



“十二五”普通高等教育本科国家级规划教材



北京高等教育精品教材

BEIJING GAODENG JIAOYU JINGPIN JIAOCAI

全国优秀畅销书

全国高校出版社优秀畅销书

21世纪软件工程专业规划教材

软件工程导论 (第6版) 学习辅导

张海藩 牟永敏 编著

10010101000101

111010010010

10010101000101

111010010010

1000101

11101001

10010101000101

111010010010



清华大学出版社



“十二五”普通高等教育本科国家级规划教材

TP311.5/4=5C



北京高等教育精品教材
BEIJING GAODENG JIAOYU JINGPIN JIAOCAI

2013

全国优秀畅销书
全国高校出版社优秀畅销书

21世纪软件工程专业规划教材

软件工程导论 (第6版) 学习辅导

张海藩 牟永敏 编著

北方工业大学图书馆



C00338993



清华大学出版社
北京

内 容 简 介

为配合《软件工程导论(第6版)》的出版,作者对《软件工程导论(第5版)学习辅导》作了适当修改,编写了《软件工程导论(第6版)学习辅导》。

本书正文共10章,每章均由3部分内容组成:第1部分简明扼要地复习本单元的重点内容;第2部分给出与本单元内容密切相关的习题;第3部分是习题解答,对典型习题的解答不是简单地给出答案,而是仔细分析题目,讲解解题思路,从而帮助读者举一反三,学会用软件工程方法学分析问题、解决问题。

正文后面有两个附录,分别给出了模拟试题和模拟试题参考答案。读者可以用这些试题自我测试,检验学习效果。

本书可以与《软件工程导论(第6版)》配合使用,也可供学习软件工程课程的读者单独使用,以加深对所学内容的理解并检测学习效果。

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

软件工程导论(第6版)学习辅导/张海藩、牟永敏编著. —北京:清华大学出版社,2013
21世纪软件工程专业规划教材
ISBN 978-7-302-33099-8

I. ①软… II. ①张… ②牟… III. 软件工程—高等学校—教学参考资料 IV. TP311.5

中国版本图书馆CIP数据核字(2013)第150347号

责任编辑:袁勤勇

封面设计:常雪影

责任校对:白蕾

责任印制:杨艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

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

印 装 者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:15.25

字 数:344千字

版 次:2013年8月第1版

印 次:2013年8月第1次印刷

印 数:1~3000

定 价:28.00元

产品编号:050165-01

前言

PREFACE

《软件工程导论》已经出了 5 个版本，颇受读者欢迎，先后被评为全国高校出版社优秀畅销书、全国优秀畅销书（前 10 名）、北京高等教育精品教材、“十二五”普通高等教育本科国家级规划教材。国内许多高校用它作为软件工程课的教材，累计销售达到 130 万册。

为配合《软件工程导论（第 6 版）》的出版，作者在针对《软件工程导论（第 5 版）》所编写的《软件工程导论（第 5 版）学习辅导》的基础上作了适当修改。牟永敏教授根据多年的教学和科研经验，对书中面向过程部分的内容进行了适量删减，同时，为了加强软件工程的实践教学，增加了面向对象设计实现服务的方法等方面的练习，并且给出了与上述内容密切配合的习题和习题解答。

本书正文共 10 章。第 1 章“软件工程概论”，涵盖教材（第 6 版，下同）第 1 章的重点内容；第 2 章“结构化分析”，涵盖教材第 2、3、4 章的重点内容；第 3 章“结构化设计”，涵盖教材第 5、6 章的重点内容；第 4 章“结构化实现”，涵盖教材第 7 章的重点内容；第 5 章“维护”，涵盖教材第 8 章的重点内容；第 6 章“面向对象方法学引论”，涵盖教材第 9 章的重点内容；第 7 章“面向对象分析”，涵盖教材第 10 章的重点内容；第 8 章“面向对象设计”，涵盖教材第 11 章的重点内容；第 9 章“面向对象实现”，涵盖教材第 12 章的重点内容；第 10 章“软件项目管理”，涵盖教材第 13 章的重点内容。

每章均由 3 部分内容组成：第 1 部分简明扼要地复习本单元的重点内容；第 2 部分给出与本单元内容密切相关的习题，其中一些题目与教材上的题目相同，另一些题目是教材上没有的，当然，也有一些教材上的题目没有包含在本书中，可作为软件工程课的练习题，留给读者独立完成；第 3 部分是习题解答，对典型习题的解答不是简单地给出答案，而是仔细分析题目，讲解解题思路，从而有助于读者举一反三，学会用软件工程方法学分析问题和解决问题。

正文后面有两个附录:附录A是模拟试题,共给出3份试卷;附录B是模拟试题参考答案。读者可以用这些试题自我测试,检验学习效果。

全书由张海藩统一定稿。

丁媛、刘梦婷、刘昂、李慧丽、张亚楠等同学对第6版增加的内容进行了测试,并提出了有益的建议,谨在此表示感谢。

编者

2013年5月

目 录

CONTENTS

第 1 章 软件工程概论	1
1.1 软件危机	1
1.1.1 软件危机简介	1
1.1.2 产生软件危机的原因	1
1.1.3 消除软件危机的途径	2
1.2 软件工程	3
1.2.1 软件工程简介	3
1.2.2 软件工程的基本原理	3
1.2.3 软件工程方法学	3
1.3 软件生命周期	4
1.4 软件过程	5
习题	11
习题解答	13
第 2 章 结构化分析	17
2.1 可行性研究的目的	17
2.2 可行性研究过程	17
2.3 需求分析的任务	18
2.4 与用户沟通的方法	18
2.5 分析建模与规格说明	19
2.6 实体-联系图	20
2.7 数据流图	20
2.8 数据字典	21
2.9 状态转换图	21
2.10 其他图形工具	22
2.11 验证软件需求	23

2.12	成本/效益分析	23
2.12.1	成本估计	23
2.12.2	成本/效益分析方法	23
2.13	形式化说明技术	24
	习题	25
	习题解答	27
第3章	结构化设计	45
3.1	软件设计的任务	45
3.1.1	概要设计	45
3.1.2	详细设计	45
3.2	分析与设计的关系	46
3.3	设计原理	47
3.3.1	模块化与模块独立	47
3.3.2	抽象	47
3.3.3	逐步求精	48
3.3.4	信息隐藏	49
3.3.5	局部化	49
3.4	度量模块独立性的标准	49
3.4.1	内聚	49
3.4.2	耦合	49
3.5	启发规则	50
3.6	描绘软件结构的图形工具	50
3.7	面向数据流的设计方法	50
3.7.1	数据流的类型	51
3.7.2	设计步骤	51
3.8	人机界面设计	52
3.8.1	应该考虑的设计问题	53
3.8.2	人机界面设计过程	54
3.8.3	人机界面设计指南	54
3.9	过程设计	55
3.10	过程设计的工具	55
3.11	面向数据结构的设计方法	57
3.12	程序复杂程度的定量度量	58
3.12.1	McCabe方法	58
3.12.2	Halstead方法	58
	习题	59
	习题解答	63

第 4 章 结构化实现	83
4.1 编码	83
4.1.1 选择程序设计语言	83
4.1.2 编码风格	84
4.2 软件测试基础	84
4.3 单元测试	85
4.4 集成测试	86
4.5 白盒测试技术	88
4.5.1 逻辑覆盖	88
4.5.2 控制结构测试	89
4.6 黑盒测试技术	90
4.6.1 等价划分	90
4.6.2 边界值分析	91
4.6.3 错误推测	92
4.7 调试	92
4.7.1 调试过程	92
4.7.2 调试途径	92
4.8 软件可靠性	93
4.8.1 基本概念	93
4.8.2 估算平均无故障时间的方法	94
习题	96
习题解答	98
第 5 章 维护	109
5.1 软件维护的定义	109
5.2 软件维护的特点	110
5.3 软件维护过程	111
5.4 软件的可维护性	112
5.4.1 决定软件可维护性的因素	112
5.4.2 文档	113
5.4.3 可维护性复审	113
5.5 预防性维护	114
5.6 软件再工程过程	114
习题	116
习题解答	117
第 6 章 面向对象方法学引论	121
6.1 面向对象方法学概述	121

88	6.1.1 面向对象方法学的要点	121
88	6.1.2 面向对象方法学的优点	122
88	6.1.3 面向对象的软件过程	124
88	6.2 面向对象的概念	125
88	6.2.1 对象	125
88	6.2.2 其他概念	126
88	6.3 面向对象建模	127
88	6.4 对象模型	128
88	6.4.1 类图的基本符号	128
88	6.4.2 表示关系的符号	129
88	6.5 动态模型	130
88	6.6 功能模型	130
88	6.6.1 用例图	131
88	6.6.2 用例建模	132
88	6.7 三种模型之间的关系	133
88	习题	133
88	习题解答	134
88	第7章 面向对象分析	141
88	7.1 面向对象分析的基本过程	141
88	7.1.1 概述	141
88	7.1.2 3个子模型与5个层次	142
88	7.2 需求陈述	142
88	7.3 建立对象模型	143
88	7.3.1 确定类与对象	143
88	7.3.2 确定关联	144
88	7.3.3 划分主题	145
88	7.3.4 确定属性	145
88	7.3.5 识别继承关系	145
88	7.3.6 反复修改	146
88	7.4 建立动态模型	146
88	7.4.1 编写脚本	146
88	7.4.2 画事件跟踪图	147
88	7.4.3 画状态图	147
88	7.4.4 审查动态模型	148
88	7.5 建立功能模型	148
88	7.6 定义服务	148
88	习题	149

习题解答	150
第 8 章 面向对象设计	155
8.1 面向对象设计的准则	155
8.2 启发规则	156
8.3 软件重用	157
8.3.1 概述	157
8.3.2 类构件	158
8.3.3 软件重用的效益	159
8.4 系统分解	159
8.5 设计问题域子系统	160
8.6 设计人机交互子系统	161
8.7 设计任务管理子系统	161
8.8 设计数据管理子系统	162
8.9 设计类中的服务	163
8.10 设计关联	163
8.11 设计优化	164
习题	165
习题解答	166
第 9 章 面向对象实现	175
9.1 程序设计语言	175
9.2 程序设计风格	176
9.3 面向对象的测试策略	177
9.4 设计测试用例	178
9.4.1 测试类的技术	178
9.4.2 集成测试技术	179
习题	179
习题解答	180
第 10 章 软件项目管理	197
10.1 估算软件规模	197
10.1.1 代码行技术	197
10.1.2 功能点技术	197
10.2 估算工作量	199
10.2.1 静态单变量模型	199
10.2.2 动态多变量模型	199
10.2.3 COCOMO2 模型	199

10.3	进度计划	200
10.3.1	估算开发时间	200
10.3.2	Gantt图	201
10.3.3	工程网络	201
10.3.4	估算工程进度	201
10.3.5	关键路径	202
10.3.6	机动时间	202
10.4	人员组织	202
10.4.1	民主制程序员组	203
10.4.2	主程序员组	203
10.4.3	现代程序员组	203
10.5	质量保证	204
10.5.1	软件质量	204
10.5.2	软件质量保证措施	204
10.6	软件配置管理	204
10.6.1	软件配置	204
10.6.2	软件配置管理过程	205
10.7	能力成熟度模型	206
	习题	207
	习题解答	209
	附录	215
附录 A	模拟试题	215
	试卷一	215
	试卷二	217
	试卷三	219
附录 B	模拟试题参考答案	222
	试卷一参考答案	222
	试卷二参考答案	224
	试卷三参考答案	226
	参考文献	229

第 1 章

CHAPTER

软件工程概论

因素致主 35

1.1 软件危机

为吸取历史经验教训,应该认真研究产生软件危机的原因,探讨消除软件危机的途径。

1.1.1 软件危机简介

通常把在计算机软件的开发与维护过程中所遇到的一系列严重问题笼统地称为软件危机。这些问题绝不仅仅是不能正常运行的软件才具有的,实际上,几乎所有软件都不同程度地存在这些问题。

概括地说,软件危机包含以下两个方面的问题。

- (1) 如何开发软件,以满足社会对软件日益增长的需求。
- (2) 如何更有效地维护数量不断膨胀的已有软件。

具体地说,软件危机主要有以下一些典型表现。

- 对软件开发成本和进度的估计常常很不准确。
- 经常出现用户对“已完成的”软件产品不满意的情况。
- 软件产品的质量往往达不到要求。
- 软件通常是很难维护的。
- 软件往往没有适当的文档资料。
- 软件成本在计算机系统总成本中所占的比例逐年上升。
- 软件开发生产率提高的速度远远不能满足社会对软件产品日益增长的需求。

鉴于软件危机周期长且难于预测,因此把它称为“软件萧条”或“软件困扰”可能更恰当一些。

1.1.2 产生软件危机的原因

1. 客观原因

软件是计算机系统逻辑部件而不是物理部件,其显著特点是缺乏

“可见性”,因此,管理和控制软件开发过程相当困难。此外,软件维护通常意味着改正或修改原有的设计,从而使得软件较难维护。

软件的另一个突出特点是规模庞大,而程序复杂性将随着程序规模增加以指数速度上升。软件可能具有的状态数通常都是天文数字,无法完全预见软件可能遇到的每一种情况。

2. 主观原因

在计算机系统发展的早期阶段,开发软件的个性化特点使得许多软件工程师对软件开发和维护有不少糊涂认识,在实际工作中或多或少地采用了错误的方法,这是使软件问题发展成软件危机的主要原因。

错误的认识和做法主要表现为:忽视软件需求分析的重要性;认为软件开发就是写程序;轻视软件维护。

事实上,对用户的需求没有完整准确的认识就匆忙着手编写程序,是许多软件开发工程失败的主要原因之一。

必须认识到,软件开发和维护要经历一个漫长的时期(称为软件生命周期),编写程序只是软件开发过程中的一个相对来说比较次要的阶段。

另一方面还必须认识到,程序只是完整的软件产品的一个组成部分,一个软件产品必须由一个完整的配置组成。软件配置主要包括程序、文档和数据等成分。

严酷的事实是,在软件开发的后期阶段引入一个变动比在早期引入同一个变动所需付出的代价高几百倍甚至上千倍。所谓软件维护,就是在软件开发工作已经结束之后在使用现场对软件进行修改。因此,维护是极端艰巨复杂的工作,需要花费很大代价。由此可见,轻视软件维护是一个最大的错误。软件工程的一个重要目标就是提高软件的可维护性,减少软件维护的代价。

1.1.3 消除软件危机的途径

首先应该树立对计算机软件的正确认识。软件是程序、数据及文档的完整集合,其中,程序是能够完成预定功能和性能的可执行的指令序列;数据是使程序能够适当地处理信息的数据结构;文档是开发、使用和维护程序所需要的图文资料。

软件开发应该是组织良好、管理严密、各类人员团结协作共同完成的工程项目,必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法,并研究能更有效地开发软件的技术和方法。

应该积极开发和使用计算机辅助软件工程(CASE)工具。

总之,为了消除软件危机,既要有技术措施(方法和工具),又要有必要的组织管理措施。软件工程正是从技术和管理两个方面研究如何更好地开发和维护软件的一门新兴的工程学科。

1.2 软件工程

1.2.1 软件工程简介

软件工程是指导计算机软件开发和维护的一门工程学科,该学科的目的是生产出能按期交付的、在预算范围内的、满足用户需求的、质量合格的软件产品。

软件工程具有以下本质特性。

- 软件工程关注于大型程序的构造。
- 软件工程的中心课题是控制复杂性。
- 软件产品交付使用后仍然需要经常修改。
- 开发软件的效率非常重要。
- 开发人员和谐地合作是成功开发软件的关键。
- 软件必须有效地支持它的用户。
- 在软件工程领域中通常由具有一种文化背景的人替具有另一种文化背景的人开发产品。

1.2.2 软件工程的基本原理

- 用分阶段的生命周期计划严格管理。
- 坚持进行阶段评审。
- 实行严格的产品控制。
- 采用现代程序设计技术。
- 结果应能清楚地审查。
- 开发小组的人员应该少而精。
- 承认不断改进软件工程实践的必要性。

1.2.3 软件工程方法学

通常把在软件生命周期全过程中使用的一整套技术方法的集合称为方法学,也称为范型。

软件工程方法学包含3个要素:方法、工具和过程。其中,方法是完成软件开发各项任务的技术方法,回答“怎样做”的问题;工具是为运用方法而提供自动的或半自动的软件工程支撑环境;过程是为了获得高质量软件所需要完成的一系列任务的框架,它规定了完成各项任务的工作步骤,回答“何时做”的问题。

目前使用得最广泛的软件工程方法学分别是传统方法学和面向对象方法学。

1. 传统方法学(结构化范型)

(1) 采用结构化技术(结构化分析、结构化设计和结构化实现)完成软件开发的各项任务。

(2) 把软件生命周期划分成若干个阶段,然后顺序完成各个阶段的任务。

(3) 每个阶段的开始和结束都有严格的标准,对于任何两个相邻的阶段而言,前一阶段的结束标准就是后一阶段的开始标准。

(4) 在每个阶段结束之前都必须正式地进行严格的技术审查和管理复审。

2. 面向对象方法学(面向对象范型)

(1) 把对象作为融合了数据及在数据上操作的软件构件。也就是说,用对象分解取代了传统方法的功能分解。

(2) 把所有对象都划分成类。

(3) 按照父类与子类的关系,把若干个相关类组织成一个层次结构的系统。

(4) 对象彼此间仅能通过发送消息互相联系。

使用结构化范型开发出的软件,在本质上是一个单元,这是用结构化范型开发大型软件产品时不甚成功的一个重要原因。相反,当正确地使用面向对象范型时,开发出的软件产品是由许多小的、相对独立的单元(对象)组成的。因此,面向对象范型降低了软件产品的复杂度,从而简化了软件开发与维护工作。

1.3 软件生命周期

概括地说,软件生命周期由软件定义、软件开发和运行维护(也称为软件维护)3个时期组成,通常把前两个时期再进一步划分成若干个阶段。

软件定义时期的基本任务是:确定软件开发工程的总目标;研究该项目的可行性;分析确定客户对软件产品的需求;估算完成该项目所需的资源和成本,并且制定工程进度表。这个时期的工作称为系统分析,由系统分析员负责完成。

通常把软件定义时期进一步划分成问题定义、可行性研究和需求分析3个阶段。其中需求分析阶段应该完成的工作包括需求获取和需求分析两部分。揭示客户需求的过程称为需求获取或需求收集;一旦确定了最初的一系列需求,就应该进一步提炼和扩展这些需求,并用软件需求规格说明书把客户需求准确地记录下来,这个过程称为需求分析。

软件开发时期具体设计和实现在前一个时期定义的软件,它通常由下述4个阶段组成:总体设计(又称为结构设计)、详细设计、编码和单元测试、综合测试。其中前两个阶段称为系统设计,后两个阶段称为系统实现。

运行维护时期的主要任务是通过已交付使用的软件做必要的修改,使软件持久地满足客户的需求。具体地说,当软件在使用过程中发现错误时应该加以改正;当环境改变时应该修改软件以适应新的环境;当用户有新要求时应及时改进或扩充软件以满足用户的新需要。通常对维护时期不再进一步划分阶段,但是每一次维护活动本质上都是一次压缩和简化了的定义和开发过程。

使用结构化范型开发软件时,软件生命周期各阶段中使用的概念及应完成的任务性质显著不同。需求分析阶段的基本任务是确定软件必须“做什么”,使用的概念主要是“功

能”。设计阶段的任务是确定“怎样做”，其中结构设计的任务是把软件分解成不同的模块，使用的概念是“模块”；详细设计的任务是设计实现每个模块所需要的数据结构和算法，使用的概念是“数据结构”和“算法”。

使用面向对象范型开发软件的基本原理是用现实世界的概念思考问题，从而自然地解决问题。面向对象分析的基本任务就是提取应用领域中有意义的对象。因为对象就是面向对象软件的模块，实际上在面向对象分析阶段就开始做结构设计工作了，可见，面向对象分析阶段比它在结构化范型中的对应阶段（需求分析）走得更远。面向对象设计的基本任务是扩充和完善面向对象分析所建立的对象模型，把它转变成解空间的对象模型。由于用面向对象范型开发软件时各阶段使用的概念和表示方法相当一致，有利于在各项开发活动之间的平滑过渡，从而降低了开发工作的难度，减少了所犯错误的数量。

1.4 软件过程

软件过程定义了运用技术方法的顺序、应该交付的文档资料、为保证软件质量和协调软件变化必须采取的管理措施，以及标志完成了相应开发活动的里程碑。

通常使用生命周期模型概括地描述软件过程。生命周期模型规定了软件过程包含的各个阶段，以及完成这些阶段的顺序。

典型的生命周期模型如下所述。

1. 瀑布模型

传统软件工程方法学的软件过程基本上可以用瀑布模型来描述。

实际的瀑布模型如图 1.1 所示。

瀑布模型的主要优点如下。

- 强迫开发人员采用规范的技术方法。
- 严格地规定了每个阶段必须提交的文档。
- 每个阶段结束前必须正式进行严格的技术审查和管理复审。

瀑布模型的主要缺点是：在可运行的软件产品交付给用户之前，用户只能通过文档来了解未来的产品是什么样的。开发人员和用户之间缺乏有效的沟通，很可能导致最终开发出的软件产品不能真正满足用户的需求。

2. 快速原型模型

所谓“快速原型”是快速建立起来的、可在计算机上运行的程序，它所能完成的功能往往

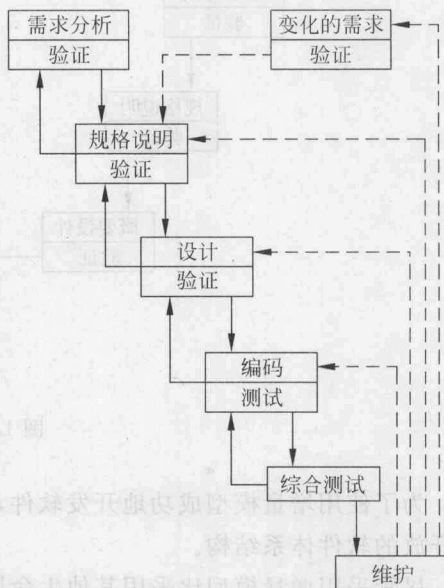


图 1.1 实际的瀑布模型

是最终的软件产品所能完成功能的子集。原型是软件开发人员与用户沟通的强有力工具,因此有助于所开发出的软件产品满足用户的真实需求。

图 1.2 描绘了快速原型模型。

快速原型模型的主要优点如下。

- 使用这种软件过程开发出的软件产品通常能满足用户的真实需求。
- 软件产品的开发过程基本上是线性顺序过程。

3. 增量模型

增量模型也称为渐增模型,如图 1.3 所示。

使用增量模型开发软件时,把软件产品作为一系列增量构件来设计、编码、集成和测试。每个构件由若干个相互协作的模块构成,并且能够完成相对独立的功能。

增量模型的主要优点如下。

- 能在较短时间内向用户提交可完成部分工作的产品。
- 逐步增加产品功能,从而使用户有较充裕的时间学习和适应新产品,减少一个全新的软件给用户所带来的冲击。

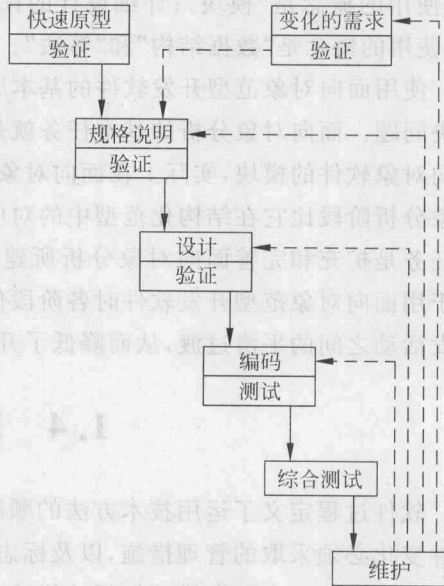


图 1.2 快速原型模型

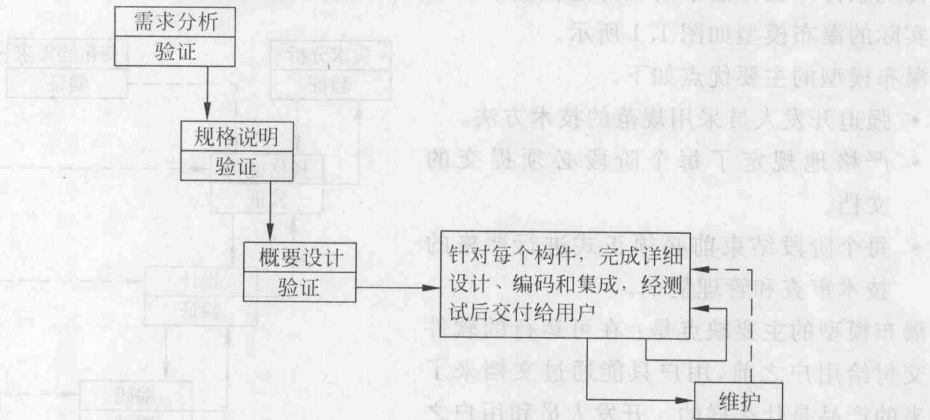


图 1.3 增量模型

为了使用增量模型成功地开发软件,软件工程师必须具有较高的技术水平,能够设计出开放的软件体系结构。

尽管采用增量模型比采用其他生命周期模型需要更精心的设计,但在设计阶段多付出的劳动将在维护阶段获得回报,因为用增量模型开发出的软件具有较好的可扩充性。