Methods for preclinical validation of software as a medical device

Methods for Preclinical Validation of Software as a Medical Device

Alice Ravizza[1], Federico Sternini[2][a], Alice Giannini[3] and Filippo Molinari[4][b]
[1]*USE-ME-D srl, I3P Politecnico di Torino, Torino, Italy*
[2]*Politecnico di Torino, Torino, Italy*
[3]*Stefanelli & Stefanelli Law Firm, Bologna, Italy*
[4]*Biolab, Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy*

Keywords: Software as a Medical Device, Preclinical Validation.

Abstract: Software as a medical device is subject to dedicated regulatory requirements before it can be used on human beings. The certification process in Europe requires that sufficient data on clinical benefits are available before the device is CE marked. This position paper describes our proposal of a risk-based approach to technical and preclinical validation of software as medical devices. This approach ensures that all technical solutions for safety are implemented in the software and that all information for safe use is consistent before the software can be made available to patients. This approach is compliant to the main international standards ISO 13485 on quality systems and ISO 14971 on risk management and therefore ensures regulatory compliance as well as patient protection. This integrated approach allows the designers of the software to integrate regulatory and safety testing in the technical testing of the candidate release version of the device. This approach ensures patient safety and regulatory compliance at the same time as technical functionality.

1 INTRODUCTION

Currently, among all software solutions related to health, only few can be considered as medical devices, at least from a regulatory point of view. For a software to be considered as a medical device it shall be specifically intended to perform a medical action: vital parameters monitoring for diagnosis; provision of information for subsequent diagnosis or therapy; therapy of a disease; alleviation of a disfunction; control of conception. Specifically, the medical device definition from World Health Organization (Executive Board, 2019) states that it shall be "used in the prevention, diagnosis or treatment of illness or disease, or for detecting, measuring, restoring, correcting or modifying the structure or function of the body for some health purpose"; this definition is also in the European Regulation 2017/745 (European Parliament & European Council, 2017).

This definition rules out all software used for wellness, self monitoring and self enhancement such as apps for the monitoring of body weight of healthy subjects, wearable sensors for the monitoring of heart beat during sport activities, software for guided meditation. So, not all software about the human body is actually a medical device: this guideline applies only to those software that fall into the definition of medical devices.

Medical software is specifically designed to provide a measurable clinical benefit to an individual patient. Being the "zero-risk" condition impossible, regulations prescribes that medical software shall not only provide a measurable and evidence based clinical benefit, but also that this expected benefit shall overweight any risk posed by the software to the patient, the software users and the general environment. We present this position paper about the preclinical validation of software as a medical device, the first step in a long process that leads from ideation to certification of a medical software.

[a] https://orcid.org/0000-0002-5510-2296
[b] https://orcid.org/0000-0003-1150-2244

Ravizza, A., Sternini, F., Giannini, A. and Molinari, F.
Methods for Preclinical Validation of Software as a Medical Device.
DOI: 10.5220/0009155406480655
In *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020)*, pages 648-655
ISBN: 978-989-758-398-8

2 EUROPEAN REGULATORY LANDSCAPE

In Europe, the core regulatory requirements on medical devices can be summarised as “the risk-benefit profile of the device is favourable to the patient” and “all risks have been adequately minimised”. Specifically, new requirements on medical software are laid out in the Medical device Regulations 2017/745 (European Parliament & European Council, 2017) for medical devices and 2017/746 (European Parliament & European Council, 2017) for in vitro medical devices.

Those regulations set very detailed requirements that ensure that device, before being provided to patients, has shown an adequate level of safety and benefit. Moreover, the software life cycle plan shall ensure an adequate level of quality over time. In this position paper, we focus particularly on safety aspects that should be ensured before the device is rendered available to patients.

We have given special attention to the use of international standards, and have given preference to European harmonized standards: this approach ensures that the methods described in this paper are adequate not only for design of medical software but also match the regulatory requirements and allow for a faster certification path in Europe and in countries non-members of the European Union that implemented or harmonized European Regulations and Directives (De Maria et al., 2018). Additionally, the use of European harmonized standards allows presumption of compliance to the European Regulations.

In terms of medical device design, the core requirements are set in the standard EN ISO 13485 (International Organization for Standardization, 2016). Software developers shall list the design inputs to describe the user needs, the expected functionality and performance and to define the main risks that should be overcome to ensure patient safety. The risk identification and reduction is managed by the application, during the input phase and also during development and testing, of the standard EN ISO 14971 (European Committee for Standardization, & International Organization for Standardization, 2012). These general standards shall be applied together with the specific standard on software life cycle management, IEC 62304 (International Electrotechnical Commission & International Organization for Standardization, 2015) that sets requirements not only for the input list, but also on methods and minimum contents of the software testing. A core requirement is that all the risk minimisation measures shall be included in the software testing plan.

Another core standard that should be taken into account is IEC 62366 (International Electrotechnical Commission, 2015) that sets methods to design and test the device for usability.

At the present time, to our knowledge, the state of the art in approaching to regulatory requirements for the European market is a “vertical approach” method. In fact, the industry standard in the management of the software life cycle is typically to approach the compliance to each international standard as a separate task, that is carried out by a different development or testing team. The Quality Assurance testing is performed according to a test plan that is based on software functionality and is not based on the assessment of the potential impact of software malfunctioning on patient health. Additionally, usability is taken care of by a dedicated, specialised team that is not involved in the setup of the functional testing. Lastly, privacy requirements are forcibly added in the software functionalities by another, separate team. A comprehensive test plan, with full coverage of all regulatory requirements, is rarely available. This is reflected in a shortage of information for Competent Authorities when they assess the compliance to European regulations. Additionally, this lack of comprehensive planning impacts those developers that have to take care of the software changes over time, as they design small sets of testing especially for the change under evaluation, while losing control of the main medical features of the device.

3 SOFTWARE LIFE CYCLE MANAGEMENT

3.1 Input Requirements

According to ISO 13485 (International Organization for Standardization, 2016) the first design step should be a clear list of requirements for the developers. The standard cites amongst others "functional, performance, safety, regulatory requirements" and clearly requires to include usability requirements and the definition of the main risks.

Every medical software is different, but we propose to define at the early stages of the development at least the following core inputs, that are presented in an order that reflects their impact on the development:

- Clear expected clinical benefit: what target patient population the software is meant for, what pathology or condition it is addressed to and what are the expected benefits on those patients and those health conditions. The clearer this definition is, the more focused the design and testing activities
- Expected users: the use by professionals, laypersons, special needs persons or different age groups shapes all aspects of design and leads the risk analysis
- Means to measure the clinical benefits: once the clinical benefits are clearly expressed, they shall be measured in a reliable and repeatable way. This means defining which data are collected, the frequency and quality of this collection, and how the data are analysed to detect variations that measure the actual improvements on the patient status
- Data storage: data should be safely stored whatever is the support of the storage. Consider a catastrophe recovery system and a backup database. In case of a centralised database, specify proper countermeasures to minimise errors in data synchronisation.
- Associated software and hardware: Changes in the Software Of Unknown Provenance (SOUP) should be controlled with appropriate policies that plan consequent changes. We propose to classify SOUPs in three risk levels: "red" SOUPs are SOUPs whose malfunctions can expose patients to risk, "yellow" SOUPs are SOUPs that affects the use of the device, "green" SOUPs are all other SOUPs. After the SOUP classification, we propose a reaction policy to SOUP changes.

We have a special policy regarding software as a medical device as medical apps on smart devices. The Operating system is under all points of view a "red" SOUP, because it sustains the Software architecture. A malfunction may interfere with the software proper use or availability and can lower the patient state of health, if the patient management is primarily based on the app itself. iOS and Android have their own versioning and the medical software shall follow the versioning; this means that the developers shall support the latest available version to iOS and Android. It should be noted that Android and iOS provide notifications in advance for major changes.

At each change, the development team shall evaluate the change and define:

- All the required changes
- Timeline for changes
- Verification testing

Table 1: Reaction policy to SOUP changes.

	RED	YELLOW	GREEN
The SOUP complies to a LEGAL REQUIREMENT	Update is top urgent and requires to repeat the technical validation	Update is top urgent and requires repeat the usability testing	Update is top urgent but re-validation may not be required
The SOUP change affects the user experience	Update is top urgent and requires to repeat the technical validation, the assessment of the risk minimization measures, and requires to repeat the usability validation	Update is top urgent and requires to repeat the usability testing	Update can be planned and re validation may not be required

For each software update, the design team shall assess the impact. Impact can be classified as a major change (update that modifies the risk profile of the device), minor change (update that changes the user experience without modifying risk or usability profile of the device,) and bugfix (any other update). While major changes require a new clinical and safety validation, minor requires only new safety validation. Also, the quality of the data transfer between modules shall be proved, including the accuracy and reliability of the measurements originating from sensors (Ravizza, De Maria, et al., 2019).

To propose this approach, our method includes the application of brain storming techniques and FMEA techniques to risk analysis, as the very first step of planning of the validation testing. In fact, the risk control measures that can be implemented at design stage (safe-by-design risk control) shall be not only implemented but also verified and therefore shall be part of the test protocol.

3.2 Identification of Main Risks and of Risk Control Measures

The standard ISO 14971 (European Committee for Standardization & International Organization for Standardization, 2012) on risk management foresees that all possible risks posed to the patient safety and also to data security are identified; the standard suggests structured methods such as Failure Mode and Effects Analysis (FMEA). Subsequently, developers are requested by the standard to implement all those control measures that are technically available, in order to lower and minimise the risks. The preferred order to the risk minimisation measures, as stated in the European Regulation, is to give priority to a safe-by- design approach. If no safe-by-design solutions are available, then developers should include adequate protections and alarms; additionally, all the required information for safe use shall be provided.

Again, every medical software is different, but we have identified some core issues that may be applicable to a vast majority of medical device software; for each we also have identified a possible risk minimization measure. Hereby we list the risks in an order that reflects the potential impact on patient safety.

Table 2: Examples of core issues and their risk control measures.

Risk	Risk control measures
Risk of delayed treatment	Streamlined use flow chart
Risk of alarm overload	Alarms should be differentiated in priority levels according to the associated risk.
Risk of improper treatment	Alarms should be triggered for any use that is inconsistent with previous data or clinical guidelines
Risk of patient incomplete or wrong profiling	Guided data insertion procedure, patient ID always visible, minimal screen cluttering
Risk of improper professional use	Clear definition of profile privileges and procedure for password management
Risk of data loss	Backup, activity log of all user interactions
Risk of data breach	Basic cybersecurity measures
Risk of patient privacy break	Data pseudonymization, strict control on profiles with data access privilege

3.3 Specific Activities for Privacy Requirements

In Europe, the privacy rights of citizens are defined and protected by the Regulation 2016/679, so called GDPR (European Parliament & European Council, 2016).

The main rights include:

- Article 15 GDPR – Right of access: function to export data upon request. Data should be exported in a common format (according to art. 20 GDPR – Right to data portability);
- Article 16 GDPR – Right to rectification: function to update data upon request with a record of the update;
- Article 17 GDPR – Right to erasure: design function that perform record deletions of all data points connected to one subject quickly and efficiently (deleting the record shouldn't affect the integrity of the whole database) with a report of deletion
- Article 18 GDPR – Right to restriction of processing.
- Art. 22 GDPR- Automated individual decision-making, including profiling: functions which allow to turn on or off certain processing parameters

As medical devices collect a great entity of personal data, it is necessary to deal with the privacy aspects of this data collecting and processing. In literature, there are applicable position papers that provide an in depth analysis of how to apply concepts of the GDPR to Artificial Intelligence for specific medical purposes (Pesapane, Volonté, Codari, & Sardanelli, 2018). Minimum measures required to be GDPR compliant might include:

- Conduction of a Data Protection Impact Assessment before the medical software is in use;
- Design the software to collect and store only data which is necessary for the core activity of the medical software;
- Regularly pseudonymize data for back-end processing.
- Restrict access and privileges of developers to sensitive data if not strictly necessary;
- Design software functions which allow for data subjects to correctly exercise their rights.
- Conduct periodical tests of effectiveness of data security measures of the software;

Pseudonymization represents, from a privacy standpoint, both a type of data processing and a measure to ensure data security. As such, it lowers the risks connected to the processing of particular sets of data such as those related to health.

Moreover, according to the GDPR, pseudonymized data are to be considered as personal data. On the other hand, anonymization consists of a processing which results into an irreversible de-identification of the data subjects. Data subjects need to be informed, in a transparent way, that their clinical data might be anonymized and used for further research. Thus, the anonymization of patients' data represents a secondary use which does not require the consent of the patient. The GDPR is not applicable to fully anonymized data but, differently from other privacy standards, it does not include a set of variables that when removed from a dataset render the data anonymous, making this specific type of processing riskier.

3.4 Output of the Design: Software Functionalities

Designers should be able to define, for each end user, a set of privileges and a list of functionalities that the software provides.

Such functionalities are those software actions that, once used on the patient or for the patient, will provide the clinical performance and therefore ensure the patient clinical benefit.

For example, a medical device software for medical adherence may have different functionalities: plan a therapy, alert the patient when the therapy should be taken, measure the amount of drug that was taken, monitor the adherence in terms of timing and dosing, coach the patient to improve adherence, provide suggestions to manage symptoms or collateral effects and so on. In this case, not all functionalities will be available to the patient directly: for example, only a user with a "physician" profile may be able to access to the therapy planning function.

On the other hand, a software for cognitive or behavioural therapy will have completely different functionalities such as provision of images, text, or sounds at specific times of the day or in reaction to patient inputs.

It should be noted that software modules can be discriminated according to the functionality for the user, where each of these features is related to a module. Some of these modules have a medical purpose, others do not.

Some features without a medical purpose are widely present in medical software, for example: collect and retain the patient's administrative details or archive the patient's medical history

Some modules may have ancillary functions, for example: audit trail, access and security, cryptography, archiving and backup of personal data.

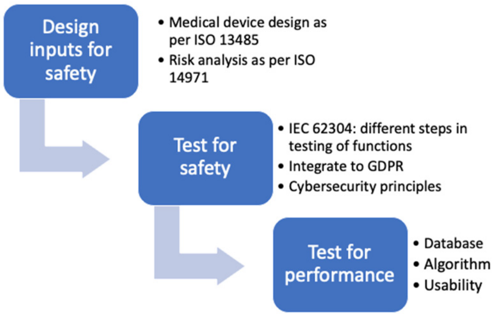
Design inputs for safety
• Medical device design as per ISO 13485
• Risk analysis as per ISO 14971
Test for safety
• IEC 62304: different steps in testing of functions
• Integrate to GDPR
• Cybersecurity principles
Test for performance
• Database
• Algorithm
• Usability

Figure 1: The risk-based approach.

4 SOFTWARE TECHNICAL VERIFICATION

4.1 Technical Verification of Functionalities and Risk Control Measures

After the development of the Minimum Viable Product of the Medical Device, its safety and clinical benefit shall be tested. The testing should follow a fixed plan built on the base of the input requirements drafted at the beginning of the design stage. This plan shall not only declare the required input, but also define, for each input, the testing method and the expected result. The tests are passed if the result is consistent with the expected pre-declared deliverable and if it does not introduce any additional risk to the patient. While some requirements may need actual functional tests on the device, others can be satisfied with documentation and procedures. Technical requirements that need functional tests may include the correct alarm activation, which requires to prove that it is activated when needed, and that does not activate when it is not needed. On the other hand, other requirements are not strictly related to the software development, such as the password policy and the personnel training.

For risk control measures that are information for safe use, the verification is in two steps. First verify the instructions for safe use is in place, then verify that it is clear (during the usability test).

The outcome of the test defines the needs for additional development. If all tests are passed, the product is ready to be tested for usability and clinical benefit. Otherwise, if some tests failed, but they do not affect risk for the patient or do not modify the usability profile of the device, the manufacturer can still test the device for usability but should refrain to use it for clinical benefit until all the technical functions have been confirmed.

4.1.1 Usability Verification

Usability verification aims to confirming the safety of the user interface and also of the provided information, that together enable effective use and protect against potentially harmful use errors. The reference standard is IEC 62366 (International Electrotechnical Commission, 2015) that proposes various methods for usability verification. We proposed in the past the choice of the most adequate methods according to the device kind and the step in the device design process (Ravizza, Lantada, Sánchez, Sternini, & Bignardi, 2019).

For validation activities, we propose to define a structured approach to user testing in real or simulated use conditions, with real users. The preparation of such user tests requires that the usability experts define, together with the designers, a list of expected use scenarios, that shall be chosen in order to simulate the most common use and also any hazardous situations in the expected use experience. We usually prefer to describe the use experience by a flow chart of each use scenario and then to provide detailed description by a task list analysis. For detailed risk management, this task list may be used as the entrance information to apply Failure Mode and Effect Analysis techniques. We also propose to involve real users in the usability evaluation activities, by applications of the use scenarios in a real or, most probably, simulated clinical setting and by observation of user errors and any safety-critical errors.

Analysis of user errors and of critical errors as observed leads to the approval of the user interface or to the identification of non-acceptable use risks, that would lead to device interface re-design (Ravizza, Lantada, et al., 2019; Zhang, Johnson, Patel, Paige, & Kubose, 2003).

4.2 Aim of the Verification Activities: Compliance with the Medical Device Regulation and to GDPR

In Medical Device Regulation (MDR), Annex I section 17 declares that medical device software shall *"be designed to ensure repeatability, reliability and performance in line with their intended use"*.

For compliance with the essential requirement 17 of the MDR, the manufacturer is required to apply management principles to the entire software life cycle: these principles are listed in the EN 62304 (International Electrotechnical Commission & International Organization for Standardization, 2015).

Another core Essential Requirement, is the point 5 of Annex I, that requires that risk minimisation measures include use error minimisation principles and usability principles. The manufacturer may apply the methods described in the standard IEC 62366; while this standard is not harmonized, it is considered as state-of-the- art and is cited as a means of compliance in both ISO 13485:2016 and IEC 62304 in its current version (indirectly cited, by citation of IEC 60601-1-6) and in its proposed draft (directly cited) (International Electrotechnical Commission & International Organization for Standardization, 2015; International Organization for Standardization, 2016).

The new cybersecurity requirements of Regulation (EU) 2019/881 (European Parliament & European Council, 2019) referred to below also apply to software:

- Consider the risks related to the possible negative interaction between the software and its IT environment (Annex I, Chapter II, Section 14.2 (d)).
- Develop the software taking into account information security (Section 17.2).
- Establish minimum requirements relating to hardware, IT network features and IT security measures, including protection against unauthorized access (Section 17.4)
- Include these instructions in the instructions for use (Annex I, Chapter III, Section 23.4 (ab)).

Furthermore, the principles of art. 24 of the GDPR for Software as Medical Device. Since software treats data: that is to say that they must be designed according to the principles of privacy by design and by default pursuant to art. 24 of the GDPR.

Very briefly, the software must comply with the principles set out in art. 5 (in particular the principles of minimization, accuracy, security, integrity and confidentiality of the data). It must also allow the user (as Data Controller or Data Processor) to respect the general principle of accountability.

4.3 Proposed Test Methods

For regulatory purposes, device validation shall give proof that the device is adequate for its intended purpose and to confirm its estimated risk benefit profile. No major modifications are expected after the device validation, in terms of design characteristics for risk minimisation. On the other hand, improvement suggestions can be collected, for the subsequent future iterations.

4.3.1 For Technical Verification

The device design process has defined the input requirements, including expected results of real use. Therefore, to complete the technical validation, a simulation of the real-world condition is proposed. The simulation should include the use of dummies that simulate typical patients/users; these dummies that can be designed to reflect what is expected by real user interactions; we propose, to simulate the real users, to define a known dataset of real user activities and then apply the principle of the mode, as opposed to the principle of the average. The proposed strategy simulates the real users with the most frequent activity, ensuring that the simulation is representative of the majority of the users. We propose to not use the average principle because it could create simulations that are statistically representative of the users but that are not consistent with any user activity. For example, if the interaction is defined as the time reaction to a stimulus, the mode provides real users time reactions, while average may be biased. In the case of high-risk applications, manufacturers should complete additional simulations with additional dummies, built to represents high-risk user/patients or worst-case scenarios.

4.3.2 For Usability

The international standard on usability requires that designers validate those parts of the interface that are meaningful for clinical use, including all the critical ones. For example, all the software interactions with the patients should be included, while some on-boarding activities of the professional users and the administrative or logistic modules may be subject to very light usability assessment.

Since all the critical use scenarios should be tested, we propose that a complete task analysis is available and checked for coherence to the user manual or instruction leaflet. This should be available for all the software intended users.

The usability validation should be performed with real users and in a very well simulated or real use environment and should include a significant number of participants. International guidelines (Center for Devices and Radiological Health, 2016) suggest a minimum of 15 participants per user profile. We suggest, in this phase, to involve as applicable the patient associations, such as for example EUPATI, that may be able to assist not only in the recruitment of the participants but also in raising awareness on the importance of their participation in the whole life cycle of the development of new technologies (Haerry et al., 2018).

During user tests, we also suggest to evaluate, if applicable to the device, the length of time needed for each task (by time-and-motion studies) and the workload of the user; additionally, the readability and understandability of the privacy disclosure documents and the privacy contents can be added as usability endpoints. In order to collect valuable information from the end users, destined to root cause analysis of any encountered user error, we suggest to include a de-briefing interview at the end of the user testing. We typically apply heuristic principles while drafting the interview questions (Zhang et al., 2003).

As for all other validation activities, risk control measures in the user interface such as passwords, pop-up and notifications, should be formally reviewed for final implementation and effectiveness and the risk-benefit profile confirmed.

5 CONCLUSIONS

An integrated risk-based approach for software validation allows to plan and execute the validation activities in a complete and efficient manner. The test plan should include requirements from the main medical international standards and should give proof that all the risk minimisation measures have been effectively implemented. Such testing, for technical features, can be based on simulation, thanks to the creation of adequate simulated clinical use conditions. Such conditions should be created by application of the principle of the mode, to define a significant simulated patient and clinical conditions.

At the completion of such activities, real users and patients should be involved to test the usability aspects of the interface. The outcome of the testing is the proof that all technical features of the device show an adequate performance and that all risks have been minimised. The device is therefore compliant to regulatory requirements and can be subject to clinical trials.

Future work for this position paper will include the application of this method to the preclinical testing of different devices and the evaluation of a similar risk-based approach for clinical testing.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the European Patients' Academy (EUPATI), a pan-European project implemented as a public-private partnership by a collaborative multi-stakeholder consortium from the pharmaceutical industry, academia, not-for-profit, and patient organisations. The Academy was started, developed and implemented as a flagship project of the Innovative Medicines Initiative (http://www.imi.europa.eu/), and continues to be led by the European Patients' Forum.

REFERENCES

Center for Devices and Radiological Health. (2016). Applying Human Factors and Usability Engineering to Medical Devices. Retrieved November 21, 2019, from U.S. Food and Drug Administration website: http://www.fda.gov/regulatory-information/search-fda-guidance-documents/applying-human-factors-and-usability-engineering-medical-devices

De Maria, C., Di Pietro, L., Díaz Lantada, A., Madete, J., Makobore, P. N., Mridha, M., … Ahluwalia, A. (2018). Safe innovation: On medical device legislation in Europe and Africa. *Health Policy and Technology*, *7*(2), 156–165. https://doi.org/10.1016/j.hlpt.2018.01.012

European Committee for Standardization, & International Organization for Standardization. (2012). *EN ISO 14971:2012, Medical devices: Application of risk management to medical devices*. London: BSI.

European Parliament, & European Council. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). , Pub. L. No. 32016R0679, 119 OJ L (2016).

European Parliament, & European Council. Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC (Text with EEA relevance.). , Pub. L. No. 32017R0745, 117 OJ L (2017).

European Parliament, & European Council. Regulation (EU) 2017/746 of the European Parliament and of the Council of 5 April 2017 on in vitro diagnostic medical devices and repealing Directive 98/79/EC and Commission Decision 2010/227/EU (Text with EEA relevance.). , Pub. L. No. 32017R0746, 117 OJ L (2017).

European Parliament, & European Council. Regulation (EU) 2019/881 of the European Parliament and of the Council of 17 April 2019 on ENISA (the European Union Agency for Cybersecurity) and on information and communications technology cybersecurity certification and repealing Regulation (EU) No 526/2013 (Cybersecurity Act) (Text with EEA relevance). , Pub. L. No. 32019R0881, 151 OJ L (2019).

Executive Board, 145. (2019). *Standardization of medical devices nomenclature: International classification, coding and nomenclature of medical devices: report by the Director-General*. Retrieved from https://apps.who.int/iris/handle/10665/328220

Haerry, D., Landgraf, C., Warner, K., Hunter, A., Klingmann, I., May, M., & See, W. (2018). EUPATI and Patients in Medicines Research and Development: Guidance for Patient Involvement in Regulatory Processes. *Frontiers in Medicine*, *5*. https://doi.org/10.3389/fmed.2018.00230

International Electrotechnical Commission. (2015). *IEC 62366-1:2015, Medical devices—Part 1: Application of usability engineering to medical devices*. Retrieved from http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/31/63179.html

International Electrotechnical Commission, & International Organization for Standardization. (2015). *IEC 62304:2006+AMD1:2015 CSV, Medical device software: Software life cycle processes* (1st ed.). Geneva: IEC.

International Organization for Standardization. (2016). *ISO 13485:2016, Medical devices—Quality management systems—Requirements for regulatory purposes* (3rd ed.). Geneva: ISO.

Pesapane, F., Volonté, C., Codari, M., & Sardanelli, F. (2018). Artificial intelligence as a medical device in radiology: Ethical and regulatory issues in Europe and the United States. *Insights into Imaging*, *9*(5), 745–753. https://doi.org/10.1007/s13244-018-0645-y

Ravizza, A., De Maria, C., Di Pietro, L., Sternini, F., Audenino, A. L., & Bignardi, C. (2019). Comprehensive Review on Current and Future Regulatory Requirements on Wearable Sensors in Preclinical and Clinical Testing. *Frontiers in Bioengineering and Biotechnology*, *7*. https://doi.org/10.3389/fbioe.2019.00313

Ravizza, A., Lantada, A. D., Sánchez, L. I. B., Sternini, F., & Bignardi, C. (2019). Techniques for Usability Risk Assessment during Medical Device Design. *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies*, 207–214. https://doi.org/10.5220/0007483102070214

Zhang, J., Johnson, T. R., Patel, V. L., Paige, D. L., & Kubose, T. (2003). Using usability heuristics to evaluate patient safety of medical devices. *Journal of Biomedical Informatics*, *36*(1), 23–30. https://doi.org/10.1016/S1532-0464(03)00060-1