dHealth 2020 – Biomedical Informatics for Health and Care
G. Schreier et al. (Eds.)

doi:10.3233/SHTI200088

A Systematic Approach to Quality Requirement Management in Medical Software

Karin MESSER-MISAK[a,1], Jeroen S. DE BRUIN[a] and Sten HANKE[a]
[a] *Institute eHealth, Department of Applied Computer Sciences, FH Joanneum University of Applied Sciences GmbH, Graz, Austria*

Abstract. Background: The quality requirements for medical software have become increasingly demanding. Several quality standards and models are already in place, but there is a debate on whether these are specific enough for medical software. Moreover, mapping requirements to quality criteria can be challenging but is required throughout the software development process. Objectives: We propose a workflow in which we apply proven methods and tools for systematic collection, analysis, and evaluation of software quality criteria based on the ISO/IEC 25010:2011. Methods: We employ affinity diagrams, Kano analysis and quality function deployment for the systematic requirement development, analysis, and management. Results: We outline a systematic approach on how to use the recommended process when developing medical software. Conclusion: The paper proposes a systematic approach for requirements management that could be used for mapping medical software quality criteria and stakeholder requirements, independent from the quality criteria (and the underlying model) itself.

Keywords. software design, reference standards

1. Introduction

Current developments in healthcare Information and Communication Technology (ICT) include the design and development of increasingly complex integrated systems comprising both software and hardware, as well as the need for connectivity of such systems across the healthcare sector. As a result, systems are more difficult to certify, more difficult to abstract from other software and hardware components more difficult to understand concerning decision-making. With increasing healthcare demands, these systems will inevitably outgrow medical professionals' capabilities to deliver safe, quality care on time.

An underlying factor that complicates the medical application of such technologies resides in the legislative landscape, which is currently undergoing a process of continuous (re-)definition of guidelines for the development and application of medical devices integrating complex software and hardware components.

What legislative and healthcare perspectives on medical software have in common, is the requirement of delivering high-quality systems, software, and services. To fulfill

[1] Corresponding Author: Karin Messer-Misak, Institute eHealth, FH Joanneum University of Applied Sciences GmbH, Eckertstraße 30i, 8020 Graz, Austria, E-Mail: Karin.Messer-Misak@fh-joanneum.at

this prerequisite, a framework is needed that comprises: i) quality criteria for medical software and hardware development, ii) a process to validate software as well as hardware requirements against these criteria and iii) a process to monitor the development process and to track, if requirements meet quality criteria, during the project.

In the end, the assessment of the effect of software quality on patient treatment quality, healthcare institution efficiency and applications like clinical trials is essential. Especially in the field of medical software development, several standards (such as ISO 13485:2016 Medical devices, IEC 62304: Software life cycle processes 29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering, etc.) have been established. What users usually lack are concrete tools to optimally use these standards.

The paper aims to introduce a method on how to apply well-established methods from quality management (QM) as a process, which might also be used in the medical domain to control, manage and evaluate systems against certain requirements. Given the current legislative landscape, available standards and best-practices, the authors have worked out which methods from QM can contribute to a better implementation of the current regulations for software development in the medical sector.

2. Methods

2.1. Current guidelines and standards

Although models that structure quality criteria for software and hardware are available, their applicability to medical software and systems is still a matter of debate; there is a discussion ongoing about the criteria in general because application and software quality is a multifaceted concept determined by several properties [1].

Over time, different qualitative models have been proposed and used as a basis to describe and delineate the different system and software quality attributes. The ISO/IEC standards 9126, 14598 and 25000 (SQaRE) are widely referred to as standard frameworks for developing complementary or alternative models for evaluating specific software or developing custom models.

The International Standard Organization (ISO) and International Electrotechnical Commission (IEC) introduced the ISO/IEC 9126 standard in 1991. This model is used for the definition and integration of various characteristics of software quality and has been used to assess the impact of software quality characteristics on healthcare outcomes. Since then, ISO/IEC 25000 was introduced, which attempts to harmonize, unify and update the international standards for evaluating software quality. This standard has replaced ISO/IEC 9126, as it is considered more suitable for evaluating the quality of ICTs for healthcare use from the perspective of institutional acquisition [2].

Software quality requirements can be complex, and their application and priority depend on the application under consideration, as well as the application environment and stakeholders. These criteria are normally assorted in functional and non-functional requirements. Aghazadeh et al. [1] proposed six healthcare indicators concerning software quality characteristics based on a questionnaire with health experts. The characteristics are 1) User satisfaction, 2) Quality of patient care, 3) Clinical Workflow and Efficiency, 4) Care provider's communication and information exchange, 5) Patient satisfaction and 6) Care costs.

In the software development process, quality criteria need to be mapped against requirements and tracked continuously. Although several models regarding quality criteria exist, currently there is no structured approach of applying them. Moreover, as requirements can vary depending on the application and application field, a process needs to be flexible to be able to adopt it to multiple use cases.

2.2. Use of quality models

Requirements management plays a decisive role in the success of software projects. These requirements are often formulated abstractly as visions or goals. Ideally, it is more productive to formulate general requirements that are valid for the entire system, special requirements that are relevant for certain parts of the system, and so-called "atomic" requirements as the most concrete and unambiguous form of the requirement.

The challenge is to identify, analyze, prioritize, agree on and document the requirements. In summary, concrete descriptions are essential for the following types of requirements: efficiency requirements, usability requirements, reliability requirements, availability, changeability, security, other operational requirements, and legal requirements.

According to [7], there are a variety of activities that can benefit from the use of quality management tools, e.g., the identification of software and system requirements, the validation of the completeness of a requirement definition, the identification of software and system design and test objectives, the definition of quality characteristics and control criteria in the context of quality assurance, and the identification of acceptance criteria for a software product and/or software-intensive computer system. These activities, methods, and tools must be carefully coordinated and consistently used throughout the entire product development process to ensure that the required quality requirements can be verified in the final product.

To identify the optimal set of software and system requirements and verify them for completeness, a comprehensive analysis of stakeholder requirements must be carried out. These requirements are to be specified as criteria, i.e. described in concrete terms, and measurement methodologies or metrics need to be defined for evaluation purposes. To achieve accurate and comprehensive results, it is recommended to work on this process in a stepwise fashion in interdisciplinary teams and to employ quality management tools and methods.

3. Results

In this section, we present our systematic approach for quality requirement management in medical software based on existing, proven standards and tools in the medical software development domain. The workflow of this approach is shown in **Figure 1**.

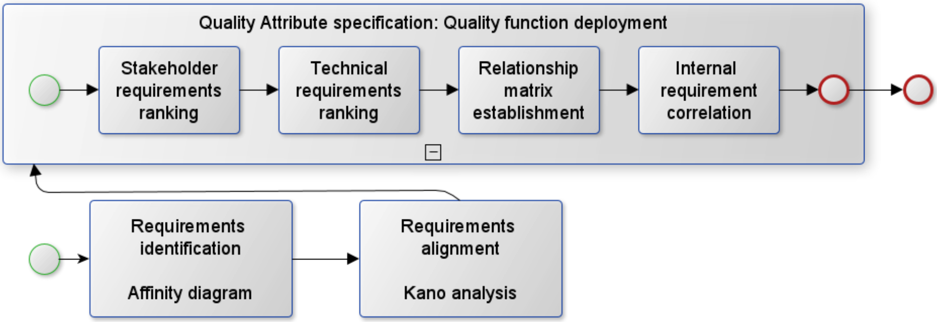
Quality Attribute specification: Quality function deployment
Stakeholder requirements ranking
Technical requirements ranking
Relationship matrix establishment
Internal requirement correlation
Requirements identification
Affinity diagram
Requirements alignment
Kano analysis

Figure 1. Business Process Modeling and Notation workflow for quality requirement management in medical software

3.1. Requirements identification and collection

Identifying software quality requirements early in the software life cycle as a part of the requirements specification is considered crucial. At present, quality requirements are commonly elicited from literature, checklists, and users. We can distinguish between primary users (people who interact directly with the system), secondary users (e.g., support or content provider, system managers) and indirect users (person who receives output without interacting with the system) [7].

Usually, a survey is employed for requirements identification, e.g., with the help of an affinity diagram (Figure 2). Complete sets of requirements are most likely to be achieved through systematic collaboration. This enables user and system requirements, design requirements, etc. to be comprehensively identified, named, and categorized. As a result. the relationships between the individual information groups become transparent.

In software development, it makes sense to use the models "Quality in Use" and "Product Quality" and their attributes mentioned in ISO/IEC 25010:2011 and then to extend and supplement them accordingly (Figure 1). Examples of the health applications applying the guideline include Alves et al. [4] who validated a poison central system as well as Kadi et al. [5] who evaluated the quality requirements of a pregnancy monitoring system, both using the ISO/IEC 25010 standard.

3.2. Use of Kano-analysis for Requirements alignment

Following requirement collection, requirements are further specified from the stakeholder perspective using a Kano analysis [6]. In this process, the relationship between the achievement of certain characteristics of the requirements and the satisfaction expected by the stakeholder/user is established.

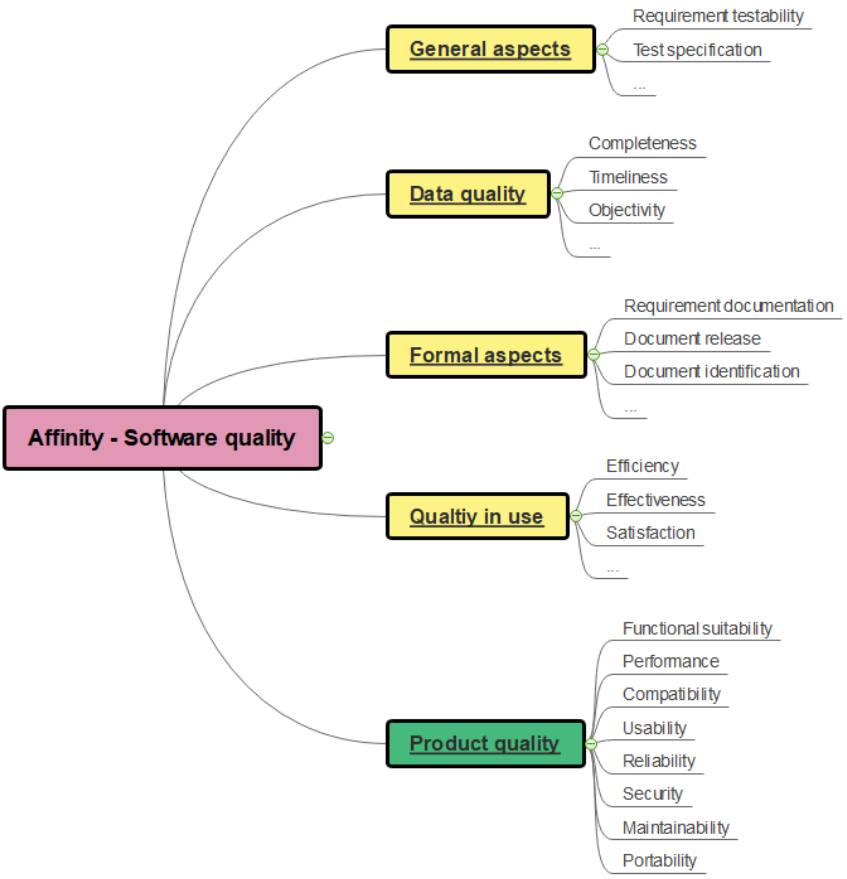

Figure 2. Example of an affinity diagram incorporating ISO/IEC 25010:2011

The basic characteristics are “implicit must” criteria, i.e., criteria that are not articulated explicitly but tacitly presupposed. Performance characteristics are requirements explicitly demanded by the stakeholders, which have a high influence on satisfaction. The “Enthusiasm characteristics” are auxiliary criteria that are not explicitly expected by the stakeholders but increases their satisfaction substantially when fulfilled. Finally, insignificant characteristics (thus characteristics, whose existence leads neither to satisfaction nor to the dissatisfaction) and rejection characteristics (characteristics that cause dissatisfaction when implemented). Each requirement is aligned to a category before it is further specified (Figure 3).

3.3. Quality attribute specification

Stakeholder requirements are translated into specifiable and measurable quality characteristics. In this process, the target criteria to be achieved are defined, control criteria for quality assurance are defined and the acceptance of the software product is presented in a matrix.

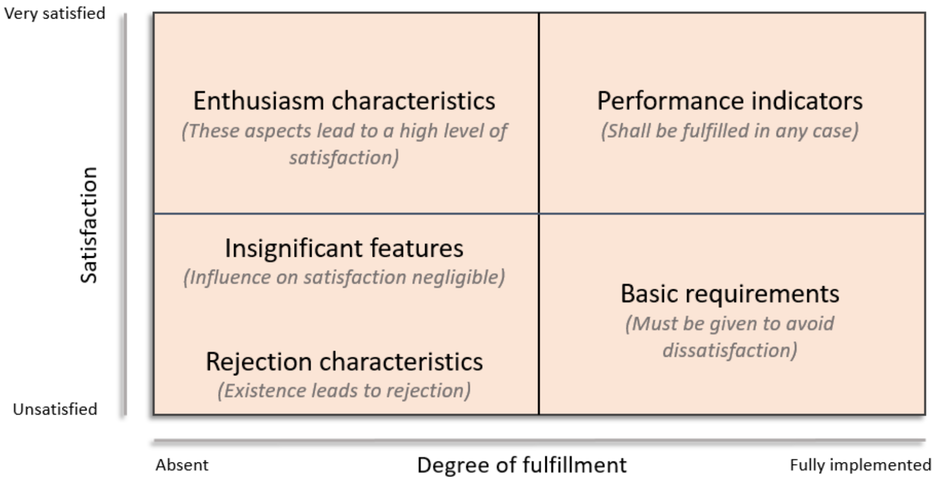
Very satisfied
Satisfaction
Unsatisfied
Enthusiasm characteristics
(These aspects lead to a high level of satisfaction)
Performance indicators
(Shall be fulfilled in any case)
Insignificant features
(Influence on satisfaction negligible)
Rejection characteristics
(Existence leads to rejection)
Basic requirements
(Must be given to avoid dissatisfaction)
Absent
Degree of fulfillment
Fully implemented

Figure 3. Kano-analysis for requirements alignment matrix to incorporate the requirements in terms of their relevance.

The "Quality function deployment" method is suitable for planning. Especially when interfaces and contradictions occur, these can be easily identified by correlations. It shows "what" the stakeholders want and "how" the requirements are technically implemented. The method is supplemented by target values to be defined, which are defined by fixed values in a corresponding unit. Here, too, a systematic approach is recommended (Figure 4):

1. Stakeholder requirements ranking: After the structuring and weighting of the requirement analysis results, these are displayed on row level. By weighting these entries (e.g., between 1-9) from the Kano analysis, the *relevance* for the stakeholders is represented
2. Technical requirements ranking: The column level lists how stakeholder requirements can be met by functional requirements
3. Relationship matrix establishment: The matrix is used to qualify the relationship between stakeholder and technical requirements. For simplicity's sake, the *rating* is bucketed in four levels: no relationship, weak relationship, medium relationship or a strong relationship. The final scores are then calculated for each stakeholder and technical requirement (rating * relevance) and summarized as target values (vertically summation) and competitive assessments (horizontal summation).
4. Internal requirement correlation: To check for mutual influences and dependencies, both the stakeholder requirements and the technical requirements can be correlated with themselves.

The method can be extended as required and can be flexibly adapted to suit specific requirements. Yoji Akao - one of the developers of the method - coined the sentence "Copy the spirit, not the form" [9]. Herzwurm, et al. [10] show that it makes sense to adapt existing methods and tools to the requirements.

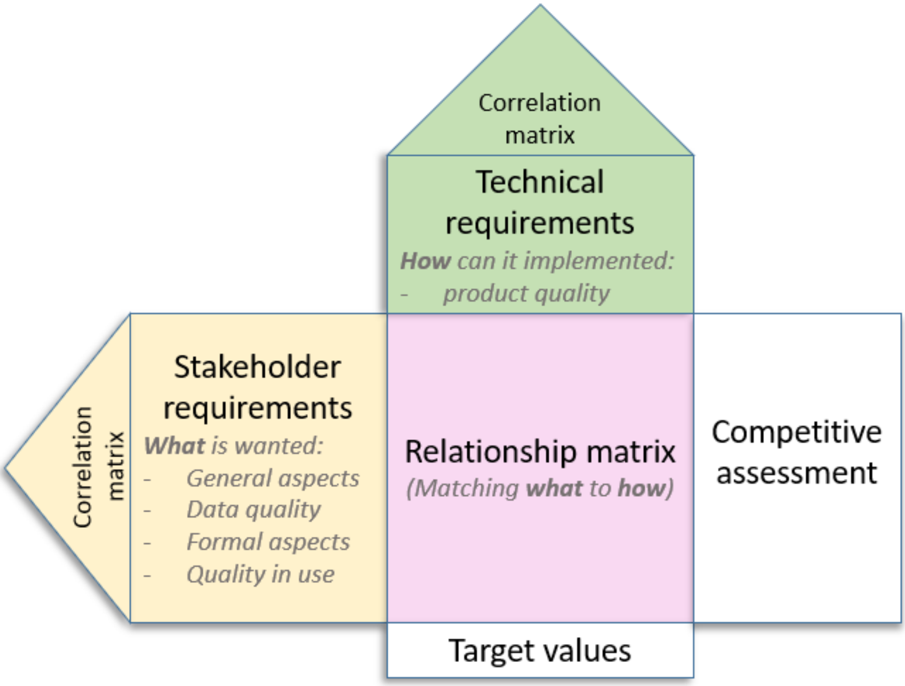

Figure 4. Quality function deployment graph following Yoji Akao [8].

4. Discussion

Research to improve the quality of the software is generated due to users' demand for software products with increasing quality. This demand especially applies to the development and design process of health and medical software as well as hardware.

The IEEE Standard Glossary of Software Engineering Terminology [12-14] defines the quality of software products as 1) the degree to which a system, component or process meets specified requirements and 2) the degree to which a system, component or process meets the needs or expectations of a user. Software quality is therefore directly linked to the requirements of different stakeholders.

As the complexity of software rises a defined process for tracking the mapping of quality criteria as well as requirements can help to provide better software quality as well as better usability. The quality aspect is an essential factor in ensuring security and stakeholder satisfaction, i.e. success in the use of software products, especially in health care. Potential negative effects should be excluded from the outset and positive aspects should be identified and promoted. Quality characteristics must be known, specified, measured and evaluated. Additionally, high-quality software systems have been realized within the planned time and at the estimated costs [11].

The paper describes a stepwise approach that could be followed to address the mentioned challenges. The suggested approach has been adapted from a well-established process and tool taken from the field quality assurance and quality management. As a next step, this approach will be used in the assessment of software quality for remote decentralized clinical trials as part of the Innovative Medicines Initiative [2] Joint Undertaking (H2020-JTI-IMI2) Trials@Home project [15, 16].

Funding statement

The research leading to these results has received support from the EU/EFPIA Innovative Medicines Initiative [2] Joint Undertaking (H2020-JTI-IMI2) Trials@Home grant n° 831458.

References

[1] S. Aghazadeh, H. Pirnejad, A. Aliev, and A. Moradkhani, Evaluating the effect of software quality characteristics on health care quality indicators, *Indicators,* **2**(3), (2015), 67–73

[2] A. V. Ordozgoiti, P. D. Hito, E. M. Guix Comellas, C. M. F. Sanchez, M. G. Hernandez, and T. L. Canut, *Software quality evaluation models applicable in health information and communications technologies. A review of the literature*, in Studies in Health Technology and Informatics, **226**, 2016, 169–172.

[3] A. Idri, M. Bachiri, and J. L. Fernández-Alemán, A Framework for Evaluating the Software Product Quality of Pregnancy Monitoring Mobile Personal Health Records, *J. Med. Syst*., **40** (3), (2016), 1–17

[4] J. M. Alves et al., *Quality evaluation of poison control information systems: A case study of the DATATOX system*, in Proceedings - IEEE Symposium on Computer-Based Medical Systems, (2016), 30–35.

[5] I. Kadi, A. Idri, and S. Ouhbi*, Quality evaluation of cardiac decision support systems using ISO 25010 standard*, in Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, (2016)

[6] N. Kano, N. Seraku, F. Takahashi, F. Tsuji, 1984, S.5, quoted after E. Sauerwein, *Das Kano-Modell der Kundenzufriedenheit*, Springer Fachmedien, Wiesbaden, (2000), 27.

[7] BSI Standards Publication. *Systems and Software engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*, BSI, (2011), 1.

[8] G. Martin, G. Möhrle, K. Hartung and W. D. Spilgies, *Quality Function Deployment for bundled services., in Innovation Management in the Service Industry. Basics, practical examples and perspectives*. Edited by Klaus Schmidt/Roland Gleich/Ansgar Richter. 39-160, Haufe Publishing House, Freiburg. 2007, 146 - quotes Akao 1992, 19.

[9] Herzwurm, Schockert, Dowie u. Breidung, *Requirements Engineering for Application Development in Volatile Project Environments Using Continuous Quality Function Deployment (CQFD)*, in Proceedings of the International Conference on Software Engineering Research and Practise, **1**, (2003) 440-447.

[10] B. Hohler, *Qualitätsmanagement bei der Softwareentwicklung*, 827-866 in: Masing Handbuch Qualitätsmanagement, Carl Hanser Verlag, 2014, München, Wien, 831.

[11] IEEE. (1990). IEEE Std 610.12-1990 (1990)- *Standard Glossary of Software Engineering Terminology,* http://web.ecs.baylor.edu/faculty/grabow/Fall2013/csi3374/secure/Standards/IEEE610.12.pdf, last access: 20.03.2020.

[12] IEEE. (1998). Standard for Software Maintenance, Software Engineering Standards Subcommittee of the IEEE Computer Society.

[13] S. Durgesh, W. Su-Hua and C. Dengjie, *Quality Models: Role and Value in Software Engineering*, 2nd International Conference on Software Technology and Engineering(ICSTE). 320-324.

[14] IEEE, IEEE Standard Glossary of Software Engineering Terminology, http://web.ecs.baylor.edu/faculty/grabow/Fall2013/csi3374/secure/Standards/IEEE610.12.pdf, last access: 20.3.2020.

[15] Innovative Medicines Initiative, Trials@Home> Center of excellence – remote decentralised clinical trials, https://www.imi.europa.eu/projects-results/project-factsheets/trialshome, last access: 20.3.2020.

[16] Trials@Home, Centre of Excellence Remote and Decentralised Clinical Trials, https://trialsathome.com/, last access: 20.3.2020.