

# Exploring Explainability: A Definition, a Model, and a Knowledge Catalogue

Larissa Chazette\*, Wasja Brunotte\*<sup>†</sup> Timo Speith<sup>‡</sup>

\*Leibniz University Hannover, Software Engineering Group, Hannover, Germany

<sup>†</sup>Leibniz University Hannover, Cluster of Excellence PhoenixD, Hannover, Germany

<sup>‡</sup>Saarland University, Institute of Philosophy and Department of Computer Science, Saarbrücken, Germany

Email: {larissa.chazette, wasja.brunotte}@inf.uni-hannover.de, timo.speith@uni-saarland.de

**Abstract**—The growing complexity of software systems and the influence of software-supported decisions in our society awoke the need for software that is transparent, accountable, and trustworthy. Explainability has been identified as a means to achieve these qualities. It is recognized as an emerging non-functional requirement (NFR) that has a significant impact on system quality. However, in order to incorporate this NFR into systems, we need to understand *what* explainability means from a software engineering perspective and *how* it impacts other quality aspects in a system. This allows for an early analysis of the benefits and possible design issues that arise from interrelationships between different quality aspects. Nevertheless, explainability is currently under-researched in the domain of requirements engineering and there is a lack of conceptual models and knowledge catalogues that support the requirements engineering process and system design. In this work, we bridge this gap by proposing a definition, a model, and a catalogue for explainability. They illustrate how explainability interacts with other quality aspects and how it may impact various quality dimensions of a system. To this end, we conducted an interdisciplinary Systematic Literature Review and validated our findings with experts in workshops.

**Index Terms**—Explainability; Explanations; Explainable Artificial Intelligence; Interpretability; Non-Functional Requirements; Quality Aspects; Requirements Synergy; Software Transparency

## I. INTRODUCTION

We live in the age of artificial intelligence (AI). Software decision-making has spread from simple daily decisions, such as the choice of a navigation route, to more critical ones, such as the diagnosis of cancer patients [2]. Systems have been strongly influencing various aspects of our lives with their outputs but can be as mysterious as black boxes to us [3]. This ubiquitous influence of black-box systems has induced discussions about the transparency and ethics of modern systems [4]. Responsible collection and use of data, privacy, and safety are just a few among many concerns. It is crucial to understand how to incorporate these concerns into systems and, thus, how to deal with them during requirements engineering (RE).

Explainability is increasingly seen as the preferred solution to mitigate a system's lack of transparency and should be treated as a non-functional requirement (NFR) [5]. Incorporating explainability can mitigate software opacity, thereby helping users understand why a system produced a particular result and supporting them in making better decisions. Explainability also has an impact on the relationship of trust in and reliance on a system [6], and it may avoid feelings of frustration [7].

Although explainability has been identified as an essential NFR for software-supported decisions [8] and one of the pillars for trustworthy AI [9], there is still a lack of an extensive overview that investigates the impact of explainability on a system.

In this paper, we investigate the concept of explainability and its interaction with other quality aspects. We use the notion of *quality aspects* to refer both to *NFRs* and to *aspects* that relate to or compose NFRs. In this regard, we follow Glinz and see NFRs as attributes of or constraints on a system [10].

Previous studies have shown that explainability is not only a means of achieving transparency and building trust but that it is also linked to other important NFRs, such as usability and auditability [11]–[13]. Explainability can, however, have both a positive and a negative impact on a system. Like other NFRs, explainability is difficult to elicit, negotiate, and validate. Eliciting and modeling NFRs is often a challenge for requirements engineers due to the subjective, interactive, and relative nature of NFRs [14]. Often there are trade-offs between NFRs in a system that must be identified and resolved during the requirements analysis [14], [15].

One of the challenges with respect to NFRs stems from the fact that information concerning them is rather tacit, distributed, and based on experience [16], [17]. To mitigate this, a usual strategy adopted by requirements engineers to deal with NFRs during RE is to make use of artifacts such as conceptual models [16], pre-established lists, or knowledge catalogues [14]. Models and catalogues can be used to help specify quality requirements [18]. They may compile knowledge about specific NFRs and their interactions with other quality aspects. Among others, such artifacts support the elicitation process. Models can be used to understand the taxonomy of a given quality aspect during analysis, while catalogues can support the analysis of trade-offs where it is essential to understand how two or more NFRs will interact in a system and how they can coexist [19]. Requirements engineers can also use models and catalogues to ask stakeholders about their interest with respect to the general system quality [20]. Existing works propose to build such artifacts to capture and structure knowledge that is scattered among several sources [14], [16], [21], [22].

Since explainability is an emerging requirement, there is still a lack of structured knowledge about this NFR. To bridge this gap, we employed a multi-method research strategy consisting of an interdisciplinary Systematic Literature Review (SLR)

and workshops. Overall, our goal is to advance the knowledge towards a common terminology and semantics, facilitating the discussion and analysis of explainability during the RE process. To this end, we distill definitions of explainability into an own suggestion that is useful for software and requirements engineering. We use this definition as a starting point to create a model that represents the impacts of explainability across different quality dimensions. Finally, we construct a knowledge catalogue of explainability and its impacts that is framed along these dimensions.

## II. BACKGROUND AND RELATED WORK

Chung et al. [16] explain the importance of conceptual models and knowledge catalogues as resources for the use and re-use of knowledge during system development. Models and catalogues can compile either abstract or concrete knowledge. At a more abstract level, such artifacts can compile knowledge about different NFRs and their interrelationships with other quality aspects. Likewise, models and catalogues can also compile more concrete knowledge, such as about methods and techniques in the field that can be used to operationalize a given NFR. The knowledge required to develop such artifacts is typically derived from literature, previous experiences, and domain expertise. By making this knowledge available in a single framework, developers can draw on know-how beyond their own fields and use this knowledge to meet the needs of a particular project. Essentially, software engineers can use models and knowledge catalogues to facilitate the software design process.

Some researchers developed catalogues for specific domains based on the premise of the NFR framework. Serrano and Serrano [21] developed a catalogue specifically for the ubiquitous, pervasive, and mobile computing domain. Torres and Martins [23] propose the use of NFR catalogues in the construction of RFID middleware applications to alleviate the challenges of NFR elicitation in autonomous systems. They argue that the use of catalogues can reduce or even eliminate possible faults in the identification of functional and non-functional requirements. Carvalho et al. [24] propose a catalogue for invisibility requirements focused on the domain of ubiquitous computing applications. They emphasize the importance of software engineers understanding the relationships between requirements in order to select appropriate strategies to satisfy invisibility and traditional NFRs. Furthermore, they discovered that invisibility may impact other essential NFRs for the domain, such as usability, security and reliability.

On a general level, Mairiza et al. [14] conducted a literature review to identify conflicts among existing NFRs. They constructed a catalogue to synthesize the results and suggest that it can assist software developers in identifying, analyzing, and resolving conflicts between NFRs. Carvalho et al. [22] identified 102 NFR catalogues in the literature after conducting a systematic mapping study. They found that the most frequently cited NFRs were performance, security, usability, and reliability. Furthermore, they found that the catalogues are represented in different ways, such as softgoal

interdependency graphs, matrices, and tables. The existence of so many catalogues illustrates their importance for RE and software design. Although these catalogues present knowledge about 86 different NFRs, none of them addresses explainability.

Since explainability has rapidly expanded as a research field in the last years, publications about this topic have become quite numerous, and it is hard to keep track of the terms, methods, and results that came up. For this reason, there have been numerous SLRs presenting overviews concerning certain aspects (e.g., used methods or definitions) of explainability research. Many of these reviews focus on a specific community or application domain. For instance, [25] focuses on explainability of recommender systems, [26] on explainability of robots and human-robot interaction, [27] on the human-computer interaction (HCI) domain, and [28] on biomedical and malware classification. Another focus of these reviews is to demarcate different, but related terms often used in explainability research (e.g., in [4] and [29]). For instance, the terms "explainability" and "interpretability" are sometimes used as synonyms and sometimes not.

Our review differs from others in the following ways. First, many other reviews do not have an interdisciplinary focus. Even if they do not focus on a specific community, they rarely incorporate views on explainability outside of computer science. Second, quality aspects are the pivotal focus of our work. To the best of our knowledge, only a few reviews explicitly include NFRs or quality aspects (most notably [25] and [30]). Finally, in contrast to preceding reviews, we do not only consider positive impacts of explainability on other quality aspects, but we also take negative ones into account.

## III. RESEARCH GOAL AND DESIGN

We frame our study into the following three RQs:

- RQ1:** What is a useful definition of explainability for the domains of software and requirements engineering?
- RQ2:** What are the quality aspects impacted by explainability in a system context?
- RQ3:** How does explainability impact these quality aspects?

Since other disciplines have a long history working on explainability, their insights should prove valuable for software engineering and enable us to delineate the scope of the term explainability for this area. Accordingly, **RQ1** focuses on harnessing the work of other sciences in the field of explainability to compile a definition that is useful for the area of software and requirements engineering.

**RQ2** focuses on providing an overview of the quality aspects that may be impacted by explainability. Similar to the work of Leite and Capelli [31], who investigated the interaction between transparency and other qualities, our goal is to offer an overview for explainability and its impact on other quality aspects within a system.

With **RQ3** we want to assess what kind of impacts explainability has on other quality aspects. More specifically, our goal is to analyze the polarity of these impacts: whether they are positive or negative. To answer RQ2 and RQ3, we build a model and a catalogue that compiles knowledge about the impacts of explainability on other quality aspects.

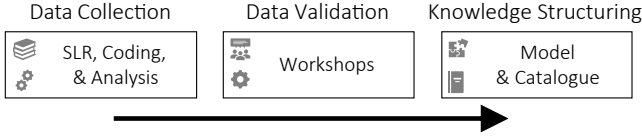


Fig. 1. Overview of the research design

An overview of our research design is shown in Fig. 1. Our research consisted of a multi-method approach that combined two qualitative methods to achieve higher reliability of our data. The first method focuses on systematic data collection and qualitative data analysis. For the data collection, we conducted an interdisciplinary *SLR* that resulted in a total of 229 papers. We *coded* the gathered data by using an open coding approach [32]. As a next step, we *analyzed* the resulting codes for definitions of explainability (RQ1), for relationships between explainability and other quality aspects (RQ2), and for information about the polarity of these relationships (RQ3). To validate and complement our findings, we employed a second qualitative method: two *workshops* with experts. Finally, we framed the obtained knowledge in a *model* by structuring and grouping the quality aspects impacted by explainability along four dimensions and developed our *catalogue* based on it.

The focus of this paper is on the results obtained through the qualitative research we conducted. For details on the SLR, especially the inclusion/exclusion criteria and the complete list of the papers analyzed through our literature review, please refer to our supplementary material [1]. A more detailed description of the workshops can also be found there. Further results and details about the methodology employed in our literature review will be addressed in a future publication.

#### A. Data Collection and Analysis

1) *Systematic Literature Review*: We followed guidelines from Kitchenham et al. [33], and Wohlin [34] when conducting our SLR. The search strategy for our SLR consisted of a manual search followed by a snowballing process.

The manual search was performed independently by the authors of this paper and resulted in 104 papers. We used Fleiss' Kappa statistics [35] to assess the reliability of the selection process. The calculated value of  $\kappa = 0.81$  showed an almost perfect agreement [36]. After the manual search, we performed snowballing to complement the search results. Our snowballing procedure included forward and backward snowballing [34] and resulted in additional 125 papers. Overall, our SLR yielded a total of 229 papers. The snowballing step was also independently conducted by the authors. The calculated value of  $\kappa = 0.87$  also shows an almost perfect agreement.

This literature review process is partially based on a grounded theory (GT) approach for literature reviews proposed by Wolfswinkel et al. [37]. The goal of using this approach to reviewing the literature is to reach a detailed and relevant analysis of a topic, following some of the principles of GT. According to [37], a literature review is never complete but at most saturated. This saturation is achieved when no new concepts or categories arise from the data. We followed this approach to decide when to conclude our snowballing process.

2) *Coding and Analysis*: We followed an open-coding approach [32] for the qualitative analysis of the papers we found during our search. This approach consists of up to three consecutive coding cycles. For our first coding cycle, we applied *Initial Coding* [38] to preserve the views and perspectives of the authors in the code. In the second coding cycle, we clustered the initial codes based on similarities, using *Pattern Coding* [39]. This allowed us to group the data from the first coding cycle into categories. Next, we discussed these categories until we reached an agreement on whether they adequately reflected the meaning behind the codes. These categories allowed us to structure the data for better analysis and to identify similarities.

For RQ2 and RQ3, we conducted a third coding cycle to further classify the categories into quality aspects. We applied *Protocol Coding* [40] as a procedural coding method in this cycle. For this method, we used a pre-established list of NFRs from Chung et al. [16]. If any correspondence between a category and an NFR was found, we assigned the corresponding code. In the specific cases where we could not assign a corresponding NFR from [16] to the data, we discussed together and selected a quality aspect that would adequately describe the idea presented in the text fragment. All coding processes were conducted independently by the authors of this paper. We had regular consensus sessions to discuss discrepancies. A list of all codes is available in our supplementary material [1].

#### B. Data Validation

We held two workshops to validate and augment the knowledge gathered during data collection: one with philosophers and psychologists, and one with requirements engineers. In both workshops, we discussed the categories and other relevant information that were identified during our coding. For RQ1, the categories consisted of competing definitions of explainability that we extracted from the literature. For RQ2, the categories consisted in the identified quality aspects that have a relationship with explainability. Finally, for RQ3, we identified the kind of impact that explainability can have on each of the extracted quality aspects.

1) *Workshop with Philosophers and Psychologists*: We validated the data related to RQ1 in a workshop with philosophers and psychologists (two professors, one postdoc, three doctoral candidates). Scholars in these disciplines have a long history in researching explanations and, thus, explainability. After consulting with experts from these disciplines on the workshop design, we decided on an open discussion. We instructed participants to hand in their notion of explainability prior to the meeting. During the workshop, we presented our coded data

with the most prominent categories concerning RQ1. After a round of discussion, we presented the submitted definitions and compared them with our findings. We debated the similarities and differences between both and reached a consensus that eventually led to our proposed definition.

2) *Workshop with Requirements Engineers*: We validated the data related to RQ2 and RQ3 in a workshop with requirements engineers (three professors, two postdocs, one practitioner, one doctoral candidate). Two experts in the field of RE with experience in the topic of NFRs and software quality were consulted about the workshop design. The participants of this workshop also had to hand in a task in advance. For the task, we sketched scenarios where an explainable system was to be developed and sent the list of quality aspects we found, as seen in Fig. 3. To avoid bias, we removed the polarities. Participants had to indicate which quality aspects were important in each system and what possible influence explainability could have on each of them. During the workshop, we discussed the outcomes of this assignment and had an open debate on the aspects on the list. Afterwards, we presented our findings and compared them with the received feedback. Overall, the experts were able to relate to each of the polarities we found.

### C. Knowledge Structuring

The last step of our research consisted of making sense of and structuring the knowledge collected in the previous stages.

1) *Framing the Results – Model*: We built a model to frame our knowledge catalogue. This model illustrates the impact of explainability on several quality dimensions (see Fig. 2; RQ2). During the workshop with requirements engineers, we discussed possible ways to classify the different quality aspects. Here, the participants offered useful ideas. To further supplement these ideas, we consulted the literature and found three promising ways to classify the results. These three ways are analogous to the suggestions made by the workshop participants and supported us in the development of our model.

2) *Catalogue Construction*: We summarized the results for RQ3 in a knowledge catalogue for explainability. Overall, we have extracted 57 quality aspects that might be influenced by explainability. We present these quality aspects and how they are influenced by explainability in Fig. 3. Additionally, we extracted a representative example from the literature for all positive and negative influences listed in our catalogue to show how this influence may come about. These examples also serve to illustrate our understanding of certain quality aspects.

We present the results for RQ1 in Sec. IV and the results for RQ2 and RQ3 in Sec. V and VI.

## IV. A DEFINITION OF EXPLAINABILITY

The domain of software engineering does not need a mere abstract definition of explainability, but one that focuses on requirements for explainable *systems*. Before requirements engineers can elicit the need for explainability in a system, they have to understand what explainability is in a system context. For this reason, we provide a definition of what makes a system explainable to answer our first RQ.

Explainability is tied to disclosing information, which can be done by giving explanations. Köhl et al. hold that what makes a system explainable is the access to explanations [5]. However, this leaves open what exactly is to be explained. In the literature, definitions of explainability vary considerably in this regard. Moreover, our review has revealed other aspects in which definitions of explainability differ. Consequently, there is not *one* definition of explainability, but several complementary ones. Similarly, Köhl et al. also found that there is not just one type of explainability, but that a system may be explainable in one respect but not in another [5]. Based on their definition of explainability, the definitions we found in the literature, and results from our workshop with philosophers and psychologists, we were able to develop an abstract definition of explainability that can be adjusted according to project or field of application.

**Answering RQ1:** A system  $S$  is explainable with respect to an aspect  $X$  of  $S$  relative to an addressee  $A$  in context  $C$  if and only if there is an entity  $E$  (the explainer) who, by giving a corpus of information  $I$  (the explanation of  $X$ ), enables  $A$  to understand  $X$  of  $S$  in  $C$ .

There were differences in the literature concerning the values of the following variables presented in the above definition: aspects of a system that should be explained, contexts in which to explain, the entity that does the explaining (the explainer), and addressees that receive the explanation. Being aware of these differences is crucial for requirement engineers, as they need to elicit the right kind of explainability for a project. We will shortly discuss some of these findings in what follows.

a) *Aspects that should be explained*: Concerning the aspects that should be explained, we found the following options in the literature and validated them during the workshop with philosophers and psychologists: the system in general [41], and, more specifically, its reasoning processes (e.g., inference processes for certain problems) [42], its inner logic (e.g., relationships between the inputs and outputs) [5], its model's internals (e.g., parameters and data structures) [43], its intention (e.g., pursued outcome of actions) [44], its behavior (e.g., real-world actions) [45], its decision (e.g., underlying criteria) [4], its performance (e.g., predictive accuracy) [46], and its knowledge about the user or the world (e.g., user preferences) [45].

b) *Contexts and Explainers*: A context is set by a situation consisting of the interaction between a person, a system, a task, and an environment [47]. Plausible influences on the context are time-pressure, the stakes involved, and the type of system [30]. Explainers refer to a system or specific parts of a system that supply its stakeholders with the needed information.

c) *Addressee's Understanding*: A vast number of papers in the literature make reference to the addressee's understanding as important factor for the success of explainability (e.g., [13], [29], [41], [48], [49]). Framing explainability in terms of understanding provides the benefit of making it measurable, as there are established methods of eliciting a person's understanding of something, such as questionnaires or usability tests [11].

## V. A MODEL OF EXPLAINABILITY

Models and catalogues compile knowledge about quality aspects and help to better visualize their possible impact on a system. Based on the data extracted from the literature and on our qualitative data analysis and validation, we were able to build a model and a catalogue for explainability. Overall, our model is divided into four dimensions. We considered three existing concepts to shape and compose these dimensions.

**Answering RQ2:** We framed the quality aspects that are impacted by explainability in a model that spans different quality dimensions of a system (Fig. 2).

The first concept connects to our definition and is based on the insight that understanding is pivotal for explainability. Langer et al. tackle explainability from the perspective of the persons who inquire after explanations [30]. Individuals differ in their background-knowledge, values, experiences, and many further respects. Accordingly, they also differ in what is required for them to understand certain aspects of a system. Furthermore, Langer et al. also hold that some persons are more likely to be interested in a certain quality aspect than others. For instance, a developer may be more interested in the maintainability of a system than a user. They categorize quality aspects that are influenced by explainability according to so-called *stakeholder classes* and distinguish the following ones:

*users, developers, affected parties, deployers, and regulators.* According to them, these classes should serve as a reference point when it comes to implementing explainability since the interests of different stakeholder classes may conflict [30].

Chazette and Schneider identified six dimensions that affect the elicitation and analysis of explainability [11]: the *needs and expectations of users, cultural values, corporate values, laws and norms, domain aspects, and project constraints*. Their results indicate that different factors distributed across these dimensions influence the identification of explainability as being a necessary NFR within a system and the design choices towards its operationalization. We adopt and extend this notion in that we consider these dimensions to be decisive not only for the RE process but also for a system in general.

Finally, the external/internal quality concept based on the ISO 25010 [50] and proposed by [51] is another way to categorize the quality aspects in our model. We consider the *external* quality characteristics as the ones which are more related to the users or the *quality in use*, and the *internal* as the ones which are more related to the developers or the *system itself*. As pointed out by McConnel [52], the difference between internal and external characteristics is not completely clear-cut and affects several dimensions. Therefore, we do not assign clear-cut internal or external dimensions, but rather acknowledge a continuous shift from external to internal.

Based on these concepts, we developed a model of explainability and its impacts on other quality aspects. We frame

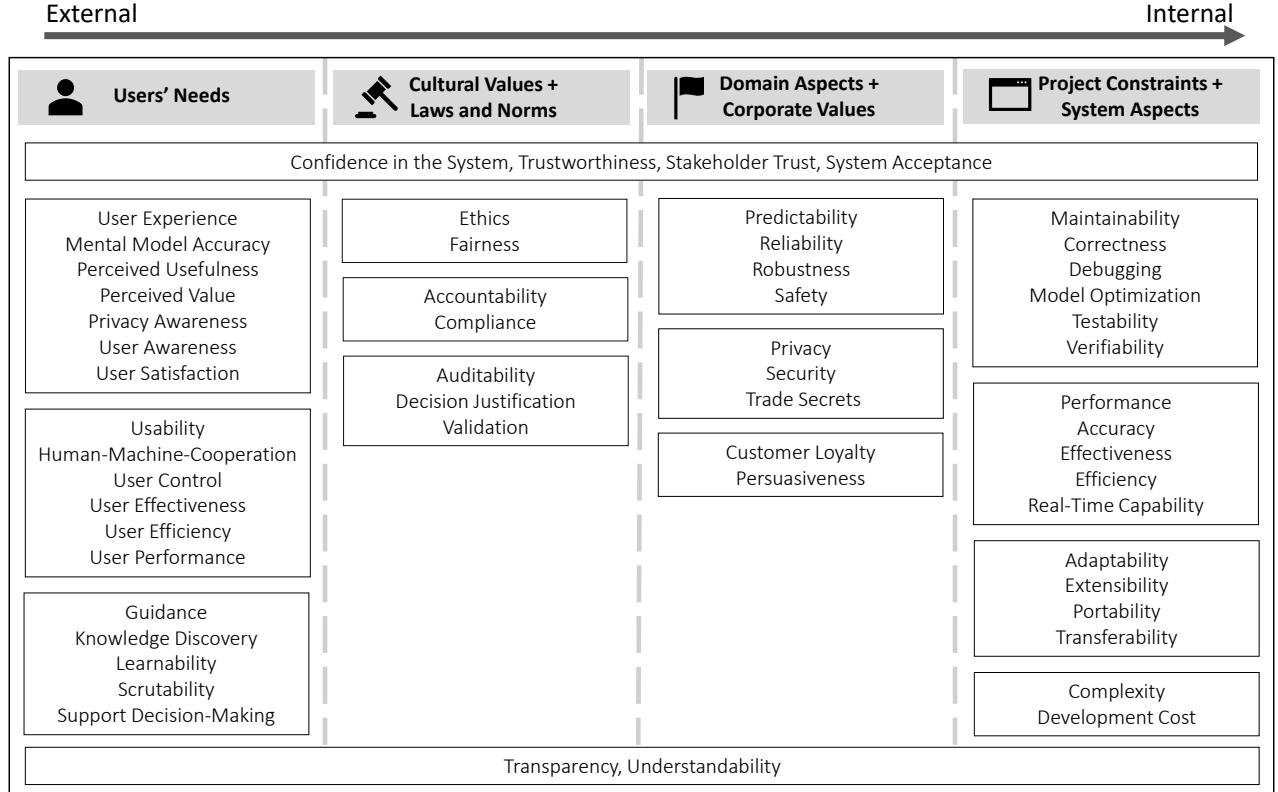


Fig. 2. A model illustrating the impact of explainability across different quality dimensions

the quality aspects along the four dimensions of our model: *user's needs*, *cultural values & laws and norms*, *domain aspects & corporate values*, and *project constraints & system aspects*. Furthermore, we also identified quality aspects that are present in all dimensions. More details on the individual dimensions will be given in the next section, when we discuss the catalogue since the dimensions are closely linked to the quality aspects they frame. The dimensions and their respective quality aspects are illustrated in Fig. 2. In the figure, the quality aspects are grouped according to similarity, based on our workshops' results. Overall, the model should support requirements engineers in understanding how explainability can affect a system, facilitating requirements analysis.

## VI. A CATALOGUE OF EXPLAINABILITY'S IMPACTS

In what follows, we will present the catalogue and discuss the quality aspects in relation to our model. To this end, we will analyze them, whenever possible, based on the three categorizations we have described above: the stakeholders involved, the dimensions that affect the elicitation and analysis of explainability, and the external/internal categorization.

**Answering RQ3:** We built a catalogue that lists all quality aspects found in our study and the kind of impact that explainability has on each one of these aspects (Fig. 3).

### A. Foundational Qualities

Explainability can influence two quality aspects that have a crucial role: **transparency** and **understandability**. These quality aspects provide a foundation for all four dimensions, thereby having an influence on the other aspects inside these dimensions. Receiving explanations about a system, its processes and outputs can facilitate understanding on many levels [53]. Furthermore, explanations contribute to a higher system transparency [54]. For instance, understandability and transparency are required on a more external dimension so that users understand the outputs of a system, which may positively impact user experience. They are also important on a more internal dimension, where they can contribute to understanding aspects of the code, facilitating debugging and maintainability.

### B. User's Needs

Most papers concerning stakeholders in Explainable Artificial Intelligence (XAI) state *users* as a common class of stakeholders (e.g., [29], [55], [56]). This, in turn, also coincides with the view from requirements engineering, where (end) users also count as a common class of stakeholders [57]. Among others, users take into account recommendations of AI systems to make decisions [12]. Members of this stakeholder class can be medical doctors, loan officers, judges, or hiring managers. Usually, users are no experts regarding the technical details and the functioning of the systems they use [30].

When explainability is integrated into a system, different groups of users will certainly have different expectations, experiences, personal values, preferences, and needs. Such

aspects mean that individuals can perceive quality differently. At the same time, explainability influences aspects that are extremely important from a user perspective.

The quality aspects we have associated with users are mostly external. In other words, they are not qualities that depend solely on the system. To be more precise, they depend on the expectations and the needs of the person who uses the system.

On a general level, the **user experience** can both profit and suffer from explainability. Explanations can foster a sense of familiarity with the system [58] and make it more engaging [59]. In this case, user experience profits from explainability. On the other side, explanations can cause emotions such as confusion, surprise [60], and distraction [49], harming the user experience. Furthermore, explainability has a positive impact on the **mental-model accuracy** of involved parties. By giving explanations, it is possible to make users aware of the system's limitations [46], helping them to develop better mental models of it [60]. Explanations may also increase a user's ability to predict a decision and calibrate expectations with respect to what a system can or cannot do [46]. This can be attributed to an improved **user awareness** about a situation or about the system [61]. Furthermore, explanations about data collection, use, and processing allow users to be aware of how the system handles their data. Thus, explainability may be a way to improve **privacy awareness** [29]. Explainability can also positively impact the **perceived usefulness** of a system or a recommendation [62], which contributes to the **perceived value** of a system, increasing users' perception of a system's competence [63] and integrity [64] and leading to more positive attitudes towards the system [65]. Finally, all of this shows that explainability can certainly positively impact **user satisfaction** with the system [60].

Explainability can also influence the **usability** of a system. On the positive side, explanations can increase the ease of use of a system [25], lead to more efficient use [61], and make it easier for users to find what they want [66]. On the negative side, explanations can overwhelm users with excessive information [67] and can also impair the user interface design [11]. Explanations can help to improve **user performance** on problem solving and other tasks [64]. Another plausible positive impact of explainability is on **user effectiveness** [68]. With explanations, users may experience greater accuracy in decision-making by understanding more about a recommended option or product [69]. However, user effectiveness can also suffer when explanations lead users to agree with incorrect system suggestions [6]. **User efficiency** is another quality aspect that can be positively and negatively influenced by explainability. Analyzing and understanding explanation takes time and effort [70], possibly reducing user efficiency. Overall, however, the time needed to make a judgment could also be reduced with complementary information [68], increasing user efficiency. Furthermore, explanations may also give users a greater sense of **control**, since they understand the reasons behind decisions and can decide whether they accept an output or not [13]. Explainability can also have a positive influence on **human-machine cooperation** [48] since explanations may



Quality Aspect	Literature	Expert	Quality Aspect	Literature	Expert	Quality Aspect	Literature	Expert
Accountability	+	+	Knowledge Discovery	+	+	Support Decision Making	+	+
Accuracy	+ -	+	Learnability	+	+	System Acceptance	+	+
Adaptability		-	Maintainability		+ -	Testability	+	
Auditability	+	+	Mental Model Accuracy	+	+	Trade Secrets	-	-
Complexity		-	Model Optimization	+	+	Transferability	+	
Compliance	+	+	Perceived Usefulness	+	+	Transparency	+	+
Confidence in the System	+ -	+ -	Perceived Value	+	+	Trustworthiness	+	+
Correctness	+	+	Performance	+ -	-	Understandability	+ -	+
Customer Loyalty	+	+	Persuasiveness	+	+	Usability	+ -	+ -
Debugging	+	+	Portability		+ -	User Awareness	+	+
Decision Justification	+	+	Predictability	+		User Control	+	+
Development Cost	-	-	Privacy	+ -	-	User Effectiveness	+ -	+
Effectiveness	+		Privacy Awareness	+		User Efficiency	+ -	
Efficiency	-		Real-Time Capability		-	User Experience	+ -	+ -
Ethics	+	+	Reliability	+	+	User Performance	+	
Extensibility		-	Robustness	+	+	User Satisfaction	+	
Fairness	+	+	Safety	+	+ -	Stakeholder Trust	+ -	+
Guidance	+	+	Scrutability	+		Validation	+	+
Human-Machine Cooperation	+	+	Security	+ -	-	Verifiability	+	+

+ positively influenced by explainability - negatively influenced by explainability

Fig. 3. The knowledge catalog for explainability: how explainability impacts other quality aspects.

provide a more effective interface for humans [71], improving interactivity and cooperation [22], which can be especially advantageous in the case of cyber-physical systems.

Explainability can have a positive influence on **learnability**, allowing users to learn about how a system works or how to use a system [69]. It may also provide **guidance**, helping users in solving problems and educating them about product knowledge [72]. As these examples illustrate, explanations can **support decision-making** processes for users [25]. In some cases, this goes as far as enabling **scrutability** of a system, that is, enabling a user to provide feedback on a system's user model so that the system can give more valuable outputs or recommendations in the future [25]. Finally, explainability can help **knowledge discovery** [13]. By making the decision patterns in a system comprehensible, knowledge about the corresponding patterns in the real world can be extracted. This can provide a valuable basis for scientific insight [46].

### C. Cultural Values & Laws and Norms

Although [11] distinguished *Cultural Values* and *Laws and Norms* as two separate dimensions and [30] did the same for regulators and affected parties, we have combined them into one dimension because they are complementary and influence each other. The dimensions form a kind of symbiosis since, e.g., legal foundations are grounded, among others, on the basis of the cultural values of a society. We adopt the same approach for the dimensions discussed in Sec. VI-D and Sec. VI-E.

Regulators commonly envision laws for people who could be affected by certain practices. In other words, regulators stipulate legal and ethical norms for the general use, deployment, and development of systems. This class of stakeholders occupies an extraordinary role, since they have a 'watchdog' function

concerning the systems and their use [30]. Regulators can be ethicists, lawyers, and politicians, who must have the know-how to assess, control, and regulate the whole process of developing and using systems.

The restrictive measures by regulators are necessary, as the influence of systems is constantly growing and key decisions about people are increasingly automated – often without their knowing [30]. Affected parties are (groups of) people in the scope of a system's impact. They are stakeholders, as for them much hinges on the decision of a system. Patients, job or loan applicants, or defendants at court are typical examples of this stakeholder class [30].

In this dimension, cultural values represent the ethos of a society or group and influence the need for specific system qualities and how they should be operationalized [73], [74]. These values resonate in the conception of laws and norms, which enforce constraints that must be met and granted in the design of systems. Explainability can influence key aspects on this dimension.

With regard to the internal/external distinction, a clear attribution is not possible. Rather, the quality aspects seem to occupy a hybrid position. Whether or not they are present does not only depend on the system itself, but it also does not depend on a person using them. Rather, it depends on general conventions (e.g., legal, societal) that are in place.

On the cultural side, explanations can contribute to the achievement of **ethical** decision-making [75] and, more specifically, ethical AI. On the one hand, explaining the agent's choice may support ensuring that ethical decisions are made [13]. On the other hand, providing explanations can be seen as an ethical aspect itself. Furthermore, explainability may also contribute to **fairness**, enabling the identification of harms and

decision biases to ensure fair decision-making [13], or helping to mitigate decision biases [46].

On the legal side, explainability can promote a system's **compliance** with regulatory and policy goals [76]. Explaining an agent's choice can ensure that legal decisions are made [13]. A closely related aspect is **accountability**. We were able to identify a positive impact of explainability on this quality that occurs when explanations allow entities to be made accountable for a certain outcome [77]. In the literature, many authors refer to this as *liability* [77] or *legal accountability* [78].

In order to guarantee a system's adherence to cultural and legal norms, regulators and affected parties need several mechanisms that allow for inspecting systems. One NFR that can help in this regard is **auditability**. Explainability positively impacts this NFR, since explanations can help to identify whether a system made a mistake [6], can help to understand the underlying technicalities and models [44], and allow users to inspect a system's inner workings to judge whether it is acceptable or not [79]. In a similar manner, **validation** can be positively impacted, since explainability makes it possible for users to validate system knowledge [69] or assess if a recommended alternative is truly adequate for them [25]. Exactly the latter aspect is essential for another quality that is helped by explainability, namely, **decision justification**. On the one hand, explanations are a perfect way to justify a decision [77]. On the other hand, they can also help to uncover whether a decision is actually justified [4].

#### D. Domain Aspects & Corporate Values

People who decide where to employ certain systems (e.g., a hospital manager decides to bring a special kind of diagnosis system into use in her hospital) are deployers. Other possible stakeholders in this dimensions are specialists in the domain, known as domain experts. People have to work with the deployed systems and, consequently, new people fall inside the range of affected people [30].

This dimension is shaped by two aspects: 1) the corporate values and vision of an organization [80], and 2) the domain aspects that shape a system's design since explanations may be more urgent in some domains as in others.

We consider this dimension as more internal to the system, since it encompasses quality aspects that are more related to the domain or the values of the corporation or the team. Generally, the integration of such aspects affects the design of a system on an architectural level. However, there are some exceptions, as the organization's vision may aim at external factors like customer loyalty.

Explainability supports the **predictability** of a system by making it easier to predict a system's performance correctly and helping to determine when a system might make a mistake [81]. Furthermore, explainability can support the **reliability** of a system [41]. In general, explainability supports the development of more **robust** systems for critical domains [82]. All of this contributes to a positive impact on **safety**, helping to meet safety standards [13], or helping to create safer systems [83].

On the negative side, explanations may also present safety risks by distracting users in critical situations.

Explanations are also seen as a means to bridge the gap between perceived **security** and actual security [42], helping users to understand the actual mechanisms in systems and adapt their behavior accordingly. However, explanations may disclose information that makes the system vulnerable to attack and gaming [3]. Explainability can also influence **privacy** positively, since the principle of information disclosure can help users to discover what features are correlated with sensitive information that can be removed [84]. By the same principle, however, privacy can be hurt since one may need to disclose sensitive information that could jeopardize privacy [61]. Explainability can also threaten model confidentiality and **trade secrets**, which companies are reluctant to reveal [29].

Explainability can contribute to **persuasiveness**, since explanations may increase the acceptance of a system's decisions and the probability that users adopt its recommendations [25]. Furthermore, explainability influences **customer loyalty** positively, since it supports the continuity of use [58] and may inspire feelings of loyalty towards the system [66].

#### E. Project Constraints & System Aspects

Individuals who design, build, and program systems are, among others, developers, quality engineers, and software architects. They count as stakeholders, as without them the systems would not exist in the first place. Generally, representatives of this group have a high expertise concerning the systems and a strong interest in creating and improving them.

This dimension is shaped by two aspects: project constraints and system aspects. The project constraints are the non-technical aspects of a system [85], while system aspects are more related to internal aspects of the system, such as performance and maintainability.

The quality aspects framed in this dimension are almost entirely internal in the classical sense, since they correspond to the most internal aspects of a system or the process through which the system is built.

Explainability can have both a positive and negative impact on **maintainability**. On the one hand, it can facilitate software maintenance and evolution by giving information about models and system logic. On the other hand, the ability to generate explanations requires new components in a system, hampering maintenance. A positive impact on **verifiability** was also identified, when explanations can work as a means to ensure the correctness of the knowledge base [69] or to help users evaluate the accuracy of a system's prediction [86]. **Testability** falls in the same line, since explanations can help to evaluate or test a system or a model [13]. Explainability has a positive influence on **debugging**, as explanations can help developers to identify and fix bugs [4]. Specifically, in the case of ML applications, this could enable developers to identify and fix biases in the learned model and, thus, **model optimization** is positively affected [28]. Overall, all these factors can help increase the **correctness** of a system, by helping to correct errors in the system or in model input data [77].



The overall **performance** of a system can be affected both positively and negatively by explainability. On the one hand, explanations can positively influence the performance of a system by helping developers to improve the system [48]. In this regard, explainability positively influences system **effectiveness**. On the other hand, however, explanations can also lead to drawbacks in terms of performance [70] by requiring loading time, memory, and computational cost [11]. Thus, as the additional explainability capacities are likely to require computational resources, the **efficiency** of the system might decrease [4]. Another quality that is impacted by explainability is **accuracy**. For instance, in the ML domain, the accuracy of models can benefit from explainability through model optimization [28]. On the negative side, there exists a trade-off between the predictive accuracy of a model and explainability [4]. A system that is inherently explainable, for instance, may have to sacrifice predictive power in order to be so [43]. Explainability may have a negative impact on **real-time capability** since the implementation of explanations could require more computing power and additional processes, such as logging data, might be involved.

**Adaptability** can be negatively impacted, for example, if lending regulations in a financial software have changed and an explanation module in the software is also affected. Next, assume that a new module should be added to a system. The quality aspect involved here is **extensibility**, which in turn is negatively impacted by explainability. Merely adding the new module is already laborious. If the explainability is also affected by this new module, the required effort increases again. Depending on the architecture of the software, it may even be impossible to preserve the system's explainability. Explanations affect the **portability** of a system as well. On the negative side, an explanation component might not be ported directly because it uses visual explanations, but the environment to which system is to be ported to has no elements that allow for visual outputs. On the positive side, explainability helps **transferability** [87]. Transferability is the possibility to transfer a learned model from one context to another (thus, it can be seen as a special case of portability for ML applications). Explanations may help in this regard by making it possible to identify the context from and to which the model can be transferred [87].

Overall, the inclusion of explanation modules can increase the **complexity** of the system and its code, influencing many of the previously seen quality aspects. In particular, as an explainability component needs additional development effort and time, it can result in higher **development costs** [5].

#### *F. Superordinated Qualities*

We were able to identify some aspects that hold regardless of dimension. These aspects are commonly seen as some kind of superordinated goals of explainability. For instance, organizations and regulators have been lately focusing on defining core principles (or "pillars") for responsible or **trustworthy** AI. Explainability has been often listed as one of these pillars [29]. Overall, many of the quality aspects we could find in the literature contribute to trustworthiness. For

instance, explanations can help to identify whether a system is safe and whether it complies to legal or cultural norms. Ideally, **confidence** and **trust** in a system originate solely from trustworthy systems. Although one could trust an untrustworthy system, this trust would be unjustified and inadequate. For this reason, explainability can both contribute to and hurt trust or confidence in a system [42], [61]. Regardless of the system's actual trustworthiness, bad explanations can always degrade trust [42]. Finally, all of this can influence the system's **acceptance**. A system that is trustworthy can gain acceptance [45] and explainability is key to this.

## VII. DISCUSSION

Explainability is a new NFR that echoes the demand for more human oversight of systems [9]. It can bring positive or negative consequences across all quality dimensions: from users' needs to system aspects. Explainability's impact on so many crucial dimensions illustrates the growing need to take explainability into account while designing a system. Currently, however, the RE community still lacks guidance on how to do so. Building appropriate elicitation techniques and developing adequate tools to capture explainability requirements are challenges that still need to be addressed.

To this end, our first contribution is a helpful definition of explainability for software and requirements engineers (Sec. IV). This definition points out what should be considered when dealing with requirements and the appropriate functionality for explainable systems: aspects that should be explained, contexts, explainers, and addressees. Being aware of these variables facilitates the software development process, supporting the elicitation and specification of explainability requirements. In this sense, the possible values (e.g., reasoning process) we found in the literature can serve as an abstract starting point during requirements analysis. Overall, our definition can serve as a template to help engineering explainable systems and to make good design choices towards explainability requirements.

In contrast, poor design choices regarding explainability can negatively affect the relationship with the user (e.g., user experience issues), interfere with important quality aspects for a corporation (e.g., damaging brand image and customer loyalty), and bring disadvantages for the project or the system (e.g., increasing development costs or hindering system performance). This kind of impact may stem from the fact that explainability might be seen as an aspect of communication between systems and humans. Depending on how it happens in practice, communication can either strengthen or harm relationships.

Research in RE can profit from insights of other disciplines when it comes to explainability. The fields of philosophy, psychology, and HCI, for example, have long researched aspects such as explanations or human interaction with systems (see [49] for research concerning explanations in several disciplines). At the same time, requirements engineers can contribute to the field of explainability by studying how to include such aspects in systems and adapt development processes. This knowledge, scattered among different areas of knowledge, must be made available and integrated into the development of systems.

To this end, additional contributions of this paper are a model (Sec. V) and a knowledge catalogue (Sec. VI). Conceptual models are useful to abstract, comprehend, and communicate information. Among others, our catalogue can serve as checklist during elicitation and also during trade-off analysis. It can help software engineers avoid conflicts between quality aspects and choose the best strategies for achieving the desired quality outcomes. Both artifacts contain information that may be used to turn explainability into a positive catalyst for other essential system qualities in modern systems.

On a general level, building these artifacts has revealed that there is much to do in the field of NFRs. On the one hand, we believe that there may be other emerging NFRs besides explainability. Aspects such as human-machine cooperation, privacy awareness, and mental model accuracy show that there are specific needs that should be better understood when developing modern systems. Furthermore, ethics, fairness, and legal compliance are all good examples of quality aspects that are gaining in importance and should be better researched [88].

On the other hand, we have identified that explainability can exhibit an impact on nearly all traditional NFRs that can be found in the ISO 25010 [50]: performance, efficiency, usability, reliability, security, maintainability, and portability. As such, the importance of explainability has to be further acknowledged. In this line of thought, the impact of other NFRs on explainability should be better researched and existing catalogues could be updated to incorporate explainability. The RE community needs to explore what kind of activities, methods, and tools need to be incorporated into the software development process in order to accommodate the necessary steps towards building explainable systems. Our work is an essential step in this direction.

### VIII. LIMITATIONS AND THREATS TO VALIDITY

Our work is exclusively based on qualitative data analysis. Consequently, there is the possibility that the results are affected by subjectivity during analysis. Therefore, we decided on a multi-method approach to produce results that are more robust and compelling than single method studies. Next, we discuss the main threats to validity in each part of our research.

*a) SLR and coding:* The review process assumed a common understanding among all researchers involved in this work with respect to the search and analysis methods used. Results could be subject to bias if the methods and concepts are misunderstood. We mitigated this threat by elaborating a review protocol and discussing it before starting the review to reach a good level of shared understanding. We have formulated inclusion and exclusion criteria to reduce biases due to subjective decisions in the selection process. Some criteria, such as the publication period, are objective, while others, focusing on the content of the papers, are still subjective. To decrease the amount of researcher bias, we conducted the analysis independently. For both the literature review and the coding process, in case of disagreement, the decision on inclusion or exclusion (for a paper) or the code assignment (for the extracted data) was taken by all researchers and validated by the Fleiss' Kappa statistic.

*b) Explainability catalogue:* The clustering and categorization of the quality aspects into their different dimensions was prone to subjective judgment. As steps to mitigate this, we rooted this categorization on well-known concepts present in the literature and conducted workshops with experts. This allowed us to inspect our clustering through internal and external reviews. During the internal reviews, the categorization was discussed among the authors to clarify ambiguities and reach agreements. During the external reviews, we compared the findings from the literature with expert knowledge. Due to these review processes, we are confident to have achieved a proper level of validity of the catalogue. Moreover, as researchers we are confident that both our catalogue and model are reasonably accurate for the field studied, developed over debates that formed our shared knowledge on the subject.

### IX. CONCLUSION AND FUTURE WORK

Explainability is increasingly seen as an appropriate means of achieving essential quality aspects in a system, such as transparency, accountability, and trustworthiness. As building these values into our systems becomes more urgent, there is a need for tools and methods that help elicit, implement, and validate related requirements. For this reason, we should be concerned with understanding explainability as a whole: its meaning, its effects, its taxonomy.

In this sense, our proposed definition can help to facilitate communication and align expectations when referring to explainability. Our model can help professionals to understand its taxonomy, and our catalogue can help to identify conflicts between explainability and other important qualities. In holding this knowledge, it is possible to think of design strategies and implementation level solutions that result in positive effects for the stakeholders involved.

As a next step, we want to create a quality model that structures and expands the gathered knowledge with specific characteristics and aspects of explanations. Furthermore, we need to investigate what kinds of explainability-related activities should be integrated into the software development process to successfully develop explainable systems. Overall, we hope that our work lays the foundation for the RE community to better understand and investigate the topic of explainability.

### ACKNOWLEDGMENTS

This work was supported by the research initiative Mobilise between the Technical University of Braunschweig and Leibniz University Hannover, funded by the Ministry for Science and Culture of Lower Saxony and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122, Project ID 390833453). Work on this paper was also funded by the Volkswagen Foundation grant AZ 98514 "Explainable Intelligent Systems" (EIS) and by the DFG grant 389792660 as part of TRR 248. We thank Martin Glinz for his feedback on our research design. Furthermore, we thank all workshop participants, the anonymous reviewers, and the colleagues who gave feedback on our manuscript.

## REFERENCES

- [1] L. Chazette, W. Brunotte, and T. Speith, "Supplementary Material for Research Paper "Exploring Explainability: A Definition, a Model, and a Knowledge Catalogue"," Jul. 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.5114922>
- [2] A. Panesar, "Ethics of intelligence," in *Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes*. Apress, 2019, pp. 207–254.
- [3] B. Lepri, N. Oliver, E. Letouzé, A. Pentland, and P. Vinck, "Fair, transparent, and accountable algorithmic decision-making processes," *Philosophy & Technology*, vol. 31, no. 4, pp. 611–627, 2018.
- [4] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.
- [5] M. A. Köhl, K. Baum, M. Langer, D. Oster, T. Speith, and D. Bohlender, "Explainability as a non-functional requirement," in *27th IEEE International Requirements Engineering Conference (RE)*. IEEE, 2019, pp. 363–368.
- [6] A. Bussone, S. Stumpf, and D. O'Sullivan, "The role of explanations on trust and reliance in clinical decision support systems," in *2015 International Conference on Healthcare Informatics*. IEEE, 2015, pp. 160–169.
- [7] J. P. Winkler and A. Vogelsang, "'What does my classifier learn?' A visual approach to understanding natural language text classifiers," in *Natural Language and Information Systems*, F. Frasincar, A. Ittoo, L. M. Nguyen, and E. Métais, Eds., 2017, pp. 468–479.
- [8] B. Abdollahi and O. Nasraoui, "Transparency in fair machine learning: the case of explainable recommender systems," in *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*. Springer, 2018, pp. 21–35.
- [9] S. Thiebes, S. Lins, and A. Sunyaev, "Trustworthy artificial intelligence," *Electronic Markets*, pp. 1–18, 2020.
- [10] M. Glinz, "On non-functional requirements," in *15th IEEE International Requirements Engineering Conference (RE)*, 2007, pp. 21–26.
- [11] L. Chazette and K. Schneider, "Explainability as a non-functional requirement: challenges and recommendations," *Requirements Engineering*, vol. 25, no. 4, pp. 493–514, 2020.
- [12] M. Hind, D. Wei, M. Campbell, N. C. F. Codella, A. Dhurandhar, A. Mojsilović, K. Natesan Ramamurthy, and K. R. Varshney, "TED: Teaching AI to explain its decisions," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2019, pp. 123–129.
- [13] A. Rosenfeld and A. Richardson, "Explainability in human-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 673–705, 2019.
- [14] D. Mairiza and D. Zowghi, "Constructing a catalogue of conflicts among non-functional requirements," in *Evaluation of Novel Approaches to Software Engineering*. Springer, 2011, pp. 31–44.
- [15] L. M. Cysneiros, "Evaluating the effectiveness of using catalogues to elicit non-functional requirements," in *Workshop em Engenharia de Requisitos (WER 2007)*, 2007, pp. 107–115.
- [16] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering*. Springer Science & Business Media, 2012, vol. 5.
- [17] R. Gacitúa, L. Ma, B. Nuseibeh, P. Piwek, A. N. D. Roeck, M. Rouncefield, P. Sawyer, A. Willis, and H. Yang, "Making tacit requirements explicit," in *Second International Workshop on Managing Requirements Knowledge (MARK@RE)*. IEEE, 2009, pp. 40–44.
- [18] F. Deissenboeck, E. Jürgens, K. Lochmann, and S. Wagner, "Software quality models: Purposes, usage scenarios and requirements," in *2009 ICSE Workshop on Software Quality (WoSQ@ICSE)*. IEEE, 2009, pp. 9–14.
- [19] P. Gutmann and I. Grigg, "Security usability," *IEEE security & privacy*, vol. 3, no. 4, pp. 56–58, 2005.
- [20] R. L. Q. Portugal, T. Li, L. Silva, E. Almentero, and J. C. S. do Prado Leite, "NFRfinder: a knowledge based strategy for mining non-functional requirements," in *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. ACM, 2018, pp. 102–111.
- [21] M. Serrano and M. Serrano, "Ubiquitous, pervasive and mobile computing: A reusable-models-based non-functional catalogue," in *Proceedings of Requirements Engineering@Brazil*, vol. 1005. CEUR, 2013.
- [22] R. M. Carvalho, R. M. C. Andrade, V. Lelli, E. G. Silva, and K. M. de Oliveira, "What about catalogs of non-functional requirements?" in *Proceedings of REFSQ-2020 Workshops*, vol. 2584. CEUR, 2020.
- [23] R. C. Torres and L. E. G. Martins, "NFR catalogues for RFID middleware," *Journal of Computer Science and Technology*, vol. 14, no. 02, pp. 102–108, 2018.
- [24] R. M. Carvalho, R. M. C. Andrade, and K. M. Oliveira, "How developers believe invisibility impacts NFRs related to user interaction," in *28th IEEE International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 102–112.
- [25] I. Nunes and D. Jannach, "A systematic review and taxonomy of explanations in decision support and recommender systems," *User Modeling and User-Adapted Interaction*, vol. 27, no. 3-5, pp. 393–444, 2017.
- [26] S. Anjomshoaie, A. Najjar, D. Calvaresi, and K. Främling, "Explainable agents and robots: Results from a systematic literature review," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, p. 1078–1088.
- [27] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, "Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda," in *Proceedings of the 2018 Conference on Human Factors in Computing Systems (CHI)*. ACM, 2018, pp. 1–18.
- [28] S. M. Mathews, "Explainable artificial intelligence applications in NLP, biomedical, and malware classification: A literature review," in *Intelligent Computing – Proceedings of the Computing Conference*. Springer, 2019, pp. 1269–1292.
- [29] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bannetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.
- [30] M. Langer, D. Oster, T. Speith, H. Hermanns, L. Kästner, E. Schmidt, A. Sesing, and K. Baum, "What do we want from explainable artificial intelligence (XAI)? – a stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research," *Artificial Intelligence*, 2021.
- [31] J. C. S. do Prado Leite and C. Cappelli, "Software transparency," *Business & Information Systems Engineering*, vol. 2, no. 3, pp. 127–139, 2010.
- [32] J. Saldaña, *The Coding Manual for Qualitative Researchers*. SAGE Publications, 2021.
- [33] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University, Tech. Rep., 2007.
- [34] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 2014, pp. 1–10.
- [35] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76, no. 5, pp. 378–382, 1971.
- [36] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.
- [37] J. F. Wolfswinkel, E. Furtmueller, and C. P. M. Wilderom, "Using grounded theory as a method for rigorously reviewing literature," *European Journal of Information Systems*, vol. 22, no. 1, pp. 45–55, 2013.
- [38] K. Charmaz, *Constructing grounded theory: A practical guide through qualitative analysis*. SAGE Publications, 2006.
- [39] M. B. Miles and A. M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*. SAGE Publications, 1994.
- [40] R. E. Boyatzis, *Transforming Qualitative Information: Thematic Analysis and Code Development*. SAGE Publications, 1998.
- [41] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019.
- [42] W. Pieters, "Explanation and trust: what to tell the user in security and AI?" *Ethics and Information Technology*, vol. 13, no. 1, pp. 53–64, 2011.
- [43] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller, "Causability and explainability of artificial intelligence in medicine," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 4, pp. 1–13, 2019.
- [44] J. Hois, D. Theofanous-Fuelbier, and A. J. Junk, "How to achieve explainability and transparency in human AI interaction," in *International*

- Conference on Human-Computer Interaction (HCI). Springer, 2019, pp. 177–183.
- [45] A. Glass, D. L. McGuinness, and M. Wolverson, “Toward establishing trust in adaptive agents,” in *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI)*. ACM, 2008, pp. 227–236.
  - [46] Q. V. Liao, D. M. Gruen, and S. Miller, “Questioning the AI: informing design practices for explainable AI user experiences,” in *Proceedings of the 2020 Conference on Human Factors in Computing Systems (CHI)*. ACM, 2020, pp. 1–15.
  - [47] P. Dourish, “What we talk about when we talk about context,” *Personal and Ubiquitous Computing*, vol. 8, no. 1, pp. 19–30, 2004.
  - [48] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should I trust you?’: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144.
  - [49] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.
  - [50] ISO Central Secretary, “ISO/IEC 25010:2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuARE) - System and Software Quality Models,” International Organization for Standardization, Standard ISO/IEC 25010:2011, 2011. [Online]. Available: <https://www.iso.org/standard/35733.html>
  - [51] S. Freeman and N. Pryce, *Growing Object-Oriented Software, Guided by Tests*. Addison-Wesley, 2009.
  - [52] S. McConnell, *Code Complete*. Microsoft Press, 2004.
  - [53] C. Henin and L. M. Daniel, “Towards a generic framework for black-box explanation methods,” in *Proceedings of the IJCAI Workshop on Explainable Artificial Intelligence (XAI)*, 2019, pp. 28–34.
  - [54] L. Chen, D. Yan, and F. Wang, “User evaluations on sentiment-based recommendation explanations,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 9, no. 4, pp. 1–38, 2019.
  - [55] A. D. Preece, D. Harborne, D. Braines, R. Tomsett, and S. Chakraborty, “Stakeholders in explainable AI,” *CoRR*, vol. abs/1810.00184, 2018.
  - [56] A. Weller, “Transparency: Motivations and challenges,” in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Eds. Springer, 2019, ch. 2, pp. 23–40.
  - [57] M. Glinz and R. J. Wieringa, “Guest editors’ introduction: Stakeholders in requirements engineering,” *IEEE Software*, vol. 24, no. 2, pp. 18–20, 2007.
  - [58] M. O. Riedl, “Human-centered artificial intelligence and machine learning,” *Human Behavior and Emerging Technologies*, vol. 1, no. 1, pp. 33–36, 2019.
  - [59] J. McNerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, and R. Mehrotra, “Explore, exploit, and explain: personalizing explainable recommendations with bandits,” in *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys)*. ACM, 2018, pp. 31–39.
  - [60] C. J. Cai, J. Jongejan, and J. Holbrook, “The effects of example-based explanations in a machine learning interface,” in *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI)*. ACM, 2019, pp. 258–262.
  - [61] J. Zhou, H. Hu, Z. Li, K. Yu, and F. Chen, “Physiological indicators for user trust in machine learning with influence enhanced fact-checking,” in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2019, pp. 94–113.
  - [62] M. Zanker, “The influence of knowledgeable explanations on users’ perception of a recommender system,” in *Proceedings of the sixth ACM conference on Recommender systems (RecSys)*. ACM, 2012, pp. 269–272.
  - [63] P. Pu and L. Chen, “Trust building with explanation interfaces,” in *Proceedings of the 11th International Conference on Intelligent User Interfaces (IUI)*. ACM, 2006, pp. 93–100.
  - [64] R. F. Kizilcec, “How much information? Effects of transparency on trust in an algorithmic interface,” in *Proceedings of the 2016 Conference on Human Factors in Computing Systems (CHI)*. ACM, 2016, pp. 2390–2395.
  - [65] H. Cramer, V. Evers, S. Ramlal, V. S. Maarten, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga, “The effects of transparency on trust in and N. Tintarev and J. Masthoff, “Effective explanations of recommendations: user-centered design,” in *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys)*. ACM, 2007, pp. 153–156.
  - acceptance of a content-based art recommender,” *User Modeling and User-adapted interaction*, vol. 18, no. 5, p. 455, 2008.
  - [67] C. Tsai and P. Brusilovsky, “Explaining recommendations in an interactive hybrid social recommender,” in *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI)*. ACM, 2019, pp. 391–396.
  - [68] N. Tintarev and J. Masthoff, “Evaluating the effectiveness of explanations for recommender systems,” *User Modeling and User-Adapted Interaction*, vol. 22, no. 4–5, pp. 399–439, 2012.
  - [69] K. Darlington, “Aspects of intelligent systems explanation,” *Universal Journal of Control and Automation*, vol. 1, no. 2, pp. 40–51, 2013.
  - [70] P. S. Kumar, M. Saravanan, and S. Suresh, “Explainable classification using clustering in deep learning models,” in *Proceedings of the IJCAI Workshop on Explainable Artificial Intelligence (XAI)*, 2019, pp. 115–121.
  - [71] J. Dodge, Q. V. Liao, Y. Zhang, R. K. E. Bellamy, and C. Dugan, “Explaining models: an empirical study of how explanations impact fairness judgment,” in *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI)*. ACM, 2019, pp. 275–285.
  - [72] V. Putnam and C. Conati, “Exploring the need for explainable artificial intelligence (XAI) in intelligent tutoring systems (ITS),” in *Joint Proceedings of the ACM IUI 2019 Workshops*. CEUR, 2019.
  - [73] A. Pacey, *The Culture of Technology*. MIT Press, 1983.
  - [74] T.-F. Kummer, J. M. Leimeister, and M. Bick, “On the importance of national culture for the design of information systems,” *Business & Information Systems Engineering*, vol. 4, no. 6, pp. 317–330, 2012.
  - [75] J. Schneider and J. Handali, “Personalized explanation in machine learning: A conceptualization,” *Proceedings of the 27th European Conference on Information Systems (ECIS)*, 2019.
  - [76] L. H. Gilpin, C. Testart, N. Fruchter, and J. Adebayo, “Explaining explanations to society,” in *NIPS Workshop on Ethical, Social and Governance Issues in AI*, 2018, pp. 1–6.
  - [77] I. Monteath and R. Sheh, “Assisted and incremental medical diagnosis using explainable artificial intelligence,” in *Proceedings of the IJCAI/ECAI Workshop on Explainable Artificial Intelligence (XAI)*, 2018, pp. 104–108.
  - [78] R. Binns, M. Van Kleek, M. Veale, U. Lyngs, J. Zhao, and N. Shadbolt, “‘It’s reducing a human being to a percentage’: Perceptions of justice in algorithmic decisions,” in *Proceedings of the 2018 Conference on Human Factors in Computing Systems (CHI)*. ACM, 2018, pp. 1–14.
  - [79] K. McCarthy, J. Reilly, L. McGinty, and B. Smyth, “Thinking positively-explanatory feedback for conversational recommender systems,” in *Proceedings of the European Conference on Case-Based Reasoning (ECCBR) Explanation Workshop*, 2004, pp. 115–124.
  - [80] S. Thomsen, “Corporate values and corporate governance,” *Corporate Governance*, vol. 4, no. 4, pp. 29–46, 2004.
  - [81] I. Lage, D. Lifschitz, F. Doshi-Velez, and O. Amir, “Exploring computational user models for agent policy summarization,” in *Proceedings of the IJCAI Workshop on Explainable Artificial Intelligence (XAI)*, 2019, pp. 59–65.
  - [82] R. Borgo, M. Cashmore, and D. Magazzeni, “Towards providing explanations for AI planner decisions,” *Proceedings of the IJCAI/ECAI Workshop on Explainable Artificial Intelligence (XAI)*, pp. 11–17, 2018.
  - [83] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, “A survey of methods for explaining black box models,” *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, 2019.
  - [84] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker, “Gamut: A design probe to understand how data scientists understand machine learning models,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. 1–13.
  - [85] J. P. Carvallo, X. Franch, and C. Quer, “Managing non-technical requirements in cots components selection,” in *14th IEEE International Requirements Engineering Conference (RE)*. IEEE, 2006, pp. 323–326.
  - [86] J. Zhou and F. Chen, “Towards trustworthy human-AI teaming under uncertainty,” in *Proceedings of the IJCAI Workshop on Explainable Artificial Intelligence (XAI)*, 2019, pp. 143–147.
  - [87] J. Chen, F. Lécué, J. Z. Pan, I. Horrocks, and H. Chen, “Knowledge-based transfer learning explanation,” in *Proceedings of the Sixteenth International Conference for Principles of Knowledge Representation and Reasoning (KR)*. AAAI Press, 2018, pp. 349–358.
  - [88] F. B. Aydemir and F. Dalpiaz, “A roadmap for ethics-aware software engineering,” in *Proceedings of the International Workshop on Software Fairness (FairWare)*. ACM, 2018, p. 15–21.