Improvement of Requirements Engineering Course – Medical Software Case Study

MIRJANA IVANOVIĆ, University of Novi Sad, Faculty of Sciences, Department of Mathematics and Informatics
AMELIA BADICA, Faculty of Economics and Business Administration, University of Craiova
MARIA GANZHA, Systems Research Institute Polish Academy of Sciences and Warsaw University of Technology
COSTIN BADICA, Department of Computers and Information technology, University of Craiova
MARCIN PAPRZYCKI, Systems Research Institute Polish Academy of Sciences and Warsaw Management Academy

Requirements engineering is one of crucial phases in developing any kind of software. For complex and demanding software it is necessary, before starting actual development, to elicit and specify requirements in form of appropriate document that will facilitate communication between developers and users. Requirement Engineering (RE) is, usually, one of master courses during ICT studies. Nowadays, society is confronted with a global aging population, and significant efforts are dedicated to the development of complex healthcare and medical software. In this paper we will discuss some important elements and characteristics of requirement document template and also critical and important lessons for collecting requirements for healthcare and medical software. We also propose some steps for improving RE course that has been delivering for more than 10 years at the University of Novi Sad.

Categories and Subject Descriptors: D.2.1 [**Requirements/Specifications**] ; D.2.9 [**Software quality assurance-SQA**]: —*Management*; J.3 [**Medical information systems**]: —*Life and medical sciences*; K.3.2 [**Curriculum**]: —*Computer and Information Science Education*

1. INTRODUCTION

Remarkable gains in life expectancy have influence on society, as they result in global aging population. Wide range of stakeholders invests serious efforts to develop different kinds of healthcare and medical software, in order to support, on the one hand, independent living of old population and, on the other hand, unhealthy and disabled population. Majority of medical devices, available today, could not function without appropriate software. So-called medical software is a special kind of software, used within medical context, which possess the following characteristics:

(1) standalone software used for diagnostic/therapeutic aim,

(2) ”medical device software” – embedded in specific medical (smart) devices,

Authors address: Mirjana Ivanović, Trg Dositeja Obradovica 4, 21000 Novi Sad, Serbia; e-mail: mira@dmi.uns.ac.rs; Amelia Badica, 13, A.I. Cuza Street, Craiova, 200585, Romania; e-mail: ameliabd@yahoo.com; Maria Ganzha, ul. Newelska 6, 01-447 Warsaw, Poland; e-mail: maria.ganzha@ibspan.waw.pl; Costin Badica, Blvd. Decebal nr. 107, RO-200440, Craiova, Romania; e-mail: cbadica@software.ucv.ro; Marcin Paprzycki, ul. Newelska 6, 01-447 Warsaw, Poland; e-mail: paprzyck@ibspan.waw.pl;

In: Z. Budimac (ed.): Proceedings of the SQAMIA 2017: 6th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications, Belgrade, Serbia, 11-13.9.2017, Also published online by CEUR Workshop Proceedings (http://ceur-ws.org, ISSN 1613-0073)

(3) software that drives medical devices or acts as an accessory to medical devices,
(4) software that supports the design, realization, and testing of medical devices; or provides quality control management for such devices.

Healthcare and medicine domains, nowadays, are facing a number of exclusive challenges in technology adoption. Very important aspect is strict connection to the regulatory issues, dealing with privacy and protection of patient data, thus seriously influencing applications and services. Contrary to some other domains of software development, healthcare and medicine domains absorb and adapts to new technologies more slowly. One of the main reasons is that the regulatory and operational circumstances don't adequately support it. It means that healthcare and medicine, contrary to information-communication technologies (ICT), move at a different tempo, dealing with complex integration of technology into rather inflexible regulatory environments.

In many health jurisdictions, unexpectedly, medical software is usually specified as medical device software (MDS) [Wang et al. 2014]. MDS is predominantly used to analyze patient data and support a diagnosis, or monitor the patient's health. Any drawback in MDS can seriously harm patient's health. So it is important that legislators, and regulatory agencies, obtain specific regulatory standards (usually in form of guidelines) to try to ensure the safety, security and reliability of MDS. These standards heavily support necessity and importance of complete and consistent requirement specifications for medical device software.

There are several worldwide established and recognized institutions (like, in Europe - European Medical Devices Directive or, in USA, FDA - Food and Drug Administration) that deal with important regulations, and related international standards, relevant to the development of healthcare, but also medical (device), software. The particular importance has to be devoted to documenting software and system requirements, for development of healthcare and medical (devices) software. In this context, it is important to introduce such topics and adequate case studie(s) within appropriate Software or Requirement Engineering courses within ICT study programs. This is particularly important since, likely, certain number of students, in their future professional life, will be part of teams that develop healthcare and medical software, or some specific components embedded into medical devices. For example, concerning medical device software it has been stated that: Even slightly erroneous behavior by such a device could lead to a grave incident. The FDA Manufacturer and User Facility Device Experience (MAUDE) database contains a large number of reports on such incidents [FDA 2017].

On the other hand, the European Medical Device Directive MDD 93/42/EEC [1993] looks at the software as a specific type and part of medical, usually smart, devices. It even specified rules under which software can be treated as a medical device. The software influences the functioning of a medical device; or it is intended for the analysis of patient data and use for/by patients to diagnose them. Rapid development of ICT, and other interconnected disciplines, influenced appearance of a new buzz word digital health incorporating: mHealth (mobile health), telehealth and telemedicine, wearable (smart) devices, and personalized medicine. Many stakeholders [FDA 2017] are involved in digital health activities: patients, healthcare practitioners, medical device industry, and recently unavoidable mobile application developers (that develop smart mobile health devices) and of course medical experts. Note that, the FDA has been working in the digital health field, to balance benefits and risks, in order to provide clarity using practical approaches in: wireless medical devices, mobile medical applications, medical device data systems and interoperability, software as a medical device (SaMD) and so on [HHS 2017]. On the other hand plain mobile applications are also promising to help people manage their own health and wellness, and gain access to useful information. The FDA encourages the development of mobile medical applications that:

(1) help patients self-manage their condition and track their health information with simple tools,

(2) provide easy access to information and help patients communicate specific medical conditions to healthcare givers,

(3) enable patients or care givers to interact with Electronic Health Record (EHR) or Personal Health Records (PHR) systems,

but also to monitor the safety and effectiveness of medical devices.

In this paper, we present some considerations and suggestions on specific requirements for healthcare and medical (device) software. Furthermore, we analyze some specifics of the medical domain and how to handle requirements to develop secure and reliable software in this area. Moreover, we will discuss main objectives for requirements template for developing specific type of medical software. All these considerations will be put in the context of a Requirement Engineering (RE) course. Here, possible advantages of introducing specific medical (device) software requirements as case study within such course will be discussed.

In this context, the paper is organized as follows. Section 2, discusses some elements about healthcare and medical software like: Standards, Guidelines and Requirement Templates. Section 3, features a brief overview concerning general objectives for MDS requirements template, as specific type of software in the medical domain. Section 4, as a central one, brings guidelines and propositions for improvement of practical part of the RE course. A Medical Case Study in RE course, as well as specific requirements and possible advantages, are discussed. We conclude the paper with some general comments and observations.

2. STANDARDS, GUIDELINES AND REQUIREMENT TEMPLATES ON MEDICAL (DEVICE) SOFTWARE

Healthcare and medical software development teams, have to use best practices for rapid product development, due to new or constantly changing regulatory requirements. Some best practices must not be neglected like: Patients are primary; Short software development cycles; Don’t try to avoid risks; Continuously validate developed parts/components.

Most important industry innovation, in software technologies and development, have led key government regulators to recognize the crucial importance of standalone medical (device) software products and applications, as well as their incorporation in bigger software architectures and workflows. This has been reflected in significant regulatory changes in multiple E.U. MDD and U.S. FDA guidance documents [Vogel 2011]. The hierarchy of international standards and regulations that are relevant for the development of, first of all, medical (device) software, but also, more generally, healthcare and medical software, is presented in Figure 1.

Slightly different regulations are proclaimed by different Institutions but, generally speaking, they are concentrated on safety, security and reliability issues. U.S. regulation requires that medical devices go through premarket approval [Vogel 2011]. In Canada, to market medical devices, manufacturers must be authorized and approved by the Canadian Medical Devices Bureau for their quality, safety, and effectiveness. For example the Medical Device Directive (MDD) 93/42/EEC (amendment MDD 2007/47/EC) regulates the implementation of medical device software. Also there are some ISO/IEC standards that cover the quality aspects and development process for medical (device) software [Jetley et al. 2013; Rust et al. 2015].

Medical (device) software is, definitely, a specific and unique kind of software, and RE tasks must be considered more seriously and comprehensively than for traditional software development. Considering the format and content of a requirements template (as important and necessary part of requirements specification), there exist multiple possibilities and different published requirements templates [Committee and Board 1998; Robertson and Robertson 2012; El Gamal and Kriedte 1996; Lempia and Miller 2009; Alspaugh et al. 1992; Ahmadi 2006; Lai 2004]. However, as commented in [Gi-

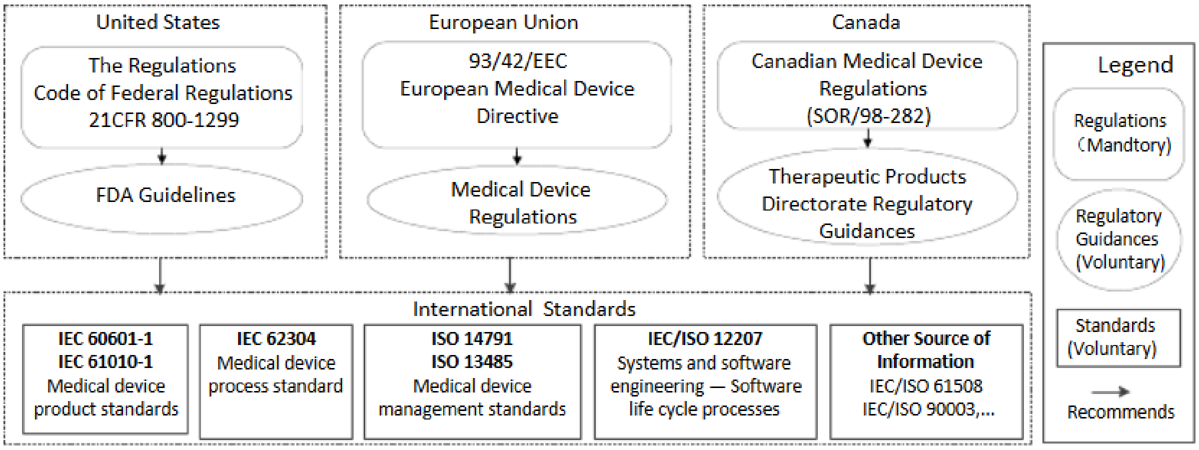
United States
The Regulations Code of Federal Regulations 21CFR 800-1299
FDA Guidelines
European Union
93/42/EEC European Medical Device Directive
Medical Device Regulations
Canada
Canadian Medical Device Regulations (SOR/98-282)
Therapeutic Products Directorate Regulatory Guidances
International Standards
IEC 60601-1 IEC 61010-1 Medical device product standards
IEC 62304 Medical device process standard
ISO 14791 ISO 13485 Medical device management standards
IEC/ISO 12207 Systems and software engineering — Software life cycle processes
Other Source of Information IEC/ISO 61508 IEC/ISO 90003,...
Legend
Regulations (Mandtory)
Regulatory Guidances (Voluntary)
Standards (Voluntary)
Recommends

Fig. 1. Important Regulations and Standards for medical (device) software by [Wang et al. 2014]

akoumakis and Xylomenos 1996], the needs of organizations working on different projects can, and do, vary, and unfortunately these templates are not fully appropriate for the specific needs for medical (device) software, and do not satisfy the requests of quality requirements documents in this area. Even above mentioned institutions that regulate healthcare and medical devices, have not proposed documentation templates for requirements for medical (device) software [Wang et al. 2014].

3. SOME GENERAL OBJECTIVES FOR MDS REQUIREMENTS TEMPLATE

In this context, let us now concentrate on particular type of requirements template devoted to the MDS. Definition, and separate parts of requirements template, are an important part of initial phases in RE. Designing a requirements template for the MDS, which is a highly specific case study, similarly to other application areas, one must start with a list of essential objectives. They have to consider, and specify, which features of the template have to be achieved. MDS regulatory systems consist of legal regulations that consider the public expectations and the dependability and safety of MDSs. There are several standards for setting the requirements of medical devices. However, they do not prescribe the explicit content of the requirements document, as they contain only recommendations concerning the general approach to produce MDSs. In [Wang et al. 2014], authors proposed specific template that has to satisfy a number of necessary objectives for development of MDSs. The objectives can be summarized as follows:

Objective 1 - elicitation of requirements should be guided by template and governed by the relevant regulations and standards.

Objective 2 – elicitation of requirements should be guided by template following several system perspectives; particular perspective should be presented by the viewpoint of one of the system's environment actor, or partner applications or systems.

Objective 3 – template decomposition should follow the essential principle of separation of concerns.

Objective 4 – template must support documenting the device safety requirements and support their ranking; it also should support articulating the safe device/environment interactions.

Objective 5 – template must support documenting the device's threat targets; it also must allow specifying the reasonable time for accessing the shared environmental resources; it should provide the needed requirements for a thorough security assessment.

Objective 6 – template must support documenting the privacy requirements; such important requirements guarantee the protection of the user's personal information.

Objective 7 – functional requirements should be formally documented but at the same time written in the form understandable for non-technical users, which is more or less standard way for requirements specification; the formalism has to support formal and automated verification of the functional requirements like: the space completeness, the dictionary, and behaviour consistency.

Objective 8 – template must be somehow general to document medical device software, i.e. to support a familiar approach.

Presented objectives have to be discussed into details with students of RE course, as a practical part of the course devoted to requirements specification and preparing requirements document. Here, it is important to actively discuss with them all necessary elements, uniqueness and particularities of healthcare and medical software especially directing their attention to privacy, security, and safety.

4. MEDICAL CASE STUDY IN RE COURSE – SPECIFIC REQUIREMENTS AND POSSIBLE ADVANTAGES

Creating successful healthcare and medical software, starting from electronic medical records and patient management systems, to medical devices, considerably differs from, so called, traditional software. Unique domain expertise is demanding, and combined with industry-specific requirements and best software practices, can significantly improve chances of success of the final software product.

Generally, efforts to bring new healthcare and medical (device) software to the market are continually underestimated. Such kind of projects have a host of industry-specific requirements. So, to increase success of such specific software, it is necessary to adopt well-planned and well-executed solutions that are highly secure, flexible, and tightly integrated with the workflow of numerous clinical users.

Therefore, we will now concentrate on critical lessons collected from years of healthcare software development presented in [Macadamian 2017], but we will also aim at changing, modifying and adjusting them, according to our experiences in requirement engineering processes regarding the realization of different scientific and professional projects, as well as in delivering the RE course for more than 10 years at the University of Novi Sad.

Presented lessons represent essential sources of useful guidelines for developing specific healthcare and medical software and understanding the unique requirements of the medical sector and by getting it right the first time. Understanding of uniqueness and proper preparation, can eventually reduce development costs and help end-users deliver better quality healthcare. Lessons can be used in introducing specific case study in the RE course.

Lesson 1: Requirements must be tailored to a practitioner's specialty – Software developers usually do not understand that requirements highly depend on specialization and/or general knowledge, of clinical users, medical staff and caregivers.

IMPORTANT: So, it is important that during the requirements elicitation, requirement engineers and software developers have to be constantly in contact with healthcare and medical specialists in the specific area, in which the software is being developed.

Advantages of the Medical Case Study for the RE course (AMCSRE): It is important to present, to the students, specific characteristics of medical software and high importance of including medical experts in all phases of the requirement engineering processes. It also is important to outline specific problems and difficulties, i.e. different state of mind of software engineers and medical experts and suggest ways to make better and more understandable communication.

Lesson 2: Privacy restrictions must be balanced with usability requirements - Personal medical data, collected from different sources and using different medical devices, has made patients privacy essential for end-users, healthcare organizations, and governments. Governments have created laws, essentially requiring the implementation of two security systems: access control (not overly restrictive - offer an override mechanism to prevent inappropriate access) and an audit trail (serves as a record of all events that can occur through the normal use of the application, and it alerts users to be careful when accessing data making them aware that their actions are being recorded).

IMPORTANT: It is important, when developing healthcare and medical software, to balance access control and audit trail mechanisms carefully. Privacy laws must be respected, but allow users to efficiently accomplishing their tasks.

AMCSRE: Special attention within the RE course must be paid to explaining to students the significance of privacy, within medical software. Also, they have to be aware of the need of inclusion of key stakeholders and legal regulations.

Lesson 3: Model hospital processes before you write a line of code – Healthcare and medical software has to map appropriately into the existing processes in medical institutions. For very complex and multifunctional healthcare and medical architectures, the separation of components of medical device software is of crucial importance. Such (new) components must not negatively affect, or significantly change, the existing processes and data processing within (existing) software.

IMPORTANT: It is critical that all processes must be understood before software is designed and well documented. Each hospital process must be fully understood before any software (or new component that will be integrated) can be designed to support it.

AMCSRE: This characteristic is important for almost all kinds of software, but it is especially important for software and particular components that are being developed for (existing) healthcare and medical software. Students have to know that all key and essential sources, for requirement elicitation, must be seriously considered and a comprehensive view of them is necessary. They must be aware of all aspects and complexities of the existing environment and architecture.

Lesson 4: Doctors have no patience for software that doesn't save them time – Healthcare and medical workers, in hospitals all over the world, have to take care of constantly growing number of patients and, on the other hand, to do their jobs as efficiently as possible. Software developers usually do not consider seriously the time restrictions healthcare and medical staff face.

IMPORTANT: When designing and implementing a new healthcare and medical software, developers have to be aware of the doctor's desire to finish clinical paperwork as quickly as possible.

AMCSRE: Healthcare and medical (device) software needs to be extremely responsive and informative. So, students must be aware of fact that, in this case, analysis and negotiation phases are essential. Healthcare practitioners and doctors must take essential part in these processes and precisely articulate their wishes and needs. Accordingly, requirement engineers and software developers must blindly follow their (doctor's) needs.

Lesson 5: Validation is far more important than verification – Generally, for health and medical software, the emphasis must be on application stability. Thus, software developers, with little healthcare experience, usually take care of software verification, but do not pay enough attention to validation. Software can always be made to do what you want it to do (verification), but it is noticeably harder to be sure you are building the right software for the job (validation) [Macadamian 2017]. In healthcare domains, software validation is critically important, requires strong domain expertise, and must be used to test all exceptional and rare medical cases.

IMPORTANT: A good software validation process will resolve multiple problems before implementation goes too far.

AMCSRE: It is necessary to emphasize to students that, in the implementation phase of healthcare and medical (device) software, validation process is extremely significant. Also, it is necessary to point out to them that obtaining adequate expertise is crucial for developing healthcare and medical (device) software.

Lesson 6: Input validation should not block users – Healthcare applications, but especially medical device software, deal with rather complex data entries. Accordingly, important software development considerations must concentrate on accurate methods for data validation and treatment, and cannot be slowed by the requirements of software.

IMPORTANT: In healthcare and medical environments it is recommended to follow effective input validation best practices. It is necessary to ensure medical care is never delayed because of software requirements.

AMCSRE: It is important to emphasize to students that, in healthcare and medical software development, input validation is necessary to take special care of: to keep the number of required fields to a minimum; to not to block users from submitting a form; to provide defaults to handle missing (not provided) data; data quality checks should account for incomplete or a 'fake' placeholder data and advise users to complete the data entry once the values are known; to continue functioning in case when incomplete data appears. It is also desirable to invite doctors and actively include them in practical classes and provoke active discussion with students, in order to try to emphasize and illustrate different styles of medical staff thinking.

Lesson 7: Database Designs must be flexible, as workplace requirements are likely to change – In health and medical institutions (predominantly hospitals), work processes and requirements are continually and constantly changing, more than in majority of traditional software or environments. It is necessary to keep these constraints in mind, while developing and deploying software, and to be able to handle changing requirements.

IMPORTANT: With the appropriate data model, it is easy to handle these challenges and allow for continuous updates.

AMCSRE: This lesson is especially important to students, in order to illustrate and emphasize importance of requirements management, especially requirements changes and tractability.

Lesson 8: Hospital processes are highly collaborative and asynchronous – In hospitals, usually, majority of individuals work autonomously, in order to provide adequate healthcare. The usual scenario is that communication is carried out between caregivers and patients. When a difficult case appears, multi-staff collaboration is needed to gain experience, start discussion and reach solution.

IMPORTANT: Ideally, good healthcare and medical software in hospitals should support an autonomous workflow process, where each member is empowered to perform her/his tasks. But, the problem usually appears as each individual may have radically different working patterns. As a general rule, doctors' schedules are more unpredictable as they are frequently getting interrupted. So, communication between healthcare and medical professionals is predominantly asynchronous.

AMCSRE: As healthcare and medical software has specific requirements, to follow radically different working patterns and specific communication needs, and thus developers must be fully aware of this. It is necessary to expose students to this uniqueness in developing healthcare and medical software, and insist to pay special attention in elicitation of requirements, as well as in trying to fully understand workflows of existing processes. Ideally, requirement engineers have access to multiple sources for requirements, especially including doctors' expertise. Preferable approach of elicitation could be to use viewpoint analysis and interviews, scenarios and observations. Furthermore, students have to be introduced to the fact that doctors are always busy and have specific way of thinking, which is very different from that of ICT developers and experts.

Lesson 9: Supporting interoperability standards is tougher than it looks - Rather complex interfaces, with external systems, have to be implemented in majority of health and medical software.

To simplify and support the processes, it is obligatory to incorporate in software solution adequate interoperability standards.

IMPORTANT: Interoperability is one of the biggest challenges in healthcare and medical ICT. So, the initial activity in realization of such kind of software is to ensure that the interoperability standards are properly planned and estimated.

AMCSRE: During presentation of medical case study to students, it is necessary to explain them that once they have obtained a very thorough, granular task breakdown structure and estimates, they have solved half the problem [Macadamian 2017]. The other important part is avoiding the high risk of reworking the software as it evolves, so that they also must precisely and cautiously consider procedures and standards starting from very early phases of requirement engineering.

Lesson 10: Not all clinical users are doctors and nurses – The most difficult groups, in developing healthcare and medical software, are doctors and nurses. So developers often neglect other groups of users who, directly or indirectly, use the software. Nevertheless, it is evident that doctors' and nurses' actions most directly affect the quality of patient care.

IMPORTANT: Software developers have to take care to satisfy the requirements of healthcare providers, but also of users who may not use the application directly and can benefit downstream from the collected data. It is also of high importance, for modern and complex multisource data collection (as smart medical devices) software. This process is known by usability experts as stakeholder identification.

AMCSRE: It is essential to explain to students that it is highly significant, in requirement engineering processes in healthcare and medical software, to include and frequently interact with a wider range of stakeholders and possible users of the software that is to be developed. Also, they have to be prepared for the fact that these initially identified stakeholders and experts can also identify other different groups of users who will also use the application. As a result, they become a valuable sources of requirement elicitation. This is especially important for medical device software, which is an essential part of modern, complex healthcare and medical software.

5. CONCLUDING REMARKS

Developing healthcare and medical software presents a set of unique software development challenges and it is usually complex and fraught with risks. Such kind of software requires precisely elicited, analyzed and specified requirements using multiple data and requirement sources and high involvement of domain experts, foresight and specialized knowledge.

This being the case, it is essential to include in a Requirement Engineering course specific hints, suggestions and issues to make students aware of the most commonly encountered issues in healthcare and medical software development, so that they can avoid typical mistakes, reduce schedule, and cost overruns, and help healthcare practitioners obtain and deliver better care for the patients, in future real-life situations. Special attention must be paid to the fact that software must be flexible and adaptable to clinical workflow processes and also that clinical end-users should be able to leverage well-designed systems that meet their needs as well.

Documenting the requirements for healthcare and medical software (especially for MDSs) is a very challenging task, due to its uniqueness and specific nature, and the stringent regulatory rules. These systems ought to exhibit the highest dependability qualities such as safety, security, and availability. A systematic approach to tackle gathering and documentation of requirements of the system, in general, is essential and it is necessary to precisely explain such elements to the students. Objectives for designing a suitable requirements template document, for healthcare and medical software, is also important and specific elements of the template have to be presented to the students. Additionally, it could be desirable to invite doctors to actively participate in classes and discuss healthcare and medical software issues with the students. It could be highly useful to students to have contact and communication with the very specific style of thinking of key medical stakeholders.

REFERENCES

Mahnaz Ahmadi. 2006. *Requirements Documentation for Manufacturing Systems: Template and Management Tool*. Ph.D. Dissertation.

Thomas A Alspaugh, Stuart R Faulk, Kathryn H Britton, R Alan Parker, and David L Parnas. 1992. *Software Requirements for the A-7E Aircraft.* Technical Report. NAVAL RESEARCH LAB WASHINGTON DC.

IEEE Computer Society. Software Engineering Standards Committee and IEEE-SA Standards Board. 1998. Ieee recommended practice for software requirements specifications. Institute of Electrical and Electronics Engineers.

Y El Gamal and W Kriedte. 1996. European Cooperation for Space Standardisation (ECSS). In *Product Assurance Symposium and Software Product Assurance Workshop*, Vol. 377. 43.

U.S. FDA. 2017. Manufacturer and User Facility Device Experience Database. https://www.fda.gov/medicaldevices/digitalhealth. (2017). Accessed: 2017-06-20.

EA Giakoumakis and G Xylomenos. 1996. Evaluation and selection criteria for software requirements specification standards. *Software Engineering Journal* 11, 5 (1996), 307–307.

U.S. HHS. 2017. Mobile Medical Devices. https://www.fda.gov/MedicalDevices/DigitalHealth/MobileMedicalApplications/default.htm#a. (2017). Accessed: 2017-06-20.

Raoul Jetley, Sithu Sudarsan, R Sampath, and Srini Ramaswamy. 2013. Medical Software–Issues and Best Practices. In *International Conference on Distributed Computing and Internet Technology*. Springer, 69–91.

Lei Lai. 2004. *Requirements documentation for engineering mechanics software: Guidelines, template and a case study*. Ph.D. Dissertation. McMaster University.

David L Lempia and Steven P Miller. 2009. Requirements engineering management handbook. *National Technical Information Service (NTIS)* 1 (2009).

Macadamian. 2017. Developing Successful Healthcare Software: 10 Critical Lessons. http://info.macadamian.com/rs/macadamian/images/Mac_10_Healthcare_Lessons.pdf. (2017). Accessed: 2017-06-20.

Suzanne Robertson and James Robertson. 2012. *Mastering the requirements process: Getting requirements right*. Addison-wesley.

Peter Rust, Derek Flood, and Fergal McCaffery. 2015. Software Process Improvement and Roadmapping–A Roadmap for Implementing IEC 62304 in Organizations Developing and Maintaining Medical Device Software. In *International Conference on Software Process Improvement and Capability Determination*. Springer, 19–30.

David A Vogel. 2011. *Medical device software verification, validation and compliance*. Artech House. 27–36 pages.

Hao Wang, Yihai Chen, Ridha Khedri, and Alan Wassyng. 2014. Envisioning a requirements specification template for medical device software. In *International Conference on Product-Focused Software Process Improvement*. Springer, 209–223.