International Journal of Engineering and Technology Innovation, vol. 14, no. 3, 2024, pp. 309-320

Quality Model and Quality Characteristics Evaluation Suitable for Software 2.0

Yoshimichi Watanabe[1,*], Yunarso Anang[2], Masakazu Takahashi[1]

[1]Department of Computer Science and Engineering, University of Yamanashi, Yamanashi, Japan

[2]Department of Computational Statistics, Politeknik Statistika STIS, Jakarta, Indonesia

Received 23 January 2024; received in revised form 19 March 2024; accepted 20 March 2024

https://doi.org/10.46604/ijeti.2024.13303

Abstract

Software systems incorporating artificial intelligence (AI) technology are called software 2.0, and their development is spreading in various fields. The purpose of this study is to enable quality engineers to easily evaluate the quality of software 2.0. For this purpose, this paper proposes a quality model that can be used by quality engineers. In order to propose a new quality model for quality engineers, this paper selects and reorganizes quality elements from general-purpose quality models and guidelines, and reconstructs quality characteristics and their evaluation methods. The quality model is considered useful for quality engineers. The paper also identifies important quality characteristics in 24 different application areas and confirms the applicability of the proposed model. The result shows that the model is useful for quality engineers to evaluate the quality of software 2.0.

Keywords: AI products, software 2.0, software quality model, quality characteristics

1. Introduction

With the rapid development of information and communication technology in recent years, software systems have become the foundation for social activities. They have also become a necessary foundation for new value-creation opportunities. In particular, the field of embedded software systems and software systems containing artificial intelligence (AI) functions are often used to create new value and research on software system quality is actively discussed. As machine learning technology has been penetrating various application software in various application areas, the quality of machine learning software has become a major concern [1-4].

Under these circumstances, it is becoming increasingly important to incorporate quality assurance in an exploratory or hypothesis-testing manner, in which quality assurance items are identified as hypotheses for goals, monitored, and continuously improved in response to changes, rather than in planned quality assurance of software systems based on clear correct answers. In addition, quality characteristics such as security, privacy, and interoperability are becoming more important factors, depending on the size and diversity of the data and its sources, requiring a much different view of quality than the traditional one.

Software that includes AI functions is referred to as software 2.0 [5], and the development of software systems that include AI functions is expanding in a variety of fields. However, software 2.0 development and development methodologies are still in their infancy, and machine learning software engineering or machine learning systems engineering, which aim to solve

* Corresponding author. E-mail address: nabe@yamanashi.ac.jp

English language proofreader: Chih-Wen Teng

various problems by systematizing development methods from a software engineering perspective, are being actively discussed. Since software systems based on software 2.0 are constructed differently from conventional software, software 1.0, the quality characteristics of software systems also need to be different from conventional software's quality characteristics.

From this perspective, several types of research are conducted to focus on quality that captures the characteristics of software 2.0 and to clarify the quality elements and characteristics necessary for software 2.0 quality assurance. SQuaRE [6] is an appropriate framework to discuss the quality. Recently. the quality models extended for AI systems have been discussed [2, 7-9].

For example, Kuwajima and Ishikawa [7] propose to update SQuaRE's quality characteristics and sub-characteristics for AI systems. This research deals with the external quality of software systems and does not discuss the internal quality. Natale [8] investigated AI features and SQuaRE quality characteristics for the AI extension of SQuaRE. While the proposal is reasonable, it does not describe the features in detail. Nakajima and Nakatani [9] proposed the training data quality model as an extension to the SQuaRE data quality model of ISO/IEC 25012 [10]. It focuses on the aspect of training data quality and does not describe how to combine training data quality with the system quality model.

This research reviews the quality characteristics extracted in previous research and defines a quality model for software 2.0 from the viewpoint of software quality engineers, who are involved in quality with the expertise required for the job, contributing to the team and development, and doing professional work that delivers value to the customer.

In addition, there are many different application areas for applied software with AI functions. The important quality items generally differ depending on the characteristics of the application areas. From this viewpoint, the authors will identify the important quality characteristics required in the application areas. It is especially intended for software quality engineers working in small and medium-sized companies.

Most of the existing standards propose quality management systems from a comprehensive systems management perspective, with a view toward standardization. Many guidelines also are based on the knowledge of practitioners and are used by companies to independently choose whether or not to adopt them based on their impact on business, etc., and to put them into practice. For example, the guidelines for quality assurance of AI-based products and services [3] and the machine learning quality management guideline [11] are issued. These guidelines focus on quality assurance as an organization and are not familiar with the software quality engineers. Also, the standards such as ISO/IEC 25059 [12] deal with the quality model. However, this standard does not consider data quality characteristics.

In addition, these guidelines and standards are oriented toward generality, which makes them difficult to apply in small organizations. There are some AI-extended approaches, but there are no universal standards or guidelines currently. Each software system must be developed while considering quality characteristics, quality levels, risks, processes, evaluation indices, technologies and systems to be used, and so on.

Systematization of technologies to master machine learning is an urgent task, requiring the establishment of data preparation processes, model technologies, and system technologies that complement machine learning. This research proposes a quality model from the viewpoint of software quality engineers and identifies important quality characteristics in specific application fields to confirm the validity of the proposed quality model.

The remaining sections of this paper are organized as follows. Section 2 presents the literature review. Section 3 proposes a quality model for software 2.0 for software quality engineers and typical methods to measure quality characteristics. Section 4 identifies the quality characteristics that are important in the application areas to confirm the usefulness of the proposed quality model. Finally, Section 5 concludes this paper.

2. Literature Review

It has been pointed out that software 2.0 has characteristics that are essentially different from those of traditional software, which is created deductively [4]. In the quality of software 2.0, the issue is not only the quality assurance of the machine learning calculation part but also the quality of the software system as a whole, including the machine learning calculation part. Various organizations are currently defining quality models and quality characteristics for software 2.0. Some of them are explained as follows.

The guidelines for quality assurance of AI-based products and services [3] define the following five classification axes for the concept of quality assurance and describe quality assurance based on them.

(1) Data integrity

(2) Model robustness

(3) System quality

(4) Process agility

(5) Customer expectation

This guideline states that it is not simply that a high evaluation value for each axis is preferable, but that it is important to strike an appropriate balance, and if the evaluation values based on the five axes are well-balanced, the situation is such that sufficient quality assurance can be expected to meet customer expectation. It provides a checklist of evaluation perspectives in five classification axes. Customer expectation and process agility are not treated as elements of the quality model. No quantitative measure of the magnitude of the ratings for each axis is provided. In addition, it is necessary to consider the customer expectation axis and other axes separately depending on the status of the customer's level of understanding. Therefore, the use of this model is not always easy.

Machine learning quality management guideline [11] defines the quality of software 2.0 using machine learning function by classifying it into the following three categories.

(1) Quality in use: This is expected quality to be satisfied overall when software 2.0 is used. It is the quality that the software system as a whole should provide to the user.

(2) External quality: This is expected quality to be satisfied by the components of the software 2.0 built by the machine learning functions. This is a quality that elements incorporating machine learning functions should possess to achieve quality when using the software system.

(3) Internal quality: This is an inherent quality of components built with machine learning functions. This is a quality that should be satisfied during the design and operation of elements incorporating machine learning functions to achieve external quality.

This guideline considers that through the improvement of the internal quality of the machine learning functions, external quality will be improved to the required level to achieve the final quality in the use of the system. The quality of use of the overall system that is ultimately realized is different for each application of the system that needs to be focused on. Each quality characteristic that constitutes the external quality and the internal quality has the following sub-characteristics, each of which has quality levels.

(1) External quality: risk avoidance, AI performance, fairness, privacy, and AI security.

(2) Internal quality: designing quality structures and datasets (sufficiency of problem domain analysis and sufficiency of data design), dataset quality (coverage of datasets, uniformity of datasets, and adequacy of data), quality of machine learning

models (correctness of trained models and stability of trained models), quality of software implementation (reliability of underlying software systems), and operational quality (maintainability of quality in operation).

This guideline presents the concept of software quality incorporating machine learning functions and the matters necessary for its realization in a cross-disciplinary, industry-independent manner. The goal is to provide a common understanding between departments that determine quality goals and quality assurance departments, and between system developers and the customers. For this reason, it does not describe in detail how to achieve the internal quality. Therefore, the model is not necessarily suitable for software quality engineers.

The AI extension of SQuaRE is published as an international standard ISO/IEC 25059 [12], which focuses on the quality characteristics of trustworthy AI derived from ethical AI [13]. This standard is an application-specific AI system extension to the SQuaRE quality model specified in ISO/IEC 25010 [6]. The quality model of this standard is considered from two perspectives, product quality and quality in use.

ISO/IEC 25059 provides a quality model for AI systems and is an application-specific extension to the SQuaRE standard. Therefore, while it is useful as a quality model specific to AI systems, it may not be directly applicable to software quality in general. Understanding and applying ISO/IEC 25059 requires a certain amount of knowledge and education. The quality assessment process requires resources and time. Also, ISO/IEC 25059 itself does not directly discuss the characteristics of training data.

Kuwajima and Ishikawa [7] propose to update SQuaRE's quality characteristics and sub-characteristics for AI systems. They identified traditional quality characteristics that are invalidated by the specifics of machine learning and new quality characteristics that should be added by trustworthy AI [13]. This paper provided an overall insight into incorporating the essence of machine learning and AI ethics from traditional software engineering into SQuaRE. This research deals with the external quality of software systems and does not discuss the internal quality.

Natale [8] investigated AI features and SQuaRE quality characteristics for the AI extension of SQuaRE. They proposed to introduce fairness, explainability, and freedom from risk as new quality characteristics. It is only a mostly superficial feature. This paper does not contain a detailed description. Therefore, it is not possible to make concrete use of it.

Nakajima and Nakatani [9] proposed the training data quality model as an extension to the SQuaRE data quality model of ISO/IEC 25012 [10]. This paper introduces two new data quality characteristics, adequacy and provenance. This research focuses on the aspect of training data quality. Therefore, it is insufficient as a quality model for the entire AI system.

The following observations were made after reviewing the literature for quality models of software 2.0.

(1) Most guidelines focus on quality assurance as an organization rather than from a quality engineer's perspective. They cover a wide range of quality characteristics and are too general for quality engineers. While many of the guidelines are useful for large companies, they are difficult to apply to smaller companies.

(2) The quality model of many standards such as ISO/IEC 25059 is considered from two perspectives, product quality and quality in use, but there is no discussion of what should be considered based on the characteristics of the training data. The distribution of training data and other factors are not considered.

(3) Many AI extensions of SQuaRE do not propose to include evaluation methods for quality characteristics concerning external quality, internal quality, and quality-in-use.

This paper proposes the quality model and quality evaluation methods suitable for software 2.0 from the viewpoint of software quality engineers. The model also takes into account the characteristics of training data and is based on three standard models: an external quality model, an internal quality model, and a quality-in-use model.

3. A Quality Model for Software 2.0 for Software Quality Engineers

This section proposes a quality model for software 2.0 from the viewpoint of software quality engineers. In order to propose a new quality model for quality engineers, this paper selects and reorganizes quality elements from general-purpose quality models and guidelines, and reconstructs quality characteristics and their evaluation methods. The proposed model consists of two models: the system quality model for software 2.0 and the data quality model for software 2.0. The system quality model for software 2.0 has internal and external quality characteristics and the data quality model for software 2.0 has internal and data quality characteristics. The evaluation methods for each quality characteristic are also described.

3.1. The system quality model for software 2.0

Among the software 2.0 quality characteristics, the quality characteristics of the system quality model for software 2.0 are classified into internal quality characteristics and external quality characteristics. Fig. 1 shows the system quality model for software 2.0 proposed in this paper. The quality evaluation methods of this model are defined as follows.

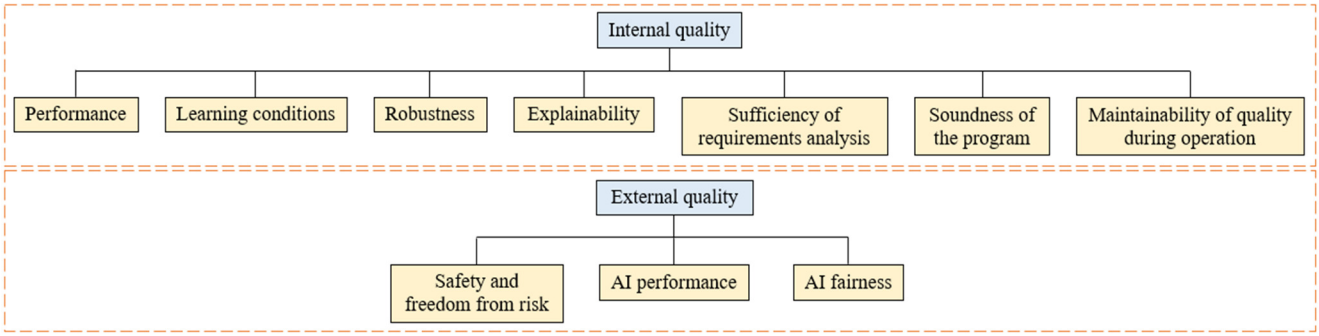

Fig. 1 The system quality model for software 2.0

3.1.1. Internal quality characteristics

This paragraph describes the internal quality characteristics of the system quality model for software 2.0. They consist of seven quality characteristics: performance, learning conditions, robustness, explainability, sufficiency of requirements analysis, soundness of the program, and maintainability of quality during operation. Each quality characteristic and its typical evaluation methods are described.

(1) *Performance*: The performance is the indicator used to evaluate the performance of software 2.0. Specific measures include accuracy, precision, recall, F-measure, and specificity [14]. To evaluate the performance based on these measures, a confusion matrix [14] is used, in which multiple patterns are shown in a cross-tabulation table based on predictions and correct answers in software 2.0.

(2) *Learning conditions*: The learning conditions are the indicators of whether the developed model is in an overfitting or underfitting state [15]. To measure the learning conditions, it is necessary to evaluate the learning state of the model, and cross-validation [16] can be used.

(3) *Robustness*: The robustness is the ability to predict properties adequately for data with properties that are not often present in the training data. One method to measure robustness is to calculate the magnitude of the noise that does not change the inference result and evaluate the result using the peak signal-to-noise ratio (PSNR) [17]. This method can be used with randomized smoothing [18]. Using a detector with an inspector trained to detect adversarial perturbations [19], it is possible to measure and evaluate the percentage of adversarial perturbations detected by the detector (detection rate) against a data set containing adversarial data.

(4) *Explainability*: Explainability is the degree to which a user using the output from a system can grasp the criteria used in obtaining the output [20]. Explainability can be evaluated in three ways: deep explanation, interpretable models, and model induction. However, the evaluation method is currently in the research phase and there is no established evaluation method yet.

(5) *Sufficiency of requirements analysis*: The sufficiency of requirements analysis is the indicator that the results of the analysis cover all assumed usage conditions when the analysis has been conducted on the properties of the actual data during operation that is assumed to be input to the machine learning elements corresponding to the real-world usage of software 2.0 [11]. The sufficiency of requirements analysis can be evaluated using a feature model, or a simplified feature model, or a method that captures specific usage conditions as a combination of these models.

(6) *Soundness of the program*: The soundness of the program is the indicator that the training program used in the training phase of machine learning and the prediction and inference program used at runtime perform correctly as a software program for given data and trained machine learning models [11]. In software 2.0, if the runtime environment is different from the environment of the training phase, it can be evaluated by testing the system using software that reproduces the same calculations as the runtime environment during the testing phase.

(7) *Maintainability of quality during operation*: The maintainability of quality during operation is the indicator of whether the internal quality satisfied at the start of operation will be maintained throughout the operation period [11]. There are two types of evaluation methods. Whenever additional training or model updates are performed in the development environment, the system can be evaluated by always performing quality inspections before the update, just as in the testing phase during initial development. In addition, when additional learning is performed and models are updated in the operational environment, the system is evaluated on whether a quality monitoring mechanism is in place and whether the operation incorporates measures to deal with deterioration.

3.1.2. External quality characteristics

This paragraph describes the external quality characteristics of the system quality model for Software 2.0. They consist of three quality characteristics: safety and freedom from risk, AI performance, and AI fairness. Each quality characteristic and its typical evaluation methods are described.

(1) *Safety and freedom from risk*: Safety and freedom from risk is the indicator of the property of avoiding adverse effects, such as human injury, economic loss, or opportunity loss, to operators, users, or third parties of the product due to undesirable decision-making behavior caused by machine learning [21]. The level of risk avoidance for software 2.0 is evaluated based on the classification of AI safety levels, which are categorized based on human risks such as injury to humans, and economic risks.

(2) *AI performance*: AI performance is the indicator of whether machine learning functions can output the expected output by the user with higher accuracy and probability on average over a longer period [21]. Specific target values to be achieved are defined as key performance indicators (KPIs) and are evaluated. The KPIs to be used will depend on the objectives of the individual software 2.0.

(3) *AI fairness*: AI fairness is the degree to which the output of machine learning functions or its distribution is unaffected by differences in some of the attributes possessed by the source of the input such as humans [21]. AI fairness criteria include equivalent odds, equal opportunity, demographic parity, fairness through recognition, fairness through unconsciousness, and conditional statistical fairness. AI fairness is then evaluated based on each of these criteria.

Fig. 2 shows the quality characteristics of the system quality model for software 2.0 and its typical evaluation methods. The system quality model is classified into two characteristic categories: internal quality and external quality. Each category includes a set of quality characteristics, such as performance, learning conditions, robustness, explainability, sufficiency of requirements analysis, soundness of the program, maintainability of quality during operation, safety and freedom from risk, AI performance, and AI fairness, along with their respective typical evaluation methods.

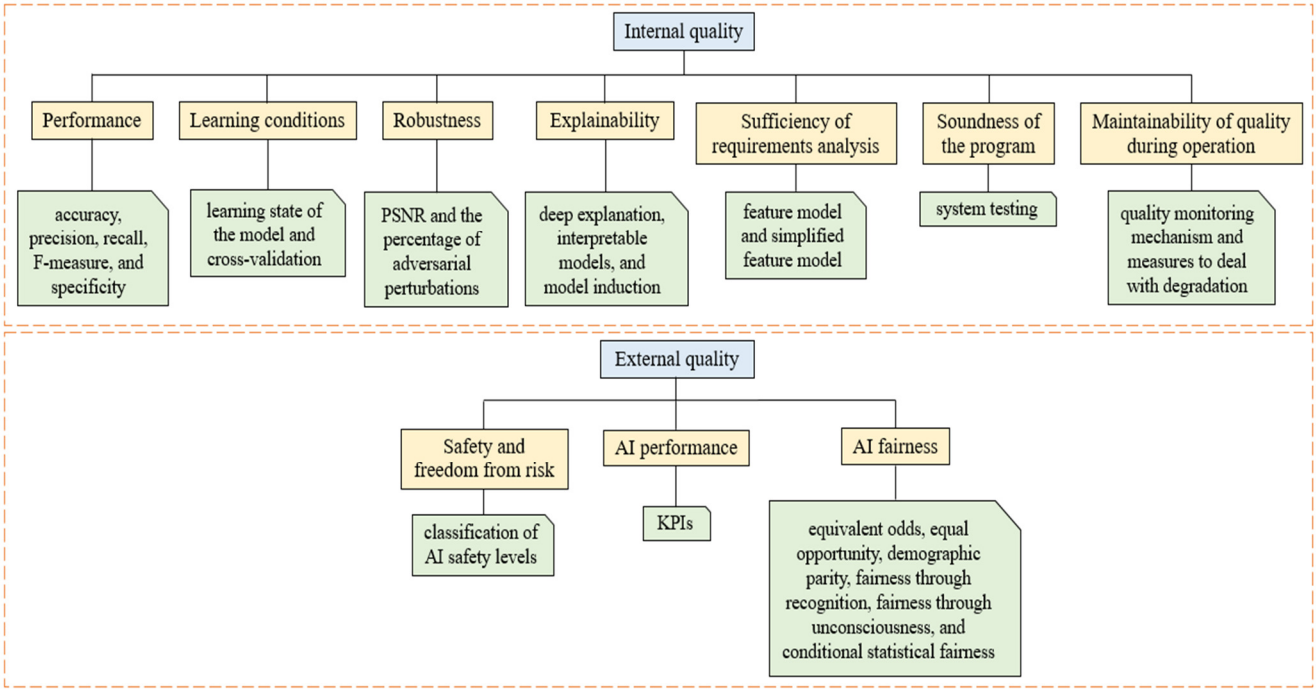

Fig. 2 The quality characteristics of the system quality model for software 2.0 and its typical evaluation methods

3.2. The data quality model for software 2.0

Next, the data quality model for software 2.0 is explained. The data quality model for software 2.0 can be classified into the following two categories: the internal quality model and the data quality model. Fig. 3 shows the data quality model. This model is applied to both training data and test data.

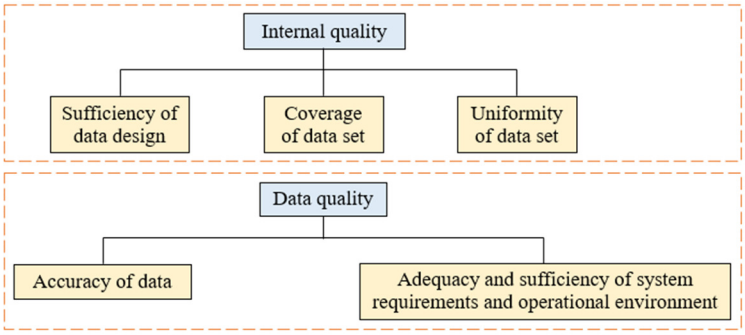

Fig. 3 The data quality model for software 2.0

3.2.1. Internal quality characteristics

This paragraph describes the internal quality characteristics of the data quality model for software 2.0. They consist of three quality characteristics: sufficiency of data design, coverage of data set, and uniformity of data set. Each quality characteristic and its typical evaluation methods are described.

(1) *Sufficiency of data design*: The sufficiency of data design is the degree of whether the training data is designed to adequately reflect high-risk situations and other conditions within the operation [11]. This can be evaluated by using the

coverage criterion to set criteria for the level of detail in the combination of attribute classifications and the degree of inspection items in the testing process.

(2) *Coverage of data set*: The coverage of the data set is the degree of whether the software 2.0 can prepare training data without omissions based on its design for the various situations in its required operation [11]. Software 2.0 can be evaluated by making sure that there is sufficient training and test data for the case it is designed for.

(3) *Uniformity of data set*: The uniformity of the data set is the degree of whether the training data is uniform to improve the performance of software 2.0 [11]. It can be evaluated whether the original data is unbiased by extracting and evaluating an unbiased sample from the original data.

3.2.2. Data quality characteristics

This paragraph describes the data quality characteristics of the data quality model for software 2.0. They consist of two quality characteristics: accuracy of data and adequacy and sufficiency of system requirements and operational environment. Each quality characteristic and its typical evaluation methods are described.

(1) *Accuracy of data*: The accuracy of data is the indicator of whether the data used in training contains errors or improprieties [11]. It is evaluated in terms of accuracy, completeness, consistency, credibility, correctness, precision, and traceability. These methods are used to evaluate the accuracy of the data and it is also checked for any imbalance between the results of each evaluation. The ideal data should be unbiased, exhaustive, and complete.

(2) *Adequacy and sufficiency of system requirements and operational environment*: The adequacy and sufficiency of system requirements and operational environment is the degree of whether the data used for training are described and whether the situation or subject that should not use software 2.0 is understood, according to the used model and the performance [11]. The evaluation is based on whether the system maintains information about the data set, such as the source of data collection, collection policy, collection criteria, annotation assignment criteria, and usage constraints, from the preparation of the data set for learning to the development and operation of software 2.0.

Fig. 4 shows the quality characteristics of the data quality model for software 2.0 and its typical evaluation methods. The data quality model is classified into two characteristic categories: internal quality and data quality. Each category includes a set of quality characteristics, such as sufficiency of data design, coverage of data set, and uniformity of data set, accuracy of data, and adequacy and sufficiency of system requirements and operational environment, along with their respective typical evaluation methods.

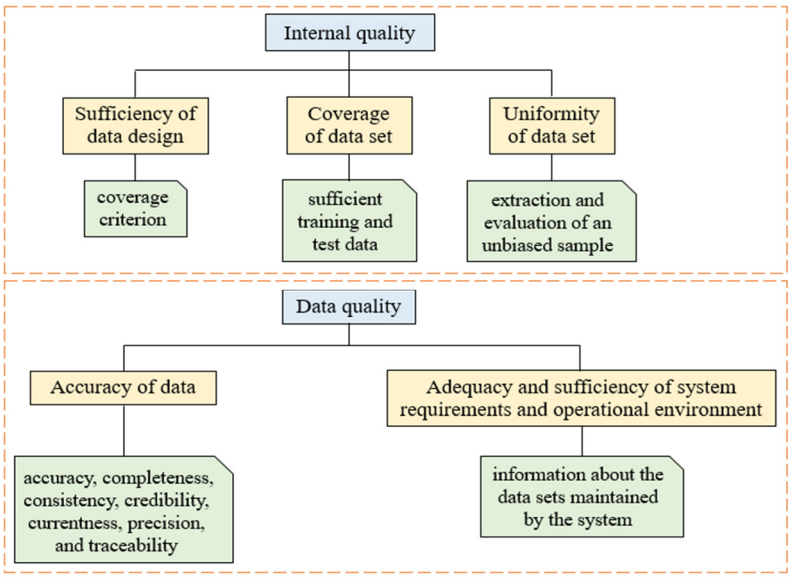

Fig. 4 The quality characteristics of the data quality model for software 2.0 and its typical evaluation methods

4. Important Quality Characteristics of the Application Areas

Software 2.0 is beginning to be used in many different fields of application. In general, however, the important quality characteristics depend on the application area. This section summarizes the quality characteristics that are considered important for the application of software 2.0 to each application area. By surveying some papers, literature, and guidelines [2-4, 21-22], Table 1 shows examples of typical application areas where software 2.0 is or could be applied.

Table 1 Examples of the fields of application

1. Manufacturing domain
(1) Manufacturing (2) Food inspection (3) Waste disposal
2. Mobility domain
(1) Automatic driving monitoring
3. Medical domain
(1) Diagnostic support (2) Drug discovery (3) Genome analysis
4. Nursing care domain
(1) Conviviality support (2) Preventive healthcare
5. Infrastructure domain
(1) Maintenance and inspection (2) Operational efficiency (3) Disaster prevention
6. Service, retail, and logistics domains
(1) Application to automobile-related service domain (2) Similar product image search (3) Application to the real estate field (4) Approach to life support robots (5) Inquiry response (6) Finance and insurance (7) Internet-related service (8) Application to the retail industry (9) Application to the logistics field
7. Agriculture, forestry, and fisheries domains
(1) Agriculture (2) Livestock industry (3) Fishing industry

In the manufacturing area, software 2.0 can be applied to product inspection such as the detection of defective automobile parts and defects in LCD panels, food inspection such as the inspection of food packaging, and waste disposal such as the automatic sorting of industrial waste, and so on. In this area, performance and robustness in the internal quality of the system quality model, AI performance in the external quality of the system quality model, and uniform of data set in the internal quality of the data quality model, and accuracy of data in the data quality of the data quality model are considered important.

In the mobility area, software 2.0 can be applied to automated driving monitoring such as in-car monitoring of automated cars, and so on. In this area, explainability in the internal quality of the system quality model, safety and freedom from risk in the external quality characteristics of the system quality model, sufficiency of data design in the internal quality of the data quality model, and accuracy of data in the data quality of the data quality model are considered important.

In the medical area, software 2.0 can be applied to diagnostic support for endoscopic imaging to detect cancer, automatic shadow detection in ultrasonography, programmed medical devices in the field of brain MRI, productivity improvement in drug discovery, genome analysis, and so on. In this area, explainability, sufficiency of requirements analysis and soundness of

the program, and maintainability of quality during operation in the internal quality of the system quality model, AI performance in the external quality of the system quality model, and coverage of data set, and uniformity of data set, and accuracy of data in the internal quality of the data quality model are considered important.

In the nursing care area, software 2.0 may be applied to conviviality support such as nursing care coaching preventive health care, and so on. For example, through software 2.0, the daily lives and health conditions of the elderly can be monitored, problems can be detected early, and caregivers can provide appropriate support. In addition, software 2.0 can be used to assist the elderly in finding areas of interest and communities. In this area, learning conditions, explainability, and sufficiency of requirements analysis in the internal quality of the system quality model, AI fairness in external quality of the system quality model, coverage of data set, and uniform of data set in the internal quality of the data quality model, and adequacy and sufficiency of system requirements and operational environment in the data quality of the data quality model are considered important.

In the infrastructure area, software 2.0 can be applied to maintenance and inspection such as power transmission tower inspection, operational efficiency of dams and other facilities by predicting rainfall, and disaster prevention such as land-slide risk prediction. In this area, performance, learning conditions, robustness in the internal quality of the system quality model, safety and freedom from risk in the external quality of the system quality model, sufficiency of data design in the internal quality of the data quality model, and adequacy and sufficiency of system requirements and operating environment in the data quality of the data quality model are considered important.

In the service, retail, and logistics areas, software 2.0 can be applied to automotive services such as used car valuation, real estate such as similar product image search, rental property image classification, lifestyle support robots such as fully automatic cleaning robots, financial and insurance support such as stock price prediction and automatic analysis of insurance policies, Internet-related services such as streaming video automatic translation systems, inquiry response and retail such as customer tracking and marketing, and logistics such as picking systems. In this area, performance, learning conditions, robustness, and soundness of the program in the internal quality of the system quality model, AI performance and AI fairness in the external quality of the system quality model, coverage of data set in the internal quality of the data quality model, and adequacy and sufficiency of system requirements and operating environment in the data quality of the data quality model are considered important.

In the areas of agriculture, forestry, and fisheries, software 2.0 can be applied to low-pesticide agriculture with pinpoint pesticide application, livestock farming with anomaly detection, and efficient fishing based on weather data. In this area, learning conditions, and maintainability of quality during the operation in the internal quality of the system quality model, safety and freedom from risk in the external quality of the system quality model, uniformity of data sets in the internal quality of the data quality model, and accuracy of data in the data quality of the data quality model are considered important.

Table 2 summarizes the correspondence between the application areas and the quality characteristics considered important in those areas. While all quality characteristics are important in each area, this table selects those that are considered particularly important. This table confirms the usefulness of the proposed quality model.

Table 2 The correspondence between the application areas and the quality characteristics

	Manufacturing	Mobility	Medical	Nursing care	Infrastructure	Service, retail, and logistics	Agriculture, forestry, and fisheries
Performance	×	-	-	-	×	×	-
Learning conditions	-	-	-	×	×	×	×
Robustness	×	-	-	-	×	×	
Explainability	-	×	×	×	-	-	-

Table 2 The correspondence between the application areas and the quality characteristics (continued)

	Manufacturing	Mobility	Medical	Nursing care	Infrastructure	Service, retail, and logistics	Agriculture, forestry, and fisheries
Sufficiency of requirements analysis	-	-	×	×	-	-	-
Soundness of the program	-	-	×	-	-	×	-
Maintainability of quality during operation	-	-	×	-	-	-	×
Safety and freedom from risk	-	×	-	-	×	-	×
AI performance	×	-	×	-	-	×	-
AI fairness	-	-	-	×	-	×	-
Sufficiency of data design	-	×	-	-	×	-	-
Coverage of data set	-	-	×	×	-	×	-
Uniform of data set	×	-	×	×	-	-	×
Accuracy of data	×	×	×	-	-	-	×
Adequacy and sufficiency of system requirements and operational environment				×	×	×	

The symbol "×" indicates that the quality characteristic is useful in the application area.

5. Conclusion

This paper proposed the quality model and quality evaluation methods suitable for software 2.0 from the viewpoint of software quality engineers. Also, this paper categorized the quality characteristics required for AI products by the area in which the products are used. The validity and usefulness of the proposed model were confirmed by illustrating the quality characteristics that are important in the application areas.

One of the future tasks is to investigate quality characteristics related to AI required in actual workplaces and to reconfirm the validity of the proposed model. This paper only considered important quality characteristics with a theoretical consideration of their use in applied fields based on a literature survey. One of the other future works is to propose a more realistic model. Another future work is to propose a more detailed evaluation method of quality characteristics. The current model does not directly take into account generative AI. Generative AI can be applied in all fields. However, generative AI is outside the scope of this model because it has not been fully tested. It is also necessary to modify the quality model accordingly to keep up with the evolution of AI.

Acknowledgment

A part of this research was supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science, grant numbers 21K04560 and 22K04616.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] OECD, Artificial Intelligence in Society, Paris: OECD Publishing, 2019.
[2] S. Nakajima, T. Nakatani, and S. Takizawa, "AI Quality Assurance," Journal of Information Processing, vol. 63, no. 11, pp. 602-605, pp. e1-e33, 2022. (In Japanese)
[3] QA4AI Consortium, Guidelines for Quality Assurance of AI-based Products and Services, 2011.
[4] S. Nakajima, Machine Learning Quality Issues Learned from Software Engineering, Tokyo: Maruzen, 2020. (In Japanese)

[5] A. Karpathy, "Software 2.0," https://karpathy.medium.com/software-2-0-a64152b37c35, November 12, 2017.
[6] Systems and Software Engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE) — Systems and Software Quality Model, ISO/IEC 25010, 2011.
[7] H. Kuwajima and F. Ishikawa, "Adapting SQuaRE for Quality Assessment of Artificial Intelligence Systems," IEEE International Symposium on Software Reliability Engineering Workshops, pp. 13-18, October 2019.
[8] D. Natale, "Possible Extension of ISO/IEC 25000 Quality Models to Artificial Intelligence in the Context of an International Governance," 27th Asia-Pacific Software Engineering Conference, pp. 22-24, December 2020.
[9] S. Nakajima and T. Nakatani, "AI Extension of SQuaRE Data Quality Model," IEEE 21st International Conference on Software Quality, Reliability and Security Companion, pp. 306-313, December 2021.
[10] Software Engineering — Software Product Quality Requirements and Evaluation (SQuaRE) — Data Quality Model, ISO/IEC 25012, 2008.
[11] Digital Architecture Research Center, "Machine Learning Quality Management Guideline, 3rd English Edition," National Institute of Advanced Industrial Science and Technology, Technical Report DigiARC-TR-2023-01, January 20, 2023.
[12] Software Engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE) — Quality Model for AI Systems, ISO/IEC 25059, 2023.
[13] "Ethics guidelines for trustworthy AI," https://data.europa.eu/doi/10.2759/346720, 2019.
[14] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation," https://arxiv.org/pdf/2010.16061.pdf, October 11, 2020.
[15] I. V. Tetko, D. J. Livingstone, and A. I. Luik, "Neural Network Studies. 1. Comparison of Overfitting and Overtraining," Journal of Chemical Information and Computer Science, vol. 35, no. 5, pp. 826-833, September 1995.
[16] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 2, pp. 1137-1143, August 1995.
[17] Y. Tsuzuki, H. Ohira, and S. Takahashi, "Techniques for Quantitative Evaluation of Noise Robustness of AI Models," Toshiba Review, vol. 76, no. 3, pp. 44-47, May 2021. (In Japanese)
[18] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified Adversarial Robustness via Randomized Smoothing," Proceedings of the 36th International Conference on Machine Learning, vol. 97, pp. 1310-1320, 2019.
[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," https://arxiv.org/pdf/1412.6572.pdf, Mar 20, 2015.
[20] Explainable Artificial Intelligence (XAI), Defense Advanced Research Projects Agency, DARPA-BAA-16-53, August 10, 2016.
[21] Guidelines on Assessment of AI Reliability in the Field of Plant Safety, 2nd ed., 2021.
[22] Japan Deep Learning Association (Ed.), The Official Textbook of the Deep Learning G Certificate, 2nd ed., Tokyo: Shoeisha, 2021. (In Japanese)