

# Preface

When computer software succeeds—when it meets the needs of the people who use it, when it performs flawlessly over a long period of time, when it is easy to modify and even easier to use—it can and does change things for the better. But when software fails—when its users are dissatisfied, when it is error prone, when it is difficult to change and even harder to use—bad things can and do happen. We all want to build software that makes things better, avoiding the bad things that lurk in the shadow of failed efforts. To succeed, we need discipline when software is designed and built. We need an engineering approach.

In the 25 years since the first edition of this book was written, software engineering has evolved from an obscure idea practiced by a relatively small number of zealots to a legitimate engineering discipline. Today, it is recognized as a subject worthy of serious research, conscientious study, and tumultuous debate. Throughout the industry, software engineer has replaced programmer as the job title of preference. Software process models, software engineering methods, and software tools have been adopted successfully across a broad spectrum of industry applications.

Although managers and practitioners alike recognize the need for a more disciplined approach to software, they continue to debate the manner in which discipline is to be applied. Many individuals and companies still develop software haphazardly, even as they build systems to service today's most advanced technologies. Many professionals and students are unaware of modern methods. And as a result, the quality of the software that we produce suffers, and bad things happen. In addition, debate and controversy about the true nature of the software engineering approach continue. The status of software engineering is a study in contrasts. Attitudes have changed, progress has been made, but much remains to be done before the discipline reaches full maturity.

The sixth edition of *Software Engineering: A Practitioner's Approach* is intended to serve as a guide to a maturing engineering discipline. The sixth edition, like the five editions that preceded it, is intended for both students and practitioners,

retaining its appeal as a guide to the industry professional and a comprehensive introduction to the student at the upper level undergraduate or first year graduate level.

The sixth edition is considerably more than a simple update. The book has been revised extensively and restructured to emphasize new and important software engineering processes and practices. In addition, a new “support system,” illustrated below, provides a comprehensive set of student, instructor, and professional resources to complement the content of the book. These resources are presented as part of a Web site ([www.mhhe.com/pressman](http://www.mhhe.com/pressman)) and CD-ROM specifically designed for *Software Engineering: A Practitioner’s Approach*.



**The Sixth Edition.** The 32 chapters of the sixth edition have been organized into five parts. This has been done to compartmentalize topics and assist instructors who may not have the time to complete the entire book in one term. Part 1, *The Process*, presents a variety of different views of software process, considering all important process models and addressing the debate between prescriptive and agile process philosophies. Part 2, *Software Engineering Practice*, presents analysis, design, and testing methods with an emphasis on UML modeling. Part 3, *Web Engineering*, presents a complete engineering approach for the analysis, design and testing of Web applications. Part 4, *Managing Software Projects*, presents topics that are relevant to those who plan, manage, and control a software development project. Part 5, *Advanced Software Engineering Topics*, presents dedicated chapters that address formal methods, cleanroom software engineering, component-based software engineering, reengineering, and future trends.

In addition to many new and significantly revised chapters, the sixth edition introduces over 120 sidebars that (1) allow the reader follow a (fictional) project team as it plans and engineers a computer-based system; (2) provide complementary discussion of selected topics; (3) outline “task sets” that describe work flow for selected software engineering activities; (4) suggest automated tools relevant to chapter topics.

The five-part organization of the sixth edition enables an instructor to “cluster” topics based on available time and student need. An entire one-term course can be built around one or more of the five parts. For example, a “methods course” might emphasize only Parts 1 and 2; a Web development course might emphasize Parts 1 and 3; a “management course” would stress Parts 1 and 4. By organizing the sixth edition in this way, I have attempted to provide an instructor with a number of teaching options. In every case the content of the sixth edition is complemented by the following elements of the *SEPA, 6/e Support System*.

**Student Resources.** A wide variety of student resources includes an extensive on-line learning center encompassing study guides, practice quizzes and a variety of Web-based resources (e.g., checklists, an evolving collection of “tiny tools,” a complete case study, work product templates) and over 800 categorized Web references that allow a student to explore software engineering in greater detail.

**Instructor Resources.** A broad array of instructor resources has been developed to supplement the sixth edition. These include a comprehensive on-line Instructor’s

Guide (also downloadable) and supplementary teaching materials including a complete set of Powerpoint slides that may be used for lectures, a test bank, and sample exams. In addition, a “reference library”, containing pointers to over 500 software engineering papers (organized by topic and downloadable in pdf format) can be used in advanced software engineering courses where in-depth discussion of specific topics is required.

The *Instructor's Guide for Software Engineering: A Practitioner's Approach* presents suggestions for conducting various types of software engineering courses, recommendations for a variety of software projects to be conducted in conjunction with a course, solutions to selected problems, and a number of useful teaching aids.

**Professional Resources.** A collection of resources available to industry practitioners (as well as students and faculty) includes outlines and samples of software engineering documents and other work products, a useful set of software engineering checklists, a catalog of software engineering (CASE) tools, a comprehensive collection of Web-based resources, and an “adaptable process model” that provides a detailed task breakdown of the software engineering process.

When coupled with its on-line support system, the sixth edition of *Software Engineering: A Practitioner's Approach*, provides flexibility and depth of content that cannot be achieved by a textbook alone.

**Acknowledgements.** My work on the six editions of *Software Engineering: A Practitioner's Approach* has been the longest continuing technical project of my life. Even when the writing stops, information extracted from the technical literature continues to be assimilated and organized. For this reason, my thanks to the many authors of books, papers, and articles (in both hardcopy and electronic media) who have provided me with additional insight, ideas, and commentary over the past 25 years.

Special thanks go to Tim Lethbridge of the University of Ottawa who performed an extremely detailed review of the sixth edition, assisted me in the development of UML and OCL examples, and developed the comprehensive case study that accompanies this book. His assistance and comments were invaluable. Special thanks also go to Bruce Maxim of the University of Michigan–Dearborn, who assisted me in developing the Web site that accompanies this book. Bruce is

responsible for much of its pedagogical content. Finally, I wish to thank the reviewers of the sixth edition: <<insert list of reviewers and affiliations here>> Their in-depth comments and thoughtful criticism have been invaluable.

The content of the sixth edition of *Software Engineering: A Practitioner's Approach* has been shaped by industry professionals, university professors, and students who have used earlier editions of the book and have taken the time to communicate their suggestions, criticisms, and ideas. My thanks to each of you. In addition, my personal thanks go to our many industry clients worldwide, who certainly have taught me as much or more than I could ever teach them.

As the editions of this book have evolved, my sons, Mathew and Michael, have grown from boys to men. Their maturity, character, and success in the real world have been an inspiration to me. Nothing has filled me with more pride. And finally, to Barbara, my love and thanks for encouraging still another edition of "the book."

Roger S. Pressman