



A software process improvement lifecycle framework for the medical device industry

McCaffery, F., McFall, D., Donnelly, P., Wilkie, G., & Sterritt, R. (2005). A software process improvement lifecycle framework for the medical device industry. In *Unknown Host Publication* (pp. 273-280). IEEE. <https://doi.org/10.1109/ECBS.2005.13>

[Link to publication record in Ulster University Research Portal](#)

Published in:
Unknown Host Publication

Publication Status:
Published (in print/issue): 01/04/2005

DOI:
[10.1109/ECBS.2005.13](https://doi.org/10.1109/ECBS.2005.13)

Document Version
Publisher's PDF, also known as Version of record

General rights
Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact pure-support@ulster.ac.uk.

A Software Process Improvement Lifecycle Framework for the Medical Device Industry

F. McCaffery¹, D. McFall¹, P. Donnelly², F.G. Wilkie¹ & R. Sterritt¹
¹Centre for Software Process Technologies, Jordanstown, Northern Ireland.
²IQ Solutions, Castlewelgan, Northern Ireland.

Email: f.mccaffery@ulster.ac.uk, d.mcfall@ulster.ac.uk, peter@iqsolution.com,
fg.wilkie@ulster.ac.uk, r.sterritt@ulster.ac.uk

Abstract

This paper describes a software process improvement (SPI) framework to ensure regulatory compliance for the software developed in medical devices. The software framework introduced here (known as MedeSPI – Medical Devices Software Process Improvement) will address an opportunity to integrate the regulatory issues and process improvement mechanisms in order to achieve improved software processes.

1. Introduction.

Software is becoming an increasingly important aspect of medical devices and medical device regulation. Medical devices can only be marketed if compliance and approval from the appropriate regulatory bodies of the Food and Drug Administration (FDA) [1] (US requirement), and the European Commission under its Medical Device Directives (MDD) [2] (CE marking requirement) is achieved.

Integrated into the design process of medical devices, is the requirement of the production and maintenance of a device *technical file*, incorporating a design history file. Design history illustrates the well documented, defined and controlled processes and outputs, undertaken in the development of medical devices and for our particular consideration with this framework - the *software* components.

Many medical device manufacturers, like businesses in most industries, have based the core of their quality system on the ISO9000:2000 [3] family of standards. The internal quality system of each medical device company has then been enhanced to ensure compliance with additional standards that relate specifically to the nature of their products and the market they address. ISO 13485:2003, Medical devices - Quality management systems - Requirements for regulatory purposes [4], for instance, is based on quality management system

requirements currently contained in medical device regulations as well as those appropriate requirements contained in ISO 9001:2000.

A recent study carried out by IQ Solutions with the medical device companies in N. Ireland has revealed that the software development process has been predominately based on the need to comply with the FDA and European MDD regulations. The software processes have not been of primary focus, only that the elements are in place that satisfy the regulatory requirements.

It is believed that a software process improvement roadmap which incorporates the goals of medical device software for meeting regulatory compliance would greatly enhance the design control procedures currently identified within company quality systems. Quality software is defined as software that meets its functional and non-functional requirements without lengthy rework, including regulatory compliance, without any inconsistencies. Reducing time to market is often based on continuous improvement of processes implemented for the design, development and manufacture of medical devices and products.

In addition, a recent survey undertaken by the Centre for Software Process Technologies (CSPT), indicated that Northern Ireland health technology companies have limited awareness of standards applied to software development, such as CMMI [5]. The CSPT [6] is a research and knowledge transfer group funded jointly by the University of Ulster and a Northern Ireland governmental organisation charged with the economic development of this geographical region. The CSPT is tasked with motivating and supporting a culture of software process improvement within the Northern Ireland software industry. The survey findings noted that few organisations are aware of the concept of software process improvement and more alarmingly, the importance of software process improvement for increasing market penetration worldwide, particularly in the U.S.A. If this is not addressed, medical device software companies will

experience difficulties in competing in markets where advanced software systems are required and time to market is crucial.

Interestingly the prime factor here is not simply cost, but the quality agenda as a whole, with competitor companies possibly having already engaged in process improvement programs. Indeed, in countries such as China [7] there is considerable recognition of the importance of quality in software development with the Chinese government not only setting up government agencies with responsibility for building the maturity of the Chinese software industry but also offering subsidies to software enterprises that engage in SPI based maturity evaluations.

2. SPI framework.

IQ Solutions and the CSPT are developing a software development framework for the medical device sector that

addresses existing regulatory requirements for the control of the design, development, maintenance and support of software.

The approach for delivering the software development framework is to establish a model (implemented as illustrated in Figure 1) that addresses the relevant regulations, and integrates those constraints within an SPI framework (i.e. MedeSPI). The model will be flexible in that relevant elements of the SPI framework may be adopted as required to provide the most significant benefit to the business. For the purpose of this paper, the SPI framework used will be that of the CMMI [8] and the regulations used to extend the CMMI framework will be those of the FDA.

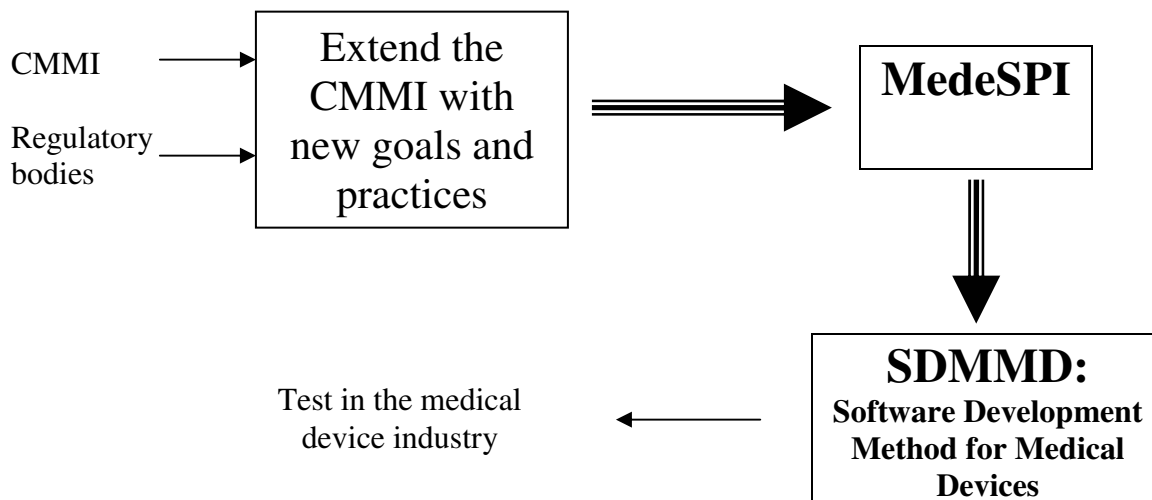


Figure 1: Software framework approach

3. Project outline.

In order to deliver an endorsed framework it was essential that a steering group was formed with members from various medical device companies and a notified body with experience in auditing medical device companies. The involvement of medical device companies also adds an ownership element to the model

and should improve its acceptance and implementation within each company.

The Software Development Method for Medical Devices (SDMMD) will be a defined set of software process models (in effect a methodology) which when utilised will meet the goals of MedeSPI. SDMMD will cover the complete lifecycle, defined by default, as the V-Model in Figure 2. No restriction will be made on the development lifecycle processes undertaken by individual companies, although it is understood that companies

within the medical device sector typically implement this V-Model. In order to achieve this the project is divided into several stages.

1. Assess the need for and commitment to the creation of SDMMMD and MedeSPI
2. Identify which parts of the CMMI are required to comply with FDA regulation and extend the

CMMI with new goals and practices that are necessary to achieve FDA compliance (i.e. creation of MedeSPI).

3. Develop process models for meeting the goals of MedeSPI (i.e.create SDMMMD)
4. Test SDMMMD with Northern Ireland medical device companies

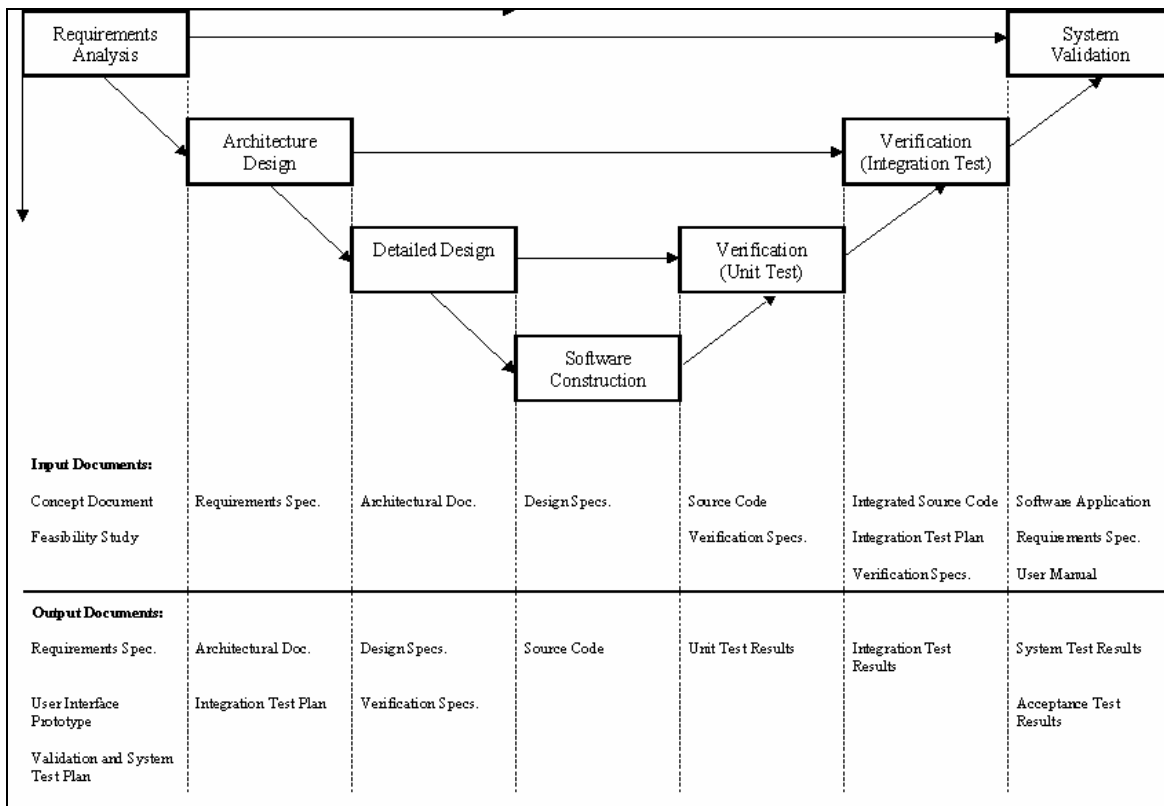


Figure 2: Software life cycle V model

We have completed stage 1 of our work and are currently performing stage 2 activities.

4. MedeSPI development.

SDMMMD will provide a software development methodology, which addresses the regulatory guidance criteria, while introducing best processes that can be selected as required.

MedeSPI will provide a means of assessing software engineering capability in twelve areas that have been defined by the FDA [9,10] as:

1. Level of Concern
2. Software Description
3. Device Hazard and Risk Analysis

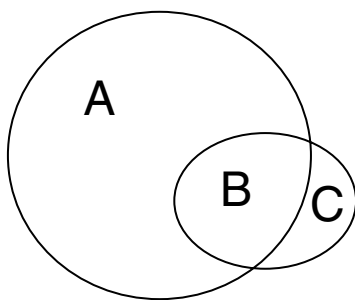
4. Software Requirements Specification
5. Architecture Design
6. Design Specifications
7. Requirements Traceability Analysis
8. Development
9. Validation, Verification and Testing [11]
10. Revision Level History
11. Unresolved Anomalies
12. Release Version Number

MedeSPI is being developed to promote software process improvement practices into the software development processes of medical device companies. This is an attempt to improve the effectiveness and efficiency of software processes used by medical device companies through investigating the mapping between twelve CMMI process areas and the twelve FDA areas

listed above. The twelve CMMI process areas that we have deemed appropriate for the medical device industry are as follows:

1. Project Planning,
2. Project Monitoring & Control,
3. Supplier Agreement Management,
4. Risk Management,
5. Requirements Management,
6. Requirements Development,
7. Technical Solution,
8. Product Integration,
9. Verification.
10. Validation,
11. Configuration Management,
12. Process and Product Quality Assurance.

The mappings between the FDA regulatory guidelines and the CMMI process areas listed above then produce twelve MedeSPI process areas which retain the CMMI process area names listed above. Each of the MedeSPI process areas will then be composed of a number of goals and practices. Goals and practices may be either generic (relating to the entire organisation) or specific (relating to the current process area). MedeSPI investigates what parts of the CMMI process areas are required to satisfy FDA regulations, but also investigates the possibility of extending the CMMI process areas with additional goals and practices that are outside the remit of CMMI, but are required in order to satisfy FDA regulations. The composition of the MedeSPI framework is illustrated in figure 3.



- A- CMMI Practices that are not mandatory for FDA compliance.
- B- CMMI Practices that are required for FDA compliance.
- C- Non-CMMI Practices that are required for FDA compliance.

Figure 3. Composition of the MedeSPI framework.

The model will help companies to measure their organisational capability and to track progression and achievements in each of the twelve process areas and against process capability levels. The MedeSPI framework has adopted the following capability levels:

- **Level 0** – Companies must demonstrate that a process area satisfies the goals and performs the practices required to achieve FDA regulatory compliance. This will involve performing some practices which the CMMI views as generic, although not to the extent of fulfilling any generic goals.
- **Level 1** - Companies must demonstrate that a process area satisfies level 0 and the CMMI capability level 1 goal of performing the CMMI base practices.
- **Level 2** – Companies must demonstrate that a process area satisfies level 1 and additionally performs CMMI Advanced Practices, as well as the CMMI capability level 2 generic goal of Institutionalising a Managed Process.
- **Level 3** - Companies must demonstrate that a process area satisfies level 2 and additionally the CMMI Generic Goal to Institutionalise a Defined Process (CMMI Generic Goal 3).
- **Level 4** – Companies must demonstrate that a process area satisfies level 3 and additionally the CMMI Generic Goal to Institutionalise a Quantitatively Managed Process (CMMI Generic Goal 4).
- **Level 5** - Companies must demonstrate that a process area satisfies level 4 and additionally the CMMI Generic Goal to Institutionalise an Optimising Process (CMMI Generic Goal 5).

What follows is a mapping of the FDA regulations to the CMMI for the Requirements Management (REQM) process area. This will demonstrate what CMMI goals and practices are required in order to satisfy FDA guidelines for requirements management. Software development within medical device companies could be improved by incorporating other CMMI practices that are not required to achieve FDA compliance. Comment is provided on how additional goals and practices (not included in the CMMI) may be added where necessary to satisfy FDA regulatory guidelines. REQM goals and practices have to be performed to satisfy each of the MedeSPI capability levels.

6. Requirements management process area.

FDA regulations [1] which have a counterpart within the goals and practices of the CMMI REQM process area

and are related to the creation of software are identified. REQM has one specific goal which is *SGI Manage Requirements*. In order that this goal is achieved it is necessary for the following five practices to be performed:

- 1.1-1 Obtain an understanding of requirements,**
- 1.2-2 Obtain commitment to requirements,**
- 1.3-1 Manage requirements changes,**
- 1.4-2 Maintain bi-directional traceability of requirements,**
- 1.5-1 Identify inconsistencies between project work and requirements.**

Obtaining an understanding of requirements involves several activities identified within CMMI as sub-practices. To fully perform 1.1-1, it is necessary to:

- Establish criteria for distinguishing appropriate requirements providers,
- Establish objective criteria for the acceptance of requirements,
- Analyse requirements to ensure that the established criteria are met,
- Reach an understanding of the requirements with the requirements provider so that the project participants can commit to them.

The FDA regulations do not mandate any criteria for distinguishing providers of requirements. The formal identification of requirements providers is necessary to avoid conflict in the gathering of requirements from multiple and possibly unauthorised sources. This is a particular issue for software requirements because problems with requirements are very expensive to rectify later in a lifecycle. Although the FDA regulations recommend the creation of procedures to address incomplete and conflicting requirements, collecting requirements in parallel from different sources can exacerbate the problem.

Establishing objective criteria for the acceptance of requirements suggests the need for potential requirements to be selected on the basis of agreed criteria. The FDA regulations do not require such criteria to be established. This is a fundamental issue during requirements gathering because of the need to weigh and prioritise the value of the software requirements to ensure only the incorporation of valid requirements into the software development process. The other side of the coin to establishing objective criteria is the inspection of the requirements to ensure compliance. Although there is a need to approve requirements within the FDA regulations, failure to stipulate a need for criteria, renders inspection against criteria impossible. There is a necessity to ensure that the design requirements relating

to a device are appropriate and address the intended use of the device but alarmingly, within the FDA regulations, there is no specific necessity to reach an understanding of requirements with the requirements provider.

Obtaining commitment to requirements (1.2-2), involves demonstrating that the following activities are being performed:

- Assess the impact of requirements on existing commitments
- Negotiate and record commitments

There is no retrospective review of existing commitments recommended by the FDA regulations whenever new requirements are established and no treatment of the negotiating and recording of commitments. The ability of an organisation to create software which meets requirements is tied to its ability to handle its commitments. However, even the CMMI treats this as an advanced practice so perhaps it is not so surprising that the FDA regulations fail to address it.

In **managing requirements changes** (1.3-1), an organisation needs to:

- Capture all requirements and requirement changes that are given to or generated by the product,
- Maintain the requirements change history with the rationale for the changes,
- Evaluate the impact of requirements changes from the standpoint of relevant stakeholders,
- Make the requirement and change data available to the project.

The FDA regulations specifically require processes to ensure the completeness of requirements. However, changes to requirements are not specifically treated beyond the context of completeness. The FDA regulations do suggest that changes to documents or indeed any specification, shall follow a process and be reviewed and approved by an individual(s) in the same function or organization that performed the original review and approval, so at least in this respect, changes to requirements are indirectly recognised.

Maintaining the requirements change history with the rationale for the changes is an issue strongly associated with configuration management. The FDA recommends the maintenance of a design history file where each manufacturer shall establish and maintain a history for each type of device. The history shall contain or reference the records necessary to demonstrate that the design was developed in accordance with the approved design plan and consequently, requirements. Further, the regulations mandate that each manufacturer shall maintain records of changes to documents including a

description of the change (presumably including requirements documents). This though should not be taken as reference to the maintenance of full and proper requirements configurations. When it is necessary to evaluate requirement changes from the standpoint of stakeholders, the regulations simply suggest a process should be followed for changes to specifications, no specific mention of stakeholders is made. Making requirement and change data available to the project has a counterpart in the FDA regulations where it suggests that documents shall be available at all locations for which they are designated, used, or otherwise necessary.

Maintaining bi-directional traceability of requirements (1.4-2) involves:

- Maintaining requirements traceability to ensure that the source of lower level requirements is documented,
- Maintaining requirements traceability from a requirement to its derived requirements and to functions, objects, people, processes and work products,
- Generating the requirements traceability matrix.

With respect to traceability, the FDA regulations require that in certain circumstances procedures are established and maintained for identifying with a control number each unit, lot, or batch of finished devices and where appropriate components. This has an implicit reference to managing configurations and traceability but some would argue that there are many meanings to the word component and that taking this to imply requirements is a step too far. Again, the CMMI regards these practices as advanced.

Identify inconsistencies between project work and requirements (1.5-1) is performed by:

- Reviewing the projects plans, activities and work products for consistency with the requirements and the changes made to them,
- Identifying the source of the inconsistency and the rationale,

- Identifying changes that need to be made to the plans and work products resulting from changes to the requirements,
- Initiating corrective action.

The FDA regulations ensure during design verification that the design output meets the design input requirements. However, dealing with the source of the inconsistency, identifying changes to plans and initiating corrective action are ignored.

The CMMI identifies a number of generic goals and practices. At a fundamental maturity or capability level it is only necessary to perform the specific base practices. It is interesting to note that FDA regulations with respect to REQM often have a counterpart in the CMMI. For REQM the generic goals and practices for capability level 2 are:

GG 2: Institutionalise a Managed Process

- GP 2.1 Establish Policy**
- GP 2.2 Plan the process**
- GP 2.3 Provide Resources**
- GP 2.4 Assign Responsibility**
- GP 2.5 Train People**
- GP 2.6 Manage Configurations**
- GP 2.7 Identify stakeholders**
- GP 2.8 M&C Process**
- GP 2.9 Evaluate Adherence**
- GP 2.10 Review**

The FDA regulations state that each manufacturer shall establish the appropriate responsibility, authority, and interrelation of all personnel who manage, perform, and assess work affecting quality. It also undertakes to ensure that all work is adequately resourced and that staff are trained.

The following table (Table 1) illustrates what REQM goals and practices have to be performed for each of the MedeSPI capability levels.

Table 1: MedeSPI Requirements management process area components and capability levels

<i>Goal</i>	<i>Practice</i>	<i>Sub-Practice</i>	<i>Level</i>
SG1 Manage Requirements.	1.1-1 Obtain an understanding of requirements	Establish criteria for distinguishing appropriate requirements providers	1
SG1 Manage Requirements.	1.1-1 Obtain an understanding of requirements	Establish objective criteria for the acceptance of requirements	1
SG1 Manage Requirements.	1.1-1 Obtain an understanding of requirements	Analyse requirements to ensure that the established criteria are met	1
SG1 Manage Requirements.	1.1-1 Obtain an understanding of requirements	Reach an understanding of the requirements with the requirements provider so that the project participants can commit to them	1
SG1 Manage	1.2-2 Obtain commitment to	Assess the impact of requirements on existing	2

Requirements.	requirements	commitments	
SG1 Manage Requirements.	1.2-2 Obtain commitment to requirements	Negotiate and record commitments	2
SG1 Manage Requirements	1.3-1 Manage requirements changes	Capture all requirements and requirement changes that are given to or generated by the product	0
SG1 Manage Requirements	1.3-1 Manage requirements changes	Maintain the requirements change history with the rationale for the changes	0
SG1 Manage Requirements	1.3-1 Manage requirements changes	Evaluate the impact of requirements changes from the standpoint of relevant stakeholders	1
SG1 Manage Requirements	1.3-1 Manage requirements changes	Make the requirement and change data available to the project.	0
SG1 Manage Requirements	1.4-2. Maintain bi-directional traceability of requirements	Maintaining requirements traceability to ensure that the source of lower level requirements is documented	1
SG1 Manage Requirements	1.4-2. Maintain bi-directional traceability of requirements	Maintaining requirements traceability from a requirement to its derived requirements and to functions, objects, people, processes and work products	0
SG1 Manage Requirements	1.4-2. Maintain bi-directional traceability of requirements	Generating the requirements traceability matrix	2
SG1 Manage Requirements	1.5-1 Identify inconsistencies between project work and requirements.	Reviewing the projects plans, activities and work products for consistency with the requirements and the changes made to them	0
SG1 Manage Requirements	1.5-1 Identify inconsistencies between project work and requirements.	Identifying the source of the inconsistency and the rationale	1
SG1 Manage Requirements	1.5-1 Identify inconsistencies between project work and requirements.	Identifying changes that need to be made to the plans and work products resulting from changes to the requirements	1
SG1 Manage Requirements	1.5-1 Identify inconsistencies between project work and requirements.	Initiating corrective action	1
GG 2:Institutionalise a Managed Process (IMP)	GP 2.1 Establish Policy		2
GG 2:IMP	GP 2.2 Plan the process		2
GG 2:IMP	GP 2.3 Provide Resources		0
GG 2:IMP	GP 2.4 Assign Responsibility		0
GG 2:IMP	GP 2.5 Train People		0
GG 2:IMP	GP 2.6 Manage Configurations		2
GG 2:IMP	GP 2.7 Identify stakeholders		2
GG 2:IMP	GP 2.8 M&C Process		2
GG 2:IMP	GP 2.9 Evaluate Adherence		2
GG 2:IMP	GP 2.10 Review		2
GG3 :Institutionalise a Defined Process	GP 3.1 Establish a defined Process		3
GG3 :Institutionalise a Defined Process	GP 3.2 Collect Improvement Information		3
GG4:Institutionalise a Quantitatively Managed Process	GP 4.1 Establish Quantitative Objectives for the Process		4
GG4:Institutionalise a Quantitatively Managed Process	GP 4.2 Stabilise Sub-process Performance		4
GG5 :Institutionalise an Optimising Process	GP 5.1 Ensure Continuous Process Improvement		5
GG5 Institutionalise an Optimising Process	GP 5.2 Correct Root Causes of Problems		5

7. Conclusion.

With respect then to the specific goals and practices of the requirements management process area, it is clear that following FDA regulations will only, at best, partially meet the goals of this CMMI process area. As might reasonably be expected, there is no support within the FDA regulations for the advanced practices of requirements management but perhaps more surprising there is little equality between more fundamental practices of requirements management and those mandated in the FDA regulations. It is difficult to come to terms with regulations which permit compliance for requirements management practices which do not cater for identifying formally a source of requirements nor for the formal assessment of requirements. Requirements are the fundamental starting point for the birth of a software system and their full and proper understanding is essential for efficient and effective software development.

Since failure to perform any specific practice implies failure to meet the specific goal, with respect to CMMI, it is clear, the goals of REQM cannot be obtained by satisfying FDA regulations during software development. But is the opposite true, can meeting the CMMI goals for REQM successfully meet FDA regulations? Certainly for REQM, meeting the goals of the process area by performing the specific practices more than meets the FDA regulations in this area. For REQM, the existing CMMI specification of goals and practices can be carried over, without extension into the MedeSPI framework.

Our work continues for stage 2 of our project. We will endeavour to examine all of the appropriate process areas within the CMMI referred to in the FDA regulations, investigating the extent to which the CMMI framework needs to be extended to create MedeSPI.

8 Acknowledgements.

The Centre for Software Process Technologies at the University of Ulster is supported by the EU Programme For Peace And Reconciliation in Northern Ireland and The Border Region Of Ireland (PEACE II).

9. References.

- [1]. FDA Regulations. "Code of Federal Regulations 21 CFR Part 820." June 1997.
- [2]. European Council, "Council Directive 93/42/EEC Concerning Medical Devices", 14 June 1993.
- [3]. Safety in the Process Industries – Guiding Principles to the implementation of IEC 61508 www.sipi61508.com
- [4]. ISO 13485:2003 Medical devices -- Quality management systems -- Requirements for regulatory purposes. www.iso.ch
- [5]. McFall, D., Wilkie, F.G. McCaffery, F. Lester, N.G. and Sterritt, R. "Software Processes and Process Improvement in Northern Ireland", 16th International Conference on Software & Systems Engineering and their Applications, Paris, France, December 2003, ISSN: 1637-5033, p1-10.
- [6]. The Centre for Software Process Technologies, University of Ulster, www.cspt.ulster.ac.uk.
- [7]. Ju Dehua., China's Budding Software Industry, IEEE Software May/June 2001. pp 92-95
- [8]. Capability Maturity Model® Integration (CMMISM) for Software Engineering (CMMI-SW, V1.1, Version 1.1, August 2002)
- [9]. FDA/CDRH Guidance Document. "Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices." May 1998.
- [10]. FDA/CDRH Guidance Document. "Guidance for Off-the-Shelf Software Use in Medical Devices." September 1999
- [11]. FDA/CDRH Guidance Document. "General Principles of Software Validation." June 1997.