Non-functional Requirements for Machine Learning: Understanding Current Use and Challenges in Industry

Khan Mohammad Habibullah, Jennifer Horkoff
Chalmers | University of Gothenburg, Sweden
{khan.mohammad.habibullah, jennifer.horkoff}@gu.se

Abstract—Machine Learning (ML) is an application of Artificial Intelligence (AI) that uses big data to produce complex predictions and decision-making systems, which would be challenging to obtain otherwise. To ensure the success of ML-enabled systems, it is essential to be aware of certain qualities of ML solutions (performance, transparency, fairness), known from a Requirement Engineering (RE) perspective as non-functional requirements (NFRs). However, when systems involve ML, NFRs for traditional software may not apply in the same ways; some NFRs may become more prominent or less important; NFRs may be defined over the ML model, data, or the entire system; and NFRs for ML may be measured differently. In this work, we aim to understand the state-of-the-art and challenges of dealing with NFRs for ML in industry. We interviewed ten engineering practitioners working with NFRs and ML. We find examples of (1) the identification and measurement of NFRs for ML, (2) identification of more and less important NFRs for ML, and (3) the challenges associated with NFRs and ML in the industry. This knowledge paints a picture of how ML-related NFRs are treated in practice and helps to guide future RE for ML efforts.

Index Terms—Non-Functional Requirements, NFRs, qualities, Machine Learning, NFR Challenges, Requirements Engineering

I. INTRODUCTION

Machine Learning (ML) is increasingly used in decision-making and prediction applications, including image recognition, language processing, and autonomous systems. ML includes algorithms that use large amounts of data to learn and automatically perform tasks that are challenging with traditional software [1]. ML-enabled software often has an influence on critical decision making (e.g., cancer detection and loan approval), and such decisions may suffer from unintended bias [2], unsafe execution [3], unexplainable decisions [4]. At the same time, ML-enabled software is becoming more complex, and exhaustive testing is expensive and time-consuming. Because of these issues, ensuring safe, fair, transparent, and accurate ML-enabled systems is challenging. From a Requirement Engineering (RE) perspective, these quality aspects are known as non-functional requirements (NFRs) [5].

For more than 40 years, work has focused on how to consider and deal with software qualities or NFRs in an effective way as part of RE and software development, e.g., [6]. Although much work has been devoted to NFRs, including exploring effective definitions of this concept [7], dealing with NFRs remains a difficult challenge in modern system development [8]. Glinz defines NFRs as "an attribute of or a constraint on a system", where attributes are performance or quality requirements [7]. In this work, we focus on NFRs as quality requirements. Despite the NFR-related challenges, progress in the area of NFR exploration has been made, including, for example, definitions (e.g., [7]), taxonomies (e.g., [9]), classification methods (e.g., [10]), modeling approaches (e.g., [5]), management methods (e.g., [11]), and industrial studies (e.g., [12]).

However, when dealing with ML-enabled software, it is not clear if our accumulated knowledge concerning NFRs is still applicable. Looking at the literature (e.g., [2]), popular science (e.g., [13]), and reading AI-related news [14], we can see that some NFRs, such as fairness, become more prominent, others, such as privacy, remain relevant, and perhaps others, such as usability, become less relevant. Further, as-yet-unexplored NFRs such as "retrainability" may also become relevant. But does knowledge gained from available literature reflect current practice? In order to begin to answer how our knowledge of NFRs should be reconsidered, we need to understand the state of practice concerning NFRs and ML in industry.

Existing work has begun to look at challenges with ML use in practice. According to a recent survey, RE is the most challenging activity for ML-enabled software development [15]. Several recent papers have identified and discussed RE-related challenges in AI-based systems [16]–[18]. However, we are unable to find work which focuses explicitly on how NFRs are dealt with in an industrial ML-context.

NFRs can be defined at over different granularity levels of a system, i.e., we can define NFRs for a system, a component, or a specific feature. Machine Learning is often only a small part of a larger system [19]. NFRs can be identified and measured over ML-related data, over the ML-model, or over the whole software which includes ML. In which of these areas do we see changes in how NFRs are identified, managed or measured? Are these distinctions in scope used in practice? Where do the NFR-related challenges lie? To begin to address these and other questions, we conducted interviews with ten practitioners who have worked with ML in industry, exploring their perceptions of NFRs in an ML context. The interview covered more or less important NFRs for ML-enabled software, how NFRs are captured and measured, and what challenges they face to

working with and measuring NFRs for ML.

Our interviewees agree that NFRs for ML are important and play a significant role in the software's overall success. They could identify many important NFRs for ML-enabled systems and some tools and techniques to measure NFRs in the ML industry. The industry practitioners also brought up challenges encountered in identifying and measuring NFRs for ML-enabled software. However, we find that currently, industry cannot offer many extensive or well-established solutions in how to identify, scope, or measure NFRs for ML.

The contributions of this paper including raising awareness of NFRs in ML amongst practitioners, bringing NFRs for ML to the forefront, and facilitating early consideration. From a research perspective, we identify a number of challenges which can direct future work in the area of NFRs for ML systems, including studying particular NFRs, considering the scope of NFRs, and identifying ML-specific NFR measurements.

The paper is structured as follows: Sec. II describes related work, while Sec. III states the research questions. We describe the research method in Sec. IV, and present our findings related to NFRs in Sec. V. Sec. VI discusses our findings and future work. We conclude our study in Sec. VII.

II. RELATED WORK

In this section, we highlight relevant related work topics, including NFRs, RE for AI, and work on SE for AI.

NFRs. Although NFRs are considered essential and critical for software success, there is no agreed upon guidelines on when and how NFRs should be elicited, documented, and validated [7]. Even so, as highlighted in Sec. I, much work in an out of RE has been devoted to NFRs, so much so that it is difficult to give a brief and complete overview.

While many papers looked into the NFRs for traditional software, a few studies focused on NFRs for ML-enabled systems in industry. Doerr et al. presented the application of a systematic, experience-based method to elicit, document, and analyze non-functional requirements. Their objective was to achieve a sufficient set of measurable and traceable non-functional requirements [20]. Ameller et al. conducted an interview-based survey with 18 different companies from six European countries. They presented the barriers to and benefits of the management of NFRs for companies, how NFRs are supported by Model-Driven Development (MDD), and which strategies are followed if some NFRs are not supported by MDD approaches. Their results show that MDD adaptation is a complex process with little or no support for NFRs, and productivity and maintainability should be supported when MDD is adopted [21]. Sachdeva et al. conducted an industrial case study over nine months and proposed a novel approach to handling performance NFRs and security for big data and cloud-related projects using Scrum. The results show that their approach helps deal with performance and security requirements both individually and accounting for conflicts between them in an agile methodology [22]. Although relevant, this body of work has mainly focused on NFRs for traditional software systems (without the use of ML).

RE for AI. While there are many approaches using ML to improve RE tasks such as model extraction [23] or prioritization [24], with much of such work reported in the AIRE Workshop Series [25], there is not as much work looking in the other direction, on RE for ML systems [18]. However, recent papers point out challenges and issues looking at RE for AI-based systems. Vogelsang & Borg have pointed out that the ML-based development process becomes more complex, with the need to effectively use large quantities of data, as well as a dependence on other quality requirements (NFRs) [18]. Belani et al. identified, discussed, and tackled the challenges for requirement engineering disciplines in developing ML and AI-based complex systems [16]. They reported that one of the challenges in ML-enabled software development is to identify NFRs throughout the software lifecycle, not only in the initial phases dealing with requirements, but as part of the whole lifecycle. Heyn et al. used three real use cases of distributed deep learning to describe system engineering challenges relating to requirements engineering [17]. They specifically focus on challenges concerning AI context, defining data quality attributes, testing/monitoring/reporting, and human factors.

Other papers begin to offer solutions, for example, Rahimi et al. introduced a RE-focused method using domain-specific concepts to find dataset gaps for safety-critical ML components [26]. Further work looking at requirements for AI focuses on specific types of requirements, such as transparency (e.g., [27]) or legal requirements (e.g., [28]). A recent workshop (RE4AI@REFSQ) has begun to explore RE for AI, but thus far, no papers have focused on the broad state of NFRs in AI industry. Nakamichi et al. focused on quality characteristics and measurement methods related to functional correctness and maturity of ML software systems (MLS). They extended the quality characteristics of conventional software defined by ISO 25010 to those unique to MLS, defining a quality measurement method. They defined a method to identify requirements to derive the quality characteristics and measurement methods for MLS [29]. Further work has focused on outlining the challenges of NFRs for ML-based solutions, including an outline of research directions. In contrast to work in this section, we focus on a wider perception of NFRs for ML in industry [30].

SE for AI. Work has looked more broadly at how SE knowledge can be applied to AI and ML system development. Previous work in collaboration with companies has identified software engineering challenges of deep learning [31]. The research used seven ML projects and identified twelve challenges categorized into three areas: development challenges, production challenges, and organizational challenges. An empirical investigation on a taxonomy of SE challenges for ML systems has been presented and discussed in [32]. Challenges include problem formulation and specifying the desired outcome, use of a non-representative dataset, lack of well-established ground truth, no deployment mechanism, and difficulties in building a highly scalable ML pipeline.

Siebert et al. presented a quality model (i.e., quality aspects and evaluation objects) for software-intensive systems contain-

14

ing one or more components that use ML in an industrial use case [33]. Martínez-Fernández et al. conducted a systematic mapping study to collect and analyze current state-of-the-art knowledge about Software Engineering for AI-based systems. They found that the most studied properties of AI-based systems are safety and dependability [34].

Further work on SE for AI can be found in workshops such a WAIN (Workshop on AI Engineering); however, most papers found in general SE venues do not focus specifically on existing requirements challenges, and we are unable to locate papers which provide a broad, industrial view of NFRs.

NFR Scope. Work by Sculley et al. focused on hidden technical debt in ML, but also pointed out that ML makes up only a small part of the system [19]. The ML component may be surrounded by code focusing on configuration, data collection, feature extraction, analysis tools, or monitoring, as well as glue code to make everything work together. Further, as emphasized by Vogelsang & Borg, RE for ML should focus not only on requirements for the system, but on requirements over the data [18]. These consideration raises the question of scope in our investigation. To simplify, in this work we focus on three possible scopes: NFRs over the ML model; NFRs over the whole system, including all the additional surrounding software as described by Sculley et al.; and NFRs for the ML-related data, as highlighted by Vogelsang & Borg.

Overall, although previous work has pointed out that handling NFRs in the development of ML systems is difficult, little work focuses specifically on NFRs, or tries to understand the state-of-the-art in handling and treating NFRs in industry.

III. Research Questions

To guide our study, we introduce a number of research questions. Our overarching research question is as follows:

RQ1. What is the perception and current treatment of NFRs in ML in industry?

We refine this RQ1 into more detail as follows:

RQ1.1. Which ML-related NFRs are more or less important in industry?
RQ1.2. Over what aspects of the system are NFRs defined?
RQ1.3. What NFR and ML-related challenges are perceived?
RQ1.4. How are NFRs related to ML currently measured?
RQ1.5. Over what aspects of the system are NFRs measured?
RQ1.6. How are NFRs and their measurements captured in practice?
RQ1.7. What measurement-related challenges for NFRs in an ML-context exist?

With RQ1.1, we aim to understand if the emphasis on certain NFRs in the literature corresponds with interest in reality. RQ1.2 is inspired by our scoping question, are NFRs defined over the ML model, the whole system, or the data? RQ1.3 aims to collect general challenges in this area. In the final four RQs, we aim to understand if and how NFRs for ML are measured, over what part of the system they are measured, how these measurements are captured, and what challenges exist in the area of NFR for ML measurement.

IV. Methodology

We choose a qualitative interview study to answer our research questions and explore experiences and perceptions in the context of NFRs for ML, following the ACM SIGSOFT Empirical Standard (tinyurl.com/QualitativeSurveys).

Sampling. The goal of the sampling was to find interviewees who had experience with ML, and who were currently working with ML in industry. The sampling method was a mix of convenience, purposive, and snowball sampling. We sent open calls to our colleagues and at industry events to find those with industrial ML experiences, then asked interviewees if they knew of further qualified people we could contact. In the end, we interviewed 10 engineers and researchers in different sectors who have experience working with ML in industry. We believe the interviewees we selected are representative of those working in the data science and ML field, including their knowledge (or lack thereof) of NFRs. The interviews lasted 30 to 35 minutes, and were conducted online between December and February 2021 via Zoom. We recorded all the interview sessions with the permission of interviewees. All interviews were transcribed, and anonymized for further analysis.

Data collection. We used semi-structured interviews, with a set of predetermined open-ended questions, so that there remained enough freedom to add follow-up questions to collect in-depth information. The interview guide can be found in I. The interview started with describing the background of the study and the research gap to help interviewees to gain a sense of their role and make them comfortable with the context of the interview. The questions were divided into three categories. In the first set of questions (Questions 1-5), we collected interviewees' demographic information as well as their experience working with NFRs. In the second set of questions, questions 6 and 7 gather the interviewees' general impressions of NFRs, and if there are differences between NFRs for ML or traditional software. These questions were meant to act as an initial check or filter: if the participants did not believe NFRs were important, or that ML brought specific differences, the rest of the interview may not provide fruitful results. We then asked about NFR-related questions in an ML context, NFR measurements in an ML context, with a final open question for more input. For identifying and defining NFRs for ML-enabled software, we relied on the interviewees' own definition of NFRs, and reporting on their perceptions.

In some cases, participants were not explicitly familiar with the term NFR. We believe this is not uncommon for those working with ML in industry. In these cases we showed them an example hierarchy of NFRs, using McCall's quality hierarchy as an example [6]. They were then able to recognize and talk about software quality aspects. We discuss this as part of our consideration of validity threats in Sec. VI.

Pre-Testing. To improve the validity and reliability of the interview process, we conducted test interviews with two Ph.D. students working with NFRs and ML. This procedure helped to remove ambiguous and redundant questions, revise unclear wordings and rearrange the questions.

15

TABLE I
INTERVIEW QUESTIONS MAPPING TO RESEARCH QUESTIONS

Interview Questions	Research Questions
Background of Interviewee (Demographic Data)	
1. Please introduce yourself and your role in this company/ organization.	N/A
2. Do you consider yourself more of academic person or industry related person?	N/A
3. Total years of experience in the industry and how long you are in current position?	N/A
4. Please describe your responsibilities (e.g., Product owner, developer, Software Architect) in your organization.	N/A
5. Please describe your experience working with NFRs.	N/A
NFR Related Questions	
6. Do you think NFRs play an important role for the success of a software? If yes, how?	RQ1.1
7. Do you think there are differences in NFRs between generic software (without ML) and ML-enabled software? If so, what? If not, why not?	RQ1.1
8. Do you think there are NFRs which are more prominent or important in ML context? If so, which ones?	RQ1.1
9. Do you think there are some NFRs less important in ML context which were prominent in generic software engineering?	RQ1.1
10. Do you think of NFRs for the whole system, for the ML model, for the data or other parts?	RQ1.2
11. What challenges do you experience with NFRs for ML?	RQ1.3
NFR Measurement Questions	
12. Do you measure NFRs over ML-enabled software?	RQ1.4
13. Of the NFRs you mentioned, how do you measure these NFRs in an ML context?	RQ1.4
14. Are these NFRs measured over the whole system, the ML implementation or some part of data?	RQ1.2
15. How do you capture NFRs and their measurement for ML-enabled systems?	RQ1.5
16. What are the challenges you face measuring NFRs for ML?	RQ1.6
17. Do you have anything else you would like to add?	any

Data Analysis. The collected data were qualitative; therefore, we used thematic analysis as a data analysis method [35]. We used a mixed form of coding, where we started with a number of high-level codes based on our RQs, then refined and adapted these codes when going through the transcripts [36]. Two authors started to code each transcribed interview separately and afterward reviewed and validated the codes for each interview with each other. We did this for the first five interviews and discussed the results and findings in several iterative meetings, reaching a good level of agreement. Then the first author coded the remainder of the transcribed interviews. We then combined data from all transcriptions into summary tables and figures, working together to find high-level categories for our codes. We made an effort to maintain the original terminology of the participants in developing our codes, e.g., we merged similar NFRs only a few clear cases.

As an example, the statement by P10,

"In terms of explainability, fairness, and other metrics, quality attributes, of course, it's a very important part of making any software as a service better."

is coded as ImportantNFR, as the interviewee discusses a number of important NFRs, and with more specific codes of NFRCorrectnessAccuracy, NFRExplainability, and NFRFairness, capturing the specific NFRs which arose.

Interview transcripts and codes can be found here https://tinyurl.com/6d96vesb.

V. RESULTS

In this section, we provide our findings from the interviews. We summarize our qualitative codes to provide an overview of NFRs in an ML context, then summarize interview demographics, examine the reported importance of NFRs, the perceived challenges related to NFRs in ML, and measurement-related findings for NFRs in ML.

A. Overview: Thematic Codes

In this section, we provide an overview of our thematic codes as derived from the interviews. A graphical overview can be found in Fig. 1. Five high-level codes were identified, subdivided into different codes, e.g., Challenges is a high-level code divided into three sub-codes: NFRChallenges, GeneralMLChallenges and NFRMeasurementChallenges.

The high-level code DefinedOver maps the interview comments that state which part of the system NFRs is defined over, subcategorized into: DefinedOverData- includes statements where interviewees said NFRs are defined over data, DefinedOverML- the interviewees said NFRs are defined over ML model, DefinedOverWhole- the interviewees said NFRs are defined over the whole software. For example, P4 said:

"To be honest, I just see the non-functional requirements just for the machine learning system."

We mapped this comment with DefinedOver and DefinedOverML as this statement describes in which part of the system the NFRs should be defined.

Similar to DefineOver, MeasurementOver mapped the statements that include comments on which part of the system NFRs are measured and categorized into MeasurementOverData, MeasurementOverML, MeasurementOverWhole.

The statements on more and less important NFRs for ML are coded as ImportantNFR and LessImportantNFR, respectively. Statements which include the name of specific NFRs for ML are coded with their name, for example, Safety, performance, and efficiency are coded as NFRsafety, NFRPerformance, and NFREfficiency. The methods for NFR and measurement capture were coded as NFRCaptured and NFRMeasurementCapture, respectively.

B. Participant Demographics

We collected our interviewees' demographic information, asking them to introduce themselves, their role in their organization, their total years of experience and experience in their current position, their responsibilities, and their experience working with NFRs. The result is shown in Table II.

Our interviewees cover a wide range of domains, industries and countries working with different roles, e.g., Section leader, Data Scientist, Team manager, Consultant. 6/10 of the interviewees are from Sweden. The interviewees' responsibilities include conducting research, developing and implementing ML algorithms, leading the development team, etc. The interviewees' experience varies between 3 to 28 years, but

TABLE II
INTERVIEWEES' DEMOGRAPHIC INFORMATION

| Partici-pant	Country	Domain	Role	Exp. (Years)	Exp. in Position (Years)	Responsibilities	NFR Experience
P1	Sweden	Medical	Section Leader	20	1.5	Research and innovative projects on strategic level	Consider NFRs while developing patient management tool
P2	Norway	email service	Requirement Engineer and Tool Expert	11	1.5	Working with requirement engineering process, setting up tools to support process	Working with requirement engineering process but not with requirements themselves
P3	Canada	Cloud Migration	Development Manager	15	2	Leading the team of software developers	Consider NFRs when creating and designing software
P4	Sweden	ML-Consulting	Senior Machine and Data Scientist	12	2	Develop and implement algorithms for ML, co-ordinate development team	Work with NFRs
P5	Sweden	e-healthcare	Chief Data Scientist	10	3	Lead team who are working for digitization and decision support system related to hospital	We use NFRs but not in very structured way
P6	Sweden	Sustainable solutions	Competence Team Manager	15	2	Designing and writing software, leading the project	Consider NFRs while designing and developing the software
P7	Sweden	Biotechnology	Consultant	3	1.5	Product owner, architect, developer	Consider NFRs while designing and developing the software
P8	Israel	Cybersecurity	Head of Research	28	1	Take care of the research and supervise the development of the model	Consider NFRs and discuss during development of a product
P9	Sweden	Automotive	Functional safety expert	3	0.5	Researcher, function development, providing functional safety knowledge	Ensure safety related NFRs
P10	Canada	Multinational and multipurpose	Research and Technology lead	4	2	Lead a group of software engineers to make ML explainabile and accountable	Try to improve ML models considering NFRs
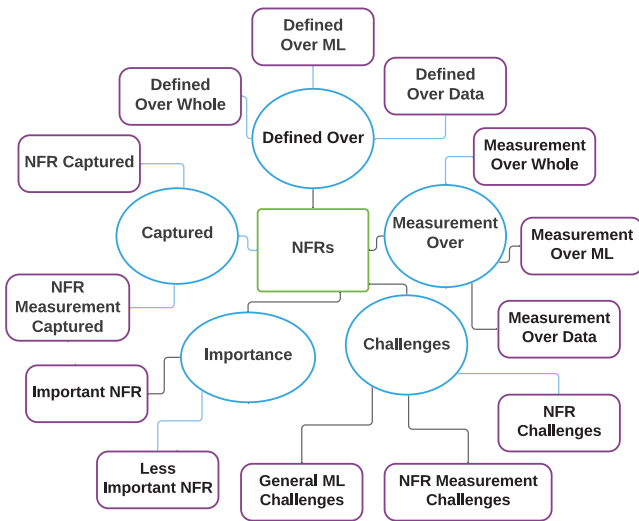
Fig. 1. Overview of Codes/Themes

most interviewees' had more than 12 years general experience. Overall, our interviewees had a leaning to more senior positions. However, on average they have only a few years experience in their current role.

We observed that the interviewees have a mix of industrial and research backgrounds, it was hard to find people to speak to who are from the only industry without academic involvement. This could be a result of our search strategy, but believe this is may be an indication of the novelty of ML applications in industry.

C. NFR Results

To start, all the interviewees indicate that NFRs play important roles in the overall success of the software and that there were differences worth considering for ML. Thus we were able to continue with the rest of our questions.

Perceived NFR Importance (RQ1.1). According to the interview codes, we identified important and less important NFRs for ML, categorizing these into the categories provided by [6]: product operation, revision, and transition. The codes related to important and less important NFRs, counts of the number of the interviewees whose interview included the code (c), and the frequency (f), a count of occurrences of the code across all transcripts, are shown in Fig. 2. We include the numbers to give an idea of frequency and ranking, however, given the small sample size, this ranking may change with more participants. We are currently working on presenting and revising the ranking with more participants via a survey.

All ten interviewees brought up important NFRs for ML-enabled software, and nine interviewees named less important NFRs. Interviewees could identify important NFRs for ML quickly compared to less important NFRs.

While talking about important NFRs for ML, P3 named a number of NFRs:

> "Repeatability, accuracy, these things are often important in ML or deep learning-based software which is not generally that much present in traditional software."

Concerning new NFRs, P4 stated:

> "Retrainability is a new non-functional requirement for the Machine Learning system. When to retrain,

how to retrain, which data use to retrain those are the requirements those you don't define in traditional software."

Considering less important NFRs for ML, P1 commented,

"Flexibility right now is not so important. If you need to scale up, you can do some changing, so we don't consider that as much important thing yet. The same with reusability. I think as AI is not so much mature yet, so we are not considering it yet".

The results illustrate that most NFRs are still considered important in an ML context, and few NFRs are considered less applicable. It is also important to note that there was a disagreement among the interviewees on which NFRs are less important. A few NFRs mentioned by some interviewees as less important are identified as important NFRs by other interviewees (colored yellow in Fig. 2). Most of the interviewees could provide answers to the related interview questions. However, not everyone could answer this question easily, and they had to be shown a standard NFR hierarchy (McCall's) [6] to illustrate possible NFRs.

> **RQ1.1: Most NFRs as defined for traditional software are still relevant in an ML context, while only a few become less prominent. Prominent NFRs according to our interviews include fairness, but also flexibility, usability, accuracy, efficiency, correctness, reliability, and testability.**

Scope of NFRs (RQ1.2). We summarize our answers and codes regarding what part of the system NFRs are defined over: the ML model, the data, or the whole system.

Out of ten interviewees, eight said NFRs are defined over the ML model. As an example quote, P4 said:

"To be honest, I just see the non-functional requirements just for the machine learning system".

Two interviewees said NFRs are defined over the data, while four participants said NFRs are defined over the whole software. As an example, P7 mentioned:

"Machine learning projects also software project. So, I guess they all match all over."

The result shows that the NFRs for ML are mostly defined over the ML model or the system as a whole. However, we see some disagreement here, and we note that this question was not so easy to answer for many participants.

> **RQ1.2 (NFRs): Most participants focused on defining NFRs over the ML model itself or the whole system, and have not explicitly considered NFRs for ML-required data.**

Reported NFR and ML-related Challenges (RQ1.3). All ten interviewees identified NFR-related challenges. These challenges are presented in Fig. 3. Leaf-level challenges include interviewee counts (c) and frequencies (f).

According to the interviewees, NFR challenges include uncertainty, domain dependence, awareness, and regulations, illustrated in Fig. 3. The interviewees also brought up specific NFRs as challenges. For example, some ML applications can be safety-critical, not entirely safe, and we can't rely blindly on NFR behavior (Safety). According to P2:

"I think that's very tricky; you can really mess with safety. I think that's why companies always afraid of using machine learning techniques over traditional system where you can really check that."

Interviewees pointed out that the transparency of a decision support system can be crucial for sensitive ML applications, and while measuring accuracy, false negatives can be dangerous. According to the interviewees, maintaining the privacy and consistency of the system can also be very challenging.

We found several challenges related to uncertainty. For ML-enabled software, it is challenging to guarantee pre-defined execution, preserve the integrity of the solutions, and there is a lack of tangible requirements. Other challenges related to domain dependence of NFRs. Some NFRs for ML depend on, for example, an Operational Design Domain (ODD), which is the specific operating domain(s) in which an automated function or system is designed to properly operate, and there can be domain-specific bias in NFR definition.

The responses also pointed out a lack of awareness of NFRs. Clients are often unaware of NFRs; therefore, they do not have expectations over NFRs. To define NFRs, special skills are needed, and the engineers and researchers lack skills in this regard. P8 said:

"Then I think that we don't have enough experience in the field (NFRs for ML) to define them well".

The interviewees also mentioned a lack of proper documentation of NFRs, which made it more challenging to define NFRs for ML. Finally, at least one interviewee reported that regulations and laws constrain definition NFRs in ML-enabled software, and that this can be challenging.

When asked about NFR-related challenges in ML, some interviewees answered with both NFR-related challenges and more general challenges with ML. Eight interviewees described challenges not specifically related to NFRs. For example, wrong training and testing data selection, complexity in data pre-processing, unexpected results over time, uncertain system behaviour, expensive and time consuming testing process, and unstructured development process.

> **RQ1.3: Our interviewees reported many NFR-related challenges. Some NFRs were pointed out as particularly challenging (safety, transparency, accuracy, consistency, privacy, completeness), but other challenges included uncertainty, domain dependence of NFRs, a lack of awareness of NFRs and regulations.**

D. NFR Measurement Results

We collected NFR measurement-related codes in the third part of the interview, and present our findings in the following.

NFR Measurements (RQ1.4). While answering the question "Do you measure NFRs over ML-enabled software?", all interviewees answered they measure or need to measure NFRs over ML-enabled software. Answers to the question, "Of the
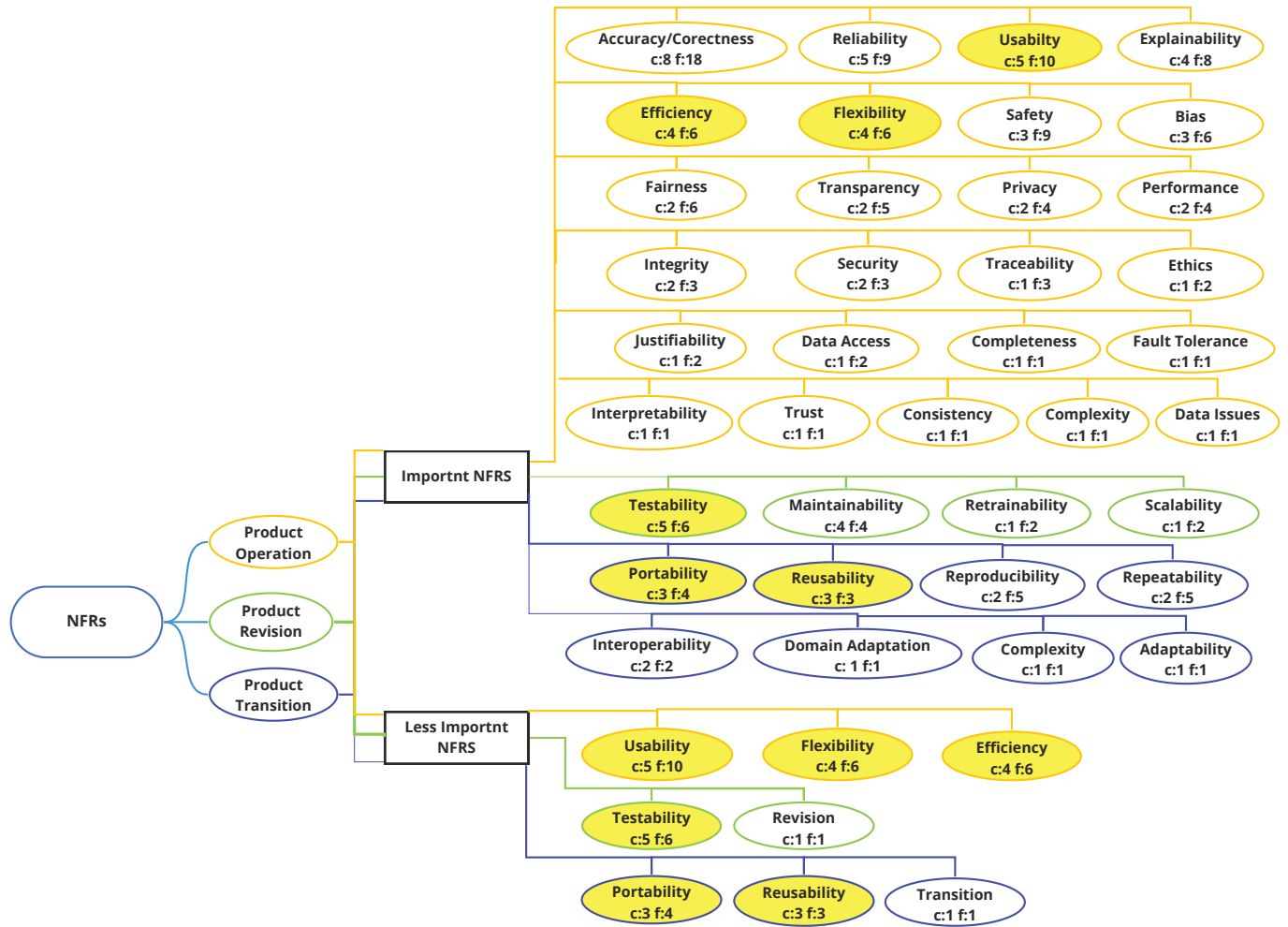
Fig. 2. Important and Less Important NFRs for ML. **c:** counts of the number of the interviewees whose interview included, **f:** count of occurrences of the code across all transcripts, **Yellow background:** NFRs mentioned by some interviewees as important are identified as less important NFRs by other interviewees.
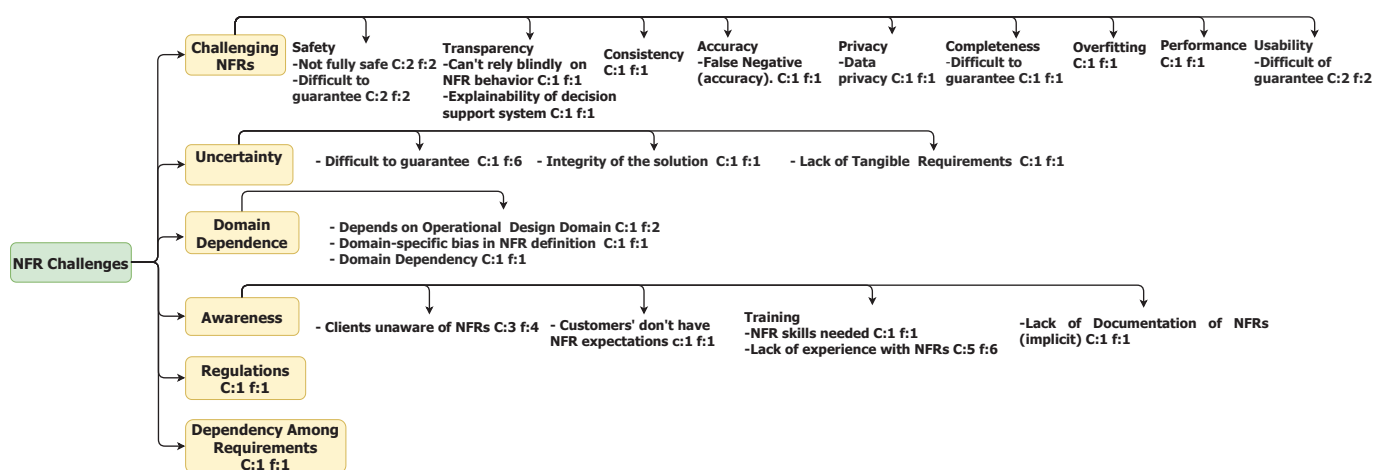

Fig. 3. NFR-Related Challenges with ML Systems.

NFRs you mentioned, how do you measure these NFRs in an ML context?" varied depending on the functionalities the software provides. For example, NFRs can be measured based on response time, statistical analysis, different performance metrics, or user feedback. According to P10:

> "Lots of NFRs (e.g., accuracy, repeatability, consistency of execution, etc.) are quantifiable, and those quantifiable NFRs can be measured by statistical analysis. For example, accuracy can be measured by accuracy matrix-like f1 score, root mean square error, etc."

Measurement can be done by machine and human judgment combined, along with statistical analysis in safety critical sensitive measurements. P1 commented:

> "If you set up a clinical trial of something, then you compare with or without machine or with a doctor's judgement with machine, then compare those and in the end if you do statistical analysis to see whether it is significant difference."

Usability can be measured by interview results (P5):

> "We do perform interviews and use the result of the system and see how they find the usability."

Additionally, usability can be measured by ad hoc methods, with some difficulty. P4 said:

> "The usability of machine learning system is a bit tricky to measure, and sometimes you have to come up with this ad hoc matrix to know about how usable the system is."

Further NFRs were also identified as challenging to measure, according to P10:

> "Measurements should be done according to standard baseline, but some measurements are not quantifiable (e.g., usability, adaptability, flexibility, etc.), therefore tricky."

RQ1.4: While some NFRs (e.g., accuracy) can be measured using ML-specific or more standard measures (e.g., precision, recall, f1 score), many are difficult to measure (e.g, fairness, explainability) as with traditional software.

NFR Measurement Scope (RQ1.5). We summarize our results concerning what part of the system NFRs are measured: the data, the ML model, or the whole system. Three interviewees said NFRs for ML are measured over data (P8):

> "Measurement is for the data. If you have labeled data for all cases, then you can measure the performance".

Six interview participants said NFRs were measured over the ML model, while four interviewees indicated measurements over the whole software. P1 explained:

> "Before you bring the system into production, you need to measure NFRs for the whole system."

Generally, 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.

RQ1.5: As with definitions, there are variances in the scope of ML-related NFR measurements, with a focus on the ML model slightly favored.

NFR Measurement Capture (RQ1.6). We asked the interviewees how NFR measurements for ML-enabled software were captured, e.g., in a tool, or via some documentation. Many interviewees had difficulties answering this question, and we discuss this further in Sec. VI. Some answered in terms of process. One respondent captures NFRs via interviews, while another mentioned use of checklists. P8 said:

> "I saw plenty of systems, and we still don't have a good enough methodology for that. Like, these are some checklists that you should go and do".

For technical means to capture measurements, engineers use different methods. For example, they implement some algorithms that run and measure the result against time. According to P6:

> "I think for this model, we should develop specific code. But we did not do it. My idea is that we have to write specific software to measure".

One participant mentioned traceability tooling as a way to measure the fulfillment of NFRs. P9 stated:

> "Well, normally we have one requirement tracing tool. So, if we have certain non-functional safety requirements, we define tests to prove that we fulfill this non-functional requirement".

In general, NFR measurement and measurement capture depend on the context. According to P1:

> "The measurement depends on their functionalities, some are time-based, and some are based on output. Sometimes measurement is captured using different tools and compared with journals in the field of healthcare".

RQ1.6: 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.

NFR Measurement Challenges (RQ1.7). Fig. 4 summarizes NFR measurement-related challenges as found via our coding process. Although many challenges could apply to both NFR challenges and NFR measurement challenges, e.g., domain dependence, the purpose is different. Here we mean the challenges specifically arise while measuring the NFRs. The first challenge concerns a missing measurement baseline and lack of a strict and effective measurement system. While measuring a certain NFR, different systems come into play. The engineers lack knowledge of measurements, and they are behind in measuring in the case of NFRs. According to P5:

> "If you compare the functional requirements, we are probably way behind when it comes to non-functional requirements. We do not have the same

NFR Measurement Related Challenges

- **Baseline**
  - -Missing measurement baseline C:1 f:1
- **Complex Ecosystem**
  - - Different systems come into play C:1 f:1
- **Lack Precise Practices**
  - - Lack of knowledge of measurements C:1 f:1
  - - Lack of strict measurement system C:1 f:1
  - - Way behind in measuring in the case of NFRs C:1 f:1
  - -Lack of effective method of measurement C:2 f:2
  - Documentation
  - - Measurements not well documented C:2 f:2
- **Lack of Knowledge**
- **Data Quality**
  - Safety
  - - Lack of plausible input C:1 f:1
  - - Over fitting of training data makes measurements invalid C:1 f:1
- **Testing**
  - - Rigorous testing is expensive. C:1 f:1
  - - Rigorous testing is not possible. C:1 f:1
- **Problematic Results**
  - - Statistical bias. C:1 f:1
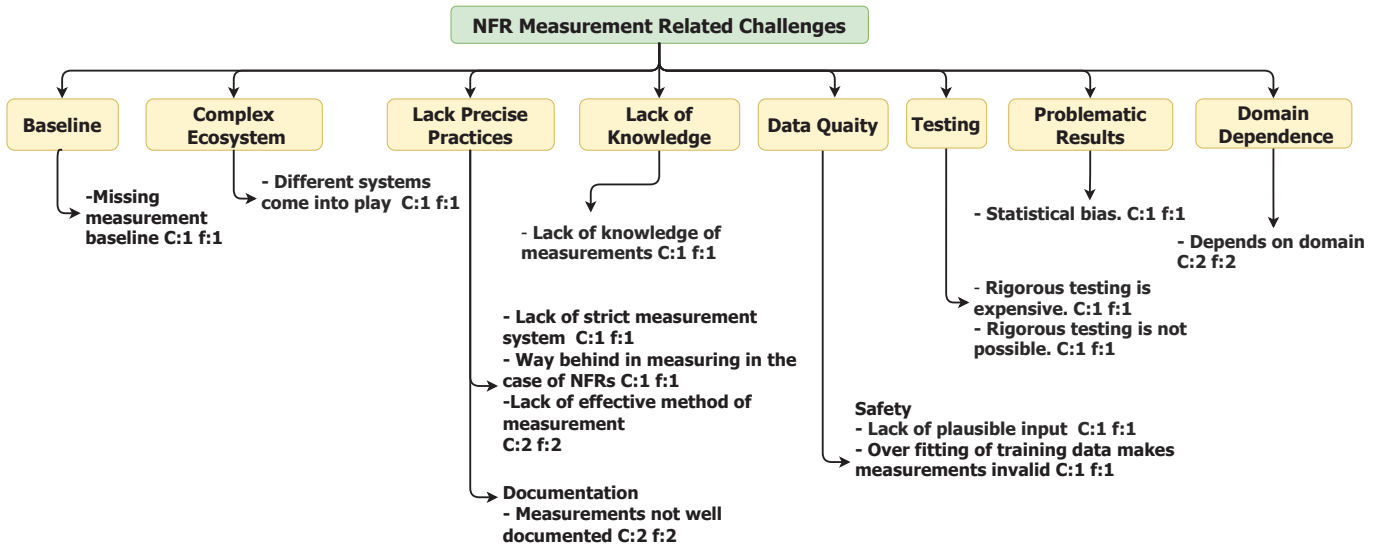- **Domain Dependence**
  - - Depends on domain C:2 f:2

Fig. 4. NFR Measurement-Related Challenges

strict system for that as we do in the functional requirements."

In one case, a lack of knowledge of measurements is a challenge (P9):

> "Not that I am aware of. I mean, testing the system of course, and based on those tests, we decide whether it is safe or not, but in Machine Learning, I am not aware of any possible measure yet".

Further challenges include that the measurement of NFRs is domain-dependent and statistical bias exists. ML systems depends on having plausible input, which can be difficult to find, and overfitting of training data makes measurements invalid. Participants also complained about the cost and plausibility of rigorous testing.

Lack of proper documentation on NFR measurements in the context of ML makes for further challenges. According to P4:

> "Sometimes it is a lack of documents that contains non-functional requirements compatible with ML-enabled systems".

> **RQ1.7: Interviewees were able to list many challenges measuring NFRs in ML-enabled systems, including a lack of knowledge, complexity, costly testing and finding data.**

VI. DISCUSSION

In this section, we discuss our interview findings. Our results show that interviewees believe that NFRs for ML play a vital role in ML-enabled software's success. We were motivated to conduct our study to understand if the increased emphasis in the literature on certain NFRs such as fairness and transparency was reflected in industrial interest. We found that our interviewees brought up many of these NFRs, including fairness, bias, and explainability. However, interviewees also discussed many other, more typical NFRs such as safety, interoperability, and retrainability. NFRs relating to performance, accuracy, and correctness were, unsurprisingly, emphasized

by our interviewees. Furthermore, some NFRs without so much emphasis in the literature made an appearance, such as retrainability, justifiablity and testability. Few NFRs were agreed to be less important. To summarize, all the old NFRs we have dealt with for years are still very important for ML-enabled systems. However, we have many further NFRs which have increased importance.

The result also indicate that both engineers and customers lack knowledge and expertise in regards to NFRs for ML. There are a lack of documentation, methods, and benchmarks to define and measure NFRs for ML-enabled software. Thus, this is a new area of exploration for our interview participants.

We have also found that there are varied answers for the scope of NFR definition in ML-enabled systems. Most participants are focusing on the ML model itself and not necessarily on the whole software, and there seems to be even less focus on NFRs over data, in contrast to recommendations found in [18] and challenges identified in [17]. The difference in results compared with [18] may be due to the difference in interviewees' profiles, with [18] focusing on data scientists, while only 2/10 of our interviewees identified as data scientists. Our results can be seen to echo the findings in [16]: although we did not ask specifically about NFRs in the software lifecycle, we found many measurement-related challenges related to system operation and testing.

From a research perspective, our findings reveal several gaps that can shape future work.

1) We need further work which focuses on those NFRs with a newly increased focus in an ML context, e.g., fairness, explainability/transparency, bias, justifiability, and testability. This includes definitions, new taxonomies, measurements, and methods. Such work has already begun for some NFRs (e.g., [37] for fairness, [27] for transparency), but it is often approached from a general SE, rather than an RE perspective.

2) Conceptualization and methods are needed to address the scope of NFRs, e.g., there can different ways to view the sub-parts of the system, and these views may affect the way

21

we categorize and define NFRs for ML [33]. We believe that many NFRs should be considered on three levels (ML model, software, data), but more work must be done to investigate such ideas.

3) Our findings have identified NFR-related challenges that can be addressed from an RE perspective by further research, e.g., the domain dependency of our NFR-ML understanding (as recently emphasized in [17]), and awareness of NFRs for ML from the client and engineer side.

4) New measurements for NFRs in an ML context are needed (e.g., work in [29]) . Many NFRs have always been difficult to measure, yet they must be measured in potentially new ways over different scopes. The rise of ML makes measuring NFRs more difficult.

5) We found further measurement-related challenges. From an RE perspective, we can apply methods to understand complex ML ecosystems, to define and refine NFRs, or to make tradeoffs between NFRs (e.g., testing costs vs. results).

From an industrial perspective, our findings provide a view of current practice, but do not yet offer concrete solutions. However, it is useful for practitioners to see the sorts of questions and challenges that others are facing, and to understand that many of their current challenges are not unique.

Overall, we see that this area is challenging for industry, yet important, and although individual companies may have some knowledge and practices, they do not have well established solutions for dealing with NFRs in an ML context.

Future Work. This work has focused more on identification and importance of certain NFRs, we did not have time in the interviews to get into specific definitions, e.g., how is usability for ML-enabled software different than for traditional software. We are currently designing a survey to validate and refine our finding with more participants. We are also working on a systematic literature review for NFRs in ML-based systems. This will allow us to more precisely compare the focus of the literature to the focus in industry. Finally, we wish to investigate NFR scope as indicated above, identifying and differentiating NFRs over different system parts, such as data, the ML model, and the whole system.

Threats to Validity. Our qualitative interview study has several threats to validity. In terms of construct validity, we noted that questions concerning how NFR measurements were captured were not easy for the interviewees to understand. They could have understood each NFR differently. In retrospect, this question should have been more clearly written, still we believe the results collected were interesting.

Several of our interviewees were not familiar with the concept of NFRs, and wanted examples. But we believe the interviewees are representative of the data science and ML field, and many working in this area may not have software engineering training. Data science engineers are working to ensure the qualities of ML-enabled software, but they don't know traditional software engineering terms. To exemplify NFRs, we showed a version of McCall's software quality hierarchy as an example [6], given its age and prominence, also used to categorize our NFR results. We could have used

a number of other available NFR hierarchies, as there are many. Reflecting on conclusion validity, showing a particular NFR hierarchy will bias the results towards that hierarchy; however, the differences between hierarchies is not extensive, and showing any one NFR hierarchy would influence the results in that direction.

Our work suffers from the typical internal validity threats associated with thematic coding. We mitigated these threats by performing independent coding over half the interviews and comparing results, finding sufficient agreement; and by using standard coding tools (NVivo) to help the process. We made the results available for further use.

In terms of the number of interviewees, we must consider whether we believe our findings were close to reaching saturation after 10 participants. We found towards the end of our analysis that the codes were generally converging to a stable set. However, the code "justifiability" was added in the last interview. An eleventh interview, conducted, but not yet coded did not reveal any new results. Thus, we believe further interviews could help to enrich our findings in a few areas, but would not produce significant additions.

Our sampling technique found a number of participants who straddle the boundaries between industry and academia. This may be a result of our circle of contacts, but 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 background. Our participants are mainly from one country, threatening external validity. However, we found participants in a diverse set of industries, and we believe Sweden has a strong and international AI-oriented industry, thus our participants are fairly representative.

VII. Conclusions

We have conducted a qualitative interview study to understand the perception of and practices for NFRs in ML-based systems in industry. The interviewees agree that NFRs for ML is an important topic. The results showed that traditional NFRs like usability can still be important for ML, as new NFRs such as transparency gain importance. The results also showed that NFRs for ML are not well-structured or well-documented. According to interviewees' industrial and research experience, it is difficult to define and measure NFRs for ML as ML-enabled systems. People are not aware of NFRs for ML-enabled software, and engineers and researchers lack knowledge and experience. Sometimes the NFRs are not tangible, which creates measurement challenges and complex, expensive testing. From an industrial perspective, NFRs for ML are not well organized and well developed and their consideration is mainly in an initial stage. The challenges and complexities of NFR-related research remain, but are intensified by ML. Much further research by the RE community is needed to overcome NFR-related challenges for ML-enabled software.

VIII. Acknowledgements

REFERENCES

[1] A. Smola and S. Vishwanathan, "Introduction to machine learning," *Cambridge University, UK*, vol. 32, no. 34, p. 2008, 2008.

[2] T. Kamishima, S. Akaho, and J. Sakuma, "Fairness-aware learning through regularization approach," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 643–650.

[3] R. Challen, J. Denny, M. Pitt, L. Gompels, T. Edwards, and K. Tsaneva-Atanasova, "Artificial intelligence, bias and clinical safety," *BMJ Quality & Safety*, vol. 28, no. 3, pp. 231–237, 2019.

[4] O. Biran and C. Cotton, "Explanation and justification in machine learning: A survey," in *IJCAI-17 workshop on explainable AI (XAI)*, vol. 8, no. 1, 2017, pp. 8–13.

[5] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering*. Springer Science & Business Media, 2012, vol. 5.

[6] J. P. Cavano and J. A. McCall, "A framework for the measurement of software quality," in *Proceedings of the software quality assurance workshop on Functional and performance issues*, 1978, pp. 133–139.

[7] M. Glinz, "On non-functional requirements," in *15th IEEE International Requirements Engineering Conference (RE 2007)*. IEEE, 2007, pp. 21–26.

[8] A. Jarzębowicz and P. Weichbroth, "A systematic literature review on implementing non-functional requirements in agile software development: Issues and facilitating practices," in *Lean and Agile Software Development*, A. Przybyłek, J. Miler, A. Poth, and A. Riel, Eds. Cham: Springer International Publishing, 2021, pp. 91–110.

[9] M. Galster and E. Bucherer, "A taxonomy for identifying and specifying non-functional requirements in service-oriented development," in *2008 IEEE Congress on Services-Part I*. IEEE, 2008, pp. 345–352.

[10] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "Automated classification of non-functional requirements," *Requirements engineering*, vol. 12, no. 2, pp. 103–120, 2007.

[11] H. Kaur, G. ASU, and A. Sharma, "Non-functional requirements research: Survey," *Int. J. Sci. Eng. Appl*, vol. 3, no. 6, 2014.

[12] J. Eckhardt, A. Vogelsang, and D. M. Fernández, "Are" non-functional" requirements really non-functional? an investigation of non-functional requirements in practice," in *Proceedings of the 38th International Conference on Software Engineering*, 2016, pp. 832–842.

[13] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, "Dissecting racial bias in an algorithm used to manage the health of populations," *Science*, vol. 366, no. 6464, pp. 447–453, 2019. [Online]. Available: https://science.sciencemag.org/content/366/6464/447

[14] R. Chandran, "Fears of 'digital dictatorship' as myanmar deploys ai," 2021. [Online]. Available: https://www.reuters.com/article/myanmar-tech-protests-idUSL8N2L90EU

[15] F. Ishikawa and N. Yoshioka, "How do engineers perceive difficulties in engineering of machine-learning systems?-questionnaire survey," in *2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)*. IEEE, 2019, pp. 2–9.

[16] H. Belani, M. Vukovic, and Ž. Car, "Requirements engineering challenges in building ai-based complex systems," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2019, pp. 252–255.

[17] H.-M. Heyn, E. Knauss, A. P. Muhammad, O. Erikssonz, J. Linder, P. Subbiah, and S. K. P. andSagar Tungal, "Requirement engineering challenges for ai-intense systems development," in *1st Workshop on AI Engineering – Software Engineering for AI (WAIN2021)*. IEEE, 2021.

[18] A. Vogelsang and M. Borg, "Requirements engineering for machine learning: Perspectives from data scientists," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2019, pp. 245–251.

[19] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," *Advances in neural information processing systems*, vol. 28, pp. 2503–2511, 2015.

[20] J. Doerr, D. Kerkow, T. Koenig, T. Olsson, and T. Suzuki, "Non-functional requirements in industry-three case studies adopting an experience-based nfr method," in *13th IEEE International Conference on Requirements Engineering (RE'05)*. IEEE, 2005, pp. 373–382.

[21] D. Ameller, X. Franch, C. Gómez, S. Martínez-Fernández, J. Araújo, S. Biffl, J. Cabot, V. Cortellessa, D. Méndez, A. Moreira *et al.*, "Dealing with non-functional requirements in model-driven development: A survey," *IEEE Transactions on Software Engineering*, 2019.

[22] V. Sachdeva and L. Chung, "Handling non-functional requirements for big data and iot projects in scrum," in *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*. IEEE, 2017, pp. 216–221.

[23] C. Arora, M. Sabetzadeh, S. Nejati, and L. Briand, "An active learning approach for improving the accuracy of automated domain model extraction," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 28, no. 1, pp. 1–34, 2019.

[24] A. Perini, A. Susi, and P. Avesani, "A machine learning approach to software requirements prioritization," *IEEE Transactions on Software Engineering*, vol. 39, no. 4, pp. 445–461, 2012.

[25] F. Dalpiaz and N. Niu, "Requirements engineering in the days of artificial intelligence," *IEEE Software*, vol. 37, no. 4, pp. 7–10, 2020.

[26] M. Rahimi, J. L. Guo, S. Kokaly, and M. Chechik, "Toward requirements specification for machine-learned components," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2019, pp. 241–244.

[27] H. Felzmann, E. F. Villaronga, C. Lutz, and A. Tamò-Larrieux, "Transparency you can trust: Transparency requirements for artificial intelligence between legal norms and contextual concerns," *Big Data & Society*, vol. 6, no. 1, p. 2053951719860542, 2019.

[28] A. Bibal, M. Lognoul, A. de Streel, and B. Frénay, "Legal requirements on explainability in machine learning," *Artificial Intelligence and Law*, pp. 1–21, 2020.

[29] K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, M. Aoyama, L. Joeckel, J. Siebert, and J. Heidrich, "Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 260–270.

[30] J. Horkoff, "Non-functional requirements for machine learning: Challenges and new directions," in *2019 IEEE 27th International Requirements Engineering Conference (RE)*. IEEE, 2019, pp. 386–391.

[31] A. Arpteg, B. Brinne, L. Crnkovic-Friis, and J. Bosch, "Software engineering challenges of deep learning," in *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2018, pp. 50–59.

[32] L. E. Lwakatare, A. Raj, J. Bosch, H. H. Olsson, and I. Crnkovic, "A taxonomy of software engineering challenges for machine learning systems: An empirical investigation," in *International Conference on Agile Software Development*. Springer, Cham, 2019, pp. 227–243.

[33] J. Siebert, L. Joeckel, J. Heidrich, K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, and M. Aoyama, "Towards guidelines for assessing qualities of machine learning systems," in *International Conference on the Quality of Information and Communications Technology*. Springer, 2020, pp. 17–31.

[34] S. Martínez-Fernández, J. Bogner, X. Franch, M. Oriol, J. Siebert, A. Trendowicz, A. M. Vollmer, and S. Wagner, "Software engineering for ai-based systems: A survey," *arXiv preprint arXiv:2105.01984*, 2021.

[35] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[36] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[37] Y. Brun and A. Meliou, "Software fairness," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 754–759.