

Documentation of Non-Functional Requirements for Systems with Machine Learning Components

Elma Bajraktari¹, Thomas Krause² and Christian Kücherer³

¹ adesso SE, Stockholmer Platz 1, 70173 Stuttgart, Germany

² Serapion GmbH, Schäufeleinstr. 7, 80687 München, Germany

³ Reutlingen University, 72762 Reutlingen, Germany

Abstract

[Context and motivation] Many of today's systems use artificial intelligence, where Machine learning (ML) is a subfield. Requirements engineering (RE) addresses the needs of the stakeholders for systems development. In particular, systems with ML components require specific non-functional requirements (NFRs) to define ML relevant details, such as quality aspects of training datasets, retrainability of ML-models or specifics of the ML training pipeline. [Problem] The specific application of RE techniques in practical use to systems with ML components is not yet completely understood. It is not clear, which techniques for elicitation, documentation of requirements can be used efficiently for ML based systems. [Ideas and results] Based on a systematic mapping study; we identify 58 NFRs used in studies to describe particular ML requirements. Through an online survey and expert interviews, we identified 30 NFRs that need to be considered in particular for systems with ML components. For the documentation of the highly relevant NFRs, a template was designed, evaluated and optimized in two IT companies. This template helps to ensure consistent documentation of the NFRs. [Contribution] Based on the systematic mapping study, the online survey and the expert interviews, we provide a list of relevant NFRs and a template for documenting the NFRs for systems with ML components. We validated the proposed template using a real world case in the context of two IT industry companies and several software projects. The evaluation shows an increased completeness of requirements.

Keywords


Requirements elicitation and documentation, machine learning, non-functional requirements

1. Introduction

Requirements Engineering (RE) is a process used in the development of software-based systems to address the needs of stakeholders. During the RE process, requirements are elicited, documented, validated, and managed. A requirement is a statement that reflects the needs of the stakeholder, such as the capabilities or characteristics that the software to be developed must have [1, 2]. A distinction is made between functional and non-functional requirements (NFRs). Functional requirements describe functionalities that must be provided by a system. NFRs are understood to be quality requirements on the one hand and constraints on the other [1, 3].

In: D. Mendez, A. Moreira, J. Horkoff, T. Weyer, M. Daneva, M. Unterkalmsteiner, S. Bühne, J. Hehn, B. Penzenstadler, N. Condori-Fernández, O. Dieste, R. Guizzardi, K. M. Habibullah, A. Perini, A. Susi, S. Abualhaija, C. Arora, D. Dell'Anna, A. Ferrari, S. Ghanavati, F. Dalpiaz, J. Steghöfer, A. Rachmann, J. Gulden, A. Müller, M. Beck, D. Birkmeier, A. Herrmann, P. Mennig, K. Schneider. Joint Proceedings of REFSQ-2024 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2024. Winterthur, Switzerland, April 8, 2024.

✉ elma.bajraktari@adesso.de (E. Bajraktari); thomas.krause@serapion.net (T. Krause); christian.kuecherer@reutlingen-university.de (C. Kücherer);

 0009-0002-2340-502X (T. Krause) 0000-0001-5608-482X (C. Kücherer)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

In this paper, only NFRs are considered. Quality requirements describe quality criteria that the system must meet, such as performance and availability, and constraints describe conditions and restrictions that can influence the development of the system, such as laws, regulations or standards [1, 3]. According to Ahmad et al. [3], RE has been sufficiently explored in the development of classical software-based systems, and it is clear which techniques can be used in RE activities. Recent software developments are increasingly using artificial intelligence (AI) components and machine learning (ML) [3]. According to Kreutzer and Sirrenberg [4] AI is understood as an overarching term for systems that automatically generate intelligent solutions to a problem [4]. ML is understood as a subfield of AI in which self-learning algorithms are used. These systems are able to learn and solve problems on their own without a human being programming them to do so [4, 5]. The difference between classical software-based systems and systems with ML components is that systems with ML components are probabilistic [6]. According to Ahmad et al. [3], the use of RE in the development of systems with AI or ML components has not been sufficiently explored. This statement is confirmed by Yoshioka et al. [6] and Villamizar et al. [7] in that research regarding techniques of RE in the development of ML-based systems is insufficient. According to Ahmad et al. [3], new techniques are needed for RE activities that can be used in the development of systems with AI or ML components. According to Rupp and the SOPHISTS [2], the specification of ML-based systems differs from classical systems in that this type of system focuses on the quality of the ML model used as well as the training, validation, and test data. Greater emphasis must be placed on NFRs in ML-based systems [2]. A ML model is built using a selected ML algorithm trained on training data for solving a particular problem [4, 5, 8]. Validation and test data are then used to check the quality of the ML model before it is deployed [9]. The quality aspects of NFRs that must be taken into account when developing classic software-based systems are defined and categorized in ISO/IEC 25010 [10]. This standard does not yet explicitly consider quality criteria related to systems with ML components. Quality criteria should not only refer to the end product, but also to the previous aspects such as the training data used, or the ML model used.

Summarizing this, there is a need for specific ML-related requirements documentation techniques. This is also confirmed by the employees of adesso SE. Due to the lack of best practices in this area, it is still unclear which documentation techniques are specifically suitable for systems with ML components. With over 9,000 employees and annual sales of EUR 900.3 million in 2022, adesso Group is one of Germany's largest IT service providers. We aim to provide a template, focusing requirements analysts to ML specifics and improve the completeness of requirements by specifying ML quality requirements. This approach promises to improve project success and to help on save development costs. The primary goal of this paper is twofold: (1) to identify the NFRs of systems with ML components that need to be specified in greater detail when applying ML and (2) to conceptualize a template for the documentation of the most relevant NFRs. This paper will answer the following research questions (RQ):

- RQ1: How are requirements for systems with ML components documented?
- RQ2: Which NFRs are specific to systems with ML components?
- RQ3: Which documentation styles for systems with ML components are most used in industrial projects at adesso SE?
- RQ4: Which problems do current elicitation and documentation techniques have for ML components in industrial projects at adesso SE?
- RQ5: What is the template design to document NFRs for ML systems?

2. Related Works

The following previous works are relevant: The systematic mapping study of Villamizar et al. [7] covers RE for ML-based systems. They provide an overview of 35 studies (2018-2020) starting from the results of a Scopus search and snowballing. They investigated specific ML based systems NFRs, based on their frequency in the primary studies, given in brackets hereinafter. The ML specific requirements are Usability (1), Scalability (1), Modularity (1), Robustness (1), Autonomy (1), Uncertainty (1), Suitability (1), Accuracy (2), Ethics (2), Accountability (2), Testability (2), Legal requirements (2), Maintainability (3), Performance (3), Safety (4), Reliability (4), Transparency (5), Fairness (5), Data quality (5), Privacy (6), Explainability (6) and Security (6). The systematic literature review (SLR) of Yoshioka et al. [6] covers 32 papers (2017-2021), investigating which techniques are used to document ML system requirements with concrete examples. They found GORE (i* and KAOS) in 10 cases, UML in 7 and safety cases in 1 case. The ML specific requirements are Overfitting (2), Fairness (2), dataset requirements (6), Robustness (6), Accuracy (7), Explainability, Transparency and Accountability (9).

The literature review from Ahmad et al. [3] covers RE for AI and ML based systems. They analyzed 27 studies (2011-2020). The authors focused on the documentation techniques of requirements for systems with AI or ML components with an emphasis on their specific NFRs. First of all, the GORE notations FLAGS, CORE, GRL, i* und GORE-MLOps are used (5), and with the same frequency UML and SysML notations (5) and Conceptual Models (CM). Further they summarized ML relevant NFRs: Transparency, Trust, Privacy, Safety, Reliability, Security, Fairness, Explainability, Ethics, Robustness, Accuracy, Uncertainty, Data quality, Testability, Legal requirements und Availability of training-, validation- and test-data.

The literature review of Gjorgjevikj et al. [54] shows an interesting mixed-method study on the use of requirements for ML in projects. They validate that RE activities are crucial to the ML development process. Requirements should cover quality ML specifics, which are Interpretability, Fairness, Robustness, Security, Privacy, and Safety, that occurs also in our research. Most importantly, they state future research should focus on adjusting the RE activities to fit the ML development. The template described in this study, provides further research to this direction. The existing SLRs show documentation forms for ML components (RQ1) and specific ML components NFRs (RQ2) till 2021. We complete this view of related works to June 2022.

3. State of the Art

In this article, we perform a systematic mapping study according to the principles of Petersen [53] to provide an overview of the state-of-the-art regarding use of NFR for ML systems, answering RQ1 and RQ2. The mapping study, particularly the data gathering, was performed by the first and reviewed by the third author. We used principles of Kitchenham und Charters [11] for study selection and search term construction. Data was gathered in June 2022.

3.1. Method of Literature Review

As RQ1 and RQ2 are closely related, we use one search term for the literature acquisition: `((("requirements engineering") AND (documentation OR specification OR notation OR "modeling language") AND "machine learning" AND (software OR application OR system)))`. For a broad search and a high level of completeness around machine learning, the

search for additional ML sub-terms was omitted. We used the following databases: (i) SpringerLink¹ as they have broad basis on RE and AI, (ii) ScienceDirect² focuses on engineering of AI systems and architectures, (iii) IEEE Xplore³ and (iv) ACM Digital Library⁴ are engineering databases for studies with emphasis on AI applications. Inclusion (I_n) and exclusion (E_n) criteria are shown in **Table 1**. $I1$ selects studies after the SLRs in related works have been published. $I2$ assures studies to have minimal scientific standard, whilst the selected databases list peer reviewed articles only. $I3$ includes papers that contribute to the topic of this article. A paper was selected if a review showed it to offer a contribution to documentation techniques or NFRs for systems with ML components. $E1$ filters studies that use ML techniques to support RE: If the paper's main focus was on approaches to support requirements activities by AI or ML, the paper was excluded, as this was not our scope. This criterion was validated by a detailed review of the article. $E2$ avoids SLRs or mapping studies, that we already addressed as related works. $E3$ de-selects similar studies from the same authorship.

Table 1. In- and exclusion criteria

ID	Criterion
I1	publication date Sept. 2021 to June 2022
I2	English language, peer reviewed
I3	contributes to RQ1 or RQ2
E1	Literature focusing on ML for RE
E2	systematic literature reviews or systematic mapping studies
E3	duplicates

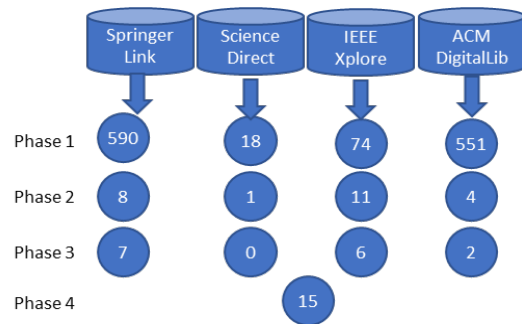


Fig. 1. Study selection chart

Study Selection. **Fig. 1.** shows the four phases of study selection. In *phase 1* we queried the selected databases with the presented search term, applied $I1$ and $I2$, resulting in 1233 publications. In *phase 2* the title, keywords and abstracts were considered using $I3$ and $E1$. In *phase 3* we reviewed the content and filtered according $I3$ and $E2$. As there were no duplicates $E3$, we summed up to 15 relevant publications shown in **Table 3**. No relevant publication could be identified from ScienceDirect. The contents of the listed publications did not cover documentation techniques or NFRs for systems with ML components. The references of the literature from phase 2 onwards are available for download in the last section.

3.2. Result of the Literature Review

Documentation Techniques of Systems with ML Components.

Four out of 15 selected studies cover techniques for requirements documentation. The other 11 studies cover NFRs of systems with ML components. Zaidi [12] shows the use of conceptual models (CM) in various phases of ML to document requirements and goals of the project. The use i^* and UML, Business Process Model and Notation (BPMN) [13] and Building Information Modeling (BIM). The latter is a domain specific notation model for building [14]. Tun et al. [15] found several requirements documentation notations: An *AI Project Canvas* is used to document decision making considerations and to capture the impact on organizational structure due to

¹ <https://link.springer.com/>

² <https://www.sciencedirect.com/>

³ <https://ieeexplore.ieee.org/Xplore/home.jsp>

⁴ <https://dl.acm.org/>

the ML components. *Safety Cases* and *System-Theoretic Accident Model and Processes* (STAMP) are used for safety analysis, modelling possible accidents and giving evidence for the systems suitability [16]. The architectural design of the system is described with SysML [17] and the functional requirements with a Goal Model (GORE) notation.

Khan et al. [18] investigates the use of NFRs for ML systems. They propose an extended SysML requirements diagram, and a GORE-MLOps model, that addresses uncertainty and unpredictability in ML systems during RE. Husen et al. [19] proposes a framework for safety-critical ML systems consisting of the documentation techniques AI project canvas, ML Canvas, KAOS, UML Component Diagram, STAMP/Systems Theoretic Process Analysis (STPA) and Safety Cases. The AI project canvas details early business requirements from the system specification. KAOS is used to document functional requirements that are detailed into an architectural component model. By using safety cases counter measures for risks through STAMP/STPA are defined, whereas STAMP is a method for risk [19, 20]. **Table 2** summarizes these techniques.

Table 2. Identified documentation techniques for ML requirements

Documentat./Study	Zaidi [12]	Tun et al. [15]	Khan et al. [18]	Husen et al.[19]
UML, SysML	CM	X	SysML Req.diagr.	X
BPMN	CM			
BIM	CM			
AI/ML Project Canvas		X		X
Safety Cases		X		X
Goal Model	i*	X	GORE-MLOps	KAOS
STAMP		X		X

In summary, GORE notations and UML or SysML diagrams are mostly used for ML requirements. Three studies provide details about GORE: i* [12], KAOS [19] and GORE-MLOps [18]. All four works use UML/SysML diagrams: in two papers this is detailed to component diagrams [19] and extended SysML requirements diagrams [18]. For the documentation of NFRs GORE [15], GORE-MLOps and an extension of a SysML requirements diagrams [18] is mentioned.

NFRs of Systems with ML Components

Within 14 of the 15 selected studies, we identified in summary 54 NFRs for ML components as shown in the extraction table available for download as open research data (see last section). The related works showed four more NFRs that were not included in the 14 selected studies. We have added the additional four NFRs to the results: (i) Autonomy of ML algorithm, (ii) Suitability, (iii) Legal requirements and (iv) dataset requirements. As there is no common set of ML specific NFRs yet, we identified NFRs with synonymous terms or describing similar phenomena. Therefore, we consolidated the 58 identified NFRs into 33 relevant NFRs with the following descriptions. An overview of these results is given in **Table 3**.

NFR-1 Suitability describes the appropriateness of the ML usage and the extent to which the ML solves the given problem [7, 21]. **NFR-2 Explainability and Interpretability** describes mechanisms that allows users to comprehend the systems results [22–25]. **NFR-3 Justifiability** Users require system’s decision to be rational [26]. **NFR-4 Transparency** describes details about ML algorithms, training- and test data to validate the systems decisions [22, 24, 26, 27]. **NFR-5 Traceability** For users and developers the source of training and validation data, artifacts and processes must be documented [22, 28]. **NFR-6 Fairness** describes that systems must

respect human rights, equal rights, equal opportunity for all users, and follow the democracy principles. The results of AI are supposed to be un-biased and discrimination free [22, 25, 29, 30]. **NFR-7 Safety** describes requirements that avoids risks to humans or the environment [23]. **NFR-8 Trust** describes users expectation to the reliability of system decisions and the correctness of results [10, 24]. **NFR-9 Efficiency of ML algorithm** are used to describes quality aspects of the ML training and prediction algorithms [26]. **NFR-10 Performance of ML model** describes the expectations to performance and correctness of the ML model for prediction and the resources to gain these predictions [22]. This incorporates *accuracy* as internal performance metric for the prediction quality [23, 31, 32] and *correctness* of the decisions or prediction results [22, 23]. **NFR-11 Latency of ML model** defines the acceptable time between data acquisition and the result of the prediction [33]. **NFR-12 Security** defines the access to used and created data (sets) [22, 23]. **NFR-13 Privacy** considers rights of human privacy due to confidentiality and protection of data [22, 23]. **NFR-14 Integrity of data** defines quality attributes to training, validation and test data, e.g. correctness, preprocessing and data integrity [22, 23, 27]. **NFR-15 Dataset requirements** capture requirements to data records such as topic, domain, context, origin of data [34], quantity of data [9, 27]. **NFR-16 Accountability** describes details to the extent to which the system takes ownership and responsibility of its decisions [26, 35]. **NFR-17 Reliability** captures the predicted functioning expectations for the system [23, 25]. **NFR-18 Reproducibility and Repeatability** address the necessity that predictions must be identical for multiple requests with the same data [23, 26, 36]. This is synonymously referred to as *consistency* [37]. **NFR-19 Fault tolerance** describes expectation to resilience to incorrect data or partial system failures to avoid complete failure [38], sometimes referred to as *robustness* [22, 39]. **NFR-20 Autonomy of ML algorithm** expresses the independence or encapsulation of ML algorithms. The retraining of a ML subsystem must be possible without depending on the application's development [4, 7]. **NFR-21 Maintainability** describes the needs regarding further development as extensions or system evolutions [10, 40]. **NFR-22 Modularity** defines the required level of maintainability. A system consist of several modules that work independent but collectively to provide the necessary functionality [10, 40]. **NFR-23 Reusability** describes details about the possibility to reuse parts or components of the system in other contexts or systems. In ML contexts this could be the reuse of ML models or data sets [40]. This NFR also covers the *domain adaptation*, where in transfer learning scenarios labelled data are used to create models of other domains [41]. **NFR-24 Modifiability** describes the extent to which a module can be changed without affecting the ML module quality [10, 40]. **NFR-25 Retrainingability** describes that ML models must be newly trained with other than initial data [27]. **NFR-26 Testability** defines, how ML models and their prediction components can be tested and what the resulting quality of decisions can be [40]. **NFR-27 Usability** describes the same aspects as in ISO/IEC 25010 with a focus on ML models and their decision or prediction presentation to users [10, 27]. **NFR-28 Interoperability** describes the requirements that allow the communication and data exchange with other systems [10, 42]. **NFR-29 Portability** defines to what extend ML models are supposed to be transferred to other contexts, e.g. a classification of blood diseases for cancer patients to non-cancer patients [10, 32]. **NFR-30 Adaptability** (portability of ISO/IEC 25010) describes how the ML model can be used in other contexts such as operating systems or system environments [10, 27]. **NFR-31 Scalability of ML pipeline** defines requirements for processing tools with regard to data volumes, large data sets and runtime is-

sues during training [37, 43]. **NFR-32 Complexity of ML model** describes the number of features the ML model is capable of. A too high model complexity leads to overfitting and a too low complexity leads to underfitting [44]. **NFR-33 Legal requirements** covers the specification of regulatory needs such as standards, acts, laws etc. [1, 45].

In addition to these NFRs, further requirements were mentioned in the primary studies, as given below. We combined these requirements to the following NFRs: *Ethics* is part of NFR-33, *Uncertainty* is part of NFR-17, *data issues* are details of dataset requirements in NFR-15 covering cleaning of datasets, *revision and transition* is part of Maintainability NFR-21. Completeness is discussed in Habibullah und Horkoff [37] but is not clearly defined; *Flexibility* covers the flexibility of the ML pipeline and is categorized as Reusability NFR-23.

4. Handling NFRs in Systems with ML Components in Practice

An online survey (12 persons) and expert interviews (4 experts) with employees were conducted at adesso SE, to identify the special needs of the documentation of NFRs for systems with ML components. We used this case to prove the relevance of the NFRs identified by the literature review and to identify further NFRs (RQ2). Moreover, we wanted to understand the documentation techniques used by adesso SE (RQ3) and related problems of the documentation (RQ4).

4.1. Relevant NFRs of Systems with ML Components

After the analysis of the systematic mapping study, the online survey and the expert interviews, the following 30 of the 33 NFRs mentioned in **Table 3** were identified as especially relevant for ML components and provide the final answer to RQ2: The NFRs marked in blue in **Table 3** have been classified as relevant in literature research but also in the online survey and expert interviews at adesso SE. The NFRs marked in grey were additionally classified as relevant in the online survey and in the expert interviews. The online survey questions are available for download, given in the last section. Based on the online survey, the following NFRs were rated as the most relevant NFRs (all of the participants gave a positive response): *Suitability*, *Integrity of data*, *Reliability*, *Latency of ML model*, *Testability*, *Explainability* and *Interpretability*, *Performance of ML model*, *Security* and *Retrainability*. Through the expert interviews, it was determined that the NFRs *Accountability* and *Transparency* must be considered relevant, even though they were not classified as relevant by the online survey. The expert interviews showed that *Integrity of data* always appears in the top 2 of the most important NFRs. This is justified by the fact that data is the basic building block of ML and therefore its integrity is necessary. The questions of the expert interviews are available for download, given in the last section.

Table 3. Frequency of NFRs from the systematic mapping study. The last column shows the number of people who have classified the NFR as relevant and the number of persons who gave an answer (*n*) in the online survey.

ID	NFR	Tun et al. [15]	Khan et al.[18]	Husen et al.[19]	Llinas et al. [24]	McDermott et al.[46]	Razaulla et al.[26]	de Oliveira Carvalho et al.[25]	Pankiewicz et al.[47]	Russell et al.[42]	Nahar et al.[27]	Zhang et al.[48]	Habibullah and Horkoff [37]	Hutchinson et al.[32]	Yurrita et al.[22]	Σ Primary Studies	Importance through online survey (#relevant/n)
NFR-1	Suitability															0	11/11
NFR-2	Explainability and Interpretability				x	x	x	x	x	x	x	x	x	x	x	11	9/9
NFR-3	Justifiability				x		x						x			3	7/8
NFR-4	Transparency		x		x	x	x	x			x		x	x	x	9	3/8
NFR-5	Traceability	x											x		x	3	6/9
NFR-6	Fairness		x			x	x	x		x	x	x	x	x	x	10	6/9
NFR-7	Safety	x		x	x	x		x	x		x		x	x		9	9/11
NFR-8	Trust				x	x		x		x		x	x	x	x	8	8/11
NFR-9	Efficiency of ML algorithm						x						x			2	7/10
NFR-10	Performance of ML model	x			x	x	x	x		x	x	x	x	x	x	11	9/9
NFR-11	Latency of ML model										x					1	10/10
NFR-12	Security					x	x	x			x		x	x	x	7	9/9
NFR-13	Privacy					x	x	x			x		x	x	x	7	10/12
NFR-14	Integrity of data			x							x	x	x		x	5	11/11
NFR-15	Dataset requirements															0	4/6
NFR-16	Accountability						x	x			x			x	x	5	4/8
NFR-17	Reliability	x				x		x				x	x	x	x	7	11/11
NFR-18	Reproducibility and Repeatability	x					x	x			x		x		x	6	8/9
NFR-19	Fault tolerance	x				x				x	x	x	x		x	7	9/10
NFR-20	Autonomy of ML algorithm															0	3/9

ID	NFR	Tun et al. [15]	Khan et al.[18]	Husen et al.[19]	Llinas et al. [24]	McDermott et al.[46]	Razaulla et al.[26]	de Oliveira Carvalho et al.[25]	Pankiewicz et al.[47]	Russell et al.[42]	Nahar et al.[27]	Zhang et al.[48]	Habibullah and Horkoff [37]	Hutchinson et al.[32]	Yurrita et al.[22]	Σ Primary Studies	Importance through online survey
NFR-21	Maintainability				x						x		x			3	8/9
NFR-22	Modularity		x													1	6/7
NFR-23	Reusability												x	x		2	4/6
NFR-24	Modifiability	x								x				x		3	6/7
NFR-25	Retrainability										x		x			2	7/7
NFR-26	Testability				x			x			x		x	x		5	10/10
NFR-27	Usability	x									x		x			3	6/9
NFR-28	Interoperability									x			x			2	5/6
NFR-29	Portability												x	x		2	4/9
NFR-30	Adaptability										x		x			2	5/8
NFR-31	Scalability of ML pipeline									x	x		x			3	6/8
NFR-32	Complexity of ML model								x	x	x		x	x	x	6	1/8
NFR-33	Legal requirements															0	8/9
Σ		8	3	2	8	10	10	12	3	9	19	7	25	15	14	-	-

Table 3 shows that NFRs occur with different frequency in the articles. In particular the work of Habibullah and Horkoff [37] show the most NFRs, as they performed a comprehensive qualitative interview study in which NFRs for ML were explored in detail. Part of the interview study was to identify NFRs that are more or less important in the industry. The Importance through online survey column shows how many people classified the respective NFR as relevant in the online survey. For each NFR, the participants were asked to state whether it is a relevant NFR for systems with ML components. The participants could respond to the statement as follows: strongly disagree, disagree, neither, agree and strongly agree. When evaluating the relevance of the NFRs, the answers with the selection Neither were sorted out in relation to neutrality. An NFR was categorized as relevant for systems with ML components if more than 50% of the participants in the online survey agreed with the statement. Only 14 of the 15 papers identified in **Fig. 1**. are listed in this table, as the paper from Zaidi [12] only deals with documentation techniques and therefore makes no reference to the NFRs of systems with ML components. This table answers RQ2.

4.2. Documentation Techniques for Systems with ML Components

The online survey at adesso SE showed that employees use the following techniques to document NFRs (in order of occurrences): user stories, sentence templates, to be concepts, UMLs (e.g. use case diagram), entity relationship models (ERM), tests, service level agreements (SLA), authorization concepts, mockups and usability standards. The first two techniques were also identified as appropriate by online survey participants. These findings answer RQ3: Using user stories, sentence templates, NFRs can be documented in natural language. The expert interviews also confirm this finding.

4.3. Problems with the Documentation of NFRs

Through the online survey and expert interviews, the following problems in documenting NFRs were identified, which provide the answer to RQ4: (1) No consistent, appropriate, or proper format in documentation, (2) definition problems in certain situations, (3) insufficient specification and thus missing information, which later led to problems during implementation, and (4) missing technical aspect in the requirements documentation.

4.4. Design of the Template for the Documentation of NFRs

Based on project examples, the specifics of documenting NFRs for systems with ML components were examined in the expert interviews. **Table 4** shows the roles, skills and the work experience in years of the experts. The projects did not use a standard template for the documentation of NFRs. For example, in one project a combination of user story and key results was used to document the requirements, and in another project the requirements were documented without using an appropriate sentence template or similar. However, each requirement was prioritized using the MoSCoW prioritization technique [1]. Based on the results of the expert interviews and a literature review, a template was designed that can be used to document NFRs as a natural language documentation technique in a uniform manner. A natural language documentation

Table 4. Details of the experts from adesso SE, data & analytics division for the interview

Expert	Professional Role	Skills	Work experience
Person 1	Data Scientist and ML-Engineer	high	five years
Person 2	Architect, Consultant, Requirements Engineer and Project Manager	high	nine years
Person 3	Data Scientist	high	ten years
Person 4	AI Consultant	high	n/a

technique was chosen because natural language documentation techniques are used most frequently at adesso SE. This does not exclude the possibility that the specified template can also be supplemented by model-based documentation techniques. After the initial conception, the template was evaluated with a further five experts in semi-structured expert interviews at adesso SE. The template used for the evaluation is available for download in the last section. The expert interviews showed beneficial aspects of the template and some need for optimization. The experts were asked whether they would recommend the first conception of the template to colleagues, measured by the Net Promoter Score (NPS). NPS is a metric that categorizes people into three groups, promoters, passives, and detractors based on a question that can be

answered on a scale of 0-10. Only the number of detractors and promoters is required to calculate the NPS, as the percentage of detractors is subtracted from the percentage of promoters [49]. Out of five experts, two experts have a neutral attitude (passives), one expert has a critical attitude (detractor) and two experts have a positive attitude (promoters) towards the present template. The expert with the critical attitude mentioned that s/he would rate the template with a much higher value if the role and benefits were taken out of the template. The *benefit* field is covered by the justification of priority, which is why this field does not offer any added value. Another expert agreed that the role should be removed, as this does not add any value. The *benefit* field and the <role> in the sentence template have therefore been removed. But the *Other notes (optional)* field was added to the template. The following NPS was determined for the first draft of the template: $NPS = ((2-1)/5) \times 100 = 20\%$. According to Lee [50], 20% is a good value. The improved template can be found in **Table 5**. and provides the answer to RQ5. Furthermore, an example can be seen in **Table 6**. The template supports the management of requirements through consistent documentation and the relation the NFR-classes in **Table 3** defined by the NFR-class and the object of consideration in the template. The NFR-class, for example, is relevant because different metrics must be defined in the acceptance criteria depending on the NFR-class. These metrics are not considered in more detail in this paper but would need to be analyzed and defined in more detail in further research work. The NFR-classes are only an example in our template. These can be extended by further NFR-classes. In addition, further research could determine whether the template could be expanded to include further elements. Additionally, the template was evaluated in a second company *Serapion*, presented below. This template is the answer to RQ5.

Table 5. Customized template for the documentation of the most relevant NFRs

Identifier	<Goal/desire related to the object of consideration>.
Acceptance criteria	AC1: <Acceptance criteria> (expandable list)
NFR-class (according Tab.3)	<NFR-class> [Suitability, Integrity of data, Reliability, Latency of ML model, Testability, Explainability and Interpretability, Performance of ML model, Security or Retrainability]
Object of consideration	<Object of consideration> (Training data, ML algorithm, ML model, overall system)
Priority	<Priority specification based on MoSCoW prioritization> [Must-Have, Should-Have, Could-Have or Won't-Have]
Justification of priority	<Justification of the priority>
Other notes (optional)	<e.g., to-do's to fulfill this requirement or other notes>.
Status	<Selection options can be defined depending on the project>
Owner	<person(s) responsible for implementing this NFR>.

Table 6. Example NFR for Integrity of data

NFR-1	The training dataset provided must consist of high-quality data.		
Acceptance criteria	AC1: 100% of duplicates in the training data set were removed. AC2: The amount of data in the data set for the two classes "must be maintained" and "must not be maintained" only differs by a maximum of 15%. AC3: At least 10% of the data from the training data set were manually checked for correct labeling as a sample and are correctly labeled 95% of the time.		
NFR-class	Integrity of data	Priority	Must-Have
Object of consideration	Training data	Owner	Data Engineer
Justification of priority	Training data is the basic building block of ML. Poor training data lead to incorrect and unrealistic predictions about maintenance of a vehicle fleet. Therefore, high quality training data is needed for creation of ML model.		
Other notes (optional)	Review of the training data set must be done within one week.		
Status	Done		

5. Evaluation of a Template for the Documentation of NFRs

The second evaluation was done with Serapion GmbH, which is an IT service provider and management consultancy with more than 100 employees that develops cross-industry solutions in Europe. In order to expand the survey, Serapion was chosen as another consulting company because this company is known as a pioneer of AI technology in the German market. The proposed NFR-template was evaluated in practice using three projects with ML components. By using the technology acceptance model (TAM) [51] the perceived usefulness was evaluated by analysts to measure the degree to which a person believes that using this template would enhance their job performance. Three randomly selected projects have been identified where ML components currently or in the future are used. For each of the three projects, the responsible project manager has been identified in the role of the Requirements Engineer, and the implementing party has been pinpointed. The template was presented to project members. After a short discussion and feedback about the template, a real NFR was documented using the template. The template was filled out together with the respective project manager and feedback was then obtained through verbal interviews with the respective project manager and a respective developer. The results illustrate the benefit (one of the experts called it a remarkable benefit) of the proposed NFR template in specifically capturing and documenting requirements in projects with ML components. Through the practical application of the project examples, the template's effectiveness in improving communication within the project team and stakeholders as well as in helping to prioritize requirements became clear.

The general feedback to the template was: (i) during a proof-of-concept phase, the template is not required. In early project stages the focus lies on validating ideas and testing concepts. (ii) The template is highly valuable in subsequent stages, where implementation and scaling occur. The NFR template represents a valuable resource for managing NFR. The recommended extensions ensure improved practicality and completeness. It is recommended to use this template in software projects and customize it according to project situation and requirements.

6. Threats to Validity

We follow the classification of Wohlin et al. [52]. Conclusion validity addresses whether the soundness of the treatment is related to the actual observed outcome. As the template was found useful by the experts, this is an indication that it contributes to capturing the requirements for ML components adequately. **Construct validity** considers whether the study measures what it claims to measure. For the systematic mapping study, the RQs were discussed in the team of authors and feedback from a university talk was considered. The online survey was conducted in one company only and contained 177 questions, which might frighten some participants. Pretests were conducted with three people. Some answers did not describe precisely how the NFRs were documented by the specified techniques. Expert interviews were conducted with adesso SE using a guideline with questions, that might have been misunderstood by experts. We used pretests with two people to mitigate misunderstanding. **Internal validity** determines the extent of conclusions that can be drawn from a study. We scoped the systematic mapping study to RE of NFRs for ML components. The search term was validated by all three authors and the process of Petersens mapping study was followed, beside the quality measurement of primary studies. The use of four databases covers publications since 2021 in a broad range. Three related SLRs cover results before 2021. The online survey of adesso SE employees gained

a small sample size of 12. Results of the expert interviews were biased upon the level of expertise of participants. Therefore, we raised questions about the participant's ML experience. Only 4 people from adesso SE participated in the expert interviews. **External validity** describes the possible generalization or transfer of the study results to other situations. The results of the mapping study might be biased through the missing quality evaluation. The scope *ML based components* limit the generality of the presented insights of the mapping study, online survey and interviews; thus, results are not valid for other types of systems. We did not distinguish different types of ML such as supervised, unsupervised, and reinforced learning. The evaluation expert interviews were only conducted in two companies, which limits transferability to other companies. For the evaluation in the second company, the template was rated by 3 people, who had not seen the template before. However, the rating of the template overall, was useful.

7. Conclusion

The peculiarities of documenting NFRs in systems with ML components were identified and 30 NFRs were considered relevant. In addition, no consistent, suitable format for the documentation of ML NFRs could be found. Expert interviews showed a missing standardized approach to documenting NFRs for ML systems. As a result, a specific template to capture NFR requirements was proposed. The template was evaluated in practice and showed significant advantages in usefulness. Criticisms have been raised wrt. early research phases of a project, where detailed requirement documentation may not be relevant. The added value of the proposed NFR template in later project phases was recognized, especially for the implementation of NFRs. Feedback during the evaluations lead to a revision of the NFR template. There is still a need to understand the effectiveness and applicability of the template in larger companies. The adaptation of the template to different ML methods is still in its early stages and requires further research. The identification and integration of relevant NFRs for these learning methods is an under-researched area. In summary, this research showed particularly important NFRs for ML components and showed the usefulness of the proposed NFR template to support the implementation of projects with ML components.

References

1. M. Glinz, H. van Loenhoud, S. Staal and S. Bühne: Handbook for the CPRE Foundation Level, IREB Standard (2020)
2. C. Rupp und SOPHISTen: Requirements-Engineering und -Management: Das Handbuch für Anforderungen in jeder Situation. Hanser, München (2021)
3. Ahmad, K., Bano, M., Abdelrazek, M., Arora, C., Grundy, J.: What's up with Requirements Engineering for Artificial Intelligence Systems? In: 2021 IEEE 29th International RE Conf., pp. 1–12. IEEE (2021). doi: 10.1109/RE51729.2021.00008
4. Kreutzer, R.T., Sirrenberg, M.: Künstliche Intelligenz verstehen. Springer Fachmedien Wiesbaden, (2019)
5. Welsch, A., Eitle, V., Buxmann, P.: Maschinelles Lernen. HMD, vol. 55, 366–382 (2018). doi: 10.1365/s40702-018-0404-z
6. Yoshioka, N., Husen, J.H., Tun, H.T., Chen, Z., Washizaki, H., Fukazawa, Y.: Landscape of Requirements Engineering for Machine Learning-based AI Systems. In: 28th Asia-Pacific Softw.Eng.Conf.Works., IEEE (2021).
7. Villamizar, H., Escovedo, T., Kalinowski, M.: Requirements Engineering for Machine Learning: A Systematic Mapping Study. In: 2021 47th Euromicro Conf. on Softw. Eng. and Adv. Appl.(SEAA), pp. 29–36. IEEE (2021).
8. Gerard, C.: Practical Machine Learning in JavaScript. Apress, Berkeley, CA (2021)
9. Haller, K.: Managing AI in the Enterprise. Apress, Berkeley, CA (2022)

10. ISO/IEC: 25010: Systems and software engineering — Systems and Software Quality Requirements and Evaluation. System and software quality models, (2011)
11. Kitchenham, B.A., Charters, S.: Guidelines for Performing Systematic Literature Reviews in Software Engineering. Tech. Rep. EBSE 2007-001. Keele, UK (2007)
12. Zaidi, M.A.: Conceptual Modeling Interacts with Machine Learning – A Systematic Literature Review. In: Gervasi, O., et al. (eds.) *Comp.Sc.and Its Appl.ICCSA 2021*. LNCS, vol. 12957, pp. 522–532. Springer International Publishing, Cham (2021).
13. Object Management Group: Business Process Model and Notation, (2014).
14. A. Borrmann, M. König, C. Koch, J. Beetz: Building Information Modeling: Technologische Grundlagen und industrielle Praxis. Springer Fachm. Wiesbaden (2015)
15. Tun, H.T., Husen, J.H., Yoshioka, N., Washizaki, H., Fukazawa, Y.: Goal-Centralized Metamodel Based Requirements Integration for Machine Learning Systems. In: 28th Asia-Pacific Soft.Eng.Conf.Works., IEEE (2021).
16. Leveson, N.: A new accident model for engineering safer systems. *Safety Science*, vol. 42, 237–270 (2004).
17. Object Management Group: Systems Modeling Language (OMG SysML™).
18. Khan, A., Siddiqui, I.F., Shaikh, M., Anwar, S., Shaikh, M.: Handling Non-Functional Requirements in IoT-based Machine Learning Systems. In: *Joint Int.Conf.on Digital Arts, Media a.Technology (ECTI DAMT & NCON)*, pp. 477–479. IEEE (2022).
19. Husen, J.H., Washizaki, H., Tun, H.T., Yoshioka, N., Fukazawa, Y., Takeuchi, H.: Traceable business-to-safety analysis framework for safety-critical machine learning systems. In: Crnkovic, I. (ed.) *Proc.1st Int. Conf. on AI Engineering: Softw.Eng. for AI*, pp. 50–51. ACM, New York, NY, USA (2022).
20. Allison, C.K., Revell, K.M., Sears, R., Stanton, N.A.: Systems Theoretic Accident Model and Process safety modelling applied to an aircraft rapid decompression event. *Safety Science*, vol. 98, 159–166 (2017). doi: 10.1016/j.ssci.2017.06.011
21. M. Hesenius, N. Schwenzfeier, O. Meyer, W. Koop and V. Gruhn: Towards a Software Engineering Process for Developing Data-Driven Applications.7th Int.Work.on Realizing AI Synergies in Softw.Eng.(RAISE), pp. 35–41.
22. Yurrita, M., Murray-Rust, D., Balayn, A., Bozzon, A.: Towards a multi-stakeholder value-based assessment framework for algorithmic systems. In: *ACM Conf.on Fairness, Accountability, and Transp.*, pp. 535–563. ACM, New York, USA (2022).
23. Jain, S., Luthra, M., Sharma, S., Fatima, M.: Trustworthiness of Artificial Intelligence 2020 6th Int.Conf.on Adv.Comp.a.Comm. Syst.(ICACCS), pp. 907–912.
24. Llinas, J., Fouad, H., Mittu, R.: Systems Engineering for Artificial Intelligence-based Systems: A Review in Time. In: Lawless, W.F., et al. (eds.) *Syst.Eng. and Artif.Int.*, pp. 93–113. Springer Internat. Publishing, Cham (2021).
25. Oliveira Carvalho, N. de, Libório Sampaio, A., Vasconcelos, D.R. de: MoReXAI - A Model to Reason About the eXplanation Design in AI Systems. In: Degen, H.,et al. (eds.) *Art.Int. in HCI*. LNCS, vol. 13336, pp. 130–148. Springer Int. Pub., (2022).
26. Razaulla, S.M., Pasha, M., Farooq, M.U.: Integration of Machine Learning in Education: Challenges, Issues and Trends. In: Satyanarayana, C., et al. (eds.), *Adv. Techn.and Soc. Change*, pp. 23–34. Springer Singapore, (2022).
27. Nahar, N., Zhou, S., Lewis, G., Kästner, C.: Collaboration challenges in building ML-enabled systems. In: Dwyer, M.B., et al.(eds.) *Proc. of the 44th Int.Conf.on Softw.Engi.*,pp. 413–425. ACM, New York, NY, USA (2022).
28. Mora-Cantalops,M.,Sánchez-Alonso,S.,García-Barriocanal,E.,Sicilia,M.-A.: Traceability for Trustworthy AI: A Review of Models and Tools. *BDCC*, vol. 5, 20 (2021).
29. Xivuri, K., Twinomurinzi, H.: A Systematic Review of Fairness in Artificial Intelligence Algorithms. In: Dennehy, D., et al. (eds.) *Resp.AI a.Anal.f.Ethical a.Inclusive Digitized Soc.*,vol.12896, pp. 271–284.Springer,Cham (2021).
30. Barton, M.-C., Pöppelbuß, J.: Prinzipien für die ethische Nutzung künstlicher Intelligenz. *HMD*, vol. 59, 468–481 (2022).
31. Xiao, C., Sun, J.: Introduction to Deep Learning for Healthcare. Springer International Publishing, Cham (2021)
32. Hutchinson, B., Rostamzadeh, N., Greer, C., Heller, K., Prabhakaran, V.: Evaluation Gaps in Machine Learning Practice. In: 2022 ACM Conf. on Fairness, Accountability, and Transparency, pp. 1859–1876. ACM, USA (2022).
33. Zhu, B., Shin, U., Shoaran, M.: Closed-Loop Neural Prostheses With On-Chip Intelligence: A Review and a Low-Latency Machine Learning Model for Brain State Detection. *IEEE Biomed.Circ.Syst.*, vol.15, 877–897 (2021).
34. Hutchinson, B., Smart, A., Hanna, A., Denton, E., Greer, C., Kjartansson, O., Barnes, P., Mitchell, M.: Towards Accountability for ML Datasets. In: *Proc.Conf.on Fairness, Account.a.Transp.*, pp. 560–575. ACM, USA (2021).

35. Lepri, B., Oliver, N., Letouzé, E., Pentland, A., Vinck, P.: Fair, Transparent, and Accountable Algorithm. *Decision-making Proc. Philos. Techn.* vol. 31, 611–627 (2018).
36. Alahmari, S.S., Goldgof, D.B., Mouton, P.R., Hall, L.O.: Challenges for the Repeatability of Deep Learning Models. *IEEE Access*, vol. 8, 211860–211868 (2020).
37. Habibullah, K.M., Horkoff, J.: Non-functional Requirements for Machine Learning: Understanding Current Use and Challenges in Industry. In: 2021 IEEE 29th International RE Conf., pp. 13–23. IEEE (2021). doi: 10.1109/RE51729.2021.00009
38. Myllyaho, L., Raatikainen, M., Männistö, T., Nurminen, J.K., Mikkonen, T.: On misbehaviour and fault tolerance in machine learning systems. *Journal of Systems and Software*, vol. 183, 111096 (2022). doi: 10.1016/j.jss.2021.111096
39. Hanif, M.A., Khalid, F., Putra, R.V.W., Rehman, S., Shafique, M.: Robust Machine Learning Systems: Reliability and Security for Deep Neural Networks. In 24th Int. Symp. on On-Line Test. a Robust Syst. Design (IOLTS), pp. 257–260. IEEE (2018).
40. Mikkonen, T., Nurminen, J.K., Raatikainen, M., Fronza, I., Mäkitalo, N., Männistö, T.: Is Machine Learning Software Just Software: A Maintainability View. In: *Fut. Persp. o. Softw. Eng. Quality*. vol. 404, pp. 94–105. Springer Int. Publishing, (2021).
41. Csurka, G.: A Comprehensive Survey on Domain Adaptation for Visual Applications. In: Csurka, G. (ed.) *Advances in Computer Vision and Pattern Recognition*, pp. 1–35. Springer International Publishing, Cham (2017).
42. Russell, S., Jalaian, B., Moskowitz, I.S.: Re-orienting Toward the Science of the Artificial: Engineering AI Systems. In: Lawless, W.F., et al. (eds.) *Syst. Eng. a. Artificial Intelligence*, pp. 149–174. Springer International, Cham (2021).
43. Migliorini, M., Castellotti, R., Canali, L., Zanetti, M.: ML Pipelines with Modern Big Data Tools for High Energy Physics. *Comp. Softw Big Sci*, vol. 4 (2020).
44. Chen, L.: *Deep Learning and Practice with MindSpore*. Springer Singapore, (2021)
45. Wong, A.: Ethics and Regulation of Artificial Intelligence. In: *Art. Int. for Knowl. Mgmt. IFIP Adv. i. Inf. a. Comm. Techn.*, vol. 614, pp. 1–18. Springer Int. Pub., (2021).
46. McDermott, T.A., Blackburn, M.R., Beling, P.A.: Artificial Intelligence and Future of Syst. Eng. In: *Syst. Eng. a. Art. Int.*, pp. 47–59. Springer Int. Pub. Cham (2021).
47. Pankiewicz, N., Wrona, T., Turlej, W., Orłowski, M.: Promises and Challenges of Reinforcement Learning Applications in Motion Planning of Automated Vehicles. In: *Art. Int. Soft Comp.* vol. 12855, pp. 318–329. Springer Int. Pub., (2021).
48. Zhang, R., Albrecht, A., Kausch, J., Putzer, H.J., Geipel, T., Halady, P.: DDE process: A requirements engineering approach for machine learning in automated driving. In: 29th RE Conference (RE), pp. 269–279. IEEE (2021).
49. Owen, R.: Net Promoter Score and Its Successful Application. In: Kompella, K. (ed.) *Marketing Wisdom. Management for Professionals*, pp. 17–29. Springer Singapore, Singapore (2019). doi: 10.1007/978-981-10-7724-1_2
50. Lee, S.: Net Promoter Score. In: Carney-Morris, M., Appling, J., Spigelmyer, A. (eds.) *Proceedings of the 2018 ACM SIGUCCS Annual Conference*, pp. 63–64. ACM, New York, NY, USA (2018). doi: 10.1145/3235715.3235752
51. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User Acceptance of Comp. Technology: A Comparison of Two Theoretical Models. *Managm. Sc.* vol. 35, 982–1003 (1989).
52. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: *Experimentation in Software Engineering*. Springer Berlin Heidelberg, (2012)
53. Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M.: Systematic mapping studies in software engineering. In 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12, pp. 1-10 (2008)
54. A. Gjorgjevikj, K. Mishev, L. Antovski and D. Trajanov: Requirements Engineering in Machine Learning Projects, in *IEEE Access*, vol. 11, pp. 72186-72208, 2023, doi: 10.1109/ACCESS.2023.3294840.

Additional material for this article can be downloaded from the zenodo online repository of the European Union: <https://10.5281/zenodo.10640638>.