Author Query Form

ENCYCLOPEDIA OF SOFTWARE ENGINEERING (ESE)

ARTICLE: 120046037

Q1	P: 1	Column 1	Line 28	The keyword “regulated software development” is not present in the entry. Please check.
Q2	P: 3	Column 2	Line 89	Please provide the expansion for “CE,” if applicable.
Q3	P: 7	Column 1	Line 12	Both “RSKM” and “RM” are provided as abbreviations for “Risk Management.” Please consider making usage consistent.
Q4	P: 13	Column 1	Line 26	The link “http://www.fda.gov/cdrh/ode/guidance/337.pdf” throws an error when typed on a browser’s address bar. Please provide a working link.
Q5	P: 11	Column 1	Line 10	Please check figure placement.

Software Process Improvement in the Medical Device Industry

Fergal McCaffery
Regulated Software Research Group, Dundalk Institute of Technology, Dundalk, Ireland

John Burton
Vitalograph Ireland Ltd., Ennis, Ireland

Valentine Casey
School of Design Engineering and Computing, Bournemouth University, Poole, U.K.

Alec Dorling
InterSPICE Ltd., Cambridge, U.K.

Abstract
This entry considers medical device software development that takes place in a regulated environment. Research has shown that medical device regulations cannot be completely satisfied by generic software process improvement models. The development of a software process assessment model (Medi SPICE) specifically for the medical device industry through extending relevant practices from ISO/IEC 15504-5 is presented. Medi SPICE consists of a Process Reference Model (PRM) and a Process Assessment Model (PAM). The Medi SPICE Process Assessment Model can be used to perform conformant assessments of the software process capability of medical device suppliers in accordance with the requirements of ISO/IEC 15504-2. The Medi SPICE Process Assessment Model is based on ISO/IEC 15504-5 but can also provide coverage of additional software development practices that are required to achieve regulatory compliance within the medical device industry.

INTRODUCTION

Medical device companies must comply with the regulatory requirements of the countries in which they wish to sell their devices. Compliance requirements stipulate that the manufacturers must produce a design history file detailing the software components and processes undertaken in the development of their medical devices. Due to the safety-critical nature of medical device software it is important that highly effective software development practices are in place within medical device companies. Although guidance exists from regulatory bodies on what software activities must be performed, no specific method for performing these activities is outlined or enforced.

This entry highlights the need for a Software Process Improvement (SPI) model within the medical device industry (Medi SPICE) and draws upon the call for specific software development standards to be developed for the medical device industry so that companies adhering to such standards will have a more streamlined pathway toward regulatory compliance. Medi SPICE has the aim of minimizing the volume of software documentation content within the premarket submission to the Food and Drug Administration (FDA) for audit and to provide global harmonization (with consistent guidance provided for all medical device software manufacture). The results of a Medi SPICE assessment may be used to indicate the state of a medical device suppliers software practices in relation to the regulatory requirements of the industry, and identify areas for process improvement. The results of these assessments may also be used as a criterion for supplier selection. The authors believe that, with the publication of the Medi SPICE Process Reference and Process Assessment Models, more specific guidance will be available for the basis of process design and assessment in the medical device industry. We describe the development of Medi SPICE based upon applicable processes from the ISO/IEC 15504-5[1] model. The ISO/IEC 15504-5 model is used as a foundation upon which to develop this model.

This entry is structured as follows: the "What Are Medical Devices?" section provides definitions of what a medical device is; the "What Is Medical Device Regulation?" section explains medical device regulation; and the "Medical Device Software Development and Associated Standards" section details the evolution of medical device software regulation. The "Issues with Existing Medical Device Software Development Standards and Guidance" section describes the issues that the medical device software standards present in terms of developing software for the industry. The next sections

Encyclopedia of Software Engineering DOI: 10.1081/E-ESE-120046037

focus on SPI. The "Formal SPI Models—CMMI® and SPICE™," section describes the formal SPI reference models and the "Specific SPI Model for the Medical Device Industry" section describes how they fail to provide sufficient coverage for the safety-critical Space, Automotive, and Medical Device industries and describes how a medical device-specific SPI model called Medi SPICE has been developed for the medical device industry. The objectives, structure, content, and the delivery phases of Medi SPICE are outlined and explained. The "Example: Medi SPICE Process—Risk Management" section contains an example of a Medi SPICE process and the entry concludes with the "Conclusions and Future Work" section.

WHAT ARE MEDICAL DEVICES?

On June 14, 1993 the Council of the European Communities published the Council Directive 93/42/EEC (1993) concerning medical devices. The directive is also known as the Medical Device Directive (MDD). The MDD is intended to ensure the safety of medical devices placed on the market of the European Union (EU) in a harmonized fashion, and has the backing of national legislation in member states. Amendments to this directive occurred via Directive 2000/70/EC (2000), 2001/104/EC (2001), 2003/32/EC (2003), and 2007/47/EC (2007).

Traditionally, the definition for a medical device has been limited to the context of "any instrument, apparatus, appliance, material or other article whether used alone or in combination, including the software necessary for its proper application intended by the manufacturer to be used for human beings…" (Council Directive 93/42/EEC). However, since the original definition of a medical device, software products have become increasingly prevalent as stand-alone devices and are increasingly being used alone for medical purposes. The Council of the European Community has recognized that the traditional definition for a medical device does not adequately address software, which as a stand-alone product, can also be used as a medical device and have therefore revised their statements as follows:

> Stand alone software is considered to be an active medical device. It is necessary to clarify that software in its own right, when specifically intended by the manufacturer to be used for one or more of the medical purposes set out in the definition of a medical device, is a medical device. Software for general purposes when used in a healthcare setting is not a medical device. (Council Directive 2007/47/EC)

To formally account for how software is being used in a stand-alone context as a medical device, the Council amended their definition of a medical device in the MDD as follows:

"Medical Device" means 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, intended by the manufacturer to be used for human beings for the purpose of:

> - diagnosis, prevention, monitoring, treatment or alleviation of disease,
> - diagnosis, monitoring, treatment, alleviation of or compensation for an injury or handicap,
> - investigation, replacement or modification of the anatomy or of a physiological process,
> - control of conception, and which does not achieve its principal intended action in or on the human body by pharmacological, immunological or metabolic means, but which may be assisted in its function by such means. (Council Directive 2007/47/EC)

The "growing importance of software in the field of medical devices, be it as stand alone or as software incorporated in a device" requires that the "validation of software [must be] in accordance with the state of the art should be an essential requirement" (Council Directive 2007/47/EC). Therefore, the following statement has also been added to the MDD: "Taking account of the growing importance of software in the field of medical devices, be it as stand alone or as software incorporated in a device, validation of software in accordance with the state of the art should be an essential requirement." The Council Directive 2007/47/EC was adopted by the European Parliament on March 29, 2007 and will become mandatory on March 21, 2010.

WHAT IS MEDICAL DEVICE REGULATION?

Medical devices and associated software are developed with the intention to increase the well-being of patients. However, if not properly controlled, some medical devices may have the potential to cause harm to operators, subjects, bystanders, or the environment. Devices may fail to perform an intended function, or there is a risk that users may attempt to misuse a function of the software in a manner not intended by the manufacturer—giving rise to risk of injury or even death. Additionally, the medical device industry and global governments are faced with the illegal sale of substandard devices that fail to meet minimum quality and safety standards—putting further lives at risk. Therefore, the medical device industry and governments are faced with the challenge of ensuring that only safe and effective devices are produced and placed on the market.

To tackle these issues, governments have put in place regulatory bodies whose job is to define regulatory systems for medical devices and to ensure that only safe medical devices are placed on the market. A safe device is one that cannot cause serious injury to a patient or end user of the device. To aid the control of medical devices, regulatory bodies adopt a classification scheme. Devices are classified against a predetermined set of criteria established by the regulatory body. The manufacturer of the device must establish design controls in line with the classification level. The higher the classification of the device, the more stringent the design controls and constraints. The classification levels range from class I-type devices (least risk) to class III devices (highest risk). Class I (lower risk) devices are subject to the least regulatory constraints and controls and are exempt from premarket submissions. They are, however, subject to general controls including—but not limited to—registration with the relevant regulatory body and the maintenance of a good manufacturing process. Class II devices must implement additional special controls with class III devices requiring premarket approval.

Before a medical device company can sell its product in a country, it must follow the registration or licensing procedure of that country. This, in turn, establishes a contract between the device manufacturer and that particular country. The device manufacturer is obliged to establish and perform both premarket and postmarket duties as defined in the quality system regulations. The quality system requirements for Europe are defined in the ANSI/AAMI/ISO 13485 standard (2003) and the requirements for the United States are defined in the 21 CFR Part 820 Quality System Regulations (2007). The quality system has been defined by the FDA[2] as a system "for the design, manufacture, packaging, labelling, storage, installation, and servicing of all finished devices intended for human use." Applicable requirements are typically directly related to the class of the device.

The regulatory or approved body, through its audits, checks conformance to the quality system requirements periodically. By conforming to the quality system requirements, the device manufacturer is proactive and demonstrates a tightly controlled manufacturing system, one that, in turn, provides for greater reliability, safety, and effectiveness of their devices.

MEDICAL DEVICE SOFTWARE DEVELOPMENT AND ASSOCIATED STANDARDS

Medical device companies are responsible for ensuring that they take adequate precautions to produce safe and efficient software—software that does not pose a severe hazard should a software-related failure occur, i.e., ensure that the software is not a potential source of harm to the operator, subject, bystander, or environment. One important issue facing medical device companies producing software is that it is not always feasible to test all paths of execution through a software application. Therefore, the testing process cannot be relied upon as the only indicator of software quality.[3] A simple change in a software component can cause unforeseen problems in other components within the system, problems that can go undetected unless a robust software design and implementation process exists. The safety of medical device software is dependent upon the manufacturer having a reliable and solid software process[4] with risk management a core practice for the production of safe and efficient software.

A primary issue with the quality system regulations is that while they state the requirements for specific activities to be established and performed, they do not provide details in relation to how these activities should be actually performed. Requirements are stipulated at a high level, by stating, for example, that manufacturers must "establish quality system procedures and instructions." From a software perspective, "software specification, software validation, and risk analysis" must be included "as part of the design process." "Each manufacturer shall maintain device master records" that include "device specifications including appropriate drawings, composition, formulation, component specifications, and software specifications" and "design validation shall include software validation and risk analysis, where appropriate."[2] In an attempt to address the vagueness of the regulatory requirements, the FDA published separate regulatory guidance documents for required software activities.[5–7] In Europe, many organizations rely on the regulatory guidance documents from the FDA due to insufficient guidance being provided for the equivalent CE marking process. Additionally, there has been a steady progression in the medical device software sector with the release of new and updated standards in an attempt to address the knowledge gap that exists between the high-level regulatory requirements and the low-level detail and knowledge required to adequately satisfy those requirements.

The following section discusses the progression of medical device software-related standards over the years, how they have evolved, and why they have evolved in the manner that they did[4] and expands on the standards discussed where appropriate.

Software Quality Audit System (1996)

In September 1996, the National Institutes of Health (NIH) held a workshop aimed at creating a formal policy for regulating medical device software such that safer devices could be produced and the number of unnecessary injuries and deaths caused by medical devices each year would be reduced. The aim of the workshop was to establish a Software Quality Audit (SQA) system and procedure, which manufacturers could follow to gain market approval and without undergoing the traditional premarket software

review. If manufacturers followed the specified FDA-approved audit process and had no negative or serious findings, they would then be allowed to market their device in the United States. A group known as *AdvaMed* commenced work on the identification of relevant software standards and associated practices. They also initiated a program to define a suitable audit process.

IEC 60601-1-4 (1996)

In May 1996, the International Electrotechnical Commission (IEC) first published the *Programmable Electrical Medical Systems*, IEC 60601-1-4 Collateral Standard (1996). This standard was to define a risk management process along with development activities. It was proposed that medical device companies that followed the standard and that could produce evidence to that effect would be seen to produce safer software.

The FDA had concerns in relation to the publication of the IEC 60601-1-4 standard. They were not satisfied that the standard sufficiently covered all of the relevant software process areas. They viewed the standard as more a risk management standard than as an all-encompassing software standard. Therefore, they rejected it as a software standard, but indicated that it was a useful risk management standard, one that medical device companies could benefit from following. The FDA's primary concerns were as follows:

1. The standard was written primarily as a risk management standard.
2. It did not adequately address software engineering.
3. It focused only on the design portion of a product's life cycle. It did not address the maintenance or retirement of the product.
4. It did not address all of the devices regulated by the FDA, specifically stand-alone software.

The AAMI Software Committee (1997–1998)

In response to the FDA's concerns over the IEC 60601-1-4 standard, the Association for the Advancement of Medical Instrumentation (AAMI) software committee immediately set about generating a new software standard. The foundation of this new standard would be the IEC 60601-1-4 standard. The AAMI sought to expand the IEC 60601-1-4 standard by incorporating those software engineering practices that the FDA believed were missing. Unfortunately, the whole process came to a halt for several reasons including the following:

- There was failure among committee members to establish a consensus of opinion.
- At the time, there was no established method for the Center for Devices and Radiological Health (CDRH) to use standards in their regulatory process. This was prior to the FDA Modernization Act (FDAMA), which was passed by the U.S. Congress in 1998.
- There was no traceability to existing IEEE software standards or IEC/ISO software engineering standards. To prove successful and robust, the AAMI had to prove ties to existing and established software engineering standards.

The FDA Modernization Act of 1997

The FDA Modernization Act, which was signed into law in November 1997, was a significant step in the regulation of medical software devices. The act advocates the use of standards in the device review process. In implementing the legislation, the FDA published a list of standards in the Federal Register to which manufacturers could declare their conformity. The purpose of the FDAMA Act was to save the agency and the manufacturers considerable time and resources by reducing the regulatory obstacles to entry in domestic and international markets.

Once the FDAMA was signed into law, the CDRH immediately commenced the establishment of Standards Technical Groups (STGs), including an STG for software. The software STG was initially responsible for evaluating existing software engineering standards and recommending suitable standards for recognition in the premarket approval process. However, during their evaluation of the standards, the STG found that the standards were too broad in scope and did not contain sufficient detail for the production of safe, reliable software. The group therefore set about providing their own guidance. Their first three major steps included the following:

1. Categorizing all of the standards reviewed into four different groups as described below
2. Establishing links between existing standards and the premarket approval process
3. Recognizing the ISO/IEC 12207 software engineering standard (1995) for general use

According to Eagles and Murray,[4] the STG established categories for the software standards that included the following:

- General process standards, which are technology independent (e.g., ISO 9000 standards).
- General process standards, which are technology specific (e.g., ISO/IEC 12207, IEC 60601-1-4, ANSI/AAMI SW68).

- Specific process implementations for the general software engineering processes (IEEE software standards).
- Specific product requirements for particular types of medical device.

Once the standards had been reviewed, accepted, and categorized, the STG defined a link between the recognized standards and the regulatory process. Manufacturers who can prove conformance with recognized standards could reduce the amount of paperwork required in their submission for premarket approval. It is important to note that the level of documentation required during the submission is directly linked to the levels of concern posed by the software-containing device. The "*level of concern*" has been defined by the CDRH as "*an estimate of the severity of injury that a device could permit or inflict, either directly or indirectly, on a patient or operator as a result of device failures, design flaws, or simply by virtue of employing the device for its intended use*."[7]

As part of the software STG work, a decision was made to recognize the ISO/IEC 12207 software engineering standard for general use. The standard was believed to contain an accurate and relevant description of terms as related to medical device software development.

Medical Device Software Development and the AAMI Software Committee (1997–2001)

The AAMI software committee carefully reviewed the ISO/IEC 12207 standard and decided it was necessary to create a new standard that would be tailored specifically for medical device software development. The decision by the AMMI to create a medical device-specific standard was due to a number of shortfalls they found in the existing standard. Specifically, the existing ISO/IEC 12207 standard required major changes for those companies that already had existing software processes in place. The standard did not account for off-the-shelf (OTS) software requirements. It did, however, contain various sections that were seen as relevant in the medical device software domain.

The AAMI did not discard the work done with the ISO/IEC 12207 standard because, as outlined, it was accepted as containing a good set of well-defined terms. The AAMI used the ISO/IEC 12207 standard as the foundation for their new standard *AAMI SW68, Medical device software—Software life cycle processes*, which is discussed in detail below. The AAMI removed objectionable sections and added their own new ones. They added specific requirements for OTS. They also added specific requirements for software that can cause safety hazards. This included defining the relationship between software engineering and risk management as outlined in the standard ISO 14971 (2007).

AAMI SW68—Software Life Cycle Processes for Medical Device Software

The AAMI SW68 standard (2001) is a standard for the software life cycle processes for medical device software. AAMI SW68 defines two major life cycle processes, i.e.,—a development process and a maintenance process. This standard was produced with both application software and embedded software in mind. Where a medical device comprises software or is used in conjunction with software, the standard considers the software to be a subsystem of the medical device itself.

AAMI SW68 separates compliance with the standard into two separate categories that include—compliance for software that can cause hazards or that controls risk (Major/Moderate level of concern) and/or compliance for software that cannot cause a hazard and that does not control risk (Minor level of concern).

ANSI/AAMI/IEC 62340—Medical Device Software—Software Life Cycle Processes

In 2006, the IEC released a new standard IEC 62304 (2006) that was based on the AAMI SW68 standard. The standard was the output of a collaborative effort between

- Subcommittee (SC) 62A, "*Common aspects of electrical equipment use in medical practice*"
- IEC Technical Committee (TC) 62, "*Electrical equipment in medical practice*"
- ISO TC 210, "*Quality management and corresponding general aspects for medical devices*"
- ISO/IEC JTC 1/SC 7, "*Software and system engineering*"

The AAMI reviewed the standard during its development and adopted it on its first release. The major difference between this IEC 62304 and AAMI SW68 with respect to software risk management is that the standard introduces the concept of software classification. Three software classes are defined:

- 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

In adopting this standard, medical device companies producing software must assign a particular class to each software system. All software units within a system inherit the systems safety class until they themselves are analyzed and reclassified. The decision to break a software item into new software items and reclassify them into a lower safety class must be part of the risk management process and documented in the risk management file. All software development processes and the requirement to perform

activities within the development processes are then linked directly to the software system class.

ISO 14971

The ISO 14971 standard has become the de facto standard on risk management for medical devices since its inception. The first release of the standard was in 2000 with the second edition being released in 2007. In the United States, the FDA regulators receive formal training in ISO 14971 as part of their software risk management training. The FDA recognize the standard and agree compliance with it as acceptable for premarket submissions in the United States. In the EU, conformance with the standard is also acceptable for meeting the requirements of the medical directive.[8] In Japan, ISO 14971 has become the Japanese Industrial Standard.

The standard specifies procedures and activities for identifying hazards in medical devices and their accessories (including software). Within its Annex section, it also provides specific consideration for in vitro medical devices and toxicological hazards.

GAMP 5

The GAMP forum is a subcommittee of the International Society of Pharmaceutical Engineering (ISPE). This forum was established in the 1990s by a number of representatives from the pharmaceutical industry. Regulatory guidance with respect to automated systems in pharmaceutical manufacturing was already in existence. However, it was felt that the process for regulatory inspection in this area was less mature than in other areas of inspection; so too was the industry's interpretation of the requirements. In 2001, the GAMP forum produced a guide incorporating good automated manufacturing practices with the aim of satisfying the regulatory requirements of that time. The Guide's aim is to provide

A Risk-Based Approach to Compliant GxP Computerized Systems through the provision of pragmatic and practical industry guidance to achieve compliant computerized systems fit for intended use in an efficient and effective manner. The first draft of GAMP was made available in March 1994.

ISSUES WITH EXISTING MEDICAL DEVICE SOFTWARE DEVELOPMENT STANDARDS AND GUIDANCE

Medical device companies must comply with the medical device regulations stipulated by regulatory bodies governing the country in which they wish to market their device. The medical device companies must be able to produce sufficient evidence to support their claims of compliance. To this end, in the United States, the CDRH has published guidance papers for industry and medical device staff that include risk-based activities to be performed during software validation,[6] premarket submission,[7] and when using OTS software in a medical device.[5] Although the CDRH guidance documents provide information on which software activities should be performed, they do not enforce any specific method for performing these activities. Much of the guidance provided is ambiguous and does not provide details on how software activities should be performed. This information is spread across various regulatory guidance papers, industry guidance papers, standards, and technical implementation reports. The obvious implication of this is that medical device manufacturers could fail to comply with the expected requirements. This view would appear to be supported by an observation made by Nadia Perreault, medical device technical manager for National Quality Assurance USA, who, in relation to software risk management, has been quoted as saying "a lot of places typically look at risk management only at the design and development function and they don't carry it through the entire lifecycle of the product or process... people think that it is just a little snippet in time during the design and development phase."[9] There has not been any comparison or link made from the medical device industry to the various existing (and proven) SPI models. This is evident in the way in which the standards for medical devices have independently evolved.

The remainder of the entry will describe current SPI models, how SPI models have been developed for other safety-critical industries, and the development of an SPI model for the medical device industry.

FORMAL SPI MODELS—CMMI® AND SPICE™

CMMI® Overview

The U.S. Department of Defense recognized a key risk in software projects with respect to increased software costs and quality issues. Therefore, in the early 1980s, they set about establishing the Software Engineering Institute (SEI), located in Carnegie Mellon University, Pennsylvania. The SEI (1991) immediately set about developing a formal SPI model for software engineering. In August 1991, the SEI released the initial version of the Capability Maturity Model for Software (SW-CMM). In 1997, efforts in the CMM model halted in preference for the more comprehensive Capability Maturity Model Integration (CMMI). Version 1.1 of the CMMI was released in 2002 by the SEI CMMI Product Team[10] with v1.2 following in August 2006.[11]

The CMMI model comprises 22 key process areas. A process area as defined by CMMI is "a cluster of related practices in an area that, when performed collectively, satisfy a set of goals considered important for making

significant improvement in that area."[11] A process area typically comprises 1–4 specific goals. The specific goals are further divided into practices that are specific to that process area. In addition, the model contains generic goals and practices that are common across all of the process areas. Capability levels are used to measure performance with respect to the specific and generic practices. The levels include Level 0—Incomplete, Level 1—Initial, Level 2—Repeatable, Level 3—Defined, Level 4—Managed, and Level 5—Optimized.

One of the key process areas addressed by CMMI is the area of Risk Management (RSKM). Risk Management appears as a Project Management process area at Maturity Level 3 with the three specific goals:

- SG 1 Prepare for Risk Management
- SP 1.1-1 Determine Risk Sources and Categories
- SP 1.2-1 Define Risk Parameters
- SP 1.3-1 Establish a Risk Management Strategy
- SG 2 Identify and Analyze Risks
- SP 2.1-1 Identify Risks
- SP 2.2-1 Evaluate, Categorize, and Prioritize Risks
- SG 3 Mitigate Risks
- SP 3.1-1 Develop Risk Mitigation Plans
- SP 3.2-1 Implement Risk Mitigation Plans

Each of the specific practices listed above is expanded into a number of subpractices in the CMMI model. The specific goals and specific practices described by the CMMI are a common approach to risk management irrespective of the industry in question. However, it is in the subpractices of the specific practices where differences may be required depending on the industry in question and whether there are any specific regulations governing that industry.

ISO/IEC 15504 Overview

ISO/IEC 15504 is the international standard for software process assessment and was developed by a Joint Technical Subcommittee of ISO and IEC. The group was formed in 1993. ISO/IEC 15504 is derived from the ISO/IEC 12207 software life cycle processes standard. ISO/IEC 15504 is published as a full international standard under the general title Information Technology—Process Assessment. It consists of the following parts:

Part 1—Concepts and vocabulary
Part 2—Performing an assessment
Part 3—Guidance on performing an assessment
Part 4—Guidance on use for process improvement and process capability determination
Part 5—An exemplar software life cycle process assessment model
Part 6—An exemplar systems life cycle process assessment model
Part 7—Assessment of organizational maturity
Part 8—An exemplar process assessment model for IT service management life cycle processes (under development)
Part 9—Target capability profiles (under development)
Part 10—Safety Extensions (under development)

ISO/IEC 15504 is used in defining and assessing an organization's capability in the areas of management and the definition of their process structures. The model is therefore broken into a process and capability dimension. The key process categories include customer–supplier, engineering, supporting, management, and organization. Similar to CMMI, the capability level for the process areas are broken into six levels, 0—5 measured as follows: Level 0—Incomplete process, Level 1—Performed process, Level 2—Managed process, Level 3—Established process, Level 4—Predictable process, and Level 5—Optimized process.

It should also be noted that a decision has been made to replace ISO/IEC 15504 with a new series of standards and this was approved at an International Standards Organization meeting in Hyderabad, India on May 29, 2009. This new series of standards will be numbered as an ISO/IEC 33001–33099 series. Work on the new 33001 series has commenced. However, the overall standards development will take several years to complete.

SPECIFIC SPI MODEL FOR THE MEDICAL DEVICE INDUSTRY

With the creation of formal SPI models such as CMMI and ISO/IEC 15504, researchers within regulated environments such as the Space and Automotive industries started to investigate how they could utilize these models to improve the practices within their industry domains. However, they discovered that although the existing models are comprehensive in their own right, neither CMMI nor ISO/IEC 15504 addressed all of the regulatory needs and constraints of their industries. Researchers therefore sought to adopt the practices within these models while also expanding on them to account for regulatory requirements within their own domains. This resulted in the production of full SPI models tailored specifically for the Space domain[12] and Automobile domain—Automotive SPICE.[13]

The authors of this entry have published a number of research papers that have investigated the gaps that exist within CMMI and ISO/IEC 15504 in relation to the practices that are required for medical device regulatory compliance for the processes of risk management and configuration management.[14,15] The SEI has published a white paper that provides a high-level description of how the CMMI model does not provide sufficient coverage of

medical device software with particular reference to the IEC 62304 standard.[16]

What Is Medi SPICE?

The authors have launched Medi SPICE, which is a software process assessment model for the medical device industry,[17] that is based upon ISO/IEC 15504-5 and provides coverage of the medical device regulatory device software regulations (with a particular focus upon the IEC 62304 standard). Medi SPICE, like ISO/IEC 15504 and Automotive SPICE™, contains both a Process Reference Model (PRM) and Process Assessment Model (PAM) containing processes that provide comprehensive coverage of the FDA and European Council guidelines, and associated standards (e.g., ISO 14971, IEC 60601-1-4, IEC 62304, TIR 32, and GAMP) for the complete software development life cycle. As safety is a primary issue for medical device software, the PRM and PAM incorporate the following:

- The safety integrity levels and the safety life cycle from the international standard for the *functional safety* of electrical, electronic, and programmable electronic equipment (IEC 61508).
- The safety processes that are present in +SAFE.[18] These are included in part 10 of ISO/IEC 15504.

The scope of Part 10 is to develop a Safety Extension that defines additional processes and guidance to support the use of the exemplar process assessment models for systems and software (ISO/IEC 15504 Parts 5 and 6) when applied to the assessment of safety-related systems in order to make consistent judgments regarding process capability and/or improvement priorities.

Medi SPICE Objective

The overall objective of Medi SPICE is to provide a conformity assessment scheme to support first-, second-, or third-party assessment results that may be recognized by the regulatory bodies. The PRM and PAM of the Medi SPICE assessment standard are derived from ISO/IEC 15504-5 processes as they are all applicable to the development of safety-critical medical device software. As the IEC 62304 standard contains the medical device software life cycle processes that have to be adhered to in order to achieve medical device regulatory compliance, a key objective is to provide coverage of all processes that are either included in or referenced from IEC 62304.

Medi SPICE Coverage

The Medi SPICE PRM and PAM are being released in phases and once complete, will consist of a defined set of software processes that will contain base practices that when utilized will assist medical device software development organizations to fulfil the regulatory guidelines and standards of the medical device industry. Medi SPICE will cover the complete medical device software development and maintenance life cycle. The Medi SPICE PAM will provide guidance in relation to assessing the software engineering capability of processes within a medical device software development organization, which will be conformant with the ISO/IEC 15504-2[19] requirements for a PAM. Medi SPICE will be based upon integrating defined IEC 62304 processes into relevant ISO/IEC 15504-5 processes to enable FDA guidelines to be fulfilled. Additionally, we will also incorporate the safety processes from +SAFE and various other medical device-related standards into relevant processes, e.g., ISO 14971 into the risk management process.

Food and Drug Administration

The FDA request documentation in relation to the following 11 software development areas are the following:

1. Level of Concern
2. Software Description
3. Device Hazard and Risk Analysis
4. Software Requirements Specification
5. Architecture Design
6. Design Specifications
7. Requirements Traceability Analysis
8. Development
9. Validation, Verification, and Testing
10. Revision Level History
11. Unresolved Anomalies

IEC 62304

The IEC 62304 Medical device software—Software life cycle processes standard defines a number of process requirements that should be adhered to, which are as follows:

Quality Management System,
Software Safety Classification,
Software Development Process Group, containing:

Software Development; Planning, Software Requirements Analysis; Software Architectural Design; Software Detailed Design; Software Unit Implementation & Verification; Software Integration & Integration Testing; Software System Testing; Software Release

Software Maintenance Process Group, containing:

Establish Software Maintenance Plan; Problem & Modification Analysis; Modification Implementation

Risk Management Process Group, containing:

Analysis of software contributing to hazardous situations; Risk Control measures; Verification of risk control measures; Risk management of software changes

Software Configuration Management Process Group, containing:

Configuration Identification; Change Control; Configuration status accounting

Software Problem Resolution Process Group, containing:

Prepare problem reports; investigate the Problem; Advise Relevant Parties; Use change control process; Maintain Records; Analyse problems for trends; Verify software problem resolution; Test documentation contents

Medi SPICE Processes

Medi SPICE has been developed to instil effective practices into the software development processes of medical device companies. This is an attempt to improve the effectiveness and efficiency of software practices used by medical device companies through investigating the mapping between relevant ISO/IEC 15504-5 processes, the 11 FDA software development areas, and the five process area groups defined in IEC 62304. At present ISO/IEC 15504-5 consists of 48 processes; however, it should be noted that three new safety-based processes will be added in Part 10 of ISO/IEC 15504.

All 48 ISO/IEC 15504-5 processes are applicable to the development of safety-critical medical device software and may be included in Medi SPICE. However, in order to progress Medi SPICE a small number of pilot processes were initially developed. This set of processes will continue to be expanded until Medi SPICE contains a complete set of processes that fulfil all the associated regulatory standards and guidelines. The approach taken was to agree upon a base set of processes that will satisfy the 11 software development areas defined by the FDA, +SAFE processes, the European Council, and the main process groups (i.e., Software Development, Software Maintenance, Software Risk Management, Software Configuration Management, Software Problem Resolution) that are defined in IEC 62304.

The Medi SPICE PRM contains a process name, a purpose, and outcomes for each of the processes. In addition to this information the PAM also contains the following information for each process:

- The set of base practices required to accomplish the process purpose and fulfil the process outcomes
- Sample output work products
- Characteristics associated with each work product

Table 1 illustrates the result of performing a high-level mapping of requested FDA premarket submission areas and IEC 62304 processes against existing ISO/IEC 15504-5 processes. As a result of performing this mapping it was clear that there would not be exact one-to-one mappings from an FDA area to both an IEC 62304 process and an ISO/IEC 15504-5 process. Table 1 groups together processes with the most significant overlaps. However, it should be noted that other less significant overlaps may also occur across groups. The table includes the 11 software areas that are defined by the FDA; these are mapped against 14 processes that are defined in IEC 62304 plus the two +SAFE processes, which are in turn mapped against 23 ISO/IEC 15504-5 processes and three additional safety-related processes (that will be included in ISO/IEC 15504 Part 10)—i.e., Safety Management, Safety Engineering & Selection, and qualification of software tools and libraries. It should be noted that while other ISO/IEC 15504-5 processes are relevant for developing medical device software, the 23 ISO/IEC 15504-5 processes listed are the most relevant (from the set of 48 processes). These processes will be extended with medical device-specific content to include safety integrity levels and to satisfy the associated medical device requirements included in both the FDA guidelines and ISO/IEC 62304.

The Medi SPICE authors initially considered developing Medi SPICE to solely focus upon the medical device regulations and to use the IEC 62304 processes as the foundation. However, the aim is for medical device companies not only to develop software that will adhere to the regulatory guidelines but also to encourage the adoption of medical device software development processes that will lead to the development of safer and more reliable medical device software. This will best be achieved through following established software engineering practices such as those documented in ISO/IEC 15504-5.

From inspection of Table 1, it may be recognized that 16 of the associated ISO/IEC 15504-5 processes appear in bold and are labeled with an "A" and the remaining 10 processes (in italics) are labeled with a "B." The "A" processes are ISO/IEC 15504-5 processes of which a high proportion of their defined base practices will be required either in their current state or in a revised state to satisfy the regulatory demands of the medical device industry. The "B" processes are ISO/IEC 15504-5 processes that contain a much smaller proportion of defined base practices that will be required either in their current state or in a revised state to satisfy the regulatory demands of the medical device industry.

The 16 "A" processes are Risk Management; Safety Management (to be included in Part 10); Safety Engineering (to be included in Part 10); Selection and qualification of software tools and libraries (to be included in Part 10); Software Requirements Analysis; Project Management; Software Design; Software Construction; Software Integration; Software Testing; Verification;

Table 1 Mapping FDA software areas and IEC 62304 processes against ISO/IEC 15504-5.

FDA areas	Associated IEC 62304 processes	Associated ISO/IEC 15504-5 processes
Risk and Safety Management Group		
1. Device Hazard and Risk Analysis	1. Risk Management	**A1. Risk Management**
2. Level of Concern	2. Software Safety Classification	**A2. Safety Management (Part 10 Safety Extensions)**
		A3. Safety Engineering (Part 10 Safety Extensions)
+SAFE—A Safety extension to CMMI CMMI Process Areas		
• Safety Management		
• Safety Engineering		
Requirements Group		
3. Software Requirements Specification	3. Software Requirements Analysis	*B.1 Requirements Elicitation*
4. Requirements Traceability Analysis		**A.4 Software Requirements Analysis**
		B.2 System Requirements Analysis
Development Group		
5. Architecture Design Chart	4. Software Development Planning	**A.5 Project Management**
6. Design Specifications	5. Software Architectural Design	**A.6. Selection and qualification of software tools and libraries (Part 10 Safety Extensions)**
7. Software Description	6. Software Detailed Design	*B.3 System Architectural Design*
8. Software Development Environment Description		**A.7 Software Design**
		B.4 Documentation
Testing and Integration Group		
9. Validation, Verification and Testing	7. Software Unit Implementation and Verification	**A.8 Software Construction**
	8. Software Integration and Integration Testing	**A.9 Software Integration**
	9. Software System Testing	**A.10 Software Testing**
		B.5 System Integration
		B.6 System Testing
		A.11 Verification
		A.12 Validation
Supporting Processes		
10. Revision Level History	10. Software Release	*B.7 Product Release*
11. Unresolved Anomalies	11. Software Configuration Management	*B.8 Product Acceptance Support*
	12. Software Maintenance	*B.9 Software Installation*
	13. Software Problem Resolution	**A.13 Configuration Management**
		A.14 Problem Resolution Management
		A.15 Change Request Management
		A.16 Software and System Maintenance
Additional Requirements		
	14. Quality Management System	ISO 13485
		B.10 Quality Assurance

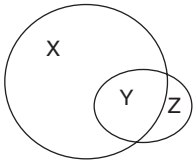

Fig. 1 Composition of Medi SPICE processes.

Validation; Configuration Management; Problem Resolution Management; Change Request Management; and Software and System Maintenance.

The 10 B processes are Requirements Elicitation; System Requirements Analysis; System Architectural Design; Documentation; System Integration; Product Release; Product Acceptance Support; System Testing; System Installation; and Quality Assurance.

Composition of Medi SPICE Processes

Like ISO/IEC 15504, each of the Medi SPICE processes consist of a purpose, a number of outcomes, and a number of base practices that will have to be performed in order to fulfil the outcomes. The performance of the base practices provides an indication of the extent of achievement of the process purpose and process outcomes. Work products are used, produced, or both, when performing the process.[1] The composition of the Medi SPICE processes is illustrated in Fig. 1.

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.

Medi SPICE highlights what additional base practices and outcomes have to be added to the associated ISO/IEC 15504-5 processes in order to satisfy medical device regulations (Z), as well as any ISO/IEC 15504-5 outcomes and associated base practices that are not required in order to satisfy medical device regulatory requirements (X). Due to the scale of the entire Medi SPICE model the remainder of this entry will present a summary of the risk management process as this is a very important process in relation to the development of safety-critical software for the medical device industry.

EXAMPLE: MEDI SPICE PROCESS—RISK MANAGEMENT

This illustrates an example of a Medi SPICE process—Risk Management (RM). The Medi SPICE RM process seeks to combine the various guidelines and standards within the medical device industry. It does so in the context of the following regulations: ISO 14971, SW68, IEC 62304, TIR32 and GAMP 4/GAMP 5, and FDA guidance documents. The Medi SPICE RM process is an extension of the ISO/IEC 15504-5 RM process that is specifically tailored to fulfil the RM regulations of the medical device software industry. The RM process may then be adopted by medical device companies to improve their software development practices by providing them with a process that will ensure that their hazard analysis and risk control procedures satisfy the current medical device regulations and guidelines.

This section briefly summarizes a mapping of the medical device standards and guidelines against the ISO/IEC 15504-5 RM process. The Medi SPICE RM process is developed through mapping regulatory medical device practices against the seven base practices specified in ISO/IEC 15504-5 for RM. These are as follows:

BP1: Establish risk management scope
BP2: Define risk management strategies
BP3: Identify risks
BP4: Analyze risks
BP5: Define and perform risk treatment actions
BP6: Monitor risks
BP7: Take preventative or corrective action

Within Medi SPICE, each base practice consists of one or more base subpractices. The following is an example of the *BP1: Establish risk management scope* base practice within Medi SPICE. The aim of this practice is to determine the scope of the RM to be performed. Both ISO/IEC 15504-5 and the medical device regulations specify that the RM scope should be defined. Additionally, the medical standards specify that the strategy should include the life cycle phases for which the strategy is applicable (see Table 2). In the following table a new base practice that was added is represented in bold italics.

Table 3 provides a high-level summary of the number of additional subpractices that had to be added to the ISO/IEC 15504-5 RM process to achieve the Medi SPICE objectives. The table illustrates that there are 31 subpractices within

Table 2 Medi SPICE RM base practices for establishing RM scope.

Base practice no	Description	Specified in ISO/IEC 15504-5?	Specified in the medical device regulations?
Man.5.BP1	Determine the scope of risk management to be performed	**Yes**	**Yes**
Man.5.BP1a	***Define the scope of the strategy and include those life-cycle phases for which the strategy is applicable***	***No***	***Yes—ISO 14971 Section 3.4(a)***

Table 3 Summary of Medi SPICE RM process.

Practice	ISO/IEC 15504-5 subpractices	ISO/IEC 15504-5 subpractices required to meet regulatory medical device requirements	Additional subpractices required to meet regulatory medical device requirements
BP1: Establish risk management scope	1	1	1
BP2: Define risk management strategies	1	1	11
BP3: Identify risks	2	1	3
BP4: Analyze risks	1	1	0
BP5: Define and perform risk treatment actions	1	1	5
BP6: Monitor risks	1	1	2
BP7: Take preventative or corrective action	1	1	1
Total	8	7	23

Medi SPICE RM. Thirty of these base practices are required within the medical device industry whereas only eight are required within the ISO/IEC 15504-5 RM process. Only one of the seven ISO/IEC 15504-5 base practices fully satisfies the requirements of the associated Medi SPICE RM practice (i.e., BP 4: Analyze Risks). Therefore, the mappings highlight that Medi SPICE needs to be more comprehensive in its coverage of the RM process than ISO/IEC 15504-5 in order to satisfy the regulatory requirements of the medical device industry.

With respect to the practices of the Medi SPICE RM process, we discovered that following the base practices of the ISO/IEC 15504-5 RM process will at best, only partially meet the regulatory requirements of the medical device industry in relation to RM. For RM, the existing ISO/IEC 15504-5 specification of outcomes and base practices can be carried over, with the extension mentioned above into the Medi SPICE framework.

CONCLUSIONS AND FUTURE WORK

This entry has described the area of SPI for the medical device industry. The initial sections in the entry focused upon the background of medical devices and described what medical devices are and how the industry is heavily regulated. The next sections explained the evolution of software-related medical device standards and the issues that these standards present in terms of developing software for the industry. The second half of the entry then focused upon SPI, initially describing the formal SPI reference models and then how they did not provide sufficient coverage for the safety-critical Space, Automotive, and Medical Device industries. The authors then described how a medical device-specific SPI model called Medi SPICE has been developed for the medical device industry. The objectives, structure, content, and the delivery phases of Medi SPICE were then outlined. Finally, an example was provided of the Medi SPICE risk management process area as this is a key process within the medical device industry and therefore requires comprehensive coverage.

Medi SPICE has been designed to be a framework that will encourage medical device companies to distance themselves from the concept of developing the software first and then completing the necessary documentation that is required to achieve FDA compliance. Instead the objective is to pursue a continuous SPI path that will produce more efficient software development and safer medical devices. Ongoing research is being carried out in all the relevant areas to fine tune and develop the Medi SPICE model further.

ACKNOWLEDGMENTS

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, and Lero—the Irish Software Engineering Research Centre (http://www.lero.ie).

REFERENCES

1. ISO/IEC 15504, Information Technology—Process Assessment—Part 5: An exemplar Process Assessment Model, ISO/IEC JTC1/SC7, International Standards Organization, 2006. http://www.iso.org/iso/iso_catalogue/

catalogue_tc/catalogue_detail.htm?csnumber=38934 (accessed June 2009).
2. Code of Federal Regulations, Title 21, Volume 8, Revised April 1, 2009 http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1.1.12.1 (accessed November 2009).
3. Lee, I.; Pappas, G.J.; Cleaveland, R.; Hatcliff, J.; Krogh, B.H.; Lee, P.; Rubin, H.; Sha, L. High-confidence medical device software and systems. IEEE Comp. April **2006**, *39* (4), 33–38.
4. Eagles, S.; Murray, J. *Medical Device Software Standards: Vision and Status, Medical Device and Diagnostic Industry*, May 2001, http://www.devicelink.com/mddi/archive/01/05/002.html (accessed February 2008).
5. FDA/CDRH Guidance Document. *Guidance for Off-the-Shelf Software Use in Medical Devices*. September 1999. http://www.fda.gov/cdrh/ode/guidance/585.pdf (accessed November 2009).
6. CDRH. 2002, *General Principles of Software Validation; Final Guidance for Industry and Medical Device Staff*, January 11, 2002. http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm085281.htm (accessed November 2009).
7. FDA/CDRH Guidance Document. *Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices*. May 2005, http://www.fda.gov/cdrh/ode/guidance/337.pdf
8. Rudolph, H. *Do We Need Medical Device Risk Management Certification? Medical Device & Diagnostic Industry*. 2003. http://www.devicelink.com/mddi/archive/03/11/001.html (accessed November 2009).
9. Wichelecki, S. *Understanding ISO 13485*, Quality Magazine, January 2, 2008, http://www.qualitymag.com/CDA/Articles/Feature_Article/BNP_GUID_9-5-2006_A_10000000000000225133 (accessed November 2009).
10. Software Engineering Institute. CMMI Product Team. *Capability Maturity Model Integration (CMMI) for Software Engineering*, Version 1.1, Continuous Representation (CMMI-SW, V1.1, Continuous), Carnegie Mellon University Software Engineering Institute, Technical Report CMU/SEI-2002-TR-028, Version 1.1, 2002, http://www.sei.cmu.edu/publications/documents/02. reports/02tr028.html (accessed November 2009).
11. Software Engineering Institute. CMMI Product Team. *Capability Maturity Model® Integration for Development*, Carnegie Mellon University Software Engineering Institute, Technical Report CMU/SEI-2006-TR-008, Version 1.2, 2006, http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html (accessed November 2009).
12. Cass, A.; Völcker, C.; Dorling, A.; Winzer, L.; Carranza. J.M. *SPiCE for SPACE: A Process Assessment and Improvement Method for Space Software Development*, ESA bulletin 107, 2001. www.esa.int/esapub/bulletin/bullet107/bul107_14.pdf (accessed November 2009).
13. Automotive SIG, 2008, The SPICE User Group, *Automotive SPICE™ Process Reference Model*, http://www.automotive-spice.com/web/download.html (accessed November 2009).
14. McCaffery, F.; Burton, J.; Richardson, I. Development and Evaluation of a Risk Management Capability Model (RMCM) for the Medical Device Industry. In Proceedings of SPICE 2008 Conference, Nuremberg, Germany, May 2008.
15. McCaffery, F; O'Connor, R; Coleman, G., Mapping Medical Device Guidelines against the CMMI for the Configuration Management, Software and Data Technologies. In First International Conference, ICSOFT 2006, Setúbal, Portugal, September 11–14, 2006, Revised Selected Papers Series: Communications in Computer and Information Science, Vol. 10, Filipe, J., Shishkov, B., Helfert, M. Eds.; 2008; XIII, 389 p.
16. Walker, D.W. CMMI and Medical Device Engineering September 29, 2009, Software Engineering Institute White paper, 2009 http://www.sei.cmu.edu/library/abstracts/whitepapers/cmmi-medical.cfm (accessed November 2009).
17. McCaffery, F.; Dorling, A. Medi SPICE: An Overview. In Proceedings of SPICE 2009 Conference, Turku, Finland, June 2–4, 2009.
18. +SAFE, V1.2: *A Safety Extension to CMMI-DEV, V1.2*, Defence Material Organisation, Australian Department of Defence, March 2007, Software Engineering Institute, TECHNICAL NOTE CMU/SEI-2007-TN-006 http://www.sei.cmu.edu/pub/documents/07.reports/07tn006.pdf (accessed June 2009).
19. ISO/IEC 15504, Information Technology—Process Assessment—Part 2: Performing an Assessment, ISO/IEC JTC1/SC7, International Standards Organization, 2003. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=37458 (accessed November 2009).