***Piotr Wojciechowski, Ph.D.***
*The School of Banking and Management in Krakow*
*piwojcie@wszib.edu.pl*

# VALIDATION OF MEDICAL DEVICE SOFTWARE

## Introduction

Medical technologies are increasingly supported by IT systems. Medial device software may have a direct impact on medical device safety. The legal definition of a medical device[1] indicates directly that software may be qualified as medical device. In such cases, the software becomes an active medical device regardless of the fact whether it controls or influences the application of the device or it is an independent device itself. This means that this type of software is subject to the same principles as all other medical devices. The validation of the software is an important issue and it is a crucial element of the compliance assessment process resulting in CE marking. In such cases, it is irrelevant whether the process concerns software used in a computer system that cooperates with the device or the software which is an integral part of the device incorporated into the circuits. Every software application should be taken into account in risk assessment. According to the standard [2] that concerns risk analysis of medical devices, a breach of data and systems safety may lead to a damage, e.g. through the loss of data, unauthorized access to the data, the damage of data, the loss of information, or the malfunction of the device. Thus, it is extremely important that proper functioning of the software should be taken into consideration in all the processes that involve the development of a medical device and the changes introduced to it, including the software update.

## 1. Software related hazards

The scope of hazards related to the application of software in medical devices is extensive. The PN-EN ISO 14971 standard that was mentioned above deals with the practical aspect of this problem. The identification of hazards to the device under analysis should involve all possible events related to the software of the device. They may differ as regards their character and the levels of significance. The objective of the analysis is to identify and assess the risk

[1] Regulation (EU) 2017/745 of the European Parliament and of the Council on medical devices
[2] PN-EN ISO 14971-2020-05 Medical devices. Application of risk management to medical devices.

level, to decrease it to the lowest possible level[3], and to plan preventive measures in the case of any hazard that can be predicted and that may result in any minor or major damage. It is also necessary to take measures against hazards whose level of risk exceeds the safety level but is impossible to decrease. Such a state can be identified as the *residual risk*. This is acceptable only when potential clinical benefits exceed significantly potential hazards and on the condition that adequate safety measures are applied. In such situations, software may play a significant role in the implementation of a hazard prevention system.

Numerous medical devices have data bases which store not only the results of examinations but also patient personal data. The connection of personal data to clinical data results in the creation of a special type of sensitive data. They have to be subject to thorough protection against unauthorized access. The accepted IT solutions should lead to the increase in safety level, which obviously does not relieve the users from the necessity to apply internal access procedures to the data. The data controller and the entity that processes sensitive data must ensure the prevention of unauthorized access by implementing appropriate technological and organizational measures [4]. Moreover, medical device manufacturers should apply adequate safeguards. Both approaches complement each other.

Although it is not related to the issue discussed above, data damage or loss is of significance to the functioning of medical devices. Medical units should have continuous access to information during treatment processes. Thus, the manufacturer should provide a data archiving system or inform the device users about the options and possible ways of implementing it. Even when data continuity is not important for individual patients, it may be important to medical professionals in their scientific research.

The third group of hazards, which is the malfunction of a medical device, is of particular significance. Software is frequently responsible for the control of important, or even critical parameters such as, for example, the dose and time of exposure to the radiation of X-ray machine. Unauthorized and uncontrolled change of any parameter may result in serious consequences.

In 2001 an event happened that was referred to as the Białystok incident. Five female patients were overexposed to radiation during radiotherapy. According to investigations, the radiation dose was be exceeded as many as several dozen times. The patients developed wounds

[3] The use of the ASAP (As Small as Possible) rule is required.
[4] M. Błażejewski, J. Behr, *Środki prawne ochrony danych osobowych*, UW, Wrocław 2018, p. 115.

that were difficult to heal, which indicates the scale of the event[5]. They had tissue and skin transplantation surgeries[6] and in 2004 the court awarded damages to them. It turned out that during the therapy that was conducted with the Neptune 10P apparatus, the device switched on and off as a result of a failure and it increased the previously intended radiation dose many times. The incident showed the importance of a rigorous control of devices that are potentially dangerous. It also illustrates the significance of the verification of the devices and their software in the elimination of the probability of conditions that are hazardous to the patients and users.

In the case of diagnostic devices, improper results may lead to wrong medical decisions, e.g. the implementation or abandonment of certain therapies. Every such case may pose danger to the health and life of patients. This is particularly important for in vitro diagnostic medical devices[7]. Automated measurement systems that are applied by analytical laboratories can perform thousands of tests every day. Every software error involves a risk of serious consequences, not necessarily in terms of the diagnostic significance but with regard to the scale of such incidents.

## 2. The goal and application scope of the EN 62304 standard

The *Introduction* to the EN 62304 standard justifies the application of the standard which results from the necessity to provide evidence that the software in a medical device will not cause any unacceptable risks. The Polish Standardization Committee informs about the application scope regarding the use of the standard in the development and maintenance of medical devices[8], which also results from chapter 1.2 of the standard. One should remember that these activities do not cover final validation of the medical device. In the development process, the validation of the medical device is independent of the software validation as software validation concerns only a component of the medical device. This is the objective of the EN 62304 standard. In Annex B (informative) to the standard, the need to develop high quality, safe software is emphasized

The EN 62304 standard distinguishes different risk classes and that is why one of the three safety classes (A, B or C) should be assigned to the software in line with the principles given in Clause 4.3:

[5] https://www.mp.pl/kurier/7741,komisja-ekspertow-badala-w-bialostockim-osrodku-onkologicznym-sprawe-nadmiernego-napromieniowania-chorych (accessed: 17.08.2021).
[6] https://bialystok.wyborcza.pl/bialystok/7,35241,1233070.html (accessed: 17.08.2021).
[7] Regulation (EU) 2017/746 of the European Parliament and of the Council on *in vitro* medical devices
[8] https://sklep.pkn.pl/pn-en-62304-2010p.html (accessed 26.08.2021).

- Class A: No injury or damage to health is possible.
- Class B: Non serious injury is possible.
- Class C: Death or serious injury is possible.

A correct classification is crucial for further steps. The more serious potential hazards, the more necessary it is to meet an increased number of requirements presented in the standard. In the process, the highest risk should be considered that may be involved with the use of a particular medical device. Clauses 5-9 describe the requirements in detail and indicate the class to which a particular requirement should be assigned. Table A.1. on page 35[9] of the standard is a major help. One can see there that comparatively few requirements are assigned to Class A software, some more to Class B and all of them to Class C. It should be remembered that the software classification is independent of the classification of complete medical devices which are assigned to one of the four classes: I, IIa, IIb, III.

The standard allows for the decomposition of the software, i.e. breaking it up into component parts in order to assign different classifications to different software parts (4.3 of the standard). This approach may be justified not only in the case of devices where the software controls, for example, radiation doses, which involves a high-risk level but also in entering and storing patient data, which obviously does not generate any clinical hazards but involves a risk of data loss or leakage.

It is also possible to lower software safety class when other technological measures of risk reduction are applied in the medical device thanks to which a software failure does not even cause a hazard to health and life of the patient or staff. For example, the software may issue a command to "replenish the pressure in the tank". However, if there is a technological safeguard that limits the maximum pressure to the value that does not exceed the maximum level acceptable in a given device and if safety valves are applied, a software malfunction will not increase the risk which will still be within the controlled range

It is stated in Clause 2. of the EN 62304 standard that its application must also involve the reference to the ISO 14971 standard on risk management medical device. The ISO 14971 draws attention of the manufacturers of the medical device and its software to risk elements that should be considered in the compliance assessment process. This standard takes into account the identification of hazards from the beginning of the medical device development process so potential hazards should be described before the software validation process. As

[9] Page number for the paper version. For a pdf version it is page 39.

software life cycle processes are closely connected with the identified and documented hazards, the application of both standards is absolutely justified. Subclause 5.4.2 of the standard does not only require a risk reevaluation after defining all software requirements but it also demands a verification of these requirements after risk control methods of the medical device are determined (subclause 5.2.6 of the standard), which even better emphasized the relationship between the two standards.

## 3. Basic definitions and their application

The practice of using the EN 62304 standard by design teams illustrates certain problems concerning the understanding of some terms used in the standard. Firstly, the significance of the term *documented* should be emphasized. The term means that all requirements recorded in this way should be reflected in the records. The following definitions in the standard should be discussed.

**Change request (3.4)**

A *documented specification change to be made to a software product* is referred to as a change request. It is obvious that some errors may be revealed in the software. This may happen not necessarily immediately before launching the medical device. Sometimes, there is an idea of a simple modification such as the change of the background color or the type or size of fonts to make the text clearer. The users note the possibility to implement some simplifications or they ask for the extension of some functionalities. This must be documented regardless of the fact whether a minor or a more significant change has been introduced. The problem with many developers is that they are reluctant to keep records. This, however, is obligatory in the case of medical devices. Subclause 5.1.8. shows that the software must have its documentation. Entering the information about changes involves the necessity of approval (Subclause 8.2.1. of the standard). This procedure is closely related to the software maintenance process which is described in Clause 6 and is applicable in all software classes – although not in all subclauses of the clause. Some changes may require a verification of the identified hazards in the documentation of medical device risk assessment. Chapter 8.2.4 of the EN 62304 standard includes the requirement the traceability of change. Traceability is the capability to differentiate various versions (configuration items, according to clause 3.34 of the standard) and to link them to particular medical devices, which in terms of traceability applied in quality management systems means the so-called backward traceability. It means in practice that it should be possible to determine where the modified software is installed. This is particularly important

when a problem arises that was not identified during the verification of the modified software version. The necessity is obvious of a full control over the versions of the software and the possibility to link them to the devices where they were installed. Detailed guidelines concerning software release are given in Annex B 5.8. to the EN 62304 standard.

**Problem report (3.13)**

The request for change described above may result from problem identification. The EN 62304 standard indicates that this may concern either actual or potential problems with a device that is believed to be inappropriate for the intended use or even unsafe. This is the reason why all reported issues must be included in the records. It should also be kept in mind that for medical devices steps are provided to identify incidents and serious medical incidents. The latter ones require official reporting. Obviously, not very problem reported by the user is justified. The user may misuse the device, which requires the manufacturer's explanation. However, this does not lift the obligation to keep records. For every software class there is the necessity to report problems, analyze them, inform the interested parties and analyze the trends.

**Regression testing (3.15)**

According to the normative definition, the term refers to *testing required to determine that a change to a system component has not adversely affected functionality, reliability or performance and has not introduced additional defects.* This type of testing is frequently considered by developers and IT specialists as unnecessary and a waste of time. On the other hand, there are user opinions that an elements that previously worked properly does not work after the software update. This is why the EN 62304 standard includes the requirement for running and recording such tests. Anomalies in medical devices may lead to serious hazards. Thus, it is not acceptable to state automatically that the change is insignificant and it certainly will not affect other functions. As this type of tests is related to safety issues, the above requirement refers only to Class B and C software. It is unnecessary in the case of Class A software.

**SOUP (software of unknown provenance) (3.29)**

The term refers to software items *that are already developed and generally available, (also known as "off-the-shelf software") and software previously developed for which adequate records of the development process are not available*. The standard describes issues concerning the integration of such software with the medical device, its compatibility, errors and the related unexpected risks. Such software should not be automatically assumed to be correct solely on

the basis of previous experience. On the contrary, it requires special control in the processes of integration and subsequent validation.

## 4. Testing medical device software

The objective of testing should be to find that the software is error-free and the medical device will not perform activities that might generate any hazards. As programming is a complex process and source codes are frequently elaborate, testing should cover all stages of software development, including the creation of program units, their integration and the final testing of the completed program. In order to test integrated software (of Class B and C), the EN 62304 standard recommends testing in non typical conditions with the consideration of improper use (clause 5.6.4. of the standard)

Predicting improper use is a difficult task. It may be helpful to use the experience gained from previous similar devices. However, there is usually no reference for new technologies or manufacturers. Every misuse prediction should be recorded so that a set of ideas can be used in all next designs. If in a particular case an EN 62366 standard-based usability report has been developed, it is possible to use the information stored there. Three types of scenarios of the medical device use[10] are developed within usability engineering. One of them is the misuse scenario, which may be an inspiration source in the identification of similar behavior of software users.

The testing of medical device software requires a close cooperation with its developers. In order to check practically program's output responses to input stimuli it may be necessary to prepare particular modules of medical devices that will enable a direct observation of their behavior. For example, switching of individual solenoid valves to provide a gas is not practical in the developer's work as they may be substituted by LED diodes which will signal clearly and safely that particular functions are triggered. There may be single modules of a given device, their functional connections or almost ready-made devices. One should remember that as the device being tested is still in its design phase, the developers become a part of the design team. A question arises sometimes whether it is not better to prepare a complete device together with its software and then to submit it to external tests conducted by medical specialists. Unfortunately, in this case the approach of beta testing cannot be applied. A medical device in its prototype phase cannot be launched on the market and put into service. This is possible only

[10] PN-EN 62366-1:2015-07E Medical devices. Part 1: Application of usability engineering to medical devices.

after all formal and legal requirements are fulfilled and the device is registered. Thus, testing the device on patients is not possible. When there is a need for a medical specialist to cooperate with the medical device design team, an appropriate agreement must be signed. Every activity of the specialist should be supervised by the manufacturer.

## Conclusions

Correct implementation of the rules of supervisory control of medical device software requires a reliable approach to the established rules of conduct. Adequate measures should be taken in organizations to ensure proper communication and cooperation between departments. Employees responsible for the development of the software should be given opportunities of training about general requirements regarding compliance assessment of medical devices. Good results are achieved through regular meetings during which information is exchanged among all parties involved.

It is not without reason that the EN 62304 standard emphasizes the need to develop a software development plan. The plan is an input document to which all subsequent activities can be referred. Self-discipline, an ongoing documentation of the whole work, problems, changes, test results and reviews make it possible to maintain the systematics of work, which results in the development of the actual quality of a safe medical device.

IT systems will be used more extensively in medical devices. This field of knowledge is developing quickly. Medical diagnostics is increasingly more automated and the robotization of medicine is a fact. Considering this development trend, the supervision of software is more and more critical. Further standardization work in this area and the development of requirements for the changing technological conditions can be expected.

## Bibliography

[1] Błażejewski M., Behr J., *Środki prawne ochrony danych osobowych*, UW, Wrocław 2018.
[2] PN-EN ISO 14971-2020-05 Medical devices. Application of risk management to medical devices.
[3] PN-EN 62366-1:2015-07E Medical devices. Part 1: Application of usability engineering to medical devices.
[4] Regulation (EU) 2017/745 of the European Parliament and of the Council on medical devices.
[5] Regulation (EU) 2017/746 of the European Parliament and of the Council on *in vitro* medical devices.

[6] https://bialystok.wyborcza.pl/bialystok/7,35241,1233070.html.
[7] https://www.mp.pl/kurier/7741,komisja-ekspertow-badala-w-bialostockim-osrodku-onkologicznym-sprawe-nadmiernego-napromieniowania-chorych.
[8] https://sklep.pkn.pl/pn-en-62304-2010p.html.

***Abstract***

Validation of the medical device software is obligatory in the compliance assessment of the device. Software that is dedicated to medical devices may have a crucial impact on patient and medical staff safety. A harmonized EN 62304 standard is dedicated to the process. It is important that the team of developers should cooperate with medical device designers, understand the essence of the validation process and properly document all the activities.

***Key words***

Software, medical, validation, testing.