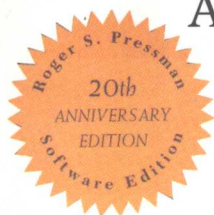


大学计算机教育丛书（影印版）

FIFTH EDITION

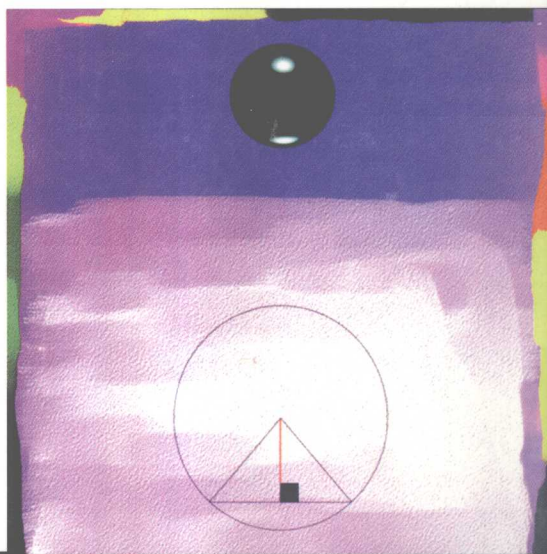
# SOFTWARE ENGINEERING

A PRACTITIONER'S APPROACH



ROGER S. PRESSMAN

## 软件工程 实践者之路 第5版



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



WCB  
McGraw-Hill

<http://www.mhhe.com>



TP311.5  
P147

# Software Engineering

**A PRACTITIONER'S APPROACH**

**FIFTH EDITION**

软 件 工 程

实践者之路

第 5 版

**Roger S. Pressman, Ph.D.**

清 华 大 学 出 版 社  
McGraw-Hill Companies, Inc.

## (京)新登字 158 号

Software Engineering: A Practitioner's Approach, 5th ed.

Roger S. Pressman

Copyright © 2001 by McGraw-Hill Companies, Inc.

Original English Language Edition published by McGraw-Hill Companies, Inc.

All rights Reserved.

For sale in Mainland China only.

本书影印版由 McGraw-Hill 出版公司授权清华大学出版社在中国境内(不包括香港、澳门特别行政区和台湾地区)独家出版、发行。

本书之任何部分未经出版者书面许可,不得用任何方式复制或抄袭。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

北京市版权局著作权合同登记号: 01-2000-3919

书 名: Software Engineering A Practitioner's Approach, 5th ed.

软件工程 实践者之路

作 者: Roger S. Pressman

出版者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京牛山世兴印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 56

版 次: 2001 年 1 月第 1 版 2002 年 10 月第 3 次印刷

书 号: ISBN 7-302-04139-3/TP·2442

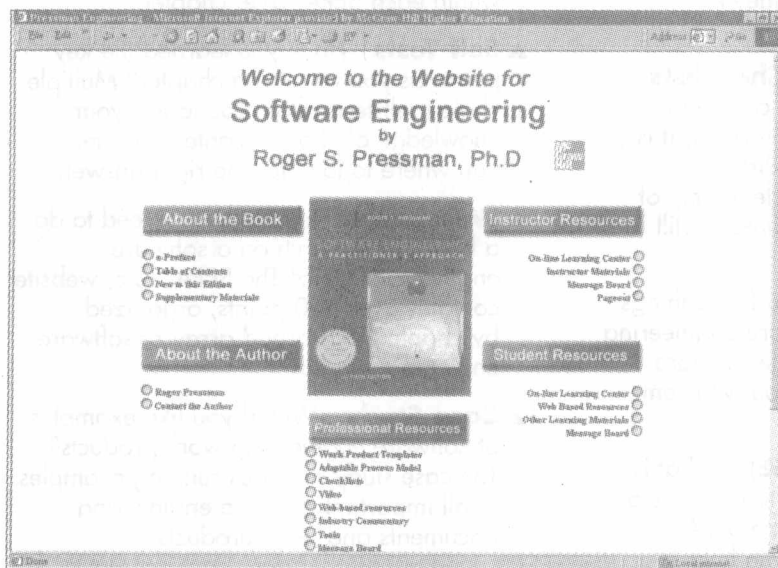
印 数: 6501~9000

定 价: 79.00 元

You get much more than a textbook when you use

# SOFTWARE ENGINEERING

## A PRACTITIONER'S APPROACH, 5/e



SEPA, 5/e, is accompanied by a comprehensive software engineering website, designed to provide valuable resources for students, instructors, and software professionals. Among the many resources you will be able to use as a reader of SEPA, 5/e, are:

### Instructor Resources

- ▲ **Instructor's Guide** - A complete Instructor's Guide for SEPA, 5/e, is available for download. The Guide includes chapter by chapter discussion of the book, suggestions for various course structures and term projects, teaching guidelines, and selected solutions to problems.
- ▲ **Powerpoint Slides** - A full set of MS powerpoint slides may be downloaded to supplement your lecture presentations.
- ▲ **Test Bank** - A collection of suggested test questions (with answers) is provided.
- ▲ **Tiny Tools** - A collection of simple software engineering tools that serve to implement approaches and techniques presented in SEPA, 5/e, is provided.
- ▲ **Page Out** - A tool that enables you to develop your own course specific website.
- ▲ **Message Board** - The message board allows communication among instructors who are using SEPA, 5/e, for teaching purposes.

The SEPA, 5/e, website is an important source of software engineering information, a gateway to additional information on the Web, and an important part of the learning experience.

Visit the SEPA, 5/e, website at <http://www.mhhe.com/engcs/compSCI/pressman>

## Professional Resources

- ▲ **Documents/Work Product Templates** - If you need examples of the format and content of software engineering work products, they're provided for download.
- ▲ **Software Engineering Checklists** - When you conduct a review or assess software engineering work products, it is a good idea to have a checklist to guide your evaluation. A wide variety of checklists are provided and links to still more are also included.
- ▲ **Tiny Tools** - There are lots of little things that lead to success in software engineering work. This collection of simple software engineering tools will help you with some of them.
- ▲ **Professional Tools (CASE)** - What is the best CASE tool for our situation? It's a commonly asked question. The SEPA, 5/e, website provides you with links to hundreds of CASE tools and more than a few tools comparisons.
- ▲ **Software Engineering Resources** - If you need to do some research or take an on-line tutorial, the SEPA, 5/e, website contains over 500 pointers to a broad array of software engineering topics.
- ▲ **Adaptable Process Model** - The adaptable process model is a comprehensive process template that can be tuned to your organization's needs.
- ▲ **An Industry Quality Video Curriculum** - An industry quality video curriculum, already used at hundreds of major companies worldwide, is available to augment SEPA, 5/e. Information about this product is available at the website.
- ▲ **Industry Commentary** - A collection of short essays, extracted from a number of industry sources, is provided to help spur thoughtful debate.

## Student Resources

- ▲ **Study Guide** - Need a quick review during exam time? The study guide can help by summarizing key points presented within each SEPA, 5/e, chapter.
- ▲ **Self-Tests** - Have you learned the key points as you've read a chapter? Multiple choice self-tests allow you to test your knowledge of chapter content and tell you where to look for the right answer.
- ▲ **Web-based Resources** - Need to do a bit more research on a software engineering topic? The SEPA, 5/e, website contains over 500 points, organized by chapter, to a broad array of software engineering topics.
- ▲ **Case Study** - Would you like examples of software engineering work products? The case study provides running examples of all important software engineering documents and work products.
- ▲ **Video** - Want to hear what Dr. Pressman has to say on a software engineering topic? Take a look!
- ▲ **Supplementary Content** - Need even more information? The site contains a collection of supplementary materials that expand on topics presented within the book.
- ▲ **Message Board** - Want to talk with other readers? The message board allows Q&A among students and others and provides a useful mechanism for informal communication.

## 出版者的话

今天,我们的大学生、研究生和教学、科研工作者,面临的是一个国际化的信息时代。他们将需要随时查阅大量的外文资料;会有更多的机会参加国际性学术交流活动;接待外国学者;走上国际会议的讲坛。作为科技工作者,他们不仅应有与国外同行进行口头和书面交流的能力,更为重要的是,他们必须具备极强的查阅外文资料获取信息的能力。有鉴于此,在原国家教委所颁布的“大学英语教学大纲”中有一条规定:专业阅读应作为必修课程开设。同时,在大纲中还规定了这门课程的学时和教学要求。有些高校除开设“专业阅读”课之外,还在某些专业课拟进行英语授课。但教、学双方都苦于没有一定数量的合适的英文原版教材作为教学参考书。为满足这方面的需要,我们陆续精选了一批国外计算机科学方面最新版本的著名教材,进行影印出版。我社获得国外著名出版公司和原作者的授权将国际先进水平的教材引入我国高等学校,为师生们提供了教学用书,相信会对高校教材改革产生积极的影响。

我们欢迎高校师生将使用影印版教材的效果、意见反馈给我们,更欢迎国内专家、教授积极向我社推荐国外优秀计算机教育教材,以利我们将《大学计算机教育丛书(影印版)》做得更好,更适合高校师生的需要。

清华大学出版社

《大学计算机教育丛书(影印版)》项目组

1999.6



## ABOUT THE AUTHOR

**R**oger S. Pressman is an internationally recognized authority in software process improvement and software engineering technologies. For over three 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 principle consultant, helping companies establish effective software engineering practices. He also designed and developed the company's software engineering training and process improvement products—*Essential Software Engineering*, a complete video curriculum that is among the industry's most comprehensive treatments of the subject, and *Process Advisor*, a self-directed system for software engineering process improvement. Both products are used by hundreds of companies worldwide.

Dr. Pressman has written many technical papers, is a regular contributor to industry periodicals, and is author of six books. In addition to *Software Engineering: A Practitioner's Approach*, he has written *A Manager's Guide to Software Engineering* (McGraw-Hill), an award-winning book that uses a unique Q&A format to present management guidelines for instituting and understanding software engineering technology; *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 is on the Editorial Boards of *IEEE Software* and the *Cutter IT Journal*, 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 has presented tutorials at the International Conference on Software Engineering and at many other industry meetings. He is a member of the ACM, IEEE, and Tau Beta Pi, Phi Kappa Phi, Eta Kappa Nu, and Pi Tau Sigma.

## PREFACE

**W**hen a 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 20 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 the most advanced technologies of the day. 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 fifth edition of *Software Engineering: A Practitioner's Approach* is intended to serve as a guide to a maturing engineering discipline. The fifth edition, like the four 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 format and style of the fifth edition have undergone significant change, making the presentation more reader-friendly and the content more easily accessible.

The fifth edition is considerably more than a simple update. The book has been revised to accommodate the dramatic growth in the field and to emphasize new and important software engineering practices. In addition, a comprehensive Web site has been developed to complement the content of the book. The Web site, which I call



*SepaWeb*, can be found at <http://www.mhhe.com/pressman>. Designed to be used in conjunction with the fifth edition of *Software Engineering: A Practitioner's Approach*, *SepaWeb* provides a broad array of software engineering resources that will benefit instructors, students, and industry professionals.

Like all Web sites, *SepaWeb* will evolve over time, but the following major content areas will always be present: (1) a broad array of *instructor resources* including a comprehensive on-line *Instructor's Guide* and supplementary teaching materials (e.g., slide presentations to supplement lectures, video-based instructional aids); (2) a wide variety of *student resources* including an extensive on-line learning center (encompassing study guides, Web-based resources, and self-tests), an evolving collection of "tiny tools," a case study, and additional supplementary content; and (3) a detailed collection of *professional resources* including 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. In addition, *SepaWeb* will contain other goodies that are currently in development.

The 32 chapters of the fifth 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 One, *The Product and the Process*, presents an introduction to the software engineering milieu. It is intended to introduce the subject matter, and more important, to present concepts that will be necessary for later chapters. Part Two, *Managing Software Projects*, presents topics that are relevant to those who plan, manage, and control a software development project. Part Three, *Conventional Methods for Software Engineering*, presents the classical analysis, design, and testing methods that some view as the "conventional" school of software engineering. Part Four, *Object-Oriented Software Engineering*, presents object-oriented methods across the entire software engineering process, including analysis, design, and testing. Part Five, *Advanced Software Engineering Topics*, presents dedicated chapters that address formal methods, cleanroom software engineering, component-based software engineering, client/server software engineering, Web engineering, reengineering, and CASE.

The five-part organization of the fifth 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 "design course" might emphasize only Part Three or Part Four; a "methods course" might present selected chapters in Parts Three, Four, and Five. A "management course" would stress Parts One and Two. By organizing the fifth edition in this way, I attempted to provide an instructor with a number of teaching options. *SepaWeb* can and should be used to supplement the content that is chosen from the book.

An *Instructor's Guide for Software Engineering: A Practitioner's Approach* is available from *SepaWeb*. The *Instructor's Guide* presents suggestions for conducting var-

ious 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 teaching aids.

A comprehensive video curriculum, *Essential Software Engineering*, is available to complement this book. The video curriculum has been designed for industry training and has been modularized to enable individual software engineering topics to be presented on an as-needed, when-needed basis. Further information on the video can be obtained by mailing the request card at the back of this book.<sup>1</sup>

My work on the five 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 as well as a new generation of contributors to electronic media (newsgroups, e-newsletters, and the World Wide Web) who have provided me with additional insight, ideas, and commentary over the past 20 years. Many have been referenced within the pages of each chapter. All deserve credit for their contribution to this rapidly evolving field. I also wish to thank the reviewers of the fifth edition: Donald H. Kraft, Louisiana State University; Panos E. Livadas, University of Florida; Joseph Lambert, Pennsylvania State University; Kenneth L. Modesitt, University of Michigan—Dearborn; and, James Purtilo, University of Maryland. Their comments and criticism have been invaluable. Special thanks and acknowledgement 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 design and pedagogical content.

The content of the fifth 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 teach me as much or more than I can 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

---

<sup>1</sup> If the request card is missing, please visit the R. S. Pressman & Associates, Inc. Web site at <http://www.rspa.com/ese> or e-mail a request for information to [info@rspa.com](mailto:info@rspa.com).

## USING THIS BOOK

The fifth edition of *Software Engineering: A Practitioner's Approach* (SEPA) has been redesigned to enhance your reading experience and to provide integrated links to the SEPA Web site, <http://www.mhhe.com/pressman/>. SepaWeb contains a wealth of useful supplementary information for readers of the book and a broad array of resources (e.g., an *Instructor's Guide*, classroom slides, and video supplements) for instructors who have adopted SEPA for classroom use.

A comprehensive video curriculum, *Essential Software Engineering*, is available to complement this book. The video curriculum has been designed for industry training and has been modularized to enable individual software engineering topics to be presented on an as-needed, when-needed basis. Further information on the video can be obtained by mailing the request card at the back of this book.<sup>1</sup>

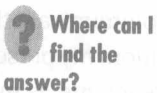
Throughout the book, you will encounter marginal icons that should be interpreted in the following manner:



Used to emphasize an important point in the body of the text.



Practical advice from the real world of software engineering.



Where can I find the answer?

### XRef

Provides an important cross reference within the book.



"Important words"

The **keypoint** icon will help you to find important points quickly.

The **advice** icon provides pragmatic guidance that can help you make the right decision or avoid common problems while building software.

The **question mark** icon asks common questions that are answered in the body of the text.

The **xref** icon will point you to another part of the book where information relevant to the current discussion can be found.

The **quote** icon presents interesting quotes that have relevance to the topic at hand.

### WebRef

For pointers that will take you directly to Web resources



A selected topic

The **WebRef** icon provides direct pointers to important software engineering related Web sites.

The **SepaWeb** pointer indicates that further information about the noted topic is available at the SEPA Web site.



The **SepaWeb.checklists** icon points you to detailed checklists that will help you to assess the software engineering work you're doing and the work products you produce.



The **SepaWeb.documents** icon points you to detailed document outlines, descriptions and examples contained within the SEPA Web site.

<sup>1</sup> If the card is missing, please visit the R.S. Pressman & Associates, Inc. Web site at <http://www.rsps.com/ese>, or e-mail to [info@rsps.com](mailto:info@rsps.com).

# CONTENTS AT A GLANCE

*Preface* xxv

## **PART ONE      The Product and the Process      1**

---

- CHAPTER 1      The Product      3  
CHAPTER 2      The Process      19

## **PART TWO      Managing Software Projects      53**

---

- CHAPTER 3      Project Management Concepts      55  
CHAPTER 4      Software Process and Project Metrics      79  
CHAPTER 5      Software Project Planning      113  
CHAPTER 6      Risk Analysis and Management      145  
CHAPTER 7      Project Scheduling and Tracking      165  
CHAPTER 8      Software Quality Assurance      193  
CHAPTER 9      Software Configuration Management      225

## **PART THREE      Conventional Methods for Software Engineering      243**

---

- CHAPTER 10      System Engineering      245  
CHAPTER 11      Analysis Concepts and Principles      271  
CHAPTER 12      Analysis Modeling      299  
CHAPTER 13      Design Concepts and Principles      335  
CHAPTER 14      Architectural Design      365  
CHAPTER 15      User Interface Design      401  
CHAPTER 16      Component-Level Design      423  
CHAPTER 17      Software Testing Techniques      437  
CHAPTER 18      Software Testing Strategies      477  
CHAPTER 19      Technical Metrics for Software      507

## **PART FOUR      Object-Oriented Software Engineering      539**

---

- CHAPTER 20      Object-Oriented Concepts and Principles      541  
CHAPTER 21      Object-Oriented Analysis      571  
CHAPTER 22      Object-Oriented Design      603

CHAPTER 23	Object-Oriented Testing	631
CHAPTER 24	Technical Metrics for Object-Oriented Systems	653

---

<b>PART FIVE</b>	<b>Advanced Topics in Software Engineering</b>	<b>671</b>
------------------	--	------------

---

CHAPTER 25	Formal Methods	673
CHAPTER 26	Cleanroom Software Engineering	699
CHAPTER 27	Component-Based Software Engineering	721
CHAPTER 28	Client/Server Software Engineering	747
CHAPTER 29	Web Engineering	769
CHAPTER 30	Reengineering	799
CHAPTER 31	Computer-Aided Software Engineering	825
CHAPTER 32	The Road Ahead	845

# TABLE OF CONTENTS

## **PART ONE—THE PRODUCT AND THE PROCESS 1**

### **CHAPTER 1 THE PRODUCT 3**

1.1	The Evolving Role of Software	4
1.2	Software	6
1.2.1	Software Characteristics	6
1.2.2	Software Applications	9
1.3	Software: A Crisis on the Horizon?	11
1.4	Software Myths	12
1.5	Summary	15
	REFERENCES	15
	PROBLEMS AND POINTS TO PONDER	16
	FURTHER READINGS AND INFORMATION SOURCES	17

### **CHAPTER 2 THE PROCESS 19**

2.1	Software Engineering: A Layered Technology	20
2.1.1	Process, Methods, and Tools	20
2.1.2	A Generic View of Software Engineering	21
2.2	The Software Process	23
2.3	Software Process Models	26
2.4	The Linear Sequential Model	28
2.5	The Prototyping Model	30
2.6	The RAD Model	32
2.7	Evolutionary Software Process Models	34
2.7.1	The Incremental Model	35
2.7.2	The Spiral Model	36
2.7.3	The WINWIN Spiral Model	38
2.7.4	The Concurrent Development Model	40
2.8	Component-Based Development	42
2.9	The Formal Methods Model	43
2.10	Fourth Generation Techniques	44
2.11	Process Technology	46
2.12	Product and Process	46
2.13	Summary	47
	REFERENCES	47
	PROBLEMS AND POINTS TO PONDER	49
	FURTHER READINGS AND INFORMATION SOURCES	50



**PART TWO—MANAGING SOFTWARE PROJECTS 53****CHAPTER 3 PROJECT MANAGEMENT CONCEPTS 55**


---

3.1	The Management Spectrum	56
3.1.1	The People	56
3.1.2	The Product	57
3.1.2	The Process	57
3.1.3	The Project	57
3.2	People	58
3.2.1	The Players	58
3.2.2	Team Leaders	59
3.2.3	The Software Team	60
3.2.4	Coordination and Communication Issues	65
3.3	The Product	67
3.3.1	Software Scope	67
3.3.2	Problem Decomposition	67
3.4	The Process	68
3.4.1	Melding the Product and the Process	69
3.4.2	Process Decomposition	70
3.5	The Project	71
3.6	The W <sup>5</sup> HH Principle	73
3.7	Critical Practices	74
3.8	Summary	74
	REFERENCES	75
	PROBLEMS AND POINTS TO PONDER	76
	FURTHER READINGS AND INFORMATION SOURCES	77

**CHAPTER 4 SOFTWARE PROCESS AND PROJECT METRICS 79**


---

4.1	Measures, Metrics, and Indicators	80
4.2	Metrics in the Process and Project Domains	81
4.2.1	Process Metrics and Software Process Improvement	82
4.2.2	Project Metrics	86
4.3	Software Measurement	87
4.3.1	Size-Oriented Metrics	88
4.3.2	Function-Oriented Metrics	89
4.3.3	Extended Function Point Metrics	91
4.4	Reconciling Different Metrics Approaches	94
4.5	Metrics for Software Quality	95
4.5.1	An Overview of Factors That Affect Quality	95
4.5.2	Measuring Quality	96
4.5.3	Defect Removal Efficiency	98
4.6	Integrating Metrics Within the Software Engineering Process	98
4.6.1	Arguments for Software Metrics	99
4.6.2	Establishing a Baseline	100
4.6.3	Metrics Collection, Computation, and Evaluation	100
4.7	Managing Variation: Statistical Quality Control	100
4.8	Metrics for Small Organizations	104
4.9	Establishing a Software Metrics Program	105
4.10	Summary	107
	REFERENCES	107

PROBLEMS AND POINTS TO PONDER	109
FURTHER READINGS AND INFORMATION SOURCES	110

## **CHAPTER 5 SOFTWARE PROJECT PLANNING 113**

5.1	Observations on Estimating	114
5.2	Project Planning Objectives	115
5.3	Software Scope	115
5.3.1	Obtaining Information Necessary for Scope	116
5.3.2	Feasibility	117
5.3.3	A Scoping Example	118
5.4	Resources	120
5.4.1	Human Resources	121
5.4.2	Reusable Software Resources	121
5.4.3	Environmental Resources	122
5.5	Software Project Estimation	123
5.6	Decomposition Techniques	124
5.6.1	Software Sizing	124
5.6.2	Problem-Based Estimation	126
5.6.3	An Example of LOC-Based Estimation	128
5.6.4	An Example of FP-Based Estimation	129
5.6.4	Process-Based Estimation	130
5.6.5	An Example of Process-Based Estimation	131
5.7	Empirical Estimation Models	132
5.7.1	The Structure of Estimation Models	132
5.7.2	The COCOMO Model	133
5.7.3	The Software Equation	135
5.8	The Make/Buy Decision	136
5.8.1	Creating a Decision Tree	137
5.8.2	Outsourcing	138
5.9	Automated Estimation Tools	139
5.10	Summary	140
	REFERENCES	140
	PROBLEMS AND POINTS TO PONDER	141
	FURTHER READINGS AND INFORMATION SOURCES	142

## **CHAPTER 6 RISK ANALYSIS AND MANAGEMENT 145**

6.1	Reactive versus Proactive Risk Strategies	146
6.2	Software Risks	146
6.3	Risk Identification	148
6.3.1	Assessing Overall Project Risk	149
6.3.2	Risk Components and Drivers	149
6.4	Risk Projection	151
6.4.1	Developing a Risk Table	151
6.4.2	Assessing Risk Impact	153
6.4.3	Risk Assessment	154
6.5	Risk Refinement	156
6.6	Risk Mitigation, Monitoring, and Management	156
6.7	Safety Risks and Hazards	158
6.8	The RMMM Plan	159
6.9	Summary	159
	REFERENCES	160

PROBLEMS AND POINTS TO PONDER	161
FURTHER READINGS AND INFORMATION SOURCES	162

## **CHAPTER 7 PROJECT SCHEDULING AND TRACKING 165**

7.1	Basic Concepts	166
7.1.1	Comments on "Lateness"	167
7.2.1	Basic Principles	168
7.2	The Relationship Between People and Effort	170
7.2.1	An Example	170
7.2.2	An Empirical Relationship	171
7.2.3	Effort Distribution	172
7.3	Defining a Task Set for the Software Project	172
7.3.1	Degree of Rigor	173
7.3.2	Defining Adaptation Criteria	174
7.3.3	Computing a Task Set Selector Value	175
7.3.4	Interpreting the TSS Value and Selecting the Task Set	176
7.4	Selecting Software Engineering Tasks	177
7.5	Refinement of Major Tasks	178
7.6	Defining a Task Network	180
7.7	Scheduling	181
7.7.1	Timeline Charts	182
7.7.2	Tracking the Schedule	185
7.8	Earned Value Analysis	186
7.9	Error Tracking	187
7.10	The Project Plan	189
7.11	Summary	189
	REFERENCES	189
	PROBLEMS AND POINTS TO PONDER	190
	FURTHER READINGS AND INFORMATION SOURCES	192

## **CHAPTER 8 SOFTWARE QUALITY ASSURANCE 193**

8.1	Quality Concepts	194
8.1.1	Quality	195
8.1.2	Quality Control	196
8.1.3	Quality Assurance	196
8.1.4	Cost of Quality	196
8.2	The Quality Movement	198
8.3	Software Quality Assurance	199
8.3.1	Background Issues	200
8.3.2	SQA Activities	201
8.4	Software Reviews	202
8.4.1	Cost Impact of Software Defects	203
8.4.2	Defect Amplification and Removal	204
8.5	Formal Technical Reviews	205
8.5.1	The Review Meeting	206
8.5.2	Review Reporting and Record Keeping	207
8.5.3	Review Guidelines	207
8.6	Formal Approaches to SQA	209
8.7	Statistical Software Quality Assurance	209
8.8	Software Reliability	212
8.8.1	Measures of Reliability and Availability	212
8.8.2	Software Safety	213