Introduction to Medical Software

Providing a concise and accessible overview of the design, implementation, and management of medical software, this textbook will equip students with a solid understanding of critical considerations for both standalone medical software (software as a medical device/SaMD) and software that is integrated into hardware devices. It includes: practical discussion of key regulatory documents and industry standards, and how these translate into concrete considerations for medical software design; detailed coverage of the medical software lifecycle process; an accessible introduction to quality and risk management systems in the context of medical software; succinct coverage of essential topics in data science, machine learning, statistics, cybersecurity, software engineering, and healthcare, bringing readers up to speed; six cautionary real-world case studies which illustrate the dangers of improper or careless software processes. Accompanied by online resources for instructors, this is the ideal introduction for undergraduate students in biomedical engineering, electrical engineering and computer science, junior software engineers, and digital health entrepreneurs.

Xenophon Papademetris is Professor of Radiology and Biomedical Imaging, and Biomedical Engineering, at Yale University. His research focuses on biomedical image analysis and software development. He has over twenty years of experience in medical software design.

Ayesha N. Quraishi is an MD serving as the Chief of Otolaryngology at the Cambridge Health Alliance; Faculty at Harvard Medical School; former Clinical Director of the Yale Center for Biomedical Innovation and Technology; member of the MIT H@cking Medicine Initiative and Lecturer in the Practice of Management at Yale School of Management. She holds an MBA from the MIT Sloan School of Management with an emphasis on global leadership and innovation.

Gregory P. Licholai teaches at Yale School of Management and is Co-Director of the Center for Digital Health. He also co-teaches Innovating Health Care at Harvard Business School. He is Chief Medical and Innovation Officer at ICON plc, a leading pharmaceutical service, healthcare data, and contract research provider. He serves on multiple company and non-profit boards including the Digital Medicine Society (DiMe) and is an advisor on the Clinical Trials Transformation Initiative (CTTI), a public–private partnership co-founded by Duke University and the Food and Drug Administration (FDA). He received his MD from Yale School of Medicine and MBA from Harvard Business School. He writes about innovation in healthcare for *Forbes*.

Cambridge Texts in Biomedical Engineering

Cambridge Texts in Biomedical Engineering provide a forum for high-quality textbooks targeted at undergraduate and graduate courses in biomedical engineering. They cover a broad range of biomedical engineering topics from introductory texts to advanced topics, including biomechanics, physiology, biomedical instrumentation, imaging, signals and systems, cell engineering, and bioinformatics, as well as other relevant subjects, with a blending of theory and practice. While aiming primarily at biomedical engineering students, this series is also suitable for courses in broader disciplines in engineering, the life sciences and medicine.

Introduction to Medical Software

Foundations for Digital Health, Devices, and Diagnostics

Xenophon Papademetris
Yale University, Connecticut

Ayesha N. Quraishi
Harvard Medical School and Yale University, Connecticut

Gregory P. Licholai
Yale University, Connecticut

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre,
New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/highereducation/isbn/9781316514993
DOI: 10.1017/9781009091725

First published 2022

Printed in the United Kingdom by TJ Books Limited, Padstow, Cornwall, 2022

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging-in-Publication Data
Names: Papademetris, Xenophon, 1971– author. | Quraishi, Ayesha N., 1975– author. | Licholai, Gregory P., 1964– author.
Title: Introduction to medical software : foundations for digital health, devices, and diagnostics / Xenophon Papademetris, Ayesha N. Quraishi, Gregory P. Licholai.
Other titles: Cambridge texts in biomedical engineering
Description: Cambridge, United Kingdom ; New York, NY : Cambridge University Press, 2022. | Series: Cambridge texts in biomedical engineering | Includes bibliographical references and index.
Identifiers: LCCN 2022000137 | ISBN 9781316514993 (hardback)
Subjects: MESH: Software | Medical Informatics | Artificial Intelligence | BISAC: TECHNOLOGY & ENGINEERING / Engineering (General)
Classification: LCC R855.3 | NLM W 26.55.S6 | DDC 610.285–dc23/eng/202202046
LC record available at https://lccn.loc.gov/2022000137

ISBN 978-1-316-51499-3 Hardback

Additional resources for this publication at www.cambridge.org/medicalsoftware

Contents

Cambridge University Press & Assessment
978-1-316-51499-3 — Introduction to Medical Software
Xenophon Papademetris , Ayesha N. Quraishi , Gregory P. Licholai
Frontmatter
More Information

CAMBRIDGE

Cambridge University Press & Assessment
978-1-316-51499-3 — Introduction to Medical Software
Xenophon Papademetris , Ayesha N. Quraishi , Gregory P. Licholai
Frontmatter
More Information

Cambridge University Press & Assessment
978-1-316-51499-3 — Introduction to Medical Software
Xenophon Papademetris , Ayesha N. Quraishi , Gregory P. Licholai
Frontmatter
More Information

Cambridge University Press & Assessment
978-1-316-51499-3 — Introduction to Medical Software
Xenophon Papademetris , Ayesha N. Quraishi , Gregory P. Licholai
Frontmatter
More Information

Preface

The goal of this book is to provide in one brief and accessible volume a survey of the critical material involved in the design, implementation, and management of medical software for both standalone software ("software as a medical device – SaMD") and software that is part of a physical medical device. One will find more detailed treatments of many of the topics covered in this book in specialized books that focus on some of the topics we cover (e.g. software engineering, systems engineering, probability theory, machine learning). Depth was not our goal; this book is explicitly designed to provide a broad survey.

Our hope is to familiarize the reader with the span of topics he or she may need in entering this field and to provide pointers to more specialized publications as this becomes necessary. For example, most computer scientists have very limited exposure to statistical decision theory, and we think that even the cursory coverage in this book will at least enable them to understand "what they do not know" and seek help as opposed to being ignorant of this entire field and attempting to reinvent the wheel in an amateurish manner!

An emerging challenge in medical software is the increasing use of big data and artificial intelligence/machine learning (AI/ML) techniques. This places an even greater stress on proper software design and management. Given that these are "black box" methods, in which the human understanding of what actually is going on is limited, a proper software quality process will be even more critical in creating reliable software tools. We introduce this topic in Section 1.3. In that section we also provide pointers to the other sections of the book in which we discuss issues related to the use of AI/ML methods.

This is an introductory book. One can and should follow the material here with further study, using both original regulatory materials, industry standards,[1] and more advanced books.[2] Our goal can be summarized by the phrase "to convert *unknown unknowns* to *known unknowns*." Our goal is to make our reader aware of important material he or she is not as familiar with as one should be, and to pursue further study to acquire such knowledge.

This is not a programming book. Our goal is to describe the enabling activities that support programmers in producing high-quality software in the context of medical applications. We are less concerned by questions such as 'How should we code?' Our focus, rather, is on answering higher-level questions such as 'How do we decide what we need to code?' and 'How should the process be organized?' There is a wealth of material available that describes the actual coding process, and, therefore, we chose not to duplicate this type of description here.[3]

This is also not a handbook for navigating the regulatory process. Our goal in writing this book was not to create a guide that one can follow to bring a product to market.[4] We aim rather to explain how the regulatory process(es) affects the process of creating software for medical purposes. This will enable both a junior programmer entering this field and, perhaps, a high-level manager supervising a medical software project to understand why things are done the way they are. The regulatory documents and the associated industry standards capture much of what the medical device/software community has learned about this process over the past 20–30 years and constitute primary sources in this field.

Much of the content of this book is an expanded version of handouts prepared for the needs of the class Medical Software Design (BENG 406b) taught at Yale University over the period 2017–2021 by two of the authors (X. Papademetris and A. Quraishi). In our experience, university graduates are badly equipped by their coursework for careers in this general area, as many of the topics that are of critical importance to the medical device/medical software industry are almost never covered in undergraduate curricula.[5]

The Structure of the Book

This book consists of four parts. The first two provide necessary background on the topic. They describe the environment that medical software lives in, be it scientific, regulatory, clinical, managerial, or financial. The third part describes the actual process of design, implementation, and testing of medical software, and the last part presents six case studies in which (for the most part) failure to follow the principles described earlier in the book led to expensive disasters and even deaths.

Part I: Regulatory, Business, and Management Background

This goal of this part of the book is to provide the necessary regulatory and business/management background for medical software. It is subdivided into the following chapters: *Chapter 1: Introduction to Medical Software* provides a brief overview of the entire field of medical software. *Chapter 2: The FDA and Software* focuses on the regulatory aspects of medical software, with a particular emphasis on the regulations and guidance issued by the US Food and Drug Administration (FDA). While this may initially appear to be a purely US-centric approach, it is worth pointing out that the most recent guidance documents from the FDA are based on the work of an international body, the International Medical Devices Regulator Forum (IMDRF), which represents an attempt at establishing international consensus in this field. Furthermore, Section 2.4 within this chapter discusses regulations from the EU and China and other countries.

Chapter 3: Operating within a Healthcare System has two goals. The first is to briefly describe the confusing and complex world of healthcare systems. The second goal is to introduce core clinical information technology standards and to discuss data privacy issues and related legislation, such as HIPAA (US) and GDPR (EU). The chapter concludes with a discussion of the critical topic of cybersecurity.

A quality management system is a core regulatory requirement for the development of safe and high-quality medical software (and any other product). This is described in ISO 9001 [131] and its more specialized derivatives (e.g. ISO 13485 [132]) for software and medical devices. We discuss this topic in *Chapter 4: Quality Management Systems*. Skill and technical competence alone do not suffice; one must perform the work in a properly managed organization whose culture is centered on quality. *Chapter 5: Risk Management* provides a description of this important topic, which is also a core regulatory requirement. Our bases here are the appropriate international standards, in particular ISO 14971 [133].

Medical software development happens in a business environment. *Chapter 6: Taking an Idea to Market: Understanding the Product Journey* is a guided tour of this area with a particular focus on startups and entrepreneurship. We cover here issues related to intellectual property, fundraising, and marketing. Even if the reader is not particular interested in starting his or her own company, these are important issues to understand as they also drive management decisions in established companies.

We conclude this part with a more forward-looking chapter. *Chapter 7: Medical Software Applications and Growth Drivers* provides a review of the current state of the industry and, in particular, describes the excitement and emerging applications based on the use of AI/ML techniques in the area of digital health.

Part II: Scientific and Technical Background

Our aim in this part is to fill in common gaps in background knowledge of undergraduate students (and practitioners) in two important areas.

Chapter 8: Mathematical Background is a fast-paced tour of probability, statistics, and machine learning. Anyone involved in the design of medical software (especially as this touches upon diagnosis and treatment) needs to have some basic background in topics such as detection theory and to understand terms such as false positive and false negative. Most computer scientists, in our experience, have minimal exposure to these topics.

Chapter 9: Topics in Software Engineering introduces basic material from software engineering for those of our readers coming from medical, engineering, and management backgrounds as opposed to computer science. Here, we discuss topics such as software life cycles, software testing, and source code management. All of these are critical in the design and implementation of medical software. We particularly highlight Section 9.3, where we provide a description of the challenges and techniques involved in the use of AI/ML modules within medical software.

Part III: An Example Medical Software Life Cycle Process

This practical application of software life cycles is the heart of this book.[6] Here, we provide a simplified recipe for designing, implementing, testing, and distributing medical software to illustrate the concepts discussed in IEC 62304 [119]. The topic is introduced in *Chapter 10: The Overall Process*.

Next, *Chapter 11: Identifying User Needs* discusses the beginning of a project. This is where we identify what needs to be done and begin to plan how to do it. In this chapter, we also present a sample project, "The Image Guided Neuro-Navigation Project," which we use to anchor our description of the various components of the process in the subsequent chapters.

Chapter 12: The System Requirements Specification discusses the critical process of creating the system requirement document, which is the master plan or anchor for the whole process. Next, *Chapter 13: The Software Design Document* discusses (for the first time) the actual, concrete process of designing software. *Chapter 14: Software Construction, Testing, and Verification* provides a brief description of software implementation and testing.

Chapter 15: Software Validation describes the validation process, which includes a brief description of clinical trials and issues related to the evaluation of AI/ML modules. This part of the book concludes with *Chapter 16: Deployment, Maintenance, and Decommissioning*, a short chapter that discusses these last steps in the software life cycle.

Part IV: Case Studies

This last part of the book consists of six case studies that relate to either software failures or the averting (in the case of the Y2K story) of crises that would have been caused by software issues that were successfully remedied in time. We use these to illustrate and reinforce some of the lessons presented earlier in the book.

Chapter 17: Therac-25 presents the famous case of the Therac-25. These accidents, which were caused by software bugs in a radiation treatment machine, resulted in serious injuries and deaths in a number of cases in the 1980s. They also resulted in the FDA taking an active interest in medical device software. Much of the regulation in this area derives directly or indirectly from the Therac-25 disasters.

Chapter 18: Mars Climate Orbiter describes the loss of the Mars Climate Orbiter. This is a classic failure of integration testing. *Chapter 19: HealthCare.gov: The Failed Launch of a Critical Website* describes the failed launch of the federal health insurance marketplace web page. During the creation of this web page, the planned software process was not followed, resulting in confusion and overruns. *Chapter 20: The 2020 Iowa Caucus App* presents the case of the 2020 Democratic Iowa Caucus App. This is both a case of management failure, effectively trying to do things "on the cheap" in a mission-critical setting, and a case of lack of awareness of the users and their environment.

Chapter 21: The Boeing 737 MAX Disasters describes the ongoing saga involving the design of this airplane. This case, which has some strong parallels to the Therac-25, is an example of an attempt to cut corners and use software to fix/rectify fundamental hardware problems, and using shortcuts and obfuscation to get around regulatory processes. Two 737 MAX airplanes crashed, resulting in hundreds of fatalities.

Our final vignette, *Chapter 22: The Averted Y2K Crisis*, is a success story. This illustrates how proper planning and thorough work averted what might have become a serious problem caused by legacy computer software design.

Target Audience

Undergraduate students in computer science and engineering: The initial stimulus for writing this book was to address the needs of our undergraduate students in the Yale class Medical Software Design (BENG 406b).

Junior software engineers in the medical device industry: Our hope is that Part I of this book will provide this group with an understanding of the environment they find themselves in and some justification as to why things are done the way they are. Part II may also be profitable to those who lack some of the technical background. Their company will have a specific process for the actual design phase, so Part III may be a little less applicable, but may also, again, provide some explanations as to why their company's process is what it is. One of the reviewers of this book also suggested that knowledge of this material will be helpful in the interview process when applying for jobs in the medical device industry.

Senior management and/or policymakers supervising and/or managing software projects: Such individuals need to be able to assess the quality of the underlying software and to account for the risks that aspects of this software pose to their own decision-making process. For these individuals, the primary utility of a book such as this one (especially Parts I and II) may be to enable them to ask the right questions of their subordinates, who are actually directly managing the software process. The case studies (Part IV) may also provide sobering reading.

Those interested in starting a digital health company: This new entrepreneur in digital health is entering the minefield of the medical world, which comes with regulatory constraints (e.g. the FDA) and environmental constraints (hospitals) that make the process significantly harder (at least at the beginning). Our hope is that this person will be able to quickly skim this book and get a sense of what their task will involve and how to seek appropriate help (e.g. consultants) as they begin this challenging (and potentially highly rewarding) task of setting up a company (e.g. quality management systems – see Chapter 4). Chapter 6 may provide them with a useful road-map for the business aspects of this task, including coverage of topics such as intellectual property, fundraising, and setting up human studies/clinical trials. Chapter 7 reviews the current state of the industry and provides examples of recent work that may serve as an inspiration for new entrepreneurs. Finally, reading through the software process part of the book (Part III) will at the very least make it plain that they must document their software designs before they start coding if they would like anything they produce to meet regulatory standards.

NOTES

1. These regulatory documents [64, 69, 70, 82, 89, 91, 92, 125–127, 188] and industry standards [1, 2, 119, 122–124, 131–134] form a good initial list.

2. Probably at the top of this list should be the book *Medical Device Software Verification, Validation, and Compliance* [261] by D.A. Vogel. Though slightly dated, it contains much valuable information.
3. In teaching a class on medical software we have consistently emphasized that “figuring out what to code is much, much, harder than actually coding.” We, obviously, recognize that, in certain cases, the actual coding process can be highly complex and requires significant expertise and experience! There are many other classes that provide training on the coding process itself.
4. The regulatory process differs from country to country and it is critical to ascertain what the requirements are for any particular country in which approval is sought. The good news is that there are ongoing efforts to harmonize these regulations through the work of the IMDRF.
5. In Chapters 11–13 there is a specific section that applies the material in that chapter to our fictional image-guided navigation software project presented in Chapter 10. The hope is that this provides students with concrete examples of how to go about completing class assignments. We also provide sample class assignments in these chapters, and also in Chapter 15. One of the authors was involved in the design and implementation of a research interface for the BrainLAB system [198], but he had no (and still does not have a) financial connection to the company. This was a purely scientific collaboration as part of an NIH-funded project.
6. When taking a class on medical software, many students expect the class to start at this point. By contrast, many of the regulations highlight the importance of quality management systems and risk management in addition to the software life cycle process.

Acknowledgments

We would first like to thank the students, the teaching fellows, and the mentors who participated in BENG 406b over the past few years. They were the “beta-testers” of the book and we thank them for their patience through the years and their helpful suggestions for improving the manuscript.

We would like to thank our anonymous reviewers for their kind and helpful feedback. In addition, we would like to thank the friends and colleagues who volunteered to read portions of this book and offer helpful comments. In alphabetical order: Tina Kapur, Sean Khozin, Phan Luu, John Onofrey, Nicholas Petrick, Steve Pieper, Saradwata Sarkar, Lawrence Staib, Don Tucker, and Rajesh Venkataraman. An Qu, postgraduate associate at Yale, double-checked the unofficial English translation of one of the Chinese regulatory documents [189]. Robert Bell took part in the many early discussions about this book with Drs. Papademetris and Quraishi. We would also like to thank the anonymous peer reviewers of the book for their many constructive comments.

We would like to thank our colleagues at Yale biomedical engineering, in particular Mark Saltzman, who had the idea for creating this class and provided the institutional support and mentorship as it was being set up.

Ellie Gabriel, a Yale biomedical engineering undergraduate, wrote the initial drafts of the six case studies in Part IV of this book as part of a summer internship (2020) she did under the supervision of Dr. Papademetris. She also created the illustration for Figure 10.4. She, together with Maria Papademetris, did much of the initial proofreading for the text.

We would also like to thank our editors at Cambridge University Press, Elizabeth Horne and Jane Adams, and our Content Manager, Rachel Norridge, for their help in shepherding us through the process in the middle of a global pandemic. Without their help, this would have been a much inferior manuscript.