The Medical Device Industry

The Medical Device Industry: Developments in Software Risk Management

By

John Burton

Edited by Ita Richardson, Fergal Mc Caffery and Mícheál Ó hAodha

CAMBRIDGE
SCHOLARS
PUBLISHING

The Medical Device Industry: Developments in Software Risk Management, by John Burton
Edited by Ita Richardson, Fergal Mc Caffery and Mícheál Ó hAodha

This book first published 2009

Cambridge Scholars Publishing

12 Back Chapman Street, Newcastle upon Tyne, NE6 2XX, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-4438-0561-0, ISBN (13): 978-1-4438-0561-2

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author wishes to acknowledge the assistance of Dr. Ita Richardson and Dr. Fergal McCaffery with this research. A special thanks also goes to the medical device company in which this research was performed.

The author would like to acknowledge that this research was funded by Science Foundation Ireland through the Cluster project GSD for SMEs (Grant no 03/IN3/1408C) within the University of Limerick, Ireland. This research is supported by the Science Foundation Ireland funded project, *Global Software Development in Small to Medium Sized Enterprises (GSD for SMEs)* within Lero - the Irish Software Engineering Research Centre (http://www.lero.ie).

CHAPTER ONE

INTRODUCTION

Risk

Several definitions exist for the term “risk” but they typically contain a common theme of loss, injury or harm. Cambridge Advanced Learners dictionary (2008) defines risk as “the possibility of something bad happening”. Webster’s (2008) dictionary defines risk as “The possibility of loss or injury”.

In 1921, Frank Knight (1921) defined risk as “measurable uncertainty” stating that in “the distinction between the measurable uncertainty and an unmeasurable one we may use the term ‘risk’ to designate the former and the term ‘uncertainty’ for the latter”. Holton (2004) suggests that Knight’s definition is flawed in that it does not address exposure in his definition. Risk entails uncertainty and exposure, the potential of both, and can therefore be defined as “exposure to a proposition of which one is uncertain”. Knight’s definition has only addressed the uncertainty aspect. Take the following scenario for example: if a person bets their house on the spin of a roulette wheel, there is a risk that they will lose their house. If a person never exposes himself or herself to gambling, then there is no risk that they will lose their house in this way. Therefore exposure is an important aspect of risk. By the same token, if a person is certain of a scenario e.g. by digesting a certain poison they will kill themselves, then there is no element of risk, for the outcome is known.

Felix Kloman (2000) argues that risk must carry a connotation of “harm”. What if the proposition defined earlier is in fact a positive one? As a consequence, Kloman defines risk as “the compound estimate of the probable frequency, probable severity and public perception of harm”. This definition would appear to be in line with the widely accepted “combination of the probability of occurrence of harm and the severity of that harm” as defined by ISO standards (ISO/IEC 1999).

Risks exist in everyday life and are a direct result of the scenarios we expose ourselves to each day and the uncertainties associated with those scenarios. Risk also exists for businesses across all industries and is measured in accordance with different contexts depending on the industry in question. For example, the financial industry may measure risk as the actual return on an investment proving substantially lower than expected. A farmer may measure risk in terms of the damage that may occur to his crops from storm damage or insect damage. The medical device industry may measure risk as the possibility of a failure in their device causing harm to a patient. Regardless of the industry, if risk is not identified, measured, and controlled as part of the business, and depending on its severity, there may be loss in terms of human life, damage to the environment, litigation, the market withdrawal of products and other severe financial impacts.

Risk Management

Risk management is an attempt to identify, measure, mitigate and control risks. In a general sense, risk management in organisations can be seen as the protection of the physical, financial and human interests of the organisation in addition to its end-users, the general public and the environment towards which the organisation's outputs are geared. The key word here is "protection" against risk. Governments across the world are aware of the need to manage risk and as a consequence many governmental and associated agencies exist to provide support in this area across the different industry sectors (EC 2008; EFSA 2008; HSA 2008; HSE 2008; IHCP 2008; NEA 2008; USDA 2008). The purpose of these agencies is the protection of the public and public business. Perhaps the best definition of risk management the researcher came across is the following: "a discipline for living with the possibility that future events may cause harm" (Kloman 2000).

Software risk management within medical device companies is a critical area. Failure of the software can have potentially catastrophic effects, leading to the injury of the patients, or even death. It is therefore no surprise to find that regulators throughout the world are penalising medical device manufacturers that do not devote sufficient attention to the areas of hazard analysis and risk management throughout the software lifecycle. If a medical device company fails to comply with the regulations of a given country, they in effect, surrender their legal right to market their device in that country. It is in everybody's best interest that the medical device manufacturer gets it right.

With the availability of so many different standards, regulatory guidance papers and industry guides on software risk management, however, the task of collating this information into a usable model is a daunting one. This research seeks to extract the most important software risk management concepts from a number of accepted standards and guides in the medical device industry in addition to the broader software engineering community - and to present them as a generic usable model for medical device software.

Research Question and Objectives

"Can the development of a new Software Process Improvement (SPI) Risk Management Capability Model (RMCM) for the medical device domain assist medical device companies in improving their software risk management practices and put them on the path to regulatory compliance?" To answer this question, the research has been divided into two distinct, yet equally important objectives. The first objective is the development of the RMCM. The RMCM is geared towards companies who produce software for the medical device industry. The RMCM provides a method for medical device software-producing companies to assess and improve their software risk management capability. This model also seeks to promote SPI practices within medical device software risk management in order to increase the effectiveness and efficiency of software risk management within the regulated medical device environment. The second objective is the evaluation of the model, through trialling it in an Irish medical device company which produces both embedded and desktop application software for their devices.

Research Contributions

This research has resulted in a range of different contributions. An extensive literature review is presented which highlights the need for an SPI model for the medical device software industry. A new model is developed as part of the research and is evaluated in the context of an Irish medical device company producing medical device software. This research investigates how current medical device regulations can be improved through the adoption of practices from existing formal SPI models. The research also highlights how the software risk management practices associated with existing SPI models are not comprehensive enough in their own right to satisfy medical device requirements.

This research also contributes to learning through:

- The publication of the knowledge gathered during the research (Burton *et al.* 2006; Burton *et al* 2008, McCaffery *et al* 2008);
- The presentation of the model to a steering group of Irish medical device companies;
- The development of a more comprehensive software risk management process for the client company in which the research was performed and the sharing of lessons learned during the research.

Outline of Volume Structure

The following is an outline of how the research has been structured and presented in this volume:

- **Chapter 1: Introduction.** The volume commences with a high-level introduction to the research domain and the critical nature of the research. The research question is then set out in conjunction with the primary two objectives of the research, objectives which are explored further in Chapter 0.
- **Chapter 2: Literature Review.** The chapter commences with a definition of risk and risk management and the applicability of risk management to the software industry. This is followed by an overview of the emergence of software risk management within the software engineering community - from its early days to the development of formal SPI models. A discussion is presented on how formal models have been adapted by other regulated industries. A history of the major medical device software risk management standards and guidance papers is so presented. Finally, a critical review of the related research is presented.
- **Chapter 3: Research Methodology.** The chapter commences with a definition for the *risk management capability model* and follows with a presentation of the client organisation in which the research occurred. The research objectives are presented and justification is made for the choice of the interpretive research paradigm. A critical evaluation of the research methods for the interpretive paradigm is performed, as resulting in the chosen method for this research, Action Research. Common criticisms

of the chosen research method are addressed and the method is then presented in the context of the client organisation which participated in the research. The chapter also covers how internal and external validity, reliability and objectivity have been obtained throughout the course of the research. Finally, data analysis and collection during in the research, is also discussed.

- **Chapter 4: Development of the RMCM.** The development and structure of the RMCM are discussed in detail, including the goals, practices and sub-practices of the model. Changes made to the model throughout the research are presented with the reasons and associated impacts of these modifications.
- **Chapter 5: Evaluation of the RMCM.** The evaluation chapter commences with the evaluation of the client's software risk management process at the start of the project. The researcher's intervention and the impact of that intervention is also evaluated. The chapter concludes with various recommendations based on the research findings. These are divided into recommendations related to the RMCM and recommendations related to the client's software risk management procedures.
- **Chapter 6: Summary and Conclusions.** The conclusions chapter summarises how the objectives at the beginning of the research have been met and how the research has provided valuable contributions on several levels. The factors which have contributed to the success of the research are discussed along with its limitations. Finally, a number of recommendations for further research are suggested.

CHAPTER TWO

LITERATURE REVIEW

Introduction

It has been suggested that "we have reached a point where testing as the primary way to gain confidence in a system is impractical or ineffective" (Lee *et al.* 2006). Therefore, software risk management has emerged as a very important method for establishing safe and efficient software. Lack of risk management within software projects can lead to failures and loss at several levels. Barry Boehm (1991) has defined the degree to which a software project may be exposed to failure i.e. risk exposure - as the "probability of an unsatisfactory outcome and the loss to the parties affected if the outcome is unsatisfactory". The two key concepts in this definition are "unsatisfactory outcome" and "loss".

In the software risk management literature, these two terms are viewed from different perspectives depending on the industry in question and the people involved in the risk management process. For risk management literature, which is industry-non-specific, "loss" is typically defined and managed in terms of schedule and budget (Mizuno *et al.* 2001) - with the primary goal of avoiding so-called "death-march" projects (Yourdon 1997). Similarly, when software risk is discussed in the context of the Defence Industry, unsatisfactory outcome and loss may be equated to delays of mission-critical systems, depletion of management reserve and budget overruns which can ultimately impact upon the effectiveness of the armed forces (Chittester and Haimes 1993). Literature in the defence forces industry focuses on risk management from the perspective of reducing over-runs in performance, support, cost and schedule (Williams *et al.* 1999). Risk is reduced by increasing performance through the utilisation of technical, management and other human resources effectively in order to achieve programme goals (Boehm 1991; Loveland Link *et al.* 1999).

Literature in the Automobile and Space industries takes a broader view of risk since the systems are required to deliver many functions in a dependable and safe manner, while also keeping costs low (Fröberg 2006). Loss in these industries is viewed from several different perspectives including budget, project delays and overruns, quality, and also in terms of safety of the end-users of the systems and associated mechanics/electronics.

Within the Medical Device industry however, risk is discussed in the literature primarily with the aim of managing software risk from a safety perspective. The medical device industry and its associated regulators view unsatisfactory outcome and loss in terms of loss of life, injuries to users or bystanders or in terms of damage to the environment. Schedule and budget over-runs do not concern the regulators. Their job is to protect the public from faulty software which may be placed into medical devices thereby reducing the risk of potential injury (FDA 2008*a*). The key point here is that the focus and targeted risks found within risk management can, to a large degree depend on the industry in question (Table 2.1). As this research is focussed on the medical device software domain, it produces a risk management model aimed at satisfying the requirements of the regulators of this industry. This risk management model is therefore geared towards improving the safety of medical device software.

Table 2.1 – Risk management by industry sector based on literature review

Industry Domain	**Focus of Risk Management**			**Drivers of Risk Management**
	Schedule	**Budget**	**Safety**	
General	Primary	Primary	Secondary	Industry
Defence	Primary	Primary	Secondary	Department of Defence
Space	Spread Evenly			Industry and regulators
Automobile	Spread Evenly			Industry
Medical	Secondary	Secondary	Primary	Regulators

Evolution of Software Risk Management

Early Software Risk Management Literature

Software risk management comprises several core practices irrespective of the industry domain in question. These include the identification, analysis, planning, tracking and control of risks (Automotive SIG 2005; Boehm 1989; Boehm 1991; Cass *et al.* 2001; Higuera and Haimes 1996; ISO 2007; ISO/IEC 1998*a*). Early research and literature such as Barry Boehm's paper (1991) and tutorial (1989) on "Software Risk Management: Principles and Practices" is devoted to the discussion and breakdown of risk management practices into more detailed sub-practices. Tools and techniques for risk identification, analysis and control are discussed by Boehm (1987), including the use of decision trees and the creation and management of a top-10 risk list which focuses on the critical success factors of the software projects. This ensures that the important issues get analysed and addressed thus reducing the likelihood of the projects coming in over-budget and over-schedule.

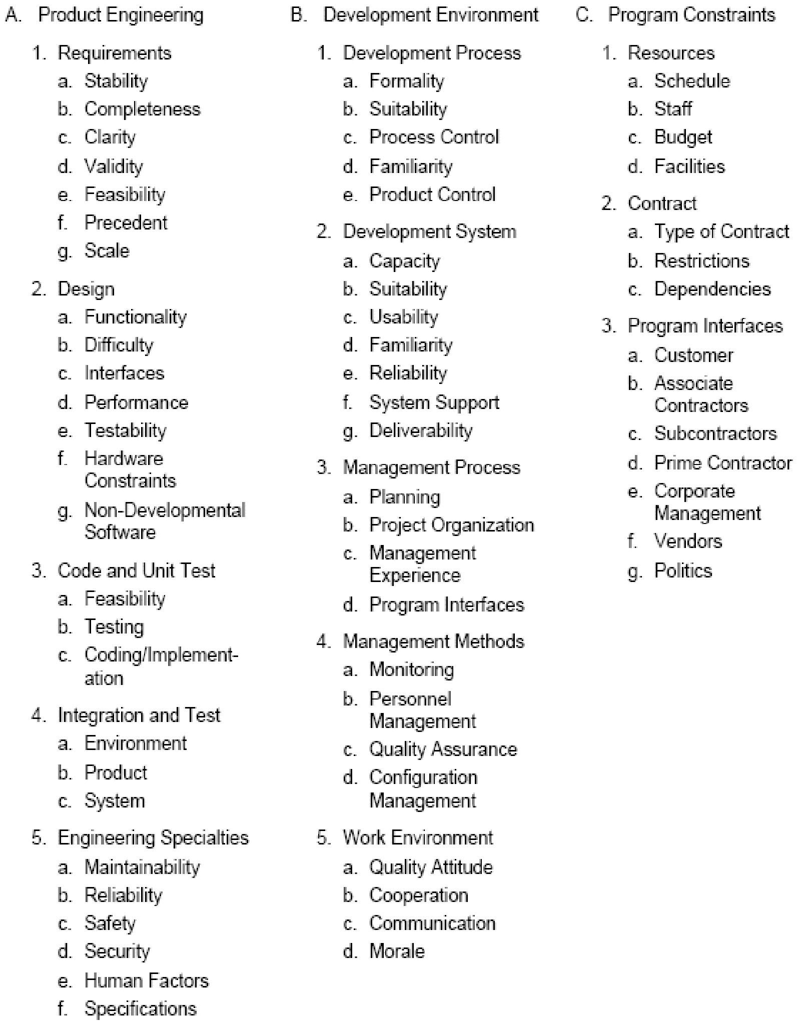
Other early researchers expanded upon some of the sub-practices identified by Boehm, including risk identification and analysis, through the introduction of specialised techniques such as FMEA, FTA, Event Tree Analysis and others (Bell 1989). It is evident from this early literature that risk management was seen as a very necessary part of software systems development and management. With complex systems being built, some of which were expected to run for a long time, industry was aware that there was "no silver bullet" (Brooks 1987) for completely eliminating all potential errors from complex systems. The software industry had to assume, therefore, that whatever could go wrong with complex software systems, would. This would be as a consequence of unexpected use by end-users and various failures and causes external to the system. Hence the focus turned to how potential errors could be reduced through the up-front identification, analysis, planning, tracking and control, and the introduction of software risk management practices. The risk management practices discussed by Boehm go a long way to providing such a framework for risk management.

A Holistic Approach to Software Risk Management

Several authors (Carr *et al.* 1993; Chittester and Haimes 1993; Higuera and Haimes 1996; Williams *et al.* 1999) at the Software Engineering Institute (SEI) in Carnegie Mellon University, explored some key issues as relating to the earliest risk management frameworks. Their primary concern was that these frameworks were being viewed and used from a single "planar" view or perspective. The SEI argued that the main concerns with early risk management frameworks and associated practices was that they did not take a "holistic" approach whereby consideration was given to several different perspectives of software risk management. They acknowledged that risk comes from several different sources and it is only when risks are viewed from these diverse perspectives that a truly comprehensive identification and analysis can be performed. Therefore, the SEI went about producing a framework and risk management literature which focussed on software risks from the following perspectives: process, lifecycle and human (Chittester and Haimes 1993; Higuera and Haimes 1996; Williams *et al.* 1999).

The framework produced by the SEI has its foundations in practices such as Software Risk Evaluation (SRE), Continuous Risk Management (CRM) and Team Risk Management (TRM). TRM involves both the customer and the supplier, whereby both have their own risk management programmes but form a team together to manage project risks. This concept is highly dependent on open communication between the parties and consequently communication is seen as central to the risk management strategy by the SEI. The SRE, as described by the SEI, is "a diagnostic and decision-making tool that provides a robust, clear, and understandable picture of the risks that may affect a project". The SRE is based on what is termed a "Risk Taxonomy" (Carr *et al.* 1993) type questionnaire/interview. The "risk taxonomy" comprises a list of where risks originate; see Figure 2.1 – Taxonomy of Software Development Risks (Carr *et al.* 1993). This is used in identifying and analysing the initial software project risks.

Figure 2.1 – Taxonomy of Software Development Risks (Carr *et al.* 1993)



A. Product Engineering

1. Requirements
   - a. Stability
   - b. Completeness
   - c. Clarity
   - d. Validity
   - e. Feasibility
   - f. Precedent
   - g. Scale
2. Design
   - a. Functionality
   - b. Difficulty
   - c. Interfaces
   - d. Performance
   - e. Testability
   - f. Hardware Constraints
   - g. Non-Developmental Software
3. Code and Unit Test
   - a. Feasibility
   - b. Testing
   - c. Coding/Implement-ation
4. Integration and Test
   - a. Environment
   - b. Product
   - c. System
5. Engineering Specialties
   - a. Maintainability
   - b. Reliability
   - c. Safety
   - d. Security
   - e. Human Factors
   - f. Specifications

B. Development Environment

1. Development Process
   - a. Formality
   - b. Suitability
   - c. Process Control
   - d. Familiarity
   - e. Product Control
2. Development System
   - a. Capacity
   - b. Suitability
   - c. Usability
   - d. Familiarity
   - e. Reliability
   - f. System Support
   - g. Deliverability
3. Management Process
   - a. Planning
   - b. Project Organization
   - c. Management Experience
   - d. Program Interfaces
4. Management Methods
   - a. Monitoring
   - b. Personnel Management
   - c. Quality Assurance
   - d. Configuration Management
5. Work Environment
   - a. Quality Attitude
   - b. Cooperation
   - c. Communication
   - d. Morale

C. Program Constraints

1. Resources
   - a. Schedule
   - b. Staff
   - c. Budget
   - d. Facilities
2. Contract
   - a. Type of Contract
   - b. Restrictions
   - c. Dependencies
3. Program Interfaces
   - a. Customer
   - b. Associate Contractors
   - c. Subcontractors
   - d. Prime Contractor
   - e. Corporate Management
   - f. Vendors
   - g. Politics

The SEI points out that risk management must be continuous, hence the CRM part of the framework. Results are achieved over time, as new risks are identified, existing risk mitigation plans are developed, and risks are also found to have dependencies. The SEI risk management framework is quite comprehensive and covers risk from several perspectives – it therefore has many advantages over the traditional "planer" analysis.

Formal Methods – CMMI® and SPICE™

CMMI® Overview and Software Risk Management

As discussed earlier, the Department of Defence recognised a key risk in software projects with respect to increased costs and quality issues. In the early 1980's therefore they set about establishing the Software Engineering Institute (SEI), located in Carnegie Mellon University, Pennsylvania. The SEI (1991) immediately set about developing a formal software process improvement (SPI) model for software engineering and in August of 1991, it released the initial version of the Capability Maturity Model for Software (SW-CMM). In 1997, development of the CMM model stopped 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 (SEI 2002*a*; SEI 2002*b*) with v1.2 following in August 2006 (SEI 2006). A key change in the CMMI® for Development V1.2 (SEI 2006) is that one document describes both the continuous and the staged models. In Version 1.1 the continuous (SEI 2002*a*) and staged (SEI 2002*b*) representations are detailed in separate documents.

The primary difference between the continuous and staged representations of CMMI is as follows. In selecting the continuous representation, an organisation may choose to focus on either single or multiple-related process areas with the goal of improving capability within these process areas. Different process areas may be progressed at different rates (capability), thereby providing the organisation with maximum flexibility to address these areas they see as critical. In implementing the staged representation of CMMI, organisations must progress through the model - one stage at a time and in a prescribed fashion. The process areas within the model are associated with maturity levels. The organisation may only progress to the next maturity level of the model once they have demonstrated capability for all process areas within their current maturity level. This method of progression through the model ensures that an adequate foundation has been laid at each maturity stage before tackling the more advanced process areas of the subsequent stages.

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" (SEI 2006). A process area typically comprises 1 to 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. Within the continuous representation of the CMMI® model capability levels are used to measure performance with respect to both the specific and generic practices. The levels include: Level 0 - Incomplete, Level 1 - Performed, Level 2 - Managed, Level 3 - Defined, Level 4 – Quantitatively Managed and Level 5 – Optimised.

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 of the staged representation with these 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 Analyse Risks;
  - SP 2.1-1 Identify Risks;
  - SP 2.2-1 Evaluate, Categorise, and Prioritise 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 sub-practices in the CMMI® model. The specific goals (SG) and specific practices (SP) described by the CMMI® are a common approach to risk management, irrespective of the industry in question. However, it is in the sub-practices of the SPs where differences may be required depending on the industry and whether there are any specific regulations governing that industry as discussed later in this chapter in the "Industry Domain Specific Models" section.

SPICE Overview and Software Risk Management

ISO/IEC 15504, also known as SPICE™ (Software Process Improvement and Capability Determination), is an international standard for software process assessment. It was developed by the Joint Technical Subcommittee between ISO and IEC. This group was formed in 1993.

SPICE™ is derived from the ISO/IEC 12207 (1995) standard and, because it was developed after CMMI, it uses many of the ideas of CMMI.

SPICE™ is used in defining and assessing an organisation's capability in the areas of management and the definition of their process structures. This model is therefore broken into both a process and capability dimension. The key process categories include: customer-supplier, engineering, supporting, management and organisation. Similar to CMMI®, the capability level for the process areas are broken into six levels, 0 – 5 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 - Optimised process.

Similar to CMMI®, risk management within SPICE™ (ISO/IEC 1988) is contained within the management process area. The risk management is spread across more than one process category however (Surie 2008) e.g. the human dimension is covered in the human resource management process area 3.2.3; continuous monitoring and performance is covered in the quality management process area 3.1.3, while customer related risks are dealt with in the customer supplier process area 1.1.

Industry Domain Specific Models

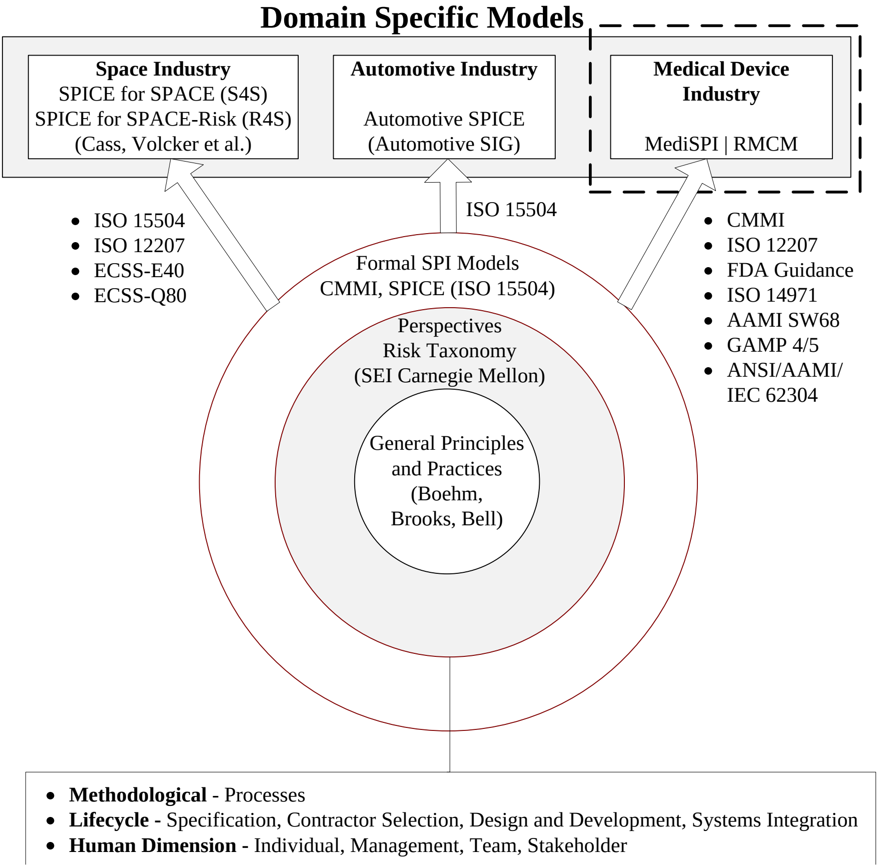
With the creation of formal SPI models such as CMMI® (SEI 2006) and SPICE™ (ISO/IEC 1998*a*), researchers within regulated environments such as the Space and Automotive industries began to investigate how they could utilise these proven SPI Models to improve the practices within their own industry domains.

The main problem with the existing models is that although they are comprehensive in their own right, they do not address all of the regulatory needs and constraints of their industries. Researchers therefore sought to adopt the new practices within these models while also expanding on them to account for regulatory requirements within their own domains. The result of these various strands of research was the production of full SPI models tailored specifically for the Space domain (SPICE for SPACE) (Cass *et al.* 2001) and the automobile domain - Automotive SPICE (Automotive SIG 2005).

Figure 2.2 portrays the emergence of industry domain specific risk management models commencing with the early research as carried out by

Boehm (i.e. in the centre). As discussed, Boehm's research was improved upon by the SEI's holistic software risk management model which, was in turn, followed by the release of formal SPI models such as SPICE™ and CMMI®. The following sub-sections discuss the emergence of the Space and Automobile industry domain-specific models and highlight a similar need for any medical device software risk management model.

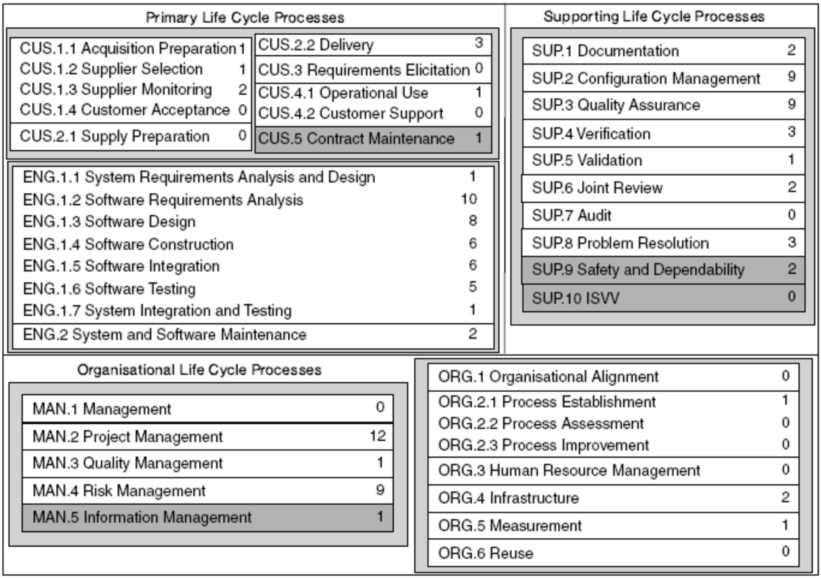
Figure 2.2 – Overview of relevant risk management research and literature areas



Space Industry

In the space industry, Cass et al. (2001) recognised the need for a proven method of assessing software suppliers in the regulated space industry. Therefore they developed a new SPI model through the adoption of existing standards ISO 12207, ISO 15504 (ISO/IEC 1995; ISO/IEC 1998*a*) and combined them with space regulation requirements as published by the European Cooperation for Space Standardisation (ECSS), ECSS-E40 and ECSS-Q80 (1999; ECSS 1996). The research led to a considerable expansion of the existing ISO 15504 model through the introduction of four new processes including SUP 9: Safety and Dependability, SUP 10: ISVV, CUS 5: Contract Maintenance, MAN 5: Information Management (see Figure 2.3). Fifty base practices and sixty new notes were also added.

Figure 2.3 - S4S process figure extracted from work by Cass *et al* (2001). Processes added to the ISO 15504 exemplar model are depicted in grey.



Primary Life Cycle Processes

CUS.1.1 Acquisition Preparation	1
CUS.1.2 Supplier Selection	1
CUS.1.3 Supplier Monitoring	2
CUS.1.4 Customer Acceptance	0
CUS.2.1 Supply Preparation	0
CUS.2.2 Delivery	3
CUS.3 Requirements Elicitation	0
CUS.4.1 Operational Use	1
CUS.4.2 Customer Support	0
CUS.5 Contract Maintenance	1
ENG.1.1 System Requirements Analysis and Design	1
ENG.1.2 Software Requirements Analysis	10
ENG.1.3 Software Design	8
ENG.1.4 Software Construction	6
ENG.1.5 Software Integration	6
ENG.1.6 Software Testing	5
ENG.1.7 System Integration and Testing	1
ENG.2 System and Software Maintenance	2

Supporting Life Cycle Processes

SUP.1 Documentation	2
SUP.2 Configuration Management	9
SUP.3 Quality Assurance	9
SUP.4 Verification	3
SUP.5 Validation	1
SUP.6 Joint Review	2
SUP.7 Audit	0
SUP.8 Problem Resolution	3
SUP.9 Safety and Dependability	2
SUP.10 ISVV	0

Organisational Life Cycle Processes

MAN.1 Management	0
MAN.2 Project Management	12
MAN.3 Quality Management	1
MAN.4 Risk Management	9
MAN.5 Information Management	1
ORG.1 Organisational Alignment	0
ORG.2.1 Process Establishment	1
ORG.2.2 Process Assessment	0
ORG.2.3 Process Improvement	0
ORG.3 Human Resource Management	0
ORG.4 Infrastructure	2
ORG.5 Measurement	1
ORG.6 Reuse	0

Using the newly-derived SPICE for Space capability model, space software companies could determine the gap between target and measured capability and derive a direct list of corrective actions for their processes. The research was then extended to focus specifically on risk management

(Volcker *et al.* 2002). Unlike the medical device sector where risk management focuses on user and patient safety, the space risk management model known as Risk for Space (R4S), focused on quantifying the risks associated with the gaps between target capability as defined by ISO 15504 and assessed capability. The output was a risk-analysis that helps software suppliers select the necessary and minimum process improvements required to meet the organisation's constraints. The authors describe the process as complementary to an organisation's existing risk analysis within software development and it is this researcher's opinion that the described model could in fact be extended and implemented in other industries where formal SPI models have been defined.

Automotive Industry

In Automotive SPICE, risk management is included as a process area, i.e. MAN.5 Risk Management. The process comprises seven base practices which are fundamental risk management practices, as shown below in Table 2.2.

Table 2.2 – MAN.5 Risk Management from Automotive SPICE

Base Practice	Description
MAN.5.BP1	Establish risk management scope
MAN.5.BP2	Define risk management strategies
MAN.5.BP3	Identify risks
MAN.5.BP4	Analyse risks
MAN.5.BP5	Define risk treatment actions
MAN.5.BP6	Monitor risks
MAN.5.BP7	Take corrective actions

It is obvious from the list of base practices that the focus here is on the core practices of software risk management. As discussed earlier in this chapter, these practices are common across the majority of software risk management models. It is in the detail of these practices i.e. the sub practices, that the critical differences lie and where one industry model distinguishes itself from another. The safety related practices within the Automotive SPICE process area are not comprehensive enough to satisfy all medical device regulations.

Take for example the base practice above i.e. MAN.5.BP2 – "Define risk management strategies". Table 2.3 shows a list of sub practices for this base practice as defined in Automotive SPICE. The sub-practices were described in narrative form as opposed to being broken down into bulleted or numbered items and have been extracted and tabulated below for clarification purposes. The table also shows how each sub-practice is also addressed by the medical device regulations and standards.

Table 2.3 – Sub-practices of MAN.5.BP2 – Define risk management strategies

Automotive SPICE Sub Practice	Source of corresponding medical device practice
Define appropriate strategies to identify risk	ISO 14971
Define appropriate strategies to mitigate risk	ISO 14971
Set acceptability levels for each risk or set of risks at a project and organisational level	ISO 14971

The next table, Table 2.4, shows additional sub-practices extracted from the medical device regulations and standards, ones which are not wholly addressed by the MAN.5.BP2 – "Define risk management strategies" practice in Automotive SPICE.

Table 2.4 – Medical Device sub-practices not wholly covered by MAN.5.BP2 – Define risk management strategies

#	Sub Practice	Source of corresponding medical device practice
1	Determine the scope of the risk management strategy. Include those lifecycle phases for which the strategy is applicable	ISO 14971
2	Include a verification plan as part of the strategy	ISO 14971
3	Outline the allocation of responsibilities	ISO 14971
4	Outline the requirements for reviewing the risk management activities	ISO 14971
5	The risk management strategy should include OTS software	IEC 62304
6	Post-production queries and bugs be should analysed	ISO 14971 and CDRH Guidelines

In the table above:

1. Determining the scope of the risk management is addressed by Automotive SPICE in MAN.5.BP1:Establish risk management scope, however, it does not go so far as to request the inclusion of the lifecycle phases to which the strategy is applicable;
2. A verification plan is required by the medical device regulations to ensure a pre-defined plan is in place for the verification of risk mitigations. The verification plan is not mentioned as a specific output work product of Automotive SPICE;
3. For medical device companies, the strategy itself must outline who is responsible for the various risk management tasks. The purpose of this is to provide documented traceability to the person who performed the associated task. This, in turn, must be traceable back to a training log which shows the person has been trained on the specific risk management task. If the person performing a risk management activity is not trained in that area, this in itself can pose a risk in that the activity may not be properly performed. This specific requirement is not addressed by Automotive SPICE;
4. The requirements for reviewing the risk management activities themselves must become part of the medical device company's strategy. If there are issues with or within the risk-management activities themselves then the whole software risk management process could potentially fail;
5. There is no specific mention of examining OTS software as potential sources of risk in the Automotive SPICE risk strategy. The reason for its inclusion in the medical device practices is because a failure in an OTS component can produce an unexpected condition or failure in the medical device software with which it is integrating;
6. Automotive SPICE does not state that post-production queries and bugs should be analysed. This is an important part of the risk management strategy. An analysis of post-production issues and bugs can produce information on unexpected uses of the software, unforeseen environmental conditions in which the software was used and other risk areas that may have been missed during the design and development lifecycle.

As illustrated, the medical device industry introduces new sub-practices and elaborations on existing sub-practices within the risk management process area. The formalities of the medical device software regulations introduce a new complexity and the need for a specific medical device model which completely addresses these requirements in a detailed manner.

Medical Device Industry

An industry specific SPI risk management model has not previously been developed for the medical device industry, i.e. - one which considers both formal SPI methods and the regulatory requirements. Research is currently underway to define an SPI model for the medical device industry, covering those process areas that are key to medical device software (McCaffery *et al.* 2005). The research presented in this volume is an integral part of this wider research area, focussing specifically on software risk and it is therefore key in providing a new SPI risk management model for medical device companies.

The research presented in this volume uses CMMI® (as opposed to SPICE™) as its foundation model for the development of a new SPI risk management model. The main reason for this is that research by others has shown risk management in CMMI® to be more comprehensive than risk management in ISO 15504 (Surie 2008). CMMI® has also been shown to satisfy the risk management models discussed by Boehm and the SEI (Surie 2008). Some ancillary reasons for choosing CMMI® include the fact that the ISO/IEC 15504 is not available as a free download but must be purchased from the ISO whereas the CMM and CMMI are available as free downloads from the SEI website (SEI 2006); The CMMI was sponsored by the US Department of Defence and is therefore the preferred model for US corporations - due to its American roots. Basing the medical device SPI risk management model on the CMMI should therefore favour Food and Drug Administration (FDA) regulated companies; The CMM was created before SPICE™ and had reached critical 'market' mass before ISO 15504 became available.

Medical Device Industry

Introduction

This section discusses software engineering within the medical device industry and associated risk management practices. The reviewed literature has contributed to an understanding of how medical device regulations have evolved to the present day, what medical device companies must do to satisfy the regulations and which practices are key to producing safer and more efficient software, culminating in the creation of the risk management software process improvement model.

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 the local 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 (Lee *et al.* 2006). A simple change in a software component can cause un-foreseen problems in other components within the system, problems which can go undetected unless a robust software design and implementation process exists. The safety of medical device software is dependant on a reliable and solid software process (Eagles 2001) with risk management a core practice for the production of safe and efficient software.

In recent years, the medical device industry has emerged as one of the fastest-growing industries in the world. In 2000, it was estimated that there were over 1.5 million different types of medical device on the market, representing a total value of over €114 billion. These figures from the World Health Organisation which incorporate medical devices containing software - were expected to be in the region of approximately €205 billion in 2006 (WHO 2003). When one considers the importance of medical devices in providing critical patient care and the potential for software failures within associated medical device software, it is inevitable that “accidents” relating to medical devices can and do happen.

Regulatory Bodies and Quality System Regulations

Software quality may be defined and measured in many ways. For medical device companies, one way to define and measure software quality is in terms of the risk of software-containing devices exposing patients, operators, bystanders, service personnel or the environment to hazards. Although medical devices and associated software are developed to increase the well-being of patients, the medical device industry and the governments are faced with challenges. Medical device software may potentially fail to perform an intended function. There is also the