


计 算 机 科 学 丛 书

软件工程 实践者的研究方法

Software
Engineering
A Practitioner's Approach
Fourth Edition

(美) Roger S. Pressman 著
黄柏素 梅宏 译

 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

(美) Roger S. Pressman 著

黄柏素 梅宏 译



机械工业出版社
China Machine Press

本书以丰富的信息资源、广泛的角度深刻地论述了软件工程的方方面面，内容包括软件工程的相关语境、管理软件项目的方法、各种软件工程的方法论、面向对象的方法、客户/服务器软件工程、CASE等。本书内容新颖、见解独到，适合大学计算机专业高年级学生、研究生学习，也适合软件工程开发者、组织者、管理者参考。

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

Authorized translation from the English language edition published by McGraw-Hill.

Copyright © 1997 by McGraw-Hill.

All rights reserved.

本书中文简体字版由机械工业出版社出版，未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

版权所有，翻印必究。

本书版权登记号：图字01-1999-1835

图书在版编目(CIP)数据

软件工程——实践者的研究方法 / (美) 普瑞斯曼(Pressman, R. S.) 著；黄柏素、梅宏译. —北京：机械工业出版社，1999. 10

(计算机科学丛书)

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

ISBN 7-111-07282-0

I. 软… II. ①普… ②黄… ③梅… III. 软件工程 IV. TP311.5

中国版本图书馆CIP数据核字(1999)第27039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：吴怡 章小莉

北京市密云县印刷厂印刷·新华书店北京发行所发行

1999年10月第1版第1次印刷

787mm × 1092mm 1/16 · 37.25印张

印数：0 001-8 000册

定价：48.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

作者简介

Roger S. Pressman是软件工程领域国际知名的咨询专家和作者。他以优异成绩从Connecticut大学获得学士学位，从Bridgeport大学获得硕士学位，从Connecticut大学获得工学博士学位。已有超过25年的产业经验。主要从事工程产品软件和系统软件的开发技术工作和管理工作。

作为产业的实践者和管理者，Pressman博士主要从事的是航空航天应用中高级工程和制造的CAD/CAM系统的开发，他也从事科学及系统程序设计方面的工作。

除了他的产业经验之外，Pressman博士还是Bridgeport大学计算机工程系的兼职副教授和该大学的计算机辅助设计和制造中心的主任。

Pressman博士是R.S. Pressman & Associates, Inc公司的总裁，这是一家专门从事软件工程方法和培训的咨询公司。他是公司主要的咨询专家，专门负责帮助其他公司建立有效的软件工程方法。他开发了RSP&A软件工程评估方法，该方法采用独特的数量和质量分析混合的方式，帮助客户评估他们软件工程实践的当前状况。

除了给500多个客户提供咨询服务外，R.S. Pressman & Associates, Inc公司还提供大量的软件工程培训及过程改善服务。公司开发了一个艺术式的录像课程“Essential Software Engineering”，它全面地讲述了产业界关于这一主题的内容。另一个产品“Process Advisor”是指导企业软件工程改进的自测系统。

Pressman博士还在产业期刊上发表了许多技术论文，是企业期刊的特约撰稿人并出版了6本书。除了本书外，还有：“Making Software Engineering Happen(Prentice Hall出版公司出版)”，这是第一本涉及到改善软件工程实施过程中的主要管理问题的书籍；“Software Shock(Dorset House出版公司出版)”，该书叙述了软件及其对商业和社会的影响；“A Manager's Guide to Software Engineering(McGraw-Hill出版公司出版)”，该书使用独特的Q&A方式表示了创立和理解技术的管理指南。Pressman博士是杂志American Programmer(美国程序员)和IEEE Software(IEEE软件)的编委，是IEEE Software的Manager(管理员)专栏的编辑。他还是ACM、IEEE、Tau Beta Pi、Phi Kappa Phi、Eta Kappa Nu和Pi Tau Sigma的会员。

译者序

20世纪末发生在我们这个星球上的最大变化之一无疑是席卷全球的信息技术(IT)革命,人们将这场革命视为21世纪——知识经济时代的前奏曲。在这场IT革命中,软件无疑扮演了极其重要的角色。软件产业作为一个独立形态的产业,正在全球经济中占据越来越举足轻重的地位。而软件工程正是软件产业健康发展的关键技术之一。

从1968年软件工程概念的正式提出到现在,软件工程已有逾30年的发展,出现了大量的研究成果,也进行了大量的技术实践。正是由于学术界和产业界的共同努力,软件工程正在逐步发展为一门成熟的专业学科,以解决软件生产的质量和效率问题为宗旨,在软件产业的发展中起到了重要的技术保障和促进作用。

本书是一本系统而全面地介绍软件工程理论、技术和实践的专著,是北美学术界和产业界的畅销书之一。本书作者Roger S.Pressman是软件工程领域国际知名的咨询专家和作者,著有多本学术专著,本书已是其第四版。本书第二版曾在国内翻译出版,并被很多学校选为软件工程教材,在我国软件工程研究、教学和实践起到了很好的借鉴和参考作用。而第四版并不仅仅是简单的修订,而是被完全重构以适应软件工程领域快速的增长并着重于新的、重要的软件工程方法。从早期版本保留的章节被全面地修订,并加入了12章新内容,以提供对当代趋势和技术的完整讨论。书中还加入了很多新例子、思考题、推荐阅读文献及其它参考信息源。本书的翻译出版旨在向国内软件工程领域的研究、教学、管理和技术人员提供一个全面的参考文献、教材或实践指南。

本书由黄柏素、梅宏组织翻译,其中梅宏负责第二部分7、8、9章和第三部分的翻译工作,黄柏素翻译了其余部分并负责全书的统稿工作。同时译者希望向参与了部分章节翻译工作的李克勤、张路、袁望洪、常继传、郭立峰、谢涛、郭耀、马黎等,以及参与了插图绘制和参考文献录入工作的徐松青、沈璞、刘洋、孟祥文等表示诚挚的感谢。

由于译者自身的知识局限及时间的仓促,译稿中难免存在错误和遗漏。谨向读者及原书作者致以歉意,并欢迎指正。

黄柏素、梅宏
1999年8月

译者简介



黄柏素(女), 博士, 北京大学计算机科学技术系副教授。1993年于西北工业大学获得博士学位。同年进入北京大学计算机科学技术系博士后流动站。1995年出站后留校工作。主要研究方向为软件工程、软件开发环境及工具、面向对象技术、用户界面管理系统等。承担了软件工程课程教学工作。目前已发表学术论文20余篇,并获得多项国家及部委科技成果奖和个人奖。



梅宏, 博士, 北京大学计算机科学技术系教授。1992年于上海交通大学获工学博士学位, 1994年从北京大学计算机科学技术系博士后出站。研究、教学工作主要涉及软件工程及软件开发环境、软件复用及软件构件技术、(分布)对象技术、软件工业化生产技术及支持系统、新型程序设计语言等。已在国内外学术刊物及国际、全国学术会议上发表学术论文60余篇。并多次获得国家及部委级科技成果奖,以及其他个人荣誉奖。

前 言

软件工程已进入到目前的第四代，它已具有许多优势，虽然它仍存在同时代人曾经历的某些弱点，但其早年的天真和热情已被更合理的经历多年培育的期望（以及甚至善意的嘲讽）所替代，软件工程正带着许多成就步入中年，然而还有大量工作需要去做，今天，它已被公认为一个重要学科，值得认真地研究、细心地学习和热烈地争论。在整个产业界，“软件工程师”已经替代“程序员”成为更受欢迎的工作头衔。产业应用软件中已广泛而成功地采用了软件过程模型、软件工程方法以及软件工具。管理者和实践者均认识到，需要一个更严谨的软件方法来支持手头的工作。

但是，在本书的早期版本中很多讨论的问题仍然存在，很多个人和公司仍然在随意地开发软件，很多专业人员和学生不知道现代方法，最终，我们生产的软件仍然存在大量质量问题。此外，关于软件工程方法的真实性质的争论仍在继续。然而，今天软件工程已成为研究的热点，人们对它的态度已有很大变化，它的发展也很明显，但是，要使软件工程最终发展成为一个完全成熟的学科还需做大量工作。

本书的第4版试图成为正逐步走向成熟的软件工程学科的一个指南。和前面三版一样，第4版的主要读者群仍然是学生和实践者，而且在写作风格上我们力图仍然保持前面各版的格式和风格。本书的基本目标仍然是：作为产业界专业人员的指南以及作为高年级大学生和一年级研究生的软件工程的全面导论。

我们在第4版中并不仅仅简单地修订了原稿，为适应本领域快速的增长我们完全重新组织了书中的内容，并着重讨论了新的重要的软件工程方法，还全面地修订了从早期版本保留的章节，加入了12章新内容，以提供对当代趋势和技术的完整讨论。加入了很多新例子、思考题，每一章中还增补了推荐阅读文献及其他信息搜索地址，包括数百个新的出版站点以及超过160个WWW信息站地。

第4版由5个部分共30章构成。这样做的目的是按专题安排内容，并使那些没有时间在一个学期内完成书中内容教学任务的老师，可以按需取用。第一部分：产品和过程，简介软件工程的相关语境，引出书中主要内容，并着重介绍了以后章节用到的概念；第二部分：管理软件项目，讨论那些与计划、管理和控制软件开发项目的人员相关的话题；第三部分：传统软件工程方法，讨论那些被视为传统软件工程不同“学派”的分析、设计和测试方法；第四部分：面向对象软件工程，讨论跨越整个软件工程过程的面向对象方法，包括分析、设计和测试方法；第五部分：软件工程高级课题，分章专门讨论形式化方法、净室软件工程、复用、重建工程、客户/服务器软件工程和CASE。

第4版比以前版本更多地强调了度量和测度方面的相关技术。有三章和软件度量相关，分别是：软件过程和项目的度量、软件的技术度量、面向对象系统的技术度量。

本书的组织使得老师可以根据时间和学生需要安排授课话题。一个学期可选择一个或多个部分。例如，“设计课程”可能只需要第三或第四部分，“方法课程”可能只需第三、第四和第五部分的部分章节，“管理课程”可能只需要第一和第二部分。按这种方式组织本书第4版，目的是给老师提供灵活的教学选择。

第4版的写作工作已成为我生活中持续最长的技术计划。即使当写作停止时，从各种技术文献中提炼、组织信息的工作也一直在进行，为此，我要感谢许多书籍、论文和文章的作者，以及新一代的电子媒体（新闻组和WWW）的贡献者们，他们在过去的15年中给我提供了大量的信息资源、思想和评注，很多信息资源已在每章的参考文献中列出，他们在这个快速发展的

领域中的贡献是值得称道的。我还要感谢第4版的审阅者：Wayne State University 的Frank H. Westervelt、The University of Connecticut的Steven A. Demurjian、California State Polytechnic University的Chung Lee、University of Colorado的Alan Davis、QSM Associates的Michael C. Mah、University of California-Irvine的Richard N. Taylor、Virginia Tech.的Osman Balci、Auburn University的James H. Cross、Portland State University的Warren Harrison、Northeastern University的Mieczyslaw M. Kokar，他们的评注和批评是无价的。

本书第4版内容的成型有赖于许多曾经使用过本书以前版本的产业界专业人员、大学教授和学生，他们花了很多时间和我通信交流他们的建议、批评和思想，我要感谢他们中的每一位。此外，我也要向我们的在北美和欧洲的许多产业客户表示感谢，他们教我的比我教他们的要多。

Roger S. Pressman

目 录

作者简介	
译者简介	
前言	

第一部分 产品和过程

第1章 产品	1
1.1 软件的发展	2
1.1.1 产业的观点	4
1.1.2 老化的软件工厂	4
1.1.3 软件的竞争	5
1.2 软件	5
1.2.1 软件特征	6
1.2.2 软件构件	8
1.2.3 软件应用	9
1.3 软件：地平线上的危机	10
1.4 软件神话	10
1.5 小结	12
参考文献	12
思考题	13
推荐阅读文献及其他信息源	13
第2章 过程	15
2.1 软件工程——一种层次化技术	15
2.1.1 过程、方法和工具	15
2.1.2 软件工程的一般视图	16
2.2 软件过程	17
2.3 软件过程模型	19
2.4 线性顺序模型	20
2.5 原型模型	22
2.6 RAD模型	23
2.7 演化软件过程模型	25
2.7.1 增量模型	25
2.7.2 螺旋模型	27
2.7.3 构件组装模型	28
2.7.4 并发开发模型	29
2.8 形式化方法模型	31
2.9 第四代技术	31
2.10 过程技术	32
2.11 产品和过程	33

2.12 小结	33
参考文献	34
思考题	35
推荐阅读文献及其他信息源	35

第二部分 软件项目的管理

第3章 项目管理的概念	39
3.1 管理的范围	39
3.1.1 人员	40
3.1.2 问题	40
3.1.3 过程	40
3.2 人员	40
3.2.1 项目参与者	41
3.2.2 项目负责人	41
3.2.3 软件项目组	42
3.2.4 协调和通信问题	44
3.3 问题	46
3.3.1 软件范围	46
3.3.2 问题分解	46
3.4 过程	47
3.4.1 合并问题和过程	47
3.4.2 过程分解	48
3.5 项目	49
3.6 小结	49
参考文献	49
思考题	50
推荐阅读文献及其他信息源	51
第4章 软件过程和项目的度量	53
4.1 测度、度量和指标	53
4.2 过程和项目领域中的度量	54
4.2.1 过程度量和软件过程改进	54
4.2.2 项目度量	57
4.3 软件测量	58
4.3.1 面向规模的度量	58
4.3.2 面向功能的度量	59
4.3.3 扩展的功能点度量	61
4.4 调和不同的度量方法	63
4.5 软件质量度量	63

4.5.1 概述影响质量的因素	64	6.3 识别风险	93
4.5.2 测量质量	64	6.3.1 产品规模风险	94
4.5.3 缺陷排除效率	65	6.3.2 商业影响风险	94
4.6 在软件过程中集成度量	66	6.3.3 客户相关的风险	95
4.7 小结	67	6.3.4 过程风险	95
参考文献	67	6.3.5 技术风险	96
思考题	68	6.3.6 开发环境风险	97
推荐阅读文献及其他信息源	69	6.3.7 与人员数目及经验相关的风险	97
第5章 软件项目计划	71	6.3.8 风险因素和驱动因子	98
5.1 对估算的观察	71	6.4 风险预测	98
5.2 项目计划目标	72	6.4.1 建立风险表	98
5.3 软件范围	72	6.4.2 评估风险影响	100
5.3.1 获取定义软件范围所需的信息	72	6.4.3 风险评估	100
5.3.2 一个范围定义的例子	73	6.5 风险缓解、监控和管理	101
5.4 资源	75	6.6 安全性风险和危险	103
5.4.1 人力资源	75	6.7 RMMM计划	103
5.4.2 可复用软件资源	76	6.8 小结	104
5.4.3 环境资源	76	参考文献	104
5.5 软件项目估算	77	思考题	104
5.6 分解技术	77	推荐阅读文献及其他信息源	105
5.6.1 软件规模估算	78	第7章 项目进度安排及跟踪	107
5.6.2 基于问题的估算	78	7.1 基本概念	107
5.6.3 一个基于LOC估算的例子	79	7.1.1 关于“延迟”的评注	108
5.6.4 一个基于FP估算的例子	80	7.1.2 基本原则	109
5.6.5 基于过程的估算	81	7.2 人员和工作量之间的关系	110
5.6.6 一个基于过程估算的例子	82	7.2.1 一个例子	110
5.7 经验估算模型	83	7.2.2 一个经验关系	111
5.7.1 估算模型的结构	83	7.2.3 工作量分布	111
5.7.2 COCOMO模型	83	7.3 为软件项目定义任务集合	111
5.7.3 软件方程式	85	7.3.1 严格度	112
5.8 自行开发或购买的决策	86	7.3.2 定义适应性准则	113
5.8.1 创建决策树	87	7.3.3 计算任务集合选择因子的值	113
5.8.2 外购	88	7.3.4 解释TSS值并选择任务集合	114
5.9 自动估算工具	88	7.4 选择软件工程任务	115
5.10 小结	88	7.5 主要任务的求精	115
参考文献	89	7.6 定义任务网络	118
思考题	89	7.7 进度安排	118
推荐阅读文献及其他信息源	90	7.7.1 时间表	119
第6章 风险管理	92	7.7.2 跟踪进度	119
6.1 被动和主动的风险策略	92	7.8 项目计划	122
6.2 软件风险	92	7.9 小结	122

参考文献	123
思考题	123
推荐阅读文献及其他信息源	124
第8章 软件质量保证	125
8.1 质量概念	125
8.1.1 质量	126
8.1.2 质量控制	126
8.1.3 质量保证	127
8.1.4 质量的成本	127
8.2 质量运动	128
8.3 软件质量保证	129
8.3.1 背景	129
8.3.2 SQA活动	130
8.4 软件复审	130
8.4.1 软件缺陷对成本的影响	131
8.4.2 缺陷的放大和消除	131
8.5 正式技术复审	133
8.5.1 复审会议	133
8.5.2 复审报告和记录保存	134
8.5.3 复审指南	134
8.6 SQA的形式化方法	135
8.7 统计质量保证	136
8.8 软件可靠性	137
8.8.1 可靠性和可用性的度量	138
8.8.2 软件的安全和危险的分析	138
8.9 SQA计划	139
8.10 ISO9000质量标准	140
8.10.1 ISO对质量保证系统的方法	141
8.10.2 ISO9001 标准	141
8.11 小结	142
参考文献	142
思考题	143
推荐阅读文献及其他信息源	144
第9章 软件配置管理	147
9.1 软件配置管理	147
9.1.1 基线	148
9.1.2 软件配置项	149
9.2 SCM过程	150
9.3 软件配置中对象的标识	151
9.4 版本控制	153
9.5 变化控制	154

9.6 配置审计	157
9.7 状态报告	157
9.8 SCM标准	157
9.9 小结	158
参考文献	158
思考题	159
推荐阅读文献及其他信息源	159

第三部分 传统软件工程方法

第10章 系统工程	161
10.1 基于计算机的系统	161
10.2 系统工程层次结构	163
10.2.1 系统建模	164
10.2.2 信息工程: 概述	165
10.2.3 产品工程: 概述	167
10.3 信息工程	168
10.4 信息策略计划	168
10.4.1 企业建模	169
10.4.2 业务级数据建模	170
10.5 业务域分析	171
10.5.1 处理建模	172
10.5.2 信息流建模	173
10.6 产品工程	173
10.6.1 系统分析	176
10.6.2 说明客户需求	176
10.6.3 可行性研究	176
10.6.4 经济分析	177
10.6.5 技术分析	179
10.7 系统体系结构建模	180
10.8 系统建模和仿真	182
10.9 系统规约	183
10.10 小结	184
参考文献	185
思考题	185
推荐阅读文献及其他信息源	187
第11章 分析概念和原则	188
11.1 需求分析	188
11.2 通信技术	189
11.2.1 过程的启动	189
11.2.2 便利的应用规约技术	190
11.2.3 质量功能部署	192

11.3 分析原则	193	12.9 小结	234
11.3.1 信息域	194	参考文献	234
11.3.2 建模	195	思考题	235
11.3.3 划分	196	推荐阅读文献及其他信息源	236
11.3.4 基本视图和实现视图	197	第13章 设计概念和原则	238
11.4 软件原型	198	13.1 软件设计和软件工程	238
11.4.1 选择原型方法	198	13.2 设计过程	239
11.4.2 原型方法和工具	199	13.2.1 设计和软件质量	239
11.5 规约	199	13.2.2 软件设计的演化	240
11.5.1 规约原则	200	13.3 设计原则	240
11.5.2 表示	200	13.4 设计概念	241
11.5.3 软件需求规约	201	13.4.1 抽象	241
11.6 规约复审	202	13.4.2 求精	242
11.7 小结	203	13.4.3 模块化	242
参考文献	203	13.4.4 软件体系结构	244
思考题	204	13.4.5 控制层次	245
推荐阅读文献及其他信息源	205	13.4.6 结构划分	246
第12章 分析建模	206	13.4.7 数据结构	247
12.1 简史	206	13.4.8 软件过程	247
12.2 分析模型的元素	207	13.4.9 信息隐蔽	248
12.3 数据建模	208	13.5 有效的模块设计	248
12.3.1 数据对象、属性和关系	208	13.5.1 功能独立性	249
12.3.2 基数和形态	210	13.5.2 内聚	249
12.3.3 实体—关系图	211	13.5.3 耦合	250
12.4 功能建模和信息流	214	13.6 针对有效模块化的设计启发	251
12.4.1 数据流图	214	13.7 设计模型	253
12.4.2 针对实时系统的扩展	216	13.8 设计文档	253
12.4.3 Ward和Mellor扩展	217	13.9 小结	254
12.4.4 Hatley和Pirbhai扩展	219	参考文献	254
12.5 行为建模	220	思考题	256
12.6 结构化分析的技巧	222	推荐阅读文献及其他信息源	256
12.6.1 创建实体—关系图	223	第14章 设计方法	258
12.6.2 创建数据流模型	225	14.1 数据设计	258
12.6.3 创建控制流模型	227	14.2 体系结构设计	259
12.6.4 控制规约	229	14.2.1 贡献者	259
12.6.5 加工规约	230	14.2.2 应用域	260
12.7 数据字典	231	14.3 体系结构设计过程	260
12.8 其他传统分析方法概述	233	14.3.1 变换流	260
12.8.1 数据结构化系统开发	233	14.3.2 事务流	261
12.8.2 Jackson系统开发	233	14.4 变换映射	261
12.8.3 SADT	234	14.4.1 一个实例	261

14.4.2 设计步骤	262	15.4 实时设计	306
14.5 事务映射	268	15.5 小结	307
14.5.1 一个实例	268	参考文献	307
14.5.2 设计步骤	268	思考题	309
14.6 设计的后处理	272	推荐阅读文献及其他信息源	309
14.7 体系结构设计优化	272	第16章 软件测试技术	311
14.8 接口设计	273	16.1 软件测试基础	311
14.8.1 内部和外部接口设计	273	16.1.1 测试目标	311
14.8.2 用户界面设计	273	16.1.2 测试原则	312
14.9 人机界面设计	273	16.1.3 可测试性	312
14.9.1 界面设计模型	274	16.2 测试用例设计	314
14.9.2 任务分析和建模	275	16.3 白盒测试	315
14.9.3 设计问题	275	16.4 基本路径测试	315
14.9.4 实现工具	277	16.4.1 流图符号	315
14.9.5 设计评估	277	16.4.2 环形复杂性	318
14.10 界面设计指南	279	16.4.3 导出测试用例	318
14.10.1 一般交互	279	16.4.4 图矩阵	321
14.10.2 信息显示	280	16.5 控制结构测试	322
14.10.3 数据输入	280	16.5.1 条件测试	322
14.11 过程设计	281	16.5.2 数据流测试	324
14.11.1 结构化程序设计	281	16.5.3 循环测试	325
14.11.2 图形设计符号	281	16.6 黑盒测试	326
14.11.3 表格设计符号	284	16.6.1 基于图的测试方法	326
14.11.4 程序设计语言	285	16.6.2 等价划分	328
14.11.5 一个PDL实例	285	16.6.3 边界值分析	329
14.12 小结	288	16.6.4 比较测试	330
参考文献	288	16.7 针对专门环境和应用的测试	330
思考题	290	16.7.1 GUI测试	330
推荐阅读文献及其他信息源	292	16.7.2 客户/服务器体系结构的测试	331
第15章 实时系统的设计	294	16.7.3 测试文档和帮助设施	331
15.1 系统考虑	294	16.7.4 实时系统测试	332
15.2 实时系统	295	16.8 小结	333
15.2.1 集成和性能问题	295	参考文献	333
15.2.2 中断处理	296	思考题	334
15.2.3 实时数据库	297	推荐阅读文献及其他信息源	335
15.2.4 实时操作系统	297	第17章 软件测试策略	337
15.2.5 实时语言	298	17.1 软件测试的策略途径	337
15.2.6 任务同步和通信	298	17.1.1 验证和确认	337
15.3 实时系统的分析和仿真	299	17.1.2 软件测试的组织	338
15.3.1 实时系统分析的数学工具	299	17.1.3 一种软件测试策略	339
15.3.2 仿真和建模技术	302	17.1.4 测试完成的标准	340

17.2 策略问题	342	18.4.1 高层设计度量	369
17.3 单元测试	342	18.4.2 构件级设计度量	371
17.3.1 单元测试考虑	342	18.4.3 界面设计度量	373
17.3.2 单元测试规程	344	18.5 源代码度量	374
17.4 集成测试	345	18.6 对测试的度量	375
17.4.1 自顶向下集成	346	18.7 对维护的度量	376
17.4.2 自底向上集成	347	18.8 小结	376
17.4.3 回归测试	348	参考文献	377
17.4.4 关于集成测试的讨论	349	思考题	378
17.4.5 集成测试文档	349	推荐阅读文献及其他信息源	379
17.5 确认测试	350		
17.5.1 确认测试的标准	350	第四部分 面向对象的软件工程	
17.5.2 配置复审	351	第19章 面向对象的概念和原则	381
17.5.3 Alpha和Beta测试	351	19.1 面向对象的范型	382
17.6 系统测试	351	19.2 面向对象概念	383
17.6.1 恢复测试	352	19.2.1 类和对象	385
17.6.2 安全测试	352	19.2.2 属性	386
17.6.3 压力测试	352	19.2.3 操作、方法和服务	386
17.6.4 性能测试	353	19.2.4 消息	386
17.7 调试的技巧	353	19.2.5 封装、继承和多态	388
17.7.1 调试过程	353	19.3 标识对象模型的元素	390
17.7.2 心理考虑	354	19.3.1 标识类和对象	390
17.7.3 调试方法	354	19.3.2 表示属性	393
17.8 小结	355	19.3.3 定义操作	394
参考文献	356	19.3.4 完成对象定义	394
思考题	357	19.4 面向对象软件项目的管理	395
推荐阅读文献及其他信息源	357	19.4.1 OO的公共过程框架	395
第18章 软件的技术度量	359	19.4.2 面向对象项目的度量和估算	396
18.1 软件质量	359	19.4.3 一种OO估算和进度安排方法	397
18.1.1 McCall的质量因素	360	19.4.4 面向对象项目的跟踪过程	398
18.1.2 FURPS	362	19.5 小结	399
18.1.3 到量化视图的变迁	362	参考文献	399
18.2 软件技术度量框架	363	思考题	400
18.2.1 技术度量的挑战	363	推荐阅读文献及其他信息源	400
18.2.2 测度原则	363	第20章 面向对象分析	402
18.2.3 有效软件度量的属性	364	20.1 面向对象的分析	402
18.3 分析模型的度量	365	20.1.1 传统方法和OO方法	402
18.3.1 基于功能的度量	365	20.1.2 OOA概述	403
18.3.2 “撞击值”度量	366	20.2 领域分析	406
18.3.3 规约质量的度量	368	20.2.1 复用和领域分析	406
18.4 设计模型的度量	369	20.2.2 领域分析过程	406

20.3 OO分析模型的类属成分	408	22.1 扩大测试的视角	447
20.4 OOA过程	409	22.2 测试OOA和OOD模型	448
20.4.1 使用实例	409	22.2.1 OOA和OOD模型的正确性	448
20.4.2 类—责任—协作者建模	410	22.2.2 OOA和OOD模型的一致性	448
20.4.3 定义结构和层次	414	22.3 面向对象的测试策略	449
20.4.4 定义主题和子系统	415	22.3.1 在OO语境中的单元测试	450
20.5 对象—关系模型	416	22.3.2 在OO语境中的集成测试	450
20.6 对象—行为模型	418	22.3.3 在OO语境中的有效性测试	450
20.6.1 用使用实例标识事件	419	22.4 OO软件的测试用例设计	450
20.6.2 状态表示	419	22.4.1 OO概念的测试用例设计的含义	451
20.7 小结	422	22.4.2 传统测试用例设计方法的可用性	451
参考文献	422	22.4.3 基于故障的测试	451
思考题	423	22.4.4 OO编程对测试的影响	452
推荐阅读文献及其他信息源	424	22.4.5 测试用例和类层次	453
第21章 面向对象设计	426	22.4.6 基于场景的测试设计	453
21.1 面向对象系统的设计	426	22.4.7 测试表层结构和深层结构	454
21.1.1 传统方法和OO方法	427	22.5 在类级别上可用的测试方法	455
21.1.2 设计问题	428	22.5.1 对OO类的随机测试	455
21.1.3 OOD概述	429	22.5.2 在类级别上的划分测试	455
21.2 OO设计模型的类属成分	432	22.6 类间测试用例设计	456
21.3 系统设计过程	433	22.6.1 多个类测试	456
21.3.1 划分分析模型	433	22.6.2 从行为模型导出的测试	457
21.3.2 并发性和子系统分配	434	22.7 小结	458
21.3.3 任务管理构件	434	参考文献	458
21.3.4 数据管理构件	435	思考题	459
21.3.5 资源管理构件	435	推荐阅读文献及其他信息源	459
21.3.6 人机界面构件	435	第23章 面向对象系统的技术度量	461
21.3.7 子系统间通信	436	23.1 面向对象度量的目的	461
21.4 对象设计过程	437	23.2 区别性的特征	461
21.4.1 对象描述	437	23.2.1 局部化	461
21.4.2 设计算法和数据结构	438	23.2.2 封装	462
21.4.3 程序构件和接口	439	23.2.3 信息隐蔽	462
21.5 设计模式	442	23.2.4 继承	462
21.5.1 描述设计模式	442	23.2.5 抽象	462
21.5.2 在设计中使用设计模式	442	23.3 对OO设计模型的度量	462
21.6 面向对象编程	443	23.4 面向类的度量	463
21.7 小结	443	23.4.1 CK度量套件	463
参考文献	443	23.4.2 Lorenz和Kidd建议的度量	465
思考题	444	23.5 面向操作的度量	466
推荐阅读文献及其他信息源	445	23.6 对面向对象测试的度量	466
第22章 面向对象测试	447	23.6.1 对面向对象项目的度量	467

23.8 小结	467
参考文献	468
思考题	468
推荐阅读文献及其他信息源	469

第五部分 软件工程高级课题

第24章 形式化方法	471
24.1 基本概念	471
24.1.1 少量的形式化方法的不足	472
24.1.2 软件开发中的数学	473
24.1.3 形式化方法概念	473
24.2 数学预备知识	477
24.2.1 集合和构造性规约	477
24.2.2 集合运算符	478
24.2.3 逻辑运算符	480
24.2.4 序列	480
24.3 应用数学符号描述形式规约	481
24.4 形式化规约语言	482
24.5 用Z表示一个软件构件的例子	483
24.6 形式化方法的十条戒律	485
24.7 形式化方法——未来之路	486
24.8 小结	486
参考文献	487
思考题	487
推荐阅读文献及其他信息源	488
第25章 净室软件工程	490
25.1 净室方法	490
25.1.1 净室策略	490
25.1.2 什么使得净室独特?	492
25.2 功能规约	492
25.2.1 黑盒规约	493
25.2.2 状态盒规约	494
25.2.3 清晰盒规约	494
25.3 设计求精和验证的优点	495
25.3.1 设计求精和验证	495
25.3.2 设计验证的优点	498
25.4 净室测试	499
25.4.1 统计的使用测试	499
25.4.2 认证	500
25.5 小结	500
参考文献	501

思考题	502
推荐阅读文献及其他信息源	502
第26章 软件复用	505
26.1 管理问题	505
26.1.1 复用的障碍	505
26.1.2 硬件类比	506
26.1.3 建立复用途径的一些建议	507
26.2 复用过程	507
26.2.1 可复用的软件制品	508
26.2.2 一个过程模型	509
26.3 领域工程	509
26.3.1 领域分析过程	510
26.3.2 领域特征	511
26.3.3 结构建模和结构点	511
26.4 建造可复用构件	512
26.4.1 为了复用的分析和设计	512
26.4.2 构造方法	513
26.4.3 基于构件的开发	514
26.5 分类和检索构件	515
26.5.1 描述可复用构件	515
26.5.2 复用环境	517
26.6 软件复用经济学	517
26.6.1 对质量、生产率和成本的影响	517
26.6.2 使用结构点的成本分析	518
26.6.3 复用度量	519
26.7 小结	519
参考文献	520
思考题	521
推荐阅读文献及其他信息源	522
第27章 再工程	524
27.1 业务过程再工程	524
27.1.1 业务过程	524
27.1.2 业务过程再工程的原则	526
27.1.3 BPR模型	527
27.1.4 几句警告	528
27.2 软件再工程	528
27.2.1 软件维护	528
27.2.2 软件再工程过程模型	529
27.3 逆向工程	532
27.3.1 理解处理的逆向工程	533
27.3.2 理解数据的逆向工程	533

27.3.3 用户界面的逆向工程	534	28.5.2 C/S测试策略	553
27.4 重构	535	28.6 小结	554
27.4.1 代码重构	535	参考文献	554
27.4.2 数据重构	536	思考题	555
27.5 正向工程	536	推荐阅读文献及其他信息源	555
27.5.1 向客户/服务器体系结构的 正向工程	537	第29章 计算机辅助软件工程	557
27.5.2 向面向对象体系结构的正向工程	538	29.1 什么是CASE?	557
27.5.3 用户界面的正向工程	538	29.2 构造CASE的积木块	557
27.6 再工程经济学	539	29.3 CASE工具分类	559
27.7 小结	539	29.4 集成化CASE环境	562
参考文献	540	29.5 集成体系结构	563
思考题	541	29.6 CASE中心库	563
推荐阅读文献及其他信息源	542	29.6.1 在I-CASE中中心库的角色	564
第28章 客户/服务器软件工程	543	29.6.2 特征和内容	564
28.1 客户/服务器系统的结构	543	29.7 小结	567
28.1.1 C/S系统的软件构件	544	参考文献	567
28.1.2 软件构件的分布	545	思考题	568
28.1.3 分布应用构件的指南	545	推荐阅读文献及其他信息源	568
28.1.4 连接C/S软件构件	546	第30章 未来之路	570
28.1.5 中间件和对象请求代理体系结构	546	30.1 软件的重要性——再论	570
28.2 对C/S系统的软件工程	547	30.2 变化的范围	571
28.3 分析建模问题	548	30.3 人及他们建造系统的方式	572
28.4 对C/S系统的设计	548	30.4 “新”的软件过程	574
28.4.1 传统设计方法	548	30.5 表示信息的新模式	574
28.4.2 数据库设计	549	30.6 技术作为推动力	576
28.4.3 某设计方法的概述	550	30.7 结束语	577
28.4.4 加工设计的迭代	551	参考文献	578
28.5 测试问题	552	思考题	578
28.5.1 整体C/S测试策略	552	推荐阅读文献及其他信息源	578

第一部分 产品和过程

在本书的这一部分中我们主要讨论什么是工程产品和如何为工程技术提出一个框架的过程。在下面的章节中，我们主要解决下列问题：

- 到底什么是计算机软件？
- 为什么我们不断努力要建造高质量的基于计算机的系统？
- 我们如何对计算机软件的应用领域分类？
- 关于软件仍存在什么样的神话？
- 什么是软件过程？
- 是否存在一般性的方法评价一个过程的质量？
- 软件开发中可以应用什么过程模型？
- 线性过程和迭代过程有何区别？
- 它们的优点和缺点是什么？
- 在软件工程中可以建议什么更高级的过程模型？

一旦回答了这些问题，读者就能够更好地理解本书其余部分给出的工程原则的管理和技术方面的知识。

第1章 产 品

本书的第1版在80年代初出版后不久，Business Weekly(《商业周刊》)杂志在头版给出如下的大标题：“软件：新的驱动力”。编辑们当时并没有意识到他们的预见是多么的正确。那时，大多数人对软件还是一无所知。大软件公司，如微软公司，还不存在；拥有15 000平方英尺专门出售包装好的软件的计算机超市闻所未闻；在电视上为计算机操作系统做60秒钟商业广告的想法是可笑的；而互联网仅为个别研究者和高等学校学生所知。但是，在不到20年的时间里，所有这些(甚至更多)已经成为现实。

计算机软件已经成为一种驱动力。它是进行商业决策的引擎；它是现代科学研究和工程问题解决的基础；它也是区分现代产品和服务的关键因素。它在各种类型的系统中应用，如交通、医药、通讯、军事、产业化过程、娱乐、办公……难以穷举。软件在现代社会中的确是必不可少的。而且当我们进入21世纪，软件将成为从基础教育到基因工程的所有各领域新进展的驱动器。

所有这一切已经改变了软件的常见概念。计算机软件是无所不在的，人们把软件看作是生活中现实的技术。在很多情况下，人们把他们的工作、他们的舒适、他们的安全、他们的娱乐、他们的决策、甚至他们的整个生活都依赖于计算机软件。软件千万可不能出错。

本书介绍的若干技术是那些想要建造正确的计算机软件的人们需要用到的。这些技术包括一个过程，一组方法和一系列工具，我们称之为软件工程。