Understanding the Relationships Within the Medi SPICE Framework

Derek Flood, Fergal Mc Caffery, Valentine Casey
Dundalk Institute of Technology,
Dundalk, Ireland
{derek.flood, fergal.mccaffery, val.casey} @dkit.ie

***Abstract*— Regulated domains, such as medical device software development, require organisations to have specific processes in place in order to secure regulatory approval. Software process improvement initiatives, such as Medi SPICE, help organisations to improve their process in conformance with these regulations. These initiatives, however, do not specify how an organisation implements these processes, instead detailing what the organisation must implement. This work proposes the development of a series of roadmaps that will guide an organisation through the implementation of the required processes in a regulatory compliant manner. This paper presents the first step towards achieving this aim, which involves an investigation of the dependencies between the base practices defined in Medi SPICE in order to ensure that the produced roadmaps form a complete software development process in line with regulatory requirements. The paper describes two complementary approaches, a structured representation and a graphical representation, to representing the links between practices in the Medi SPICE framework.**

Keywords-Software Process; Medical Device Regulation; Software Process Improvement Roadmaps.

I. Introduction

Advancements in technology have allowed medical practitioners to provide a greater level of care to patients by offering a wider range of treatment options. However, when technology is used, there is a risk to the patient if that device should fail. For this reason, strict regulations must be followed during the design and development of medical device software. In order for an organisation to market medical devices they must comply with the regulatory requirements of the country in which the device is to be sold [1]. For example, an organisation wishing to market new medical devices, unlike anything on the market, within the US must first submit a pre-market submission to the Food and Drug Administration (FDA) for approval prior to the distribution of the medical device. If a similar medical device is already on the market, then the medical device organisation must submit a 510k application. One exception to this is Medical Device Data Systems (MDDS), which do not require pre-approval, but must have been developed using defined development processes and have a Quality Management System (QMS) in place [2].

Increasingly, software is becoming a more important component of medical devices. This is partially due to its flexibility and its ability to enable complex changes to be made to the medical device, without the need for changes to the hardware [3] and also due to the fact that standalone software in its own right may also be considered a medical device [4]. Consequently, this increase in the proportion of software within medical devices has resulted in increased medical device software complexity [5].

In order to assist organisations improve their processes to meet regulatory compliance, Medi SPICE [6] (a medical device specific software process improvement framework) provides organisations with the goals of the required processes and a number of base practices that must be implemented in order to achieve these goals.

The Medi SPICE framework is divided into a number of processes each detailing a different aspect of the software development process. However, there are a number of dependencies between these processes making it more difficult to focus upon individual processes in isolation. This work aims to identify these dependencies through an analysis of the base practices defined within Medi SPICE both internally within individual processes and externally across different processes.

Upon obtaining a detailed understanding of these dependencies a series of roadmaps may then be developed that will guide organisations through the implementation and improvement of their medical device software development processes in an efficient manner.

In this paper, we detail the process used for the identification of these links and how the representation scheme that has been used will allow for validation upon the completed roadmaps. In addition, the paper outlines the types of relationships that were identified in the Medi SPICE framework and provides examples of each type of relationship.

The paper is structured as follows: Section II outlines the importance of medical device software. Section III introduces the Medi SPICE framework. Section IV outlines the objectives of this research. Section V describes how the relationships in Medi SPICE were modelled using both a human readable and machine readable representation. Section VI discusses how these representations will be used during the construction of a series of process roadmaps to guide organisations through the implementation of the necessary standards for developing medical device software. Section VII contains our conclusions for this research.

II. Medical Device Software

Software is playing an increasingly integral part in medical devices and is now included in approximately 50% of the medical devices available for sale in the US [7]. Consequently, generic software development organisations are now becoming medical device software organisations both due to the software development opportunities within this domain and also because their software development applications may now be classified as medical devices if they meet the Medical Device Directive's (MDD's) definition of a medical device [8]. The MDD defines a medical device as

> *"any instrument, apparatus, appliance, material or other article, whether used alone or in combination, including software 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."*

This means that software development organisations creating applications which meet this definition must now conform to the same regulatory requirements as traditional medical device manufacturers.

Therefore, organisations that are new to medical device software development must be aware of the relevant regulations that are applicable to the medical device domain within the particular region they wish to market their device [9]. Medical devices marketed in the US must comply with the FDA regulations, while devices to be marketed within the European Union (EU) must conform to the regulations set out by the European Council.

As part of these regulations [2], a QMS must be in place during the design, development, delivery, installation and servicing of medical devices. The QMS ensures that high quality processes are used through-out the entire product lifecycle and that adequate documentation is maintained for review by the appropriate authority.

To guide these organisations a number of regulations and standards have been produced by the relevant regulatory authorities. In the EU, the *ISO 13485- Medical Devices - Quality management systems – Requirements for regulatory purposes* [10], has been produced outlining the main requirements of a QMS. Similarly, the FDA has produced the FDA 21 CFR Part 820 Quality Systems Regulations (QSR) [11].

III. MEDI-SPICE

Despite the regulatory bodies outlining the necessary regulations, standards, technical reports and guidance documents for medical device software development, they do not provide specific methods for performing the required activities to achieve regulatory approval. This often leads to medical device organizations becoming compliance centric in their approach to software development. As a result, there has been very limited adoption of software process improvement within the medical device domain [12]. Previously, this was not a critical issue due to the limited proportion of software contained within medical devices, but this is no longer the case. Today, there is a particular requirement for highly effective and efficient software development processes to facilitate medical device software development [13].

Existing generic Software Process Improvement (SPI) models are available, which include the Capability Maturity Model Integration (CMMI®) [14] and ISO 15504-5:2006 [15] (SPICE), but these models were not developed to provide sufficient coverage of all the areas required to achieve medical device regulatory compliance [16]. To address the requirement for a medical device software process assessment and improvement model the Regulated Software Research Group at Dundalk Institute of Technology undertook extensive research in this area [6] [17]. This initiated the development of Medi SPICE, a medical device specific SPI framework, which is being developed in collaboration with the SPICE User Group [19].

The objective of undertaking a Medi SPICE assessment is to determine the state of a medical device organisation's software processes and practices. Medi SPICE is an integration of the regulatory requirements of the medical device industry and software engineering best practice [14]. It can also be used as part of the supplier selection process when an organisation wishes to outsource or offshore part or all of their medical device software development to a third party or remote division [16].

Medi SPICE is based upon the latest version of ISO/IEC 15504-5 (currently under ballot) and ISO/IEC 12207:2008 [17]. It is being developed in line with the requirements of ISO/IEC 15504-2:2003 [18] and contains a Process Reference Model (PRM) and Process Assessment Model (PAM). It also incorporates the requirements of the relevant medical device regulations, standards, technical reports and guidance documents.

The Medi SPICE PRM consists of 42 processes and 15 subprocesses which are fundamental to the development of regulatory compliant medical device software. Each process has a clearly defined purpose and outcomes that must be accomplished to achieve that purpose.

Medi SPICE also contains a PAM, which is based upon the PRM, which forms the basis for collecting evidence that may be used for rating the 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 and purpose as defined in the PRM; this is termed the process dimension. In the second dimension, the PAM describes capabilities that relate to the process capability levels and process attributes, this is termed the capability dimension.

IV. Research Objectives

The aim of this research is to understand and identify the relationships between the base-practices defined within the Medi SPICE PAM. In order to achieve this aim, two research questions (RQs) were constructed to examine the relationships between base-practices both within individual processes and across different processes.

- RQ1: What relationships exist between base-practices in each process included within the Medi SPICE framework?
- RQ2: What relationships exist across processes of the Medi SPICE framework?

RQ1 was posed to examine each process in isolation to determine the relationships that exist between the base practices.

In contrast to RQ1, RQ2 examined the relationships between the processes by identifying base-practices that are dependent upon base practices in other processes. For example Fig. 1 shows the relationship between Eng1, which details a process for obtaining stakeholder requirements, and ENG2, which defines the system requirements analysis process. It can be seen that, before establishing the system requirements (*ENG2.BP1*), an organisation must first agree on the requirements with stakeholders (*ENG1.BP7*).

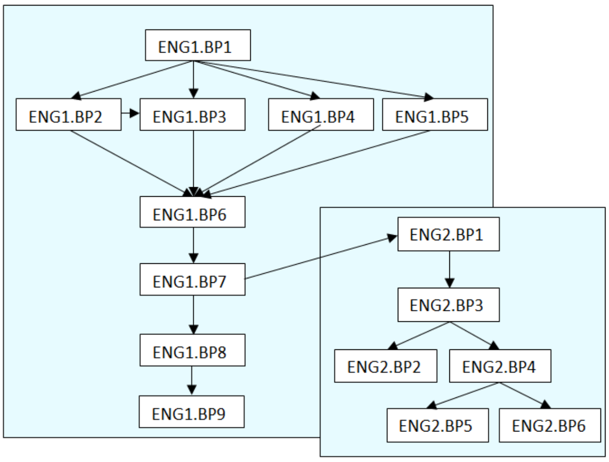

Figure 1. Across Process Relationship

In order to answer the research questions posed above, an analysis of the Medi SPICE PAM was performed. The base practices in each process were examined and the relationships between the practices were determined. The identified relationships were then independently validated by the authors of Medi SPICE. The identified relationships were represented using both a human readable (graphical) representation and a machine readable (structured) representation (XML).

V. Representation of Links

Once the links were identified between the practices, they were represented in two ways. To aid the understanding of the relationships between practices in each process, a graphical representation of each process was produced. In addition to this, a machine readable structured representation was also produced to allow for a quick identification of practice dependencies.

A. Human readable representation

As one of the aims of this work was to understand how the base practices in the Medi SPICE framework relate to one another, a human readable representation of each process was created.

In this representation, each practice is represented as a rectangle and the links between them are represented as an arrow pointing to the depending process. It can be seen in Fig. 2 that there is an arrow pointing from *AGR1B.BP1* to *AGR1A.BP2*. This means that base practice 2 *(AGR1B.BP2)* is dependent upon base practice 1 *(AGR1B.BP1)*.

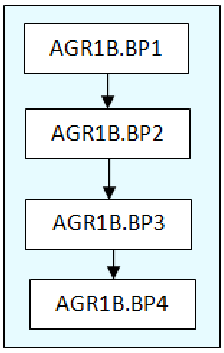

Figure 2. Human-Readable Visualisation

In this representation, it was decided to use the full process ID to help users distinguish between practices of different processes when the graph is used to represent a relationship between multiple processes.

The nature of the dependencies between the base practices usually stems from the need of information to pass from one base practice to another. For this reason, the dependency graphs designed during this work were produced to replicate the information flow between the base practices.

B. Structured representation

In addition to the visual representation, it was necessary to produce a machine readable format that could be used during the production of the roadmaps to identify all base practices necessary to meet those required by the standards.

It was decided to use a custom XML schema to represent the links as most languages provide support for reading in XML files. An example process is presented below.

```
<process title="Acquisition Preparation"
id="AGR1A">
    <basePractice
    id="AGR1A.BP1">Establish the
    need</basePractice>
    <basePractice id="AGR1A.BP2">Define
    the requirement</basePractice>
    <basePractice id="AGR1A.BP3">Review
    Requirements</basePractice>
    <basePractice id="AGR1A.BP4">Develop
    Acquisition strategy</basePractice>
    <basePractice id="AGR1A.BP5">Define
    selection criteria</basePractice>
    <basePractice
    id="AGR1A.BP6">Communicate the
    need</basePractice>

    <InProcessLink PID="AGR1A.BP2"
    dependantOn="AGR1A.BP1"/>
    <InProcessLink PID="AGR1A.BP3"
    dependantOn="AGR1A.BP2"/>
    <InProcessLink PID="AGR1A.BP4"
    dependantOn="AGR1A.BP3"/>
    <InProcessLink PID="AGR1A.BP5"
    dependantOn="AGR1A.BP4"/>
    <InProcessLink PID="AGR1A.BP6"
    dependantOn="AGR1A.BP5"/>
    <ExProcessLink PID="AGR1A.BP5"
    DependantOn="AGR1B.BP1"
    type="equivalent"/>
</process>
```

Each process is comprised of four tags; <Process/>, <BasePractice/>, <InProcessLink>, and <ExProcessLink>. The <Process> tag represents a process in the Medi SPICE framework and includes two attributes; the title of the process and the ID used to identify the process within the Medi SPICE framework. All other tags are nested within the <Process/> tag.

The <BasePractice/> tag is used to represent the base practices within the process. There are between 3 and 18 base practices within each process. Each base practice is comprised of an ID and the title of the practice.

The <InProcessLink/> tag represents a link between two practices within a single process. The tag contains two attributes; the first attribute identifies the practice which is dependent upon another practice and is given the attribute name *PID* while the second attribute identifies the practice which is depended upon, known as *dependantOn*.

The final tag is used to represent external links. This tag, titled *<ExProcessLink/>*, contains three attributes. The first two attributes are the same as those used within the *<InProcessLink/>; PID* and *dependantOn*, while the third attribute, titled *type*, denotes the class of link that exists between the practices. A detailed examination identified three types of links within the Medi SPICE framework: *breakdown*, *equivalent*, and *dependent*.

In some cases, a sub process is used to implement a base practice in another process. For example, *AGR1.BP3* defines the practice "Select Supplier" while sub process *ARG1B* defines the base practices that should be used to select a supplier, as illustrated in Fig. 3. This type of link is known as *breakdown*, as this type of link breaks down one practice into multiple base practices.

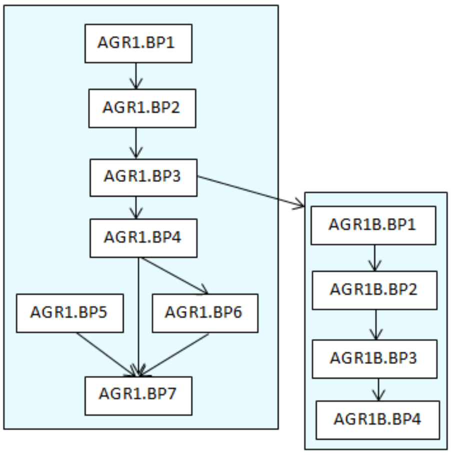

Figure 3. Between Process Link of type *breakdown*

In addition, some practices are semantically equivalent to practices in other process areas. For example, *AGR1A.BP5* is to "Define the selection criteria" while *ARG1B.BP1* states "Establish supplier selection criteria". Although the terminology is different between the two practices the underlying meaning is the same. This is depicted in Fig. 4. In this type of relationship, the type attribute is given a value of *equivalent*.

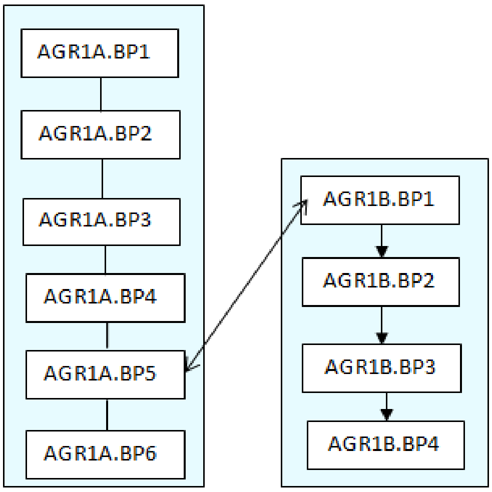

Figure 4. Between Process Link of type *equivalent*

The value given to the type attribute in the final class of relationship is *dependant*. In this case, a base practice must be performed before a subsequent base practice can be implemented. For example, the stakeholder requirements should be established *(ENG1.BP7)* before the establishment of the system requirements *(ENG2.BP2),* as illustrated in Fig. 1.

VI. Using the dependency graph

The next phase of this work will be to identify the base practices that are necessary to fulfil the requirements of multiple medical device software development regulations and standards such as ISO 13485 and ISO 14971. These standards define the requirements that are necessary to secure regulatory approval in order to sell medical devices.

Before a medical device can be marketed in the US, it may be required to first secure premarket approval from the FDA. To assist medical device organisations the FDA have produced a document entitled "Guidance for the content of Premarket Submissions for Software Contained in Medical Devices" in which they outline what is required in order to prepare a premarket submission.

In this document, the FDA state that an organisation must implement a QMS in order to sell their devices on the market in the US. This requirement is also necessary for sales within the EU. The International Organization for Standardisation (ISO) have produced the International standard *ISO 13485 – Medical devices - Quality management systems – Requirements for regulatory purposes* which details the requirements for a QMS. The FDA has produced a similar regulation, but have said, recently, that QMSs compliant with the ISO 13485 would also be acceptable.

In order to assist organisations to implement a quality management system, the base practices necessary to fulfil the requirements of the ISO 13485 will be identified through a thorough examination of the standard. Subsequently, through the use of the dependency graph, described above, all supporting base practices necessary to implement the identified base practices will be identified.

Using these base practices, a software process improvement roadmap will be developed that will guide an organisation through the implementation of a QMS. Each of the base practices will be grouped into one of three phases; Planning phase, SDLC phase and On-Going activities phase.

The planning phase occurs at the beginning of a medical device software development project. During this phase the organisation will define the lifecycle that will be used during the project and define strategies for a number of activities performed during the development of the medical device software. This phase will also include the definition of the quality objectives and the assignment of responsibility for the QMS to a member of the management team.

The second phase is performed during the development of the software. During this phase, the practices necessary to be compliant with the QMS are defined. The practices in this phase relate to the systems development lifecycle and include activities such as requirements analysis, design, and testing.

The ISO 13485 standard requires that a number of activities need to be performed at regular intervals during the development process. These activities do not belong to a single phase of the lifecycle but can occur during any phase of the development process. The practices belonging to these activities will be placed in the third phase of the roadmap, the On-Going activities phase. Examples of this type of activity are quality assurance activities, risk assessment activities and problem resolution activities.

In addition, this phase also includes an optional process that may be required during the development of a medical device software system, namely, Acquisition. It may be necessary for a medical device software organisation to acquire components that will be used in the produced medical device software. To assist these organisations the roadmap will include an optional process that will guide the organisation through the acquisition of the necessary components.

When the practices have been assigned to each of the three groups described above, they will then be sub-divided into steps that will allow the organisation to implement them in a sequential manner.

The dependency graphs described in this paper will play an important role in validating the proposed roadmaps. In addition to identifying necessary practices, the dependency graphs will also help to ensure that the activities are performed in the correct order. Using the machine readable format, each practice in a step will be validated to ensure that it does not depend on a step that is performed at a subsequent step.

VII. Conclusion

Medical device software is required to be developed and maintained through following high quality processes during the construction and distribution of the software. Depending upon the region in which the software is to be sold, local regulations must be adhered to in order to secure approval for sale. The Medi SPICE framework has been developed to assist medical device software organisations improve the quality of their processes.

This work complements the Medi SPICE framework through the development of a series of SPI roadmaps that medical device organisations can use to guide their software improvement activities. An important first step in this work has been the identification of the relationships that exist between base practices within the Medi SPICE framework. These relationships have been modelled in both a human and machine readable format allowing for quick analysis of these relationships during the creation of the roadmaps.

VIII. 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 10/CE/I1855.

IX. References

[1] Burton, J., McCaffery, F., and Richardson, I., A risk management capability model for use in medical device companies. in International Workshop on Software quality (WoSQ '06). 2006. Shanghai, China: ACM

[2] US FDA Center for Devices and Radiological Health, Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. 2005, CDRH: Rockville.

[3] Lee, I., Pappas, G., Cleaveland, R., Hatcliff, J., Krogh, B., Lee, P. , Rubin, H. , and Sha, L., High-Confidence Medical Device Software And Systems. Computer, 2006. 39(4): pp. 33 - 38.

[4] European Council, Council Directive 2007/47/EC (Amendment). 2007, Official Journal of The European Union: Luxembourg.

[5] Rakitin, R., Coping with defective software in medical devices. Computer, 2006. 39 (4): pp. 40 - 45.

[6] McCaffery, F., Dorling, A., “Medi SPICE Development”, Software Process Maintenance and Evolution: Improvement and Practice Journal. Volume 22 Issue 4, pp. 255 – 268 http://www3.interscience.wiley.com/journal/122580316/abstrac

[7] Faris, T. H., (2006) “Safe and Sound Software: Creating an Efficient and Effective Quality System for Software Medical Device Organizations” ASQ Quality Press, 2006

[8] Mc Hugh, M., Mc Caffery, F. & Casey, V. 2011. Standalone Software as an Active Medical Device In: O'CONNOR ET AL (ed.) The 11th International SPICE Conference Process Improvement and Capability dEtermination. Dublin: Springer.

[9] Ge, X., Paige, R. F. & McDermid, J. A. 2010. An Iterative Approach for Development of Safety-Critical Software and Safety Arguments. Agile 2010.

[10] ISO 13485:2003 (2003) Medical devices — Quality management systems — Requirements for regulatory purposes. Second ed. Geneva, Switzerland, ISO.

[11] FDA 2007. Title 21--Food and Drugs Chapter I --Food and Drug Administration Department of Health and Human Services subchapter h--Medical Devices part 820 Quality System Regulation. U.S. Department of Health and Human Services.

[12] Denger, C., Feldmann, R., Host, M., Lindholm, C. & Shull, F. (2007) A Snapshot of the State of Practice in Software Development for Medical Devices. First International Symposium on Empirical Software Engineering and Measurement. Madrid, Spain.

[13] McCaffery, F., Burton, J., Casey, V., and Dorling, A., (2010) Software Process Improvement in the Medical Device Industry, in Encyclopedia of Software Engineering, P. Laplante, Editor. 2010, CRC Press Francis Taylor Group: New York. pp. 528 - 540.

[14] CMMI Product Team (2006) Capability Maturity Model® Integration for Development Version 1.2. Software Engineering Institute, Pittsburch PA.

[15] ISO/IEC 15504-5:2006 (2006) Information technology - Process Assessment - Part 5: An Exemplar Process Assessment Model. Geneva, Switzerland, ISO.

[16] Casey, V. (2010) Virtual Software Team Project Management. Journal of the Brazilian Computer Society, 16, pp 83 – 96.

[17] ISO/IEC 12207:2008 (2008) Systems and software engineering - Software life cycle processes. Geneva, Switzerland, ISO.

[18] ISO/IEC 15504-2 (2003) - Software engineering — Process assessment — Part 2: Performing an assessment. 2003: Geneva, Switzerland.

[19] Spice user Group, http://www.spiceusergroup.org/, accessed 31/09/2012