

SOFTWARE ENGINEERING

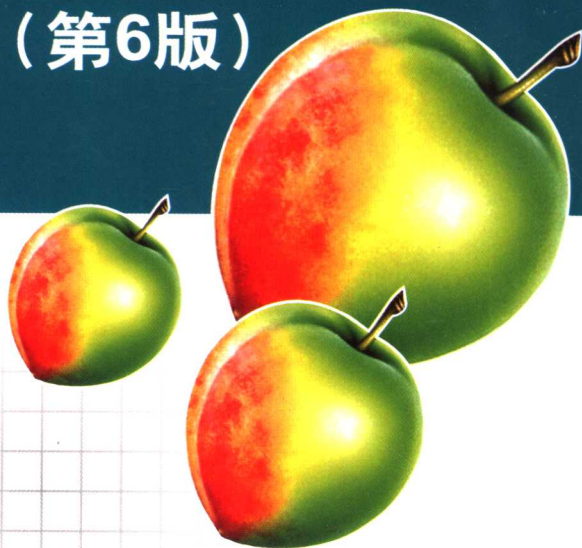
A PRACTITIONER'S APPROACH

SIXTH EDITION

软件工程

实践者之路 (第6版)

Roger S. Pressman 著



大学计算机教育国外著名教材系列（影印版）

Software Engineering

A Practitioner's Approach

Sixth Edition

软 件 工 程

实践者之路

（第 6 版）

Roger

江苏工业学院图书馆
藏书章

清华大学出版社

北 京

Roger S. Pressman

Software Engineering: A Practitioner's Approach, Sixth Edition

EISBN: 0-07-285318-2

Copyright © 2005 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Authorized English language edition jointly published by McGraw-Hill Education (Asia) Co. and Tsinghua University Press. This edition is authorized for sale only to the educational and training institutions, and within the territory of the People's Republic of China (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书英文影印版由清华大学出版社和美国麦格劳-希尔教育出版(亚洲)公司合作出版。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾地区)针对教育及培训机构之销售。未经许可之出口,视为违反著作权法,将受法律之制裁。

未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字 01-2006-2134

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有 McGraw-Hill 公司防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

软件工程:实践者之路 = Software Engineering: A Practitioner's Approach (第6版)/(美)普雷斯曼(Pressman, R. S.)著. —影印本. —北京:清华大学出版社, 2006. 6

(大学计算机教育国外著名教材系列)

ISBN 7-302-12836-7

I. 软… II. 普… III. 软件工程—高等学校—教材—英文 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2006) 第 033480 号

出版者:清华大学出版社 地址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮编:100084

社总机:010-62770175 客户服务:010-62776969

印刷者:北京市清华园胶印厂

装订者:北京国马印刷厂

发行者:新华书店总店北京发行所

开本:185×230 印张:58.25

版次:2006年6月第6版 2006年6月第1次印刷

书号:ISBN 7-302-12836-7/TP·8169

印数:1~3000

定价:79.00元

出版说明

进入 21 世纪, 世界各国的经济、科技以及综合国力的竞争将更加激烈。竞争的中心无疑是对人才的竞争。谁拥有大量高素质的人才, 谁就能在竞争中取得优势。高等教育, 作为培养高素质人才的事业, 必然受到高度重视。目前我国高等教育的教材更新较慢, 为了加快教材的更新频率, 教育部正在大力促进我国高校采用国外原版教材。

清华大学出版社从 1996 年开始, 与国外著名出版公司合作, 影印出版了“大学计算机教育丛书(影印版)”等一系列引进图书, 受到国内读者的欢迎和支持。跨入 21 世纪, 我们本着为我国高等教育教材建设服务的初衷, 在已有的基础上, 进一步扩大选题内容, 改变图书开本尺寸, 一如既往地请有关专家挑选适用于我国高校本科及研究生计算机教育的国外经典教材或著名教材, 组成本套“大学计算机教育国外著名教材系列(影印版)”, 以飨读者。深切期盼读者及时将使用本系列教材的效果和意见反馈给我们。更希望国内专家、教授积极向我们推荐国外计算机教育的优秀教材, 以利我们把“大学计算机教育国外著名教材系列(影印版)”做得更好, 更适合高校师生的需要。

清华大学出版社

前 言

随着计算机应用面向更广阔和更深入的方向发展,如今计算机软件已涉及到人们社会生活的各个领域。作为计算机系统的核心和灵魂,软件的重要性越来越被人们认识。软件是不可见的智力产品,完全不同于普通制造业的产品。专业人员一直在探讨如何获得高质量和低成本的软件产品,如何把软件工作组织得更加有效且更为合理。人类几千年前就已开始积累起来的工程知识和经验,可以被软件研制工作吸收的成分十分有限。年青的软件工程学科只能在探索中走出自己的道路。

Roger Pressman 博士的这本书对软件工程学科的发展起了重要的推动作用。本书在国际软件工程界产生了巨大而深远的影响,从而树立了它无可置疑的权威地位。一本优秀的著作,特别是一本成功的教学用书可以影响一代人,甚至几代人的业务成长。本书从 1982 年的第 1 版开始,就受到我国软件工程界的重视,成为高等学校计算机专业软件工程课的重要教学参考书。20 多年来,它的各个后继版本一直都是我国软件专业人士熟悉的读物。它在全面而系统、概括而清晰地介绍软件工程有关的概念、原则、方法和工具方面获得了广大读者的好评。此外,本书在给出传统的、对学科发展具有深刻影响的方法时,又适当地介绍了当前发展的且有着生命力的新技术。

本书第 6 版共 32 章,分为 5 个部分,分别属于 5 个专题:

- (1) 第一部分为软件过程,给出了软件过程的不同论点,考虑了所有重要的过程模型,涉及到传统的过程和(近年发展起来的)敏捷过程在指导思想上的争论。
- (2) 第二部分为软件工程实践,给出了分析、设计和测试方法,其中重点讨论了面向对象技术和 UML 建模。
- (3) 第三部分为应用 Web 工程,提供了 Web 应用系统的分析、设计和测试的全面工程方法。
- (4) 第四部分为软件项目管理,给出的是计划、管理和控制软件项目的人员关心的问题。
- (5) 第五部分是软件工程的高级课题,涉及到形式化方法、净室软件工程、基于构件的软件工程、再工程以及软件工程未来的发展趋势等问题。

本书为方便阅读,在页面安排上做了精心设计,包括:

- (1) 每章首页除给出本章要点(Quick Look)外,在左边栏中还给出了本章涉及的关键术语(KEY CONCEPTS)。
- (2) 各章在文中适当的位置,提供了带图标的简要注释,均以纵向长方框形状置于页面的左侧。例如:



指出该段内容的要点。

WebRef

指出网上资源的站点。



指出实际工作需要注意的地方，给予指导或建议。



此处所提问题的答案在右侧文中。

(3) 文中插入了说明，以文中自然段之间的通栏长方框形式出现：

- **SAFE HOME** 该框以对话形式给出软件工程人员在完成相关工作，将软件工程原则和方法应用到实际问题时的考虑。其意义为“住宅保安系统”，给读者以通俗且生动的印象。
- **INFO** 该栏内给出了所讨论内容的补充和扩展信息，使读者对相关概念获得更为广泛而深刻的理解。
- **SOFTWARE TOOL** 推荐了当前最为流行且具有代表性的商品化软件工具，指出了相应的工具开发商及其相关信息。
- **TASK SET** 简明地列举出需要完成的任务。

文中插入的简短的经典文献引语，也被置于自然段之间的通栏长方框内，其精炼而简明的语句必能给读者以有益的启发。

每章后面均附有参考文献、习题与思考题以及推荐读物与阅读信息。

本书可直接为高校计算机相关专业的教学服务，它为教师备课和学生自学提供了丰富的材料，包括书中可看到的以及间接可以找到的材料。此外，本书可供软件工程研究人员和承担软件项目的专业人员做参考之用。总之，请认真阅读这本优秀的软件工程读物，它会成为你的良师益友。

ABOUT THE AUTHOR

Roger 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 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 QAI India to develop a comprehensive Internet-based "eSchool" in software engineering.

Dr. Pressman has written many technical papers, is a regular contributor to industry periodicals, and is author of six technical books. In addition to *Software Engineering: A Practitioner's Approach*, he has 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 ACM, 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 is a serious tennis player (NTRP 4.5) and a single-digit handicap golfer. He has written two novels, *The Aymara Bridge* and *The Puppeteer*.

目 录

第1章 软件工程介绍1	
1.1 软件角色的演化.....2	
1.2 软件.....4	
1.3 软件特性的变化.....8	
1.4 遗产软件.....10	
1.4.1 遗产软件的质量.....11	
1.4.2 软件演化.....11	
1.5 软件神话.....13	
1.6 这一切是如何开始的?.....15	
1.7 小结.....16	
参考文献.....17	
习题与思考题.....17	
推荐读物与阅读信息.....18	
第一部分 软件过程	
第2章 过程综述20	
2.1 软件工程:一种层次化技术.....21	
2.2 过程框架.....22	
2.3 能力成熟度模型集成 CMMI.....27	
2.4 过程模式.....31	
2.5 过程评估.....34	
2.6 个人过程模型和团队过程模型.....36	
2.6.1 个人软件过程.....36	
2.6.2 团队软件过程.....38	
2.7 过程技术.....39	
2.8 产品与过程.....40	
2.9 总结.....41	
参考文献.....42	
习题与思考题.....43	
推荐读物与阅读信息.....43	
第3章 过程模型45	
3.1 惯例过程模型.....46	
3.2 瀑布模型.....47	
3.3 增量过程模型.....48	
3.4 演化过程模型.....51	
3.4.1 原型开发.....51	
3.4.2 螺旋模型.....54	
3.4.3 并行开发模型.....56	
3.4.4 演化模型的最终评述.....57	
3.5 专用过程模型.....59	
3.5.1 基于构件的开发.....59	
3.5.2 形式化方法模型.....60	
3.5.3 面向方面的软件开发.....61	
3.6 统一过程.....62	
3.6.1 简史.....63	
3.6.2 统一过程的阶段.....64	
3.6.3 统一过程工作产品.....66	
3.7 小结.....67	
参考文献.....68	
习题与思考题.....69	
推荐读物与阅读信息.....70	
第4章 敏捷视角下的过程71	
4.1 敏捷是什么?.....73	
4.2 敏捷过程是什么?.....74	
4.2.1 敏捷开发的动机.....75	
4.2.2 人的因素.....76	
4.3 敏捷过程模型.....77	
4.3.1 极限编程(eXtreme Programming, XP).....78	
4.3.2 自适应软件开发(Adaptive Software Development, ASD).....82	
4.3.3 动态系统开发方法(Dynamic System Development Method, DSDM).....84	
4.3.4 Scrum.....85	
4.3.5 Crystal.....87	
4.3.6 特征驱动开发(Feature Driven Development, FDD).....88	
4.3.7 敏捷建模(Agile Modeling, AM).....89	
4.4 小结.....91	
参考文献.....92	

习题与思考题	93	7.2.3 精化	145
推荐读物与阅读信息	94	7.2.4 协商	146
第二部分 软件工程实践		7.2.5 规格说明	147
第5章 软件工程实践	96	7.2.6 确认	147
5.1 软件工程实践	97	7.2.7 需求管理	148
5.1.1 实践的精髓	97	7.3 启动需求工程过程	149
5.1.2 核心原则	99	7.3.1 确认共同利益者	150
5.2 沟通实践	101	7.3.2 识别多种观点	150
5.3 策划实践	104	7.3.3 协同合作	151
5.4 建模实践	107	7.3.4 首次提问	151
5.4.1 分析建模原则	108	7.4 导出需求	152
5.4.2 设计建模原则	109	7.4.1 协同收集需求	153
5.5 构造实践	112	7.4.2 质量功能部署	156
5.5.1 编码原则和概念	113	7.4.3 用户场景	157
5.5.2 测试原则	114	7.4.4 导出工作产品	158
5.6 部署	116	7.5 开发用例	159
5.7 小结	118	7.6 构建分析模型	164
参考文献	119	7.6.1 分析模型的元素	164
习题与思考题	120	7.6.2 分析模式	168
推荐读物与阅读信息	120	7.7 协商需求	169
第6章 系统工程	122	7.8 确认需求	171
6.1 基于计算机的系统	123	7.9 小结	172
6.2 系统工程层次结构	125	参考文献	172
6.2.1 系统建模	126	习题与思考题	173
6.2.2 系统仿真	128	推荐读物与阅读信息	174
6.3 业务过程工程概述	129	第8章 构建分析模型	175
6.4 产品工程概述	130	8.1 需求分析	176
6.5 系统建模	132	8.1.1 整体目标和原理	177
6.5.1 Hatley-Pirbhai 建模	133	8.1.2 分析的经验规则	178
6.5.2 UML 系统建模	135	8.1.3 域分析	178
6.6 小结	139	8.2 分析建模的方法	179
参考文献	140	8.3 数据建模概念	181
习题与思考题	140	8.3.1 数据对象	181
推荐读物与阅读信息	141	8.3.2 数据属性	182
第7章 需求工程	142	8.3.3 关系	182
7.1 连接设计和构造的桥梁	143	8.3.4 基数和形态	183
7.2 需求工程任务	144	8.4 面向对象的分析	185
7.2.1 起始	144	8.5 基于场景建模	186
7.2.2 导出	145	8.5.1 编写用例	186
		8.5.2 开发活动图	191

8.5.3 泳道图	192	9.5 基于模式的软件设计	248
8.6 面向流的建模	194	9.5.1 描述设计模式	248
8.6.1 创建数据流模型	194	9.5.2 在设计中使用模式	249
8.6.2 创建控制流模型	197	9.5.3 框架	249
8.6.3 控制规格说明	198	9.6 小结	250
8.6.4 处理规格说明	200	参考文献	251
8.7 基于类的建模	201	习题与思考题	251
8.7.1 识别分析类	201	推荐读物与阅读信息	252
8.7.2 描述属性	204	第 10 章 创建体系结构设计	254
8.7.3 定义操作	205	10.1 软件体系结构	255
8.7.4 类—责任—协作者 (CRC) 建模	208	10.1.1 什么是体系结构	255
8.7.5 关联和依赖	214	10.1.2 为什么体系结构如此重要	256
8.7.6 分析包	215	10.2 数据设计	257
8.8 生成行为模型	216	10.2.1 体系结构级的数据设计	257
8.8.1 识别用例事件	217	10.2.2 构件级的数据设计	258
8.8.2 状态表现	218	10.3 体系结构风格和模式	259
8.9 小结	221	10.3.1 体系结构风格的简单分类	260
参考文献	222	10.3.2 体系结构模式	264
习题与思考题	223	10.3.3 组织和精化	265
推荐读物与阅读信息	224	10.4 体系结构设计	266
第 9 章 设计工程	226	10.4.1 系统的环境表示	266
9.1 软件工程中的设计	227	10.4.2 定义原始模型	268
9.2 设计过程和设计质量	229	10.4.3 将体系结构精化为构件	269
9.3 设计概念	233	10.4.4 描述系统实例	271
9.3.1 抽象	233	10.5 评估可选的体系结构设计	272
9.3.2 架构	233	10.5.1 体系结构权衡分析方法	272
9.3.3 模式	234	10.5.2 体系结构复杂性	274
9.3.4 模块化	235	10.5.3 体系结构描述语言	274
9.3.5 信息隐藏	236	10.6 映射数据流到软件体系结构	275
9.3.6 功能独立	236	10.6.1 变换流	276
9.3.7 精化	237	10.6.2 事务流	276
9.3.8 重构	238	10.6.3 变换映射	277
9.3.9 设计类	239	10.6.4 事务映射	284
9.4 设计模型	242	10.6.5 精化体系结构设计	288
9.4.1 数据设计元素	243	10.7 小结	288
9.4.2 架构设计元素	243	参考文献	289
9.4.3 接口设计元素	244	习题与思考题	290
9.4.4 构件级设计元素	246	推荐读物与阅读信息	291
9.4.5 部署级设计元素	247	第 11 章 构件级设计建模	292
		11.1 什么是构件	293

11.1.1 面向对象的观点	294	推荐读物与阅读信息	353
11.1.2 传统观点	295	第 13 章 软件测试策略	354
11.1.3 过程相关的观点	298	13.1 软件测试的策略性方法	355
11.2 设计基于类的构件	298	13.1.1 验证与确认	356
11.2.1 基本设计原则	299	13.1.2 软件测试的组织	356
11.2.2 构件级设计指导方针	302	13.1.3 常规软件体系结构的 测试策略	358
11.2.3 内聚性	303	13.1.4 面向对象软件体系结构的测 试策略	359
11.2.4 耦合性	305	13.1.5 测试完成的标准	360
11.3 实施构件级设计	307	13.2 策略问题	361
11.4 对象约束语言	313	13.3 传统软件的测试策略	362
11.5 设计传统构件	315	13.3.1 单元测试	362
11.5.1 图形设计表示	316	13.3.2 集成测试	365
11.5.2 表格设计表示	317	13.4 面向对象软件的测试策略	372
11.5.3 程序设计语言	318	13.4.1 面向对象语境中的单元 测试	372
11.5.4 设计表示方法的比较	320	13.4.2 面向对象语境中的集成 测试	373
11.6 小结	321	13.5 确认测试	374
参考文献	322	13.5.1 确认测试准则	374
习题与思考题	322	13.5.2 配置评审	374
推荐读物与阅读信息	323	13.5.3 α 测试与 β 测试	374
第 12 章 完成用户界面设计	324	13.6 系统测试	376
12.1 黄金规则	325	13.6.1 恢复测试	377
12.1.1 置用户于控制之下	325	13.6.2 安全测试	377
12.1.2 减轻用户的记忆负担	327	13.6.3 压力测试	377
12.1.3 保持界面一致	328	13.6.4 性能测试	378
12.2 用户界面的分析与设计	329	13.7 调试艺术	379
12.2.1 界面分析和设计模型	330	13.7.1 调试过程	379
12.2.2 界面分析和设计过程	331	13.7.2 心理因素	381
12.3 界面分析	333	13.7.3 调试策略	382
12.3.1 用户分析	333	13.7.4 错误改正	384
12.3.2 任务分析和建模	335	13.8 小结	384
12.3.3 显示内容分析	340	参考文献	385
12.3.4 工作环境分析	341	习题与思考题	385
12.4 界面设计步骤	341	推荐读物与阅读信息	386
12.4.1 应用界面设计步骤	342	第 14 章 测试战术	388
12.4.2 用户界面设计模式	343	14.1 软件测试基础	389
12.4.3 设计问题	345	14.2 黑盒测试与白盒测试	391
12.5 设计评估	349		
12.6 小结	351		
参考文献	351		
习题与思考题	352		

14.3	白盒测试	392	习题与思考题	427
14.4	基本路径测试	393	推荐读物与阅读信息	428
14.4.1	流图表示	393		
14.4.2	独立程序路径	394		
14.4.3	测试用例的设计	396		
14.4.4	图矩阵	399		
14.5	控制结构测试	400	第 15 章 产品度量	429
14.5.1	条件测试	400	15.1 软件质量	430
14.5.2	数据流测试	400	15.1.1 McCall 的质量因素	431
14.5.3	循环测试	401	15.1.2 ISO 9126 质量因素	432
14.6	黑盒测试	402	15.1.3 向量化视图的变迁	433
14.6.1	基于图的测试方法	403	15.2 产品度量框架	434
14.6.2	等价划分	405	15.2.1 测度、度量和指标	434
14.6.3	边界值分析	406	15.2.2 产品度量的挑战	434
14.6.4	正交数组测试	407	15.2.3 测量原则	435
14.7	面向对象测试方法	410	15.2.4 面向目标的软件测量	436
14.7.1	OO 概念的测试用例设计的 含义	410	15.2.5 有效软件度量的属性	437
14.7.2	传统测试用例设计方法的可 应用性	411	15.2.6 产品度量风景	438
14.7.3	基于故障的测试	411	15.3 分析模型的度量	440
14.7.4	测试用例与类层次	412	15.3.1 基于功能的度量	440
14.7.5	基于场景的测试	412	15.3.2 规格说明质量的度量	444
14.7.6	表层结构和深层结构的 测试	414	15.4 设计模型的度量	445
14.8	类层次可应用的测试方法	415	15.4.1 体系结构设计度量	445
14.8.1	面向对象的随机测试	415	15.4.2 面向对象设计的度量	448
14.8.2	类层次的分割测试	416	15.4.3 面向类的度量——CK 度 量套	449
14.9	类间测试用例设计	417	15.4.4 面向类的度量——MOOD 度量套	452
14.9.1	多个类测试	417	15.4.5 Lorenz 与 Kidd 提出的面向 对象度量	453
14.9.2	从行为模型中生成的测试	418	15.4.6 构件级设计度量	454
14.10	针对特定环境、体系结构和应用 的测试	420	15.4.7 面向操作的度量	456
14.10.1	用户界面测试	420	15.4.8 用户界面设计度量	457
14.10.2	客户/服务器体系结构	420	15.5 源代码的度量	458
14.10.3	测试文档和帮助设施	421	15.6 测试的度量	459
14.10.4	实时系统的测试	422	15.6.1 应用于测试的 Halstead 度量	459
14.11	测试模式	424	15.6.2 面向对象测试的度量	459
14.12	小结	425	15.7 维护的度量	460
参考文献		426	15.8 小结	461
			参考文献	462
			习题与思考题	464
			推荐读物与阅读信息	465

第三部分 应用 Web 工程

第 16 章 Web 工程	468
16.1 基于 Web 的系统及应用的特点	469
16.2 WebApp 工程的层次	472
16.2.1 过程	472
16.2.2 方法	473
16.2.3 工具与技术	474
16.3 Web 工程过程	474
16.3.1 定义框架	475
16.3.2 精化框架	477
16.4 Web 工程的最佳实践	478
16.5 小结	479
参考文献	480
习题与思考题	480
推荐读物与阅读信息	481
第 17 章 开始一个 WebApp 项目	482
17.1 表达基于 Web 的系统	483
17.1.1 表达问题	483
17.1.2 WebApp 的需求收集	485
17.1.3 分析模型的过渡	489
17.2 策划 Web 工程项目	490
17.3 Web 工程团队	491
17.3.1 人员	491
17.3.2 组建团队	492
17.4 Web 工程的项目管理问题	493
17.4.1 WebApp 策划——外包	494
17.4.2 WebApp 策划——内部 Web 工程	498
17.5 Web 工程与 WebApp 的度量	500
17.5.1 Web 工程工作的度量	501
17.5.2 评估商业价值的度量	502
17.6 WebApp 项目的“最坏实践”	502
17.7 小结	504
参考文献	504
习题与思考题	505
推荐读物与阅读信息	506
第 18 章 WebApp 分析	507
18.1 WebApp 的需求分析	508
18.1.1 用户层次	509

18.1.2 开发用例	510
18.1.3 精化用例模型	512
18.2 WebApp 的分析模型	513
18.3 内容模型	513
18.3.1 定义内容对象	514
18.3.2 内容关系与层次	514
18.3.3 WebApp 的分析类	515
18.4 交互模型	516
18.5 功能模型	519
18.6 配置模型	521
18.7 关系导航分析	521
18.7.1 关系分析——关键问题	522
18.7.2 导航分析	523
18.8 小结	524
参考文献	525
习题与思考题	525
推荐读物与阅读信息	526
第 19 章 设计 WebApp	527
19.1 Web 工程的设计问题	528
19.1.1 设计与 WebApp 质量	528
19.1.2 设计目标	531
19.2 WebE 设计金字塔	532
19.3 WebApp 界面设计	533
19.3.1 界面设计原则与指导方针	534
19.3.2 界面控制机制	539
19.3.3 界面设计工作流	539
19.4 美学设计	541
19.4.1 布局问题	542
19.4.2 美术设计问题	542
19.5 内容设计	543
19.5.1 内容对象	543
19.5.2 内容设计问题	544
19.6 体系结构设计	545
19.6.1 内容体系结构	545
19.6.2 WebApp 体系结构	547
19.7 导航设计	549
19.7.1 导航语义	549
19.7.2 导航语法	551
19.8 构件级设计	552
19.9 超媒体设计模式	552

19.10	面向对象的超媒体设计方法 (OOHDM)	554
19.10.1	OOHDM 的概念设计	554
19.10.2	OOHDM 的导航设计	555
19.10.3	抽象界面设计与实现	556
19.11	WebApp 的设计度量	556
19.12	小结	557
	参考文献	558
	习题与思考题	560
	推荐读物与阅读信息	561
第 20 章	WebApp 测试	562
20.1	WebApp 的测试概念	563
20.1.1	质量维度	563
20.1.2	WebApp 环境中的错误	564
20.1.3	测试策略	565
20.1.4	测试计划	566
20.2	测试过程——概述	566
20.3	内容测试	569
20.3.1	内容测试的目标	569
20.3.2	数据库测试	571
20.4	用户界面测试	573
20.4.1	界面测试策略	573
20.4.2	测试界面机制	574
20.4.3	测试界面语义	576
20.4.4	可用性测试	576
20.4.5	兼容性测试	578
20.5	构件级测试	579
20.6	导航测试	581
20.6.1	测试导航语法	581
20.6.2	测试导航语义	582
20.7	配置测试	583
20.7.1	服务器端问题	584
20.7.2	客户端问题	584
20.8	安全性测试	585
20.9	性能测试	587
20.9.1	性能测试的目标	587
20.9.2	负载测试	588
20.9.3	压力测试	588
20.10	小结	590
	参考文献	591

习题与思考题	592
推荐读物与阅读信息	593

第四部分 管理软件项目

第 21 章	项目管理	596
21.1	管理涉及的范围	597
21.1.1	人员	597
21.1.2	产品	598
21.1.3	过程	598
21.1.4	项目	598
21.2	人员	599
21.2.1	共同利益者	599
21.2.2	团队负责人	600
21.2.3	软件团队	601
21.2.4	敏捷团队	604
21.2.5	协调和通信问题	605
21.3	产品	606
21.3.1	软件范围	606
21.3.2	问题分解	607
21.4	过程	608
21.4.1	合并产品和过程	608
21.4.2	过程分解	609
21.5	项目	610
21.6	W ⁵ HH 原则	612
21.7	关键实践	612
21.8	小结	613
	参考文献	614
	习题与思考题	614
	推荐读物与阅读信息	615
第 22 章	过程和项目度量	617
22.1	过程和项目领域中的度量	618
22.1.1	过程度量和软件过程改进	618
22.1.2	项目度量	621
22.2	软件测量	622
22.2.1	面向规模的度量	623
22.2.2	面向功能的度量	624
22.2.3	调和代码行和功能点的 度量方法	624
22.2.4	面向对象的度量	626
22.2.5	面向用例的度量	627

22.2.6 Web 工程项目度量.....	627	23.9.2 Web 工程项目的估算.....	665
22.3 软件质量度量.....	629	23.10 自行开发或购买的决策.....	666
22.3.1 测量质量.....	630	23.10.1 创建决策树.....	667
22.3.2 缺陷排除效率.....	631	23.10.2 外包.....	668
22.4 在软件工程过程中集成度量.....	632	23.11 小结.....	669
22.4.1 支持软件度量的论点.....	633	参考文献.....	670
22.4.2 建立基线.....	633	习题与思考题.....	671
22.4.3 度量收集、计算和评估.....	634	推荐读物与阅读信息.....	671
22.5 小型组织的度量.....	634	第 24 章 项目进度安排.....	673
22.6 制定软件度量大纲.....	636	24.1 基本概念.....	674
22.7 小结.....	638	24.2 项目进度安排.....	676
参考文献.....	638	24.2.1 基本原则.....	677
习题与思考题.....	639	24.2.2 人员与工作量之间的关系.....	678
推荐读物与阅读信息.....	640	24.2.3 工作量分配.....	680
第 23 章 估算.....	642	24.3 为软件项目定义任务集合.....	681
23.1 对估算的观察.....	643	24.3.1 一个任务集合的例子.....	682
23.2 项目计划过程.....	644	24.3.2 主要任务的求精.....	682
23.3 软件范围和可行性.....	645	24.4 定义任务网络.....	683
23.4 资源.....	645	24.5 进度安排.....	684
23.4.1 人力资源.....	646	24.5.1 时序图.....	685
23.4.2 可复用软件资源.....	646	24.5.2 跟踪进度.....	686
23.4.3 环境资源.....	647	24.5.3 跟踪 OO 项目的进展.....	688
23.5 软件项目估算.....	648	24.6 获得值分析.....	690
23.6 分解技术.....	649	24.7 小结.....	691
23.6.1 软件规模估算.....	649	参考文献.....	691
23.6.2 基于问题的估算.....	650	习题与思考题.....	692
23.6.3 基于 LOC 估算的实例.....	651	推荐读物与阅读信息.....	693
23.6.4 基于 FP 估算的实例.....	653	第 25 章 风险管理.....	694
23.6.5 基于过程的估算.....	654	25.1 被动和主动的风险策略.....	695
23.6.6 基于过程估算的实例.....	655	25.2 软件风险.....	696
23.6.7 基于用例的估算.....	656	25.3 风险识别.....	697
23.6.8 基于用例的估算实例.....	657	25.3.1 评估整体项目风险.....	698
23.6.9 调和不同的估算方法.....	658	25.3.2 风险因素和驱动因子.....	699
23.7 经验估算模型.....	659	25.4 风险预测.....	700
23.7.1 估算模型的结构.....	660	25.4.1 建立风险表.....	701
23.7.2 COCOMO II 模型.....	660	25.4.2 评估风险影响.....	703
23.7.3 软件方程式.....	662	25.5 风险求精.....	705
23.8 面向对象项目的估算.....	663	25.6 风险缓解、监测和管理.....	705
23.9 特殊的估算技术.....	664	25.7 RMMM 计划.....	708
23.9.1 敏捷开发的估算.....	664		

25.8 小结	709	27.1.4 软件配置项	743
参考文献	710	27.2 SCM 中心存储库	745
习题与思考题	710	27.2.1 中心存储库的作用	745
推荐读物与阅读信息	711	27.2.2 一般特征和内容	746
第 26 章 质量管理	712	27.2.3 SCM 特征	747
26.1 质量概念	713	27.3 SCM 过程	748
26.1.1 质量	714	27.3.1 软件配置中对象的标识	749
26.1.2 质量控制	714	27.3.2 版本控制	750
26.1.3 质量保证	715	27.3.3 变更控制	752
26.1.4 质量成本	715	27.3.4 配置审核	755
26.2 软件质量保证	716	27.3.5 状态报告	756
26.2.1 背景	717	27.4 Web 工程的配置管理	756
26.2.2 SQA 活动	717	27.4.1 WebApp 的配置管理问题	757
26.3 软件评审	719	27.4.2 WebApp 的配置对象	758
26.3.1 软件缺陷对成本的影响	720	27.4.3 内容管理	758
26.3.2 缺陷放大和消除	720	27.4.4 变更管理	761
26.4 正式的技术评审	722	27.4.5 版本控制	763
26.4.1 评审会议	722	27.4.6 审核和报告	764
26.4.2 评审报告和记录保存	723	27.5 小结	765
26.4.3 评审指导原则	724	参考文献	766
26.4.4 样本驱动评审	725	习题与思考题	767
26.5 SQA 的形式化方法	727	推荐读物与阅读信息	768
26.6 基于统计的软件质量保证	727	第五部分 软件工程高级课题	
26.6.1 一个普通的例子	728	第 28 章 形式化方法	770
26.6.2 软件工程中的六西格玛	729	28.1 基本概念	771
26.7 软件可靠性	730	28.1.1 非形式化方法的缺陷	772
26.7.1 可靠性和可用性的测量	731	28.1.2 软件开发中的数学	773
26.7.2 软件安全	731	28.1.3 形式化方法概念	773
26.8 ISO 9000 质量标准	733	28.2 数学预备知识	776
26.9 SQA 计划	734	28.2.1 集合与构造性规格说明	776
26.10 小结	735	28.2.2 集合运算符	778
参考文献	736	28.2.3 逻辑运算符	780
习题与思考题	737	28.2.4 序列	780
推荐读物与阅读信息	737	28.3 应用数学表示法描述形式化 规格说明	781
第 27 章 变更管理	739	28.4 形式规格说明语言	783
27.1 软件配置管理	740	28.5 对象约束语言 (OCL)	784
27.1.1 SCM 场景	741	28.5.1 OCL 语法及语义概述	784
27.1.2 配置管理系统元素	742	28.5.2 使用 OCL 举例	786
27.1.3 基线	743		

28.6 Z 规格说明语言	788	30.5.1 描述可复用构件	827
28.6.1 Z 语法及语义概述	788	30.5.2 复用环境	829
28.6.2 使用 Z 举例	788	30.6 CBSE 经济学	830
28.7 形式化方法的十条戒律	791	30.6.1 对质量、生产率及成本的 影响	830
28.8 形式化方法——未来之路	792	30.6.2 运用结构点的成本分析	831
28.9 小结	793	30.7 小结	832
参考文献	793	参考文献	833
习题与思考题	794	习题与思考题	834
推荐读物与阅读信息	795	推荐读物与阅读信息	835
第 29 章 净室软件工程	796	第 31 章 再工程	837
29.1 净室方法	797	31.1 业务过程再工程	838
29.1.1 净室策略	798	31.1.1 业务过程	839
29.1.2 净室方法的特异之处	800	31.1.2 BPR 模型	839
29.2 功能规格说明	801	31.2 软件再工程	841
29.2.1 黑盒规格说明	802	31.2.1 软件维护	841
29.2.2 状态盒规格说明	803	31.2.2 软件再工程过程模型	842
29.2.3 清晰盒规格说明	803	31.3 逆向工程	846
29.3 净室设计	804	31.3.1 理解数据的逆向工程	848
29.3.1 设计求精与验证	804	31.3.2 理解处理的逆向工程	848
29.3.2 设计验证的优点	808	31.3.3 用户界面的逆向工程	849
29.4 净室测试	809	31.4 重构	850
29.4.1 统计使用测试	810	31.4.1 代码重构	850
29.4.2 认证	811	31.4.2 数据重构	851
29.5 小结	812	31.5 正向工程	852
参考文献	812	31.5.1 客户 / 服务器体系结构(C/S) 的正向工程	853
习题与思考题	813	31.5.2 面向对象体系结构的正向 工程	854
推荐读物与阅读信息	814	31.5.3 用户界面的正向工程	855
第 30 章 基于构件的开发	815	31.6 再工程经济学	855
30.1 基于构件系统的工程	816	31.7 小结	856
30.2 CBSE 过程	818	参考文献	857
30.3 领域工程	819	习题与思考题	858
30.3.1 领域分析过程	819	推荐读物与阅读信息	859
30.3.2 特征化函数	820	第 32 章 未来之路	860
30.3.3 结构化建模与结构点	821	32.1 软件的重要性——再论	861
30.4 基于构件的开发	822	32.2 变化的范围	861
30.4.1 构件合格性检验、适应性 修改与组装	822	32.3 人及其构造系统的方式	863
30.4.2 构件工程	825		
30.4.3 复用的分析与设计	826		
30.5 构件分类与检索	827		