A Systematic Review of Research in Medical Software Certification

by

Qi Huang
BSc, South China Agricultural University, 1996

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science

in the School of Health Information Science

© Qi Huang, 2011
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.
Supervisory Committee

A Systematic review of research in medical software certification

by

Qi Huang
BSc, University of Victoria, 1996

Supervisory Committee

Dr. Abdul V. Roudsari, (Department of Health Information Science)
Co-Supervisor

Dr. Jens H. Weber, (Department of Computer Science)
Co-Supervisor
Abstract

Supervisory Committee
Dr. Abdul V. Roudsari, Department of Health Information Science
Co-Supervisor
Dr. Jens H. Weber, Department of Computer Science
Co-Supervisor

During the past two decades, there has been an explosive volume of software applied in the field of health care. As medical software becomes pervasive in all facets of health care services, the risk of software related patient injuries and patient deaths is also on the rise. To assure the quality of medical software, rigorous validation and verification methods must be employed to analyze all phases of development and final products. In this thesis, a systematic review was conducted to examine and summarize research in the area of medical software certification, which is the primary quality assurance approach taken by regulatory bodies. Key findings indicate that research in the field of medical software certification is sparse, with a limited range of focus and research methodologies. Greater effort using empirical research approaches is necessary for the improvement of current research in medical software certification.
# Table of Contents

Supervisory Committee ........................................................................................................ ii
Abstract ................................................................................................................................. iii
Table of Contents .................................................................................................................. iv
List of Tables ........................................................................................................................ vi
List of Figures ....................................................................................................................... vii
Acknowledgments ................................................................................................................ viii
Dedication ............................................................................................................................ ix

Chapter 1 Introduction ......................................................................................................... 1
  1.1 The importance of software certification in the medical field ....................................... 1
  1.2 Research objectives and rationale to examine research in medical software certification .................................................................................................................. 2
  1.3 Definitions .................................................................................................................... 5
    1.3.1 Software certification ............................................................................................. 5
    1.3.2 Medical software .................................................................................................. 5

Chapter 2 Background ......................................................................................................... 6
  2.1 Brief history of medical software ................................................................................ 6
  2.2 Medical devices and medical software ........................................................................ 8
  2.3 Certification of software ............................................................................................. 9
    2.3.1 Activities of software certification ....................................................................... 9
    2.3.2 Forms of certification ........................................................................................ 10
    2.3.3 Process-oriented vs product-oriented approach of software certification ......... 11
    2.3.4 Domains of the evaluation ................................................................................. 14
  2.4 Brief summary of current major certification bodies, associated approaches and standards .................................................................................................................. 15
  2.5 Introduction of systematic reviews in software engineering ....................................... 20
  2.6 Related work .............................................................................................................. 21
    2.6.1 Grand challenges ............................................................................................... 21
    2.6.2 Literature surveys .............................................................................................. 22

Chapter 3 Research Questions and Rationale ................................................................... 27
  3.1 Research questions .................................................................................................... 27
  3.2 Rationale to conduct a systematic review .................................................................. 28
  3.3 Expectation of results .............................................................................................. 28

Chapter 4 Research Methodology ....................................................................................... 29
  4.1 Source identification .................................................................................................. 29
  4.2 Search protocol ......................................................................................................... 29
  4.3 Search period ............................................................................................................ 31
  4.4 Literature selection ................................................................................................... 31
    4.4.1 Selection language ............................................................................................... 31
    4.4.2 Selection criteria ................................................................................................ 32
    4.4.3 Selection procedure ........................................................................................... 34
  4.5 Data extraction and monitoring ................................................................................ 34
List of Tables

Table 1. Summary of related work with respect to study methods, topic area, aim and methodological limitations ................................................................. 24
Table 2. The name and search period for each database ................................................................. 31
Table 3. The relevant papers covered in the systematic review of medical software certification from all years to present ................................................................. 38
Table 4. A list of secondary relevant papers uncovered in the systematic review of medical software certification ................................................................. 40
Table 5. Important aspects of article categories listed by Montesi el al 2008 ......................... 41
Table 6. Paper categories and associated proper layouts in each category ......................... 43
Table 7. Characterization of relevant papers by the presence of a description of research or non-research activities ................................................................. 44
Table 8. Relevant papers in short paper or full length paper category ......................................... 45
Table 9. Relevant papers in conference paper or journal paper category ........................... 46
Table 10. Regions with corresponding article number and article category of relevant papers found on the topic of medical software certification ................................................................. 55
Table 11. Topics and medical software categories involved in the relevant papers on medical software certification ................................................................. 56
Table 12. Topics of relevant papers and the number of papers in each category encompassing medical software certification ................................................................. 57
Table 13. ID of relevant papers and the number of times they are cited ........................................ 59
Table 14. Feature analysis of relevant papers on medical software engineering ................ 78
List of Figures

Figure 1. Quality characteristics defined in the quality model of ISO/IEC 9126 ............13
Figure 2: Relationships between medical software and other sets of software.................33
Figure 3. Two-phase searching, three-step filtering and final number of relevant papers..37
Acknowledgments

I owe my deepest gratitude to my committee members:

Dr. Jens H. Weber, who provided me support, sound advice and careful guidance from the formative stages to the final draft of this thesis. Furthermore, I am also very appreciative for Dr. Jens H. Weber's contribution to this thesis as a second reviewer.

Dr. Abdul V. Roudsari, who provided me encouragement, prudent advice and feedback on my thesis.

I am heartily thankful to subject librarians Rebbecca Raworth and Katy Nelson, who gave me advice on systematic review methodologies and resources.
Dedication

For my family, who offered me unconditional support, encouragement and love throughout the course of this thesis.
Chapter 1 **Introduction**

This chapter highlights the importance of software certification in the field of health care by providing a brief history of medical software and outlining current software related safety issues. Additionally, this chapter provides a rationale for the present thesis and brief definitions required to frame the objectives of the thesis.

1.1 The importance of software certification in the medical field

Since the introduction of computer software into the field of medicine in the 1950s (Collen, 1994), the number of computer programs deployed in health care has increased steadily (Crumpler & Rudolph, 1997; Munsey, 1995; Munsey, 1995). Today computer software has become essential and pervasive in many facets of medicine (Doi, 2007). At the same time, computer software used in medicine has evolved from programs of simple logical processes to computer programs that are highly complex (Cooper & Pauley, 2006). Examples of complex tasks performed by computer programs currently include controlling safety critical medical devices, monitoring output of patient data from devices, calculating treatment dosage, analyzing patient data, and making risk assessment and treatment plans (Young, 1987). These computer programs in health care bring new opportunities of better patient care; however, they also pose new risks of patient injuries and death (Crumpler & Rudolph, 1997; Munsey, 1995; Wetter, 2008). The Food and Drug Administration (FDA) disclosed that from 2007 to 2009, 260 reports of adverse medical events were related to Health Information Technology (HIT) (Huffington Post, 2010). Among these adverse events, 6 resulted in (Koppel & Kreda, 2009) deaths and 44 resulted in patient injuries (Huffington Post, 2010). Due to the fact that these reports are voluntary,
they may represent a small portion of the adverse events that are related to HIT (Silverstein, 2010). Undoubtedly, flaws in software products used in the medical field can potentially cause patient harm.

As patient injuries and deaths resulting from medical malpractice lead to law suits, one would naturally assume that manufacturers of faulty computer software should be subject to law suits and held accountable if the program causes patient harm. However, contrary to this assumption, through a legal doctrine known as “learned intermediate,” the current legal systems protect manufacturers from liabilities of HIT errors and shift them entirely onto medical professionals (Koppel & Kreda, 2009; Silverstein, 2009b). With this stipulation, manufacturers are virtually “liability-free” and unmotivated to produce quality products at the expense of profits (Silverstein, 2009b). Moreover, this stipulation counteracts the principles of good engineering (Silverstein, 2009a). Under these circumstances, the attention given to and the current approach to quality and safety assurance of computer software in health care are believed to be inadequate (Cooper & Pauley, 2006).

Clearly, software in the health care domain should be subject to the most rigorous quality assurance process. Software certification, as an important measure of software quality assurance, is commonly employed by regulatory bodies in the health care domain.

To contribute towards the quality and safety assurance of computer programs in health care, this thesis through a systematic review summarizes and presents an overview of current research in the certification of medical software.

The subsequent sections of this chapter introduce the research objectives and provide a rationale of the thesis. Brief definitions to provide readers a gist of the thesis are also introduced in the remaining sections of this chapter.

1.2 Research objectives and rationale to examine research in medical software certification

The America Recovery and Reinvestment Act of 2009 gave a huge boost to health care information technology (HIT) by allocating nearly $20 billion to the expansion of health electronic records in th US
health care ("American Recovery Act", 2009). This is one of the many steps of the increased conjoining of information technology and health care in the 21st century. Despite the increasing trend of HIT application in health care, there is substantial disparity between the HIT users and vendors in knowledge about the design, faults, software operations and glitches (Koppel & Kreda, 2009), which has become detrimental to both the soundness of the HIT industry and reduction of patient harm (Bowman, 2010; Koppel & Kreda, 2009). The “hold harmless/learned intermediary” clauses are thought to be a major contributing factor to the knowledge disparity between HIT users and vendors (Koppel & Kreda, 2009). Learned Intermediary is a major defense doctrine used in most jurisdictions of the US. It states that the manufacturer of a product has fulfilled his duty of care by providing all of the necessary information to a “learned intermediary”, who interacts with the product. “Learned intermediaries” are medical experts who make informed and individualized medical judgments based on knowledge of medical practice and patients. According to this doctrine, HIT vendors or manufacturers are not held responsible for errors introduced by their products in patient treatments. “Hold harmless” is an contractual term that can be blindly or unwillingly endorsed by chief information officers (CIO) of hospitals and clinics. It thus absolves the HIT vendors or manufacturers of any liabilities. With the protection of the hold harmless/learned intermediary doctrine, the HIT vendors can craft contractual terms to effectively conceal from users the full knowledge of critical faults. “Gag orders” are private orders placed in contracts by vendors to prevent comments and information from being made public. It is common for medical software vendors to place “gag orders” in vendor contracts to prevent communication among HIT application users about problems with HIT systems. Consequently reports, analysis and resolutions of these issues are precluded. Hold harmless/learned intermediary doctrine and “gag orders” are counterproductive and arguably unethical steps (Koppel & Kreda, 2009; Silverstein, 2009b) that contribute to the lack of incentives among HIT manufacturers to get software right and to the reluctance of health care professionals in increased use of HIT (Koppel &
There are growing concerns around the world over substantial numbers of computer programs related clinical accidents and the lack of attention on quality issues of HIT products. During the HIT adoption and certification workshop of February 25, 2010, Jefferey Shuren, the Director of FDA's Center for Devices and Radiological Health, expressed his worry about HIT related hazards. As stated in his testimony prepared for a government meeting, the voluntary reports of 260 HIT related adverse events, including 44 injuries and six deaths over the last 2 years represent “only the tip of the iceberg” (Schulte & Schwartz, 2010). Similarly, the letter written by Iowa Senator Charles Grassley to 31 US hospitals set forth his concerns about software-related issues and faults in the US health care systems and the lack of attention to these issues (Grassley, 2010). He stated that HIT product problems reported by health care providers are often ignored and dismissed. In addition, he pointed out that the “gag order” prohibits HIT issues from being reported and resolved (Grassley, 2010). Furthermore, the Senator called to question the “hold harmless” clause that renders HIT vendor liability free. The suppression of communication about problematic health care software, the lack of pressure and incentive in the assessment of HIT product quality indubitably leads to the prevailing belief that research activities in this area are insufficient. This thesis project aims to confirm or disprove this notion by quantifying the research activities and evaluating the quality of these activities in the area of medical software certification through a systematic literature survey of research studies on the certification of medical software. The second objective of this thesis project is to summarize research in medical software certification and provide a frame of reference, in order to position future research activities.
1.3 **Definitions**

1.3.1 Software certification

Software certification is defined as a series of software product evaluation activities with the goal of assurance of product conformance to specified standards. Issuing of certifications is the final phase of a process of testing, evaluating, assessing and confirming software characteristics of interest.

1.3.2 Medical software

Medical software is defined in the present thesis as any software intended for use:

1) in diagnosis of disease and other medical conditions,
2) in the cure, treatment, mitigation or prevention of diseases associated with individual patients,
3) intended to affect the structure of any body functions (e.g., software that regulates cardiac pacemaker or brain pacemaker activity).

The present thesis is organized as follows: Chapter 2 provides readers background information on the history of medical software, software certification, current medical software assurance approaches and methodology of systematic reviews in software engineering. The rationale and research questions of this thesis are presented in Chapter 3. Details of research methodology are described in Chapter 4. Chapter 5 presents the findings of this study. Finally, Chapter 6 concludes with a discussion of the limitations of the current thesis, related works and future directions.

**Summary:** The data on software related adverse events from the FDA and relevant literature clearly show that software related safety is one of the causes of patient injuries and death. Thus, certification of medical software, as a primary quality assurance measure, is of utmost importance.

The present thesis contributes to the research of medical software certification by summarizing current research and providing references for future research in the mentioned topic area.
Chapter 2  Background

This Chapter presents background information and context of the present thesis, such as the history of medical software, software certification domains, approaches and activities, and regulatory approaches of medical software in different countries. In addition, this chapter also introduces the methodology of systematic reviews in software engineering.

2.1  Brief history of medical software

A job title called “computer” in the 1940s, documented by Bruce Blum, was a position taken by a person who works with a set of formulas and a calculator. By the 50's a “computer” referred to an electronic device that could carry out mathematical functions such as: subtraction, division, multiplication and addition (Collen, 1994). The advance of information technologies today, however, is not solely due to the evolution of computer hardware technology, but more importantly the computer software that programs computers and makes them applicable in various fields. In the medical field, the first usage of computer software was initiated in the early 1950s by Robert Ledley (Collen, 1994), who used computer programs for dental projects at the National Bureau of Standards in Washington (Ledley & Lusted, 1959). It was not until the 1960s that computer software was used in hospitals for administrative purposes such as billing and organization of services. Although there was an increase in the number of software products in the 70s, the software products used in health care were intended for stand alone computers that did not communicate with each other, and the information gathered by computers in health care organizations was not shared among health care providers (Collen, 1994). The breakthrough of such isolation of health care information came in the 80s, when networking technology made asynchronous and synchronous communication possible between computers in dispersed locations. With the emergence of the worldwide high speed internet and the increased
prevalence of PC, the trend of integration of isolated information in health care organizations emerged in 1990s (Collen, 1994).

Since the earliest application in 1950s, the usage of computer software in health care has grown considerably. First, the number of software embedded medical devices has increased significantly (Alfred, Nora & Luis, 2005; Bassen, Silberberg, Houston & Knight, 1985; Crumpler & Rudolph, 1997; Munsey, 1995). For example, the total value of production for embedded medical device in the US, Japan and Germany were $36 billion, $7 billion and $6 billion for year 2002 (Alfred et al., 2005) and their increase in production values were 15%, 11% and 25% since 1998, respectively. Second, the application domain of medical software has become wider. Contemporary medical devices that contain software are virtually ubiquitous, addressing a continuum of diagnoses and treatments (Jones, Jetley & Abraham, 2010). Thirdly, the content and the complexity of software applications used in medicine has increased substantially. For example, the software embedded in contemporary pacemakers can have up to 80,000 lines of code while state of the art software for regulating infusion pumps can have more than 160,000 lines of code (Jones et al., 2010). As health care organizations are increasingly reliant on software to overcome hurdles of cost, time, geography and complexity (Crounse, 2010), the safety aspects of computer application in health care have accordingly become critical.

Despite the efforts of software developers, software defects exist in all computer programs, for the reason that there are theoretical limits to the minimum number of defects introduced in the development process of software (Peterson, 1996) and medical software is no exception to this rule. Thus, with the widespread use of software in health care, there is a growing concern on safety issues related to software in medicine. Although not all of the software faults cause medical errors, some software defects can lead to severe consequences, such as patient injuries and death. For example, in 1993, software defects in the Therac-25 line of medical linear accelerator caused an overdose of radiation, which resulted in four deaths and left two people severely disfigured (Leveson & Turner,
On December 2009, the FDA received a report of unplanned breakdown of hospital-wide computerized physician order entry system (CPOE) and electronic health record (EHR), which resulted in tardiness in treatments and led to a patient death (U.S Food and Drug Administration, 2010). Inadequate software safety assurance measures is a major contributor of these software errors (Bassen et al., 1985). The FDA’s list of medical device recalls reveals more than 470 health device hazard alerts (Majchrowski, 2009) and 380 medical device recalls that were related to software failure in the United States over the past 30 years (Wallace & Kuhn, 2001). In Oct 2006, there was a recall on Identity pacemakers. It was later found that the cause was a software fault of the battery indicator of the pacemaker (Taft, 2007). In February 2007, a class I recall on Defibtech’s automatic external defibrillator was found to be software related (Taft, 2007). In June of the same year, a recall of 4500 infusion pumps from a manufacturer was caused by a software anomaly that erroneously ceased infusion (McCaffery, Burton & Richardson, 2010). The U.S. Food and Drug Administration (FDA) examined recalls caused by software failures from 1983 to 1991 and estimated that 90% were due to inadequate design and 19% were caused by inadequate change control (McCaffery et al., 2010). This suggests that pre-implementation certification to assure software quality conformity is crucial to the reduction of software related medical errors. A change control in software development process is a formal process used to ensure that changes to a system are introduced in a control and coordinated manner, so that the chance of introducing faults into the system is reduced.

2.2 Medical devices and medical software

Often functioning as component or accessories of medical devices, medical software are considered as “components, parts or accessories” of medical devices and regulated by medical device quality standards in the US, Canada and Europe. However, medical software also have the following assets that set them apart from physical medical devices (Bovee, Paul & Nelson, 2001; Kim, 1993):
1) Comprised of sequences of interconnected logical processes, software “can exhibit discontinuities with jumps and branches of such complexity that repeatability is difficult and impossible to prove” (Murfitt, 1990). By all means, a seemingly simple and small software program may be logically complex. The cardiac-defibrillator is an exemplary software embedded device which can potentially have $10^{45}$ different programmable settings, despite having only a few kilobytes of preset software and less than a gigabyte of software in the external interrogator (Kim, 1993).

2) The tasks that software performs can be of high sophistication and complexity. For example, radiation treatment planning software can delineate tumor contour and plan the radio treatment protocol for a cancer patient (Fraass et al., 1998). Furthermore, the radiation treatment can be verified and delivered by computerized systems to ensure proper delivery (Fraass et al., 1998).

3) The personal choice of designs by a software developer may not be readily revealed by external examination (Kim, 1993).

4) Bugs or flaws in medical software may be triggered only under unexpected conditions or means of use. Examples of software failure related accidents includes: death and injuries due to the malfunction of cardiac defibrillators caused by software defects (Coppess, Miller, Zipes & Groh, 2007; Kaczmarek, Beaulieu & Kessler, 2000) and insufficient or excess volume delivered by infusion systems with software flaws (Mc Caffery et al., 2010).

2.3 Certification of software

2.3.1 Activities of software certification

With the ultimate goal of assessing software quality and making pass/fail decisions, certification of software is a complex process that is comprised of various activities to ensure conformance with specified standards and requirements. Issuing of certifications is the final phase of a process of testing,
evaluating, assessing and confirming software characteristics of interest. The process of certification (Rae, Robert & Hausen, 1994) includes the following activities:

1) Software quality measurement, which is the process that involves verification and validation techniques, tests, static/dynamic analysis, and measurements to determine the quality of the software (process/product)

2) Software quality assessment, which is the process of comparing the actual tests and measurements of the characteristics of interest with the specifications of those characteristics

3) Issuing the certification, which is the procedure by which a certifying body gives written assurance that a software product or process conforms to specified characteristics

2.3.2 Forms of certification

Forms of certification can be classified into three categories, based on the relationships between the manufacturer of the software, the user/buyer and the certification body:

1) Self-certification: when the producer of the software is the certification body and claims that its product conforms to specified standards.

2) Buyer requested certification:

A certification body is specified by the buyer of the software product. In this form of certification, the specified certification body can be affiliated with the buying organization.

3) Third-party certification:

A certification body is independent of both the buyer and the software producer.

The third form of certification is the most common approach taken by regulatory bodies to assess software quality conformance in the medical field. Thus, we concentrate on the third party certification in our systematic review.
2.3.3 Process-oriented vs product-oriented approach of software certification

Current opinions on how to best evaluate software product quality are divided. The two approaches are process-centered and product-centered evaluations. The view that a quality process suggests the production of a quality product forms the conceptual basis of the process oriented approach of certification. Process oriented software certification focuses on the assessment of conformance to established or standard development processes. In contrast to the process oriented view, the product oriented view believes that evidence of adopting a good quality process is not sufficient to guarantee software quality and that only the quality attributes of the software product provide evidence of the software quality itself. Product-centered evaluation stresses techniques of measurement, verification and validation. Currently software certification processes mostly focus on the development process rather than the conformance of the product to specifications (Rae et al., 1994; Wassyng, Maibaum & Lawford, 2010). The section below gives a brief outline of activities involved in the two certification approaches and some of the established quality standards associated with these two approaches:

A) Process oriented approach:

The following activities may be included in process certification (Vermesan, 1998):

1) Identification of hazards and assessment of the risks associated with these hazards in case of software failure.

2) Thorough analysis and review to ensure that the design, development, validation and testing of software adhere to established methodologies.

3) Granting a certification upon the approval of the documented results of the analysis and review process.

At present, the prevailing standards in Europe on quality management system are the International Standards Organization's generic 'Quality Systems' series: ISO 9000 to ISO 9004. It is worth mentioning that these standards only verify that a quality management system is in place and has
been fully documented by the manufacturer, they do not provide assurance of the adherence of software development to the specified procedures (Rae et al., 1994). The more recent UK TickIT and the US Capability Maturity Model (CMM) address this problem by using independent audits to ensure that the procedures are followed (Rae et al., 1994).

Process control and certification are important in the scheme of evaluation of the final product. One can safely say that no complex software can be developed without adherence to established development methodologies based on a quality management scheme. However, it is imperative to understand that process quality does not directly bring to light any information about the product. The drawback of process oriented approach is the absence of proof of the final product quality (Rae et al., 1994).

B) Product oriented approach

As opposed to process certification approach, the product oriented approach assesses the quality characteristics of the final product. In this approach, various products of the process such as specification, source codes, user manuals, technical documented are checked against a predefined set of standards. According to ISO/IEC 9126, software quality may be decomposed into the following 6 “top level” characteristics:

• reliability

• usability

• efficiency

• maintainability

• portability

• availability
Each of the 6 “top level” characteristics are divided into sub-characteristics. These sub-characteristics are then broken down into attributes which can be measured or verified. A quality model of ISO/IEC of the quality characteristics listed above is given in Figure 1. The 6 features of software quality are defined below (International Organization for Standardization, 2001):

**Functionality:** A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy needs.

**Reliability:** A set of attributes that bear on the capability of software to maintain its level of

**Efficiency:** A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.

**Usability:** A set of attributes that bear on the effort needed for use and on the individual assessment of such use by a stated or implied set of users.
Portability: A set of attributes that bear on the ability of the software to be transferred from one environment to an other.

Maintainability: A set of attributes that bear on the effort needed to make specified modifications.

The process of a product oriented software certification may include the following steps (Rae et al., 1994):

1) Identifying important characteristics and defining the quality requirements of a product. The requirements may refer to standards or regulations the product should comply with.

2) Selection of proper metrics to measure the identified important software characteristics.

3) Collection of measurements from each intermediate or final product.

4) Comparison of the collected values of the selected metric against acceptance criteria.

2.3.4 Domains of the evaluation

Medical software comprise a great variety of software applications, ranging from programs that consist of small amount of code and simple logical procedures to stand alone information systems of great complexity. The evaluation of complex medical software systems may include many domains: the technical performance of the software—compatibility with other systems, functionality, usability, reliability, efficiency, portability, upgradability, maintainability, adaptability, safety, security, etc; the professional domain—the impact of the software on professional work, user-friendliness of the software, support to professionals’ needs, and improvement of work procedures; the organizational domain—impact on the work process and organization as a whole, effects on organizational strategy and health services provided, adjustments of the organization needed for implementation, and unexpected negative effects; the economic domain—the cost required to implement the software, the cost to train personnel for the system and to maintain the software; the ethical domain—impact of the software application on the doctor-patient relationship, decision making and data security; and the legal domain—effect of the software system on legal status of patient data and liability.
Although all these dimensions of medical software systems are important, this systematic review focuses on the evaluation in the technical performance domain. In other words, our research interests focus on technical requirements, designs, codes and aspects of the development process related to the mentioned technical domain. Specifically, this study examines knowledge in the literature regarding important technical performance characteristics of medical software, the requirements, constituent measures and the standards of these characteristics, test procedures, evaluation methods and tools, certification strategies and certification schemes.

From this point on, the phrase “medical software certification” used in the present thesis implies “medical software certification in the technical performance domain”.

2.4 Brief summary of current major certification bodies, associated approaches and standards. Software related patient safety is assured in many countries through a variety of administrative and legal measures. In the United States, the FDA started regulation on software related to medical devices as early as 1976 (U.S. Food and Drug Administration, 1976). At that time, only two types of medical software were regulated: software embedded in medical devices and software developed for the purpose of testing medical devices. Software that required “competent human intervention before any impact on human health occurs”, however was exempt from these regulations. An example of medical software in this category is decision support systems. In addition, the 1976 Federal Food, Drug and Cosmetic Act emphasized premarket approval of the mentioned types of medical software. In the late 80's, a series of medical software issues, including the disastrous event caused by a design defect in the blood bank systems (Kim, 1993), triggered some regulatory actions to the 1976 Act by FDA to tighten the regulation of safety-critical software. In 1990, the Medical Device Amendment was enacted to provide significant changes to the 1976 Act in order to remove safety-critical software, such as blood bank software from the exemption of regulation. In addition to premarket approval, the Amendment also emphasized post market surveillance to mitigate the risk of software related medical accidents.
For premarket approval, the FDA combines product oriented and process oriented approaches (Abdeen, Kahl & Maibaum, 2008) to assess the software produce quality, where FDA emphasizes on a well documented and robust quality assurance practice to ensure a “rational” software development process. The most important document used in the FDA's premarket approval review of medical software is “Reviewer Guidance for Computer Controlled Medical Devices” (RGCCMD). The guidance defines what data are required for computer controlled device submission and the three “levels of concerns” to categorize computer controlled devices. With regards to the post-market surveillance, the FDA oversees marketed computer controlled devices through Good Manufacturing Practice (GMP). Along with the assurance of device component acceptability and labellings, GMP requires manufacturers to establish QA programs, documentation for their activities products, updates and revisions. In terms of regulations, the GMP is compatible with the international standards ISO 9001, which is adopted by the European Committee for specification of quality system requirements of medical devices.

In Europe, before a medical device is put on the market, the device is required to be affixed with a CE mark (European Commission, n.d.), which is an indicator of the product's compliance with EU legislation. Medical software were considered intangible and functioned as integral parts of medical devices, when the first draft of European Directives was created in the early 1990s (Klumper & Vollbregt, 2010). It was not until 2007, in the preamble of Directives 2007/47/EC4 (European Parliament and the Council, 2007) that European legislators viewed certain types of software, which are “intended for one or more medical purposes set out in the definition for medical devices”, as medical devices. Accordingly, regulatory requirements for these categories of medical software have been changed. These changes are reflected in the Active Implantable Medical Device Directives (AIMD) and the Medical Device Directives (MDD) (Klumper & Vollbregt, 2010). In the amended AIMD and MDD, stand alone software and accessory software of medical devices are considered
medical devices, if they are intended to satisfy the purposes explicitly defined in AIMD and MDD. As with physical devices, this type of software must be CE marked and subject to their own respective conformance assessment. Examples of this type of software include: dose planning software, pacemaker and decision support systems that support diagnosis. Software that is not related to the core functions of a medical device, but as a component or integral part of the medical device, is not considered a medical device by the E.U. legislator. This type of software does not require a CE mark. However, such software is required to undergo the overall conformance assessment of the medical device of which the software is an auxiliary part. Examples of this group of software include software that regulate the cooling system or power of a medical device. The third group of software, which is intended for administrative or educational purposes in health care settings, is not covered by AIMD and MDD. Along with the FDA, the regulatory bodies in Europe take into account the development process in addition to the pure product-related aspects in the assessment of the medical software. The CE mark certification process requires a manufacturer of medical devices to attest conformity with all relevant New Approach Directives (NAD). However, it is important to recognize that a CE mark, by itself, does not indicate conformity to a particular standard, it only indicates conformity to the legal requirements of E.U. Directives. For most of the medical software, CE marking Directives require ISO 9000 series standards, which are standards for quality system assessment.

In Australia, medical devices are regulated under the federal Therapeutic Goods Act 1989, which is administered by the Therapeutic Goods Administration (TGA). Unlike the classification of medical software in Europe and the US, software that are not accessories, components or integral parts of medical devices are not considered as medical devices by the Australian legislator. Thus, this type of medical software is not covered by the Therapeutic Goods Act. Examples of software in this category include electronic patient records and information technology systems. On the other hand, software that are integral components of or accessory parts to physical medical devices are classified
according to the risk level of the medical device the software associates with. Among software in this category, only the software associated with high risk medical devices is subject to extensive documentary review and quality management system certification, where the process control and design control of the software is assessed. Some low risk medical devices also require quality management system certification, but design control for the device is not covered, let alone design control of the software embedded in it. In other words, only the software associated with high risk devices is subject to process oriented certification prior to supply in Australia (Jamieson, 2001). Examples of this type of software include software in drug infusion systems, active implants, extracorporeal systems and some in vitro diagnostic devices (Jamieson, 2001).

In Canada, medical devices are classified into class I to IV, based on the risk level of the associated medical device (Minister of Justice, 2010). Manufacturers selling II, III and IV medical devices must be registered by a quality registrar accredited under the Canada Medical Device. In addition, medical devices must undergo a quality system conformity assessment (ISO 13485 or ISO 13488) in order to be licensed and deployed (Health Canada, 2009), medical software that are a components of or accessories to medical devices are subject to regulation of the associated medical devices as a whole. However, on August 31, 2009 the Medical Devices Bureau of Health Canada [1] issued an announcement (file number: 09-22095-69; Subject:Classification of Medical Devices Class I or Class II patient management software), which stated for the first time Health Canada has officially classified certain types of patient management software as a class II medical device. As defined in the Food and Drug Act, patient management software used only for the purpose of storage, acquisition, transfer and viewing data or image is considered as a Class I medical device and patient management software with capability beyond the previously mentioned functions is considered a Class II medical device. According to this classification, medical software systems that contain decision supporting functions, manipulate or alter patient data are Class II medical devices and subject to ISO 13485: 2003
(CMDCAS) quality system certification, while Class I medical software is subject to establishment licensing.

Summing up, there are a few salient points that can be drawn from the review of regulatory approaches instituted by the aforementioned countries:

1) In most of these countries, the medical software that is a component of or an integral part of a medical device is not a medical device itself. These software programs are required to undergo software conformity assessment that is part of the overall conformity assessment of medical devices. As medical devices are classified into different categories according to the risk levels associated with these medical devices, so are the embedded software. Thus, this type of medical software must comply to quality standards corresponding to various risk levels.

2) As for stand-alone medical software, recent changes in the current regulatory frameworks of medical software, indicate the trend that stand-alone medical software, whether as a medical device or accessories to medical devices, is required to undergo its own conformity assessment. Recent developments in the regulation of stand-alone medical software in Europe and Canada reflect this trend. In Australia, regulation of medical software lags behind the US, Canada and Europe in that stand-alone medical software systems, such as electronic patient records and expert systems, are not covered by regulatory regimes.

3) Although process oriented software certification does not provide proof of the final quality of products, it is the predominant approach taken in most countries to assure medical software quality. The domination of the process oriented approach over the product oriented approach as medical software conformity assessment is manifested by the quality standards adopted in regulatory regimes. The aforementioned standards such as ISO 9000 standards for CE marks, the RGCCMD for premarket approval, GMP for post-market surveillance, and the ISO 13485:2003 that patient management software are subject to in Canada are all characterized by
emphasis on quality systems that lead to sound development processes that are likely to produce high quality medical software.

4) Network information systems are outside the framework of the afore-mentioned regulatory regimes (Niinimäki & Forsström, 1997).

2.5 Introduction of systematic reviews in software engineering

Systematic review is a research methodology to objectively evaluate, synthesize and summarize empirical results of relevance to a particular research question or area of interest (Brereton et al., 2007). As opposed to the informal and selective citations to reinforce preconceived ideas in narrative literature reviews (Pai et al., 2004), a systematic review includes an exhaustive search of primary studies on focused questions and uses clear eligibility criteria to select studies, critically appraise them and synthesize results according to predetermined methods. Objective aggregation of evidence through systematic reviews can be valuable in that it offers new insights and orientation of future primary studies.

Systematic reviews first emerged in the field of medical science and the methodology of systematic review in this field is sophisticated and well established. However, differences (Biolchini, Mian, Natali & Travassos, 2005) between medical science and software engineering rule out the direct application of this methodology in the software engineering field. One of these major differences is that unlike doctors or patients who may not be aware of the effects of a prescribed treatment/drug, it is difficult to blind software practitioners from the techniques that are to be applied as an intervention. To demonstrate, it is hard for a software architect, who is the administrator of a new software design to be blinded from the new design pattern he is going to use. The second major difference lies in the fact that most of the software engineering techniques have impact on the life-cycle of software, therefore it is difficult to isolate the individual effect of a technique. To explain, when the targeted technique interacts with or is influenced by other techniques or procedures in the development, it is generally
difficult to determine the causal relationship between the targeted techniques and the desired project outcome (Braccini, Fabbrini & Fusani, 1997). The third major difference is that there is no agreed standards on how to conduct systematic reviews in software engineering, thus the results of systematic reviews are fragmented and hard to integrate. The last but not the least difference is that researchers conducting systematic reviews in software engineering do not have technological and scientific support equivalent to the Cochrane Collaboration in the medical field. For these reasons, Kitchenham (Kitchenham, 2004) reformulated the procedure (Sackett et al., 2000) of systematic reviews in the medical field to address the needs of evidence-based software engineering. Nonetheless, the procedure described by Kitchenham et al (2004) was abstract and devoid of detail. In an attempt to address this weakness and facilitate the planning and conduction of systematic reviews, Biolchini et al (2005) proposed a systematic review template with more concrete steps. We use the procedure proposed by Kitchenham (Kitchenham, 2004) as a guideline and also follow the steps of Biolchini et al's template (Biolchini et al., 2005) in the definition and execution of our review protocol.

2.6 Related work

There are several approaches to conduct empirical inquiries on the current body of knowledge of software certification. The discussion in this section will cover some of these approaches with emphasis on literature surveys.

2.6.1 Grand challenges

In order to answer the question of what standards are to be met and how to assure compliance of those standards for medical software certification, the Software Quality Research Laboratory of McMaster University initiated a “Software Grand Challenge” to the software engineering communities. The “Software Grand Challenge” invites software engineers to use formal methods to specify, design and implement a heart pacemaker, which is a medical device that is controlled by embedded critical software. At the same time, the software certification consortium (SCC) through McMaster University
also encourages participants to submit supporting evidence used in certification activities to design an
evidence based, product focused certification process to assess their solutions. Likewise, the FDA
launched the Generic Infusion Pump Project, inviting researchers and software developers to help
develop a set of infusion pump safety models and specifications that can be used to verify safety
properties of infusion pumps.

2.6.2 Literature surveys

There have been two literature survey studies of software certification in the software engineering
literature. Both studies present a timeline and a summary of previous research in certification of
software components. In addition, trends, major changes and directions in the research area are noted
in these studies.

A study by Alvaro and colleagues (Alvaro, Almeida & Meira, 2005) focused on the theory of
research in software component certification. In this study, a time line of the research in the software
certification is presented:

1) Early stage: Mathematical and Test-based Models (1993-2001)
2) Second stage: Techniques and Models based on predicting quality requirements.

Summaries of previous research studies and theories during these two periods are presented in a
chronological manner, along with comments on limitations and highlights on important contributions.
Furthermore, two failures in the initiative of the US government and the IEEE committee on software
component standards are noted. Alvaro and colleagues identify the following two main directions in
the component certification area:

(i) Formalism, concerning the development of a formal way to predict component properties and
building components with fully proved correctness properties.

(ii) Component Quality Model, concerning how to establish a well-defined component quality model
and the component properties that can be certified.
Alvaro et al conclude that “components certification is still immature and further research is needed in order to develop processes, methods, techniques, and tools aiming to obtain well defined standards for certification.” With the objective to provide an overview of the research in software component certification, this literature survey conducted by Alvaro et al does not follow a defined and systematic search protocol. In addition, no link between the individual paper and the conclusions is shown in the survey.

Carvalho and colleagues undertook a literature survey on the research of embedded software component certification, focusing on the practical embedded software component certification experience (Carvalho, Meira, Freitas & Eulino, 2009). However, the literature survey of Carvalho et al is a full reproduction of the survey by Alvaro et al, with the additional information of two PhD theses (Karlson, 2006; Larsson, 2004) published after the survey of Alvaro et al. Thus, this survey by Carvalho et al inherits the same limitations of the survey of Alvaro et al and did not provide much more insight than Alvaro et al.

Apart from the two literature surveys mentioned above, there are a few other studies (Bellini, Pereira & Becker, 2008; Catal & Diri, 2009; Engstrom, Runeson & Skoglund, 2010; Gómez, Oktaba, Piattini & García, 2008; Kitchenham, 2010) that review research on one or more components of the software certification process. The method, topic area, aim and methodological limitations of these studies are summarized in Table 1.
<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Author(s)</th>
<th>Method</th>
<th>Topic area</th>
<th>Aims</th>
<th>Methodological limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS1</td>
<td>A systematic review of software fault prediction studies</td>
<td>Catal 2009</td>
<td>Systematic review</td>
<td>Fault prediction</td>
<td>Mapping study to classify primary studies with respect to metrics, methods and dataset used</td>
<td>Search process is not well defined and classification information of each paper is not shown</td>
</tr>
<tr>
<td>RS2</td>
<td>A Systematic Review Measurement in Software Engineering: State-of-the-Art in Measures</td>
<td>Gomez et al 2008</td>
<td>Systematic review</td>
<td>Measurement</td>
<td>Mapping study to classify primary studies by entities, metrics, methods, focus of measurement, life cycle phrases</td>
<td>Primary studies are not explicitly cited</td>
</tr>
<tr>
<td>RS3</td>
<td>Measurement in software engineering: from the road map to the crossroads</td>
<td>Bellini et al 2008</td>
<td></td>
<td></td>
<td>Identify major concepts, research area and trends in software measurement research</td>
<td>No explicit discussion about the search result and lack of clear link between individual papers and the authors' conclusions</td>
</tr>
<tr>
<td>RS4</td>
<td>What's up with software metrics?-A preliminary mapping study</td>
<td>Kitchenham 2010</td>
<td></td>
<td></td>
<td>Identify trends in software metric research and assess the possibility of using secondary studies to integrate results</td>
<td>No major limitation.</td>
</tr>
<tr>
<td>RS5</td>
<td>A systematic review of search-based testing for non-functional system properties</td>
<td>Afzal et al 2009</td>
<td></td>
<td>Testing</td>
<td>Mapping non function properties, metaheuristic techniques used in testing these non-functional properties, the limitation of these techniques and the fitness function used with respect to each non-functional property</td>
<td>No major limitation, except that search strings are not explicitly defined.</td>
</tr>
<tr>
<td>RS6</td>
<td>A systematic review on regression test selection techniques</td>
<td>Engstrom et al 2010</td>
<td></td>
<td></td>
<td>Identify and classify regression test techniques</td>
<td>No major limitation.</td>
</tr>
<tr>
<td>RS7</td>
<td>A systematic review of security requirements engineering</td>
<td>Mellado et al 2010</td>
<td></td>
<td>Requirements</td>
<td>Identify initiatives and experience in security requirement engineering</td>
<td>Search process and selection of primary study is not well defined.</td>
</tr>
<tr>
<td>RS8</td>
<td>Product metrics for object-oriented systems</td>
<td>Purao et al 2003</td>
<td>Narrative style survey</td>
<td></td>
<td>Compile and organize knowledge about product metrics of object-oriented systems</td>
<td>no defined search process, data selection, extraction and synthesis.</td>
</tr>
<tr>
<td>RS9</td>
<td>Reviewing 25 Years of Testing Technique Experiments</td>
<td>Juristo et al 2004</td>
<td>Testing</td>
<td>Compile and organize the body of knowledge in testing and identify areas that require further research.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS10</td>
<td>What Do We Know about Defect Detection Methods?</td>
<td>Runeson et al 2006</td>
<td>Testing and inspection</td>
<td>Compile and organize current body of knowledge to help practitioners to decide which test method to use and for what purpose</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Together with the literature survey of Alvaro et al (2005), related studies show an trend of improvement in the methodological vigor of the literature survey; all related studies published from 2008 onwards are systematic reviews, which in general have greater methodological rigor than narrative literature reviews. Among the systematic review studies, research methods in RS4 and RS6 are better defined and organized than the others. Thus, the present thesis has most similarity with RS4 and RS6 in terms of methodology. With respect to the scope and objectives, the present thesis has greater degree of kinship with the literature survey of Alvaro et al (2005) in that they both aim at providing an overview of the research in a sub-domain of software certification. However, the present thesis differs from Alvaro et al (2005) in that it has a systematic approach, as opposed to the narrative review by Alvaro et al, to gather, assess and organize evidence in the literature, in order to present an overview of the state-of-art in research of software certification.

**Summary:** Background and context in formation in this chapter can be summarized in the following bullet points:

- Since the first use of software in health care, medical software have been widely deployed and evolved to diverse and highly complex software applications.
- Currently, process-oriented certification approach is the predominant medical software quality assurance approach taken by most of the listed countries.
• In Europe and Canada, the trend of classifying and regulating stand-alone patient management software as medical device has emerged.

• In light of differences between software engineering and medical science, the well-established systematic review methodology in medical science was reformulated and adapted for its application in software engineering.
Chapter 3 Research Questions and Rationale

This chapter elaborates the five research questions addressed in the present thesis, the rationale to conduct a systematic review to answer the proposed research questions and the expected results of the systematic review.

3.1 Research questions

The objective of this systematic review is to gain an overview of the current research in the field of medical software certification by measuring the amount of activity, identifying research topics, research approaches and techniques in medical software certification and estimating the importance of contribution by each primary study. The research questions addressed by this systematic review are:

RQ1. How much research activity is in the area of medical software certification?
RQ2. What issues/topics of medical software certification have been studied?
RQ3. What research approaches, techniques, methods and tools are used by researchers to address these issues?
RQ4. Does this research contribute to practice by providing guidelines or frameworks for medical software certification?
RQ5. What are the limitations of the current research?

Framework is defined as a structured body of knowledge or concepts constituting a view of medical software certification. Thoroughness of the guidelines is indicated by its coverage of issues found in the literature and their dimensions. The impact of guidelines or frameworks is reflected by how frequently they are cited by other papers.

In the result section of this thesis, sub-titles of passages containing information relevant to these questions begin with their identities (RQ1-RQ5).
3.2 **Rationale to conduct a systematic review**

As stated previously in section 1.1 of Chapter 1, certification is the primary quality assurance measure applied by regulatory bodies in the filed of medicine. Thus, research in the area of medical software certification is paramount to the improvement of software related patient safety.

A systematic literature review is a literature review that exhaustively summarizes the literature relevant to one or more research questions. Compared to other form of empirical enquirers that aggregates evidence of current research, such as non-systematic literature surveys or expert surveys, systematic reviews are more rigorous in terms of methodology and consequently provide more comprehensive evidence in an unbiased approach (Pai *et al.*, 2004). Narrative reviews, typically lacking formal and subjective ways to collect and interpret evidence (Pai *et al.*, 2004), often serve the purpose of orienting readers to a field of study. Because of its objectivity, rigor and thoroughness, systematic review is chosen to obtain the most comprehensive information for an accurate assessment of the research in medical software certification.

To search for answers to our research questions, we conducted a preliminary literature search. The results of this search of relevant literature indicated that there is no systematic review that can provide answers to our research questions.

3.3 **Expectation of results**

We expect that through the synthesis and analysis of data extracted from primary studies in this systematic review, we will gain an overview of the current state of research in the area of medical software certification.

**Summary:** This chapter introduced the research questions which center around the topics, the amount of activities and the estimated impact of research in medical software certification. The fact that the current literature is devoid of a literature review justifies the present thesis.
Chapter 4 Research Methodology

To achieve the objective of the present thesis, namely to provide an overview of the past and current research in the field of medical software certification, a systematic review of the literature is conducted. This Chapter presents details of the protocol we followed to conduct the systematic review. This protocol is comprised of five main steps: 1) identification of data source; 2) search of relevant papers; 3) selection of relevant papers; 4) data extraction; and 5) data analysis and synthesis.

The aim of the search in a systematic literature review is to identify as many relevant research papers as possible in an unbiased approach. The rigor of the search is one of the factors that distinguishes systematic reviews from narrative reviews (Kitchenham, 2004). Sections 4.1, 4.2 and 4.3 below outline the search strategies, which allow readers to assess their rigor and fairness.

4.1 Source identification

The search was performed in indexing databases and digital libraries of publishers and organizations of software engineering. Search engines used to uncover primary studies were: IEEE Xplore, ACM Digital Library, Wiley InterScience (computer science area), Science Direct (computer science area) and SpringerLink. In addition to databases of these major publishers, the search was also performed in indexing databases of major scientific and technical literature such as Inspec, Compendex and Computer Science Index. Public search engines for scientific and academic papers, such as Google scholar and Citeseerx, were also searched as sanity checks. Finally, cross-references of the primary studies were examined to reveal additional papers that were not found by search of the indexing databases and digital libraries.

4.2 Search protocol

With limited manpower and time, automated web engines were used to perform searches, as manual searches are less efficient. The basic search string: “medical software certification” was used for the
automated web engines. To ensure thoroughness of searches, synonyms were derived from keywords in the basic search string to form a more comprehensive search string. Particularly, the search string “(medical OR patient care OR clinical or health care) AND (software OR information system) AND (certification OR licensing OR (quality AND (assurance OR assessment OR validation)))” was used in Computer Science Index and all major publisher's databases but Wiley Interscience to reveal studies of our interest. Depending on the organizational models of the corresponding databases, different sets of search strings were also derived from the keywords and the synonyms. For instance, the Wiley Interscience database search engine does not search tables and bullet points embedded in the full text. In order to uncover relevant papers with the basic keywords and associated synonyms presented only in tables or bullet points, we include additional search keywords that are associated with, but not central to medical software certification, in the search string. Accordingly, the search string “(medical OR clinical OR (patient care) OR (health care)) AND (software OR (information systems)) AND ((quality assessment) OR (quality control) OR (quality assurance) OR (conformance testing) OR (compliance AND regulations))” was used in the Wiley Interscience database.

In addition to the differences in organization models of databases, we also used controlled vocabularies to create search strings for indexing databases. Controlled vocabularies are used in the fields of library and information science. These vocabularies are carefully selected lists of words or phrases that represent the subject headings of documents or articles so that these documents can be retrieved from databases. The advantage of using controlled vocabularies is that they reduce ambiguity inherent in human languages. In indexing databases where papers are tagged by controlled vocabularies, search strings comprised of controlled terms and search strings derived from the basic keywords were used. In the case of Inspec and Compendex, the search strings “((medical computing) OR (safety critical software) OR (medical information systems ) OR (medical applications) OR
(decision support systems)) AND certification” was used, instead of the basic search string or search strings derived from it using synonyms.

4.3 **Search period**

<table>
<thead>
<tr>
<th>Database</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Xplore</td>
<td>1884</td>
<td>Present (April, 2010)</td>
</tr>
<tr>
<td>Web of Science</td>
<td>1955</td>
<td>Present (April, 2010)</td>
</tr>
<tr>
<td>ACM Digital Library</td>
<td>1947</td>
<td>Present (April, 2010)</td>
</tr>
<tr>
<td>Wiley Interscience</td>
<td>All dates</td>
<td>Present (April, 2010)</td>
</tr>
<tr>
<td>Computer Science Index (EBSCOhost)</td>
<td>1956</td>
<td>Present (April, 2010)</td>
</tr>
<tr>
<td>Science Direct</td>
<td>1823</td>
<td>Present (April, 2010)</td>
</tr>
<tr>
<td>Inspec</td>
<td>1896</td>
<td>Present (April, 2010)</td>
</tr>
<tr>
<td>Compendex</td>
<td>1896</td>
<td>Present (April, 2010)</td>
</tr>
</tbody>
</table>

For all databases, the search periods were performed from the time of earliest available literature until April 2010. Search time period for each database is listed above in Table 2.

4.4 **Literature selection**

After the potential primary papers are identified, their relevance needs to be evaluated through a predefined selection process. Selection criteria are intended to identify research papers that provide evidence to answer the research questions. This section introduces the criteria and procedures that are followed to conduct the systematic review.

4.4.1 **Selection language**

The systematic review limits the search in English literature that are externally peer-reviewed. Peer-reviewed papers are scholarly publications, for which the submitted manuscripts have undergone high standards of scrutiny by academics or professionals considered to be authorities in the field and capable of providing critical feedback and assessment of the manuscript. Externally peer-reviewed papers are
peer-reviewed papers that have gone through critical assessment by experts other than those in the editorial board.

4.4.2 Selection criteria

The returned papers from automated searches were filtered by a three-step filtration process. The selection criteria for each step of filtration and selection criteria is listed below:

1) Abstract and title filtration by keywords or associated synonyms as defined in section 4.2 of this chapter. Papers that do not have all of the search key terms/synonyms in the title, keyword list and abstract were excluded.

2) Abstract and title filtration by target certification domain, forms of certification and software set.

We include papers that discuss topics in the technical performance domain (see Chapter 2, section 2.4) of certification. In other words, papers regarding certification issues of requirements, constituent measures and the standards of technical performance characteristics, the associated test procedures, evaluation methods and tools, certification strategies and certification schemes are included.

Papers that address issues of third party certification (see Chapter 2, section 2.3) are considered to be relevant. For papers without specification of certification forms, we assume the form of certification is third party certification.

In addition, only papers with targeted software set that falls within our definition of medical software (see Chapter 1, section 1.3.2) are included. Since our research interests focus on technical requirements, testing methods and techniques and certification schemes pertaining specifically to medical software, we do not review studies with target software set outside the scope of our definition. For example, we do not review papers on certification of
biomedical information systems or papers with target software set overlapping the medical and other fields, such as studies on knowledge-based systems or mission critical systems, unless the addressed issues, techniques, methods or are specifically stated to be applicable to medical software.

Figure 2 illustrates the relations between targeted software set of this systematic review and the examples of other software sets. For papers that do not provide a definition of medical software though is referred to in the manuscript, we assume that the definition is consistent with what is defined in the present thesis.

Figure 2: Relationships between medical software and other sets of software

3) Full-text filtration

For papers of which the domain of certification, forms of certification, targeted software set can not be identified in the abstract and title, full-text was examined, using the same criteria mentioned in step 2. In the full-text filtration, we also assess the applicability of addressed issues to the certification of medical software. Since software certification comprises a broad range of topics, including general software engineering issues, we only include papers that explicitly state that the addressed issue is applicable to medical software certification. For example, papers that discuss software test techniques that can be used in the development
process will not be selected, unless it is indicated that the techniques can also be used for medical software certification.

4.4.3 Selection procedure

Stage I

1) Reject irrelevant papers by examining titles and abstracts of papers identified in initial searches.

2) Review full text of the papers that are not rejected against the detailed inclusion/exclusion criteria.

Stage II

3) Check the reference list of identified primary studies to identify other possible candidate primary studies. Potential relevant studies found through cross reference are subject to the same procedures in the stage I of selection.

4.5 Data extraction and monitoring

The objective of data extraction is to accurately record evidence found in the relevant research papers that addresses the research questions. A clearly defined data extraction protocol helps reviewers to reduce potential bias and increase consistency in the process of collecting information. Sections 4.5.1 and 4.5.2 identify the information elements to be recorded and describe the data extraction procedures followed by this systematic review.

4.5.1 Data extraction

Data extracted from the relevant papers are titles, authors, publication dates, targeted medical software categories, article types, area/country of institutes, topics in relation to software certification, aspects or issues presented, dimensions of these aspects/issues, solutions proposed, tools suggested to facilitate the proposed solutions, results of the studies, limitations found by authors, presence or absence of a framework or guideline and the number of times cited by other papers. It is worth noting that the
number of citations changes quickly and the figures of citations may have changed since April 2010, when we conducted this review.

4.5.2 Monitoring of paper selection and data extraction

In this thesis, the selection of relevant papers is conducted primarily by the first reviewer (Qi Huang). A second reviewer (Jens H. Weber) validates the selection process by selecting relevant studies from a subset of the automated search results. Appendix A provides details of this procedure. This procedure, as opposed to reviewers' selecting relevant studies independently, is more practical when resources and time are limited and the volume of papers returned by automated search is large.

Data extraction of systematic reviews is recommended (Kitchenham, 2004) to be performed independently by two or more reviewers. Accordingly, the data of each primary study in this systematic review was independently extracted by two reviewers to ensure inter-reviewer consistency.

4.6 Data analysis and synthesis

The aim of data analysis and synthesis in a systematic review is to collect and summarize the results of the relevant papers. Statistical analysis is often used to yield quantitative summaries of results (meta-analysis). Due to the scarcity of relevant papers and variation of report protocols in the area of medical software engineering, formal meta-analysis is not used to synthesize results. Nonetheless, quantitative analysis is done to summarize and interpret findings from the relevant papers in this thesis.

4.7 Validation of the systematic review protocol

Prior to the execution of the review protocol, it was discussed with two UVic subject librarians in the fields of medicine and engineering to guarantee the comprehensiveness of the search protocol. One expert from the field of software engineering was asked to review and approve the complete systematic review protocol.
Summary: This chapter introduced the systematic review protocol of this thesis. The data sources were identified. The scope of the thesis, search and selection protocol of relevant papers and the data extraction protocol were presented in this chapter.
Chapter 5 Research Findings

This chapter reports the analysis and findings of this thesis with regard to the proposed research questions. The opening section presents the results of each search and selection step and the resultant relevant papers. Relevant papers are summarized and characterized to create an overview of the research of the topic area of interest.

5.1 Summary of search results
Table 3. The relevant papers covered in the systematic review of medical software certification from all years to present.

<table>
<thead>
<tr>
<th>Paper ID</th>
<th>Title</th>
<th>Authors</th>
<th>Publisher /Journal</th>
<th>Publication date</th>
<th>Aspects/issues presented</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3</td>
<td>Why certification of medical software would be useful?</td>
<td>Forsström, Jari</td>
<td>International Journal of Medical Informatics</td>
<td>1997</td>
<td>Discussion on the changing role of medical software, the need for evaluation and the possible use of drug evaluation models as evaluation models for medical software</td>
</tr>
<tr>
<td>P5</td>
<td>Quantitative evaluation of clinical software, exemplified by decision support systems</td>
<td>Wyatt, Jeremy</td>
<td>International Journal of Medical Informatics</td>
<td>1997</td>
<td>1) 3 major aspects of DSS to be evaluated: structure, performance and impact; 2) Choice of measures of field trials; 3) Possible bias in field trials; 4) Approaches to contain evaluation cost.</td>
</tr>
<tr>
<td>P6</td>
<td>Software quality assessment for health care systems</td>
<td>Braccini, G. and Fabbrini, F. and Fusani, M.</td>
<td>Medical Informatics Europe '97</td>
<td>1997</td>
<td>A quality model was presented for the evaluation of medical software</td>
</tr>
<tr>
<td>P7</td>
<td>A framework for assessing the use of third-party software quality assurance standards to meet FDA medical device software process control guideline's</td>
<td>Bovee, M. W. and Paul, D. L. and Nelson, K. M.</td>
<td>Engineering Management, IEEE Transactions on</td>
<td>2001</td>
<td>A framework to access the difference between FDA requirement, ISO (9001, 9000, 13485) and CMM of SEI,</td>
</tr>
<tr>
<td>P8</td>
<td>A formal methods approach to medical device review</td>
<td>Jetley, R. and Purushothaman Iyer, S. and Jones, P.</td>
<td>Computer</td>
<td>2006</td>
<td>A usage model to enhance the premarket review process of medical software, using the program-slicing technique for forensic analysis of medical software</td>
</tr>
</tbody>
</table>
The first phase of search in the on-line databases yielded over 1700 papers. A possible explanation of the resultant large number of papers is that the multiple dimensions of quality assessment of medical software certification. The second phase of search through cross-referencing yielded 4 secondary relevant studies. Despite the large amount of returned papers, only 14 (0.8%) of them were identified as relevant papers, screened according to the predefined targeted software categories, the domain of certification and the selection criteria. The execution of multi-step filtering of studies and the number of resultant studies are showed in figure 3. Table 3 presents the relevant papers found and identification of these papers.
Some of the relevant papers refer to research studies that are subdivisions of the original research project. To elaborate, some authors of the 14 relevant papers have multiple publications of the same research project and refer to these other publications in the 14 relevant papers. We consider these other publications as part of the original research studies that are in the 14 relevant papers. The papers of these referenced studies are categorized as secondary relevant papers in this systematic review. Secondary relevant papers and identification of these papers are listed in Table 4.
Due to the low amount of relevant papers found in the search, we decided to include all the supporting peer-reviewed evidence that we can find in the literature, in order to obtain the most informative answer our research questions. Thus, the implementation of relevant paper selection deviate from the

<table>
<thead>
<tr>
<th>Table 5. Important aspects of article categories listed by Montesi et al 2008.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major category</td>
</tr>
<tr>
<td>Empirical research study</td>
</tr>
<tr>
<td>Field/Case study</td>
</tr>
<tr>
<td>Experimental study</td>
</tr>
</tbody>
</table>

References: (1) (Basili, Shull & Lanubile, 1999); (2) (Zelkowitz & Wallace, 1998); (3) (Montesi & Lago, 2008); (4) (Pickard, Kitchenham & Jones, 1998);
predefined selection protocol (the guidelines proposed by Kitchenham and Biochili) in that quality criteria was not applied to exclude irrelevant papers. In other words, only the content criteria were used to filter out irrelevant papers.

5.2 Research patterns

Unlike other fields of science and engineering, where research guidelines are well-established, the field of software engineering lacks established research paradigms (Tichy, Lukowicz, Prechelt & Heinz, 1995) and agreed-on criteria (Wieringa, Maiden, Mead & Rolland, 2006) to judge the quality of research papers. Without the consensus of the software engineering community, it is difficult to definitively assess the quality of research reports in the field. Thus, instead of using definitive quality criteria to assess the relevant papers, we characterized the relevant papers in this systematic review by examining their important features that reveal research patterns.

Some classifications are presented by (Zelkowitz & Wallace, 1998) and (Glass, Vessey & Ramesh, 2002) to classify papers in the computing field as a whole. More recently, Montesi et al (Montesi & Lago, 2008) have proposed a paper classification scheme. Instead of the computing field, this classification is intended to outline paper categories in software engineering, thus it is more relevant to the current thesis. However, Montesi et al do not provide practical guidelines on how to apply the classification, and the definitions of the classification of paper categories are not definitive. Thus, instead of applying the classification to the relevant studies in this systematic review, we adopted the important features summarized by Montesi et al to facilitate the analysis of research patterns. Nonetheless, we summarize Montesi's classification scheme since it provides valuable information, which formed the basis of our establishing a list of features describing various research activities that were used in the assessment. Montesi et al developed the list of important features of each paper category based on the following information sources (Montesi & Lago, 2008): 1) relevant publication
on classification; 2) software engineering classification in Journal Citation Report 2006 (JCR); 3) instructions to authors by other relevant journals listed in JRC and 4) the calls for papers of major software engineering conferences. The proposed classification consists of important aspects of each type of paper category in terms of study objectives and approaches. The important aspects of each paper category proposed by Montesi et al (Montesi & Lago, 2008) are listed in table 5.

In addition to important aspects of each paper category, Montesi et al also note proper layout of major paper categories. The proper layout of each major paper category suggested by Montesi et al are summarized in table 6.

Table 6. Paper categories and associated proper layouts in each category

<table>
<thead>
<tr>
<th>Major paper categories</th>
<th>Proper layouts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empirical research paper</td>
<td>(a) the experimental context of the research, (b) the experimental design, (c) the context of the experiment and the collection of data, (d) the analysis, (e) the presentation of results, and (f) their interpretation (1)(2).</td>
</tr>
<tr>
<td>Theoretical research paper</td>
<td>Theoretical papers should describe the problem tackled, the background literature to frame the problem, the solution suggested and the potential or evaluated benefits of the contribution (2).</td>
</tr>
<tr>
<td>Experience report</td>
<td>Soundness of experience report involves description of context (problem domain, size of project, size of system, etc) of experience, showing that the experience is not based on opinion, but observation and analysis (1).</td>
</tr>
<tr>
<td>Opinion/Position paper</td>
<td>Opinion paper may cover general topics from the field, comment on previous papers or explain a technical position. Requirements include brevity, interestingness and convincing arguments (2).</td>
</tr>
</tbody>
</table>

References: (1) (Montesi & Lago, 2008); 2) (Kitchenham et al., 2002)

In the absence of commonly agreed and more detailed quality standards of each paper category and a lack of practical guidelines, we use the proper layout noted by Montesi et al to assist in identifying key features of the studied research papers.

Based on the proper layouts proposed by Montesi et al, We developed a check list for reviewers to identify important features of each paper. Subsequently, research activities and patterns are characterized based on the result of the aforementioned identified features. The feature identifying
check list and its protocol can be found in appendix C. Table 7 shows the results of the characterized features of the 14 relevant papers.

Table 7: Characterization of relevant papers by the presence of a description of research or non-research activities

<table>
<thead>
<tr>
<th>Activities</th>
<th>Description</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>P11</th>
<th>P12</th>
<th>P13</th>
<th>P14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research activities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical research studies</td>
<td>Specific research question</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Empirical Study type</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Study design</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data collection procedure</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Analysis</td>
<td>*#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Results</td>
<td>*#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interpretation/evaluation of the results</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development of methodology/Computation</td>
<td>Problem tackled</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Background literature/conditions to frame the problem tackled</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
</tr>
<tr>
<td></td>
<td>Solution Suggested</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
</tr>
<tr>
<td></td>
<td>Evaluation of solution</td>
<td>*(p,artia l)#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
<td>**#</td>
</tr>
<tr>
<td>Non-research activities</td>
<td>View point/opinion</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
<td>*#</td>
</tr>
<tr>
<td></td>
<td>Experience</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note:

1. a blank cell stands for the absence of an important component in the description of a research or non-research activity.
2. * or # stands for the presence of an important component in the description of a research or non-research activity. * represents an important component identified by the first reviewer and # represents an important component identified by the second reviewer.

3. Cells highlighted in gray reflect differences between the two reviewers.

The result of the characterization shows that 13 of the relevant papers (P2-P14) are devoid of a description of any empirical research studies. Eight relevant papers (P2, P5-P8, P12-P14) contain a description of methodologies, concepts or computation. Five papers (P3, P4, P9-P11) contain opinion only. Among these 8 papers that describe methodologies or theories, one paper contains description of some computation (P2) and the other papers describe methodologies. For the seven papers (P5-P8, P12-P14) that describe methodologies, two papers (P5 and P7) have all of the important components of a description of methodologies and 5 papers (P6, P8, P12-P14) are without the “evaluation of solution suggested” components. Further, one of these papers (P6) is also devoid of information regarding the “background literature/conditions to frame the problem to be tackled”.

Second, to differentiate full length papers from short communications, short contributions and short submissions, etc., we classify the relevant papers into two groups: short papers and full length papers. Full length papers are papers that have more than 5000 words. Table 8 shows the relevant papers in these two groups. There are a total of 9 relevant papers (64.3%) that are classified as short papers and 5 papers (35.7%) that are full length papers.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Paper ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short paper</td>
<td>P2, P4, P5, P6, P8, P10, P11, P12, P14</td>
</tr>
<tr>
<td>Full length paper</td>
<td>P1, P3, P7, P9, P13</td>
</tr>
</tbody>
</table>

Third, since journal papers are subject to more critical evaluation than conference or workshop papers, differentiating journal papers from conference papers and workshop papers allows us to assess their quality and authority. Table 9 shows the relevant papers in the mentioned two categories.
### Table 9. Relevant papers in conference paper or journal paper category

<table>
<thead>
<tr>
<th>Categories</th>
<th>Paper ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference/workshop paper</td>
<td>P1, P2, P6, P10, P11, P12, P14</td>
</tr>
<tr>
<td>Journal paper</td>
<td>P3, P4, P5, P7, P8, P9, P13</td>
</tr>
</tbody>
</table>

#### 5.3 Brief summary of each relevant paper

Based on the data extracted, we provide a brief outline of each of the relevant paper shown in table 2. in the following section. Sub-titles of the passages containing the summaries of relevant papers begin with their corresponding ID (P1-P14)

P1: Zywietz et al “Quality assurance of interpretative electrocardiographs: development of testing services in the European communities-first results” (Zywietz, Mertins & Willems, 1991) and “Stability of computer ECG amplitude measurements in the presence of noise” (Zywietz et al., 1990)

Zywietz et al present the CTS-ECG project launched in June 1989 in Europe. This project aimed to develop systematic and harmonized testing procedures and common standards for qualitative electrocardiographs (CSE). This paper describes the basic CTS-ECG test system set up, which delineates items to be tested, types of test ECGs provided by the CTS test signal generators and the steps in the software performance test. The authors also report the first results of the project, which includes a system testing program written in C-language for protocol preparation, execution of hardware and software tests and examples of a few tests. Zywietz et al conclude that the system testing methods developed seemed to work reliably, although QRS onset and offset detection require adjustments. Regarding the acceptance or rejection standards, the authors note that the acceptance/rejection values are from the CSE study and that they are still subject to discussion.

revealing power of a test case” (Voas, Payne, Michael & Miller, 1993a) and “Experimental evidence of sensitivity analysis predicting minimum failure probabilities.” (Voas et al., 1993a).

Voas et al propose a statistical technique called “software sensitivity analysis” (SST), which is complementary to software testing. The authors believe that this statistical technique, when used together with testing, can improve the estimate of reliability to a much higher precision than using testing alone. In order to predict a program's minimum fault size, the SST, which is based on a three part software fault model (propagation, infection and execution), involves program mutations, data state mutations and repeated execution according to the test distribution. In this paper, Voas et al briefly outline three phases of the proposed software sensitivity testing, which emphasized on faults that can be isolated in a single location of software codes to be examined. Using an automated system that controls the execution and bookkeeping, the authors performed a series of syntactic mutations on each location of the code to be tested. To estimate the probability of alteration of data state that is due to software fault, the mutant programs are re-executed with random inputs. During the execution of these mutant programs, the data states at each location will be instantly compared to that of the execution of original program. A different data state of the mutant program at a location in question indicates an infection. The final phase of the SST determines whether there is an propagation of the previously found infection. During this phase, the data state at the location where infection occurs is changed to random values within a predetermined distribution. The final output of the program resulted from the changed data state is then compared with the output that would have resulted without the change. If the output differs, propagation of the identified infection has occurred. Voas et al recommend using the SST analysis tool PiSCES to perform the proposed statistical technique for applications written in C-language. After summarizing the results of three studies (Voas, 1992; Voas & Miller, 1992; Voas et al., 1993a), Voas et al conclude that it appears that SST can set an upper bound on the likely size of any software fault in a software program.
P3: Forsström “Why certification of medical software would be useful?” (Forsström, 1997)

Forsström discusses various of aspects in this paper, however we will concentrate on the sections related to medical software certification only. In the first half of this paper, Forsström discusses the evolving roles of medical software and provides the background of regulatory frameworks of medical software in the US and in Europe. In the second half of this paper, Forsström suggests using a evaluation model similar to that of human drugs. In Forsström's opinion, it is necessary to conduct clinical trials, just as in human drug evaluations, to demonstrate the usefulness of medical software in health care. He suggests that quality of patient care and cost of health service should be measured to gage the measurable effects of medical software. In addition to premarket clinical trials, the author advocates post market surveillance just as in the human drug industry. In the course of comparison of the evaluations between human drugs and medical software, Forsström highlights some difficulties in evaluation of medical software, such as resolving the conflict between dynamically evolving medical software and the enormous resources required for medical software evaluation, the timing of the evaluation of ever changing medical software and the challenge to obtain statistically significant data to show the usefulness of medical software on rare diseases. Albeit Forsström notes these difficulties, he does not offer advice on how to resolve these issues.

P4: Niinimäki et al “Approaches for certification of electronic prescription software” (Niinimäki & Forsström, 1997)

Just as the Forsström's paper, Niinimäki et al address issues related to software certification. Following a brief description of the Finnish regulatory framework and the electronic prescription-TROPPI project, Niinimäki et al, in section 4 of this paper, suggest technical aspects, such as the integrity, security and reliability of data, the usability of client interface, the reliability of the recovery and support services, the technical functionality, the medical relevance of the knowledge base of the system to be included in the evaluation of medical software certification. With regard to certification
approaches, Niinimäki et al suggests: 1) certification of development process; 2) voluntary evaluation and 3) post-market surveillance. For the certification of development process, Niinimäki et al recommends established process standards such as ISO 9001 and The Medical Devices Good Manufacturing Process. For the voluntary evaluation, the authors recommend evaluation done by a body of voluntary experts in the clinical aspects proposed by (Wyatt, 1994). Lastly, Niinimäki et al argue that report systems similar to that of adverse drug events can be used for post-market surveillance. Additional to the three certification approaches, the authors emphasize that different facets of medical software can be subject to different evaluation and certification protocols and that other means to ensure best possible quality of medical software should be considered, while complete evaluation of complex systems is often impossible. Niinimäki et al conclude that “Evaluation and registration of medical software systems could be analogous to evidence based care where the scientific evidence is used as a quality measurement of a protocol but the lack of waterproof scientific evidence does not prevent us from treating patients.”

P5: Wyatt “Quantitative evaluation of clinical software, exemplified by decision support systems” (Wyatt, 1997)

The discussion of Wyatt on evaluation of decision support systems (DSS) encompasses three dimensions: the structure, the performance and the impact of the system. Regarding the first dimension, Wyatt focuses on the medical knowledge of the system, of which he highlights the importance in evaluating the sources, representation, completeness and implementation. With regard to the second dimension, the author elaborates on formal testing of the performance aspects of the system such as the speed and accuracy of advice and the length of explanation. In this section, the testing of ACORN, a hybrid and Bayesian rule-based DSS is discussed to illustrate the formal testing of system performance. As for the third dimension, Wyatt argues that structural, process and outcome measures should be used to verify system performance and that randomized controlled trial should be used in the
evaluation of DSS to eliminate bias. To explain his views, Wyatt presents in greater detail the ACORN field trial and lessons learned. Finally, Wyatt advises containing the cost for evaluation by titrating the intensity of evaluation needs to the maturity of the system, using surrogate outcome measures and exploiting resources wisely. The author recapitulates this paper by iterating the paramount importance of a randomized controlled trial in the evaluation of clinical software.

P6: Braccini et al “Software quality assessment for health care systems” (Braccini et al., 1997)
In this conference paper, Braccini et al classify software components of health care software systems into five groups according to their functionalities. The five groups are: 1) networking; 2) archiving; 3) scientific; 4) clinical; 5) administration. Subsequent to the description of the classification scheme, Braccini et al outline five quality profiles for software components in the aforementioned categories by applying the ISO/IEC 9126, as described in section 2.3 in Chapter 2 of the present thesis, as a checklist to define the relevance of each sub-characteristics to each of the five categories. Each of the intersection is then assigned a value of a five-level scale (very low, low, medium, high, very high). In spite of the absence of references to relevant empirical studies, Braccini et al believe that the proposed quality profiles can be used as selection criteria to choose a system, pass/fail criteria for certification procedures and criteria to verify that a developed software meets specified targets. It is worth noting that interactions between software components are not taken into consideration in the quality profiles proposed by Braccini et al.

P7: Bovee et al “A framework for assessing the use of third-party software quality assurance standards to meet FDA medical device software process control guideline's” (Bovee et al., 2001)
Bovee et al propose to use third party software quality assurance (SQA) standards as a means to streamline or enhance the current approaches of medical device software (MDS) regulation by FDA. To this end, Bovee et al describe a framework to assess whether ISO 9000 standard series and ISO 13485 (referred as ISO standards below with ISO 9000 series and ISO 13485 implied) combined together or
the level 2 and 3 of the Software Engineering Institute's Capability Maturity Model (SEI/CMM) meet FDA's requirements (combination of federal SQA requirements and FDA guidelines for MDS). This framework compares the third party SQA standards with FDA requirements in four dimensions: 1) process management; 2) requirement specifications; 3) design control and 4) change control. The results of the assessment of the ISO standards and level 2 and 3 of the CMM show that level 2 and 3 of the CMM largely meet the FDA requirements, except in the process management aspects, where FDA requirements can be potentially met by adaption of CMM requirements. Regarding the ISO standards, the authors conclude that they meet or exceed the FDA requirement except in the change control, design control and requirement specification categories, where the FDA requirements are only partially covered.

P8: Jetley et al “A formal methods approach to medical device review” (Jetley, Purushothaman Iyer & Jones, 2006)

The first formal method proposed in this paper includes using formal usage models of which the correctness and adherence to the safety requirement has been rigorously verified to enhance pre-market review of medical device software. To demonstrate the usefulness of this method, the authors present a study on the usage model for infusion pump software. The infusion pump model described by Jetley et al contains a core model that captures the common features of infusion pumps of various types and wrappers to abstract characters of different categories of infusion pumps as sets of input values and events that the core generic infusion pump usage allows. The application of this usage model to laboratory simulation of the infusion pumps software by testing the prototype of a pump software revealed over 400 errors. The second formal method proposed involves combining program slicing technique with model abstraction to improve the efficiency the forensic-analysis of complex medical software. The authors believe that combination of program slicing and model abstraction provides better understanding of the code than using static slicing alone. To show the effectiveness of this
method, Jetley et al report a forensic analysis on the Xio software using both approaches. The result of this study shows that both approaches trace errors successfully, but combining program slicing with model abstraction require 39% less work than static slicing alone.

P9: Classen et al “Evaluation and certification of computerized provider order entry systems” (Classen, Avery & Bates, 2007)

This paper covers the EHR certification approach of Certification Commission for Health Information technology (CCHIT) and the CPOE certification approach of Leapfrog Group, emphasizing the development time lines of both organization and their evaluation frameworks and methodology. In the concluding remarks, the authors call for further refinement and improvement of the certification process and ongoing monitoring and evaluation of implemented systems. Giving the recommendation of future directions, Class et al provide no advice on how to proceed in these directions.

P10: Abdeen et al “FDA: between process product evaluation” (Abdeen et al., 2008)

Abdeen et al claim that the vagueness and inconsistency in FDA’s evaluation approach of medical software leads to confusion in both medical software vendors and engineers. The authors back up this statement by showing evidence of obscurity and impracticality in the guidance document. The evidence include absence of definition of attributes to be measured and associated acceptable values. Further, Abdeen et al compare the risk assessment plan, quality assurance plan and level of concern in another process-oriented evaluation approach, the Common Criteria (CC) for Information Technology Security Evaluation, with those of FDA’s approach to highlights the weak points of FDA’s approach. The authors conclude that resolution of these issues are required by the FDA to ameliorate the current certification function.

In this paper, Rohloff et al propose to specify and enforce certifiable behavior as a measure of certification argument construction for distributed, adaptable medical systems by the way of analysis, architecture and design. The authors briefly outline the four constituent elements of their approach: 1) identify and separate uncertifiable behaviors according to system observables; 2) use interface standards to enhance certification; 3) regulation of component interaction and 4) dynamically constraining behaviors into localized, certifiable operating regions. For each of the element, Rohloff et al present a few different techniques.

P12: Hoerbst et al “A structural model for quality requirements regarding electronic health records - state of the art and first concepts” (Hoerbst & Ammenwerth, 2009)

This paper reported partial results of an ongoing project to develop a certification framework for electronic patient record (EHR). The authors introduce a service and requirement model to explicitly describe the quality requirement of EHR systems in this paper. Although not stated explicitly, Hoerbst et al took the grounded theory approach to gather data required to construct the quality requirement model from literature and expert survey. The resulted data are then structured to develop a structure model to describe the basic structure of a EHR service. This structure model presents a service-oriented view of the EHR, where a specific service may consist of or make use of several other services. Besides the service model for EHR, Hoerbst et al also put forward a EHR requirement model that contains four basic components: quality objectives, generic requirements, implementation and metrics.

To illustrate the concept of this model applied in requirement engineering and quality assurance of medical software, Hoerbst et al present a brief sample of the user interface requirements for a portal service. In the final section of the paper, the authors briefly describe a successful evaluation project where the models were applied. The authors lastly conclude that the models are effective in supporting quality assurance, but the effectiveness on EHR with more complex services needs to be proven by additional testing.
P13: Caffery et al “Medi SPICE development” (Caffery & Dorling, 2010)

This paper describes the initial results of the project of adapting the general software process standards ISO/IEC 15504. specifically for medical software by integrating FDA and the European Council guidelines and associated standards. The general software process standards ISO/IEC 15504 is also known as SPICE. The authors outline three phases of the project, and report results of the first phase, which contains 16 processes from the extension of associated ISO/IEC 15504 processes. The 16 Medi SPICE processes are: 1) risk management; 2) safety management; 3) safety engineering; 4) software requirements analysis; 5) project management; 6) selection and quantification of software tools and libraries; 7) software design; 8) software construction; 9) software integration; 10) software testing; 11) verification; 12) validation; 13) configuration management; 14) problem resolution management; 15) change request management and 16) software and system maintenance.

P14: Pajic et al “A Platform for implantable medical device validation” (Pajic et al., 2010)

A real time Virtual Heart Model (VHM) has been presented to demonstrate as a method to verify correct operation of pacemakers. The VHM, which was implemented on a hardware platform, was developed to mimic the electrophysiological operation of the normal and abnormal functions of a heart. Using this example, the authors claim that VHM demonstrates the need of integrated functional and formal modeling of medical devices with closed-loop interaction of a validated physical model. Further, Pajic et al believe that the presented testing method will facilitate testing in the process of medical device /software certification.
5.4 Answers to research questions to be addressed by this thesis

5.4.1 RQ1-Research activities over time period and regions

The results of this study show that the earliest research activities (Zywietz et al., 1991) in the area of medical software certification began in the early 90's. The number of papers published are almost identical in the previous two decades. There are 6 papers published on this topic in the previous decade and 8 papers in the past decade, which probably suggests little increase in the amount of research activity in this research area. The relevant papers have been published from 8 countries over North America and Europe. 6 papers are from the United States, 1 paper is from Canada and 7 papers are from European institutes. To estimate the amount of research activities in this field, the relevant papers are grouped into two categories based on the findings in table 7. These two categories are papers that

<table>
<thead>
<tr>
<th>Country</th>
<th>Papers that describe or hint research activities</th>
<th>Papers that contain viewpoints/positions only</th>
<th>Total number of papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>4</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Finland</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Austria</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Belgium</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Canada</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ireland</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Italy</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>UK</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
hint or describe research activities and papers that have no bearing of research activities. Table 10 shows the regions together with the corresponding article numbers P12, P13 and P14) pertaining to research activities, 4 (P2, P7, P8 and P14) are from the US and 5 (P1, P5, P6, P12 and P13) are from Europe. Taken together, the results from the analysis of the number of papers over time period and regions may be indicative of equal level of interest in the research topic between Europe and the United States. As a country, the United States has the greatest amount of research activity in the topic area.

<table>
<thead>
<tr>
<th>Software categories</th>
<th>Software types</th>
<th>Topics</th>
<th>Paper ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific software</td>
<td>Electrocardio-graph</td>
<td>Quality standards and test procedures</td>
<td>P1</td>
</tr>
<tr>
<td></td>
<td>Pacemakers</td>
<td>Test techniques</td>
<td>P14</td>
</tr>
<tr>
<td></td>
<td>Electronic prescription systems</td>
<td>Evaluation strategies</td>
<td>P4</td>
</tr>
<tr>
<td></td>
<td>Electronic Health Records</td>
<td>Quality requirement model</td>
<td>P12</td>
</tr>
<tr>
<td></td>
<td>Decision Support Systems</td>
<td>Evaluation strategies</td>
<td>P5</td>
</tr>
<tr>
<td>General software categories</td>
<td>High risk medical software</td>
<td>Test techniques</td>
<td>P2</td>
</tr>
<tr>
<td></td>
<td>Distributed and adaptable medical systems</td>
<td>Evaluation strategies</td>
<td>P11</td>
</tr>
<tr>
<td></td>
<td>Medical Device software</td>
<td>Evaluation strategies</td>
<td>P3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test strategies</td>
<td>P8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirements and standards</td>
<td>P7 and P13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clarity and practicality of certification schemes</td>
<td>P10</td>
</tr>
<tr>
<td></td>
<td>Health care software</td>
<td>Quality model</td>
<td>P6</td>
</tr>
</tbody>
</table>
5.4.2 RQ2-Topics, coverage and issues studied

Table 11 shows the topics in terms of certification related issues and coverage in terms of software categories of the relevant papers.

Based on the type of targeted software, the relevant papers can be divided into two general categories: specific software and general medical/health software. There are more papers addressing issues related to the general category of medical/health software than those addressing issues related to a specific type of medical software, such as electronic health records, physician order entries etc. For each specific medical software, there is only one paper in each sub-category.

Table 12. Topics of relevant papers and the number of papers in each category encompassing medical software certification.

<table>
<thead>
<tr>
<th>Topics</th>
<th>Test procedure</th>
<th>Test techniques</th>
<th>Quality model</th>
<th>Requirements and standards</th>
<th>Certification scheme</th>
<th>Evaluation strategies</th>
<th>Test strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Papers</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Paper ID</td>
<td>P1</td>
<td>P2</td>
<td>P6, P12</td>
<td>P7, P13</td>
<td>P10, P9</td>
<td>P3, P4, P5, P11</td>
<td>P8, P14</td>
</tr>
</tbody>
</table>

In terms of certification issues, the topics of papers in general medical software category and specific medical software category both include testing or evaluation strategies, quality models and certification schemes. There is only one paper that addresses test procedure and one other paper addresses test techniques. These two papers both fall in the category of specific software. The number of papers in each topic is summarized in Table 12.

There are 11 papers (P3-P13) that discuss high level issues, i.e. quality models, requirements and standards, test and evaluation strategies. Conversely, there are only 3 papers (P1, P2 and P14) that deal with lower level topics and provide practical information on test procedures and techniques.
5.4.3 RQ3-Research approaches

The fact that out of nine papers pertaining to research activities, there are eight papers describing methodologies or computation indicates that theoretical reasoning is the primary research approach taken in the area of medical software certification. Although some of the papers (P2, P5, P7, P8, P12) hint empirical studies, the insufficiency of information provided on the empirical studies prevents assessment of the application of empirical research methodologies in the topic area. To give an example, P8 presented only the results of 2 experiments to support the test strategy of using abstract usage model and integral static program slicing. This paper omitted the methodological details including experimental design, conduct and data collection. Similarly, the brevity of P12 and P14 precludes the assessment of the associated empirical studies. In contrast, P7 provides detailed information on the experimental context, design, conduct of experiment, data collection and presentation of results. However, P7 does not document any attempt to validate the results. Thus, we conclude that the relevant papers found in this study do not comprise of adequate information for us to assess the effectiveness of research methods applied and the validity of the correspondent results.

With regard to theoretical studies, Table 7 shows only two papers (P5 and P7) containing all the important components of a proper description of theoretical studies and the other papers (P6, P8, P12 and P14) do not have all of the important components of a theoretical study. This result reflects the fact that the field of software engineering lacks established research paradigms or guidelines. It agrees with findings of Tichy, Walter, Lukowicz, Prechelt and Heinz (Tichy et al., 1995) and the findings of Wieringa, Maiden, Mead and Rolland (Wieringa et al., 2006).

5.4.4 RQ4-Presence of well-established guidelines or frameworks and their contribution in the research

For analysis, we look for papers with high number of citations, as high frequency of citations is an indicator of important contribution to research. Assuming papers with well-established frameworks or
guidelines are cited often by researchers in the same domain, the number of times a paper cited is also an indicator of possible presence of well established frameworks or guidelines. The aforementioned data on citation were obtained from Web of Science. The identification of each relevant paper and the number of times it is cited is showed in Table 13.

*A look at the more frequently cited papers (P4, P5 and P9) showed that these papers, which discuss high level issues such as evaluation strategies and certification schemes, are given more attention. In addition, all of the more frequently cited papers address issues on a specific type of medical software. This may suggest that papers address high level issues on a specific type of software have greater impact on the research of medical software certification than papers on low level aspects or on general medical software categories.*

With regards to the presence of well-established frameworks and guidelines, a close examination of the three papers mentioned did not uncover any frameworks. To elaborate, P9, although cited most often, summarizes two different certification approaches taken by the Certification Commission for Health Information Technology (CCHIT) and the Leapfrog group, without proposing any frameworks or guidelines on medical software certification. With limited section addressing the certification of electronic prescription software in P4, the authors proposed some technical aspects to be included in the evaluation. Nevertheless, no constituent attributes of these technical aspects were identified, no to mention guidelines to measure these attributes. Again, P5 presented some aspects of quantitative evaluation of decision support systems (DSS) and some dimensions of these aspects, but the author failed to identify the quantitative evidence required to support the evaluation. To illustrate, the author proposed three major aspects to be evaluated: the structure, performance and impact.

### Table 13. ID of relevant papers and the number of times they are cited.

<table>
<thead>
<tr>
<th>Paper ID</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>P11</th>
<th>P12</th>
<th>P13</th>
<th>P14</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of times cited</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Among the many performance aspects of a DSS, the author's discussion is limited to the testing of accuracy. In short, we did not find any well-established framework or guideline that is broad in scope in the relevant papers.

5.4.5  

5.4.5 Limitations of research in medical software certification

1. The number of relevant papers found is small and the numbers of papers over years show no trend of increased attention given to the topic. Despite the advance in software technologies and the increasingly important role software plays in the field of medicine, there has not been a corresponding increase in attention to certification of medical software.

2. There were limited empirical research activities identified in the relevant papers and their associated studies, using two different empirical research approaches (survey study and experimental study). The limited amount of empirical research papers and the limited number of research approaches show that the research in this topic is still in the inception stage.

3. Discussion topics are unbalanced between high level and low level issues. There are a total of 11 relevant papers (P3-13) discussing issues of medical software certification at a high level, such as evaluation strategies, requirement models and standards and certification schemes. On the other hand, there are only 3 papers (P1-2, P14) that address practical and low level issues such as test techniques and test procedures.

4. With regards to coverage, although medical software includes a broad range of computer programs, the specific medical software discussed amongst the relevant papers is largely limited to a few types of medical software. There are only 2 types of medical software, namely EHR and electronic prescription systems, covered by 3 papers (P4, P9 and P12) and 2 other medical software that work in tandem with medical devices, namely software of electrocardiograph and pacemakers are covered by 2 papers (P1 and P14). The fact that the papers we found in this
study do not adequately cover the domain of medical software corroborates the possible indication of a lack of attention in this research area.

The above results of this systematic review manifest an inception stage of the research in medical software certification and further research is needed in order to develop techniques, methods, strategies, processes and tools, aiming to establish a standard framework for the certification of medical software.

**Summary:** Research results of this systematic review show that the research effort in medical software certification is inadequate. Current research effort is limited in a few types of medical software and there is a deficiency in diversity of research methodologies, particularly, empirical research methodologies are not utilized.
Chapter 6 Discussion and Conclusions

Related works and limitations of this systematic review are presented in this chapter. Based on the limitations, a few recommendations were made to improve the research in medical software certification.

6.1 Limitations and validity threats

Threats to this systematic review are analyzed in the following order: conclusion validity, internal validity, construct validity and external validity.

Conclusion validity concerns whether there is a relationship between treatment and outcome (Wohlin et al., 2000). One possible threat to this validity is bias in the selection of relevant papers using the predefined exclusion/inclusion criteria. A second possible threat to the conclusion validity is bias in data extraction. In this study, we tried to minimize the first threat by defining detailed exclusion/inclusion criteria. Although we believe that there are adequate details in the selection criteria to provide an assessment of how we reach the final set of relevant papers, the chance that a relevant paper being excluded still exists. Another strategy we used to minimize this threat is having a second reviewer to test the predefined inclusion/exclusion criteria by independently repeating the screening for relevant papers from a random selection of the initial automated search results. The index of inter rater reliability (Cohen Kappa K=.84) for independent screening of relevant papers between two reviewers denotes excellent agreement between the reviewers on the selection of relevant paper using the predefined selection criteria. With regard to data extraction, two reviewers independently extracted data from the relevant papers to ensure consistency of the data, the results of independent data extraction by two reviewers suggests desirable agreement (Cohen Kappa k=.85). Protocols of steps to measure the degree of agreement of reviewers are included in Appendix A and Appendix B.
Internal validity concerns whether there is a causal relationship between treatments and outcomes (Wohlin et al., 2000). To minimize the threat to internal validity, we need to be sure that no other factors, other than treatment in the study, cause the outcome. In other words, the treatment causes the outcome. In this systematic review, treatment of data includes selection of relevant studies and extraction of data. The search domain of relevant studies of the present thesis is limited to peer-reviewed publications. Therefore, the exclusion of unpublished studies (which have significant outcome) and research in proprietary literature or non peer-viewed publications (but have significant impact or contribution to the research of medical software) may attribute to the threat of internal validity. We admit that the inclusion of the aforementioned literature would have increased the internal validity of this study. In addition, we link the conclusions as clearly as possible to the data extracted from the relevant studies, in order to minimize the possibility of threats to internal validity.

Construct validity concerns the agreement between a theoretical concept and a specific measuring procedure or device. Firstly, incorrect exclusion of relevant papers is one possible threat to construct validity. To minimize this threat, we defined a vigorous two-phased search protocol, which uses both keywords and control terms in multiple publisher indexing databases. Secondly, terminology is an other issue where the threat to construct validity arises. Since this systematic review is based on medical software certification, which is a process that consists of steps such as defining quality standards, product testing, measurement, evaluation and assessment, we might miss other relevant papers on any of these topics, but are not specifically aimed for medical software certification.

External validity is related to generalization of the study results outside of the scope of a study (Wohlin et al., 2000). The threats to external validity arise when the time period and the scope of literature where primary studies are selected from are unusual, as opposed to the usual time period and the scope of literature where the results of the study, are generalized for. For example, the results of smoking cessation study done the week after the FDA well-publicized results of the last studies on
smoking and cancer may be different from the smoke cessation study done the week before. In our study, the search of literature spans from the oldest electronic copies of publication available to present. Although we are confident that this study has included most of the relevant peer-reviewed studies, the inclusion of earlier relevant studies that are not electronically available may increase the external validity of this study. Similar to time span, the inclusion of peer-reviewed publication of all languages, instead of English as defined in the selection protocol, may also increase the external validity.

6.2 Future directions

The overall goal of this systematic review is to measure the amount of research activity in medical software certification, summarize research in the area and provide a framework of reference to position future research. As medical software certification encompasses a wide range of topics such as issues related to standards, measurement, testing, evaluation and assessment, the fact that only limited amount of research on a few topics was found and that the absence of well-established guidelines or frameworks clearly shows that the attention given to medical software certification research is inadequate. The insufficiency of research in software medical certification undoubtedly is detrimental to the improvement of software related safety in health care. Today, with the increasing use and reliance of medical software by medical professionals and the increase complexity of medical software, software related risks of patient injuries and death have grown substantially. This continuous trend will incite changes to the status quo of research on medical software certification.

We suggest the following changes in the research on medical software certification:

1. Learn from other mission critical software industry.

In terms of requirements and standards, the field of medicine shares a lot of common characteristics with other safety and mission critical industries where human lives are at stake. In some of these other industries, much more research has been done on software certification at a more mature level, which
resulted in highly robust software assessment frameworks. In order to bootstrap current research on medical software, it is important to use what we have already known about software certification in other safety and mission-critical fields by examining domains such as the military, aerospace, aviation and nuclear power industries. For example, we can enhance the current medical software safety standards and the application of these standards by exploring the applicability and specific ways of applying established mission-critical software standards, such as NASA software safety standards, DO-178B for airborne system and equipment certification, and the guidelines to the application of these standards to the development of software.

2. Development of frameworks.

The results of this systematic review indicate that research on medical software certification is devoid of conceptual frameworks to support empirical inquiries. In order to connect all aspects of medical certification process, we recommend the development of a comprehensive framework, beginning with medical software fault models, safety requirements, standards and certification criteria and ultimately moving towards a certification process. An example of this type of initiative is the project currently undertaken by the EuroRec Institutes to develop a repository of certification criteria for EHR. The EuroRec approach is comprehensive in that there is no limit to the criteria of certification domains of EHR, however, the EuroRec project is still at an early stage and no certification process has been developed by the EuroRec Institute. Similarly to the initiatives in the research of EHR certification, future research efforts of certification of all other medical software can be directed to towards building frameworks to support certification development, testing and assessment.

3. Develop new strategies to contain the cost of medical software certification.

As medicine is a complex and safety-critical domain. The inherent complexity and safety-criticality of software used in this domain makes pre-implementation validation and verification (VV) highly time consuming and costly. On the other hand, the rapidly changing environment of medical software
requires that software certification prior to the application to be done in a timely and economical manner. Thus, it is imperative for research in this area to develop new VV techniques and strategies to contain the cost of medical software certification. For instance, more research can be done on titrating the intensity of VV according to maturity levels of the quality system of the software. Other cost containing strategies include surrogating outcome measures and exploiting resources to reduce the amount of time and cost involved in medical software certification process.

6.3 Conclusions

As computer software proliferates in health care and professionals increasingly rely on software to improve work efficiency and health quality, medical software quality control plays a critical role in patient safety. A systematic review summarizing and presenting the current research status of this field not only supply researchers a framework of reference, but also help identifying new research questions. To achieve these objectives and gather evidence to answer our research questions, we exhaustively searched the research literature. The results of the searches total 14 papers. Among these relevant papers, only 9 of them have baring on research activities. The results of a low number of relevant papers show insufficient attention in this area. Further inadequate report quality of research activities in most of these papers affirms that there is a deficiency in the research of medical software certification.

To our knowledge, this thesis is the first literature survey study that measures the maturity level of and summarizes research in medical software certification in a systematic manner. It identifies trends, methodological patterns and areas of deficiency in the current research. More importantly this thesis confirms the hypothesis that there is lack of research activity in the area of medical software certification, despite the increasingly important role software plays in health care services. Furthermore, results synthesized in this literature survey supply researchers with a summary of existing information about research in medical software certification in a thorough manner. Finally, it
provides references and justifications to researchers who wish to contribute to the research area by focusing on issues that have not been considered or given adequate attention. The results of current thesis demonstrate that research of medical software certification is at the beginning stage. To meet the challenge of patient safety risks posed by our growing reliance on the increasingly complex and variable medical software, we call for greater attention and undertaking in the research of medical software certification.
Bibliography


Medical Devices, Software and Systems (HCMDSS) and Medical Device Plug-and-Play (MD Pnp) Interoperability(173-175). Boston, MA: IEEE, BBN Technol., Cambridge, USA.


Appendix A
Protocol to measure the degree of agreement between reviewers on the selection of relevant papers between reviewers

Step 1:
All of the relevant papers (P1-P14) were mixed with 36 other papers acquired from the automated search of the databases to form the test data set.

Step 2:
A Bibtex file that contains the 50 papers from step 1 was created. Each entry of a paper has the following information of the associated paper:
Title, author(s), publication data, publisher, abstract, journal and full text.
The order of this set of relevant and irrelevant papers are randomized.

Step 3:
The Bibtex file generated in step 2 was given to the second reviewer. The second reviewer then independently group papers into two categories: relevant papers and irrelevant papers, based on the predefined selection criteria

Step 4:
Cohen's Kappa index was calculated to estimate the degree of agreement on the selection of relevant papers between two reviewers, based on the predefined categorization scheme.
Appendix B
Protocol to measure the degree of agreement between reviewers on data extracted from the relevant papers

Step 1:
All of the relevant papers and secondary relevant papers were included to form the test data set.

Step 2:
The first and second reviewers independently identified the software categories covered, the topics in relation to certification and the important features of each article types (predefined by the first reviewer) of each relevant paper.

Step 3:
Cohen's Kappa indexes were calculated between the reviewers on the software categories covered, the topics in relation to certification and the article types.
Appendix C
Protocol to extract data of research patterns

Data to be extracted from relevant papers to answer the research question on research methodology patterns and the amount of research activities.

*Table 14.* Feature analysis of relevant papers on medical software engineering

<table>
<thead>
<tr>
<th>Categories of research activity</th>
<th>Features</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>P11</th>
<th>P12</th>
<th>P13</th>
<th>P14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A) Empirical research study</td>
<td>Specific research question</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Empirical Study type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Study design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data collection procedure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interpretation/ evaluation of the results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B) Methodology/Computation</td>
<td>Problem tackled</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Background literature/conditions to frame the problem tackled</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Solution Suggested</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evaluation of solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C) Experience</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D) View point/opinion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Instructions:**
This table is **not** intended to be used to classify each relevant paper into a paper type. The research study types in the first column is listed together with the second column to supply users information about major features (listed in the second column) of each type of research studies (listed in column one). This form is used to characterize the relevant papers by identifying the presence of major features (second column) in an empirical study or theory paper.
1. Type * in the box to indicate that the paper contains the feature or leave blank if it does NOT contain the feature.
   
   One page (600 words) or more description constitutes the presence of a research activity (A-D). If the description of a research activity is less than 600 words, leave the corresponding column of cells within a research activity (A, B, C or D) blank.

2. Empirical study types (highlighted in blue) can be zero or more of the following categories:
   - O-Observational study
   - F-Field study
   - S-Resurvey
   - E-Experimental study
   - M-Meta analysis

   Either leave blank if you judge the paper not an empirical study. If you judge the study being an empirical study, enter any of the letters O, F, S, E and/or M.

3. Important features of the study types:

   **Empirical study:**
   
   An act or operation for the purpose of discovering some thing unknown or testing a hypothesis, which involves an investigator gathering and performing analysis to determine what the data mean. Empirical research studies can employ different research strategies such as case studies, survey, experimental research, or quantitative research.

   **Theory**
   
   A theory is a set of statements or principles devised to explain a group of facts or phenomena, especially one that has been repeatedly tested or is widely accepted and can be used to make predictions about natural phenomena.

   **Methodology/computation study:**
   
   A methodology is a body of practice, procedures and rules used by those who work in a discipline or engage in an inquiry.
   
   Computation is the procedure of calculating, determining something by mathematics or logic.
   
   Methodology and computation studies should describe the problem tackled, the background literature to frame the problem, the solution suggested and the potential or evaluated benefits of the contribution.

   **View point/opinion:**
   
   An opinion/view point is a belief or conclusion held with confidence, but not substantiated by proof or positive knowledge.
   
   View point and opinion can cover general topics of interests of the field, comment on previously published papers or explain a technical position.

   **Observational study:**
   
   Observational study collects relevant data as the project develops. There is relatively little control over the development process other than using the new technology that is being studied.

   **Field/case study:**
A field/case study describes the application of one or more software engineering practices in an industrial or organizational setting and provide detailed description of the application process and reasons of the application. A distinguishing feature of case/field studies is their descriptive feature.

**Survey:**
A survey summarizes previous literature on a theoretical or systems research topic or explains approaches implemented in commercial systems and answers the question “What is currently known about this area and what does it mean to researchers and practitioners?” It should supply the basic knowledge for new researchers to enter the area, current researchers to continue developments, practitioners to apply the results.

**Experimental study:**
Experimental studies are studies whereby “the research has some control over some of the conditions in which the study takes place and control over the independent variables being studied; an operation carried out in order to test a hypothesis against observation”.

**Meta data analysis:**
Meta analysis studies use statistical techniques to combine results of various empirical studies, aiming to “provide quantitative and objective procedure for combining information from different studies”.

**Experience report:**
Experience reports describe the experience of applying programming techniques to real world problems or how particular techniques, practice or tools were used in a practical setting. Experience reports have less academic content, but more experience-related information. Although experience reports do not have to make novel contribution, they bridge the gap between theory and application.