MEDINFO 2019: Health and Wellbeing e-Networks for All
L. Ohno-Machado and B. Séroussi (Eds.)

doi:10.3233/SHTI190347

IEC 62304 Ed. 2: Software Life Cycle Standard for Health Software

Alpo Värri[a], Patty Kranz-Zuppan[b], Richard de la Cruz[c]

[a]Tampere University, Tampere, Finland
[b] Medtronic, plc, Mounds View, Minnesota, USA
[c] Silver Lake Group, Inc. (SLGI), Minnetonka, Minnesota, USA

Abstract

The quality of software is high in medical devices due to the strict regulatory requirements and their implementation in the software development processes through the use of the IEC 62304 standard. The goal of this standard revision project was to extend the scope of the standard to all health software and also to bring the requirements of the 12 year old standard back to the state-of-the-art including provisions for cybersecurity. The joint IEC/SC62A and ISO/TC215 project team revised the standard and adapted its risk management, usability, and security requirements to serve both the medical device industry and the overall health software industry. The resulting second version of the standard has gone through a multistage global voting process to achieve a consensus of the requirements to serve both these communities. The resulting standard has potential to have a major impact on the quality of software used in health care globally.

Keywords:
Standards; Health Care Technology; Software Engineering

Introduction

The quality of software in medical devices is very important because a failure in medical device software can have fatal consequences to the patients in some cases. A classic example of this is the case of the Therac-25 radiation therapy machine in the 1980's [1]. The weaknesses in the production processes of the software to this device resulted in undetected programming errors which led to massive radiation overdoses in at least six accidents and three dead patients. It is a well-known fact in the software engineering field that the testing of the software does not alone ensure sufficient quality of the software, but the production process of the software needs to be of high quality, as well. Few, if any other documents have a greater impact on how the software development processes have been arranged in the medical device industry than the IEC standard 62304 Medical device software – Software life cycle processes [2]. This standard is recognized by the Food and Drug Administration in the USA. It is also on the list of harmonized standards under Directive 93/42/EEC for Medical devices in the European Union; although, it does not necessarily cover all the requirements of the Directive 2007/47/EC for medical devices [3]. The medical device industry applies this standard because it may be the easiest way to demonstrate the conformity of the company's medical device software production process with the regulatory requirements. The regulatory requirements suggest that the standard is typically used in conjunction with the quality system standard ISO 13485 [4], the risk management standard ISO 14971 [5], and the medical device usability standard IEC 62366-1 [6], which are harmonized standards as well. The standard has a three class software safety classification A, B, and C of which class C represents the highest risk class software.

The growing needs to improve the quality of all the software used in health care has led to the idea to extend the scope of the 62304 standard beyond medical devices to all health software. The term health software is defined to include all software used for managing, maintaining, or improving the health of individual persons, or the delivery of care, still also including the software in medical devices. In the regulatory meaning, the medical device software can be embedded to a device or it can be a medical device as such, so-called Software as a Medical Device (SaMD). Software with a health purpose runs as a service, e.g. in a cloud service, could be a medical device if it fulfills the definition of a medical device, but if not, it is still covered by the definition of health software. This independence of the hardware platform means that this definition and the new version of the 62304 standard cover the popular mobile health apps.

After the approval of the revision proposal, the work on the second edition of the 62304 standard was (re)started in October 2014. The revision work was carried out in a project team of the Joint Working Group seven (JWG7), which is a co-operation working group between IEC subcommittee 62 A *Common aspects of electrical equipment used in medical practice* (IEC/SC62A) and ISO technical committee 215 *Health Informatics* (ISO/TC215) because both committees are interested in contributing to this work. The project team consisted of experts nominated by the national standards bodies of the member countries of these standardization committees.

The health informatics standards are not very well-known by the health informatics research community, partly because the standards are relatively expensive. For this reason, it is important to introduce the new version of the 62304 standard here also to the scientific community, because health informatics researchers may later face it anyway if they want to commercialize their research results.

Methods

Design Specification

The work began by defining the target of the second version of the standard. The highlights of the planning were:

- The scope of the standard would be extended from medical device software to cover all health software
- The standard would remain a life cycle standard in contrast to being a product standard
- The software maintenance process requirements would remain in the standard
- The same software safety classification would remain in the standard and the required rigor to the safety class C software would be preserved
- Significant changes to the requirements should be avoided unless there is a compelling motivation for the change
- The new version of the standard should be applicable to fulfill the regulatory requirements of medical device software

The expansion of the scope to all health software expands the user base of the standard significantly to new audiences. The designers of non-medical device health software have often not applied equally strictly controlled processes as the medical device manufacturers and the required level of rigor may come as a surprise to these companies when their customers begin to require the use of 62304 ed. 2 in the software production.

Keeping the standard as a life cycle standard instead of a product standard leaves out two main activities of the standard. A product standard such as the IEC 82304-1 [7] contains also software use requirements and the validation that the use requirements are met but these are out of the scope of this standard.

The need to keep essential changes minimal comes from the high costs that major changes could cause to the industry when they would have to change their processes and train the personnel for the changes. Major changes might also make it more difficult for the regulatory bodies to accept the standard as a way to fulfill the regulatory requirements, particularly if the changes are considered to result in less safe software than before.

Literature Review

When the revision was started, a literature review was carried out as well. For a standardization project, other related standards are the most relevant literature, because a standard can refer to other standards normatively if the other standards contain material that is not feasible to repeat in the current standard. As the list of potentially relevant standards spans a few pages, only a small subset of them is mentioned here. An earlier version of the ISO/IEC/IEEE 12207 [8] standard was used in the drafting the first version of 62304. Its latest developments were checked to maintain a certain level of compatibility to general software development standards. The IEC 60601-1 standard [9] contains requirements for medical device software and they were compared to the requirements in 62304. The risk management subclauses of 62304 were checked against ISO 14971 [5] and its draft versions under development. A similar comparison was carried out with ISO 13485 [4]. ISO 90003 [10] was checked as an alternative to ISO 13485 for health software which is not classified as a medical device. In addition to the harmonized usability standard IEC 62366-1 [6], the corresponding medical device standard IEC 60601-1-6 [11] was also considered.

The IEC 61508-3 functional safety of programmable electronic systems standard [12] was checked for comparison although its field of application is not in the medical domain. The ISO/IEC TR 29110 series [13] was identified to contain recommendations for software life cycle management in small organisations but its requirements were not sufficiently strict for high risk medical device software production. SWEBOK V3 [14] was used as the latest state-of-the-art document for software engineering.

During the course of the work, it became apparent that the cybersecurity issues in medical devices and health software in general became more and more important. A cybersecurity problem in a medical device can develop into a patient safety problem too, if the problem is in a safety critical device. Previously, when medical devices were isolated to their own networks, the cybersecurity issues could be rather safely ignored by the medical device community, but in today's interconnected world, their addressing has become mandatory. Regulatory cybersecurity guidance for manufacturers of medical devices has been published and already revised in the USA [15] and corresponding guidance is expected from the European Commission as well. Moreover, the health software which is not embedded in a medical device is typically at risk of cybersecurity attacks as personal medical data is more valuable to cybercriminals than credit card numbers [16].

A large number of security-related standards have been identified by the project team members. The NIST Cybersecurity Framework [17] can be taken as the starting point for improving security in software development but other alternatives exist too. The project team investigated the applicability of several security standards. The ISO 27799 health informatics security standard [18] refers to the general ISO/IEC 27002 standard [19] but they are not very much related to the software life cycle processes. The IEC/ISO 80001 series of standard documents [20] and particularly TR 80001-2-2 [21] and TR 80001-2-8 [22] are also relevant because they inform the health delivery organizations how to manage security risks with networked medical devices which run software developed according to 62304. These standards introduce security requirements to software which need to be addressed in the software development life cycle. The IEC 62443 series of standards, particularly the IEC 62443-4-1 [23] from the industrial automation field is a good reference source for secure development life cycle requirements. The Microsoft Security Development Lifecycle model [24] is an alternative security-aware software design model from the general software development field. Finally, the ISO/TR 17791 [25] contained a survey of standards for enabling safety in health software; thus, the project team had a good comprehension of the existing literature to be considered.

Working Method

At the outset of the work, the project team had the draft amendment [26] of the 62304 standard in its disposal. It updated parts of the first version of the 62304 without the scope extension to health software. The project team began to introduce changes to the document based on the design specification and also on the feedback received from the field in the application of the 62304.

The work was carried out in a series of meetings. In between the meetings, the core project team members formulated new paragraphs of text to the subsections to be revised. The project team has produced three drafts of the standards, which have been circulated for voting in the both standardization committees IEC/SC62A and ISO/TC215. The national standardization bodies have collected the comments of each member country, and they have been delivered to the project team for handling. The project team is responsible of

addressing each one of the comments and either (partially) approve the comment or reject it with sufficient motivation. For example, the most recent vote resulted in about fifty pages of comments to be considered, indicating great interest in the contents of the standard. The governing working group needs to accept the project team's disposition of comments before the draft standard can proceed to the next stage. When all the comments have been addressed, the standard draft is sent to the Final Draft International Standard (FDIS) vote, which decides the approval of the document as an international standard. This is the normal working method of the international standardization organizations like IEC and ISO.

Results

At the time of writing, the latest draft addresses around 95 per cent of the received comments in agreement of the stakeholders. The scope has been extended to all health software and the necessary changes relating to this change have been implemented to the standard. The draft 62304 second edition standard covers now the following stages for the software development process:

1. Software risk management process
2. Software development planning
3. Software requirement analysis
4. Software architectural design
5. Software detailed design
6. Software unit implementation
7. Software integration and integration testing
8. Software system testing
9. Software release
10. Software configuration management
11. Software problem resolution

The software maintenance process is similar to the development process, but the stages 2 and 3 have been replaced with the establishment of the software maintenance plan and the problem and modification analysis.

Both the old and new versions of the 62304 require the use of a quality management system in the software production. The old version of the standard required that the manufacturer applied a risk management process complying with ISO 14971 [5]. The new version requires the conformity to ISO 14971 only when a software failure can contribute to a hazardous situation, which in turn can lead to injury or damage to the health of people, or damage to property or the environment. The new risk management process must also manage risks associated with security. The new version does not make any security standard mandatory, but it suggests some that can be used.

The new version makes the requirement to apply a usability engineering process explicit. The use of the IEC 62366-1 [6] is not mandatory but it is given as an example of how to demonstrate the conformity to the usability design requirement.

The software safety classification still has the three classes, A, B, and C. The assignment of the software to these classes has be clarified. The procedure can be seen in Figure 1.

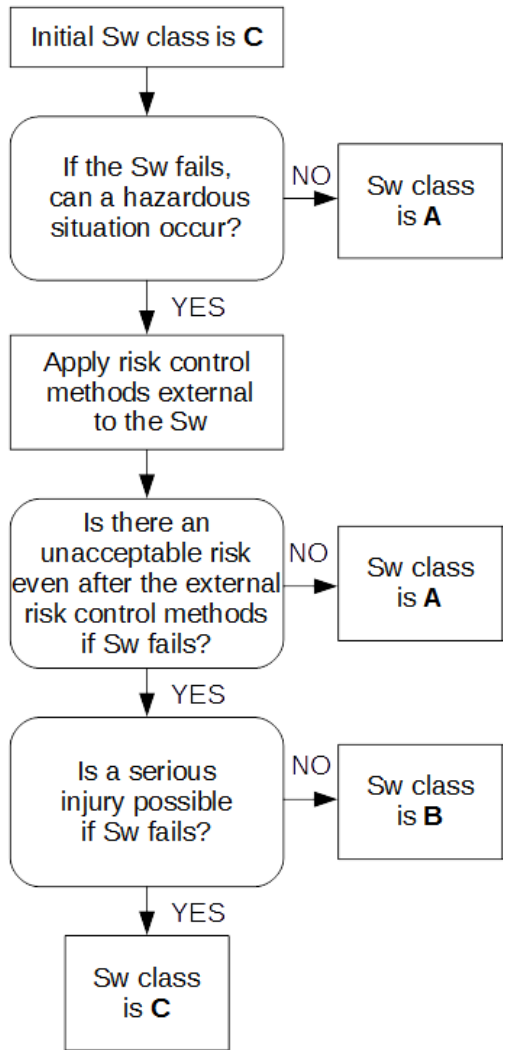

Figure 1. The Procedure to Assign the Safety Class to Software.

Legacy software is software that was produced before the new version of the standard was published, and it was therefore not possible to produce the software according to this standard. The new version has provisions how to demonstrate the compliance of the legacy software with respect to the new version.

The changes in the requirements for the software development process are relatively minor. There are a few additional requirements for the software system test record contents, but otherwise the changes are not very significant.

Similarly, the changes in the requirements for the software maintenance process are few. The new version requires that the change requests are analyzed also with safety class A software, which was not a requirement before.

The software risk management process has new requirements. More potential causes of contributions to hazardous situations must now be identified, including those related to cybersecurity.

The changes to the software configuration management process are again minor. The same applies to the changes in the software problem resolution process.

Over half of the document contains rationale and guidance in the implementation of the standard. Much of the text has remained the same or has been revised for additional clarity. The extended risk management sections now include new material which explains IT risks too. The relationships between the software safety classification and risk management are explained in more detail as well. The risk management of legacy software contains also additional new information.

The informative Annex C about the relationship to other standards has been partially rewritten. New standards are introduced to the table of standards to be considered. Accordingly, the bibliography now contains references to updated sources.

Discussion

The new version of the 62304 software life cycle processes standard extends its scope to all software production for health purposes while maintaining the current user base of medical device software producers. It remains to be seen, how widely the new audiences begin to use the standard voluntarily, because its implementation to the processes of a software company can be a significant effort. This is particularly the case with small software companies that have not produced regulated software before. Larger software companies with established quality systems may have an easier task in adjusting their processes.

The benefits of beginning to apply the revised 62304 standard to a company which has not applied a standard-based process before are the following:

1. The planning, programming, testing, and the documentation of the software becomes more controlled
2. As a result of the above, the final product will have fewer errors
3. The company will have a well-structured process to handle software updates, planned or corrective updates
4. The reputation of the company improves because the audited proof of high-quality software production processes is not so easy to achieve

For the customer, the benefit is that the products produced under the state-of-the-art software production standard are likely to be of higher quality than those produced without this methodology.

When the health software buying organizations hear about the completion of the second edition of the 62304 standard with its extended scope, they may decide to use it in procurement in order to improve the quality of the software they buy. When this happens, software companies wishing to stay competitive need to begin to apply this standard. The national regulators of health software in developed countries may also recognize their opportunity to improve the health software quality in their countries by requiring conformity to this standard. This would ensure the success of this standard in the field.

The additional requirements regarding cybersecurity do not lengthen the standard much, but their inclusion demands changes in the organizations applying the standard. The many publicly well-known cybersecurity incidents have probably triggered most members of the industry to react to the threats even without the requirements from this standard.

As the new standard has not yet passed the final vote, all the details of the standard have not been frozen. There is still discussion about the requirements regarding risk management, but the discussion will converge to a conclusion which the majority of the national standards bodies can approve.

The update of a standard causes consequences in other standards, as well. At least documents like IEC 82304-1 [7] and IEC/TR 80002-1 [27], and IEC/TR 80002-3 [28] need to be partly revised.

It is interesting to note how little direct impact the scientific literature actually has in the final version of this kind of a standard. The standard preparation work needs to consider other closely related standards more than the scientific works. Additionally, comments from the national ISO member country votes need to be considered in order to reach consensus. In this process, new radical ideas from the research community can easily get lost.

Conclusions

This paper presented the goals, methods, and results of the IEC 62304 Software life cycle processes standards for health applications. The revision of the standard brings the standard to the state-of-the-art in software production including provisions to cybersecurity and extends its potential user base to all health software producers. Thus, it has potential to have a major impact to the quality of software used in health care.

Acknowledgements

The authors want to thank all the 62304 edition 2 project team members for their contributions to the work. The support of the Finnish Standards Association (SFS) to the travel expenses to the project team meetings is also acknowledged.

References

[1] Therac-25, http://en.wikipedia.org/wiki/Therac-25 [accessed March 30, 2019]
[2] IEC 62304:2006 Medical device software – Software life cycle processes, https://www.iso.org/standard/38421.html [accessed November 22, 2018]
[3] European Commission, Harmonized Standards, Medical Devices, http://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/medical-devices_en [accessed November 22, 2018]
[4] ISO 13484:2016 Medical devices -- Quality management systems -- Requirements for regulatory purposes, https://www.iso.org/standard/59752.html [accessed November 22, 2018]
[5] ISO 14971:2007 Medical devices -- Application of risk management to medical devices, https://www.iso.org/standard/38193.html [accessed November 22, 2018]
[6] IEC 62366-1: 2015 Medical devices -- Part 1: Application of usability engineering to medical devices, https://www.iso.org/standard/63179.html [accessed November 22, 2018]
[7] IEC 82304-1 Health Software -- Part 1: General requirements for product safety,

https://www.iso.org/standard/59543.html, [accessed November 22, 2018]
[8] ISO/IEC 12207:2017 Information technology – Software life cycle processes, https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:12207:ed-1:v1:en [accessed November 22, 2018]
[9] IEC 60601-1:2005, Medical electrical equipment – Part 1: General requirements for basic safety and essential performance, IEC 60601-1:2005/AMD1:2012.
[10] ISO/IEC 90003:2014, Software engineering – Guidelines for the application of ISO 9001:2008 to computer software, https://www.iso.org/obp/ui/#iso:std:iso-iec:90003:ed-2:v1:en [accessed November 22, 2018]
[11] IEC 60601-1-6:2010+AMD1:2013 CSV, Medical electrical equipment – Part 1-6: General requirements for basic safety and essential performance – Collateral standard: Usability, https://webstore.iec.ch/publication/2596 [accessed November 22, 2018.
[12] IEC 61508-3, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements.
[13] ISO/IEC TR 29110-1:2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 1: Overview, JTC1/SC07, http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html [accessed November 10, 2014]
[14] P. Bourque, R.E. Fairley, Eds., Guide to the Software Engineering Body of Knowledge Version 3.0, SWEBOK V3, https://www.computer.org/web/swebok/v3 [accessed November 22, 2018]
[15] FDA, Content of Premarket Submissions for Management of Cybersecurity in Medical Devices, Draft Guidance for Industry and Food and Drug Administration Staff, USA 18.10.2018,https://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM623529.pdf [accessed November 22, 2018]
[16] C. Humer, J. Finkle, Your medical record is worth more to hackers than your credit card. Reuters Technology News, 24.9.2014, https://www.reuters.com/article/us-cybersecurity-hospitals/your-medical-record-is-worth-more-to-hackers-than-your-credit-card-idUSKCN0HJ21I20140924 [accessed November 22, 2018]
[17] NIST Cybersecurity Framework, version 1.1, April 2018, USA, https://doi.org/10.6028/NIST.CSWP.04162018 [accessed November 22, 2018]
[18] ISO 27799:2016 Health informatics -- Information security management in health using ISO/IEC 27002, https://www.iso.org/obp/ui/#iso:std:iso:27799:ed-2:v1:en [accessed November 22, 2018]
[19] ISO/IEC 27002:2013 Information technology - Security techniques - Code of practice for information security controls, https://www.iso.org/obp/ui/#iso:std:iso-iec:27002:ed-2:v1:en [accessed November 22, 2018]
[20] IEC 80001-1:2010 Application of risk management for IT-networks incorporating medical devices -- Part 1: Roles, responsibilities and activities, https://www.iso.org/obp/ui/#iso:std:iec:80001:-1:ed-1:v1:en [accessed November 22, 2018]
[21] IEC 80001-2-2:2012 Application of risk management for IT-networks incorporating medical devices -- Part 2-2: Guidance for the communication of medical device security needs, risks and controls, https://www.iso.org/obp/ui/#iso:std:iec:tr:80001:-2-2:ed-1:v1:en [accessed November 22, 2018]
[22] IEC 80001-2-8:2016 Application of risk management for IT-networks incorporating medical devices -- Part 2-8: Application guidance-- Guidance on standards for establishing the security capabilities identified in IEC 80001-2-2, https://www.iso.org/obp/ui/#iso:std:iec:tr:80001:-2-8:ed-1:v1:en [accessed November 22, 2018]
[23] IEC 62443-4-1:2018 Security for industrial automation and control systems - Part 4-1: Secure product development lifecycle requirements, https://webstore.iec.ch/publication/33615 [accessed November 22, 2018]
[24] Microsoft Corporation, What is the Security Development Lifecycle? https://www.microsoft.com/en-us/sdl/default.aspx [accessed November 22, 2018]
[25] ISO/TR 17791:2013 Health informatics -- Guidance on standards for enabling safety in health software, https://www.iso.org/obp/ui/#iso:std:iso:tr:17791:ed-1:v1:en [accessed November 22, 2018]
[26] IEC 62304:2006/Amd.1:2015 Medical device software -- Software life cycle processes AMENDMENT 1, https://www.iso.org/obp/ui/#iso:std:iec:62304:ed-1:v1:amd:1:v2:en,fr [accessed November 22, 2018]
[27] IEC TR 80002-1:2009 Medical device software - Part 1: Guidance on the application of ISO 14971 to medical device software, https://webstore.iec.ch/publication/7488, [accessed November 22, 2018.
[28] IEC/TR 80002-3:2014 Medical device software - Part 3: Process reference model of medical device software life cycle processes (IEC 62304), https://webstore.iec.ch/publication/7489 [accessed November 22, 2018]

Address for Correspondence

Alpo Värri, Tampere University, Korkeakoulunkatu 3, FI-33720 Tampere, Finland, Alpo.Varri@tuni.fi.