

Fourth Edition

# SOFTWARE ENGINEERING

## A PRACTITIONER'S APPROACH

# 软件工程

## 实践者的研究方法

(英文版·第4版)

(美) Roger S. Pressman 著  
R. S. Pressman & Associates公司



机械工业出版社  
China Machine Press



McGraw-Hill

计算机科学丛书

# 软件工程 实践者的研究方法

(英文版·第4版)

Software Engineering  
A Practitioner's Approach

(Fourth Edition)

(美) Roger S. Pressman 著  
(R. S. Pressman & Associates 公司)



机械工业出版社  
China Machine Press



*McGraw-Hill*

Roger S. Pressman : Software Engineering, A Practitioner's Approach, Fourth Edition.

Copyright © 1997,1992,1987,1982 by The McGraw-Hill Companies, Inc. All rights reserved.  
Jointly published by China Machine Press/McGraw-Hill. This edition may be sold in the People's Republic of China only. This book cannot be re-exported and is not for sale outside the People's Republic of China.

RISBN 007-1166947

本书英文影印版由 McGraw-Hill 公司授权机械工业出版社在中国大陆境内独家出版发行, 未经出版者许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

版权所有, 侵权必究。

本书版权登记号: 图字: 01-99-0112

### 图书在版编目(CIP)数据

软件工程:实践者的研究方法(第4版): 英文/(美)普莱斯曼(Pressman, R. S.)著, -影印版. -北京: 机械工业出版社, 1999.3

(计算机科学丛书)

ISBN 7-111-06711-8

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 (1999) 第02386号

出 版 者: 马九荣(北京市西城区百万庄大街22号 邮政编码 100037)  
北京第二外国语学院印刷厂印刷·新华书店北京发行所发行  
1999年3月第1版第1次印刷

787mm × 1092mm 1/16 · 55印张

印数: 0001-5000册

定价: 68.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

## ABOUT THE AUTHOR

**R**oger S. Pressman is an internationally recognized consultant and author in software engineering. He received a B.S.E. (cum laude) from the University of Connecticut, an M.S. from the University of Bridgeport and a Ph.D. in engineering from the University of Connecticut, and has over 25 years of industry experience, holding both technical and management positions with responsibility for the development of software for engineered products and systems.

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

In addition to his industry experience, Dr. Pressman was 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 President of R.S. Pressman & Associates, Inc., a consulting firm specializing in software engineering methods and training. He serves as principal consultant, specializing in helping companies establish effective software engineering practices. He developed the RSP&A software engineering assessment method, a unique blend of quantitative and qualitative analysis that helps clients assess their current state of software engineering practice.

In addition to consulting services rendered to many Fortune 500 clients, R.S. Pressman & Associates, Inc. markets a wide variety of software engineering training products and process improvement services. The company has developed a state-of-the-art video curriculum, *Essential Software Engineering*, which is among the industry's most comprehensive treatments of the subject.

Another product, *Process Advisor*, is a self-directed system for software process improvement.

Dr. Pressman is author of 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 *Making Software Engineering Happen* (Prentice Hall), the first book to address the critical management problems associated with software engineering process improvement, *Software Shock* (Dorset House), a treatment of software and its impact on business and society, and *A Manager's Guide to Software Engineering* (McGraw-Hill), a book that uses a unique Q&A format to present management guidelines for instituting and understanding the technology. Dr. Pressman is on the editorial boards of *American Programmer* and *IEEE Software*, and is editor of the "Manager" column in *IEEE Software*. He is a member of the ACM, IEEE, and Tau Beta Pi, Phi Kappa Phi, Eta Kappa Nu, and Pi Tau Sigma.

# PREFACE

**A**s software engineering moves into its fourth decade of existence, it suffers from many of the strengths and some of the frailties that are experienced by humans of the same age. The innocence and enthusiasm of its early years have been replaced by more reasonable expectations (and even a healthy cynicism) fostered by years of experience. Software engineering approaches its mid-life with many accomplishments, but with significant work yet to do. Today, it is recognized as a legitimate discipline, one 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. Managers and practitioners alike recognize the need for a more disciplined approach to software.

But many of the problems discussed in earlier editions of this book remain with us. Many individuals and companies still develop software haphazardly. Many professionals and students are unaware of modern methods. And as a result, the quality of the software that we produce suffers. 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 fourth edition of *Software Engineering: A Practitioner's Approach* is intended to serve as a guide to a maturing engineering discipline. The fourth edition, like the three editions that have preceded it, is intended for both students and practitioners, and maintains the same format and style of its predecessors. The book retains its appeal as a guide to the industry professional and a com-

prehensive introduction to the student at the upper level undergraduate or first year graduate level.

The fourth edition is considerably more than a simple update. The book has been completely restructured to accommodate the dramatic growth in the field and to emphasize new and important software engineering methods. Chapters that have been retained from earlier editions have been revised and updated. Twelve new chapters have been added to provide more complete treatment of contemporary trends and techniques. Many new examples, problems and points to ponder have been included. The *Further Readings and Other Information Sources* sections (one of the more popular tidbits in earlier editions) have been expanded for every chapter. Hundreds of new published sources and over 160 sources from the World Wide Web<sup>1</sup> have been included.

The 30 chapters of the fourth 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 importantly, 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 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, reuse, reengineering, client/server software engineering, and CASE.

It is important to note that the fourth edition has a much greater emphasis on metrics and measurement than earlier editions. Three separate chapters on software metrics address measurement of the software process, technical metrics for analysis, design, and testing using conventional methods, and technical metrics for object-oriented software engineering.

The five-part organization of the fourth 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 III or Part IV; a “methods course” might present selected chapters in Parts III, IV, and V. A “management course” would stress Parts I and II. By organizing the fourth edition in this way, I have attempted to provide an instructor with a number of teaching options.

---

<sup>1</sup>Because World Wide Web addresses change frequently, only those that are likely to persist have been noted in *Further Readings and Other Information Sources*. However, even these may become invalid as time passes. An up-to-date “hot list” of the URLs presented in this book can be found at <http://www.rspa.com>.

Like the first three editions, an *Instructor's Guide for Software Engineering: A Practitioner's Approach* is available from McGraw-Hill. The *Instructor's Guide* 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 transparency masters to aid in teaching selected topics. In addition, 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>2</sup>

My work on the four 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 and the World Wide Web) who have provided me with additional insight, ideas, and commentary over the past 15 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 fourth edition: Frank H. Westervelt, Wayne State University; Steven A. Demurjian, The University of Connecticut; Chung Lee, California State Polytechnic University; Alan Davis, University of Colorado; Michael C. Mah, QSM Associates; Richard N. Taylor, University of California-Irvine; Osman Balci, Virginia Tech; James H. Cross, Auburn University; Warren Harrison, Portland State University; Mieczyslaw M. Kokar, Northeastern University. Their comments and criticism have been invaluable.

The content of the fourth 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 throughout North America and Europe, who certainly teach me as much as 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 and character have been an inspiration to me. Nothing has filled me with more pride. And finally, to Barbara, my love and thanks for tolerating my travel schedule, understanding the evenings at the office, and encouraging still another edition of "the book."

**Roger S. Pressman**

---

<sup>2</sup>Instructors at accredited universities should see their *Instructor's Guide* for special options associated with the video curriculum. If the reply card is missing, please e-mail a request for information to [ESEinfo@rspa.com](mailto:ESEinfo@rspa.com) or fax a request to R.S. Pressman & Associates, Inc. at (203) 799-1023.



# CONTENTS AT A GLANCE

## PREFACE

<b>PART ONE</b>	<b>THE PRODUCT AND THE PROCESS</b>	<b>1</b>
	CHAPTER 1 THE PRODUCT	3
	CHAPTER 2 THE PROCESS	22
<b>PART TWO</b>	<b>MANAGING SOFTWARE PROJECTS</b>	<b>55</b>
	CHAPTER 3 PROJECT MANAGEMENT CONCEPTS	57
	CHAPTER 4 SOFTWARE PROCESS AND PROJECT METRICS	76
	CHAPTER 5 SOFTWARE PROJECT PLANNING	102
	CHAPTER 6 RISK MANAGEMENT	132
	CHAPTER 7 PROJECT SCHEDULING AND TRACKING	153
	CHAPTER 8 SOFTWARE QUALITY ASSURANCE	179
	CHAPTER 9 SOFTWARE CONFIGURATION MANAGEMENT	209
<b>PART THREE</b>	<b>CONVENTIONAL METHODS FOR SOFTWARE ENGINEERING</b>	<b>229</b>
	CHAPTER 10 SYSTEM ENGINEERING	231
	CHAPTER 11 ANALYSIS CONCEPTS AND PRINCIPLES	270
	CHAPTER 12 ANALYSIS MODELING	298
	CHAPTER 13 DESIGN CONCEPTS AND PRINCIPLES	341

CHAPTER 14	DESIGN METHODS	371
CHAPTER 15	DESIGN FOR REAL-TIME SYSTEMS	423
CHAPTER 16	SOFTWARE TESTING TECHNIQUES	448
CHAPTER 17	SOFTWARE TESTING STRATEGIES	487
CHAPTER 18	TECHNICAL METRICS FOR SOFTWARE	517

---

<b>PART FOUR</b>	<b>OBJECT-ORIENTED SOFTWARE ENGINEERING</b>	<b>549</b>
------------------	---	------------

CHAPTER 19	OBJECT-ORIENTED CONCEPTS AND PRINCIPLES	551
CHAPTER 20	OBJECT-ORIENTED ANALYSIS	581
CHAPTER 21	OBJECT-ORIENTED DESIGN	614
CHAPTER 22	OBJECT-ORIENTED TESTING	644
CHAPTER 23	TECHNICAL METRICS FOR OBJECT-ORIENTED SYSTEMS	664

---

<b>PART FIVE</b>	<b>ADVANCED TOPICS IN SOFTWARE ENGINEERING</b>	<b>679</b>
------------------	--	------------

CHAPTER 24	FORMAL METHODS	681
CHAPTER 25	CLEANROOM SOFTWARE ENGINEERING	707
CHAPTER 26	SOFTWARE REUSE	728
CHAPTER 27	REENGINEERING	756
CHAPTER 28	CLIENT/SERVER SOFTWARE ENGINEERING	784
CHAPTER 29	COMPUTER-AIDED SOFTWARE ENGINEERING	805
CHAPTER 30	THE ROAD AHEAD	826

<b>INDEX</b>	<b>840</b>
--------------	------------

# TABLE OF CONTENTS

## PREFACE

<b>PART ONE</b>	<b>THE PRODUCT AND THE PROCESS</b>	<b>1</b>
	CHAPTER 1 THE PRODUCT	3
	1.1 THE EVOLVING ROLE OF SOFTWARE	4
	1.1.1 An Industry Perspective	7
	1.1.2 An Aging Software Plant	8
	1.1.3 Software Competitiveness	9
	1.2 SOFTWARE	9
	1.2.1 Software Characteristics	10
	1.2.2 Software Components	13
	1.2.3 Software Applications	14
	1.3 SOFTWARE: A CRISIS ON THE HORIZON	16
	1.4 SOFTWARE MYTHS	17
	1.5 SUMMARY	19
	REFERENCES	19
	PROBLEMS AND POINTS TO PONDER	20
	FURTHER READINGS AND INFORMATION SOURCES	21
	CHAPTER 2 THE PROCESS	22
	2.1 SOFTWARE ENGINEERING—A LAYERED TECHNOLOGY	22
	2.1.1 Process, Methods, and Tools	23
	2.1.2 A Generic View of Software Engineering	24

2.2	THE SOFTWARE PROCESS	26
2.3	SOFTWARE PROCESS MODELS	28
2.4	THE LINEAR SEQUENTIAL MODEL	29
2.5	THE PROTOTYPING MODEL	32
2.6	THE RAD MODEL	34
2.7	EVOLUTIONARY SOFTWARE PROCESS MODELS	37
2.7.1	The Incremental Model	37
2.7.2	The Spiral Model	39
2.7.3	The Component Assembly Model	42
2.7.4	The Concurrent Development Model	43
2.8	THE FORMAL METHODS MODEL	45
2.9	FOURTH GENERATION TECHNIQUES	46
2.10	PROCESS TECHNOLOGY	47
2.11	PRODUCT AND PROCESS	48
2.12	SUMMARY	49
	REFERENCES	49
	PROBLEMS AND POINTS TO PONDER	50
	FURTHER READINGS AND OTHER INFORMATION SOURCES	51

## **PART TWO      MANAGING SOFTWARE PROJECTS      55**

CHAPTER 3	PROJECT MANAGEMENT CONCEPTS	57
3.1	THE MANAGEMENT SPECTRUM	58
3.1.1	People	58
3.1.2	The Problem	58
3.1.3	The Process	59
3.2	PEOPLE	59
3.2.1	The Players	60
3.2.2	Team Leaders	60
3.2.3	The Software Team	61
3.2.4	Coordination and Communication Issues	65
3.3	THE PROBLEM	66
3.3.1	Software Scope	66
3.3.2	Problem Decomposition	67
3.4	THE PROCESS	68
3.4.1	Melding the Problem and the Process	69
3.4.2	Process Decomposition	70
3.5	THE PROJECT	71
3.6	SUMMARY	72
	REFERENCES	72
	PROBLEMS AND POINTS TO PONDER	73
	FURTHER READINGS AND OTHER INFORMATION SOURCES	74
CHAPTER 4	SOFTWARE PROCESS AND PROJECT METRICS	76
4.1	MEASURES, METRICS, AND INDICATORS	77
4.2	METRICS IN THE PROCESS AND PROJECT DOMAINS	77

4.2.1	Process Metrics and Software Process Improvement	78
4.2.2	Project Metrics	82
4.3	SOFTWARE MEASUREMENT	83
4.3.1	Size-Oriented Metrics	84
4.3.2	Function-Oriented Metrics	85
4.3.3	Extended Function Point Metrics	87
4.4	RECONCILING DIFFERENT METRICS APPROACHES	90
4.5	METRICS FOR SOFTWARE QUALITY	92
4.5.1	An Overview of Factors That Affect Quality	92
4.5.2	Measuring Quality	93
4.5.3	Defect Removal Efficiency	94
4.6	INTEGRATING METRICS WITHIN THE SOFTWARE PROCESS	95
4.7	SUMMARY	97
	REFERENCES	98
	PROBLEMS AND POINTS TO PONDER	99
	FURTHER READINGS AND OTHER INFORMATION SOURCES	100
CHAPTER 5 SOFTWARE PROJECT PLANNING		102
5.1	OBSERVATIONS ON ESTIMATING	102
5.2	PROJECT PLANNING OBJECTIVES	104
5.3	SOFTWARE SCOPE	104
5.3.1	Obtaining Information Necessary for Scope	104
5.3.2	A Scoping Example	106
5.4	RESOURCES	108
5.4.1	Human Resources	109
5.4.2	Reusable Software Resources	109
5.4.3	Environmental Resources	110
5.5	SOFTWARE PROJECT ESTIMATION	111
5.6	DECOMPOSITION TECHNIQUES	112
5.6.1	Software Sizing	112
5.6.2	Problem-Based Estimation	113
5.6.3	An Example of LOC-Based Estimation	115
5.6.4	An Example of FP-Based Estimation	116
5.6.5	Process-Based Estimation	118
5.6.6	An Example of Process-Based Estimation	118
5.7	EMPIRICAL ESTIMATION MODELS	120
5.7.1	The Structure of Estimation Models	120
5.7.2	The COCOMO Model	121
5.7.3	The Software Equation	124
5.8	THE MAKE-BUY DECISION	125
5.8.1	Creating a Decision Tree	126
5.8.2	Outsourcing	127
5.9	AUTOMATED ESTIMATION TOOLS	128
5.10	SUMMARY	129
	REFERENCES	129
	PROBLEMS AND POINTS TO PONDER	129
	FURTHER READINGS AND OTHER INFORMATION SOURCES	130

CHAPTER 6 RISK MANAGEMENT	132
6.1 REACTIVE VS. PROACTIVE RISK STRATEGIES	133
6.2 SOFTWARE RISKS	133
6.3 RISK IDENTIFICATION	134
6.3.1 Product Size Risks	135
6.3.2 Business Impact Risks	136
6.3.3 Customer-Related Risks	136
6.3.4 Process Risks	137
6.3.5 Technology Risk	139
6.3.6 Development Environment Risks	139
6.3.7 Risks Associated with Staff Size and Experience	140
6.3.8 Risk Components and Drivers	141
6.4 RISK PROJECTION	141
6.4.1 Developing a Risk Table	141
6.4.2 Assessing Risk Impact	144
6.4.3 Risk Assessment	145
6.5 RISK MITIGATION, MONITORING, AND MANAGEMENT	146
6.6 SAFETY RISKS AND HAZARDS	148
6.7 THE RMMM PLAN	149
6.8 SUMMARY	149
REFERENCES	150
PROBLEMS AND POINTS TO PONDER	150
FURTHER READINGS AND OTHER INFORMATION SOURCES	151
CHAPTER 7 PROJECT SCHEDULING AND TRACKING	153
7.1 BASIC CONCEPTS	154
7.1.1 Comments on "lateness"	154
7.1.2 Basic Principles	156
7.2 THE RELATIONSHIP BETWEEN PEOPLE AND EFFORT	157
7.2.1 An Example	158
7.2.2 An Empirical Relationship	158
7.2.3 Effort Distribution	159
7.3 DEFINING A TASK SET FOR THE SOFTWARE PROJECT	160
7.3.1 Degree of Rigor	161
7.3.2 Defining Adaptation Criteria	161
7.3.3 Computing a Task Set Selector Value	162
7.3.4 Interpreting the TSS Value and Selecting the Task Set	163
7.4 SELECTING SOFTWARE ENGINEERING TASKS	164
7.5 REFINEMENT OF MAJOR TASKS	165
7.6 DEFINING A TASK NETWORK	168
7.7 SCHEDULING	170
7.7.1 Timeline Charts	170
7.7.2 Tracking the Schedule	172
7.8 THE PROJECT PLAN	174
7.8 SUMMARY	175
REFERENCES	176
PROBLEMS AND POINTS TO PONDER	176
FURTHER READINGS AND OTHER INFORMATION SOURCES	177

CHAPTER 8 SOFTWARE QUALITY ASSURANCE	179
8.1 QUALITY CONCEPTS	180
8.1.1 Quality	181
8.1.2 Quality Control	181
8.1.3 Quality Assurance	182
8.1.4 Cost of Quality	182
8.2 THE QUALITY MOVEMENT	184
8.3 SOFTWARE QUALITY ASSURANCE	185
8.3.1 Background Issues	185
8.3.2 SQA Activities	186
8.4 SOFTWARE REVIEWS	187
8.4.1 Cost Impact of Software Defects	188
8.4.2 Defect Amplification and Removal	189
8.5 FORMAL TECHNICAL REVIEWS	190
8.5.1 The Review Meeting	191
8.5.2 Review Reporting and Record Keeping	192
8.5.3 Review Guidelines	193
8.6 FORMAL APPROACHES TO SQA	194
8.7 STATISTICAL QUALITY ASSURANCE	195
8.8 SOFTWARE RELIABILITY	197
8.8.1 Measures of Reliability and Availability	198
8.8.2 Software Safety and Hazard Analysis	198
8.9 THE SQA PLAN	200
8.10 THE ISO 9000 QUALITY STANDARDS	202
8.10.1 The ISO Approach to Quality Assurance Systems	202
8.10.2 The ISO 9001 Standard	202
8.11 SUMMARY	203
REFERENCES	204
PROBLEMS AND POINTS TO PONDER	205
FURTHER READINGS AND OTHER INFORMATION SOURCES	206
 CHAPTER 9 SOFTWARE CONFIGURATION MANAGEMENT	 209
9.1 SOFTWARE CONFIGURATION MANAGEMENT	210
9.1.1 Baselines	210
9.1.2 Software Configuration Items	212
9.2 THE SCM PROCESS	214
9.3 IDENTIFICATION OF OBJECTS IN THE SOFTWARE CONFIGURATION	215
9.4 VERSION CONTROL	218
9.5 CHANGE CONTROL	220
9.6 CONFIGURATION AUDIT	223
9.7 STATUS REPORTING	224
9.8 SCM STANDARDS	224
9.9 SUMMARY	224
REFERENCES	225
PROBLEMS AND POINTS TO PONDER	226
FURTHER READINGS AND OTHER INFORMATION SOURCES	226

## **PART THREE      CONVENTIONAL METHODS FOR SOFTWARE ENGINEERING      229**

CHAPTER 10 SYSTEM ENGINEERING	231
10.1 COMPUTER-BASED SYSTEMS	232
10.2 THE SYSTEM ENGINEERING HIERARCHY	234
10.2.1 System Modeling	235
10.2.2 Information Engineering: An Overview	237
10.2.3 Product Engineering: An Overview	239
10.3 INFORMATION ENGINEERING	241
10.4 INFORMATION STRATEGY PLANNING	241
10.4.1 Enterprise Modeling	242
10.4.2 Business-Level Data Modeling	244
10.5 BUSINESS AREA ANALYSIS	245
10.5.1 Process Modeling	247
10.5.2 Information Flow Modeling	247
10.6 PRODUCT ENGINEERING	250
10.6.1 System Analysis	253
10.6.2 Identification of Need	253
10.6.3 Feasibility Study	253
10.6.4 Economic Analysis	255
10.6.5 Technical Analysis	256
10.7 MODELING THE SYSTEM ARCHITECTURE	259
10.8 SYSTEM MODELING AND SIMULATION	262
10.9 SYSTEM SPECIFICATION	264
10.10 SUMMARY	264
REFERENCES	266
PROBLEMS AND POINTS TO PONDER	266
FURTHER READINGS AND OTHER INFORMATION SOURCES	268
CHAPTER 11 ANALYSIS CONCEPTS AND PRINCIPLES	270
11.1 REQUIREMENTS ANALYSIS	270
11.2 COMMUNICATION TECHNIQUES	272
11.2.1 Initiating the Process	273
11.2.2 Facilitated Application Specification Techniques	274
11.2.3 Quality Function Deployment	277
11.3 ANALYSIS PRINCIPLES	278
11.3.1 The Information Domain	279
11.3.2 Modeling	281
11.3.3 Partitioning	282
11.3.4 Essential and Implementation Views	284
11.4 SOFTWARE PROTOTYPING	285
11.4.1 Selecting the Prototyping Approach	285
11.4.2 Prototyping Methods and Tools	287
11.5 SPECIFICATION	288
11.5.1 Specification Principles	288
11.5.2 Representation	289
11.5.3 The Software Requirements Specification	290
11.6 SPECIFICATION REVIEW	292



11.7 SUMMARY	293
REFERENCES	294
PROBLEMS AND POINTS TO PONDER	295
FURTHER READINGS AND OTHER INFORMATION SOURCES	296
 CHAPTER 12 ANALYSIS MODELING	 298
12.1 A BRIEF HISTORY	299
12.2 THE ELEMENTS OF THE ANALYSIS MODEL	300
12.3 DATA MODELING	301
12.3.1 Data Objects, Attributes, and Relationships	301
12.3.2 Cardinality and Modality	304
12.3.3 Entity-Relationship Diagrams	305
12.4 FUNCTIONAL MODELING AND INFORMATION FLOW	309
12.4.1 Data Flow Diagrams	309
12.4.2 Extensions for Real-Time Systems	312
12.4.3 Ward and Mellor Extensions	312
12.4.4 Hatley and Pirbhai Extensions	315
12.5 BEHAVIORAL MODELING	316
12.6 THE MECHANICS OF STRUCTURED ANALYSIS	320
12.6.1 Creating an Entity-Relationship Diagram	321
12.6.2 Creating a Data Flow Model	323
12.6.3 Creating a Control Flow Model	325
12.6.4 The Control Specification	328
12.6.5 The Process Specification	330
12.7 THE DATA DICTIONARY	330
12.8 AN OVERVIEW OF OTHER CLASSICAL ANALYSIS METHODS	334
12.8.1 Data Structured Systems Development	334
12.8.2 Jackson System Development	335
12.8.3 SADT	335
12.9 SUMMARY	336
REFERENCES	336
PROBLEMS AND POINTS TO PONDER	337
FURTHER READINGS AND OTHER INFORMATION SOURCES	339
 CHAPTER 13 DESIGN CONCEPTS AND PRINCIPLES	 341
13.1 SOFTWARE DESIGN AND SOFTWARE ENGINEERING	341
13.2 THE DESIGN PROCESS	343
13.2.1 Design and Software Quality	343
13.2.2 The Evolution of Software Design	344
13.3 DESIGN PRINCIPLES	345
13.4 DESIGN CONCEPTS	346
13.4.1 Abstraction	347
13.4.2 Refinement	348
13.4.3 Modularity	348
13.4.4 Software Architecture	351
13.4.5 Control Hierarchy	352
13.4.6 Structural Partitioning	353