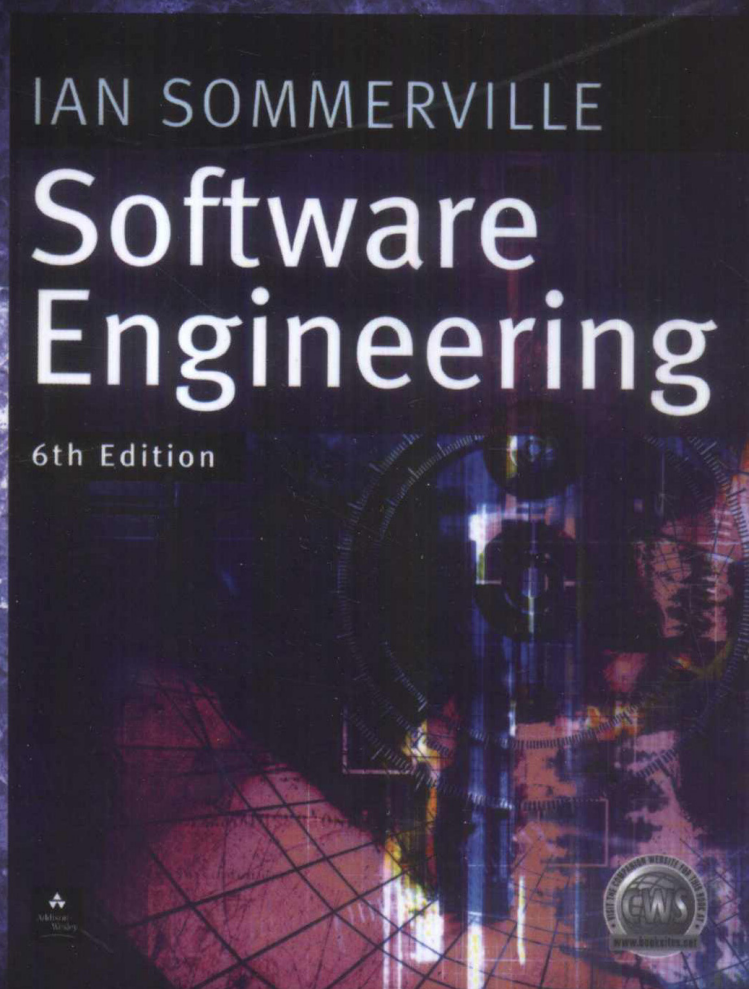


计 算 机 科 学 丛 书

原书第6版

软件工程

(英) Ian Sommerville 著 程成 陈霞 等译



Software Engineering
6th Edition



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE



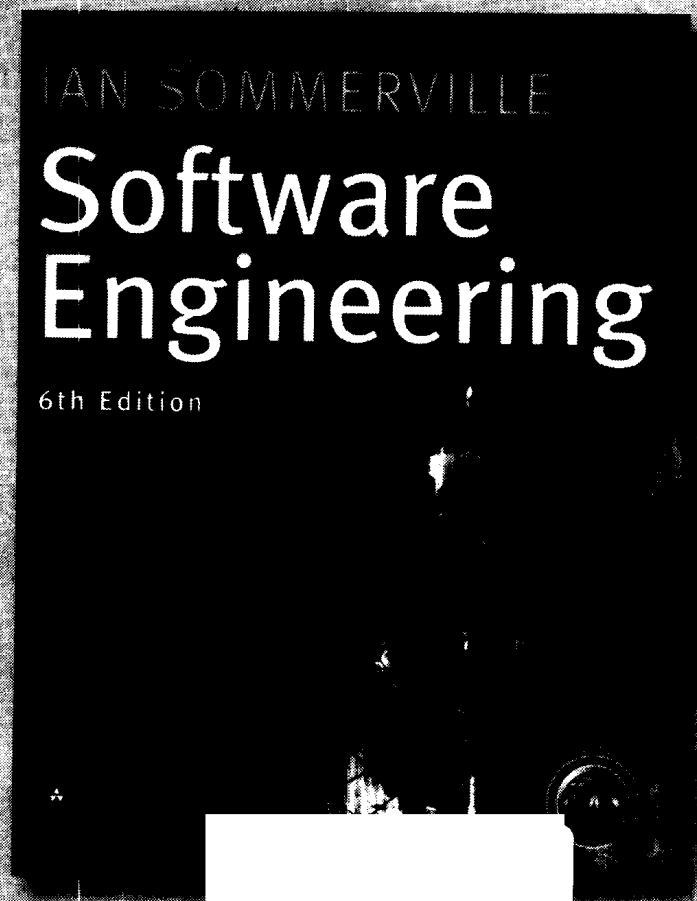
21

计 算 机 科 学 丛 书

原书第6版

软件工程

(英) Ian Sommerville 著 程成 陈霞 等译



Software Engineering
6th Edition



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

本书是英国著名软件工程学家 Ian Sommerville 系统介绍软件工程理论的力作,以要求极高的一类系统为实例,精辟透彻地阐述了软件工程的内涵。作者在第 2 篇中详细阐述了软件系统需求的内容、过程和描述方法,尤其是对形式化描述有详细的介绍和分析;在第 3 篇中针对主要类型系统的设计问题和重要的设计方法展开介绍,主要介绍了分布式系统设计、实时系统设计和用户界面设计,同时从体系结构、面向对象以及复用等视角对设计过程进行了精辟阐述;在第 4 篇和第 5 篇中,主要以实时系统为对象,介绍了系统的一系列重要特性及其度量方法,以及生命周期中包含的软件工程过程(如验证和测试等)内容和方法,内容翔实而透彻。在第 7 篇中作者对遗留系统处理和软件进化理论给出了高屋建瓴的分析,对软件改造的实践者来说无疑是一盏指路明灯;除此之外,作者还在第 1 篇和第 6 篇中对软件工程过程的诸多管理方面(如人员管理、项目管理、过程管理、成本估计和质量管理等重要方面)有极为详细的介绍和分析。

本书是软件工程学界的经典,被众多世界知名大学所采用,既有理论高度又有具体操作方法,对于软件工程的管理者和参与者来说都是不能不看的好教材。

Ian Sommerville; Software Engineering (ISBN:0-201-39815-X)

Copyright © 1982, 1984 by Addison-Wesley Publishers Ltd, Copyright © 1989, 2001 by Pearson Education Ltd.

This translation of Software Engineering is published by arrangement with Pearson Education Limited.

本书中文简体字版由英国 Pearson Education 培生教育出版集团授权出版。

版权所有,侵权必究。

本书版权登记号:图字:01-2002-0608

图书在版编目(CIP)数据

软件工程/(英)萨默维尔(Sommerville, I.)著;程成等译.-北京:机械工业出版社, 2003.1

(计算机科学丛书)

书名原文:Software Engineering

ISBN 7-111-11118-4

I. 软… II. ①萨…②程… III. 软件工程-教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2002)第 083430 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:温莉芳

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2003 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·31.75 印张

印数:0001-6000 册

定价:49.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“国外经典教材”是响应教育部提出的使用外版教材的号召，为国内高校的计算机本科教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这20多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995265

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

前言

软件系统现在已经变得无处不在。事实上,几乎所有的电气设备都内含某种软件,软件已被广泛用于制造业、学校和大学、保健、财政和政府部门,还有许多人使用不同类型的娱乐和教育软件。而所有这些软件系统的描述、开发、管理和进化构成了软件工程的基本内容。

即使是最简单的软件系统也有其固有的复杂性,因此,必须在软件开发中使用工程原则。软件工程是这样一个工程学科,即软件工程师用计算机科学中的方法和理论,在考虑成本效益的情况下,将其用于解决软件开发中的复杂问题,这些复杂问题的存在意味着许多软件开发项目的进行并不总是成功的。然而,绝大多数的现代软件能为用户提供好的服务,我们不应该让某些失败遮挡了过去 30 年中软件工程师们的巨大成功。

软件工程在为国防、政府和工业部门开发大型的、定制的软件系统中得到了发展。我们现在所开发的软件较以前具有更广的范围,从专门游戏设备上的游戏软件、个人计算机上运行的软件和基于 Web 的网络系统再到大规模的分布式系统。虽然有些技术是从定制系统开发中发展出来的,例如:面向对象的开发方法,但这些技术也同样是普遍适用的,新的软件工程技术正渗透到各种不同类型的软件开发过程当中。在这本书中不可能概括所有技术,因此,本书只针对那些大型系统开发技术进行分析,而不对个别产品中的技术进行过多阐述。

虽然本书是对软件工程的介绍,但由于作者对软件需求工程和要求极高的一类系统具有浓厚的兴趣,因此,本书对这两个方面有较大的侧重。在 21 世纪,我们面临的挑战是在确保软件能满足用户需要的同时不会对用户及其周围环境造成伤害。作者认为这些对于软件工程来说是至关重要的。

本书是要将软件工程全方位地展现给读者,而不是介绍某种特别的方法或具体的工具。作者不赞同对学院派的布道式的形式化方法的过分热衷,也不赞同对兜售的任何声称能解决软件开发问题的工具或方法的迷信。事实上,并不存在解决软件工程问题的灵丹妙药,我们需要的是广泛的技术和工具以解决软件工程问题。

每本书都不可避免地反映作者的个人观点甚至是偏见。可能有些读者与我观点相左,对我书中选择的材料不满意。存在不同意见是非常正常的,对软件工程的发展也是有益的。尽管如此,我还是期待所有的软件工程师和软件工程的学生能从本书中找到自己感兴趣的東西。

对第 5 版的修改

像许多软件系统一样,本书自 1982 年第 1 版问世以来进行了多次的修改和完善。对第 5 版的一个修改目标是要减少本书的字数,这对我来说是一件非常困难的选择,我必须在压缩字数的同时保留书中那些重要的内容,同时还要添加一些重要的新素材。结果是第 6 版较第 5 版的字数缩减了 10%。其中主要的改动如下:

- 由前一版的 8 大部分改为 7 大部分,包含软件工程介绍、描述、设计、要求极高的系统开发、校验和有效性验证、管理和软件进化。
- 增加了有关软件过程、分布式系统体系结构、可靠性和遗留系统等几章。形式化描述部

分缩减成一章,同时将 CASE 的有关材料进行了适当缩减并分散在不同的章节中介绍。功能设计的介绍放在遗留系统那一章。校验和有效性验证合并成一章。

- 对所有的章节都进行了修改,对部分章节进行了全面的改写。对复用的研究集中在基于复用的开发、基于模式信息的开发以及基于组件的开发;面向对象的设计已经不仅仅局限为过程的焦点;有关需求的那部分章节分成需求和需求工程过程两大部分。成本估计已更新为 COCOMO 2。
- 概览篇包括 4 章。原先分散在第 5 版中的介绍材料全在这一篇介绍。第 1 章完全改写成有关软件工程的常见问题。
- 有关要求极高的系统的内容被重新组织和整理以便对可靠性、安全性和可用性进行集中的阐述。将保密性作为要求极高的一类系统的一个属性进行了相应的阐述。
- 书中的程序用 Java 书写,对象模型全部用 UML 来描述。Ada 和 C++ 的程序实例已经被删除,但仍可以从作者的网站上得到。

每章中有关进一步阅读部分进行了适当的改动,然而,在某些情况下,作者在 20 世纪 80 年代写的一系列文章现在看来仍然是相关主题的最佳介绍。

读者对象

本书适合作为在校本科生和研究生的教科书及软件工程师的参考书,也可以作为通用软件工程课程或高级程序设计、软件描述、软件设计或软件管理课程的教科书。软件从业者同样可以将其作为一般阅读材料来充实自己在某一方面的知识,如:需求工程学、体系结构设计、可靠的系统开发和过程改善等专题。不管用于哪些方面,本书都针对软件工程师必须掌握的不同应用类型给出了相应的实例。

本书的读者应当熟悉程序设计、现代计算机系统知识和基本数据结构,如堆栈、列表和队列等基本知识。

用本书作为教材

有三种主要类型的软件工程课程可以用本书作为教材:

1. **对软件工程的一般性介绍** 对在此之前没有任何软件工程经验的学生,可以从第 1 篇开始,对后续章节可以大胆取舍,以合适为原则。这将会给所有学生一个总体介绍,同时给那些有浓厚兴趣的学生进一步阅读创造了条件。

2. **对软件工程的某些专题的介绍** 可以作为软件需求描述、软件设计、软件工程管理、可靠的系统开发和软件进化专题的教材。书中每个专题部分都可以根据需要作为该专题的一般性教材。另外还介绍了一些附加的阅读材料。

3. **对软件工程一些专题的进一步阅读** 对软件工程的各专题,本书中的相应部分是其进一步学习的基础,其他一些进一步的阅读资料将会更详细地探讨。每章都给出进一步的阅读的书目,另外,在作者的网站上还可以看到另外一些附加的阅读书目。

本书内容全面,可作为多种相关专题课程的教材。在兰卡斯特,我们用这本书作为软件工程概论、系统描述、系统设计、要求极高的系统分析以及软件管理等课程的补充教材。通过阅读本书,学生们可以获得对这个学科的全面认识。他们喜欢这样覆盖面广的书,因为一书在手,就不用再花钱另购其他书了。

本书覆盖了 ACM/ IEEE 2001 文档中规划的有关计算机科学中软件工程方面的全部内容,同时也与 2000 年或 2001 年出版的 IEEE/ACM 软件工程知识体系纲要规定的内容相一致。

相关 Web 站点

作者的网址是 <http://www.software-engin.com>,在这里可以查到本书作为教材所需用到的相关材料。下面这些补充材料可以通过直接下载得到:

- 教师指南,包括教学提示、学期和课程安排、案例研究和实例分析以及练习的答案,并以 Adobe PDF 格式提供。
- 为每章安排了一组幻灯片,以 Adobe PDF 格式和微软的 PowerPoint 格式提供,教师可以根据自己的偏好进行任意取舍。
- 用 Java 语言编写的程序实例代码,包括编译所需的其他代码。
- 一些先前版本中有关代数描述、Z 方法、面向功能的设计,第 5 版中的 Ada 和 C++ 例子也可以得到。

这个站点还能链接到有关系统工程的文章和幻灯片上,也能链接到其他的软件工程的主页上,以进一步查询其他的书目。

我总是非常高兴接受对本书的反馈意见,读者可以用电子邮件(地址: ian@software-engin.com)与我联系。然而,我很遗憾地说我没有时间来答复每个人在课后作业中的问题。

Ian Sommerville
2000 年 2 月于 Lancaster

目 录

出版者的话	进一步阅读资料	25
专家指导委员会	练习	26
前言		
第1篇 概 览		
第1章 概述		1
1.1 软件工程中的常见问题		2
1.1.1 什么是软件		3
1.1.2 什么是软件工程		3
1.1.3 软件工程和计算机科学 有何区别		4
1.1.4 软件工程和系统工程有 何区别		4
1.1.5 什么是软件过程		4
1.1.6 什么是软件过程模型		5
1.1.7 什么是软件工程的成本		5
1.1.8 什么是软件工程方法		6
1.1.9 什么是CASE		7
1.1.10 什么是优良软件的属性		7
1.1.11 软件工程所面临的主要 问题是什么		8
1.2 职业和道德上的责任		8
要点		10
进一步的阅读资料		11
练习		11
第2章 基于计算机的系统工程		12
2.1 系统总体特性		13
2.2 系统及其环境		15
2.3 系统建模		16
2.4 系统工程过程		18
2.4.1 系统需求定义		19
2.4.2 系统设计		20
2.4.3 子系统开发		21
2.4.4 系统整合		21
2.4.5 系统安装		22
2.4.6 系统操作		22
2.4.7 系统进化		23
2.4.8 系统退役		23
2.5 系统获得		23
要点		25
第3章 软件过程		27
3.1 软件过程模型		28
3.1.1 “瀑布”模型		29
3.1.2 进化式开发		30
3.1.3 形式化系统开发		31
3.1.4 面向复用的开发		32
3.2 过程反复		33
3.2.1 增量式开发		33
3.2.2 螺旋式开发		35
3.3 软件描述		36
3.4 软件设计和实现		37
3.4.1 设计方法		38
3.4.2 程序设计和调试		39
3.5 软件有效性验证		39
3.6 软件进化		41
3.7 自动化的过程支持		42
要点		44
进一步的阅读资料		45
练习		45
第4章 项目管理		47
4.1 管理活动		48
4.2 项目规划		50
4.2.1 项目计划		51
4.2.2 项目里程碑和可交付的 文档		51
4.3 项目进度		52
4.4 风险管理		57
4.4.1 风险识别		58
4.4.2 风险分析		59
4.4.3 风险规划		60
4.4.4 风险监控		60
要点		61
进一步的阅读资料		61
练习		62
第2篇 需 求		
第5章 软件需求		65

5.1 功能需求和非功能需求	67	8.1 软件过程中的原型开发	119
5.1.1 功能需求	67	8.1.1 进化式原型开发	120
5.1.2 非功能需求	68	8.1.2 抛弃式原型开发	122
5.1.3 领域需求	70	8.2 快速原型技术	123
5.2 用户需求	71	8.2.1 使用动态高级语言的开发	124
5.3 系统需求	73	8.2.2 数据库程序设计	125
5.3.1 结构化语言描述	74	8.2.3 组件和应用集成	126
5.3.2 使用 PDL 的需求描述	75	8.3 用户界面原型开发	129
5.3.3 接口描述	75	要点	129
5.4 软件需求文档	77	进一步的阅读资料	130
要点	79	练习	130
进一步的阅读资料	80		
练习	80	第 9 章 形式化描述	132
第 6 章 需求工程过程	82	9.1 软件过程中的形式化描述	134
6.1 可行性研究	83	9.2 接口描述	136
6.2 需求导出和分析	84	9.3 行为描述	140
6.2.1 面向视点的导出	85	要点	145
6.2.2 场景	90	进一步的阅读资料	145
6.2.3 深入实际	92	练习	145
6.3 需求有效性验证	93		
6.4 需求管理	95	第 3 篇 设 计	
6.4.1 持久的和易变的需求	96	第 10 章 体系结构的设计	147
6.4.2 需求管理规划	96	10.1 系统构成	150
6.4.3 需求变更管理	98	10.1.1 容器模型	150
要点	98	10.1.2 客户机/服务器模型	152
进一步的阅读资料	99	10.1.3 抽象机模型	153
练习	99	10.2 控制模型	153
第 7 章 系统模型	101	10.2.1 集中式控制	154
7.1 上下文模型	103	10.2.2 事件驱动系统	155
7.2 行为模型	104	10.3 模块化分解	157
7.2.1 数据流模型	105	10.3.1 对象模型	157
7.2.2 状态机模型	106	10.3.2 数据流模型	158
7.3 数据模型	108	10.4 领域相关的体系结构	159
7.4 对象模型	110	10.4.1 类模型	160
7.4.1 继承模型	111	10.4.2 参考体系结构	161
7.4.2 对象聚合	113	要点	162
7.4.3 对象行为建模	113	进一步的阅读资料	162
7.5 CASE 工作平台	114	练习	163
要点	115	第 11 章 分布式系统体系结构	164
进一步的阅读资料	115	11.1 多处理器体系结构	167
练习	116	11.2 客户机/服务器体系结构	167
第 8 章 软件原型系统开发	117	11.3 分布式对象体系结构	171
		11.4 CORBA	173

要点	176	15.4.2 帮助系统的设计	236
进一步的阅读资料	177	15.4.3 用户文档	238
练习	177	15.5 界面评价	238
第 12 章 面向对象的设计	178	要点	240
12.1 对象和对象类	180	进一步的阅读资料	240
12.2 面向对象设计过程	183	练习	241
12.2.1 系统上下文和使用模型	185		
12.2.2 体系结构的设计	186	第 4 篇 要求极高的系统	
12.2.3 对象识别	186	第 16 章 可依赖性	243
12.2.4 设计模型	188	16.1 要求极高的系统	245
12.2.5 对象接口描述	191	16.2 可用性和可靠性	247
12.3 设计进化	192	16.3 安全性	251
要点	192	16.4 保密性	252
进一步的阅读资料	193	要点	254
练习	193	进一步的阅读资料	254
第 13 章 实时软件的设计	195	练习	254
13.1 系统设计	197	第 17 章 要求极高的系统的描述	256
13.1.1 实时系统建模	198	17.1 软件可靠性描述	257
13.1.2 实时程序设计	199	17.1.1 可靠性度量	258
13.2 实时执行者	200	17.1.2 非功能性的可靠性需求	259
13.3 监控系统	202	17.2 安全性描述	261
13.4 数据采集系统	206	17.2.1 危险和风险分析	262
要点	207	17.2.2 缺陷树分析	264
进一步的阅读资料	208	17.2.3 风险评估	265
练习	208	17.2.4 降低风险	266
第 14 章 使用复用的设计	210	17.3 保密性描述	267
14.1 以组件为基础的开发	213	要点	268
14.1.1 应用框架	216	进一步的阅读资料	268
14.1.2 商业现成产品的复用	217	练习	269
14.1.3 面向复用的组件开发	218	第 18 章 要求极高的系统的开发	270
14.2 应用系列	219	18.1 缺陷最小化	271
14.3 设计模式	221	18.1.1 错误避免	272
要点	223	18.1.2 信息隐藏	274
进一步的阅读资料	224	18.1.3 可靠的软件过程	275
练习	224	18.2 容错	275
第 15 章 用户界面设计	225	18.2.1 异常处理	276
15.1 用户界面设计原则	227	18.2.2 缺陷检测	279
15.2 用户交互	229	18.2.3 损害评估	280
15.3 信息表示	230	18.2.4 缺陷恢复	282
15.4 用户支持	234	18.3 容错体系结构	283
15.4.1 错误消息	235	18.4 安全系统设计	286
		要点	286

进一步的阅读资料	287	进一步的阅读资料	336
练习	287	练习	337
第 5 篇 检验和有效性验证方法		第 6 篇 管 理	
第 19 章 检验和有效性验证	289	第 22 章 人员管理	339
19.1 检验和有效性验证规划	292	22.1 思维的局限性	340
19.2 软件检查	294	22.1.1 记忆结构	340
19.3 自动静态分析	298	22.1.2 问题的解决	342
19.4 净室软件开发方法	300	22.1.3 工作动力	344
要点	302	22.2 小组协作	345
进一步的阅读资料	303	22.2.1 小组的构成	345
练习	303	22.2.2 小组的凝聚力	346
第 20 章 软件测试	304	22.2.3 小组的沟通	347
20.1 缺陷测试	306	22.2.4 小组的结构	348
20.1.1 黑盒测试	306	22.3 选择和留住职员	349
20.1.2 等价划分	307	22.4 人员能力成熟度模型	352
20.1.3 结构化测试	310	要点	353
20.1.4 路径测试	311	进一步的阅读资料	353
20.2 集成测试	313	练习	354
20.2.1 自顶向下的和自底向上的 测试	314	第 23 章 软件成本估算	355
20.2.2 接口测试	315	23.1 生产率	357
20.2.3 强度测试	317	23.2 估算技术	360
20.3 面向对象的测试	317	23.3 算法成本建模	362
20.3.1 对象类测试	318	23.3.1 COCOMO 模型	363
20.3.2 对象集成	318	23.3.2 在项目规划中的算法 成本模型	368
20.4 测试工作平台	320	23.4 项目的工期和人员配备	369
要点	321	要点	370
进一步的阅读资料	322	进一步的阅读资料	371
练习	322	练习	371
第 21 章 要求极高的系统的验证	324	第 24 章 质量管理	373
21.1 形式化方法和要求极高的系统	325	24.1 质量保证和质量标准	376
21.2 可靠性验证	326	24.1.1 文档编写标准	377
21.2.1 操作简档	327	24.1.2 过程 and 产品质量	379
21.2.2 可靠性预测	328	24.2 质量规划	379
21.3 安全性保证	330	24.3 质量控制	380
21.3.1 检验和有效性验证	331	24.4 软件测量和度量	381
21.3.2 安全性论证	331	24.4.1 测量过程	383
21.3.3 过程保证	333	24.4.2 产品度量	384
21.3.4 运行时安全检查	335	24.4.3 测量的分析	385
21.4 保密性评估	335	要点	386
要点	336	进一步的阅读资料	386

练习	387	练习	430
第 25 章 过程改善	388	第 28 章 软件再工程	432
25.1 过程 and 产品质量	390	28.1 源代码转换	435
25.2 过程分析和建模	391	28.2 反向工程	436
25.3 过程度量	394	28.3 程序结构改善	437
25.4 SEI 的过程能力成熟度模型	396	28.4 程序模块化	439
25.5 过程分类	400	28.5 数据再工程	440
要点	401	要点	443
进一步的阅读资料	401	进一步的阅读资料	444
练习	402	练习	444
第 7 篇 软件进化			
第 26 章 遗留系统	403	第 29 章 配置管理	446
26.1 遗留系统的结构	405	29.1 配置管理规划	448
26.2 遗留系统设计	408	29.1.1 配置项识别	448
26.3 遗留系统评估	411	29.1.2 配置数据库	449
26.3.1 业务价值评估	412	29.2 变更管理	450
26.3.2 系统质量评估	413	29.3 版本和发布管理	452
要点	415	29.3.1 版本标识	453
进一步的阅读资料	415	29.3.2 发布管理	455
练习	415	29.4 系统构建	456
第 27 章 软件变更	417	29.5 用于配置管理的 CASE 工具	457
27.1 程序进化的动态特性	418	29.5.1 变更管理支持	457
27.2 软件维护	420	29.5.2 版本管理支持	457
27.2.1 维护过程	423	29.5.3 系统构建支持	458
27.2.2 维护预测	424	要点	459
27.3 体系结构的进化	426	进一步的阅读资料	460
要点	430	练习	460
进一步的阅读资料	430	参考文献	462
		索引	477
		译后记	487

第 1 篇 概 览

第 1 章 概 述

目标

本章的目标是介绍软件工程这门学科,读完本章,你将了解以下内容:

- 什么是软件工程,为什么它很重要;
- 软件工程入门的几个关键问题;
- 道德和职业问题对于软件工程的重要性。

内容

- 1.1 软件工程中的常见问题
- 1.2 职业和道德上的责任

目前,所有国家都在使用复杂的计算机系统。越来越多的产品把计算机和控制软件以一定的方式结合起来,其中,软件在整个系统的成本中所占份额也越来越大。因此,以高性价比的方式生产软件对于国家和世界经济的运作是必不可少的。

软件工程作为一门工程学科,其目标在于使软件系统向高性价比发展。软件是抽象的、不可触摸的,它不受物质材料的限制,也不受物理定律或加工过程的制约,这一方面可使软件工程得到简化,因为软件的潜能不受物理因素的限制;另一方面,由于缺乏自然约束,软件也就很容易变得极为复杂,理解它就很难。

软件工程同时又是一门比较年轻的学科。“软件工程”这一概念是在1968年召开的一个当时被称作“软件危机”的会议上首次提出的。该“软件危机”直接源于强大的(在当时)第3代计算机硬件的问世,它的问世使迄今为止难以实现的计算机应用成为现实,由此产生的软件与原有软件系统相比在数量上更庞大也更复杂。

构建这些软件系统的早期经验是:非正规的软件开发并不奏效。多数软件项目有时甚至要推迟几年才能完成,而且比预计的费用高、不可靠、难以维护,总之,做得很差。软件开发陷入危机,结果是硬件成本在下降而软件成本却呈快速增长之势,因而必须要有新的技术和方法来控制大型软件系统固有的复杂性。

这些新技术和方法就构成了软件工程的一部分,尽管还没有普及但现在已经广泛使用了。然而,制作既满足用户要求又能按期完成并在预算范围之内的复杂软件仍然存在很多困难。许多软件项目仍存在问题,这自然使某些评论家(Pressman,1997)认为软件工程正处在漫长的艰难状态。

软件制作能力提高的同时,软件系统的复杂程度也在提高,由计算机和通信系统相融合产生的新技术也对软件工程师提出了更新的要求。再加上许多企业不能有效利用软件工程技术,所以存在的困难很多。当然,事情并不像预言家所说的那么糟糕,我们仍有完善的余地。

作者认为自1968年以来我们已经取得长足的进步,软件工程的发展已经极大地完善了我们的软件,对于软件开发活动也有了更深的了解,并且已经找到了进行软件描述、设计和实现的有效方法。新的标记法和工具大大降低了制作大型、复杂系统的工作量。

软件工程师应该为自己所做出的成绩感到自豪,没有复杂的软件我们就不能探索太空,也就没有因特网和现代的远程通信,各种形式的旅行就会很危险且花费很高。软件工程在它诞生以后的不长的时间里就已做出了巨大贡献。我坚信:随着软件工程这门学科的成熟,它对21世纪的贡献将是不可限量的。

1.1 软件工程中的常见问题

这一节回答了关于软件工程的一些基本问题,给出了作者对这门学科的观点。本书使用的格式是列出“FAQ(常见问题)表”,在因特网新闻组中常用此法给新手列出常见问题的答案,相信这对简明地介绍这门学科很有效。

本节的问题及答案列于表1-1。

表 1-1 软件工程中常见的问题

问 题	答 案
什么是软件	计算机程序和相关文档。软件产品可为特定客户或通用市场开发

(续)

问 题	答 案
什么是软件工程	软件工程是关于软件生产的各个方面的工程学科
软件工程和计算机科学有何区别	计算机科学侧重理论和基础,而软件工程则侧重于软件开发和交付的实际活动
软件工程和系统工程有何区别	系统工程侧重基于计算机系统开发的所有方面,包括硬件、软件和处理工程。软件工程只是它的一部分
什么是软件过程	以软件开发和进化为目的的一系列活动
什么是软件过程模型	从特定角度提出的软件过程的简化表示形式
什么是软件工程的成本	软件开发成本约占 60%,测试成本占 40%。对于定制软件而言,进化成本常常高于开发成本
什么是软件工程方法	软件开发的结构化研究方法,包括:系统模型、标记法、规则、设计忠告和过程指南
什么是 CASE(计算机辅助软件工程)	旨在使软件过程活动自动化的软件系统。CASE 常用作方法支持
什么是优良软件的特点	软件应具有用户所需的功能与性能,而且应该可维护、可靠、可用
软件工程面临的主要挑战是什么	要面临遗留系统、不断增长的多样性以及减少递交次数等问题的挑战

1.1.1 什么是软件

许多人把软件这一术语等同于计算机程序,其实这种理解是很狭隘的。软件是程序和所有使程序正确运行所需要的相关文档和配置信息。一个软件系统通常包含大量独立的程序、用于设置这些程序的配置文件、描述系统结构的系统文档和如何使用该系统的用户文档,以及告知用户下载最新产品信息的 Web 站点。

软件工程人员不仅要关心软件产品的开发,还应该关心这些产品是否能够卖出去。软件产品有以下两类:

1. **通用软件产品** 由软件开发机构制作,在市场上公开销售,可以独立使用,有时也称为缩卷软件(shrink-wrapped software)。这类软件产品有:数据库软件、字处理软件、绘图软件以及工程管理工作等。
2. **定制软件产品** 这些产品受特定的客户委托,由软件承包商专门为这类客户开发。这类软件有:电子设备的控制系统、特定的业务处理系统和空中交通管制系统等。

这两类产品的一个重要区别在于:在通用软件中,软件描述由开发者自己完成,而定制软件产品,其软件描述通常是由客户给出,开发者必须按客户要求开发。

1.1.2 什么是软件工程

软件工程是一门工程学科,涉及软件生产的各个方面,从最初的系统描述一直到使用后的系统维护,都属于其学科范畴。在软件工程的定义中有两个关键词:

1. **工程学科** 干什么事情都离不开工程人员,他们既能恰当地应用理论、方法和工具,又能有选择地利用它们,即使在没有可用的理论和方法的情况下,也力求找出解决问题的方法。同时他们也认识到必须在机构或财政状况所允许的限度内工作,即要在此限度