

软件工程 习题与解答

Schaum's Outlines of Software Engineering

最佳的复习资料，实用的辅助教材

与国外高校计算机水平保持同步

为考研和出国深造奠定坚实基础

(美) David Gustafson 著

钟鸣 王君 等译

本系列丛书
全球销售超过
3000万册!

全美经典
学习指导系列

软件工程 习题与解答

Schaum's Outlines of Software Engineering

(美) David Gustafson 著

钟鸣 王君 等译

本书以简明扼要的语言介绍了软件工程的基本概念和基本方法,涉及软件生命周期、软件过程模型和其他模型、软件项目管理、软件度量、风险分析和管理、软件质量保证、软件设计和软件测试等内容。通过阅读本书,读者能迅速了解软件工程的相关知识,并将这些概念和技术用于实际的系统开发中。本书内容全面、实例和习题极为丰富,是软件工程学科的一本极佳的教辅书籍。本书适合软件工程专业的本科生、研究生使用,也是急需了解软件工程知识的技术人员的入门书籍。

David Gustafson: Schaum's Outlines of Software Engineering (ISBN: 0-07-137794-8) .

Copyright © 2002 by The McGraw-Hill Companies, Inc.

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

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。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。

本书封面贴有McGraw-Hill公司防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2002-4821

图书在版编目(CIP)数据

软件工程习题与解答 / (美) 古斯塔夫森 (Gustafson, D.) 著; 钟鸣等译. - 北京: 机械工业出版社, 2003.7

(全美经典学习指导系列)

书名原文: Schaum's Outlines of Software Engineering

ISBN 7-111-12245-3

I. 软… II. ①古… ②钟… III. 软件工程-解题 IV. TP311.5-44

中国版本图书馆CIP数据核字(2003)第039045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: 朱 劼

北京中加印刷有限公司印刷·新华书店北京发行所发行

2003年7月第1版第1次印刷

787mm×1092mm 1/16·14.25印张

印数: 0 001-5 000册

定价: 29.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

软件工程是计算机应用科学中极为重要的一门学科。任何一个大中型信息系统的研制都离不开软件工程的指导。目前有关软件工程或信息系统工程的书籍很多，但这些书籍往往都是大部头，少则几十万字，多则上百万字。

当然，这些书籍之所以占用大量的篇幅是有充分理由的。因为软件工程要研究和处理的内容实在太多，所涉及的学科也不少，而软件工程的各个阶段都有许多问题需要研究。就系统需求分析来说，如何获得领域知识，如何与领域专家面谈，如何把握要点等就存在各种各样的方法。这些方法各有优缺点，适合不同的情况和不同的系统。如何对这些方法进行权衡，选择适合于要研制的系统的方法也需要进行很好的研究。而且，在获得领域知识后，如何表示也有诸多办法。在软件系统研制中如何保证质量？如何安排软件研制的进展？如何计算软件工作量？程序编制出来后，如何调试以保证其正确性？诸如此类的问题不胜枚举。可以说，针对软件工程的每个阶段都可以写出一本大部头的书籍。

但是，在目前这个快节奏的社会中，许多人往往没有时间读大部头的书籍。他们需要一本简明扼要，能很快掌握软件工程知识的书籍。他们学习软件工程的目的是不是做学术性的研究，而是要用于工程实践。本书正是适合这种目的的一本书籍。阅读本书，能够迅速对软件工程这门学科有一个总体性的了解。通过完成书中的习题能很快了解和掌握软件工程的大多数概念和方法，并在实际的信息系统研制中加以运用。

本书是作者集25年给研究生、本科生讲授软件工程的教学经验编写而成的，特点是简明扼要，内容全面，用很少的篇幅就介绍了软件工程的主要概念和方法，软件工程的历史研究情况、最新的进展和存在的问题。

本书可以作为研究生、本科生的软件工程课程的习题集。由于本书的理论知识浅显易懂，也可以作为急需了解软件工程知识的技术人员的入门书籍。

参加本书翻译工作的主要成员有：钟鸣、王君。全书由刘晓霞负责审校。同时担任部分翻译及校对工作的还有石永平、张文、田晓涛、梅刚、孙登峰、樊伟、文卫东等同志。

由于译者水平有限，难免有错误或不妥之处，敬请读者批评指正。

前 言

软件工程并不仅仅是技术和术语的综述，它包括了学生必须掌握的技术。本书供在校大学生和研究生学习软件工程使用。笔者一直教授大学的软件工程课程，在25年的教学过程中认识到，要想帮助学生掌握软件工程的技术，必须讲解样例并给予指导。

本书应该与软件工程的课本或讲义结合使用。本书不包括图表、符号及技术的基础知识和说明，仅介绍正确构造图表的规则。本书还给出了使用各项技术的说明，包括运用技术的规则。最重要的是，对于图表、符号及技术，给出了例题和详细的解答。

本书得以撰写成功并不完全源于个人努力。许多人对本书都有贡献。我特别想感谢下列人员：**Karen**，我的好妻子。在本书的创作过程中，她提供了大力的支持和帮助。没有她的帮助，本书不可能完成。**Steve**，他从自己的博士学习中抽时间对本书的许多章节提出了意见。我的学生们提供了撰写本书的最初灵感，并阅读了各章，发现了错误，提出了许多建议。我还想谢谢**Ramon**，是他建议撰写此书的，还要谢谢McGraw-Hill的编辑人员，感谢他们给予的帮助和提出的建议。

软件工程技术丛书书目

丛书
编号

英文书名

中文书名

作者

1	Object-Oriented and Classical Software Engineering, 5E	面向对象与传统软件工程(原书第5版)	Stephen R. Schach
1	Object-Oriented Software Engineering	面向对象的软件工程	Ian Sommerville
1	Software Engineering: A Practitioner's Approach, 5E	软件工程:实践者的研究方法(原书第5版)	Roger S. Pressman
1	Software Engineering, 6E	软件工程(原书第6版)	Ian Sommerville
1	Software Engineering with Java	软件工程:Java语言实现	Stephen R. Schach
1	Project-Based Software Engineering: An Object-Oriented Approach	基于项目的软件工程:面向对象研究方法	Evelyn Stiller
1.1.1	Software Process Improvement: Practical Guidelines for Business Success	软件过程改进	Sami Zahran
1.1.1	Making Process Improvement Work	软件过程改进简明实践	Neil S. Potter
1.1.2	The Road to the Unified Software Development Process	统一软件开发过程之路	Ivar Jacobson
1.1.2	The Unified Software Development Process	统一软件开发过程	Jacobson/Booch/Rumbaugh
1.1.2	The Rational Unified Process: An Introduction, 2E	统一过程引论(原书第2版)	Philippe Kruchten
1.1.2	UML and The Unified Process: Practical Object-Oriented Analysis & Design	UML和统一过程:实用面向对象的分析和设计	Jim Arlow
1.1.3	Managing Global Software Projects: How to Lead Geographically Distributed Teams, Manage Processes and Use Quality Models	国际化软件项目管理	Gopalaswamy Ramesh
1.1.3	Software Project Management: A Unified Framework	软件项目管理:一个统一的框架	Walker Royce
1.1.3	How to Run Successful Projects III: The Silver Bullet	成功的软件项目管理:银弹方案(原书第3版)	Fergus O'Connell
1.1.3	Successful Software Development, 2E	成功的软件开发(原书第2版)	Scott E. Donaldson
1.1.3	Six Sigma Software Development	6σ软件开发	Christine B. Tayntor
1.1.3	IT Project Management: On Track from Start to Finish	IT项目管理:从开始到结束的历程	Joseph Phillips
1.1.3	Successful IT Project Delivery: Learning the lessons of Project Failure	IT项目成功交付的秘诀	David Yardley
1.1.3	Software Project Management, 3E	软件项目管理(原书第3版)	Bob Hughes
1.1.3	Architecture-Centric Software Project Management	软件项目管理实用指南:以体系结构为中心	Daniel J. Paulish
1.1.4	Handbook of Software Quality Assurance, 3E	软件质量保证(原书第3版)	Gordon G. Schulmeyer
1.1.4	Software Reliability Engineering	软件可靠性工程	John Musa
1.1.4	Implementing ISO 9001:2000 The Journey from Conformance to Performance	ISO 9001:2000 实施指南	Tom Tairnubg
1.1.4	CMMI Distilled: A Practical Introduction to Integrated Process Improvement	CMMI精粹:集成化过程改进实用导论	Dennis M. Ahern
1.1.4	CMM Implementation Guide	CMM实施与软件过程改进	Kim Caputo
1.1.4	Implementing the Capability Maturity Model	CMM实施指南	James R. Persse
1.1.4	Object-Oriented Defect Management of Software	面向对象软件的差错管理	Houman Younessi
1.1.4	Metrics and Models in Software Quality Engineering	软件质量工程:度量与模型	Stephen H. Kan
1.1.4	Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software	软件性能工程	Connie U. Smith

丛书

编号

英文书名

中文书名

作者

1.1.4	Solid Software	Solid Software	Shari Pfleeger
1.1.4	Peer Reviews in Software: A Practical Guide	同级评审	Karl E. Wiegers
1.1.5	Practical Software Measurement	软件度量实用教程	John McGarry
1.1.5	Software Metrics: A Rigorous and Practical Approach, 2E	软件度量(原书第2版)	Norman E. Fenton
1.1.5	Software Assessments, Benchmarks, and Best Practices	软件评估、基准测试与最佳实践	Capers Jones
1.2.1	The Object Primer: The Application Developer's Guide to Object Orientation and the UML, 2E	面向对象软件开发教程(原书第2版)	Scott W. Ambler
1.2.1	UML and C++: A Practical Guide to Object-Oriented Development, 2E	C++面向对象开发(原书第2版)	Richard C. Lee
1.2.1	Object-Oriented Methods: Principles & Practices, 3E	面向对象方法:原理与实践(原书第3版)	Ian Graham
1.2.1	Principles of Object-Oriented Software Development, 2E	面向对象软件开发原理(原书第2版)	Anton Eliëns
1.2.1	Object Solutions: Managing the Object-Oriented Project	面向对象项目的解决方案	Grady Booch
1.2.1	An Introduction To Object-Oriented Programming, 3E	面向对象程序设计导引(原书第3版)	Timothy Budd
1.2.1	The Object Advantage: Business Process Reengineering with Object Technology	对象优势:采用对象技术的业务过程再工程	Ivar Jacobson
1.2.1	The Unified Modeling Language User Guide	UML用户指南	Booch/Rumbaugh/Jacobson
1.2.1	The Unified Modeling Language Reference Manual	UML参考手册	Rumbaugh/Jacobson/Booch
1.2.1	Applying UML and patterns: An Introduction to Object-Oriented Analysis and Design, 1E	UML和模式应用(原书第1版)	Craig Larman
1.2.1	Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, 2E	UML与模式应用(原书第2版)	Craig Larman
1.2.2	Software Reuse: Architecture, Process and Organization for Business Success	软件复用:结构、过程和组成	Ivar Jacobson
1.2.2	Software Reuse Techniques: Adding Reuse to the Systems Development Process	软件复用技术:在系统开发过程中考虑复用	Carma McClure
1.2.2	Practical Software Reuse: Strategies for Introducing Reuse Concepts in Your Organization	实用软件复用方法	Donald J. Reifer
1.2.2	Large-Scale Component-Based Development	大规模基于构件的软件开发	Alan W. Brown
1.2.2	Component-based Product Line Engineering with UML	基于构件的产品线工程:UML方法	Colin Atkinson
1.4.1	Object-Oriented Analysis and Design with Applications, 2E	面向对象的分析与设计:实例应用(原书第2版)	Grady Booch
1.4.1	Object-Oriented Analysis & Design	面向对象的分析与设计	Andrew Haigh
1.4.1	Analysis Patterns: Reusable Object Models	分析模式:可复用的对象模型	Martin Fowler
1.4.1	Requirements Analysis and System Design: Developing Information Systems with UML	需求分析与系统设计	Leeszek A. Maciaszek
1.4.1	Systems Analysis and Design in a Changing World	系统分析与设计	John W. Satzinger
1.4.1	Advanced Use Case Modeling, Vol I: Software Systems	高级用例建模 卷1: 软件系统	Frank Armour
1.4.1	Requirements Engineering: A Good Practice Guide	需求工程	Ian Sommerville
1.4.1	Software Requirements and Estimation	软件需求与估算	Swapna Kishore
1.4.1	Effective Requirements Practices	有效需求实践	Ralph R. Young
1.4.1	Applying Use Cases: A Practical Guide, 2E	用例分析技术(原书第2版)	Geri Schneider
1.4.1	Managing Software Requirements	软件需求管理:统一方法	Dean Leffingwell
1.4.1	Writing Effective Use Cases	编写有效用例	Alistair Cockburn
1.4.1	Problem Frames Analyzing and Structuring Software Development Problems	Problem Frames Analyzing and Structuring Software Development Problems	Michael Jackson

丛书

编号

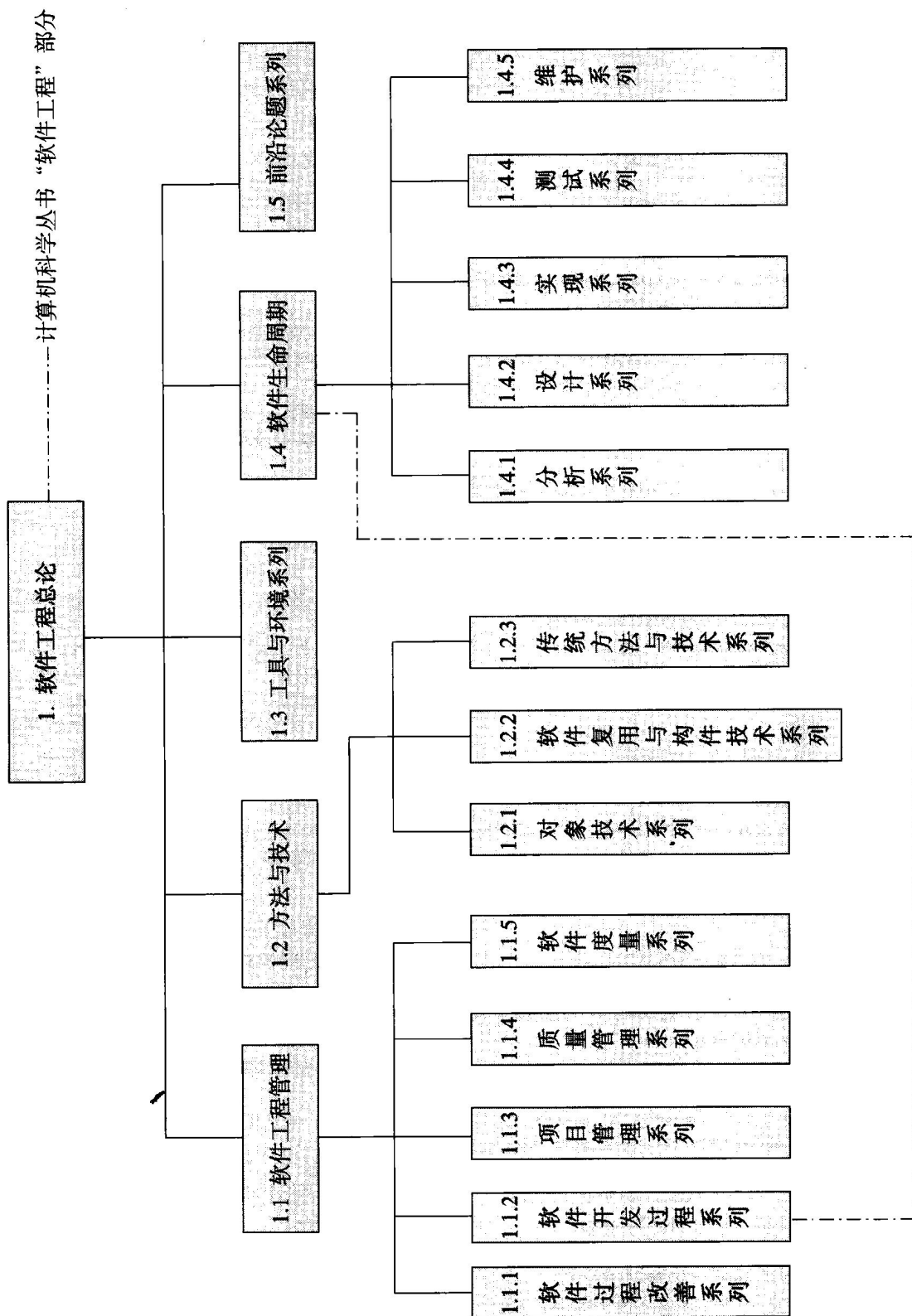
英文书名

中文书名

作者

1.4.2	Object-Oriented Software Construction, 2E	面向对象的软件结构(原书第2版)	Bertrand Meyer
1.4.2	Pattern-Oriented Software Architecture, Vol I: A System of Patterns	面向模式的软件体系结构 卷1:模式系统	Frank Buschmann
1.4.2	Pattern-Oriented Software Architecture, Vol II: Patterns for Concurrent and Networked Objects	面向模式的软件体系结构 卷2:用于并发和网络化对象的模式	Douglas Schmidt
1.4.2	Server Component Patterns	面向模式的软件体系结构 卷3:服务器组件模式	Markus Volter
1.4.2	DesignPatterns: Elements of Reusable Object-Oriented Software	设计模式:可复用面向对象软件的基础	Gamma/Helm/Johnson /Vlissides
1.4.2	Patterns of Enterprise Application Architecture	企业级应用体系结构模式	Martin Fowler
1.4.2	AntiPatterns and Patterns in Software Configuration Management	软件配置管理中的模式与反模式	William J. Brown
1.4.2	Software for Use: A Practical Guide to The Models and Methods of Usage-Centered Design	面向使用的软件设计	Larry L. Constantine
1.4.2	Software Architecture: Organizational Principles and Patterns	软件架构:组织原则与模式	David Dikel
1.4.2	The UML Profile for Framework Architectures	框架体系结构的UML档案	Marcus Fontoura
1.4.2	Software Architect's Profession: An Introduction	软件架构师入门必读	Marc Sewell
1.4.2	Business Modeling with UML: Business Patterns at work	UML业务建模:实用业务模式	Hans-Erik Eriksson
1.4.3	Software Fabrication:Automating Application Development	软件构造:自动化软件开发	Jack Greenfield
1.4.3	Building J2EE Applications With The Rational Unified Process	用RUP构建J2EE 应用程序	Peter Eeles
1.4.3	Programming from Specifications	从规范出发的程序设计	Carroll Morgan
1.4.4	Testing IT: An Off-the-Shelf SoftwareTesting Process Handbook	实用软件测试过程之路	John Watkins
1.4.4	Lessons Learned in Software Testing	软件测试经验与教训	Cem Kaner
1.4.4	Testing Computer Software: The Bestselling Software Testing Book Of All Time, 2E	计算机软件测试(原书第2版)	Cem Kaner
1.4.4	Software Testing in the Real World: Improving the Process	软件测试过程改进	Edward Kit
1.4.4	Effective Methods for Software Testing, 2E	有效的软件测试方法(原书第2版)	William E. Perry
1.4.4	Beta Testing for Better Software	软件Beta测试	Michael R. Fine
1.4.4	A Practical Guide to Testing Object-Oriented Software	面向对象的软件测试	John D. McGregor
1.4.4	Managing the Testing Process, 2E	测试过程管理(原书第2版)	Rex Black
1.4.4	Software Testing: A Craftsman's Approach, 2E	软件测试:工艺师的方法(原书第2版)	Paul C. Jorgensen
1.4.4	Just Enough Software Test Automation	软件测试自动化	Daniel J. Mosley
1.4.4	The Craft of Software Testing: Subsystem Testing Including Object-Based and Object-Oriented Testing	软件测试实用技术	Brian Marick
1.5	JAVA Tools for Extreme Programming: Mastering Open Source Tools, Including Ant,JUnit, and Cactus	极限编程的JAVA工具	Richard Hightower
1.5	Agile Software Development Ecosystems	敏捷软件开发生态系统	Tom DeMarco
1.5	Agile Modeling: Effective Practices For eXtreme Programming and The Unified Process	敏捷建模:极限编程和统一过程的有效实践	Scott W. Ambler
1.5	Model-Driven Development: Automating Component Design, Implementation, and Assembly	模式驱动的软件开发	David Frankel
1.5	A Practical Guide to Feature-Driven Development	特征驱动开发方法:原理与实践	Steve R. Palmer

软件工程技术丛书结构图



目 录

译者序

前言

第1章 软件生命周期1

1.1 概述.....1

1.1.1 软件生命周期活动的类型1

1.1.2 典型文档2

1.2 软件生命周期模型.....2

1.2.1 线性模型2

1.2.2 原型实现模型3

1.2.3 增量模型3

1.2.4 Boehm的螺旋模型3

习题.....3

习题答案.....4

第2章 软件过程模型和其他模型7

2.1 软件过程模型.....7

2.2 数据流程图.....8

2.3 petri网模型9

2.4 对象模型10

2.4.1 存在依赖.....11

2.4.2 实例图.....13

2.5 用例图13

2.6 场景14

2.7 时序图14

2.8 层次结构图15

2.9 控制流程图15

2.10 状态图.....16

2.11 网络模型.....18

习题18

补充问题19

习题答案21

补充问题答案23

第3章 软件项目管理29

3.1 概述29

3.2 管理方法29

3.3 小组方法29

3.4 重要准则30

3.5 能力成熟度模型32

3.6 个人的软件过程33

3.7 实现值分析33

3.7.1 基本的量.....33

3.7.2 进度指示器.....34

3.8 错误跟踪35

3.9 事后回顾36

习题37

补充问题37

习题答案39

补充问题答案40

第4章 软件项目计划45

4.1 项目计划45

4.2 WBS: 工作分解结构45

4.3 PERT: 程序评估和评审技术48

4.3.1 完成时间的算法.....49

4.3.2 关键路径.....50

4.3.3 构造关键路径的算法.....50

4.3.4 宽松时间.....51

4.3.5 宽松时间的算法.....51

4.4 软件成本估算52

4.4.1 代码行估算.....52

4.4.2 基于LOC的成本估算.....53

4.4.3 成本构成模型.....54

4.4.4 功能点分析.....55

4.4.5 生产率.....57

4.4.6 判定估算.....57

4.4.7 自动估算工具.....58

习题58

补充问题59

习题答案	61	7.5 SQA计划的IEEE标准	98
补充问题答案	63	习题	101
第5章 软件度量	69	补充问题	101
5.1 概述	69	习题答案	102
5.2 软件度量理论	70	补充问题答案	103
5.2.1 单调性	71	第8章 需求	111
5.2.2 度量尺度	71	8.1 概述	111
5.2.3 统计	72	8.2 对象模型	111
5.3 产品度量	72	8.3 数据流程建模	112
5.3.1 McCabe的环数	73	8.4 行为建模	112
5.3.2 Halstead的软件科学	75	8.4.1 用例	113
5.3.3 Henry-Kafura信息流	79	8.4.2 场景	114
5.4 过程度量	80	8.4.3 状态图	114
5.5 GQM方法	80	8.5 数据字典	115
习题	80	8.6 系统图	116
补充问题	81	8.7 软件需求规格说明的IEEE标准	117
习题答案	82	习题	118
补充问题答案	83	补充问题	119
第6章 风险分析和管理	89	习题答案	119
6.1 概述	89	补充问题答案	120
6.2 风险确定	89	第9章 软件设计	127
6.3 风险估算	89	9.1 概述	127
6.4 风险揭露	90	9.2 设计过程的各个阶段	128
6.5 风险缓解	90	9.3 设计概念	130
6.6 风险管理计划	91	9.4 度量内聚性	132
习题	92	9.4.1 程序片	132
补充问题	92	9.4.2 粘合权标	133
习题答案	93	9.5 度量耦合性	135
补充问题答案	93	9.6 需求的可溯性	136
第7章 软件质量保证	95	习题	139
7.1 概述	95	补充问题	140
7.2 形式化检查和技术评审	95	习题答案	140
7.2.1 检查的角色	96	补充问题答案	142
7.2.2 检查的步骤	96	第10章 软件测试	145
7.2.3 检查表	96	10.1 概述	145
7.3 软件的可靠性	97	10.2 软件测试基础知识	145
7.3.1 错误率	97	10.3 测试覆盖准则	146
7.3.2 概率论	98	10.3.1 包含	146
7.4 统计质量保证	98	10.3.2 功能测试	146

10.3.3 测试度量	147	12.2.5 度量5: 类的响应	183
10.3.4 结构测试	148	12.2.6 度量6: 方法缺乏内聚力	183
10.4 数据流测试	152	12.3 MOOD度量	187
10.5 随机测试	154	12.3.1 封装	187
10.5.1 操作预置文件	154	12.3.2 继承因子	188
10.5.2 测试的统计推断	155	12.3.3 耦合因子	189
10.6 边界测试	155	12.3.4 多态性因子	190
习题	156	习题	191
补充问题	157	补充问题	191
习题答案	159	习题答案	192
补充问题答案	160	补充问题答案	193
第11章 面向对象的软件开发	167	第13章 面向对象的测试	197
11.1 概述	167	13.1 概述	197
11.1.1 继承	167	13.2 MM测试	197
11.1.2 多态性	168	13.3 函数对的覆盖	199
11.2 确定对象	169	习题	203
11.2.1 noun-in-text方法	169	补充问题	203
11.2.2 确定继承	171	习题答案	204
11.2.3 确定重用	172	补充问题答案	204
11.2.4 用例方法	173	第14章 形式化表示方法	207
11.3 确定关联	173	14.1 概述	207
11.4 确定多重性	175	14.2 形式化的规格说明	207
习题	177	14.2.1 前提条件	208
补充问题	177	14.2.2 后置条件	208
习题答案	178	14.2.3 不变式	208
补充问题答案	178	14.3 对象约束语言	208
第12章 面向对象的度量	181	14.3.1 导航	209
12.1 概述	181	14.3.2 不变式	210
12.1.1 传统的度量	181	14.3.3 属性	210
12.1.2 面向对象的抽象	181	14.3.4 预定义操作	210
12.2 面向对象设计的度量套件	182	14.3.5 前提条件和后置条件	211
12.2.1 度量1: 每个类的加权方法	182	习题	211
12.2.2 度量2: 继承树的深度	182	补充问题	211
12.2.3 度量3: 孩子的数目	182	习题答案	213
12.2.4 度量4: 对象类之间的耦合	182	补充问题答案	214

第1章 软件生命周期

1.1 概述

软件生命周期是在软件开发中发生的不同活动的序列。在开发过程中会产生不同的交付内容。虽然交付内容可以是协议或评价，但交付内容一般是实物，如源代码或用户手册。通常，活动和交付是密切相关的。里程碑（milestone）是用来说明项目状态的事件。例如，完成用户手册的事件可以是一个里程碑。为了便于管理，里程碑很关键，因为里程碑的完成使管理人员能评价软件开发的进程。

1.1.1 软件生命周期活动的类型

- 1.1.1.1 可行性——确定所建议的开发是否有价值。
 - 市场分析——确定本产品是否有潜在的市场。
- 1.1.1.2 需求——确定软件应该包括什么功能。
 - 需求引入——从用户处获得需求。
 - 领域分析——确定本问题中最常见的任务及结构。
- 1.1.1.3 项目计划——确定怎样开发软件。
 - 成本分析——确定成本估算。
 - 时间规划——建立开发时间表。
 - 软件质量保证——确定有助于保证产品质量的活动。
 - 工作分步结构——确定开发产品所必需的子任务。
- 1.1.1.4 设计——确定软件应该如何提供功能。
 - 结构设计——设计软件系统的结构。
 - 接口设计——指定系统各部件之间的接口。
 - 详细设计——设计用于软件系统各个部件的算法。
- 1.1.1.5 实现——建造软件。
- 1.1.1.6 测试——用数据来运行软件以帮助确保软件正确工作。
 - 单元测试——由原开发人员进行的测试。
 - 集成测试——在软件的集成中进行的测试。
 - 系统测试——在一个与运行环境相匹配的环境中对软件进行的测试。
 - 阿尔法测试——顾客在开发人员处对软件进行的测试。
 - 贝塔测试——在顾客处对软件进行的测试。
 - 接收测试——满足买方需求的测试。

回归测试——对以前的版本进行测试以确保新版本仍然保持以前的功能。

1.1.1.7 交付——给顾客提供一个有效的软件解决方案。

安装——使软件在顾客处可以使用。

培训——教用户使用软件。

帮助桌面——回答用户的问题。

1.1.1.8 维护——更新和改进软件以确保软件继续有用。

1.1.2 典型文档

1.1.2.1 工作陈述——所需功能的初步描述，通常由用户提供。

1.1.2.2 软件需求规格说明——描述最终的软件能做什么工作。

对象模型——说明主要的对象/类。

用例场景——从用户的角度说明可能的一系列行为。

1.1.2.3 项目进度——描述任务的顺序以及估计的时间和工作。

1.1.2.4 软件测试计划——描述怎样测试软件以确保软件行为正确。

接收测试——顾客为确定系统的可接受性而设计的测试。

1.1.2.5 软件设计——描述软件的结构。

体系结构设计——具备互连性的高级结构。

详细设计——低级模块或对象的设计。

1.1.2.6 软件质量保证计划（SQA计划）——描述用来保证质量的活动。

1.1.2.7 用户手册——描述怎样使用最终的软件。

1.1.2.8 源代码——实际的产品代码。

1.1.2.9 测试报告——描述进行了什么测试以及系统运转情况如何。

1.1.2.10 缺陷报告——描述顾客对特定系统行为不满意的地方，一般是指软件故障或错误。

1.2 软件生命周期模型

以下各节给出了四种最常见的软件生命周期模型。

1.2.1 线性模型

这个模型也称为瀑布模型（如图1-1所示），因为其图形像瀑布而得名。此模型由Royce在1970年首先提出，它是第一个描述了任务的标准序列。

瀑布模型有许多版本。虽然某些开发任务在每个开发中几乎都会出现，但划分开发阶段的方法有许多种。请注意，在图1-1所示的瀑布模型中，项目计划活动包含在需求阶段中，且交付及维护阶段也省略了。

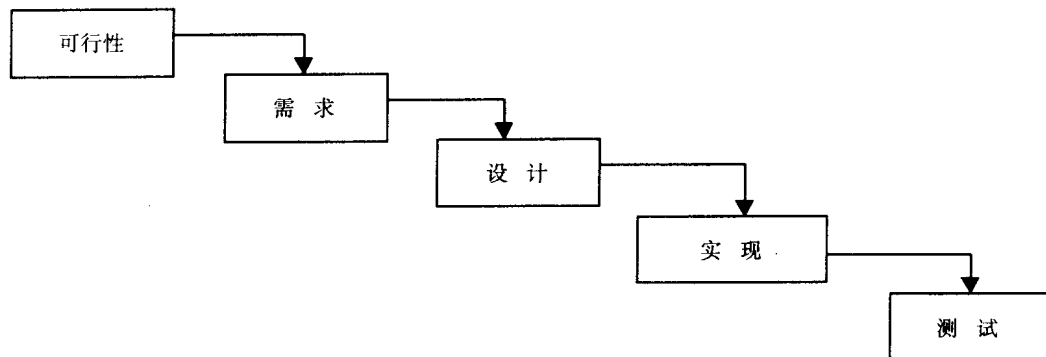


图1-1 瀑布模型

1.2.2 原型实现模型

软件生命周期模型往往先建立一个临时性的版本（原型）。这个原型主要用于测试概念和需求。此原型用来向顾客演示建议的行为。在取得顾客的认可后，相应的软件开发通常遵循与线性模型开发相同的阶段。这样做可避免开发不必要的功能。

1.2.3 增量模型

增量模型是由D. L. Parnas提出的（D.Parnas. “Designing Software for Ease of Extension and Contraction.” *IEEE Transactions on Software Engineering (TOSE)* 5:3, March 1979, 128-138）。其目标是设计和向用户提交整个系统的一个最小子集，而且此子集仍然是一个有用的系统。这个设计和提交的过程在整个生命周期中以最小的额外增量重复进行。此模型的优点可以给用户提供一个早期的工作系统以及工作增量。

1.2.4 Boehm的螺旋模型

B. Boehm引入了螺旋模型（B.Boehm. “A Spiral Model for Software Development and Enhancement.” *IEEE Computer*. 21:5, May 1988, 61-72）。此模型的图像是一个螺旋结构，此结构从中间开始，不断地重新访问客户沟通（面谈等）、规划、风险分析、工程、构造和释放以及客户评估等基本任务。

习题

1. 阶段性的生命周期模型是怎样支持软件管理的？
2. 里程碑的两个必需的特征是什么？
3. 指出下列文档是在软件生命周期的哪个阶段产生的：最终用户手册、体系结构设计、SQA计划、模块规格说明、源代码、工作陈述、测试计划、初级用户手册、详细设计、成本

估计、项目计划、测试报告、文档。

4. 指出瀑布模型中下列任务的顺序：接收测试、项目计划、单元测试、需求评审、成本估计、高级设计、市场分析、初级设计、系统测试、设计评审、实现、需求规格说明。

5. 画出一个表现循环生命周期模型的图表。

习题答案

1. 阶段性的生命周期模型是怎样支持软件管理的？

阶段性的生命周期提高了项目的可视性。项目可利用各个阶段作为里程碑来进行管理。更详细地划分阶段能够更密切地监控项目的进展。

2. 里程碑的两个必需的特征是什么？

里程碑必须与软件开发的进展相关联且必须在完成时非常明显。

3. 指出下列文档是在软件生命周期中的哪个阶段产生的？

最终用户手册	实现阶段
体系结构设计	设计阶段
SQA计划	项目计划阶段
模块规格说明	设计阶段
源代码	实现阶段
工作陈述	可行性阶段
测试计划	需求阶段
初级用户手册	需求阶段
详细设计	设计阶段
成本估计	项目计划阶段
项目计划	项目计划阶段
测试报告	测试阶段
文档	实现阶段

4. 各任务的顺序如下：

市场分析

项目计划、成本估计、需求规格说明（可同时完成）

需求评审

高级设计

初级设计

设计评审

实现

单元测试

系统测试

接收测试

5. 画出一个表现循环生命周期模型的图表。请参考图1-2。

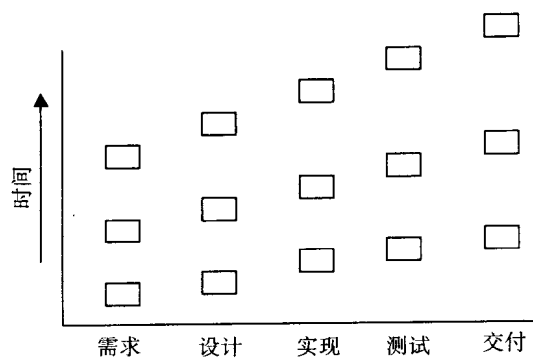


图1-2 循环生命周期模型