SQuaRE Matters: Reflection of Software Quality Evaluation, Benchmark, Pattern Classification and Practitioners' Perception through SQuaRE

Hironori Washizaki [1,2,3,4,]

[1] *Waseda University, 3-4-1 Okubo, Shinjuku City, Tokyo, 169-8555, Japan*
[1] *National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda City, Tokyo, 101-0003, Japan*
[1] *SYSTEM INFORMATION CO., LTD., 1-7-3 Kachidoki, Chuo City, Tokyo, 104-0054, Japan*
[1] *eXmotion Co., Ltd., 3-4-3 Kitashinagawa, Shinagawa City, Tokyo, 141-0032, Japan*

Abstract

The ISO/IEC 25000 Systems and software Quality Requirements and Evaluation (SQuaRE) series is a valuable framework to measure and evaluate quality from more multifaceted, objective, and standardized criteria across products and organizations. This talk introduces successful use cases of SQuaRE: software systems quality evaluation and benchmarking, and machine Learning and IoT system design patterns classification with practitioners' perception.

Keywords

Software quality, quality measurement, quality evaluation, standard, software development, machine learning systems

1. Introduction

The ISO/IEC 25000 Systems and software Quality Requirements and Evaluation (SQuaRE) series is a useful framework to measure and evaluate quality from more multifaceted, objective, and standardized criteria across products and organizations [1]. SQuaRE is independent of the domain or product. It assembles important quality characteristics, measurement values, and evaluation methods.

SQuaRE should be a valuable standard for various use cases, such as software evaluation and classification, from the viewpoint of quality attributes. This talk introduces successful use cases of SQuaRE: software systems quality evaluation and benchmarking, and machine Learning and IoT system design patterns classification with practitioners' perception.

2. Quality evaluation and benchmarking

Conventional quality evaluations of software concentrate on specific quality characteristics. Moreover, the measurement data are limited to particular products and organizations. Consequently, the present state of product quality and quality in use characteristics are not fully understood, preventing effective decision-making for software stakeholders. To alleviate this problem, ISO/IEC defined the SQuaRE series for comprehensive quality measurement and evaluation. However, these standards remain rather general and abstract, making them difficult to apply.

In these papers [2]–[4], we established a SQuaRE-based comprehensive software quality evaluation framework, Waseda Software Quality Framework (WSQF), which concretizes many product quality and quality in use measurement methods originally defined in the SQuaRE series.

4th International Workshop on Experience with SQuaRE Series and its Future Direction, December 06, 2022, Tokyo, Japan
EMAIL: washizaki@waseda.jp (A. 1)
ORCID: 0000-0002-1417-9879 (A. 1)

By applying the WSQF to 21 commercial ready-to-use software products, we revealed the status of software product quality. A resulted comprehensive benchmark includes trends of the quality measurement values, relationships among quality characteristics, the relationship between quality-in-use and product quality, and the relationship between the quality characteristics and product contexts within the limits of an application.

3. Machine learning design pattern classification with practitioners' perception

Machine learning (ML) software engineering design patterns encapsulate reusable solutions to commonly occurring problems within the given contexts of ML systems and software design. These ML patterns should help develop and maintain ML systems and software from the design perspective. However, to the best of our knowledge, there was no study on the practitioners' insights on the use of ML patterns for the design of their ML systems and software. In these papers [5], [6], we reported the results of a literature review to identify ML design patterns. We also reported a questionnaire-based survey on ML system developers' state-of-practices with concrete ML patterns. Furthermore, we described most of the identified patterns in previous papers [7]–[10].

Any design pattern should address one or more quality characteristics that are associated with design problems. For ML design patterns, we assumed that the product quality characteristics defined in the SQuaRE quality model (i.e., ISO/IEC 25010:2011), as well as ML model and prediction quality characteristics, can be addressed. We analyzed the quality characteristics by reading problems and solutions descriptions of the 15 ML design patterns and identifying related specific descriptions or keywords. Many ML design patterns address maintainability. Most operation patterns address model and prediction quality characteristics [6].

Furthermore, we surveyed 300+ software and ML developers who participated in an online seminar on ML patterns in July 2020 in terms of perception of quality characteristics considered in ML system design and development [5]. Out of the 300+ participants, 52 answered our questions, which corresponds to a response rate of around 17%. Most considered the functional suitability of the ML systems and software during design. This seems natural since the functionality is the most fundamental attribute of any system and software. In addition, more than 40% of the respondents considered the maintainability, reliability, security, and usability of the ML systems and software. In contrast, portability and compatibility were rarely considered. According to our pattern analysis, maintainability and reliability are well addressed in existing ML patterns, while security and usability are less addressed; more ML patterns focusing on security and usability are anticipated by accumulating more design cases since these characteristics are majorly concerned.

In terms of ML model and prediction quality characteristics considered when designing ML systems, the top concern was model robustness, followed by model explainability and prediction accuracy [5]. According to our pattern analysis, model robustness and prediction are well addressed in existing ML patterns, but model explainability is less addressed. These major characteristics are expected to be reflected in the future development and revision of ML-related quality model standards, such as the SQuaRE quality model for AI systems ISO/IEC DIS 25059 and the related AI-specific data quality measures ISO/IEC AWI 5259-2.

4. IoT system design patterns classification

We have applied a similar analysis to the Internet of Things (IoT) design patterns [11], [12]. IoT patterns, including IoT design and architecture patterns, have been published to document the successes (and failures) in IoT systems and software development [13].

IoT design patterns should mostly address interoperability, which is defined as a sub attribute of compatibility in SQuaRE since, by definition, IoT is about ensuring interoperability among objects. To classify IoT patterns, we used all quality attributes except for functional suitability defined in the SQuaRE quality model and selected terms from software engineering: performance, compatibility, usability, reliability, security, maintainability, and portability. We excluded functional suitability because certain functional requirements are often satisfied by concrete system and software design decisions, including the reuse of IoT platforms and software libraries, instead of the reuse of abstract architecture or design patterns.

We observed that some IoT patterns are dedicated to one or a few quality characteristics, while others address many characteristics [12]. According to SQuaRE, performance, usability, reliability, and security significantly influence the quality in use for primary users, while compatibility, maintainability, and portability greatly impact quality in use for secondary users who maintain the system. The former is an important concern of primary users, while the latter is about the ease of extending a system by maintainers in terms of performance. Furthermore, we identified potential additional quality characteristics for IoT as privacy and scalability.

5. Conclusion

This talk introduced successful use cases of SQuaRE: software systems quality evaluation and benchmarking and machine Learning and IoT system design patterns classification with practitioners' perception.

Future works can include further analysis of software systems and design patterns from the quality viewpoints, a relationship model among different quality characteristics, and suggesting new quality characteristics to be considered for quality models targeting ML and IoT.

6. Acknowledgements

This work was partially conducted as a part of the Research Initiative on Advanced Software Engineering in 2015, supported by Software Reliability Enhancement Center (SEC), Information Technology Promotion Agency Japan (IPA). This work was also supported by JST-Mirai JPMJMI20B8 Engineerable AI (eAI), JSPS JPJSBP 120209936, KAKENHI 21KK0179, and enPiT-Pro Smart SE.

7. References

[1] International Organization for Standardization, "ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, institution = International Organization for Standardization," ISO/IEC, Tech. Rep., 2011.

[2] H. Nakai, N. Tsuda, K. Honda, H. Washizaki, and Y. Fukazawa, "Initial framework for software quality evaluation based on ISO/IEC 25022 and ISO/IEC 25023," in QRS Companion. IEEE, 2016, pp. 410–411.

[3] ——, "A square-based software quality evaluation framework and its case study," in IEEE TENCON. IEEE, 2016, pp. 3704–3707.

[4] N. Tsuda, H. Washizaki, K. Honda, H. Nakai, Y. Fukazawa, M. Azuma, T. Komiyama, T. Nakano, H. Suzuki, S. Morita, K. Kojima, and A. Hando, "WSQF: comprehensive software quality evaluation framework and benchmark based on square," in ICSE (SEIP). IEEE / ACM, 2019, pp. 312–321.

[5] H. Washizaki, H. Takeuchi, F. Khomh, N. Natori, T. Doi, and S. Okuda, "Practitioners' insights on machine-learning software engineering design patterns: a preliminary study," in ICSME. IEEE, 2020, pp. 797–799.

[6] H. Washizaki, F. Khomh, Y. Gu´eh´eneuc, H. Takeuchi, N. Natori, T. Doi, and S. Okuda, "Software-engineering design patterns for machine learning applications," Computer, vol. 55, no. 3, pp. 30–39, 2022. [Online]. Available: https://doi.org/10.1109/MC.2021.3137227

[7] H. Washizaki, F. Khomh, and Y.-G. Gu´eh´eneuc, "Software engineering patterns for machine learning applications (sep4mla)," in 9th Asian Conference on Pattern Languages of Programs (AsianPLoP 2020). Hillside, Inc., 2020, pp. 1–10.

[8] H. Washizaki, F. Khomh, Y.-G. Gueheneuc, H. Takeuchi, S. Okuda, N. Natori, and N. Shioura, "Software engineering patterns for machine learning applications (sep4mla) – part 2," in 27th Conference on Pattern Languages of Programs in 2020 (PLoP'20). Hillside, Inc., 2021, pp. 1–10.

[9] J. Runpakprakun, S. R. O. Peralta, H. Washizaki, F. Khomh, Y.-G. Gueheneuc, N. Yoshioka, and Y. Fukazawa, "Software engineering patterns for machine learning applications (sep4mla) – part 3 – data processing architectures," in 28th Conference on Pattern Languages of Programs in 2021 (PLoP'21). Hillside, Inc., 2021, pp. 1–11.

[10] H. Washizaki, F. Khomh, and Y.-G. Gueheneuc, "Software engineering patterns for machine learning applications (sep4mla) – part 4 — ml gateway routing architecture," in 29th Conference on Pattern Languages of

Programs in 2022 (PLoP'22). Hillside, Inc., 2022, pp. 1–8.
[11] H. Washizaki, N. Yoshioka, A. Hazeyama, T. Kato, H. Kaiya, S. Ogata, T. Okubo, and E. B. Fern´andez, "Landscape of iot patterns," in SERP4IoT@ICSE. IEEE / ACM, 2019, pp. 57–60.
[12] H. Washizaki, S. Ogata, A. Hazeyama, T. Okubo, E. B. Fern´andez, and N. Yoshioka, "Landscape of architecture and design patterns for iot systems," IEEE Internet Things J., vol. 7, no. 10, pp. 10 091–10 101, 2020. [Online]. Available: https://doi.org/10.1109/JIOT.2020.3003528
[13] H. Washizaki, A. Hazeyama, T. Okubo, H. Kanuka, S. Ogata, and N. Yoshioka, "Analysis of iot pattern descriptions," in SERP4IoT. IEEE, 2021, pp. 21–26