

ULRR

A proposed framework for software quality in the healthcare and medical industry.

Item Type	Meetings and Proceedings
Authors	Shroff, Vispi;Reid, Louise;Richardson, Ita
Citation	European systems and Software Process Improvement and Innovation Conference.;2011
Download date	2026-05-09 14:43:15
Item License	https://creativecommons.org/licenses/by-nc-sa/1.0/
Link to Item	https://hdl.handle.net/10344/2081

A Proposed Framework for Software Quality in the Healthcare and Medical Industry

Vispi Shroff, Louise Reid, Ita Richardson

Abstract

In this paper, we propose a theoretical framework for software quality within the healthcare sector. The aim of this framework is to improve the quality of software produced within the healthcare and medical device industries, while assisting the manufacturers in achieving compliance with existing regulation. To develop this framework, we undertook an evaluation of the healthcare and medical legislation. This was followed by an evaluation of existing software quality standards and models. The CMMI model was chosen as the model on which to base this framework. An initial mapping between CMMI and the US FDA Code of Federal Regulations was undertaken. We present some of the examples the mapping here.

Keywords

CMMI, 21CFR820, medical software development, healthcare software development, software quality

1 Introduction

One of the prominent issues in the software industry is the development of high-quality software [1]. When implementing software within the healthcare and medical industry the issues of software quality become more serious. As software is used increasingly within medical devices and in hospitals, any software quality issues can have significantly detrimental effects on a person's recovery, health and well-being. Some of the earliest and most prominent cases were the Therac-25 accidents between 1985 and 1987[2]. There is also a cost involved for organizations and companies when medical device recalls need to be made. Between 1999 and 2005, it was found that "one in every three medical devices, making use of software for their operation, has been recalled due to failure in the software itself" [3].

Compliance with legislation is another major hurdle which companies have to overcome at present. Many companies are utilizing software development methodologies such as waterfall and V-model and are reticent to embrace agile methodologies. Having achieved certification and compliance many companies find the potential risk of failing compliance with regulation due to extensive changes in practices and processes to be an inhibiting factor in adopting new processes.

In this paper, we have proposed a software quality framework which looks at software quality, software standards, healthcare regulation and implementation within a software development life cycle which can facilitate continuous software process improvement. This framework has been proposed after carrying out a literature review and a study of existing practices in a clinical environment. It is developed within the context of implementation within a European healthcare system research project, the aim of which is to develop a "rapid learning healthcare system' driven by advanced computational infrastructure that can improve both patient safety and the conduct and volume of clinical research in Europe" [4].

2 Research Methodology

To develop an understanding of the domain, case study research was undertaken on a number of systems in an Irish Hospital. This provided an insight into the quality issues that are unique to the clinical environment. This case study research was further strengthened by reviews into high profile hospital investigations such as the Lourdes Hospital Inquiry [5], the Shipman Inquiry [6], the Tallaght Hospital Review [7] and the investigation into the Therac 25 incidents. Findings from our research pointed to the importance of quality processes to ensure the success of existing systems. The Bristol enquiry found that the hospital was 'awash with data but that there was very little information'; this sentiment was echoed in the Lourdes Hospital Enquiry. We found systems that are in place but underutilised. We found cases where the staff that should be making use of these systems did not trust them and that this led to loop of quality degeneration due to underuse of existing information. All of the investigations highlighted the fact that things can and do go wrong in the clinical environment but that the risk can be offset by robust quality processes.

Through interviews with Irish medical device manufacturers and software vendors for the healthcare industry, we learnt that ensuring compliance with healthcare regulation can be a significant barrier to entry and a driver of effort and costs within the industry. A literature review was then carried out to study the current state of healthcare legislation. The focus of this review was primarily on US and European legislation followed by a study of Irish regulation. Some issues around relevance and ambiguity were identified within the legislation. As increasing the quality of the products manufactured appeared to be the purpose behind a lot of the regulation requirements, software quality standards and models such as the ISO 15504 and the various CMMI models were also studied.

The initial research has focussed around creating a mapping between selected process areas within the CMMI model and the Code of Federal Regulations Title 21 Part 820 regulatory requirements. The aim of creating this mapping is to provide software vendors to the healthcare industry a set of software processes which when undertaken, may not only improve the quality of the software manufactured, but also assist them in becoming compliant with regulation.

2.1 Healthcare Legislation

The US FDA Code of Federal Regulations was the core legislation studied. In particular, Subchapter H which focused on Medical Devices and Part 820 Quality System Regulation was studied in great depth [8]. This regulation is used as the foundation upon which the software quality models are mapped. European Council directives such as the Council Directive 90/385/EEC on Active Implantable Medical Devices (AIMDD), Council Directive 93/42/EEC on Medical Devices (MDD), Council Directive 98/79/EC on In Vitro Diagnostic Medical Devices (IVDMD) and subsequent amendments via Council Directives 98/79/EC, 2000/70/EC, 2001/104/EC, 2003/32/EC and 2007/47/EC were also studied.

Historically, the legislative requirements in the healthcare industry revolved around monitoring the processes and practices involved in the development of physical devices, not software. However, over the last couple of decades software has played a larger and increasingly significant role in the operation of medical devices. While the legislation was primarily developed to suit physical and hardware manufacturing processes, it has struggled to keep up with the changes in the manufacturing practices and the different problems and challenges which software development presents.

The EU Council Directive 2007/47/EC has come closest to including standalone software into legislation by adding “Stand alone software is considered to be an active medical device” and “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” to the regulation. [9]

Some of the other key issues we found within the medical legislation included

- The lack of clear lines of distinction between hardware and software, and the impact of changes in software are not outlined.
- The focus of the healthcare industry is on patient privacy and security with very little focus on integrity and accuracy.
- Medical devices at present are categorized into three (US) or four (EU) different categories based on the potential impact they may have on patient health. These categories are shown in the table below, and there remains ambiguity in the definitions of these risk factors.

Table 1. Risk-based Medical Device Categorization in Legislation

EU Legislation		US Legislation	
Class	Risk	Class	Risk
Class I	Low	Class I	Low
Class IIa	Low-Medium	Class II	Medium
Class IIb	Medium-High	Class III	High
Class III	High		

Also, the proliferation of smartphones and the wide-spread usage of the application stores have seen a large increase in healthcare and medical applications becoming readily and openly available. At present, there is no means of implement and monitoring any regulatory or quality guidelines on these applications and this is becoming a major area of concern.

2.1.1 Code of Federal Regulations Title 21 Part 820 (21 CFR 820)

As mentioned earlier, the 21CFR820 regulation document was adopted for the initial draft of the framework. To provide systems and solutions in the healthcare industry in the United States, products manufactured by companies need to be compliant with this regulation. The focus of this regulation is on the quality systems utilised in the manufacturing processes. There are certain artefacts which need to be created according to the 21CFR820 regulation. These artefacts are similar to and complement certain software engineering artefacts that are created as part of the software development lifecycle. Some examples of 21CFR820 artefacts include:

- Design history file (DHF) means a compilation of records which describes the design history of a finished device.

- Device history record (DHR) means a compilation of records containing the production history of a finished device.
- Device master record (DMR) means a compilation of records containing the procedures and specifications for a finished device.

From the descriptions, we can see that these artefacts are similar to the requirements specifications, design documents and other material which may be created during the software development lifecycle. So, if software development best practices are correctly followed, the creation of these artefacts and the documentation of the procedures and methods should take place, thereby assisting the software vendor in achieving compliance with the regulation. This is where recommendations by software quality standards and models can be of assistance in developing good software development practices which assist a vendor in becoming compliant with regulation.

2.2 CMMI

For the purposes of this software quality framework, the Software Engineering Institute's Capability Maturity Model Integrated (CMMI) for Development [10] was utilized. The CMMI model provides flexibility as it allowed us to focus on very specific process areas. Also, this modularity would theoretically allow this framework to be implemented with any software development methodology. As the CMMI has its origins in Software Quality, and is freely and openly available it is widely adopted and validated by numerous companies and across various different industries. While it is heavily used in the software industry, there are also case studies available on the implementation of CMMI in the health care services industry [11]. We also found that the CMMI model and its ability to measure capability and maturity of certain processes mapped very easily to the Quality and Risk Management Standard (QRMS) adopted by Irish Health Service Executive[12]. Apart from this, CMMI's lays focus on and facilitates continuous SPI as compared to the IEC 62304 and other ISO standards. All these factors contributed towards the selection of CMMI as the primary model to base our framework on.

To reduce the scope of the investigation, the six process areas within the CMMI were focused on for mapping to the 21CFR820 legislation. These included the process areas of Requirements Development (RD), Requirements Management (REQM), Technical Solution (TS), Verification (VER) and Validation (VAL) and Configuration Management (CM). These process areas were chosen following discussions with the software developers and health informatics personnel involved in the healthcare system research project. These process areas were explored in depth and the specific practices and work products were mapped to the 21CFR820 legislation to see how they would satisfy regulatory requirements.

3 The CMMI – 21CFR820 Mapping

The fundamentals of the software quality framework lie in the mapping between the software models and the legislation. In the sections below, we have provided the mapping between the chosen process areas and how they satisfy different sections of the 21CFR820 legislation. Each process area within CMMI has a set of specific goals, each containing a set of specific practices. These specific practices are designed to assist in the accomplishment of the goal. The adoption of each of these specific practices creates certain outputs and deliverables. These outputs and deliverables can then be utilised to achieve compliance with the existing regulations.

We will illustrate this mapping using the specific goal of Analysing and Validating Requirements within the Requirements Development Process Area.

3.1 Requirements Development Process Area

The purpose of Requirements Development is to elicit, develop and analyse customer, product and product component requirements. It involves studying and understanding the customers need, expectations and constraints which take into consideration all the stakeholder needs. Customer require-

ments are further developed into product requirements and product component requirements which are consistent with customer requirements.

CMMI provides three specific goals; develop customer requirements, develop product requirements and to analyse and validate requirements. The latter goal of validating and analysing the requirements is designed to assist in the accomplishment of the former two goals.

An example of this mapping, the CMMI-21CFR820 mapping is shown in the table below.

Table 2. Analyze and Validate Requirements

No.	Specific Practice	Deliverables	Legislation
SP 3.1	Establish operational concepts and scenarios	Operational concepts of the systems	DMR
SP 3.1	Establish operational concepts and scenarios	Product or product component installation, operational, maintenance and support concepts	DMR
SP 3.1	Establish operational concepts and scenarios	Disposal concepts	DMR
SP 3.1	Establish operational concepts and scenarios	Use Cases	DMR
SP 3.1	Establish operational concepts and scenarios	Timeline scenarios	DMR
SP 3.1	Establish operational concepts and scenarios	New requirements	DMR
SP 3.2	Establish a definition of required functionality	Functional Architecture	DMR
SP 3.2	Establish a definition of required functionality	Activity diagrams, use cases	DMR
SP 3.2	Establish a definition of required functionality	Object oriented analysis; services and methods defined	DMR
SP 3.3	Analyse Requirements	Key Requirements	DMR
SP 3.3	Analyse Requirements	Technical performance measures to measure progress	DMR
SP 3.3	Analyse Requirements	Requirements defects reports	DMR
SP 3.3	Analyse Requirements	Proposed Requirements Defects Reports	DMR
SP 3.4	Analyse Requirements to Achieve Balance	Document assessing risks related to requirements defined	21CFR820 Sec 820.30
SP 3.5	Validate Requirements	Records of requirements validation analysis and results	21CFR820 Sec 820.65
SP 3.5	Validate Requirements	Change reports for requirements which need changes based on validation results.	21CFR820 Sec 820.65

In this case, when following the specific practices for achieving the Requirements Development goal of analysing and validating requirements, the software manufacturer may create use cases, timeline scenarios, a functional architecture and much more. Many of these artefacts can be then be utilised in the creation of the Device Master Record (DMR) of the software for compliance with 21CFR820.

4 Future Research

There is great scope for future research to enhance the framework. The key challenges at present lie in addressing the applicability of such a framework within industry. We will be using the healthcare system research project as a test bed to evaluate the applicability of such a framework within software development teams. For evaluation purposes, it will be necessary to utilize a software development methodology to implement the framework.

Other areas of research would include extending the framework to include European legislation primarily, and other regional legislation may follow. European and US legislation tend to be leading the world as far as healthcare and medical devices are concerned, encompassing these bodies of legislation should lead to a framework which relatively encompasses universal legislation.

After extending the framework to include other legislation, one may look at other software quality models and standards such as the ISO standards to enhance aspects which may be missing from this model. For example, when compared to IEC 62304, CMMI was found “lacking specific safety related requirements that have been derived over time through regulatory monitoring of software intensive medical devices.”[13]

Finally, due to the intensive nature of the quality framework, as well as the amount of regulation involved, it would be worthwhile creating different models of the framework based on risk assessment of the medical device or software being produced. If a software or medical device is evaluated to be a low risk device, i.e. Class I device, then it may not be necessary to follow all the recommendations within the framework. In such situations, it might be worthwhile having a model of the framework which is designed for the different classes of risk. However, interviews with medical device manufacturers have shown us that the default class assumed by regulators is the highest risk class, and the onus lies on the manufacturers to demonstrate the reduced risk. Also, even though individual components may be fall under a lower risk category, the final evaluation is based on the overall risk classification of the medical device or software. So, even though software may play a miniscule and trivial role in the overall functioning of a medical device, if that medical device is classified as a Class III device, the software will need to pass Class III regulation guidelines.

Finally, one of the prominent issues which arose during the investigation into the case studies and subsequent interviews with hospital clinicians was that many hospital systems remained underused or stagnant due to the lack of trust in the systems and the general sentiment that the systems were not ‘fit-for-purpose’. After the framework has been developed in further depth, the study of software development lifecycles may be carried out with a particular focus on agile processes. Within the European healthcare system research project, the Scrum methodology was proposed for implementation and will be observed for effectiveness against the scale and the nature of the project. We will also observe how the Scrum methodology facilitates this framework with the least challenges while allowing modification and improvement of the framework over time.

5 Conclusion

We have laid the foundations for a software quality framework for the healthcare and medical device industry. In a world where software is playing an increasingly pivotal role in the delivery of healthcare and patient well-being, the quality of the software being used is paramount. While there has been extensive work done on the patient privacy and data security side of the spectrum, there has been little focus on the data integrity and accuracy within systems which are becoming increasingly integrated. In this framework, our focus has not been on the outputs of the software development processes, but rather on the software development processes themselves. The proposed framework provides flexibility as a single component of it can be used by itself, or it can be implemented in its entirety. As the medical and healthcare industry is a heavily regulated industry, an equal amount of focus was laid on the existing legislation. It is the authors’ goal to provide a framework which is flexible, practical and easily implementable. We believe this framework allows the manufacturers to be compliant with regulation while developing high quality software.

6 Acknowledgments

This work is partially supported by Science Foundation Ireland grant 3/CE2/I303_1 to Lero. This work is also partially supported by TRANSFoRM which is funded by the European Commission – DG INSFO (FP7 247787).

7 Literature

- [1] A. C. Gillies, *Software Quality: Theory and Management*. London, UK, UK: Chapman & Hall, Ltd., 1992.
- [2] N. Leveson and C. Turner, "An Investigation Of The THERAC-25 Accidents," *Computer*, vol. 26, no. 7, pp. 18 - 41, Jul. 1993.
- [3] Z. Bliznakov, G. Mitalas, and N. Pallikarakis, "Analysis and Classification of Medical Device Recalls," in *World Congress on Medical Physics and Biomedical Engineering 2006*, vol. 14, Springer Berlin Heidelberg, 2007, pp. 3782-3785.
- [4] "TRANSFoRm." [Online]. Available: <http://www.transformproject.eu/Home.html>. [Accessed: 25-Feb-2011].
- [5] M. Harding Clark, *Lourdes Hospital Inquiry : An Inquiry into peripartum hysterectomy at Our Lady of Lourdes Hospital, Drogheda*. Our Lady of Lourdes Hospital, 2006.
- [6] J. Smith, *The Shipman Inquiry, Fifth Report - Safeguarding Patients: Lessons from the Past - Proposals for the Future*. UK: , 2004.
- [7] M. Hayes, *Report of the review of radiology reporting and the management of GP referral letters at Adelaide and Meath Hospital (Dublin), incorporating the National Children's Hospital, (AMNCH) [Tallaght Hospital]*. Tallaght: Health Service Executive, 2010.
- [8] US Food & Drug Administration, "Code of Federal Regulations Title 21 Part 820 Quality System Regulation." Apr-2010.
- [9] European Council, *Directive 2007/47/EC of the European Parliament and of the Council of 5 September 2007 amending Council Directive 90/385/EEC on the approximation of the laws of the Member States relating to active implantable medical devices, Council Directive 93/42/EEC concerning medical devices and Directive 98/8/EC concerning the placing of biocidal products on the market Text with EEA relevance*. 2007, pp. 21-55.
- [10] SEI, "CMMI for Development : Version 1.2." Aug-2006.
- [11] E. Forrester, "CMMI for Services - Overview Presentation." Software Engineering Institute, 2008.
- [12] HSE, "Quality & Risk Management Standard." Health Service Executive, Nov-2007.
- [13] D. W. Walker, "CMMI and Medical Device Engineering." Software Engineering Institute, 29-Sep-2009.

8 Author CVs

Vispi Shroff

Vispi Shroff is currently a part-time postgraduate researcher in Lero – The Irish Software Engineering Research Centre in the University of Limerick. Vispi has numerous years of experience in industry in the Irish SME sector and was responsible for project management and quality assurance of the software products manufactured. Having completed an MSc in Software Engineering from the University of Limerick in 2005, Vispi is also a Certified Scrum Master and has an interest in software process improvement, agile methodologies and software project management. Vispi is also pursuing a part-time MBA with the University of Liverpool.

Email: vispi.shroff@ul.ie

Louise Reid

Louise Reid is a Quality Auditor with the Health Service Executive, the national body providing healthcare in Ireland. A qualified nurse, Louise has worked in Irish hospitals over the past two decades and has first-hand experience of the software usage and quality within the Irish hospitals and healthcare system. Louise has completed a MSc in Health Informatics from Trinity College, Dublin and is currently doing her PhD with Lero – The Irish Software Engineering Research Centre in the University of Limerick.

Email: louise.reid@hse.ie

Ita Richardson

Ita Richardson is currently the project leader on numerous Irish and EU research projects within Lero – The Irish Software Engineering Research Centre in the University of Limerick. With almost two decades of lecturing and research experience, Ita has an interest in the areas of medical device software engineering, software quality, software processes, requirements engineering and global software development amongst others.

Email: ita.richardson@ul.ie