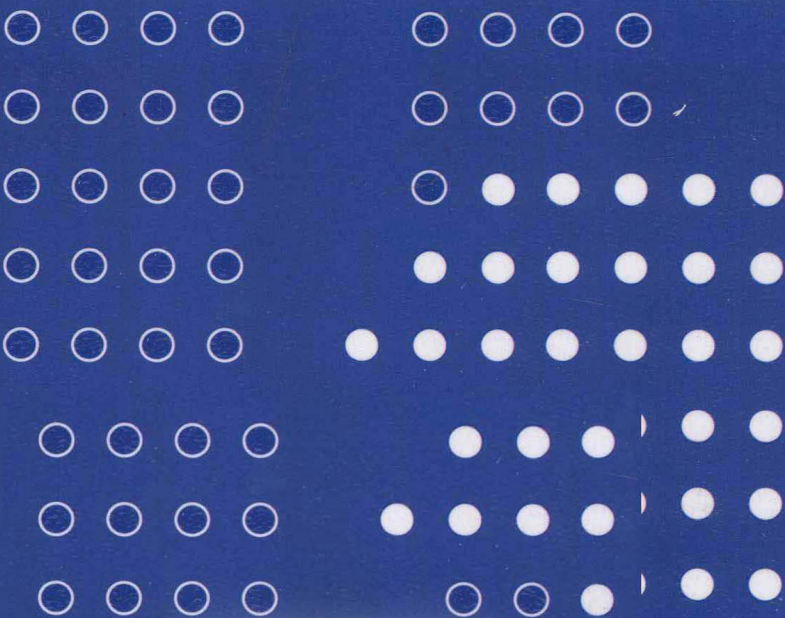
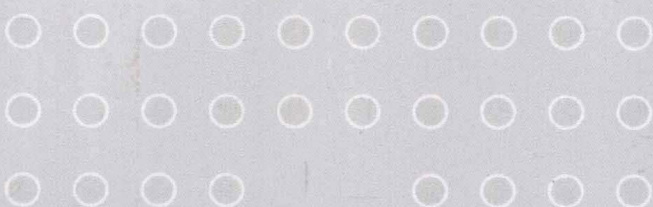




普通高等教育“十一五”国家级规划教材 计算机系列教材

实用软件工程

(第三版)



殷人昆 郑人杰 马素霞 白晓颖 编著



清华大学出版社

殷人昆 郑人杰 马素霞 白晓颖 编著

实用软件工程

(第三版)

清华大学出版社
北京

内 容 简 介

本书是《实用软件工程》的第三版。本书的第二版出版后,在国内被许多学校和培训班用作教材,部分内容被其他相关教材多次引用,受到普遍好评。由于本领域在近年来发展极快,新的知识和技术不断涌现,如果限于一本教材,难于反映全貌,也无法授人以渔,故第三版分为“开发篇”——《实用软件工程(第三版)》与“管理篇”——《实用软件工程高级教程(第三版)》两册。本册“开发篇”共10章,系统地介绍了软件工程的概 念、方法和技术,包括软件生存周期、需求分析、设计、编码、测试、维护等;另一册“管理篇”共8章,系统地介绍了软件工程管理、软件过程、质量和质量保证、可靠性、软件标准和文档、软件工具、MDA 和 SOA 等。本书力图让学习者不但能理解相关知识,而且能学会运用相关技能。

本册适用于计算机专业的本科生、非计算机专业的本科生和研究生;“管理篇”适用于计算机专业的研究生和其他学习软件工程的专业人员,也可用作培训班的教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

实用软件工程 / 殷人昆等编著. —3 版. —北京:清华大学出版社, 2010.11

(计算机系列教材)

ISBN 978-7-302-22200-2

I. ①实… II. ①殷… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2010)第 036525 号

责任编辑:郑寅堃

责任校对:白 蕾

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京鑫海金澳胶印有限公司

装 订 者:三河市李旗庄少明装订厂

经 销:全国新华书店

开 本:185×260

印 张:33.5

字 数:805 千字

版 次:2010 年 11 月第 3 版

印 次:2010 年 11 月第 1 次印刷

印 数:1~4000

定 价:49.00 元

产品编号:032414-01

普通高等教育“十一五”国家级规划教材

计算机系列教材

主编：周立柱、王志英、李晓明

书名	作者	定价
计算机组织与体系结构(第4版)解题指南	白中英	19
计算机组织与体系结构(第4版 立体化教材)	白中英	43
计算机操作系统教程(第3版)	张尧学、史美林	25
计算机操作系统教程(第3版)习题解答与实验指导	张尧学	15
高档微机原理与技术	毛国君、方娟	21
计算机组成原理与汇编语言	易小琳 等	39
微型机原理与技术(第2版)	戴梅萼、史嘉权	33
微型机原理与技术——习题、实验和综合训练题集(第2版)	戴梅萼、史嘉权	18
编译原理	陈英 等	32
MPI 并行程序设计实例教程	张武生 等	39.5
计算机组成与设计	薛宏熙	36
数字逻辑设计	薛宏熙	33
计算机英语(第四版)	刘兆毓、郑家农	35
数据库系统设计与原理(第2版)	冯建华、周立柱	24
数据库专题训练	冯建华、周立柱	20
C/C++ 与数据结构(第3版)上册	王立柱	38
C/C++ 与数据结构(第3版)下册	王立柱	17
C/C++ 与数据结构(第3版)(上册)习题解答与实验指导	刘志红 等	18

主 任：周立柱

副 主 任：王志英 李晓明

编委委员：（按姓氏笔画为序）

汤志忠 孙吉贵 杨 波

岳丽华 钱德沛 谢长生

蒋宗礼 廖明宏 樊晓桢

责任编辑：马瑛珺

本书第二版自 1997 年发行以来,已经历了十多个年头。近年来它成为许多高等学校计算机相关专业软件工程课的首选教材,已经拥有了二十几万的读者。在这十几年中,软件技术以及与其相关的计算机系统和网络技术都已取得了长足的进步,原书内容的调整与更新自然已成为适应趋势之必需。然而,在进行第三版的修订之时,作者仍然力图坚守第一、第二版的选材原则,希望继续体现简明与实用。当然,过于简短并不能充分地阐述软件工程的基本概念、基本原则与基本方法,也将无法满足广大读者的需求,指导软件工程实践。

为此第三版保留了软件生存期过程的基本内容,包括软件需求、设计、实现、测试和维护等,同时增加了统一建模语言 UML,扩充了面向对象方法,以此来满足高校计算机相关专业本科教学的要求。另一方面,把涉及软件管理,包括项目管理、配置管理、质量管理、软件过程、软件工程标准以及软件工具的内容分离出来,另成一册,并增加了体现软件面向服务(SOA)的内容,以期适应研究生教学的要求。

关于如何把握好内容的更新,我们认为,正是由于本书的主要读者对象是初学者,他们在学习更应着重掌握好软件工程的基本知识和基本内容,而不是一味地求新。例如,当前在一些软件开发组织中流行着“敏捷开发方法”,尽管该方法具有简单、灵活的优点,我们仍然不主张初学者从敏捷开发入手。毋庸置疑,传统的软件工程方法有助于培养严谨的思维、规范化的作风和工程实践,而这一点恰是软件工程师的职业生涯中必不可缺和至关重要的。还是先学走再学跑为妥,否则本末倒置,也许掌握了技巧却丢掉了基本功,那种“捷径”是不可取的。

在内容更新方面需要说明的另一点是软件标准的更新。由于软件工程的规范化实践很大程度上体现在能否遵循和参考软件工程标准,为此,必须及时地跟踪新的国际标准以及最新修订的国家标准。本书的第三版已尽可能选用了最新标准版本予以介绍,并希望通过这一部分让读者建立软件工程标准化的概念。

以下三位作者参与了第三版的工作:马素霞教授负责编写软件设计工程、体系结构设计与设计模式、软件配置管理及软件工具与环境等部分;本人负责编写软件质量管理、软件过程、软件工程标准及软件文档部分;其余部分由殷人昆教授和白晓颖副教授编写。

读者若有反馈意见请径告作者或由清华大学出版社(zhengyk@tup.tsinghua.edu.cn)转告。

郑人杰
2010 年 8 月

第1章 软件工程概述 /1

- 1.1 软件的概念、特点和分类 /1
 - 1.1.1 软件的概念及特点 /1
 - 1.1.2 软件的分类 /2
 - 1.1.3 软件的发展及软件危机 /3
- 1.2 软件工程 /5
 - 1.2.1 软件工程的定义 /5
 - 1.2.2 软件工程的框架 /6
 - 1.2.3 软件工程知识体系及知识域 /7
 - 1.2.4 软件工程的基本原理 /10
- 1.3 软件生存周期与软件过程 /11
 - 1.3.1 软件生存周期的基本任务 /11
 - 1.3.2 软件过程 /13
- 1.4 软件过程模型 /13
 - 1.4.1 瀑布模型 /14
 - 1.4.2 快速原型模型 /15
 - 1.4.3 形式化系统开发模型 /16
 - 1.4.4 面向复用的开发模型 /17
 - 1.4.5 增量模型 /18
 - 1.4.6 螺旋模型 /19
 - 1.4.7 喷泉模型 /21
 - 1.4.8 智能模型 /21
 - 1.4.9 快速应用开发模型 /22
 - 1.4.10 Rational 统一开发过程 /24
- 1.5 问题解决和范型 /26
 - 1.5.1 范型 /27
 - 1.5.2 流行的范型 /27

第2章 计算机系统工程 /32

- 2.1 基于计算机的系统 /32
- 2.2 计算机系统工程 /33

2.2.1	识别用户的要求	/33
2.2.2	系统分析和结构设计	/35
2.2.3	可行性研究	/36
2.2.4	建立成本和进度的限制	/36
2.2.5	生成系统需求规格说明	/36
2.3	系统分析与结构设计	/38
2.3.1	系统分析的层次	/38
2.3.2	业务过程工程和产品工程建模	/39
2.3.3	系统模型模板	/41
2.3.4	系统文档与评审	/45
2.4	可行性研究	/46
2.4.1	经济可行性	/46
2.4.2	技术可行性	/49
2.4.3	法律可行性	/50
2.4.4	用户操作可行性	/50
2.4.5	方案的选择和折衷	/50
2.4.6	可行性研究报告	/51
2.5	其他系统描述方法	/52
2.5.1	系统框图和系统流程图	/52
2.5.2	HIPO 建模	/53
 第3章 面向对象方法与 UML /57		
3.1	面向对象系统的概念	/57
3.1.1	面向对象系统的概念	/57
3.1.2	对象	/58
3.1.3	类与封装	/59
3.1.4	继承	/60
3.1.5	多态性和动态绑定	/61
3.1.6	消息通信	/62
3.1.7	对象生存周期	/63
3.2	统一建模语言 UML 概述	/63
3.2.1	什么是建模	/63
3.2.2	UML 发展历史	/64

3.2.3	UML 的特点	/65
3.2.4	UML 的视图	/66
3.3	UML 的模型元素	/67
3.3.1	UML 的事物	/68
3.3.2	UML 中的关系	/69
3.4	UML 中的图	/75
3.4.1	外部视图	/75
3.4.2	内部视图	/80
3.5	UML 的元模型结构	/86
3.6	UML 建模工具 Rational Rose	/87
3.6.1	Rose 的特点	/88
3.6.2	Rose 简介	/89
3.6.3	Rose 的基本操作	/90
3.6.4	在 Rose 环境下建立 UML 模型	/93
第 4 章	软件需求工程	/108
4.1	软件需求工程基础	/108
4.1.1	软件需求的定义和层次	/108
4.1.2	软件需求工程过程	/111
4.1.3	需求工程方法	/114
4.2	需求获取	/115
4.2.1	需求获取的任务和原则	/115
4.2.2	需求获取的过程	/116
4.2.3	需求的表达	/120
4.2.4	用逆向沟通改善需求的质量	/123
4.3	传统的分析建模方法	/124
4.3.1	数据建模	/125
4.3.2	功能建模	/127
4.3.3	行为建模	/132
4.3.4	数据字典	/136
4.3.5	基本加工逻辑说明	/139
4.4	面向对象的分析建模方法	/142
4.4.1	面向对象分析建模概述	/142

4.4.2	识别类或对象	/143
4.4.3	识别关系(结构)	/149
4.4.4	标识类的属性和服务	/150
4.4.5	分析模型评审	/152
4.5	原型化方法	/153
4.5.1	软件原型的分类	/153
4.5.2	快速原型开发模型	/154
4.5.3	原型开发技术	/157
4.6	需求规格说明	/159
4.6.1	软件需求规格说明的目标	/159
4.6.2	软件需求规格说明编制的原则	/159
4.6.3	软件需求规格说明模板	/161
4.6.4	SRS 和 DRD 的质量要求	/163
4.7	软件需求评审	/165
4.7.1	正式的需求评审	/165
4.7.2	需求评审中的常见风险	/167
4.8	软件需求管理	/167
4.8.1	需求管理的概念	/167
4.8.2	需求规格说明的版本控制	/168
4.8.3	需求跟踪	/169
4.8.4	需求变更请求的管理	/172
 第 5 章 软件设计工程 /175		
5.1	软件设计的目标与准则	/175
5.1.1	性能准则	/175
5.1.2	可靠性准则	/175
5.1.3	成本准则	/176
5.1.4	维护准则	/176
5.1.5	最终用户准则	/177
5.2	软件设计工程的任务	/177
5.2.1	软件设计的概念	/177
5.2.2	软件设计的阶段与任务	/178
5.2.3	软件设计的过程	/179

5.3	创建良好设计的原则	/180
5.3.1	分而治之和模块化	/180
5.3.2	模块独立性	/181
5.3.3	尽量降低耦合性	/181
5.3.4	尽量提高内聚性	/184
5.3.5	提高抽象层次	/186
5.3.6	复用性设计	/187
5.3.7	灵活性设计	/187
5.3.8	预防过期	/188
5.3.9	可移植性设计	/188
5.3.10	可测试性设计	/188
5.3.11	防御性设计	/189
5.4	传统的面向过程的设计方法	/189
5.4.1	结构化设计与结构化分析的关系	/190
5.4.2	软件结构及表示工具	/190
5.4.3	典型的数据流类型和系统结构	/194
5.4.4	变换流映射	/197
5.4.5	事务流映射	/200
5.4.6	软件模块结构改进的方法	/201
5.4.7	接口设计	/205
5.5	面向对象的系统设计	/205
5.5.1	子系统分解	/206
5.5.2	问题域部分的设计	/208
5.5.3	人机交互部分的设计	/210
5.5.4	任务管理部分的设计	/213
5.5.5	数据管理部分的设计	/214
5.6	对象设计	/216
5.6.1	使用模式设计对象	/216
5.6.2	接口规格说明设计	/220
5.6.3	重构对象设计模型	/222
5.6.4	优化对象设计模型	/222
5.7	处理过程设计	/223
5.7.1	结构化程序设计	/223

- 5.7.2 程序流程图 /224
- 5.7.3 N-S图 /227
- 5.7.4 PAD图 /228
- 5.7.5 程序设计语言 PDL /230
- 5.7.6 判定表 /230
- 5.7.7 HIPO /232
- 5.8 软件设计规格说明 /232
 - 5.8.1 软件(结构)设计说明(SDD) /232
 - 5.8.2 数据库(顶层)设计说明(DBDD) /233
 - 5.8.3 接口设计说明(IDD) /234
- 5.9 软件设计评审 /235
 - 5.9.1 概要设计评审的检查内容 /235
 - 5.9.2 详细设计评审的检查内容 /236

第6章 体系结构设计与设计模式 /238

- 6.1 软件体系结构的概念 /238
 - 6.1.1 什么是体系结构 /238
 - 6.1.2 体系结构的重要作用 /239
 - 6.1.3 构件的定义与构件之间的关系 /239
- 6.2 体系结构设计 with 风格 /241
 - 6.2.1 体系结构设计的过程 /241
 - 6.2.2 系统环境表示 /241
 - 6.2.3 体系结构的结构风格 /242
 - 6.2.4 体系结构的控制模型 /247
 - 6.2.5 体系结构的模块分解 /249
- 6.3 特定领域的软件体系结构 /250
 - 6.3.1 类属模型 /250
 - 6.3.2 参考模型 /251
- 6.4 分布式系统结构 /252
 - 6.4.1 多处理器体系结构 /252
 - 6.4.2 客户机/服务器体系结构 /252
 - 6.4.3 分布式对象体系结构 /256

6.4.4	代理	/257
6.4.5	聚合和联邦体系	/258
6.5	软件体系结构的评价	/260
6.6	体系结构描述语言	/261
6.7	设计模式	/262
6.7.1	什么是设计模式	/263
6.7.2	设计模式分类	/264
6.7.3	创建型设计模式	/264
6.7.4	结构型设计模式	/271
6.7.5	行为型设计模式	/281
6.7.6	设计模式如何解决设计问题	/294
6.7.7	如何使用设计模式	/298
第7章	软件实现	/300
7.1	软件实现的过程与任务	/300
7.2	程序设计方法概述	/301
7.2.1	结构化程序设计	/302
7.2.2	面向对象的程序设计方法	/304
7.2.3	极限编程	/308
7.3	编程风格与编码标准	/312
7.3.1	源程序文档化	/312
7.3.2	数据说明规范化	/314
7.3.3	程序代码结构化	/315
7.3.4	输入/输出风格可视化	/318
7.3.5	编程规范	/320
7.4	编程语言	/324
7.4.1	编程语言特性的比较	/325
7.4.2	编程语言的分类	/328
7.4.3	编程语言的选择	/334
7.5	程序效率与性能分析	/335
7.5.1	算法对效率的影响	/335

7.5.2	影响存储器效率的因素	/336
7.5.3	影响输入/输出的因素	/336
7.6	程序复杂性	/336
7.6.1	代码行度量法	/337
7.6.2	McCabe 度量法	/337
7.6.3	Henry-Kafura 的信息流度量	/339
7.6.4	Thayer 复杂性度量	/339
7.6.5	Halstead 的软件科学	/341
7.6.6	软件复杂性的综合度量	/343
第8章	软件测试工程	/344
8.1	软件测试的任务	/344
8.1.1	软件测试的目的和定义	/344
8.1.2	软件测试的原则	/345
8.1.3	软件测试的对象	/347
8.1.4	测试信息流	/347
8.1.5	软件测试的生存周期模型	/348
8.1.6	软件的确认和验证	/349
8.1.7	软件测试文档	/349
8.2	软件错误	/352
8.2.1	按错误的影响和后果分类	/352
8.2.2	按错误的性质和范围分类	/352
8.2.3	按软件生存周期阶段分类	/353
8.2.4	错误统计	/354
8.3	人工测试	/354
8.3.1	桌面检查	/354
8.3.2	代码检查	/356
8.3.3	走查	/358
8.4	软件开发生存周期中的测试活动	/359
8.4.1	软件需求分析阶段的测试活动	/360
8.4.2	软件设计阶段的测试活动	/361
8.4.3	编程及单元测试阶段的测试活动	/363
8.4.4	集成测试阶段的测试活动	/364

8.4.5	系统测试阶段的测试活动	/366
8.4.6	验收测试	/366
8.4.7	运行和维护阶段的测试活动	/367
8.4.8	回归测试	/368
8.5	面向对象的测试	/369
8.5.1	面向对象软件测试的问题	/369
8.5.2	面向对象软件测试的模型	/371
8.5.3	面向对象分析的测试	/372
8.5.4	面向对象设计的测试	/372
8.5.5	面向对象编程的测试	/373
8.5.6	面向对象程序的单元测试	/373
8.5.7	面向对象程序的集成测试	/373
8.5.8	面向对象软件的系统测试	/374
8.6	单元测试	/374
8.6.1	单元测试的定义和目标	/374
8.6.2	单元测试环境	/375
8.6.3	单元测试策略	/376
8.6.4	单元测试分析	/377
8.6.5	面向对象程序的单元测试	/379
8.7	集成测试	/381
8.7.1	集成测试的定义和目标	/381
8.7.2	集成测试环境	/381
8.7.3	集成测试策略	/382
8.7.4	集成测试分析	/387
8.7.5	面向对象程序的集成测试	/390
8.8	系统测试	/391
8.8.1	系统测试的定义与目标	/391
8.8.2	系统测试环境	/391
8.8.3	系统测试策略	/392
8.8.4	系统测试分析	/400
8.9	程序调试	/401
8.9.1	程序调试的步骤	/401
8.9.2	几种主要的调试方法	/402

8.9.3 调试的原则 /404

第9章 软件测试用例设计 /406

9.1 测试用例设计概述 /406

9.1.1 测试用例的重要性 /406

9.1.2 测试用例数和软件规模的关系 /407

9.1.3 测试用例设计说明的书写规范 /407

9.2 软件测试用例设计方法 /409

9.2.1 黑盒测试方法(Black-Box Testing) /409

9.2.2 白盒测试方法(White-Box Testing) /410

9.3 白盒测试用例设计方法 /411

9.3.1 逻辑覆盖 /411

9.3.2 判定和循环结构测试 /416

9.3.3 基本路径测试 /418

9.4 黑盒测试用例设计方法 /420

9.4.1 等价类划分 /420

9.4.2 边界值分析 /424

9.4.3 判定表法 /426

9.4.4 因果图法 /428

9.4.5 其他黑盒测试用例设计方法 /431

9.4.6 选择测试方法的综合策略及工作步骤 /432

9.5 单元测试用例设计 /433

9.5.1 单元测试用例设计的步骤 /433

9.5.2 单元测试用例设计方法 /434

9.5.3 构建类声明的测试用例 /437

9.5.4 根据状态图构建测试用例 /440

9.6 集成测试的测试用例设计 /442

9.6.1 集成测试用例设计的步骤 /442

9.6.2 基于协作图生成集成测试用例设计 /443

9.6.3 继承关系的测试用例设计 /449

9.7 系统测试用例的设计 /450

- 9.7.1 基于场景设计测试用例 /450
- 9.7.2 基于功能图设计测试用例 /455
- 9.7.3 基于有限状态机的系统级线索设计测试用例 /457
- 9.7.4 基于 UML 的系统级线索测试用例设计 /461

第 10 章 软件维护 /462

- 10.1 软件维护的概念 /462
 - 10.1.1 软件维护的定义 /462
 - 10.1.2 影响维护工作量的因素 /463
 - 10.1.3 软件维护的策略 /464
 - 10.1.4 维护成本 /465
- 10.2 软件维护的活动 /466
 - 10.2.1 维护机构 /466
 - 10.2.2 软件维护申请报告 /466
 - 10.2.3 软件维护过程模型 /467
 - 10.2.4 软件维护的一般工作流程 /468
 - 10.2.5 维护记录文档 /469
 - 10.2.6 维护评价 /469
- 10.3 程序修改的步骤及修改的副作用 /470
 - 10.3.1 结构化维护与非结构化维护 /470
 - 10.3.2 软件维护面临的问题 /471
 - 10.3.3 分析和理解程序 /472
 - 10.3.4 评估修改范围 /474
 - 10.3.5 修改程序 /474
 - 10.3.6 重新验证程序 /476
- 10.4 面向对象软件的维护 /478
- 10.5 软件可维护性 /480
 - 10.5.1 可维护性的外部视图 /480
 - 10.5.2 影响可维护性的内部质量属性 /481
 - 10.5.3 其他可维护性的度量 /483
- 10.6 提高可维护性的方法 /488