Preface

"Software" is regularly blamed for a large variety of failures—both grand and petty—in our modern world. It is, therefore, ironic that many software developers pay so little attention to placing their profession on a sound engineering basis by using codified approaches that are known to increase the probabilities of success. Fortunately, there are some signs of progress. Two of them provide the basis for this book: the software engineering standards of the IEEE Computer Society and the Society's *Guide to the Software Engineering Body of Knowledge*.

Software Engineering Standards

The first basis for this book is the work of largely anonymous individuals who cooperate in the development of software engineering standards, capturing processes and practices that are effective in improving software development, maintenance, and operation. Considered individually, each of these standards is a technical gem. Taken collectively, though, the corpus of hundreds of standards from dozens of organizations, using a variety of jargon, and having various scopes, can leave the hopeful user with questions rather than answers. Two organizations have stepped up to this challenge. The Software and Systems Engineering Standards Committee (S2ESC) of the IEEE Computer Society (IEEE-CS) has committed to a long-term program to place its 40 or so standards on a consistent basis. As a part of this goal, it cooperates with the international standards committee for software and systems engineering (ISO/IEC JTC 1/SC 7) to "harmonize" their respective collections so that users may choose standards from either collection without fear of contradiction.

The integration of the S2ESC collection and its harmonization with that of SC 7 is a long-term project that may never be finished. There are always new communities joining the scope of the harmonization effort. For example,

for roughly a decade, both S2ESC and SC 7 have worked to incorporate and harmonize systems engineering practices into the corpus of software engineering standards. As progress is made in this area, the need to harmonize with standards for enterprise business processes has become apparent and represents a new set of challenges.

Despite the never-ending challenge presented by harmonization, substantial progress has been made in integrating the S2ESC collection. In some sense, this book is a progress report of that integration effort.

Guide to the Software Engineering Body of Knowledge

The second basis for this book is the new 2004 version of the IEEE Computer Society's *Guide to the Software Engineering Body of Knowledge*. A few years ago, one of my colleagues went to a used-book store and returned with a circa 1910 "handbook of electrical engineering." In a single volume of several hundred pages, it summarized the information that an electrical engineer of that era should know and provided reference to additional knowledge. The modern-day *SWEBOK Guide* is the software equivalent of that volume. It provides a basis for the scope and content of software as an engineering discipline. Although software engineers need to know things from other disciplines, the *SWEBOK Guide* describes that portion of their necessary knowledge that characterizes software engineering per se.

Codified Practice

Together, these two sources provide the basis for what I like to call "Codified Software Engineering," a set of knowledge and practices that are applicable most of the time on most projects and which should be applied most of the time. To be sure, there is always a need for continued research, continued experimentation, and continued empirical validation to produce more effective approaches. However, managers of expensive projects who ignore the accepted techniques in favor of alternatives should be aware—and those who are paying the bill should be aware—that experimental engineering carries along its own set of risks. Perhaps the most important contribution of codifying software engineering is to make it clear which aspects of the practice are widely accepted as effective so that innovative alternatives receive the attention and risk management that they deserve.

The Role of This Book

This book provides a single overview of codified software engineering by placing the IEEE software and systems engineering standards—as well as a few standards from other sources—into two important contexts: the context of knowledge provided by the *SWEBOK Guide*, and the context of organizationally adopted processes provided by two important umbrella standards.

The book is intended to appeal to several audiences:

- The most important audience is software engineering practitioners. Because the standards are placed in both the context of topical knowledge

areas and the context of commonly recognized processes, it should be easy for practitioners to quickly locate the standards pertinent to questions arising in real projects.

- Closely related to the first audience are professionals interested in their own development as software engineers. The topical organization assists readers in finding "gaps" in their own knowledge and addressing those gaps by consulting the normative literature.
- Another important audience is students of software engineering and their instructors. For this audience, the book is a compendium of "industrial-strength" approaches set in the context of the knowledge that a student should seek to attain.

In considering this book for the latter usage, instructors should note that the *SWEBOK Guide* was one of the principal sources for the joint IEEE/ACM joint curriculum on software engineering.* So the *SWEBOK Guide* itself, as well as this book, should be broadly compatible with courses based on this curriculum.

The book is organized into three parts:

- As is usually the case, the first part of the book is introductory material. Chapter 1 provides an overview of the emerging engineering discipline for software; Chapter 2 provides an overview of software engineering standards-makers, with particular focus on two organizations of particular relevance to this book. The most important chapter—Chapter 3—describes the principles behind the organization of the S2ESC collection. Chapter 4 describes some of the approaches to organizing an integrated collection of software engineering standards.
- The second part of the book organizes software and systems engineering standards by the knowledge areas of the *SWEBOK Guide*.
- The third part of the book organizes software and systems engineering standards by the processes of ISO/IEC 12207, *Software Life Cycle Processes*, and ISO/IEC 15288, *System Life Cycle Processes*—both prepared with the cooperation of S2ESC and both now adopted by IEEE as standards.

Organization by Knowledge Area

One of the many contributions of the *SWEBOK Guide* was the designation of ten *knowledge areas* characterizing the content of software engineering as well as eight *related disciplines* that form the boundary of software engineering. That classification is directly reflected in this part of the book. Chapter 5 gives an overview of the *SWEBOK Guide*. The ten subsequent chapters—Chapters 6 through 15—are each devoted to a single knowledge area. Each one of these chapters begins with a statement of scope and a summary, both excerpted from the *SWEBOK Guide*. The largest part of each chapter is a description of the knowledge area, organized similarly to the treatment in the *SWEBOK Guide*.

* http://sites.computer.org/ccse/

Whenever a software or systems engineering standard would make a contribution to the knowledge area, I note its relevance by inserting a reference that looks like this example:

IEEE Std 1062, 1998 Edition (R2002)
IEEE Recommended Practice for Software Acquisition
This document recommends a set of useful practices that can be selected and applied during software acquisition. It is primarily suited toacquisitions that include development or modification rather than off-the-shelf purchase.
Allocated to: Software Engineering Management Knowledge Area

The icon on the left—and its label—classify the subject of the standard into one of seven categories: *document, measure, plan, process, terminology, tool*, and, in one case, the *world wide web* application area.

The top few lines on the right give the formal designation and title of the standard. (It should be noted that the designations of IEEE standards are trademarked, e.g., IEEE Std 1062™.) The next few lines on the right give a very brief summary of the content of the standard. The final line records a forced allocation of each standard into a single knowledge area. That's your cue that a more detailed description of the standard can be found in the chapter of this book devoted to that knowledge area.

At the end of each chapter is a section providing individual descriptions of the standards that have been allocated to that knowledge area. The descriptions provide information regarding history and content of the standards, and, in some cases, plans for revision of the standards.

Chapter 16 describes the eight related disciplines listed by the *SWEBOK Guide*. This is important because twelve of the standards described by this book are allocated to the related disciplines rather than to the software engineering knowledge areas. At the end of Chapter 16, you will find descriptions of those standards.

Two standards don't fall cleanly into any of the knowledge areas or related disciplines, one because it is too general (a vocabulary standard) and the second because it is too specific (an application area guide). They are both described in Chapter 17.

Organization by Process

Recent years have seen great emphasis on software process in the conviction that improved processes will lead to improved products. Organizations interested in process definition and process improvement inevitably ask how standards can contribute to this endeavor. So Part 3 of this book organizes the software and systems engineering standards by process. Chapter 18 provides a history and some important concepts. Chapter 19 organizes the standards by the software life cycle processes and Chapter 20 organizes them by system life cycle processes. The icons mentioned above are used to refer to standards of interest. Their descriptions can be found in the chapters devoted to knowledge areas.

Goals of This Book

Although this book pays great attention to integrated collections of standards, it is not my intention to encourage readers to adopt collections in their entirety. Such a step would cause an individual or an organization to attempt to digest a few thousand pages of highly technical material in a single meal. Instead, the purpose of providing the knowledge and process context is to assist readers in selecting a few standards that suit their needs today, confident that today's selection of standards will not be contradicted by others yet to be adopted.

This book has another important goal. The IEEE Computer Society, in partnership with John Wiley & Sons, has an overall project to provide a series of book devoted to software engineering standards in different areas, e.g., maintenance, verification and validation, testing, quality assurance, etc. We can expect to see those books in future years. In fact, one has already been published—Carma McClure's book on *Software Reuse: A Standards-Based Guide*. In some senses, this book is an entrée to that series—by providing an overview of the entire discipline.

A Few Disclaimers

It's my duty to warn the reader of a few things that may not be completely accurate or precise.* Because I wanted this book to be published at roughly the same time as the 2004 version of the *SWEBOK Guide*, I had to rely on prepublication drafts of the guide. The final published version of the guide may be slightly different from material excerpted in this book.

I think that it makes for turgid prose to always refer to standards in their full, glorious designation, e.g., "IEEE Std 1062™, 1998 Edition (R2002), *IEEE Recommended Practice for Software Acquisition*." The book might double in size if I followed that practice consistently. Therefore, despite the advice of well-meaning editors, I have elided many references. Almost without exception, I omit the trademark designation (™). I typically omit the information regarding the provenance of the standard, e.g., 1998 Edition (R2002). I sometimes shorten the title to its essentials and I often omit the abbreviation "Std." Often, I simply jump to the heart of the matter and refer to the standard by its bare number, 1062. I doubt that readers will be confused by this practice; it certainly improves the readability of the prose.

ISO/IEC 12207 is a special case. When IEEE adopted the standard, it added two parts (volumes), added some appendices to the base volume, and even corrected a few errors. Nevertheless, IEEE/EIA 12207.0 (part 0!) can be regarded as equivalent to ISO/IEC 12207. When I write simply "12207." I mean that either the ISO/IEC or the IEEE/EIA standard apply equally well.

Despite the fact that the software engineering standards development committee of the IEEE Computer Society has evolved in name and scope from SESS to SESC and then S2ESC, I typically use the abbreviation S2ESC unless there is a historical reason to make a distinction.

* It's the duty of any engineer—and an exercise for the reader—to learn the difference between the words *accurate* and *precise*.

During the period that I wrote this book, new standards were being developed and existing standards were being revised. I have attempted to "freeze" each standard in time and describe it accordingly so that the treatment is consistent. My strategies for doing this have varied on a case-by-case basis. For example, I have anticipated the 2004 revision of IEEE Std 1012 but have ignored the late-2004 adoption of ISO/IEC 15288 by the IEEE. Nevertheless, it is virtually certain that some inconsistencies have crept into the text.

Finally, I should mention the relationship of this book with my previous one: *Software Engineering Standards: A User's Road Map*, IEEE Computer Society Press, 1997. Both books were intended to fill the same role—to provide an overview of the S2ESC collection of standards at their respective points in time. The books turned out to be very different, though. Lacking a body of knowledge, the previous book paid a lot of attention to the context of software engineering in the hope that the context would provide insight regarding the core. It also paid relatively more attention to the international standards of SC 7. This book is organized on a completely different basis—that of the *SWEBOK Guide*—and is more strongly focused on S2ESC as an integrator of standards, even standards developed by other organizations. Nevertheless, the reader will find some similar material, mostly in the introductory chapters and the chapter on the processes of 12207. I insisted that the words "Road Map" appear in the title of this book as a small warning to buyers that they might already own material similar to some of what they are buying in this book.

My Thanks

Having now written **two** books, I can report that they are the **two** hardest things I have done in my life. (I was disappointed that the second one was not any easier than the first one!) So, I owe thanks to many people—if only for my bad temper during the past several months.

First, I must thank the responsible people at the IEEE Computer Society Press, Angela Burgess and Deb Plummer, for their gentle encouragement during my "slow periods," and the S2ESC book series editor, Roger Fujii. (I also appreciate, but cannot individually acknowledge, the contributions of the anonymous reviewers of this manuscript.)

The development of the *SWEBOK Guide* was a landmark accomplishment in the maturing of software engineering. Alain Abran served with me as an executive editor. Pierre Bourque and Robert Dupuis served as editors. Two dozen or so other individuals—mentioned in the knowledge area chapters—edited particular portions of the *Guide*. Robert Dupuis was primarily responsible for editing the 2004 edition. (Dale Strok and Bob Werner, staffers of the IEEE Computer Society Press, provided editorial services—as volunteers!) Approximately 500 individuals provided review and comment on the guide as it progressed through various stages toward completion. They are named on the SWEBOK web site.*

The IEEE Computer Society has been steadfast in its support of the professionalization of software engineering. A few of the people who have been

* http://www.swebok.org

generous to me with their time and attention include Don Bagert, Carl Chang, Kathy Land, Steve McConnell, Fernando Naveda, and Steve Seidman.

During my ten years of service on the Executive Committee of S2ESC, I learned much from my colleagues; Paul Croll, David Schultz, Scott Duncan, Mark Henley, Joe Jarzombek, Dennis Lawrence, Claire Lohr, and John Walz deserve special mention for their contributions to my education.

I have participated in the work of ISO/IEC JTC 1/SC 7 for nearly as many years—most recently as the IEEE Computer Society's liaison representative; I appreciate the support of SC 7's chair, François Coallier, and two successive chairs of its Working Group 7, Stan Magee and Doug Thiele, as well as WG7's current secretary, Bob Johnson. I have learned much from the various national body representatives and other participants in SC 7; unfortunately, they are far too numerous to mention.

An important aspect of software engineering is how organizations define, adopt, assess, and improve their processes. In this area, I have learned much from Terry Rout, David Kitson, Jeanie Kitson, and Perry Deweese.

As the work of SC 7 and S2ESC has extended into systems engineering, there were new things to learn. Ken Crowder, Jerry Lake, Stuart Arnold, Alain Faisandier, and Terry Doran have all been patient with me.

The US Technical Advisory Group to SC 7 has also been a source of knowledge, notably Mike Gayle, its current chair, and Garry Roedler, the US lead to WG7. Finally, Bob Pritchard, the IEEE's administrator of the US TAG to SC 7, insists that I thoroughly understand all the nuances of JTC 1 standardization. (I'm not quite there yet.)

In these standards activities, I met Jack McGarry, Cheryl Jones, and David Card, who taught me most of what I know about software measurement, and Bob Charette, who taught me risk management.

Various staff members at the IEEE Standards Association (IEEE-SA) have been very helpful in providing access to necessary materials, including Executive Director Judy Gorman, Terry DeCourcelle, Mary Lynne Nielsen, Claudio Stanziola, David Ringle, Yvette Ho Sang, and Angela Ortiz. On the JTC 1 side, Witold Suryn (secretary of JTC 1/SC 7) and Keith Brannon (ISO Central Secretariat) have been helpful.

Many of these strands lead back to a single source. Leonard Tripp (now an IEEE Fellow) has served as President of the Computer Society, Chairman of S2ESC, Chairman of the US TAG to SC 7, member of the IEEE-SA Standards Board, Chairman of the CS Professional Practices Committee, founder of the Certified Software Development Professional certification, and "champion" of the SWEBOK project. As far as I know, he was the first to articulate the concept that the integration of software engineering standards into a coherent whole was more important than the individual standards.

I gratefully acknowledge the interest of my employer for the past ten years, The MITRE Corporation, in supporting my participation in many of the standards activities described in this book.* John Slaybaugh, Linda Rosa, Steve

* In accordance with the procedures of The MITRE Corporation, I include the following disclaimer: The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions, or viewpoints expressed by the author.

Huffman, and Lyn van Hoozer have all been notably supportive. MITRE, along with several other organizations, also provided financial support to the Software Engineering Body of Knowledge project and appointed a representative, Chuck Howell, to its Industrial Advisory Board.

Finally and most importantly, I must note that writing a book requires the sacrifice of time that might be spent with the family. I appreciate the support of my wife, Barbara, and my son, David.

Potomac, Maryland JIM MOORE