HZ BOOKS
华章教育

Mc Graw Hill Education

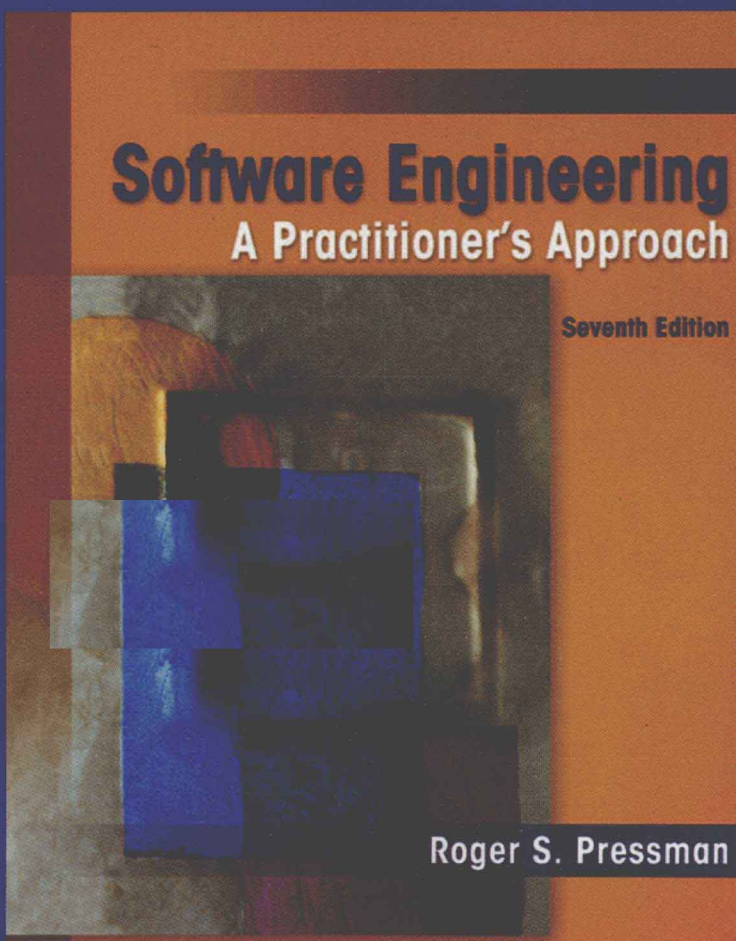
经 典 原 版 书 库

软件工程

实践者的研究方法

（美）Roger S. Pressman 著

（英文精编版·第7版）



机械工业出版社
China Machine Press

软件工程

实践者的研究方法

Software Engineering

A Practitioner's Approach

经典原版书库

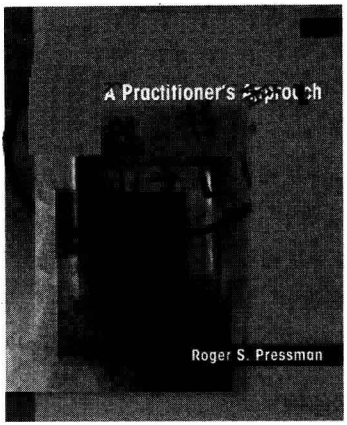
软件工程

实践者的研究方法

（英文精编版·第7版）

Software Engineering

A Practitioner's Approach (Seventh Edition · English Abridgement)



（美）Roger S. Pressman 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

软件工程：实践者的研究方法（英文精编版·第7版）/（美）普雷斯曼（Pressman, R. S.）著. —北京：机械工业出版社，2011.10

（经典原版书库）

书名原文：Software Engineering: A Practitioner's Approach, Seventh Edition

ISBN 978-7-111-35965-4

I. 软… II. 普… III. 软件工程－英文 IV. TP311.5

中国版本图书馆CIP数据核字（2011）第194817号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：姚　蕾

北京京师印务有限公司印刷

2011年10月第1版第1次印刷

150mm×214mm·18印张

标准书号：ISBN 978-7-111-35965-4

定价：49.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：（010）88379604
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

华章教育
华章科技图书出版中心

Adapter's Foreword

Purpose

The original of this book is an excellent work of Dr. Pressman. The 7th edition emphasizes on the new software engineering process and techniques, and is definitive on basically all the subjects that are listed by the Software Engineering Body of Knowledge (SWEBOK, http://www.swebok.org/). The extensive examples, references and online exercises are also quite helpful to the students. There are numbers of universities and colleges in China that are taking this book as their textbooks and keep tracking through its different versions.

As the idea of bilingual teaching is promoted in the higher education institutes in China, more and more schools are getting interested in this book. However, over nine hundred pages of the original book make it too difficult for a Chinese student with an average English reading ability to comprehend the fundamentals of software engineering in a semester. In order to introduce this book to more Chinese college students, a compression of the original book is made to better fit the book into the syllabus of an undergraduate course with only the core contents, and to reduce the students' reading load.

What's Compressed

First of all, since most of the undergraduate Software Engineering course is mainly a "methods course" plus some fundamentals on management, only the parts 1, 2, 3 and 4 of the original book are kept. Parts 5 (Advanced Topics) are for graduate courses and hence will not appear in the compressed version, neither Chapter 12 (Pattern-Based Design), Chapter 21 (Formal Modeling and Verification) and Chapter 29 (Maintenance and Reengineering).

Secondly, all the chapters, sections and sub-sections related to Web

Engineering and WebApp are for graduate courses and hence will be removed. They are: Chapter 13 (WebApp Design), Chapter 20 (Testing Web Applications), Section1.2 (The Unique Nature of WebApps), Section 7.5 (Requirements Modeling for WebApps), Section 10.4 (Component-Level Design for WebApps), Section 11.5 (WebApp Interface Design), Section 17.5 (Test Strategies for Object-Oriented Software), Section 22.4 (Configuration Management for WebApps), Sub-section 25.2.6 (WebApp Project Metrics) and Sub-section 27.5.4 (Scheduling for WebApp Projects).

Thirdly, according to the feedbacks from the students in the past eight years, we realize that as undergraduate students who have no experience on doing any large-scaled project, it is quite difficult for them to fully understand the point of doing measurements and collecting statistical data. It would be enough to simply introduce some basic concepts on metrics and estimation. Hence the following contents are removed: Sub-section 23.2.2 and Sections 23.3-23.7 (Product Metrics for analysis, design, source code, testing and maintenance), Sections 25.4-25.6 (Metrics for Process and Projects - integrating metrics within the software process, metrics for small organizations, and establishing a software metrics program), and Sections 26.9-26.10 (Estimation for Software Project - specialized estimation techniques and the make/buy decision).

Other minor compressions are made at the following chapters:

1. In Chapter 1, Section 1.7 (How It All Starts) is removed to reduce the students' reading load.

2. In Chapter 2, Section 2.2 (Process Assessment and Improvement) and Section 2.6 (Personal and Team Process Models) are considered as the advanced topics and hence are not kept in the compressed version.

3. The content of Chapter 4 (Principles that Guide Practice) will eventually be introduced in the following chapters and is removed to save the students some reading time.

4. In Chapter 18, Sections (and sub-sections) 18.4.4 (Graph Matrices), 18.6.1 (Graph-Based Testing Methods), 18.6.4 (Orthogonal Array Testing),

18.7 (Model-Based Testing), 18.8 (Testing for Specialized Environments, Architectures, and Applications) and 18.9 (Patterns for Software Testing) are considered as the advanced topics and hence will not appear in the compressed version.

Acknowledgments

We feel grateful to Roger Pressman, the author of the original book, and McGraw Hill, the original publisher, for allowing us to compress the original 928-page book into about five hundred pages. It is their understanding and generosity that make it possible for more Chinese students to enjoy this distinguished book. Since the compression is made by simply removing some chapters or sections, we hope to keep as far as possible the elegancy of Dr. Pressman's writing style.

Thanks to everyone at China Machine Press who have put in great effort to make this kind of cooperation possible.

The compression is made according to my own experience in teaching and practicing in software engineering. It is certainly wild open for discussions and suggestions on further improvements. Your comments are important to us, and would be very much appreciated.

Yue Chen
Zhejiang University
(chenyue@cs.zju.edu.cn)

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.

It has been almost three decades since the first edition of this book was written. During that time, 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 segments.

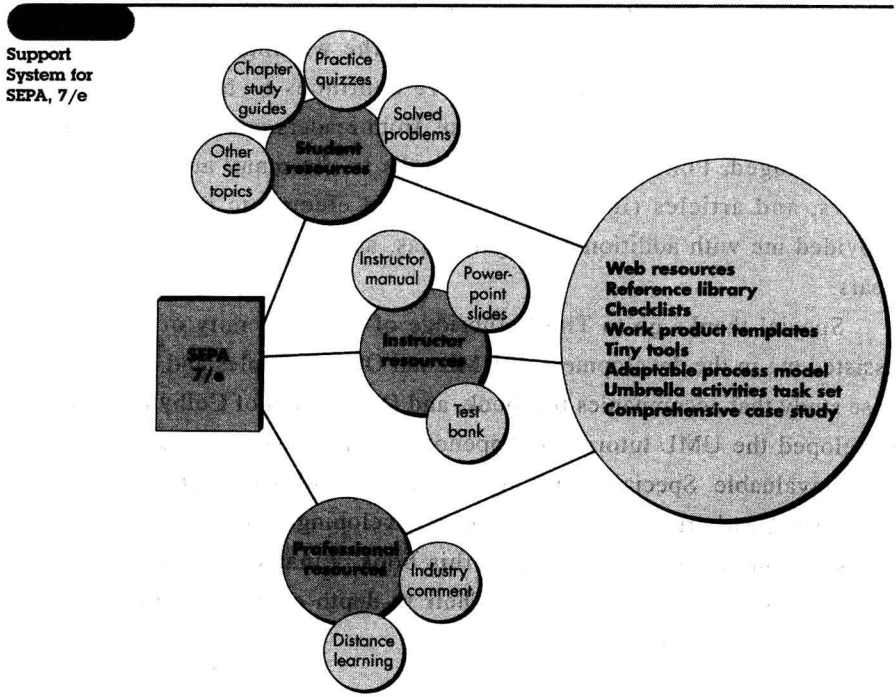
Although managers and practitioners alike recognize the need for a more disciplined approach to software, they continue to debate the manner in which discipline is to be applied. Many individuals and companies still develop software haphazardly, even as they build systems to service today's most advanced 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 seventh edition of *Software Engineering: A Practitioner's Approach* is

intended to serve as a guide to a maturing engineering discipline. Like the six editions that preceded it, the seventh edition 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 seventh edition is considerably more than a simple update. The book has been revised and restructured to improve pedagogical flow and emphasize new and important software engineering processes and practices. In addition, a revised and updated "support system," illustrated in the figure, 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 website (www.mhhe.com/ pressman) specifically designed for *Software Engineering: A Practitioner's Approach.*

The Seventh Edition. The 22 chapters of this edition have been reorganized into four parts.



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, *Modeling*, presents analysis and design methods with an emphasis on object-oriented techniques and UML modeling. Part 3, *Quality Management*, presents the concepts, procedures, techniques, and methods that enable a software team to assess software quality, review software engineering work products, conduct SQA procedures, and apply an effective testing strategy and tactics. Part 4, *Managing Software Projects*, presents topics that are relevant to those who plan, manage, and control a software development project. Continuing in the tradition of past editions, a series of sidebars is used throughout the book to present the trials and tribulations of a (fictional) software team and to provide supplementary materials about methods and tools that are relevant to chapter topics.

Acknowledgments. My work on the seven 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, and criticism and suggestions from readers worldwide is evaluated and cataloged. 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 nearly 30 years.

Special thanks go to Tim Lethbridge of the University of Ottawa, who assisted me in the development of UML and OCL examples and developed the case study that accompanies this book, and Dale Skrien of Colby College, who developed the UML tutorial in Appendix 1. Their assistance and comments were invaluable. Special thanks also go to Bruce Maxim of the University of Michigan-Dearborn, who assisted me in developing much of the pedagogical website content that accompanies this book. Finally, I wish to thank the reviewers of the seventh edition: Their in-depth comments and thoughtful criticism have been invaluable.

Osman Balci,
Virginia Tech University
Max Fomitchev,
Penn State University
Jerry (Zeyu) Gao,
San Jose State University
Guillermo Garcia,
Universidad Alfonso X Madrid
Pablo Gervas,
Universidad Complutense de Madrid
SK Jain,
National Institute of Technology Hamirpur
Saeed Monemi,
Cal Poly Pomona
Ahmed Salem,
California State University
Vasudeva Varma,
IIIT Hyderabad

The content of the seventh 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 tolerating the many, many hours in the office and encouraging still another edition of "the book."

Roger S. Pressman

About the author

Roger S. Pressman is an internationally recognized authority in software process improvement and software engineering technologies. For almost four decades, he has worked as a software engineer, a manager, a professor, an author, and a consultant, focusing on software engineering issues.

As an industry practitioner and manager, Dr. Pressman worked on the development of CAD/CAM systems for advanced engineering and manufacturing applications. He has also held positions with responsibility for scientific and systems programming.

After receiving a Ph.D. in engineering from the University of Connecticut, Dr. Pressman moved to academia where he became Bullard Associate Professor of Computer Engineering at the University of Bridgeport and director of the university's Computer-Aided Design and Manufacturing Center.

Dr. Pressman is currently president of R.S. Pressman & Associates, Inc., a consulting firm specializing in software engineering methods and training. He serves as principal consultant and has designed and developed *Essential Software Engineering*, a complete video curriculum in software engineering, and Process Advisor, a self-directed system for software process improvement. Both products are used by thousands of companies worldwide. More recently, he has worked in collaboration with EdistaLearning in India to develop comprehensive Internet-based training in software engineering.

Dr. Pressman has written many technical papers, is a regular contributor to industry periodicals, and is author of seven technical books. In addition to *Software Engineering: A Practitioner's Approach*, he has co-authored *Web Engineering* (McGraw-Hill), one of the first books to apply a tailored set of software engineering principles and practices to the development of Web-based systems and applications. He has also written the award-winning *A Manager's Guide to Software Engineering* (McGraw-Hill); *Making Software*

Engineering Happen (Prentice Hall), the first book to address the critical management problems associated with software process improvement; and *Software Shock* (Dorset House), a treatment that focuses on software and its impact on business and society. Dr. Pressman has been on the editorial boards of a number of industry journals, and for many years, was editor of the "Manager" column in *IEEE Software*.

Dr. Pressman is a well-known speaker, keynoting a number of major industry conferences. He is a member of the IEEE, and Tau Beta Pi, Phi Kappa Phi, Eta Kappa Nu, and Pi Tau Sigma.

On the personal side, Dr. Pressman lives in South Florida with his wife, Barbara. An athlete for most of his life, he remains a serious tennis player (NTRP 4.5) and a single-digit handicap golfer. In his spare time, he has written two novels, The *Aymara Bridge and The Puppeteer*, and plans to begin work on another.

Table of Contents

出版者的话
Adapter's Foreword
Preface
About the Author

CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 1

1.1 The Nature of Software 3
1.1.1 Defining Software 4
1.1.2 Software Application Domains 7
1.1.3 Legacy Software 9
1.2 Software Engineering 10
1.3 The Software Process 12
1.4 Software Engineering Practice 15
1.4.1 The Essence of Practice 16
1.4.2 General Principles 17
1.5 Software Myths 19
1.6 Summary 22
PROBLEMS AND POINTS TO PONDER 22
FURTHER READINGS AND INFORMATION SOURCES 23

PART ONE THE SOFTWARE PROCESS

CHAPTER 2 PROCESS MODELS 26

2.1 A Generic Process Model 27
2.1.1 Defining a Framework Activity 28
2.1.2 Identifying a Task Set 30
2.1.3 Process Patterns 31

2.2 Prescriptive Process Models 32
2.2.1 The Waterfall Model 34
2.2.2 Incremental Process Models 36
2.2.3 Evolutionary Process Models 37
2.2.4 Concurrent Models 43
2.2.5 A Final Word on Evolutionary Processes 44
2.3 Specialized Process Models 45
2.3.1 Component-Based Development 45
2.3.2 The Formal Methods Model 46
2.3.3 Aspect-Oriented Software Development 47
2.4 The Unified Process 48
2.4.1 A Brief History 49
2.4.2 Phases of the Unified Process 49
2.5 Process Technology 51
2.6 Product and Process 52
2.7 Summary 53
PROBLEMS AND POINTS TO PONDER 54
FURTHER READINGS AND INFORMATION SOURCES 55

CHAPTER 3 AGILE DEVELOPMENT 56
3.1 What Is Agility? 58
3.2 Agility and the Cost of Change 58
3.3 What Is an Agile Process? 59
3.3.1 Agility Principles 60
3.3.2 The Politics of Agile Development 61
3.3.3 Human Factors 62
3.4 Extreme Programming (XP) 63
3.4.1 XP Values 63
3.4.2 The XP Process 64
3.4.3 Industrial XP 68
3.4.4 The XP Debate 69
3.5 Other Agile Process Models 71
3.5.1 Adaptive Software Development (ASD) 72
3.5.2 Scrum 73