Dublin Institute of Technology
ARROW@DIT

Articles | School of Computing

2012-08-17

Software Process Improvement to Assist Medical Device Software Development Organisations to Comply with the Amendments to the Medical Device Directive

Martin Mc Hugh
Dublin Institute of Technology, martin.mchugh@dit.ie

Fergal McCaffery
Dundalk Institute of Technology, fergal.mccaffery@dkit.ie

Valentine Casey
Dundalk Institute of Technology, val.casey@dkit.ie

Follow this and additional works at: http://arrow.dit.ie/scschcomart

Part of the Computer Sciences Commons

Recommended Citation

Mc Hugh, Martin and Mc Caffery, Fergal and Casey, Valentine (2012) Software Process Improvement to assist Medical Device Software Development Organisations to comply with the amendments to the Medical Device Directive. IET Software Special Issue on European Software Process Improvement.

This Article is brought to you for free and open access by the School of Computing at ARROW@DIT. It has been accepted for inclusion in Articles by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License

Software Process Improvement to assist Medical Device Software Development Organisations to comply with the amendments to the Medical Device Directive

Martin Mc Hugh, Fergal Mc Caffery, Valentine Casey
Regulated Software Research Group, Department of Computing and Mathematics,
Dundalk Institute of Technology
martin.mchugh@dkit.ie, fergal.mccaffery@dkit.ie, val.casey@dkit.ie

Abstract

A recent revision to the European Medical Device Directive (MDD) 2007/47/EC made fourteen amendments to the original directive (93/42/EEC). A number of these changes directly affect the development of software for use in healthcare. The most significant change in relation to medical device software development is that standalone software is now seen as an active medical device and should be developed following state of the art medical device software development processes. State of the art medical device software processes is understood within the industry as developing software in accordance with IEC 62304 and standards that are aligned with it. This paper identifies how changes to the MDD affect medical device software development companies and recommendations are made as to how medical device software development companies can conform to the latest regulatory requirements. Additionally, the paper provides an overview of how Medi SPICE is currently being developed to provide organisations with a single point of reference for the practices that should be implemented in order to produce regulatory compliant medical device software.

Keywords

IEC 62304, Medical Device Directive (MDD), Software Process Improvement, 2007/47/EC, AIMD, Medi SPICE, Agile Software Development

1 Introduction

Today the role and importance that software plays in the provision of healthcare continues to grow [1]. This has resulted in a substantial increase in the functionality, complexity and reliance on software components in medical devices [2]. As this reliance on software is increasing, incidents involving software failures in medical devices are brought to the fore. These include incidents such as Therac-25 [3] and the over exposure of Panamanian Teletherapy patients [4]. To reduce the potential risk a medical device poses to patient's safety, regulatory bodies enforce regulations that medical device manufacturers must conform to.

All medical devices used within the European Union (EU) must conform to the current MDD to achieve the CE conformance mark. The MDD revision (2007/47/EC) [5], amends European directives MDD (93/42/EEC) [6], AIMD (90/385/EEC) [7] and the Biocides Directive (98/8/EC) [8, 9] .The revision to the MDD (2007/47/EC) covers all areas relevant to medical devices including risk and quality management. This latest amendment allows for standalone software to be used as an active medical device. With this amendment, incidents involving medical devices such as Therac-25 become more relevant, as now software may be the only element in a medical device subject to conformance requirements.

Consequently methods used to ensure that software is safe and fit for purpose must be reviewed. A number of benefits can be gained by manufacturers employing Software Process Improvement (SPI) techniques, one of which is a reduction in software faults that could potentially result in device recalls [10]. SPI is a continuous cycle of performing an assessment, implementing the recommendations of the assessment and restarting the cycle [9]. This process of continuous assessment and improvement can help reduce the amount of defective software being developed. Essentially the safety of medical device software is determined through the software processes followed during development [11].

IEC 62304:2006 [12] is a harmonised standard as part of the MDD. IEC 62304 contains a number of processes for medical device software development which organisations are recommended to follow in order to implement medical device software best practices and to streamline the process of achieving regulatory approval. As IEC 62304 is a software development standard it does not provide guidance on system level activities. As a result, IEC 62304 hands off the system processes to aligned

standards such as ISO 13485:2003 [13], IEC 60601-1:2010 [14], IEC 61010-1:2010 [15], ISO/IEC 90003:2004 [16], IEC 61508-3:2010 [17], ISO 14971:2009 [18], ISO/IEC 12207:1995 [19] AMD1 [20] and AMD2 [21],and ISO/IEC 15288:2008 [22] [12]. The relationship between these standards and the EU medical device regulations is shown in table 1.

European Regulations	Active Implantable Medical Device (AIMD Directive 90/385/EEC, Medical Device Directive 93/42/EEC (Amendment 2007/47/EC), Biocides Directive 98/8/EC
Applicable Standards	ISO 13485 – Quality Management Systems, IEC 62304 – Software Lifecycle Processes, IEC 60601-1 Medical Electrical Equipment, IEC 61010-1* – Safety Requirements for Electrical Equipment, ISO 15288* – Systems and Software Engineering, ISO 90003* - Software Engineering, ISO 12207* – Systems and Software Engineering, ISO 14971 – Application of Risk Management ***Not Specific to the development of medical devices**
Technical Reports	IEC TR 80002-1

Table 1: European Regulations, Applicable Standards and Technical Reports applicable to the development of medical devices

As part of this research the changes to the MDD that have an impact on the development of medical device software are identified and explained. Also, recommendations are made as to how medical device software development organisations can conform to this latest MDD amendment and to other important international standards.

The remainder of this paper is structured as follows; section 2 examines the revision to the MDD and highlights what this means with respect to medical device software development. This will include particular reference to the development of standalone software as an active medical device. In section 3, we discuss the existing standards that are appropriate to the development of medical devices with emphasis on satisfying the requirements of the MDD (2007/47/EC). In section 4, we discuss the importance of SPI techniques and recommend a specific SPI model (Medi SPICE) to follow in order to achieve compliance with medical device software regulations. Section 5, contains the conclusions from this research.

2 European Medical Device Directive Amendment 2007/47/EC

The recent MDD [5] revision has made a number of amendments to previous directives i.e. [6, 7, 23]. The MDD revision came into force on March 21st 2010. In total there are fourteen changes introduced within the revised MDD [24]. There are three areas within the amendment of the revised MDD with important significance to medical device software development:

- Standalone Software as an active medical device;
- Validation of software as an active medical device;
- Outsourced Design and Manufacturing;

2.1 Standalone Software as an Active Medical Device

Prior to the release of the revised MDD provision had been made within the MDD (93/42/EEC) for software to be used as a component of a medical device. However, the revised MDD Article 1 Section 2 makes explicit reference to software (used alone or in combination) being a medical device.

*"any instrument, apparatus, appliance, **software**, material or other article, whether used alone or in combination, including the software intended by its manufacturer to be used specifically for diagnostic and/or therapeutic purposes and necessary for its proper application"*

To accompany this change provision has also been made for standalone software to be used as an active medical device. Within the revised MDD Annex IX Section 1.4 amendment M5 states:

"Stand-alone software is considered to be an active medical device"

This can be difficult to understand particularly in relation to when software is or is not a medical device. An example of software as an active medical device is a software package which is used to calculate treatment doses for oncology treatment devices. A caveat has also been included into the revised MDD to avoid ambiguity in determining if a software package is a medical device.

"software for general purpose when used in a healthcare setting is not a medical device"

This caveat provides some clarity surrounding particular software used in healthcare. The European Commission released a MEDDEV guidance document in January 2012 [25]. This document provides clarity as to which types of software used in healthcare meet the criteria of being standalone software which is subject to regulatory conformance.

2.2 Software Validation

As standalone software is now classified as an active medical device under the revised MDD, safeguards must be put in place to ensure that such software is safe and fit for purpose. To ensure this the MDD Annex I Section 12.1a (amendment M5) states;

"For devices which incorporate software or which are medical software in themselves, the software must be validated according to the state of the art taking into account the principles of development lifecycle, risk management, validation and verification."

"State of the Art" is used here to mean what is generally accepted as good practice. Since this requirement was introduced, software development companies must now validate software whether integrated or standalone, regardless of device class. IEC 62304 is aligned with standards: ISO 13485, IEC 62366 [26], EN 60601 and ISO 14971; and is harmonised under the MDD. Therefore, medical device software organisations adopting IEC 62304 together with the standards it is aligned with are considered to be following "state of the art" practices in relation to medical device software development and maintenance.

2.3 Outsourced Design and Manufacturing

Outsourcing is becoming more and more common place in all industries This is particularly relevant with regard to software development which due to the popularity of outsourcing can be now be considered a globally sourced commodity [27, 28]. As part of the latest MDD amendment, should a device manufacturer outsource any part of the design or manufacturing process, then the manufacturer must be able to demonstrate that adequate controls have been put in place to ensure the supplier is fully utilising a quality management system. Currently, there is no internationally accepted assessment model utilised for medical device manufacturers when assessing the ability of a medical device software supplier to develop software in a regulatory compliant manner. However, in Section 4, we introduce Medi SPICE, which is currently under development and once completed may be used by medical device manufacturers as a method for accessing the capability of different software suppliers in relation to developing software using regulatory compliant software processes.

3 IEC 62304 – Software Lifecycle Processes

Medical device manufactures wishing to achieve regulatory conformance are advised to follow the relevant applicable standards. Evidence of following the applicable standards can improve the process of achieving regulatory conformance.

3.1 Processes

IEC 62304 is a medical device software development lifecycle process standard. It was created to provide assistance in terms of the safe design and maintenance of medical device software. Software developed that adheres to IEC 62304 activities and tasks is founded upon the assumption that the software is developed in accordance with a quality management standard (e.g. ISO 13485), a risk management standard (ISO 14971) and a product level standard (EN 60601-1). This standard provides a framework of processes divided into activities which are further divided into tasks.

IEC 62304 provides guidance on the development of software as part of a medical device. However, IEC 62304 does not provide guidance on all of the necessary processes required to develop standalone software as an active medical device. IEC 62304 states:

"This standard does not cover validation and final release of the medical device, even when the medical device consists entirely of software"

With the MDD revision a medical device can consist only of software. As validation is required to ensure reliability and safety another method of validating standalone software as a medical device is required. In relation to this an international joint working standards group IEC TC62A JWG 7 has been created with members from both the medical device and software standards communities to develop a new standard IEC 82304 for Healthcare Software. IEC 82304 will treat standalone software as a complete system. IEC 62304 is currently under revision and the developers of Medi SPICE [29] are working towards providing the Medi SPICE process reference model as an annex within the revision of IEC 62304.

3.2 Safety Classification

IEC 62304 classifies medical device software based on the risk posed to the patient or user. The devices are classified as follows:

- 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.

This classification is similar to that of ISO 14971 Clauses 4.4, 5 and 6.1. Safety critical software systems can be divided into items running a different software element each with its own safety classification. These items can be further sub-divided into additional software elements. The overall software system assumes the highest classification contained within all of the software elements. For example, if a software system contains five software elements, four of which may be classified as Class A, but one may be classified as Class C then the overall device receives a classification of Class C. This can be seen in figure 1.

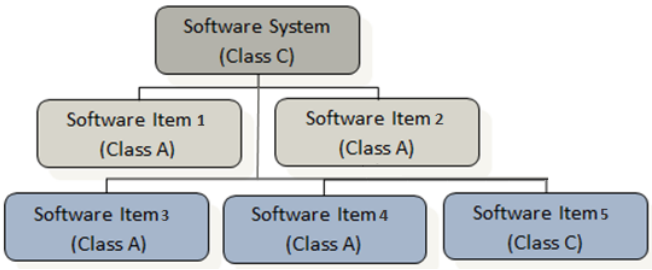


Figure 1 Classification of software items within complete software system

However, IEC 62304 makes provision for a software item to be segregated from the overall software system. This allows for the segregated software item to independently receive a lower safety classification.

4 Software Process Improvement and Medi SPICE

Software Process Improvement is an important element within any software development lifecycle. Many organisations have difficulty in consistently developing high quality software. There are many benefits to be gained by using SPI including [30]:

- Improvements to overall quality;
- Increased on-time delivery;
- Budget consistency;
- Reduced development costs.

4.1 Importance of SPI

SPI places the emphasis on defining processes that are appropriate to the project and ensures that these processes are consistently followed. SPI maturity models focus on what has to be done, rather than how it should be done. The benefits of utilising SPI can be seen in many companies e.g. Siemens [31], Alcatel [32], NASA [33] and Motorola [34].

In order for SPI to be successful within an organisation, it relies on a number of critical factors. In 2005 a survey of one hundred and twenty software organisations identified six organisational factors as being crucial to ensure the success of SPI [35]:

- Business orientation;
- Involved leadership;
- Employee participation;
- Concern for measurement;
- Exploitation of existing knowledge;
- Exploration of new knowledge.

Research carried out by Embedded Market Forecasters in 2010 [36] provided a comparison between software developed by the embedded industry and software developed by medical device producers. This research showed that 12.9% of medical device projects were cancelled, whilst 11.2% of embed-

ded industry projects were cancelled. This research also revealed that on average 19.4% of the overall budget is wasted due to months lost during project development. The primary reason cited for these problems occurring is incomplete or vague requirements.

An empirical study in 2007 revealed how much importance medical device software development companies place upon SPI [37]. The study surveyed organisations developing software for medical devices and medical information systems with the majority of respondents coming from Germany, USA and Sweden. Seventy-one percent of respondents came from small and medium companies with between ten and two hundred & fifty employees. The remainder of the respondents came from organisations with over two hundred & fifty developers. With ninety-eight percent rating software as either an important or very important part of their products. However, only fourteen percent had a CMMI (Capability Maturity Model Integration) or ISO 15504 (SPICE) rating. This survey also asked participant's which process or activities cause the most issues for a software development project.

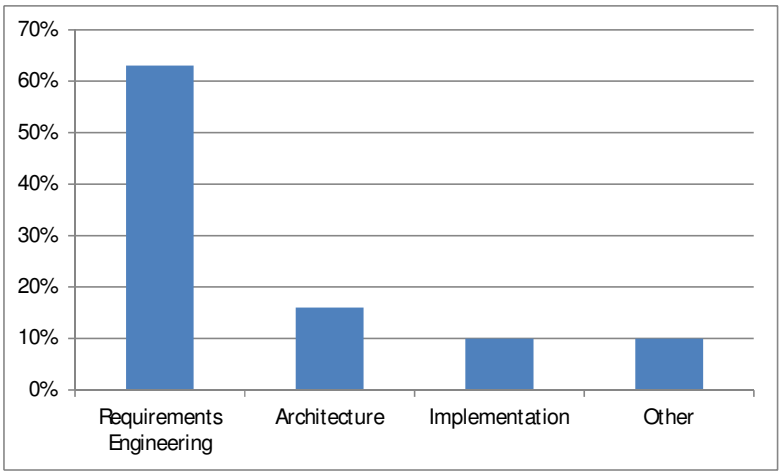


Figure 2 Activities & Processes most difficult to software development projects [37]

Figure 2 shows that the Requirements Engineering process is seen as causing problems in the majority of medical device software development. Denger's [37] research reveals that organisations typically following a defined set of processes or activities contained within an SPI model will have minimal problems with areas such as requirements engineering as they are provided guidance on all areas of development. This survey shows an inversely proportional relationship between importance placed on SPI rating/activities and difficulties caused by specific process areas.

4.2 Medi SPICE

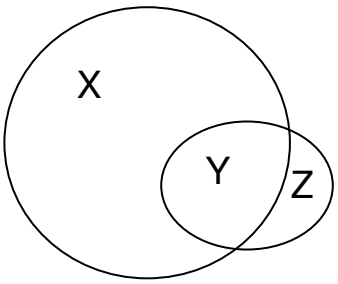
Medical device software development companies face requirements and difficulties not faced by developers of traditional software [38]. A primary requirement for medical device software development companies is the achievement of regulatory conformance. This is essential as organizations must achieve compliance and approval by the regulatory body of the country where they wish to market their medical device. Current SPI models such as CMMI and SPICE were developed for generic software development and therefore were not designed to provide sufficient coverage of all of the necessary areas needed in order to achieve medical device regulatory approval. The is recognized by a comparison made by Walker [39] between CMMI and the medical device regulations which states that CMMI can be used, but there is a requirement for amended and extended processes to be put in place to meet the specific requirements of this domain. Similarly Burton et al [40] and McCaffery and Dorling [41] have each identified that existing SPI models can be used in the development of medical device software, however, these models do not provide sufficient guidance in key development areas such as risk management and need to be amended and extended. Each of these studies also identify the need for a single point of reference which medical device software development companies can follow when developing medical device software. To address this requirement extensive research was undertaken [42] and as a result Medi SPICE [29] is currently under development by the Regulated Software Research Group (RSRG) at Dundalk Institute of Technology (DkIT) in collaboration with the SPICE User Group. The objective of Medi SPICE is to provide a framework which facilitates the implementation of high quality medical device software processes that will enable seamless conformity to the medical regulatory standards. This approach is in line with the development of a domain specific SPI model for the automotive industry - Automotive SPICE [43].

Medi SPICE aims to minimise the volume of software documentation and provide global harmonisation for all medical device software manufactures. The results of a Medi SPICE assessment may be used to indicate the current state of a medical device supplier's software development practices in relation to regulatory expectations. Therefore, the results of a Medi SPICE assessment may also be used as a criterion for medical device software supplier selection. This is particularly relevant with regard to the amendment's requirement that manufactures should demonstrate that adequate controls have been put in place to ensure the supplier is fully utilising a quality management system. Medi SPICE will also address the specific requirements of designing and developing medical device

SPICE will also address the specific requirements of designing and developing medical device software in a globally distributed environment [44, 45]

4.2.1 Medi SPICE Structure

Medi SPICE is being developed in line with the requirements of ISO/IEC 15504-2:2003 [46] and incorporates the requirements of IEC 62034 and the other relevant medical device regulations, standards and guidance documents. Medi SPICE contains a Process Reference Model (PRM) and a Process Assessment Model (PAM). Medi SPICE consists of forty-three processes and twelve sub-processes each of these processes contain a process purpose, a number of outcomes and a number of specific practices that will have to be performed in order to fulfil the outcomes. The performance of the specific practices provides an indication as to the extent of achievement of the process purpose and outcomes. Work products which are either used, produced or both, when performing the process are also recorded. The composition of the Medi SPICE processes is illustrated in figure 3.



X- ISO/IEC 15504-5 Practices that are not mandatory for regulatory compliance.
Y- ISO/IEC 15504-5 Practices that are required for regulatory compliance.
Z- Non-ISO/IEC 15504-5 Practices that are required for regulatory compliance.

Figure 3 Composition of Medi SPICE processes [41]

4.2.2 Process Reference Model

Given the importance of conformance to the latest standards the decision was taken to develop the Medi SPICE PRM in line with ISO/IEC 12207:2008 [47] and the next release of ISO/IEC 15504-5 (currently under ballot). The first step in the development of the Medi SPICE PRM was the selection of the relevant processes. In order to achieve this objective two key requirements needed to be addressed:

1) The selection of effective life cycle processes;

2) The selection of processes that facilitate conformance to the necessary medical device regulation standards and guidance documents.

The structure of ISO/IEC 12207:2008, and the next release of ISO/IEC 15504-5 were both reviewed in detail. Extensive analysis of the relevant medical device regulations and standards were undertaken. Based on this work forty-three Medi SPICE processes and twelve sub-processes were identified and defined. These were then released for review by interested parties from the SPICE User Group and industry experts. Following their approval the Medi SPICE PRM was structured as follows:

- The System Life Cycle Processes contains:
    - 3 Agreement Processes and 7 Sub-processes;
    - 6 Organizational Project - Enabling Processes and 6 Sub-processes;
    - 7 Project Processes;
    - 10 Technical Processes and 2 Sub-processes.
- The Software Life Cycle Processes contains:
    - 6 Software Implementation Processes;
    - 9 Software Support Processes;
    - 1 Supplementary Process.

Having defined the processes and structure of the PRM the developers of Medi SPICE were invited to participate in the current revision of IEC 62304, to both assist with the alignment of IEC 62304 with ISO/IEC 12207:2008 and also to provide details to the medical device software development community of the relationship between IEC 62304 and other medical device standards and guidelines.

Work has commenced on the development of the contents of the Medi SPICE PRM processes. The initial focus was on the IEC 62304 relevant processes. In line with the requirements of ISO 15504-2 each process was assigned an ID and name, with a process purpose also being defined. Based on the process purpose, outcomes were identified. The purpose and outcomes addressed the requirements for an effective process and those of the medical device standards and regulations. The regulatory aspects were addressed by undertaking a detailed analysis of the relevant standards and guidance documents with reference to each process. The Medi SPICE PRM in addition to the normal content of a PRM details the source (standard or guidance document) of each outcome and where relevant an outcome is given a safety classification (if the source of the outcome is IEC 62304).

The sixteen processes which constitute the subset of the Medi SPICE PRM for inclusion in the next release of IEC 62304 have been completed. These are currently being reviewed by interested

parties from the SPICE User Group, industry experts and the IEC SC62A JWG3 Standards working group (the IEC 62304 development team). It is planned that this subset of the Medi SPICE PRM will be included in the Appendix of the forthcoming release of IEC 62304. The development of the remaining Medi SPICE PRM processes is currently under way**.**

4.2.3 Process Assessment Model

The Medi SPICE PAM is also structured in line with ISO/IEC 12207:2008 and the next release of ISO/IEC 15504-5 (currently under ballot). The PAM is related to the Medi SPICE PRM and forms the basis for collecting evidence and the rating of process capability. This is achieved by the provision of a two-dimensional view of process capability. In one dimension, it describes a set of process specific practices that allow the achievement of the process outcomes defined in the PRM; this is termed the process dimension.

In the other dimension, the PAM describes capabilities that relate to the process capability levels and process attributes, this is termed the capability dimension. Process attributes are used to determine whether a process has reached a given capability. Each attribute measures a particular aspect of the process capability Indicators for process capability are generic practices that are applicable to any process and are associated with process attributes, generic work products and generic resources that can be observed when a particular process attribute is achieved. In line with ISO/IE 15504-2 the Medi SPICE process capability is defined over 6 levels:

- Level 0 Incomplete;
- Level 1 Performed;
- Level 2 Managed;
- Level 3 Established;
- Level 4 Predictable;
- Level 5 Optimizing.

4.2.4 Medi SPICE Validation

The Medi SPICE PRM and PAM are being released in stages and each stage is extensively reviewed and validated prior to release. A key aspect of the development of Medi SPICE is that it is being un-

dertaken in a collaborative manner and input and approval are sought from interested parties in the SPICE User Group, international standards bodies and representatives from the medical device software industry. In this way we ensure that each release of the Medi SPICE PRM and PAM are validated by both the SPI community and industry experts.

In addition we have also developed and successfully implemented a lightweight medical device software process centric assessment model with the objective of validating key aspects of Medi SPICE. Known as Medi SPICE Adept [48] this method currently enables the assessment of ten Medi SPICE processes, although it will be extended to include all Medi SPICE processes whenever they have been fully developed. It allows organizations to have each of these processes assessed regarding their conformance to the specific requirements of the medical device regulations, standards, and guidance documents. It also provides a method for carrying out an assessment against the requirements for best software engineering practice. The successful implementation of this method has allowed key aspects of Medi SPICE to be tested and validated in an industrial setting. It has also provided relevant feedback which has been documented and incorporated into the on-going development of the Medi SPICE. Based on this work future releases of Medi SPICE will be trialled and validated in an industrial setting.

5 Conclusions

The recent revision to the MDD allows for standalone software to be an active medical device and states that software must be validated in accordance with state of the art practices. This paper focuses on changes as part of the MDD which have a direct impact on the development of medical device software. The most significant change as part of this recent amendment was that standalone software previously exempt from regulatory control now falls under the umbrella of requiring regulatory approval. To accompany this, recommendations are made as to how medical device software development companies can conform to the latest regulatory requirements by utilising SPI models such as Medi SPICE.

Currently, no single point of reference is available that provides medical device software development companies guidance on end to end development of both embedded software and standalone software that conforms to the latest regulatory requirements. Existing standards only include

practices specific to themselves and practices in other standards are not included. For example, IEC 62304 does not currently address the systems aspect of software development and therefore does not address the requirements elicitation stage of development. IEC 62304 hands off these system level processes to its aligned standards. However, research has shown that the requirements elicitation stage is the development phase which presents medical device software development companies with the most issues [36, 37]. The authors of this paper are currently developing Medi SPICE to provide guidance for all stages of development of both embedded and standalone medical device software. Additionally, Medi SPICE will provide a single source for obtaining all medical device regulatory requirements. The development of Medi SPICE is on-going, but it has been piloted in a number of Irish and Australian organisations using a lightweight software process assessment method Medi SPICE Adept [48].

Acknowledgements

This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/I1299, the SFI Principal Investigator Programme, grant number 08/IN.1/I2030 (the funding of this project was awarded by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund), and supported in part by Lero - the Irish Software Engineering Research Centre (http://www.lero.ie) grant 03/CE2/I303_1.

6 References

1. Abraham, C., E. Nishiharas, and M. Akiyama, *Transforming healthcare with information technology in Japan: A review of policy, people, and progress.* International Journal of Medical Informatics 2011. **80**(2011): p. 157-170.
2. Lee, I., G. Pappas, R. Cleaveland, J. Hatcliff, B. Krogh, P. Lee, H. Rubin, and L. Sha, *High-Confidence Medical Device Software and Systems.* Computer, 2006. **39**(4).
3. Leveson, N., *Safeware: System Safety and Computers*. 1995: Addison Wesley.
4. Borrás, C., *Overexposure of Radiation Theraphy in Panama* Articles and special reports, 2006: p. 173-187.
5. European Council, *Directive 2007/47/EC of the European Parliament and of the Council of 5 September 2007*, European Council, Editor. 2007.
6. *European Council, "Council Directive 93/42/EEC Concerning Medical Devices", 14-June-1993*.
7. *European Council, 1990 "Council Directive 90/385/EEC", 5 September 2007*.
8. *European Council, 2009 "Implementation of Directive 2007/47/EC amending Directives 90/385/EEC, 93/42/EC and 98/8/EC ", 5 June 2009*.

9. McCaffery, F. and G. Coleman, *The need for a software process improvement model for the Medical Device Industry.* International Review on Computers and Software, 2007. **2**(1): p. 10-15.
10. Patra, P. and S. Bartaki, *Productivity Improvement Using Ten Process Commandments.* PMI Virtual Library, 2009.
11. Jones, P.L., J. Jorgens, A.R.T. Jr, and M. Weber, *Risk Management in the Design of Medical Device Software Systems.* Biomedical Instrumentation & Technology: July 2002, 2002. **36**(4): p. 237-266.
12. AAMI, *ANSI/AAMI/IEC 62304, Medical device Software - Software life cycle processes*. 2006.
13. *EN ISO 13485:2003 Medical Device: Quality Management Systems. Requirements for the Regulatory Process, 24-July-2003*.
14. *EN 60601-1 Medical Electrical Equipment. General requirements for basic safety and essential performance. Collateral standard. Usability 31-May2010*.
15. *IEC 61010-1:2010 Safety requirements for electrical equipment for measurement, control and laboratory use, June 2010*.
16. *ISO/IEC 90003:2003 Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software, February 2004*.
17. *IEC 61508-3:2010 Functional Safety of electrical\electronic\programmable electronic safety related systems, May 2010*.
18. *EN ISO 14971:2009 Medical Devices. Application of Risk management to medical devices 31-July-2009*.
19. ISO, *ISO/IEC 12207 Information technology -- Software life cycle processes*. 1995: International Organisation for Standards - Geneva, Switzerland.
20. ISO, *ISO 12207:1995/AMD1:2002*. 2002, International Standards Organisation: Geneva Switzerland.
21. ISO, *ISO 12207:1995/AMD2:2004*. 2004, International Standards Organisation: Geneva Switzerland.
22. ISO, *ISO/IEC 15288:2008 Systems and software engineering -- System life cycle processes*. 2008: Geneva, Switzerland.
23. *European Council, 1998 "Council Directive 1998/8/EEC", February 1998*.
24. http://www.emergogroup.com/newsletters/europe-2007-47-ec-oct2007 Accessed February 2011
25. European Commission, *Medical Devices Guidance Document - Qualification and Classification of stand alone software MEDDEV 2.1/6*. 2012.
26. *IEC 62366 Medical Devices - Application of usability engineering to medical devices, 13-November-2007*.
27. Casey, V., *Virtual Software Team Project Management*. Journal of the Brazilian Computer Society, 2010. **16**(2): p. 83-96.
28. Damian, D. and M. D, *Global Software Development: How Far Have We Come?* IEEE Software, 2006. **23**(5): p. 17-19.
29. McCaffery, F. and A. Dorling, *Medi SPICE Development.* Software Process Maintenance and Evolution: Improvement and Practice Journal, 2010. **22**(4): p. 255-268.
30. O'Regan, G., *Introduction to Software Process Improvement* 2010: Springer.
31. Messer, T., P. Paul, F. Paulisch, P. Schless, T. Mehner, and A. Volker, *Siemens Process Assessment and Improvement Approaches: Experiences and Benefits*, in *22nd International Computer Software and Applications Conference*. 1998: Vienna, Austria.
32. Debou, C., D. Courtel, H.B. Lambert, N. Fuchs, and M. Haux, *Alcatel's Experience with Process Improvement*, in *Better Software Practice for Business Benefits: Principles and Experience* 1999, CS Press. p. 281-301.
33. Basili, V.R., F.E. McGarry, R. Pajerski, and M.V. Zelkowitz, *Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory*, in *24th International Conference on Software Engineering*. 2002: Orlando Florida. p. 69-79.
34. Daskalantonakis, M.K., *A Practical View of Software Measurement and Implementation Experiences Within Motorola.* IEEE Trans. Softw. Eng., 1992. **18**(11): p. 998-1010.
35. Dyba, T., *An Empirical Investigation of the Key Factors for Success in Software Process Improvement.* IEEE Trans. Softw. Eng., 2005. **31**(5): p. 410-424.
36. Embedded Forecasters, *Embedded Market Forecasters Survey (2010)*. 2010.

37. Denger, C., R.L. Feldman, M. Host, C. Lindholm, and F. Schull, *A Snapshot of the State of Practice in Software Development for Medical Devices*, in *First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007*. 2007: Madrid.
38. Ge, X., R.F. Paige, and J.A. McDermid, *An Iterative Approach for Development of Safety-Critical Software and Safety Arguments*, in *Agile 2010*. 2010.
39. Walker, D., *CMMI and Medical Device Engineering* SEI Library, 2009.
40. Burton, J., F. McCaffery, and I. Richardson, *A risk management capability model for use in medical device companies*, in *Proceedings of the 2006 international workshop on Software quality*. 2006: Shanghai, China. p. 3-8.
41. McCaffery, F. and A. Dorling, *Medi SPICE: An Overview.* Journal of Software Maintenance and Evolution: Research and Practice, 2010. **22**(4): p. 255-267.
42. McCaffery, F., J. Burton, V. Casey, and A. Dorling, *Medical Device Industry: Software Process Improvement*, in *Encyclopedia of Software Engineering*, P. Laplanye, Editor. 2010, CRC Press Francis Taylor Group: London. p. 528-540.
43. *Automotive SIG, Automotive SPICE Process Assessment V 2.2. 21 August 2005*.
44. Casey, V., *Trust in Virtual Software Development Teams* Special Issue and on Trust and Trust Management of the Journal of Theoretical and Applied Electronic Commerce Research, 2010. **5**(2): p. 41-58.
45. Richardson, I., V. Casey, J. Burton, and F.McCaffrey, *Global Software Engineering: A Software Process Approach in Collaborative Software Engineering*. 2010, Springer London. p. 35-56.
46. ISO, *ISO/IEC 15504-2:2003 Information technology -- Process assessment -- Part 2: Performing an assessment*. 2003: Geneva Switzerland.
47. ISO, *ISO/IEC 12207:2008 Systems and software engineering -- Software life cycle processes*. 2008: Geneva, Switzerland.
48. McCaffery, F., V. Casey, and M. McHugh, *How Can Software SMEs Become Medical Device Software SMEs*, in *European Systems and Software Process Improvement and Innovation Conference, EuroSPI* 2011, Springer: Roskilde, Denmark p. 247-258.