FDA, medical software, recall,
safety of medical devices.

Leszek DREWNIOK[1], Ewelina PIEKAR[1],
Mirosław STASIAK[1], Remigiusz MANIURA[1]

ANALYSIS OF REASONS CAUSING DEVELOPMENT OF MALFUNCTIONING SOFTWARE IN MEDICAL EQUIPMENT

In the world today, the vast majority of medical electronic equipment contains software. Very often even the computer software is classified as an independent medical part. Because of the ease of making changes to the element of a large functional complexity, there is a high risk of introducing errors in the modified software. For example, just entering the wrong one filter parameter can make the biomedical signal processing circuit work incorrectly. As noted in [5], "the lessons learned from … disasters can do more to advance engineering knowledge than all the successful machines and structures in the world". This statement is also true in the software domain. The main goal of this paper is – basing on a database of medical devices with software defects – to draw conclusions and guidance for the design and maintenance of software for new medical equipment.

1. INTRODUCTION

When a medical device is defective, or when it may exists a risk to health, or when it is both defective and arises a risk to health, responsible object (manufacturer, distributor, or other responsible party) has to take an action to address a problem**.** In United States this action is called „recall” [2]. A recall does not always mean that the product must be withdrawn from using or returned to a company. A recall sometimes means that the medical device needs to be checked, adjusted, or fixed. If an implanted device is recalled, it does not always have to be removed. In this situation discussed is the risk of removing the device compared to the risk of leaving it in place. To be on the safe side, a company can recall an entire lot, model, or product line especially when the problem concerns a group of products, but it cannot predict which individual devices may be affected. In most cases, a company recalls a medical device on its own (voluntarily, recalls the device through correction or removal and notifies U.S. Food and Drug Administration (FDA)). Legally, FDA can demand to recall a device. This could happen if a company refused to recall a device that was associated with significant health problems or death. However, in practice, FDA has rarely needed to require a medical device recall [2].

Nowadays, most electronic systems – from straight lines of sensors to complex body scanners – contain software. Many requirements, such as short time to market, sophisticated functionality, speed of processing, upgradeability, reduced costs; influence on quality of development and maintenance of the medical software.

Some of the tools that prevent development of software with failures are international regulations and standards (established by different documents in various groups of countries but all containing very similar requirements). Direct requirements implementation in practice is very difficult, therefore each firm that produce medical devices with software, establishes own internal procedures and instructions that help fulfil appropriate regulations.

In all related standards, the main headwords are software validation and verification. This activity should be conducted throughout the entire software life cycle [1]. Software verification provides objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase. The main activities of verification are: testing, various static and dynamic analyses, code and document inspections, walkthroughs, and other techniques. Software validation means „confirmation by examination and provision of objective evidence that software

[1] Department of electrostimulation techniques, Institute of Medical Technology and Equipment, Zabrze, Poland.

specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled” [1].

One of the FDA’s analysis of 3140 medical device recalls conducted between 1992 and 1998 says that 242 of them (7.7%) were related to software failures. Of those software related recalls, 192 (79%) were caused by software defects that had been introduced with changes of the software made after its initial production and distribution [1]. Results of this analysis should be for manufacturers of medical devices the very important lesson. Why the software maintenance process is so difficult? Please notice that fast changes in software, without suitable analysis of all aspects and influences can lead to release device with errors.

2. METHOD

2.1. INTRODUCTION

The FDA recalls database has been selected for considerations because it’s the wide source of information about very instructive cases. To protect the privacy of the manufacturers, this paper does not contain any specific information about the manufacturer or the product name. The main purpose is to understand the types of software problems and to abstract generic guidance about preventing and detecting the software faults before systems are released. The study has analyzed 12 917 recall cases from the years 2006 to 2011 basing on FDA database [2]. The FDA recall data consists of the recall number and class, recalling manufacturer and product name and reason for recall [7]. Detailed analysis refers software related recalls.

2.2. SOFTWARE RELATED RECALLS ANALYSIS

As previously mentioned, the study has analyzed 12 917 recall cases, of which 924 are related to faulty operating software. This means that software related recalls are 7.2% of all recalls. Number of software related recalls referenced to all recalls for each year from period 2006-2011 are shown in Tab. 1.

Table 1. No. of software related recalls referenced to no. of all recalls.

Year	No. of recalls	No. of software related recalls	%
2006	1473	177	12.0
2007	1275	158	12.4
2008	2318	136	5.9
2009	2357	109	4.6
2010	2753	166	6.0
2011	2741	178	6.5
Total	**12917**	**924**	**7.2**

It may be noted that since 2008, the percentage software related recalls have remained at a similar level. This is a very interesting information, but only limiting to this statement, without taking into account other parameters (market, economic, quantities of produced equipment, etc.) it is difficult to draw the right conclusions. One can only suppose – assuming a significant increase in the complexity of medical devices – that implemented software management procedures play an increasingly positive role. The percentage of software related recalls referenced to all recalls are shown on Fig. 1.

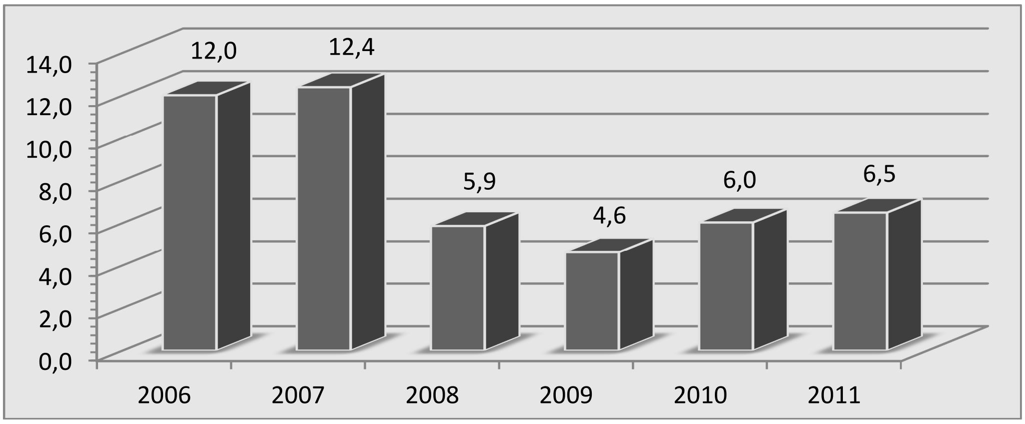

Fig. 1. The percentage of software related recalls referenced to all recalls.

2.3. CATEGORIES OF SOFTWARE RELATED RECALLS

During study the reasons for recall ten main symptoms have been defined. Presented below classification is based only on the descriptions named reasons for recall so the actual cause may be a little different than indicated by the symptom. The division into categories allows to analyse and draw conclusions, in which of them is most likely to make a software mistake.

Number of all software related recalls divided into classes according to the symptoms are shown in Tab. 2 and the percentage representation of them are shown in Fig.2.

Table 2. No. of all software related recalls divided into classes according to the symptoms.

No.	Symptom	2006	2007	2008	2009	2010	2011	2006-2011	%
1	Result / imaging / reporting / dosage	41	25	57	26	19	59	227	24
2	Configuration / requirements	19	20	17	23	26	25	130	14
3	Alarms	18	29	5	9	55	2	118	13
4	Interface/data read and processing	19	19	12	16	23	22	111	12
5	Database / memory	17	5	26	9	9	35	101	11
6	Calculation	12	30	13	5	11	12	83	9
7	Software (more exactly not specified)	19	9	3	10	20	21	82	9
8	Calibration	7	15	1	1	0	1	25	3
9	Usability	20	1	0	3	0	0	24	3
10	Output control	5	5	2	7	3	1	23	2
Total		**177**	**158**	**136**	**109**	**166**	**178**	**924**	

Result, imaging, reporting, dosage – category includes all symptoms related with display, printing or dosing medium (e.g. drug, radiation) – in general, some kind of rather complex output operations. For example: image display general problem, patient image orientation markers problem, shifting or distorted images, loss of synchronisation between text and images, loss of text problems, incorrect report data or dosage.

Configuration, incompatibility, requirements – includes incompatibility between devices or software cases, configuration problems or non-compliance with the requirements (data loss during transfer between devices, flags configured incorrectly, incorrect default configuration settings). Problems of this type occur especially after software upgrade.

Alarms – class of anomalies associated with non-alarming or incorrect alarm or warning message generation (e.g. about battery needs service, audible alarms inappropriate).

Interface/data read and processing – group of malfunction related with software response to incoming data or user control (e.g. incorrect reading of the bar code, no response to a keystroke, general problem with the software support for touch panels, allowing the operator to enter incorrect or out of range data).

Database / memory – category includes all of the symptoms associated with operations on the databases of patient data or storage of parameters related to the course of treatment in the memory (e.g. images from different patients combined in one patient folder, images can be overwritten, incorrect history of the patient read from memory, loss of image, image saved with corrupted data).

Calculation – all situations in which a description of the reasons for recall clearly indicated a calculation error (e.g. dose calculation error, wrong algorithm, rounding error).

Calibration – set, which contains cases of incorrect calibration of the device (related with software anomaly; e.g. erroneously copied calibration data during service mode, not standardized correctly calibrator, fail calibration prior to use due to incorrect software).

Usability – category includes software negative impact on device usability (e.g. not ergonomic user interface, easy to run the wrong function, user documentation does not adequately characterise the use of some device feature, another language of patient handbook version than the country in which the equipment was sold).

Output control – similar category to “Result, imaging, reporting, dosage”, but covers well-described recalls related with output controls (e.g. a pump shut-off prematurely, machine may produce unexpected motions).

Software (more exactly not specified) – software related recalls for which it is not possible – based on the recall description – specify the reason.

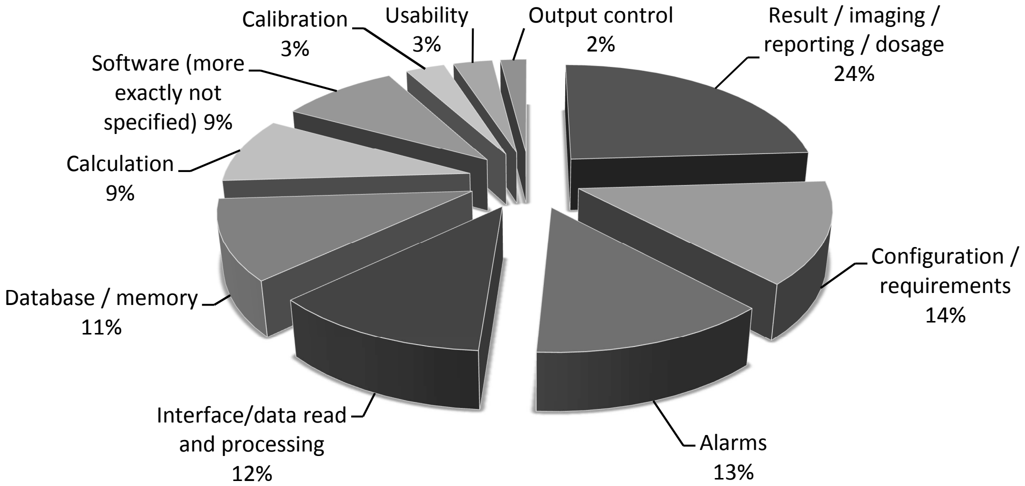

Fig. 2. The percentage of all software related recalls divided into classes according to the symptoms.

Analysis of the FDA recalls database has shown that manufacturers often report a few or even several recall events for one product, but different lots. Therefore, it has been decided to analyse the recalls database wherein multiple reporting are counting as a single. Since the results of both statements have come very close, it has been proven that multiple reporting has little impact on the final conclusions.

Total number of software related recalls divided into classes according to the symptoms wherein only one counting for the same products are shown in Tab. 3 and the percentage representation of them are shown in Fig. 3.

2.3.1. RESULTS ANALYSIS

Analysis of the recalls database has revealed that most cases apply to the category including all symptoms related with display, printing or dosing medium. Indeed, the presentation of a number of medical parameters has correlated with the degree of complexity of the device, so the occurrence software anomalies are also more likely. Of course, due to no additional information, it is possible that this class contains the cases of the other categories (e.g. calculation errors). Despite this uncertainty, the results of the analysis clearly indicate that this area of software application needs special controls in both the design and maintenance.

Table 3. No. of software related recalls divided into classes according to the symptoms (one counting for the same products).

No.	Symptom	2006	2007	2008	2009	2010	2011	2006-2011	%
1	Result / imaging / reporting / dosage	24	22	34	23	14	28	145	28
2	Configuration / requirements	16	14	11	17	14	15	87	16
3	Interface/data read and processing	15	11	12	9	16	9	72	13
4	Database / memory	17	5	15	6	8	14	65	12
5	Alarms	12	16	5	9	5	2	49	9
6	Calculation	8	17	13	3	3	7	51	9
7	Software (more exactly not specified)	3	6	3	9	3	14	38	7
8	Output control	4	4	1	4	2	1	16	3
9	Usability	6	1	0	3	0	0	10	2
10	Calibration	4	1	1	1	0	1	8	1
Total		**109**	**97**	**95**	**84**	**65**	**91**	**541**	

Special attention is needed not only during software development but also – and perhaps even more – while keeping the product on the market. As previously mentioned, the results of the other study indicate that the more often faulty software is introduced into the market as a result of various upgrades, so during maintaining part of the software life cycle.

In order to meet customer requirements, manufacturers are continually improving their products. This is closely related to the second category – configuration and requirement errors. Changing – often at the request of customers – one of the features on the device does not take into account all the negative impacts of this change on the other device feature.

Another very important category of mistakes are errors associated with the user interface, data entry into the system and running processes based on the entered data. A very common mistake is no filtering the input data, which leads to the possibility of introducing unexpected parameters. Following the acceptance by the software the value of the input for which the behavior of the system has not been established, may cause a serious threat to the life or health of the patient or the operator. Testing the user interface is not easy – especially when, for example, input parameters and their ranges are interdependent. In such situations, testing of all combinations of settings is very difficult and sometimes even impossible.

Quite a large part of all the software related recalls are database malfunctions. Especially dangerous are errors, the result of which parameters or medical images are assigned to improper patient in the database. This issue also requires to spend a lot of time, testing software – preferably on a large database of patients and data.

Here attention has been paid mainly to the problem of testing, but it should be noted that the well-defined processes such as specification of requirements, the division of software to the modules, modification, software modules integration, risk management, software release and its maintenance, are essential for minimizing the likelihood of arising on the market medical device with the failing software [3].

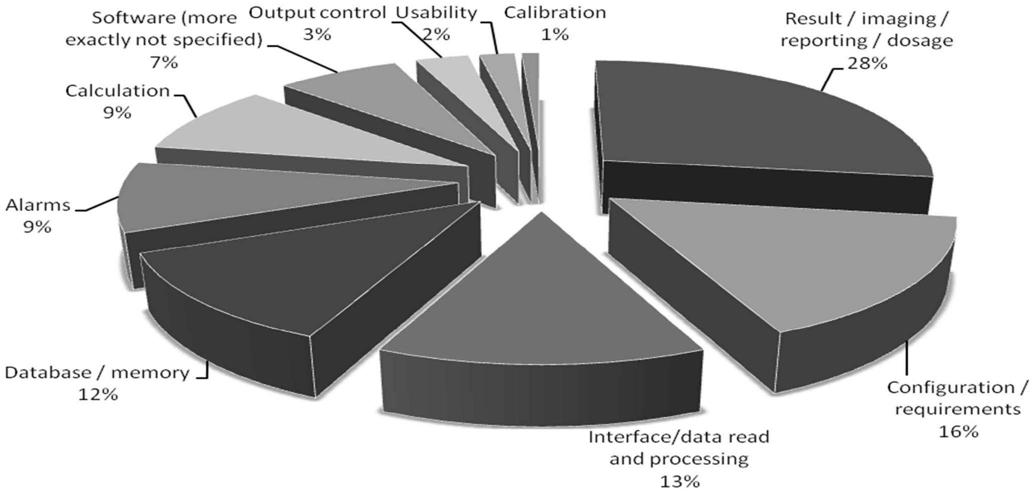

Fig. 3. The percentage of software related recalls divided into classes according to the symptoms (one counting for the same products)

3. CONCLUSIONS

The healthcare sector is one of the fastest growing economic sectors today and the medical device domain is one part of the sector. Medical devices and systems have been developed over many years but these types of products are now containing and are based on more and more software. On the medical device's path from development to market, the device is also affected by business and marketing decisions [4].

Software can be fast and easily changed (in referenced to the hardware) so it can be constantly updated and modified, such improvements are sometimes countered by new defects introduced into the software during the change [1]. This factor can cause both software and non-software professionals to believe that software problems can be at any time corrected fast and easily. Combined with a lack of understanding of software, it can lead managers to believe that tightly controlled engineering is not needed as much for software as it is for hardware. In fact, the opposite is true. Because of its complexity, the development process of software should be even more tightly controlled than for hardware, in order to prevent problems that cannot be easily detected later in the development process.

Testing of all program functionality does not mean all of the program elements have been tested. Testing of whole program's code does not mean all necessary functionality is present in the program. Testing of all program functionality and all program code does not mean the program is 100% correct [1].

Within software development there are a lot of possibilities but also difficulties. When developing software it is possible to make substantial changes late in the development process, which can be beneficial but can also cause serious incidents. In accordance with the principle that the best learning is learning from the mistakes made, the best when they are the mistakes of others, it is good to take into account the conclusions of this lesson in order to minimize the likelihood placing on the market defective medical software.

4. FURTHER RESEARCH

As previously mentioned, the FDA recalls database has been selected for considerations because it is the wide source of many cases with software related recalls. However, similar databases are maintained in most countries all over the world. For example in Poland, The Office for Registration of Medicinal Products, Medical Devices and Biocidal Products registers so-called *Safety Messages for medical devices* [6]. A glance at the database is shown in Tab. 4.

Table 4. No. of Safety Messages for medical devices in Poland.

Year	**No. of Safety Messages for medical devices**
2007	40
2008	69
2009	65
2010	101
2011	178
2012	192
Total	**645**

Thus, for example in 2012 from January up to now, 192 Safety Messages for medical devices have been registered, wherein almost 22% are software related problems. Detailed analysis should be the subject of further work.

BIBLIOGRAPHY

[1] CRUMPLER S., SNIPES CH., GODZIEMSKI A., General Principles of Software Validation; Final Guidance for Industry and FDA Staff, U.S. Department Of Health and Human Services, Food and Drug Administration, Center for Devices and Radiological Health, Center for Biologics Evaluation and Research, 2002.

[2] http://www.fda.gov/Training/CDRHLearn/ucm209125.htm.

[3] IEC 62304:2006, Medical device software - Software life cycle processes.

[4] LINDHOLM CH., Software Development and Risk Management in the Safety Critical Medical Device Domain, Licentiate Thesis, Department of Computer Science Lund University Faculty of Engineering, 2009.

[5] PETROSKI H., To Engineer Is Human, Vintage Books of Random House, Inc., 1992, New York.

[6] http://www.urpl.gov.pl/komunikaty-wyroby-medyczne.

[7] WALLACE D., KUHN R., Lessons from 342 Medical Device Failures, Information Technology Laboratory National Institute of Standards and Technology Gaitersburg.