Medical Device Software as a Subsystem of an Overall Medical Device

The MDevSPICE® Experience

Fergal McCaffery, Marion Lepmets, Paul Clarke
Regulated Software Research Centre & Lero,
Department of Computing & Mathematics
Dundalk Institute of Technology
Co. Louth, Ireland
Email{fergal.mccaffery, marion.lepmets, paul.clarke}@dkit.ie

***Abstract*—Embedded software is a sub-system that needs to be integrated with the electrical and mechanical subsystems for a functional medical device to be developed and marketed. In order to be able to develop a medical device system through integrating its sub-systems, the complete system requirements should be known at the start of the project and managed throughout development. Software requirements are then derived from the systems requirements. We have developed and piloted a medical device software process assessment framework called MDevSPICE® that integrates processes from various medical device software standards as well as generic software development standards. This paper describes how the MDevSPICE® framework has been designed so as to enable medical device software developers to produce software that will be safe and easily integrated with other sub-systems of the overall medical device. We also describe the lessons learned from piloting MDevSPICE® in the medical device industry and challenges medical device software developers meet in tracing requirements and risks to and from the system level.**

Keywords- software integration; medical device software; MDevSPICE®; medical device risks; medical device.

I. INTRODUCTION

Safety-critical software systems are increasingly affecting our lives and welfare as more and more software is embedded into medical devices, cars and airplanes each day. New approaches and international standards are being developed to ensure the safety of these systems before they are delivered. In order to market a medical device, for example, the manufacturer has to satisfy a number of regional regulatory requirements commonly achieved by following international standards and guidance issued by international standardizing bodies and regional regulatory authorities. To help software companies in the medical device domain in their attempt to reach regulatory compliance, we have developed an integrated framework of medical device software development best practices called MDevSPICE®. This framework integrates generic software development best practices with medical device standards' requirements enabling robust software process assessments to be performed. The "SPICE" in MDevSPICE® reflects its foundation in the ISO/IEC 15504 (SPICE) [25] series of standards for process assessment. Through validating the MDevSPICE® framework we provide evidence of the importance of traceability between the system and software levels of development – and explain how the establishment of robust requirements interfacing between these levels can support more effective software integration

In Section II, we describe the regulatory requirements medical device software development companies face before they are able to market their devices. In Section III we describe the development of the MDevSPICE® framework We then focus in Section IV on the lessons we learned when validating the framework in expert reviews and in industry through MDevSPICE® pilot assessments. We also discuss the importance of traceability between system and software development processes when developing an embedded medical device software system as it increases the safety and quality of the developed medical device. The paper concludes in section V.

II. MEDICAL DEVICE REGULATION

A medical device can consist entirely of software or have software as a component of the overall medical device system. In order to be able to market a medical device within a particular region it is necessary to comply with the associated regulatory demands. Two of the largest global bodies responsible for issuing and managing medical device regulation belong to the central governing functions of the US and EU. In the US, the Food and Drug Administration (FDA) issues the regulation through a series of official channels, including the Code of Federal Regulation (CFR) Title 21, Chapter I, Subchapter H, Part 820 [1]. Under US regulation, there are three medical device safety classifications: Class I, Class II and Class III. The medical device safety classification is based on the clinical safety of the device. Class I devices are not intended to support or sustain human life, and may not present an unreasonable risk of harm. A thermometer is a Class I device. Class II devices could cause damage or harm to humans. An example of a Class II medical device is a powered wheelchair. Class III medical devices are usually those that support or sustain human life, and are of significant importance in the prevention of human health impairment. An example of a Class III device is an implantable pacemaker. All

implantable devices are Class III medical devices as the surgery required carries with itself additional high risks from anesthesia and possible infections that go beyond the safety risks of the medical device.

In the EU, the corresponding regulation is outlined in the general Medical Device Directive (MDD) 93/42/EEC [2], the Active Implantable Medical Device Directive (AIMDD) 90/385/EEC [3], and the *In-vitro* Diagnostic (IVD) Medical Device Directive 98/79/EC [4] - all three of which have been amended by 2007/47/EC [5]. Similarly to the US, the EU device safety is also based on the clinical safety of the device embodying similar classifications and limitations, where Class I in the EU corresponds to Class I in the US, Class IIa and IIb to Class II, and Class III to Class III.

A further safety classification applies to the software in medical devices as outlined in IEC 62304:2006 [6], where the safety classification is determined based on the worst possible consequence in the case of a software failure. In the case of failure of software that is of safety Class A, no injury or damage to health of a patient can occur. When software of safety class B fails, injury may occur but it is not serious or life-threatening. Class C medical device software is of highest risk and in the case of failure of such software death or serious injury can happen. Depending on the functionality of software within the medical device, the software safety classification may vary from the overall medical device safety class. When software is of critical functionality of the medical device, it will carry the same classification as the device, i.e., Class C software in Class III device. The safety classification of software may be lower but cannot be higher than the overall medical device safety class, e.g., software of safety Class B, may be embedded in Class III device but there cannot be software of safety Class C, in a Class I or Class II device.

Medical device manufacturers in the US as well as in EU must satisfy quality system requirements to market their developed devices. In the medical device domain, ISO 13485:2003 (ISO 13485 from hereon) [7] outlines the requirements for regulatory purposes from a Quality Management System (QMS) perspective in medical device domain. ISO 13485, which is based on ISO 9001 [8], can be used to assess an organization's ability to meet both customer and regulatory requirements in the medical device domain. ISO 13485 does not, however, include requirements for software development. IEC 62304, which can be used in conjunction with ISO 13485, does offer a framework for the lifecycle processes necessary for the safe design and maintenance of medical device software. As a basic foundation, IEC 62304 assumes that medical device software is developed and maintained within a QMS such as ISO 13485, but does not require an organization to be certified against ISO 13485. Therefore, IEC 62304 can be considered to be a software development specific supplement to ISO 13485, similar to ISO 90003 for ISO 9001.

IEC 62304 is based on ISO/IEC 12207:1995 [9] which although a comprehensive standard for software development lifecycle processes has effectively been decommissioned following the publication of the more extensive ISO/IEC 12207:2008 [10]. Furthermore, other developments in the ISO and IEC communities for software development, such as ISO/IEC 15504 [11], have provided significant additional levels of software process detail to support ISO/IEC 12207:2008. IEC 62304 is a critical standard for medical device software developers as it is the only standard that provides recommendations for medical device software implementations based on the worst consequences in the case the software failure causing hazards. Furthermore, for general medical device risk management, IEC 62304 is used in conjunction with ISO 14971 [12] and IEC 80002-1 [13] that provides guidance on the application of ISO 14971 for software development.

Since IEC 62304 considers a medical device system to consist of software as a sub-system, the system or product level requirements are not included within IEC 62304 but instead within the medical device product standard IEC 60601-1 [14]. Due to the increasing importance of usability of devices within the medical device industry, organizations should also adhere to the medical device usability engineering process requirements outlined in IEC 62366 [15]. When Medical Device Directives were amended in 2007 [5], it allowed standalone software to be defined as a medical device in its own right. Previously, software had always been seen as a subsystem embedded in a medical device. This amendment revealed a gap in international standards as none of the published standards were addressing the concerns for standalone software as a medical device. Today, IEC CD 82304-1 [16] applies to the safety of health software that is designed to operate on general purpose IT platforms and that is intended to be placed on the market without dedicated hardware, e.g., iPad applications.

All companies planning to market a medical device in the United States need to register their product with the US FDA. Most Class I devices can be self-registered but most Class II devices require a 510(k) submission. For Class III devices, a Pre-Market (PMA) submission is needed. To support manufacturers in addressing the relevant guidance, the FDA has issued an overview of their guidance documents for medical device manufacturers and software developers [17]. The FDA Guidance on Premarket Submissions [18] provides guidance and recommendation for premarket submissions for software devices, including standalone software applications and hardware-based devices that incorporate software. Premarket submission includes requirements for software-related documentation that should be consistent with the intended use of the Software Device and the type of submission. The FDA Guidance on Off-The-Shelf Software Use in Medical Devices [19] was published in 1999 with the purpose of describing the information that should be provided in a medical device application that uses off-the-shelf (OTS) software. Many of the principles outlined in this guidance document may also be helpful to device manufacturers in establishing design controls and validation plans for use of off-the-shelf software in their devices. The FDA General Principles of Software Validation [20] outlines general validation principles that the FDA considers to be applicable to the validation of medical device software or the validation of software used to design, develop, or manufacture medical devices. This guidance describes how

certain provisions of the medical device Quality System regulation apply to software. The scope of this guidance is somewhat broader than the scope of validation in the strictest definition of that term to support a final conclusion that software is validated.

The challenge that software development companies in the medical device domain face when they want to market a device is in the adherence to a large number of regulatory requirements specified in various international standards (that can often become overwhelming). In order to help these companies better prepare for demanding and costly regulatory audits, we developed the MDevSPICE® framework. MDevSPICE® includes requirements from the previously mentioned standards and guidance documents rendering the task of regulatory compliance much less complex. Following is a description of the development of the MDevSPICE® framework that integrates the requirements from various international medical device standards and guidance documents with the generic software development best practices while providing a possibility to assess processes.

III. MDEVSPICE® FRAMEWORK

This section describes the development of the MDevSPICE® process reference model, how MDevSPICE® provides support for integration, and how MDevSPICE® was piloted in industry.

A. *Development of the MDevSPICE® Process Reference Model*

A process reference model (PRM) describes a set of processes in a structured manner through a process name, process purpose and process outcomes where the process outcomes are the normative requirements the process should satisfy to achieve the purpose of the process. In order to develop a PRM that integrates requirements from various standards allowing the processes to be evaluated in terms of their achievement of their purpose statements, we followed the format of the process description illustrated in ISO/IEC 24774 [21]. With that in mind, we first mapped and integrated the requirements from ISO/IEC 12207:2008 and IEC 62304 into what today is called the PRM for IEC 62304 that also reflects the updates to ISO/IEC 12207 from the 1995 to the 2007 version. A systematic approach of memoing and constant comparison, which is based on the principles of Grounded Theory [22] was followed when developing the PRM, further details of which are to be found in [23]. The Process Reference Model of IEC 62304 was published in June 2014 as IEC TR 80002-3 [24].

While IEC 62304 describes only the software life cycle processes, additional processes should be in place for system development in the case where software is not embedded as part of an overall medical device. These additional processes were derived from ISO/IEC 12207:2008. Design and development related requirements from ISO 13485 and ISO 14971 were also added to the MDevSPICE® Process Reference Model. Both ISO 13485 and ISO 14971 are *de facto* standards for medical device software organizations. ISO 13485 requirements are primarily related to system level processes and ISO 14971 is concerned with risk management (and therefore aligned with the Software Risk Management process of the PRM.

The final MDevSPICE® PRM consists of 23 processes of which 10 are system life cycle processes, 8 are software life cycle processes and the remaining 5 support both the system and life cycle processes as can be seen in Figure 1.

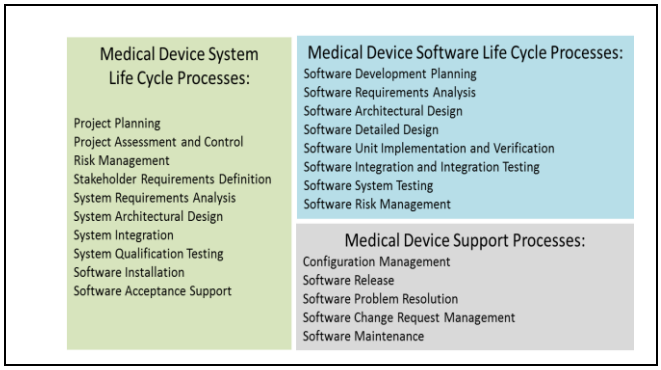

Figure 1. Processes of MDevSPICE® PRM

The MDevSPICE® PRM was then extended with additional elements to create a process assessment model (PAM). The aim of the MDevSPICE® PAM is to provide a comprehensive model for assessing the software and systems development processes against the widely recognized medical device regulations, standards and guidelines that a software development organization in the medical device domain has to adhere to. The MDevSPICE® PAM, similar to ISO/IEC 15504-5 (SPICE) [25], has two dimensions – a process dimension and a capability dimension. The process dimension lists three groups of processes from various models and standards, i.e., systems life cycle processes, software life cycle processes and support processes. Each process is described in terms of a Process Name, Process Purpose, Process Outcomes, Base Practices, Work Products and Work Product Characteristics.

The MDevSPICE® PRM is based on IEC 62304, ISO/IEC 12207:2008, ISO 14971 and ISO 13485. The MDevSPICE® PAM then extends this PRM with base practices and work products, some of the latter also being normative as they are described in IEC 62304, ISO 14971 or ISO 13485 as requirements. Where process outcomes are derived from ISO/IEC 12207:2008, their corresponding base practices and work products are derived from ISO/IEC 15504-5. Where process outcomes are derived from ISO 14971, their corresponding base practices are derived from IEC 80002-1. In addition to these sources, FDA guidance on premarket submissions, software validation and off-the-shelf software have been added to the informative base practices where the base practice did not already address the requirements of the corresponding FDA guidance. Product safety requirements have been added to the MDevSPICE® PAM from both IEC 60601-1 and IEC CD 82304-1, while

the usability engineering requirements have been incorporated from IEC 62366.

The capability dimension of the MDevSPICE® PAM is derived directly from ISO/IEC 15504 together with the Capability Levels, Process Attributes, Generic Practices, Generic Resources and Generic Work Products.

While integrating processes from different standards and guidance documents for the MDevSPICE® PRM and PAM , a focus on the traceability between and within system and software life cycle processes was maintained [26]. Both the FDA General Principles of Software Validation [20] and ISO/IEC 12207 [10] incorporate traceability of risks, changes and requirements throughout the development life cycle. This interaction and traceability of requirements is a key enabler of subsequent integration, and it has a vital role to play in raising the safety of medical device software.

B. MDevSPICE® Framework support for integration

The MDevSPICE® framework contains key facilities for integrating medical device software. Since MDevSPICE® is grounded in IEC 62304, the software sub system decomposition is consistent with the requirements of IEC 62304, meaning that the language of a *software unit*, a *software item* and a *software system* is adopted.

A software system is the integrated collection of software items to accomplish a specific function or set of functions; a software item is any identifiable part of a computer program; and a software unit is a software item that is not subdivided into other items. This software system hierarchy has an important role to play when a software developer wishes to decompose a system into parts of varying software safety classification. A benefit of such decomposition is that those parts of the software subsystem that are vital for safety (and which require additional safety activities when under development) can be isolated until they are later integrated with the other software components. It is also important that when the components are integrated that the safety implications are reflected in test cases that are pre-defined, then tested and the results are checked to ensure that they match the expected results. Otherwise sign-off cannot take place at the various levels – unit tests, integration tests and system tests.

Integration activities in the MDevSPICE® framework start by integrating software units into software items, and thereafter software items are further integrated with each other (and possibly with other units as well) into the software subsystem (which in turn is integrated into the overall medical device system). There are therefore several levels of integration and they must take into consideration the safety implications at each step. It is further the case that the bi-directional traceability of requirements (including requirements related to safety) from the product level right down to the individual software units is supported in MDevSPICE® thus further supporting medical device software safety at the integration stage and beyond.

C. Piloting the MDevSPICE® Framework

The MDevSPICE® framework has been validated in various stages of its development by different parties through both international expert review and industrial trials. The foundation of the MDevSPICE® PAM, IEC TR 80002-3 (the development of which was led by the authors), was published after several iterations of development and analysis by the standardization working group responsible for the publication of IEC 62304 (i.e., ISO/IEC Sub-Committee 62A, Joint Working Group 3). An international standard is published only after the national delegates of the standard's working group have agreed on every detail of that standard.

In addition to working with the international medical device standards community, the MDevSPICE® PAM has also been developed together with and analyzed by experts in process assessment working group 10 of ISO/IEC Joint Technical Committee 1, Sub-Committee 7, responsible for the development and maintenance of the series of process assessment standards. These standards are currently being revised from ISO/IEC 15504 series to ISO/IEC 330xx series of standards. MDevSPICE® framework keeps abreast of these updates as well as with the updates of any other standard and guidance document information from which is contained in the MDevSPICE® framework.

Upon successful completion of international expert review, the MDevSPICE® process assessment framework was then validated in the medical device software industry through pilot assessments over the past two years. MDevSPICE® process assessments were conducted in different types of organizations: (1) a small software company wishing to supply software to a large medical device manufacturer who wants them to demonstrate that they are capable of developing safe medical device software and provide the medical device manufacturer with a feeling that they will not jeopardize the safety of their overall medical device or the reputation of their organization; (2) three different assessments (across a 2 year period) were performed in two different international sites of a multinational medical device manufacturer who wants to ensure that they are incorporating best practices within their software development processes to not only achieve regulatory compliance but also reduce the likelihood of recalls through developing better quality and more robust software; (3) a software development company seeking to achieve regulatory compliance against IEC 62304 so that they can become medical device software suppliers; and (4) a large automotive manufacturer experienced in developing safety-critical embedded automotive software now wishing to also develop embedded medical device software.

IV. LESSONS LEARNED FROM PILOTING MDEVSPICE®

As a result of the MDevSPICE® pilot assessments we have witnessed different types of needs and challenges in companies where MDevSPICE® pilot assessments were conducted.

In companies that manufacture medical devices as well as develop the embedded software for their devices, the traceability and integration between system and software life cycle processes is well managed. This might be due to systems and software engineers working closely together for safe medical device development where the software developers are aware of the system risks and requirements.

For software companies that develop software for large medical device manufacturers though, it can be difficult for the third party software developers to become aware of the overall system level requirements and risks before software development project commences. When the system requirements are not provided to the software developers, this hinders the traceability engineering and integration of the subsystems of the medical device. But medical device manufacturers working on innovative devices are sometimes reluctant to provide their software subcontractors with the details of their device design as this could jeopardise device novelty or competitive advantage. Yet, the safety risks related to the performance of medical devices can outweigh the business risks, which can be diminished with proper legal knowhow, for the medical device manufacturer. We would therefore recommend medical device manufacturers to more openly communicate with their software subcontractors in order to best support risk and requirements management throughout their device design – even if this only encompasses those product requirements which are related to software requirements (and especially those which are safety related). The ultimate goal for all device providers is to have a safe medical device on the market and not risk liability or damage of their brand as a result of a recall of a faulty device.

V. CONCLUSION

Safety-critical domains are characterized by heavy regulatory demands that companies have to adhere to before they can place their devices on the market. Regulatory audits are conducted regularly to evaluate these companies and the safety of their devices. In order to pass these audits, medical device manufacturers have to ensure that all regulatory requirements have been adhered to in the design and development of each of the medical device subsystems.

In this paper, we have explained the medical device regulatory requirements and the related standards and guidance documents, and how MDevSPICE® addresses all of these concerns in a single medical device software framework. Key to developing this framework was an acknowledgement that the overall medical device requirements have a direct impact on the safety of the device, and it is therefore critical that top level product requirements are fully realized in the software system and its related requirements. This can be especially difficult to achieve in environments where device manufacturers may choose to outsource software development without necessarily sharing all top level product requirements subcontractors. To address this critical interface, the MDevSPICE® framework incorporates not just software development lifecycle processes but also system level process. Hence, system requirements that have an impact on software requirements are identified in MDevSPICE®, and through the implementation of bilateral requirements traceability, decisions taken during the software subsystem development are fed back to the top level system requirements – thus providing a closed loop for requirements management which can help to raise the overall safety of the device.

Requirements management is a key activity when integrating software subsystems and when integrating software into higher level systems (such as is the case for embedded medical device software). Closely aligned to requirements management is the management of safety related risks, and these too are supported in a bilateral top-to-bottom (system to subsystem) mechanism in MDevSPICE, with the result that software integration for medical devices is conducted in an environment that fully harmonises both general requirements and safety concerns. While such steps may not be desirable or economically viable in the case of general non-safety critical software, they do provide a mechanism for thorough requirements management, even in the case where subcontracting is undertaken – and this is a positive development in terms of supporting robust and effective software integration on all levels.

ACKNOWLEDGMENT

This research is supported by the Science Foundation Ireland 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 by Lero - the Irish Software Research Centre (http://www.lero.ie) grant 10/CE/I1855 & 13/RC/20194.

REFERENCES

[1] FDA. Chapter I - Food and drug administration, department of health and human services subchapter H - Medical devices, Part 820 - Quality system regulation. Available from: http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820. Last date accessed - 28th May 2015.

[2] Directive 93/42/EEC of the European Parliament and of the Council concerning medical devices. 1993. European Commission, Brussels, Belgium. pp. 43.

[3] Council directive 90/385/EEC on active implantable medical devices (AIMDD). 1990. Brussels, Belgium. pp. 35.

[4] Directive 98/79/EC of the european parliament and of the council of 27 october 1998 on in vitro diagnostic medical devices. 1998. Brussels, Belgium. pp. 43.

[5] Directive 2007/47/EC of the European Parliament and of the Council concerning medical devices. 2007. EC: Brussels, Belgium. pp. 35.

[6] IEC 62304: Medical Device Software - Software Life-Cycle Processes. 2006. IEC: Geneva, Switzerland. pp. 151.

[7] ISO 13485: Medical Devices - Quality Management Systems - Requirements for Regulatory Purposes. 2003. ISO: Geneva, Switzerland. pp. 57.

[8] ISO 9001:2000 - Quality Management Systems - Requirements. 2000. Geneva, Switzerland. pp. 27.

[9] ISO/IEC 12207:1995 - Information Technology - Software Life-Cycle Processes. 1995. ISO/IEC: Geneva, Switzerland. pp. 106.

[10] ISO/IEC 12207:2008 - Systems and Software Engineering - Software life cycle processes. 2008. ISO/IEC: Geneva, Switzerland. pp. 138.

[11] ISO/IEC 15504-2. 2004. Information technology - Software process assessment - A reference model for processes and process capability, in 15504.
[12] ISO 14971 - Medical Devices - Application of Risk Management to Medical Devices 2009. ISO: Geneva, Switzerland. pp. 82.
[13] IEC TR 80002-1 - Medical Device Software - Part 1: Guidance on the Application of ISO 14971 to Medical Device Software. 2009. IEC: Geneva, Switzerland. pp. 58.
[14] IEC 60601-1 - Medical electrical equipment – Part 1: General requirements for basic safety and essential performance 2005. IEC: Geneva, Switzerland. pp. 20.
[15] IEC 62366 - Medical devices - Application of usability engineering to medical devices. 2007. IEC: Geneva, Switzerland. pp. 104.
[16] IEC 82304-1: Health software -- Part 1: General requirements for product safety. 2012. IEC: Geneva, Switzerland. pp. 30.
[17] FDA. 2015. Guidance Documents (Medical Devices and Radiation-Emitting Products).
[18] FDA Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. 2005. FDA: USA. pp. 20.
[19] FDA's Guidance for industry, FDA reviewers and compliance on - Off-The-Shelf Software Use in Medical Devices. 1999. FDA: USA. pp. 26.
[20] FDA's General Principles of Software Validation; Final Guidance for Industry and FDA Staff. 2002. FDA: USA. pp. 43.
[21] ISO/IEC 24774 - Systems and Software Engineering - Life Cycle Management - Guidelines for Process Description. 2010. Geneva, Switzerland. pp. 15.
[22] B. Glaser, and A. Strauss 1976. The Discovery of Grounded Theory: Strategies for Qualitative Research, ed. A.d. Gruyter. Hawthorne, NY, USA.
[23] M. Lepmets,P. Clarke, F. McCaffery, A. Finnegan, and A. Dorling 2014. Development of a Process Assessment Model for Medical Device Software Development, in Industrial Proceedings of the 21st EuroSPI Conference: Luxembourg. p. 2.25-2.35.
[24] IEC TR 80002-3: Medical device software -- Part 3: Process reference model of medical device software life cycle processes (IEC 62304). 2014. IEC: Geneva, Switzerland. pp. 23.
[25] ISO/IEC 15504-5. Information technology - process assessment - Part 5: an exemplar process assessment model. 2012. pp. 211.
[26] G. Regan, M. Biro, F. Mc Caffery, K. McDaid, and D. Flood 2014. A Traceability Process Assessment Model for the Medical Device Domain, in EuroSPI 2014, Springer: Luxembourg. p. 206-216.