Article

A Requirement Quality Assessment Method Based on User Stories

Xiangqian Xu [ID]**, Yajie Dou ***[ID]**, Liwei Qian** [ID]**, Zhiwei Zhang, Yufeng Ma and Yuejin Tan**

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China; xuxiangqian18@163.com (X.X.); qianliwei21@163.com (L.Q.); zhangzhiweizzw21@nudt.edu.cn (Z.Z.); mashuang9707@163.com (Y.M.); yjtan@nudt.edu.cn (Y.T.)
*   Correspondence: yajiedou_nudt@163.com

Abstract: Agile development processes based on user stories often face issues such as incomplete, inconsistent, and inaccurate user requirements, which increase the workload of agile development teams and reduce the efficiency of product function development, ultimately resulting in the inability to respond quickly to user requirements. This paper proposes a user requirement quality assessment method based on user stories to address these problems. This method relies on the agile development process, constructs a user requirement quality assessment framework, defines a user story model and a user requirement quality model, develops seven user requirement quality assessment criteria, and designs a user requirement quality assessment process. A data experiment exploring the development of smartphone requirements is used to validate the feasibility and effectiveness of the method. The experimental results demonstrate that the method improves user requirement quality to some extent, providing an automated solution for agile development teams to enhance user requirement quality.

Keywords: user requirements; quality assessment; user stories; agile development

1. Introduction

Agile development is a prominent research direction in requirements engineering. Recently, novel advancements in theory have emerged, while abundant practical experience has been obtained in various enterprise operations in the Internet and big data environments [1]. At its core, agile development is a development method that responds to quickly changing user requirements through iterative, small-scale, and rapid development and functionality delivery, ultimately allowing for a rapid response to user requirements and the continuous optimization of function design [2]. Compared to the traditional waterfall approach, agile development has a significant advantage in dealing with unclear or changing requirements, and can greatly minimize the risk of costly user requirement changes in the later stages of project development [3].

The successful implementation of agile development in a project relies on the efficient, rapid, and accurate flow of information between organizations or members [4]. Of all kinds of information, user requirements are of the utmost importance, serving as the core of agile development teams. Inaccurate requirements can impede progress, drastically impact the quality of the delivered product, and even damage the morale of the development team. Consequently, high-quality user requirements can be seen as a "catalyst" that provides a solid foundation in the early stages of a project, bolsters the positive feedback of iterative development, leads the entire project development process in a virtuous cycle, and ultimately results in a product that effectively meets user needs. Requirements play a crucial role in project development across various fields. For example, in research on optical wireless communication systems based on smartphone cameras, the technology requirements of smartphone cameras play a critical and leading role in generating solutions for optical camera communication [5]. In the field of the model-based design of cyber-physical systems, traceability from requirements to the model to the simulation results has

become increasingly important [6]. In the development of mobile learning applications, inappropriate technology requirements will affect the quality and increase the development cost of mobile learning applications [7]. Additionally, in the Internet environment, users can add their comments as feedback in the mobile application store, and this feedback can be considered a requirement and analyzed through natural language processing to improve the software quality and functionality of mobile applications [8,9].

In agile development, user stories are commonly used as the source of user requirements instead of traditional requirement specifications. A user story is composed of three elements, namely role, activity, and value, and is usually expressed as "What 'Activity' a 'role' wants to accomplish to achieve what 'Value'" [10,11]. By collecting a large number of user stories, agile development teams can quickly comprehend user requirements, allowing for rapid response and iteration. However, in many project practices, the agile development process with user stories as a tool is not always implemented smoothly [12,13]. This is mainly due to the wide source of user requirements, as well as different cultural backgrounds, abilities, and cooperation degrees of users, causing most of the collected user stories to be incomplete, inconsistent, and inaccurate, thereby reducing the usability of user requirement information and seriously affecting the efficiency of agile development.

The traditional approach to these problems involves setting guidelines prior to collecting user stories. The INVEST (independent, negotiable, valuable, estimable, small, testable) principles provide a framework for evaluating the quality of user stories [14]. Agile teams typically analyze user stories through manual review after collection and by filtering out qualified stories for development. While these strategies improve quality to some extent, they are impractical when the number of user stories is large. Investing too much time and effort in review can burden the agile team, impeding iteration speed and making it difficult to meet requirements on time. Thus, given the pragmatic requirements of agile development practices, it is critical to construct a scientific, automatic, and effective user requirement quality assessment method to assist agile teams in improving requirement quality and development efficiency.

The main contribution of this paper is the construction of a user requirement quality assessment method based on user stories in agile development. By constructing a user story model and combining it with the INVEST criteria, seven user requirement quality assessment criteria are established, and a user requirement quality model is generated. This achieves a transformation of the concept of "user requirement quality" from abstraction to concreteness and from qualitative to quantitative, providing an automated solution for user requirement quality assessment in agile development. Through a data experiment on smartphones, the feasibility and effectiveness of the method proposed in this paper are confirmed.

The remaining parts of the paper are organized as follows: related works are reviewed in Section 2; the framework of user requirement quality assessment is introduced in Section 3; the model design is detailed in Section 4; the experimental results are discussed in Section 5; the summary and prospects of this study are discussed in Section 6.

2. Related Works

As an important part of the entire project development lifecycle, requirement engineering plays a crucial role in project quality assurance. Requirement quality is considered one of the most significant factors in determining the success of project development. In recent years, a growing body of research has been devoted to the study of requirement quality. This section provides a review of recent research on requirement quality, with a focus on the definition of requirement quality, its assessment, and its relationship with project development outcomes.

(1)   Definition of requirement quality

Requirement quality refers to the degree to which requirements satisfy the needs of stakeholders and meet the goals of project development. In other words, high-quality requirements are those that clearly and unambiguously describe what the system needs to

carry out without any inconsistencies, conflicts, or omissions [15]. Quality requirements are typically characterized by completeness, consistency, correctness, clarity, and testability [16]. They should be able to provide a clear and precise specification of what a system should do, what it should not do, and how it should behave under different conditions [17]. The definition of requirement quality has been given new connotations through the continuous efforts of researchers. Calazans et al. [18] defined low-quality requirements as "requirements smells" and confirmed the existence of requirements smells in the quality requirement specifications classified by ISO/IEC 25010 by analyzing 26 institutional regulations of a large public financial institution. Eugenio et al. [19] extended the definition of requirement quality in the temporal dimension and developed an industrial method for analyzing the evolution of requirement quality, which is implemented in the system quality analyzer tool. Although the definition of requirement quality is constantly being refined and innovated, the applicability of requirement quality definitions has not yet been matched and modified in the field of agile development, and relevant mathematical models and assessment methods for requirement quality have not been established.

(2)   Assessment of requirement quality

Assessing requirement quality is an essential task in project development [20]. Numerous methods and techniques have been proposed for requirement quality assessment. These methods can be broadly classified into two categories: subjective and objective. Subjective methods involve human judgment and expert opinions, such as manual reviews [21,22], walkthroughs [23,24], and surveys [24,25]. Objective methods, on the other hand, employ automated tools and formal techniques, such as static analysis [26], natural language processing [27–30], machine learning [20,31], and deep learning [32]. Objective methods have gained increasing attention due to their advantages of automation, repeatability, and scalability. Wang et al. [33] proposed an approach to automatically detect defects in requirements. By utilizing structural, syntactic, and semantic analysis, this approach can automatically improve the quality of requirements. Valentin et al. [31] developed a novel and flexible approach for assessing requirement quality based on machine learning techniques that can be adapted to various environments, projects, organizations, and quality standards. Gregorius et al. [32] utilized deep learning methods to investigate the impact of diverse software domains on quality attribute prediction. This approach can assist users in identifying quality attributes in their requirement writing. Although these objective methods have improved the efficiency of requirement quality assessment, the evaluation results of requirement development teams are still important indicators that cannot be ignored in agile development practices. Therefore, it is necessary to combine subjective and objective requirement evaluation methods to improve the applicability and effectiveness of addressing requirement quality issues in agile development.

(3)   Relationship with project development outcomes

Requirement quality has a significant impact on project development outcomes, such as development costs, user satisfaction, and project success [30,34,35]. Low-quality requirements can lead to project development failures, project delays, cost overruns, and customer dissatisfaction [25,36]. High-quality requirements, on the other hand, can facilitate the development of systems that meet user needs, are easy to use, and have good performance and reliability [26,37]. Therefore, ensuring requirement quality is essential to achieving project development success. Eman et al. [25] conducted a survey of Egyptian software development practitioners through questionnaires and revealed the association between requirement quality and software development success rates. Furthermore, Amalinda et al. [30] evaluated the efficacy of IBM's requirement quality assistant tool in aiding the requirement quality assessment of Bosch requirement engineers, and suggested applicable improvement recommendations. Emil et al. [36] interviewed 20 project practitioners, including requirement engineers, developers, and testers, and found that when they dealt with requirements they considered to be of low quality, they experienced negative emotions, more work, and additional communication, which indirectly led to problems such as slow

project progress and low completion quality of development tasks. As seen, requirement quality is a key factor that influences project development outcomes. Researchers in agile development also need to focus on requirement quality issues, and targeted requirement quality assessment methods need to be proposed.

In general, there exists a plethora of research on requirement quality in requirements engineering, yet few studies have focused on modeling and analyzing the user requirement quality assessment process in agile development. Agile development is distinct from other requirement engineering approaches because it emphasizes faster response and iteration of user requirements and functional design, thereby necessitating higher requirement quality. Drawing on the extant research, this paper proposes a user requirement quality assessment process and constructs a user requirement quality assessment method, which is a pioneering and significant endeavor in agile requirement engineering.

3. User Requirement Quality Assessment Framework

The agile development process is illustrated in Figure 1, which is extended with the overall framework of requirement quality assessment in this paper. When the requirements are not clearly specified, the project team should start by identifying target users and forming a target user group. Subsequently, user story cards are distributed to the target user group, and the purpose and standards of filling in the cards are explained to them. After the user story cards are filled out, they are collected back by the project team. Then, the quality of user requirements is automatically evaluated by computers, and only qualified user requirements are fed into fast development. Finally, the product or function is delivered to the user, and the rapid iteration of product development is achieved through user feedback.
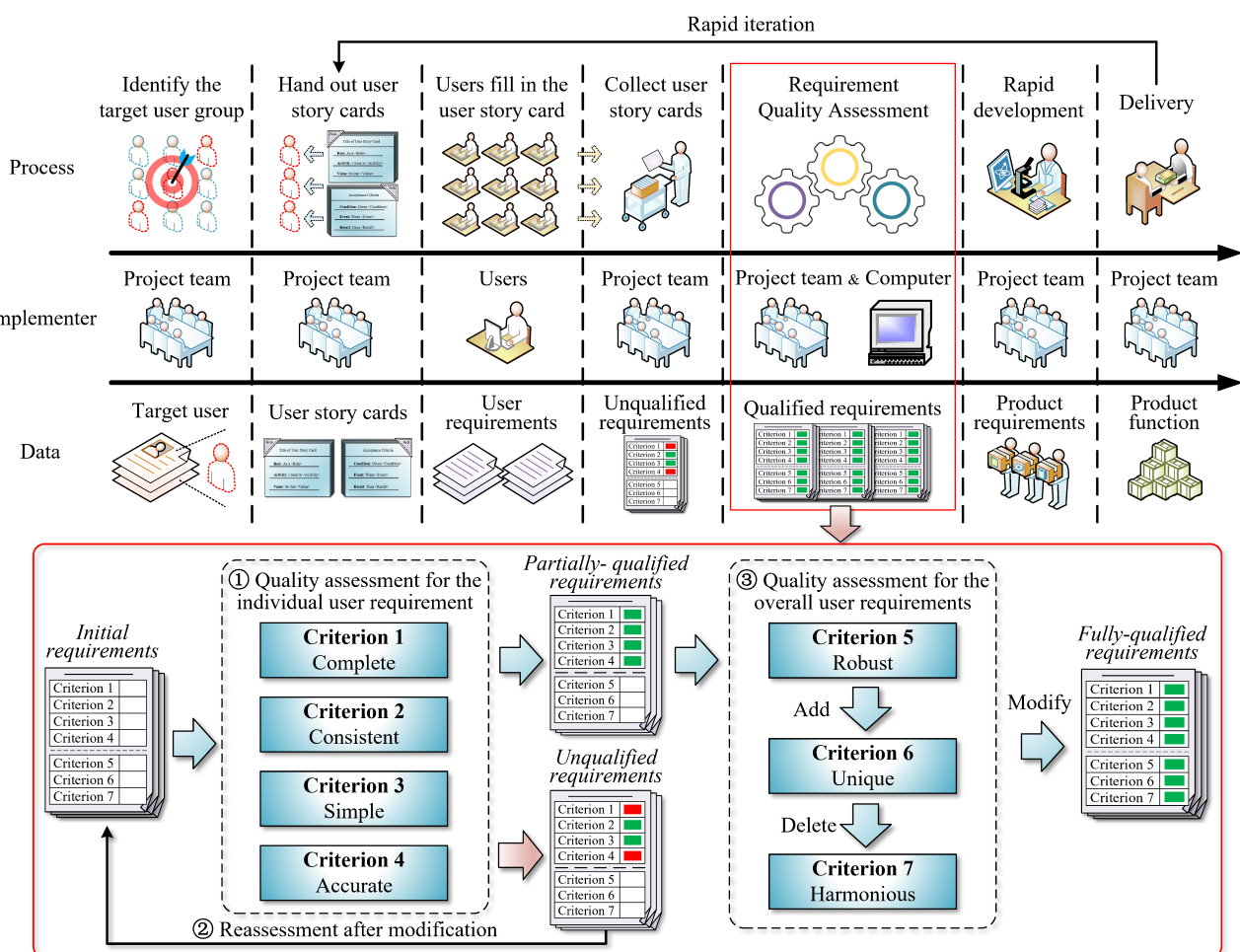

Figure 1. Framework for user requirement quality assessment in agile development.

As illustrated in Figure 1, the quality assessment of user requirements is conducted from two perspectives: individual user requirements and overall user requirements. Based on seven quality assessment criteria, the process is divided into three steps. First, an individual user requirement is assessed from four aspects: complete, consistent, simple, and accurate. Second, unqualified requirements are fed back to the users and modified. Finally, the overall quality assessment of all requirements with partially qualified quality is conducted from three aspects: robust, unique, and harmonious. Fully qualified user requirements are generated by adding, deleting, and modifying requirements. The seven assessment criteria are further explicated in the following section.

4. Model Design

In agile development, a user story is the primary vehicle for conveying user requirements. Agile development teams organize target users to fill in user story cards and obtain more detailed user requirements after analyzing a large number of user story cards. Therefore, this paper constructs a user story model and user requirement quality model to realize the quantitative analysis of requirement quality assessment.

4.1. User Story Model

The user story is an essential tool utilized in agile development to enable developers to rapidly acquire user requirements. An illustration of a user story card is presented in Figure 2.
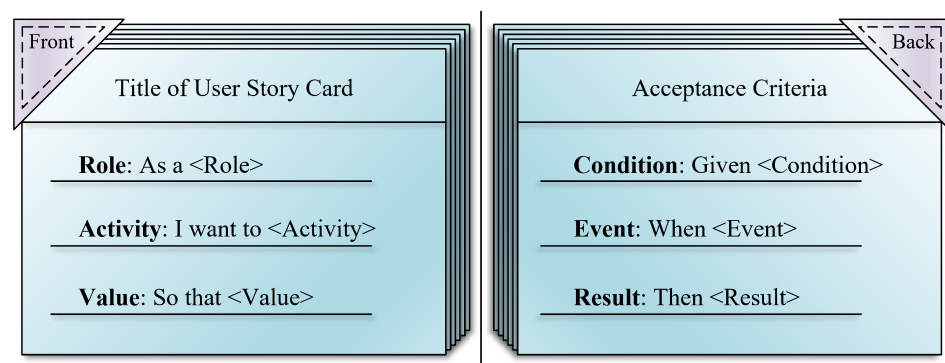

Figure 2. User story card.

As illustrated in Figure 2, the front of the card outlines the three essential elements of the user story: "role", "activity", and "value". "Role" explains the individual (or user) of the user story. "Activity" specifies the activity the user wishes that needs to be complete or the demands of the feature. "Value" outlines the aim or the benefit to the user after completing this activity. The back of the card provides the regulations and acceptance criteria for the user story. If the system's response adheres to the rules or acceptance criteria of the user story when the events given in the acceptance criteria occur, then the user story's requirements have been met.

Combined with the definition of the user story, the user story model is constructed, as illustrated in Equation (1):

$$US = (\langle R, A, V \rangle, \langle C, E, RE \rangle) \tag{1}$$

where R is the "role" of the story, A is the "activity" of the story, V is the "value" of the story, and V points to a requirement goal in the requirement goal set G, denoted as $V \to G_i$. $\langle R, A, V \rangle$ forms the description on the front of the user story card. C represents the "condition" of the user story, E represents the "event" of the user story, and RE represents the "result" of the user story. $\langle C, E, RE \rangle$ forms the validation information on the back of the user story card.

New energy vehicles (NEVs) are vehicles that utilize alternative energy sources as their primary power source. These alternative energy sources include electricity, hydrogen fuel cells, and hybrid technologies. NEVs can reduce—or even eliminate—emissions and dependence on fossil fuels, and play an important role in the transition toward sustainable transportation. Table 1 is an exemplary user story model used to elucidate the functional requirements of NEVs.

Table 1. Example of a user story model.

The description on the front of the user story card	
R	Camping enthusiast
A	Use the new energy vehicle to connect external electrical equipment and use batteries for the power supply
V	Can use electrical equipment more easily while camping
The verification information on the back of the user story card	
C	The new energy vehicle has the function of reverse power supply
E	Go camping and use appliances
RE	Can directly use my new energy car to power my appliances

4.2. User Requirement Quality Model

User requirements are expressed through user stories, and the quality of these requirements is determined by the accuracy and clarity of these stories. Therefore, INVEST guidelines for writing user stories can be used as a reference for assessing the quality of user requirements. Combined with several key attributes of quality requirements outlined by Lloyd [16], this paper proposes seven criteria for assessing the quality of user requirements from both individual user requirement and overall user requirement perspectives, as depicted in Table 2.

Table 2. Assessment criteria for the quality of user requirements.

Object	No.	Criteria	Statement
Individual user requirements	1	Complete	The six variables of the user story model are described completely.
	2	Consistent	The user story model is expressed by specified formats and keywords.
	3	Simple	The information in the variables of the user story model cannot be divided.
	4	Accurate	There are no fuzzy words in the user story model.
Overall user requirements	5	Robust	Each requirement goal is covered by at least one user story model's value.
	6	Unique	There are no duplicate user story models.
	7	Harmonious	There is no conflict between any two user story models.

According to the above criteria, the user requirement quality model is constructed, as shown in Equation (2):

$$URQ = (\langle S_1, S_2, S_3, S_4 \rangle, \langle S_5, S_6, S_7 \rangle) \tag{2}$$

where $S_1 \sim S_7$ represent the performance of user requirement quality under criterion 1 to criterion 7, respectively. $\langle S_1, S_2, S_3, S_4 \rangle$ constitutes the quality performance set of the individual user requirements, while $\langle S_5, S_6, S_7 \rangle$ constitutes the quality performance set of the overall user requirements.

Since the processing of user requirement expression involves natural language processing technology, this paper further defines the calculation methods of $S_1 \sim S_7$ in combination with the natural language toolkit in Python, as illustrated in Table 3.

Table 3. Calculation methods of $S_1 \sim S_7$.

Symbol	Calculation Method	Mathematical Expression		
		Qualified		Unqualified
S_1	When none of the six variables in a user story model is empty, it is regarded as qualified and unqualified in other cases.	$(R \neq \varnothing) \cap (A \neq \varnothing) \cap (V \neq \varnothing) \cap$ $(C \neq \varnothing) \cap (E \neq \varnothing) \cap (RE \neq \varnothing)$	(3)	Others
S_2	When all four conditions are met, it is regarded as qualified and unqualified in other cases.	$S_2C_1 \cap S_2C_2 \cap S_2C_3 \cap S_2C_4$ $S_2C_1 : R$ contains only nouns and quantifiers $S_2C_2 : A$ contains the subject $-$ verb $-$ object structure $S_2C_3 : V$ contains the verb $-$ object structure $S_2C_4 :$ nouns are derived from the specific corpus	(4)	Others
S_3	When all three conditions are met, it is regarded as qualified and unqualified in other cases.	$S_3C_1 \cap S_3C_2 \cap S_3C_3$ $S_3C_1 : R$ contains only one noun $S_3C_2 : V$ contains only one verb $-$ object structure $S_3C_3 :$ not contain "and", "or", "( )", "[ ]"	(5)	Others
S_4	When there are no fuzzy words FW in the six variables, it is regarded as qualified and unqualified in other cases.	$(FW \not\subset R) \cap (FW \not\subset A) \cap (FW \not\subset V) \cap$ $(FW \not\subset C) \cap (FW \not\subset E) \cap (FW \not\subset RE)$	(6)	Others
S_5	When each requirement goal is covered by at least one user story model's value, it is regarded as qualified and unqualified in other cases.	$\forall G, \exists V \to G$	(7)	Others
S_6	When the semantic similarity between each pair of US is less than the preset semantic similarity threshold sim_p, it is regarded as qualified and unqualified in other cases.	$\forall sim_{ij}\left(US_i, US_j\right) < sim_p$	(8)	Others
S_7	When there are no two user story models with the same E and RE but different C, it is regarded as qualified and unqualified in other cases.	$\neg \exists \left[US_i(E, RE) = US_j(E, RE), US_i(C) \neq US_j(C) \right]$	(9)	Others

It should be noted that the calculation method of S_6 utilizes a semantic similarity calculation method based on the vector space model [38]. This method is capable of rapidly calculating semantic similarity within a certain scope of text, and the principle will not be discussed in this paper. Through this model, the abstract nature of user requirements can be made concrete. Coupled with the customization algorithm, a large number of user stories can be processed by a computer automatically to quickly calculate the qualified quality of individual user requirements and the overall user requirements under the seven criteria. In conclusion, the automatic quality assessment of user requirements in agile development with user stories as the primary source of requirements can be achieved.

5. Experimental Results

To verify the effectiveness and feasibility of the proposed method in this paper, this study invited 20 smartphone users from different industries and age groups to fill out

user cards, and used all the user cards filled out by the users as input data to validate the effectiveness and feasibility of the model. The specific process is as follows:

Step 1—Generating the user requirement framework of the smartphone: By analyzing the functional features and specification parameters of the smartphones, this paper summarizes the user requirements framework of a smartphone, as shown in Table A1 in Appendix A. This framework contains a total of 34 requirement items divided into 4 categories (appearance, usability, economy, and entertainment). These requirement items can help users fill out user story cards, and serve as the requirement goal of agile development teams to continuously guide the research and development of products and the design of functions.

Step 2—Organizing users to fill out user cards based on the smartphone requirement framework: The research group organized a special seminar, and invited 20 smartphone users from various industries and age groups to participate. The user requirement framework for smartphones was provided as a reference, and user story cards filled out by the participants were collected and analyzed. A total of 136 user stories about smartphones were obtained, and Table 4 presents an example of one of these user stories.

Table 4. A user story created by a smartphone user.

No. 2.5.1	Category: Usability
The Description on the Front of the User Story Card	
R	As a travel enthusiast
A	I want to use the smartphone to wireless charge other devices at 100 watts of power
V	So that I can wireless charge other devices quickly without a power outlet outdoors using the smartphone.
The Verification Information on the Back of the User Story Card	
C	Given the smartphone is equipped with high-power wireless reverse charging
E	When I use the smartphone to charge other devices wirelessly
RE	Then other devices can be charged quickly

Step 3—Manually assessing and annotating the quality performance of user stories: The 136 user stories were manually assessed according to seven criteria, and their compliance with the quality assessment was marked. The results of the manual assessment were used as a benchmark dataset to support the validation of the effectiveness of the method, as shown in Figure 3, where the circled numbers indicate the criteria numbers for requirement quality assessment, the black squares represent "unqualified", and the white squares represent "qualified".

Step 4—Assessing the quality of user stories with the proposed model and determining the statistical experimental results: By evaluating the requirement quality of 136 user stories using the proposed method in this paper and comparing the results with the manually labeled data, experimental data were obtained to verify the effectiveness of the model, as shown in Table 5, where *TP* represents the number of cases that were manually annotated as "qualified" while the model assessment result was also "qualified"; *FP* represents the number of cases that were manually annotated as "unqualified" while the model assessment result was "qualified"; *FN* represents the number of cases that were manually annotated as "qualified" while the model assessment result was "unqualified"; and *TN* represents the number of cases that were manually annotated as "unqualified" while the model assessment result was also "unqualified".
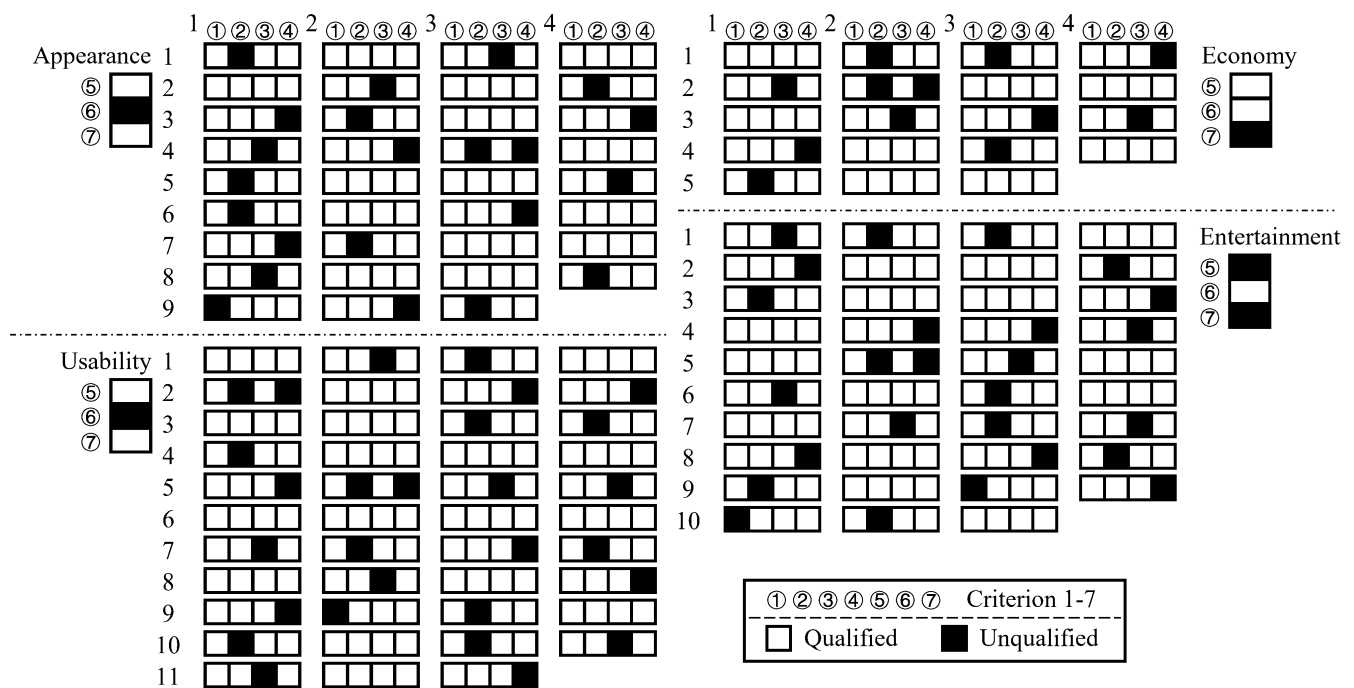
**Figure 3.** Manual assessment results of requirement quality for 136 user stories.

Table 5. The experimental results data after statistical analysis.

Criteria		Individual User Requirement				Overall User Requirements			Total
Category		①	②	③	④	⑤	⑥	⑦	
Appearance	TP	34	18	23	25	1	0	1	102
	FP	0	3	2	2	0	0	0	7
	FN	0	8	7	3	0	0	0	18
	TN	1	6	3	5	0	1	0	16
Usability	TP	42	25	28	29	1	0	1	126
	FP	0	4	2	2	0	0	0	8
	FN	0	7	8	5	0	0	0	20
	TN	1	7	5	7	0	1	0	21
Economy	TP	19	11	12	12	1	0	0	55
	FP	0	2	1	1	0	0	1	5
	FN	0	3	4	3	0	1	0	11
	TN	0	3	2	3	0	0	0	8
Entertainment	TP	37	20	26	25	0	1	0	109
	FP	0	4	2	2	0	0	0	8
	FN	0	9	7	6	0	0	0	22
	TN	2	6	4	6	1	0	1	20

Step 5—Calculating the metrics used to validate the effectiveness of the model: In this paper, the accuracy rate *A*, precision rate *P*, recall rate *R*, and F1 score are used to analyze the effectiveness of the model. The F1 score harmonizes the precision rate *P* and recall rate *R*, providing a more comprehensive assessment of the effectiveness of the model. The calculation results of the metrics for model effectiveness are presented in Table 6 and Figure 4.

Table 6. The calculation results of the metrics for model effectiveness.

Criteria	Individual User Requirements				Overall User Requirements			Total
Index	①	②	③	④	⑤	⑥	⑦	
TP	132	74	89	91	3	1	2	392
FP	0	13	7	7	0	0	1	28
FN	0	27	26	17	0	1	0	71
TN	4	22	14	21	1	2	1	65
A	100.00%	70.59%	75.74%	82.35%	100.00%	75.00%	75.00%	82.19%
P	100.00%	85.06%	92.71%	92.86%	100.00%	100.00%	66.67%	93.33%
R	100.00%	73.27%	77.39%	84.26%	100.00%	50.00%	100.00%	84.67%
F1	1.0000	0.7872	0.8436	0.8835	1.0000	0.6667	0.8000	0.8879
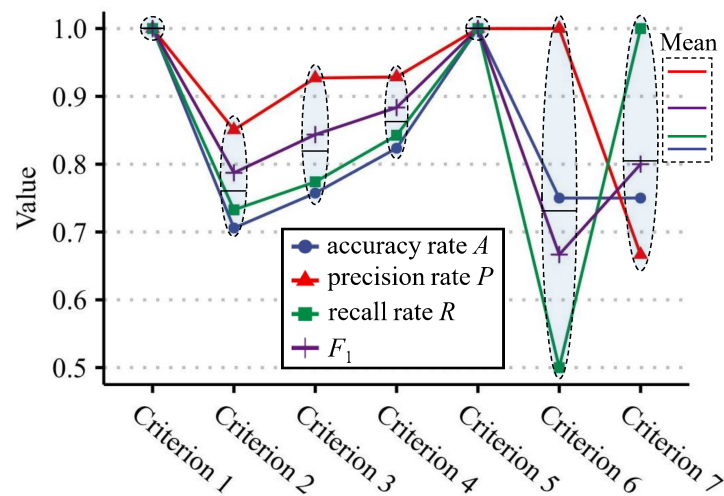
Figure 4. Line charts for *A*, *P*, *R*, and F1 scores.

Step 6—Calculating the effect of the model on improving requirement quality: To verify the feasibility of the proposed model in this paper, the improvement in requirement quality after applying the model was calculated. The effect of the quality improvement is shown in Figure 5, where subfigures (a)–(e) represent the quality improvement in the aspects of appearance, usability, economy, entertainment, and average values found in the user stories, with red bars indicating the original requirement quality and blue bars indicating the improved requirement quality.
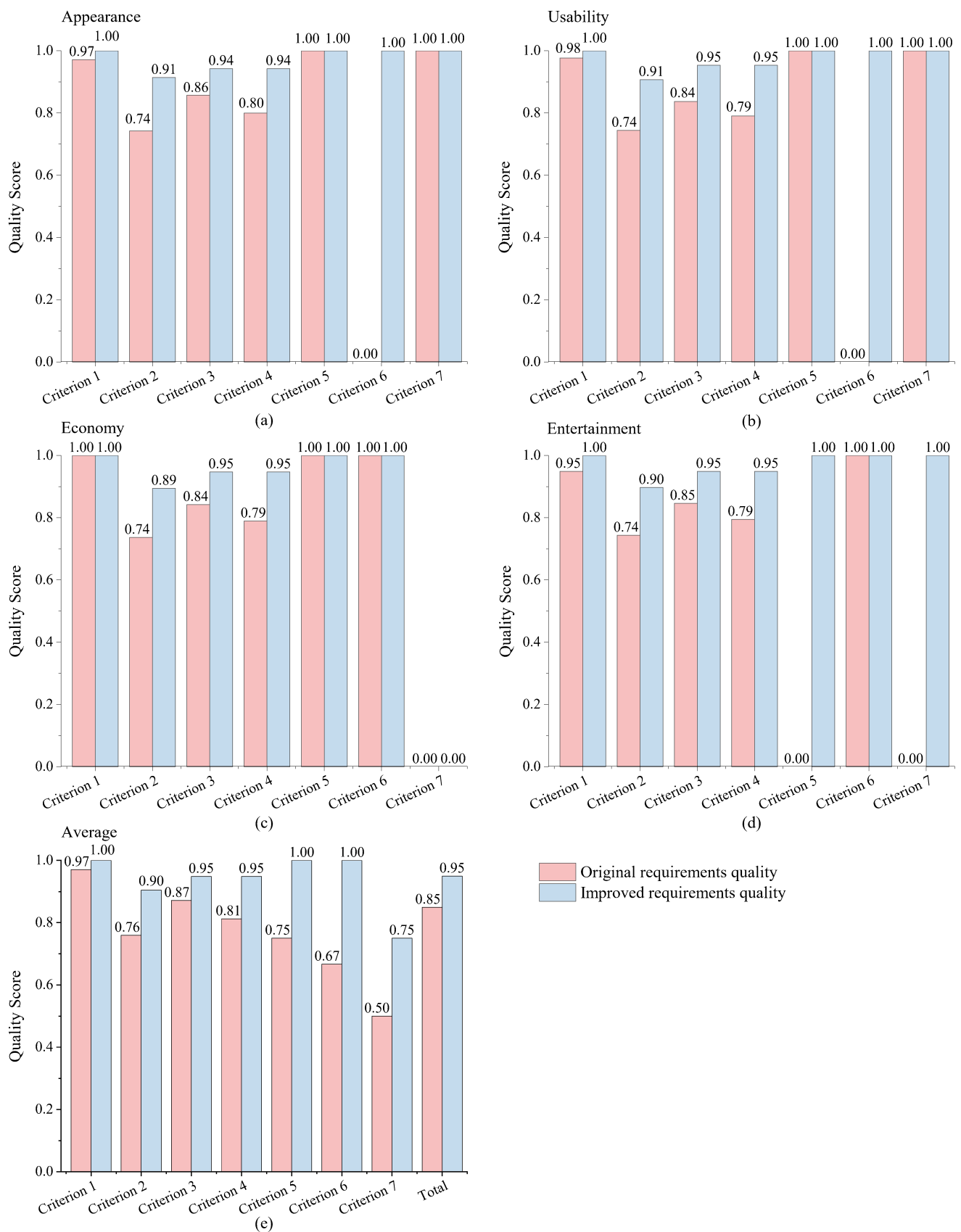
Figure 5. Bar chart of the requirement quality improvement effect.

Step 7—Analyzing the experimental data and drawing conclusions: By analyzing the above experimental results, the following three conclusions can be drawn:

(1) The model demonstrated good overall effectiveness. Based on Table 6 and Figure 4, the proposed user requirement quality assessment method in this paper achieved average *A*, *P*, and *R* of 82.19%, 93.33%, 84.67%, respectively, and an average F1 score of 0.8879, on 136 samples of user stories, demonstrating its effectiveness on this test dataset. The experimental data also showed the following characteristics: (a) The model achieved 100% accuracy and an F1 score of 1 in the testing of criteria 1 and 5. This is because the user story cards were used to collect user requirements, and the user story model can clearly divide the various elements in the user stories, which can then be processed by the computer. This method greatly reduces the burden of manual review and the possibility of errors. (b) The model performed relatively poorly in the testing of criterion 6, with an F1 score of only 0.6667. This is because the value of sim_p in Equation (8) was set too low, resulting in two user stories with slightly similar semantics being judged as unqualified. Therefore, by testing the F1 score under multiple sim_p values, the accuracy of the model on criterion 6 can be improved.

(2) The model has a significant improvement effect on the quality of user stories. Figure 5 illustrates that overall, user stories in the four aspects (appearance, usability, economy, and entertainment) showed a significant quality improvement in the seven assessment criteria, with improvements ranging from 2% to 100%, and the average requirement quality score was improved by 10.01%. This reflects the feasibility of the proposed model in improving the quality of user story requirements. However, in criterion 7 of economy, there was an outlier in the effect of quality improvement. After analysis, it was found that since criterion 7 evaluates the overall quality of the requirements, a single misjudgment by the model can lead to the entire requirement being deemed unqualified. This problem can be solved by improving the user requirement quality model, which will be the focus of future research.

(3) The model design that integrates multiple technologies demonstrates improvements. Compared to the existing research on requirement quality definitions [15–19], this study innovatively adopts a modeling approach to model the seven commonly recognized requirement quality attributes in the field of requirement engineering. Mathematical formulas are used to express the definition and calculate the requirement quality, prompting a shift from qualitative to quantitative approaches in requirement quality research. Compared to subjective methods for requirement quality assessment [20–33], this study integrates natural language processing to design an automated process for requirement quality assessment, which greatly improves the efficiency of the assessment process while demonstrating its effectiveness and feasibility. Compared to objective methods for requirement quality assessment [25,26,30,34–37], this paper collects user requirements as inputs for the requirement quality evaluation model through user stories used in agile development. This solves the problem of nonstandardized requirement descriptions due to broad sources, improves the consistency of data, and further enhances the model's feasibility. The above three aspects collectively demonstrate the innovation and advancement of this study.

6. Summary and Prospects

This paper proposes a method for assessing user requirement quality in agile development. By constructing the user requirement quality assessment framework, designing the user requirement quality assessment process, constructing the user story model based on user story cards, building the user requirement quality model, and designing the requirement quality assessment method, the problem of low-quality user requirements encountered in agile development is addressed to some extent. The method demonstrated good performance in a data experiment on smartphone user requirements, with *A*, *P*, and *R* of 82.19%, 93.33%, 84.67%, respectively, and an F1 score of 0.8879. The average requirement quality score improved by 10.01%, demonstrating the effectiveness and feasibility of the method. This provides an automated solution for agile development teams to effectively improve user requirement quality.

However, there are still some shortcomings in our work. For example, the assessment of the overall requirement quality needs to be further quantified, further optimization needs to be conducted for the parameter settings in the model, the scalability of user requirement quality assessment criteria needs to be improved, and the applicability of the user requirement quality assessment method in large-scale complex high-end equipment agile development needs to be further explored. In future work, we aim to advance the user requirement quality assessment model to achieve higher levels of accuracy and applicability. Specifically, we plan to apply the model in the acquisition of high-end equipment requirements, such as those associated with new energy vehicles. Additionally, we aim to develop a user requirement quality improvement tool by integrating natural language processing. The tool will provide agile development teams with improved work efficiency, allowing them to more effectively address the challenges involved in improving user requirement quality.

Appendix A

The user requirement framework of a smartphone is shown in Table A1.

Table A1. User requirement framework of a smartphone.

Requirement Category	Functions/Attributes	Unit
1. Appearance	1.1 Screen size	inch
	1.2 Screen curvature	degree
	1.3 Screen ratio	percentage
	1.4 Foldable screen	Boolean
	1.5 Secondary screen size	inch
	1.6 Body color	type
	1.7 Camera arrangement	type
	1.8 Body material	type
	1.9 Weight	gram

Table A1. *Cont.*

Requirement Category	Functions/Attributes	Unit
2. Usability	2.1 Screen quality (refresh rate/resolution/color accuracy)	hertz /ppi /ΔE
	2.2 Storage	GB
	2.3 Battery endurance	hour
	2.4 Charging time (0–100%)	minute
	2.5 Wireless charging/reverse wireless charging	watt
	2.6 Dual SIM/5G	type
	2.7 WIFI/infrared/Bluetooth/NFC	type
	2.8 Biometric identification	type
	2.9 Waterproof and dustproof grade	IPXX
	2.10 3.5 mm headphone jack	Boolean
	2.11 Communication encryption	Boolean
3. Economy	3.1 Selling price	EUR/USD
	3.2 Failure rate	percentage
	3.3 Warranty period	month
	3.4 Maintenance price	EUR/USD
	3.5 Recycling price	EUR/USD
	3.6 Free charger/protective case	boolean
4. Entertainment	4.1 Large game framerate/temperature	f/s/°C
	4.2 Game mode	type
	4.3 Loudspeaker power	watt
	4.4 Photographic pixel	pixel
	4.5 Zoom magnification	number
	4.6 Various photography modes	type
	4.7 Equipped with a stylus	Boolean
	4.8 Operating system ecology	type

Note: "type" indicates multiple optional types, and "Boolean" indicates equipped or not equipped.

References

1. Monte, I.; Lins, L.; Marinho, M. Communities of Practice in Large-Scale Agile Development: A Systematic Literature Mapping. In Proceedings of the 2022 48th Latin American Computing Conference, CLEI 2022, Armenia, Colombia, 17–21 October 2022.
2. Prenner, N.; Klunder, J.; Schneider, K. Defining Frames to Structure Agile Development in Hybrid Settings—A Multi-Case Interview Study. In Proceedings of the ACM International Conference Proceeding Series, Pittsburgh, PA, USA, 20–22 May 2022; pp. 34–44.
3. Uluda, O.; Philipp, P.; Putta, A.; Paasivaara, M.; Lassenius, C.; Matthes, F. Revealing the state of the art of large-scale agile development research: A systematic mapping study. *J. Syst. Softw.* **2022**, *194*, 111473. [CrossRef]
4. Kumar, D.; Dhir, S. *Requirement Barriers to Implement the Software Projects in Agile Development*; Springer Series in Reliability Engineering; Springer: Berlin/Heidelberg, Germany, 2022; pp. 341–353.
5. Shahjalal, M.; Hasan, M.K.; Chowdhury, M.Z.; Jang, Y.M. Smartphone Camera-Based Optical Wireless Communication System: Requirements and Implementation Challenges. *Electronics* **2019**, *8*, 913. [CrossRef]
6. Mengist, A.; Buffoni, L.; Pop, A. An Integrated Framework for Traceability and Impact Analysis in Requirements Verification of Cyber-Physical Systems. *Electronics* **2021**, *10*, 983. [CrossRef]
7. Almaiah, M.A.; Hajjej, F.; Lutfi, A.; Al-Khasawneh, A.; Alkhdour, T.; Almomani, O.; Shehab, R. A Conceptual Framework for Determining Quality Requirements for Mobile Learning Applications Using Delphi Method. *Electronics* **2022**, *11*, 788. [CrossRef]
8. Yahya, A.E.E.; Gharbi, A.; Yafooz, W.M.S.; Al-Dhaqm, A. A Novel Hybrid Deep Learning Model for Detecting and Classifying Non-Functional Requirements of Mobile Apps Issues. *Electronics* **2023**, *12*, 1258. [CrossRef]
9. Muellner, N.H. Karnaugh-Veitch Maps as Minimal Formal Contract between Textual Requirements and Tests: A Use-Case Based Technical Analysis. *Electronics* **2022**, *11*, 2430. [CrossRef]
10. Yang, B.; Ma, X.; Wang, C.; Guo, H.; Liu, H.; Jin, Z. User story clustering in agile development: A framework and an empirical study. *Front. Comput. Sci.* **2023**, *17*, 176213. [CrossRef]
11. Ferreira, A.M.; da Silva, A.R.; Paiva, A.C. Towards the Art of Writing Agile Requirements with User Stories, Acceptance Criteria, and Related Constructs. In Proceedings of the International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE–Proceedings, Online Streaming, 25–26 April 2022; pp. 477–484.
12. Amna, A.R.; Poels, G. Ambiguity in user stories: A systematic literature review. *Inf. Softw. Technol.* **2022**, *145*, 106824. [CrossRef]
13. Ananjeva, A.; Persson, J.S.; Bruun, A. Integrating UX work with agile development through user stories: An action research study in a small software company. *J. Syst. Softw.* **2020**, *170*, 110785. [CrossRef]

14. Ussery, S.J. A Construct for Assessing and Improving the Quality of Agile User Stories. Ph.D. Thesis, The George Washington University, Washington, DC, USA, 2021.

15. Femmer, H.; Vogelsang, A. Requirements quality is quality in use. *IEEE Softw.* **2019**, *36*, 83–91. [CrossRef]

16. Montgomery, L.; Fucci, D.; Bouraffa, A.; Scholz, L.; Maalej, W. Empirical research on requirements quality: A systematic mapping study. *Requir. Eng.* **2022**, *27*, 183–209. [CrossRef] [PubMed]

17. Unterkalmsteiner, M.; Gorschek, T. *Requirements Quality Assurance in Industry: Why, What and How?* Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer International Publishing: Berlin/Heidelberg, Germany, 2017; Volume 10153 LNCS, pp. 77–84.

18. Seidel Calazans, A.T.; Paldes, R.A.; Canedo, E.D.; Seidel Masson, E.T.; De Albuquerque Guimaraes, F.; Fialho Rezende, K.M.; De Souza Goncalves, F.; Mariano, A.M. Quality requirements and the requirements quality: The indications from Requirements Smells in a Financial Institution Systems. In Proceedings of the XXXIII Brazilian Symposium on Software Engineering, Salvador, Brazil, 23–27 September 2019; pp. 472–480.

19. Parra, E.; De La Vara, J.L.; Alonso, L. Poster: Analysis of requirements quality evolution. In Proceedings of the International Conference on Software Engineering, Gothenburg, Sweden, 27 May–3 June 2018; pp. 199–201.

20. Hauser, S.; Muller-Schmeer, S.; Franczyk, B.; Radomski, S. An Improved Assessing Requirements Quality with ML Methods. In Proceedings of the 2022 14th IEEE International Conference on Computational Intelligence and Communication Networks, CICN 2022, Al-Khobar, Saudi Arabia, 4–6 December 2022; pp. 424–429.

21. Kang, J.; Park, J. Data anonymization for requirements quality analysis: A reproducible automatic error detection task. In Proceedings of the LREC 2018—11th International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018; pp. 4432–4436.

22. Anu, V.; Walia, G.; Hu, W.; Carver, J.C.; Bradshaw, G. Using a Cognitive Psychology Perspective on Errors to Improve Requirements Quality: An Empirical Investigation. In Proceedings of the International Symposium on Software Reliability Engineering, ISSRE, Ottawa, ON, Canada, 23–27 October 2016; pp. 65–76.

23. Dehghani, R.; Wnuk, K.; Mendez, D.; Gorschek, T.; Ramsin, R. *On Understanding the Relation of Knowledge and Confidence to Requirements Quality*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer International Publishing: Cham, Switzerland, 2021; Volume 12685 LNCS, pp. 208–224.

24. Gallego, E.; Chale-Gongora, H.G.; Llorens, J.; Fuentes, J.; Álvarez, J.; Genova, G.; Fraga, A. Requirements quality analysis: A successful case study in the industry. In *Complex Systems Design & Management: Proceedings of the Seventh International Conference on Complex Systems Design & Management, CSD&M Paris 2016*; Springer: Cham, Switzerland, 2016; pp. 187–201.

25. Osama, E.; Abdelsalam, M.; Khedr, A. The effect of requirements quality and requirements volatility on the success of information systems projects. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 420–425. [CrossRef]

26. Dargan, J.L.; Wasek, J.S.; Campos-Nanez, E. Systems performance prediction using requirements quality attributes classification. *Requir. Eng.* **2016**, *21*, 553–572. [CrossRef]

27. Timoshchuk, E.; Kuznetsov, S.; Usmonov, Z.; Utih, A.; Farah, A.; Saliu, H. Application of Natural Language Processing with GQM and AHP Approaches for Requirements Quality Assessment; CEUR Workshop Proceedings. 2019; Volume 2525. Available online: https://ceur-ws.org/Vol-2525/ITTCS-19_paper_30.pdf (accessed on 8 January 2023).

28. Singh, M. Using Machine Learning and Graph Mining Approaches to Improve Software Requirements Quality: An Empirical Investigation. Ph.D. Thesis, North Dakota State University, Fargo, ND, USA, 2019.

29. Reyterou, C. Requirements Quality in the Incremental Design Processes: Problems and Perspectives; CEUR Workshop Proceedings. 2016; Volume 1564. Available online: https://ceur-ws.org/Vol-1564/paper4.pdf (accessed on 8 January 2023).

30. Post, A.; Fuhr, T. Case study: How Well Can IBM's "requirements Quality Assistant" Review Automotive Requirements; CEUR Workshop Proceedings. 2021; Volume 2857. Available online: https://ceur-ws.org/Vol-2857/nlp4re8.pdf (accessed on 8 January 2023).

31. Moreno, V.; Genova, G.; Parra, E.; Fraga, A. Application of machine learning techniques to the flexible assessment and improvement of requirements quality. *Softw. Qual. J.* **2020**, *28*, 1645–1674. [CrossRef]

32. Airlangga, G.; Liu, A. Investigating Software Domain Impact in Requirements Quality Attributes Prediction. *J. Inf. Sci. Eng.* **2022**, *38*, 295–316.

33. Wang, C.; Jin, Z.; Zhao, H.; Cui, M. An Approach for Improving the Requirements Quality of User Stories. *Jisuanji Yanjiu yu Fazhan/Comput. Res. Dev.* **2021**, *58*, 731–748.

34. Griva, A.; Byrne, S.; Dennehy, D.; Conboy, K. Software Requirements Quality: Using Analytics to Challenge Assumptions at Intel. *IEEE Softw.* **2022**, *39*, 80–88. [CrossRef]

35. Gopal, D.; Lyytinen, K. Effects of Social Structures in Requirements Quality of Open Source Software Project Development. In Proceedings of the ICIS 2017: Transforming Society with Digital Innovation, Seoul, Republic of Korea, 10–13 December 2017.

36. Lind, E.; Gonzalez-Huerta, J.; Alegroth, E. Requirements Quality vs Process and Stakeholders' Well-being: A Case of a Nordic Bank. *arXiv* **2022**, arXiv:2211.06122.

37.   Aziz, R.A.; Wong, B. The Impacts of Requirements Relationships Knowledge on Requirements Quality and Software Development Project Success; CEUR Workshop Proceedings. 2021; Volume 3062, pp. 41–50. Available online: https://opus.lib.uts.edu.au/bitstream/10453/158585/2/Paper07_QuASoQ.pdf (accessed on 8 January 2023).

38.   Singh, R.; Singh, S. Text Similarity Measures in News Articles by Vector Space Model Using NLP. *J. Inst. Eng. Ser. B* **2021**, *102*, 329–338. [CrossRef]