



21世纪高职高专规划教材
软件专业系列

实用软件工程 与实训

杜文洁 白 萍 主编



清华大学出版社



21世纪高职高专规划教材

软件专业系列

实用软件工程 与实训

杜文洁 白萍 主编

清华大学出版社
北京

内 容 简 介

本书着重从实用角度讲述软件工程的基本概念、原理和方法,系统地介绍了目前较成熟的、广泛使用的软件工程技术。本书内容包括:软件工程概述、软件需求分析、软件设计、程序编码、面向对象技术、统一建模语言 UML、软件测试、软件维护、软件工程管理、软件复用与构件技术、软件工具与集成化环境、综合实例——网站新闻管理系统。

本书结合教学实例突出基本知识和基本概念的表述,注重内容的先进性、系统性和实用性,力求反映软件工程技术发展的最新成果。本书兼顾理论与实践相结合,从而达到培养应用型人才的目标。

本书可作为高等院校、高职高专院校相关软件专业和计算机相关专业的教材,也可作为软件工程技术培训的教材,同时还可供从事软件开发及应用的广大科技人员参考使用。

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

实用软件工程与实训/杜文洁主编. —北京:清华大学出版社,2009.4

21 世纪高职高专规划教材. 软件专业系列

ISBN 978-7-302-19586-3

I. 实… II. 杜… III. 软件工程—高等学校:技术学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2009)第 020734 号

责任编辑:田 梅

责任校对:李 梅

责任印制:李红英

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:010-62776969, c-service@tup.tsinghua.edu.cn

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

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260

印 张:16.75

字 数:374 千字

版 次:2009 年 4 月第 1 版

印 次:2009 年 4 月第 1 次印刷

印 数:1~4000

定 价:25.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:028833-01

出版说明

高职高专教育是我国高等教育的重要组成部分,担负着为国家培养并输送生产、建设、管理、服务第一线高素质技术应用型人才的重任。

进入 21 世纪后,高职高专教育的改革和发展呈现出前所未有的发展势头,学生规模已占我国高等教育的半壁江山,成为我国高等教育的一支重要的生力军;办学理念上,“以就业为导向”成为高等职业教育改革与发展的主旋律。近两年来,教育部召开了三次产学研交流会,并启动四个专业的“国家技能型紧缺人才培养项目”,同时成立了 35 所示范性软件职业技术学院,进行两年制教学改革试点。这些举措都表明国家正在推动高职高专教育进行深层次的重大改革,向培养生产、服务第一线真正需要的应用型人才的方向发展。

为了顺应当前我国高职高专教育的发展形势,配合高职高专院校的教学改革和教材建设,进一步提高我国高职高专教育教材质量,在教育部的指导下,清华大学出版社组织出版了“21 世纪高职高专规划教材”。

为推动规划教材的建设,清华大学出版社组织并成立了“高职高专教育教材编审委员会”,旨在对清华版的全国性高职高专教材及教材选题进行评审,并向清华大学出版社推荐各院校办学特色鲜明、内容质量优秀的教材选题。教材选题由个人或各院校推荐,经编审委员会认真评审,最后由清华大学出版社出版。编审委员会的成员皆来源于教改成效大、办学特色鲜明、师资实力强的高职高专院校、普通高校以及著名企业,教材的编写者和审定者都是从事高职高专教育第一线的骨干教师和专家。

编审委员会根据教育部最新文件和政策,规划教材体系,比如部分专业的两年制教材;“以就业为导向”,以“专业技能体系”为主,突出人才培养的实践性、应用性的原则,重新组织系列课程的教材结构,整合课程体系;按照教育部制定的“高职高专教育基础课程教学基本要求”,教材的基础理论以“必要、够用”为度,突出基础理论的应用和实践技能的培养。

本套规划教材的编写原则如下:

- (1) 根据岗位群设置教材系列,并成立系列教材编审委员会;
- (2) 由编审委员会规划教材、评审教材;
- (3) 重点课程进行立体化建设,突出案例式教学体系,加强实训教材的出版,完善教学服务体系;
- (4) 教材编写者由具有丰富教学经验和多年实践经历的教师共同组成,建立“双师型”编者体系。

本套规划教材涵盖了公共基础课、计算机、电子信息、机械、经济管理以及服务等大类的主要课程,包括专业基础课和专业主干课。目前已经规划的教材系列名称如下:

• 公共基础课

公共基础课系列

• 计算机类

计算机基础教育系列

计算机专业基础系列

计算机应用系列

网络专业系列

软件专业系列

电子商务专业系列

• 电子信息类

电子信息基础系列

微电子技术系列

通信技术系列

电气、自动化、应用电子技术系列

• 机械类

机械基础系列

机械设计与制造专业系列

数控技术系列

模具设计与制造系列

• 经济管理类

经济管理基础系列

市场营销系列

财务会计系列

企业管理系列

物流管理系列

财政金融系列

国际商务系列

• 服务类

艺术设计系列

本套规划教材的系列名称根据学科基础和岗位群方向设置,为各高职高专院校提供“自助餐”形式的教材。各院校在选择课程需要的教材时,专业课程可以根据岗位群选择系列;专业基础课程可以根据学科方向选择各类的基础课系列。例如,数控技术方向的专业课程可以在“数控技术系列”选择;数控技术专业需要的基础课程,属于计算机类课程的可以在“计算机基础教育系列”和“计算机应用系列”选择,属于机械类课程的可以在“机械基础系列”选择,属于电子信息类课程的可以在“电子信息基础系列”选择。依此类推。

为方便教师授课和学生学习,清华大学出版社正在建设本套教材的教学服务体系。本套教材先期选择重点课程和专业主干课程,进行立体化教材建设:加强多媒体教学课件或电子教案、素材库、学习盘、学习指导书等形式的制作和出版,开发网络课程。学校在选用教材时,可通过邮件或电话与我们联系获取相关服务,并通过与各院校的密切交流,使其日臻完善。

高职高专教育正处于新一轮改革时期,从专业设置、课程体系建设到教材编写,依然是新课题。希望各高职高专院校在教学实践中积极提出意见和建议,并向我们推荐优秀选题。反馈意见请发送到 E-mail:gzzgztup.tsinghua.edu.cn。清华大学出版社将对已出版的教材不断地修订、完善,提高教材质量,完善教材服务体系,为我国的高职高专教育出版优秀的高质量的教材。

高职高专教育教材编审委员会

前 言

实用软件工程与实训

软件工程是介绍软件开发方法、提高学生软件开发能力的一门课程。随着软件应用日益广泛,软件规模日益扩大,人们开发、使用、维护软件开始采用工程的方法,以求经济有效地解决软件问题。在这种形式下,计算机专业学生对软件工程知识的学习与应用就显得尤为重要,因此一本好的软件工程教材对于学习者是必不可少的。

软件工程是一门正在迅速发展的交叉性学科,有关软件工程的新概念、新技术、新方法不断涌现。我们依据高职高专软件工程课程教学大纲所规定的教学要求编写本书,把多年软件工程教学经验和教学实践成果溶入本教材中,在内容分布上充分考虑理论与实践相结合的原则。

本书选材注意把握高职高专学生的专业知识背景与接受能力,以“实例教学”的方法激发学生的学习兴趣。在教材编写上,力求做到结合实际、注重应用、便于教学,注意内容的新颖、实用和系统性。在结构安排上,深入阐述软件工程的基础理论知识,循序渐进,做到理论和实际相结合。在每章内容后面均附有小结和课后习题。

本书共分 13 章,系统地介绍了软件工程的主要内容,安排如下:

第 1 章 初识软件工程

介绍软件工程的产生和发展、软件的开发方法、软件生存周期及软件开发模型。

第 2 章 软件需求分析

介绍需求分析的任务和步骤、需求分析的各种描述工具及需求分析的文档。

第 3 章 软件设计

具体介绍了软件设计的任务与策略、概要设计、模块和模块化、详细设计、用户界面设计及软件设计规格说明书。

第 4 章 程序编码

具体介绍了结构化程序设计、程序设计风格、程序设计效率及程序设计语言的选择。

第 5 章 面向对象技术

具体介绍了面向对象技术的概念、面向对象分析、面向对象设计及面向对象系统的实现。

第 6 章 统一建模语言 UML

具体介绍了 UML 的内容与应用领域、UML 的静态建模机制及 UML 的动态建模机制。

第7章 软件测试

具体介绍了软件测试的基本概念、软件测试方法、黑盒测试用例设计、白盒测试用例设计、软件测试步骤及面向对象的软件测试。

第8章 软件维护

具体介绍了软件维护过程、软件的可维护性及软件逆向工程与再工程。

第9章 软件工程管理

具体介绍了软件工程管理的内容、软件规模估算方法、软件项目的组织与计划、软件质量特性及保证措施、ISO 9000 软件质量体系、软件配置管理、CMM 软件能力成熟度模型、软件工程标准与软件文档。

第10章 软件复用与构件技术

具体介绍了软件复用的定义和过程、基于构件的软件开发及面向对象的软件重用技术。

第11章 软件工具与集成化环境

具体介绍了软件开发工具、软件工程环境及 CASE 技术。

第12章 综合实例——网站新闻管理系统

通过开发一个网站新闻管理系统实例,介绍用软件工程的方法开发软件的全过程。

第13章 实训指导

通过 5 个实训内容,加深读者对软件工程课程基础理论、基本知识的理解,提高分析和解决软件项目问题的能力。

本书由杜文洁、白萍担任主编,王晓红担任副主编。第 8 章、第 11 章由杜文洁编写,第 2 章、第 3 章、第 6 章、第 7 章、第 12 章由白萍编写,第 1 章、第 4 章、第 5 章、第 9 章、第 10 章由王晓红编写,第 13 章由白萍、王晓红共同编写。另外,刘明国、丛国凤、李迎春、张玉龙、董生忠、马岩也参与了本书部分内容的编写。全书由杜文洁统稿完成。

由于水平和时间有限,书中难免存在错误和不足之处,敬请读者批评指正。

作 者

2008 年 6 月

目 录

实用软件工程与实训

第 1 章 初识软件工程	1
1.1 软件与软件危机	1
1.1.1 理解软件的概念	1
1.1.2 软件的特点	2
1.1.3 产生软件危机的原因	3
1.2 软件工程的由来	5
1.2.1 软件工程的产生和发展	5
1.2.2 浅谈软件工程的定义	6
1.2.3 理解软件工程的作用	6
1.2.4 软件工程研究的内容	8
1.3 软件的开发方法	8
1.3.1 面向过程的方法	8
1.3.2 面向数据的方法	9
1.3.3 面向对象的方法	10
1.4 软件生存周期	11
1.5 了解软件开发模型	12
1.5.1 瀑布模型	13
1.5.2 快速原型模型	13
1.5.3 渐增模型	15
1.5.4 螺旋模型	15
1.5.5 快速应用开发模型	17
1.6 小结	17
1.7 习题	18
第 2 章 软件需求分析	19
2.1 需求分析的任务	19
2.2 需求分析的步骤	20
2.3 实体—关系图	22

2.4	数据流图	24
2.5	数据字典	25
2.6	需求分析图形工具	26
2.7	需求分析的文档	28
2.8	案例分析	30
2.9	小结	33
2.10	习题	34
第3章	软件设计	35
3.1	软件设计概述	35
3.1.1	软件设计概念与重要性	35
3.1.2	软件设计的任务与策略	36
3.1.3	软件设计基本原则	37
3.2	概要设计	38
3.2.1	概要设计的任务和步骤	38
3.2.2	面向数据流图的设计方法	39
3.2.3	面向对象的设计方法	43
3.3	模块和模块化	46
3.3.1	模块	46
3.3.2	模块化	46
3.3.3	模块分割评价标准	48
3.3.4	模块设计规则	50
3.4	详细设计	51
3.4.1	详细设计的目标与任务	51
3.4.2	面向过程详细设计	52
3.4.3	面向数据详细设计	55
3.4.4	面向对象详细设计	55
3.5	Jackson 程序设计方法	60
3.6	Warnier 程序设计方法	62
3.7	用户界面设计	63
3.8	软件设计规格说明书	66
3.9	案例分析	68
3.10	小结	69
3.11	习题	70
第4章	程序编码	71
4.1	程序编码概述	71
4.2	结构化程序设计	72

4.3	程序设计风格	73
4.3.1	源程序文件	73
4.3.2	语句构造方法	74
4.3.3	输入/输出技术	75
4.4	程序设计效率	75
4.5	程序设计语言的选择	77
4.6	小结	82
4.7	习题	83
第5章	面向对象技术	85
5.1	面向对象技术的概念	85
5.1.1	面向对象方法概述	85
5.1.2	面向对象的概念	86
5.1.3	面向对象方法的主要优点	91
5.2	面向对象分析	91
5.2.1	面向对象的分析过程	91
5.2.2	面向对象的分析原则	97
5.3	面向对象设计	98
5.3.1	系统设计	99
5.3.2	对象设计	102
5.3.3	面向对象设计的准则	103
5.3.4	面向对象设计的启发规则	103
5.4	面向对象系统的实现	104
5.5	案例分析	105
5.6	小结	109
5.7	习题	109
第6章	统一建模语言 UML	111
6.1	UML 概述	111
6.1.1	UML 的形成	111
6.1.2	UML 的内容	112
6.1.3	UML 的特点	114
6.1.4	UML 的应用领域	114
6.2	UML 的静态建模机制	115
6.2.1	用例图	115
6.2.2	类图、对象图和包	117
6.2.3	构件图和配置图	123
6.3	UML 的动态建模机制	125

6.3.1	状态图	126
6.3.2	顺序图	127
6.3.3	协作图	127
6.3.4	活动图	127
6.4	案例分析	128
6.5	小结	131
6.6	习题	131
第7章	软件测试	133
7.1	基本概念	133
7.1.1	软件测试的目标	133
7.1.2	软件测试的原则	134
7.2	软件测试方法	134
7.2.1	静态分析与动态测试	135
7.2.2	黑盒测试与白盒测试	135
7.3	黑盒测试用例设计	136
7.3.1	等价类划分法	136
7.3.2	边界值分析法	137
7.3.3	错误推测法	138
7.3.4	因果图法	138
7.4	白盒测试用例设计	141
7.4.1	逻辑覆盖法	141
7.4.2	路径覆盖法	143
7.4.3	循环覆盖法	144
7.5	软件测试步骤	145
7.5.1	单元测试	146
7.5.2	集成测试	147
7.5.3	确认测试	150
7.5.4	系统测试	151
7.6	面向对象的软件测试	153
7.6.1	OOA 和 OOD 模型的测试	153
7.6.2	面向对象的测试策略	156
7.7	案例分析	157
7.8	小结	158
7.9	习题	159
第8章	软件维护	161
8.1	软件维护过程	161

8.1.1	软件维护的种类	161
8.1.2	软件维护的策略	163
8.1.3	软件维护的成本	163
8.1.4	软件维护的困难	164
8.1.5	软件维护的副作用	164
8.2	软件可维护性	166
8.2.1	决定软件可维护性的因素	166
8.2.2	可维护性的度量	167
8.2.3	提高软件的可维护性	167
8.3	软件逆向工程与再工程	169
8.4	小结	170
8.5	习题	170
第9章	软件工程管理	171
9.1	软件工程管理概述	171
9.2	软件规模估算	172
9.2.1	软件开发成本估算方法	172
9.2.2	代码行技术和任务估算技术	173
9.2.3	COCOMO 模型	174
9.3	软件项目的组织与计划	175
9.3.1	软件项目管理的特点	176
9.3.2	软件开发进度计划	177
9.3.3	人员配备与组织	180
9.4	软件质量	182
9.4.1	软件质量特性	182
9.4.2	软件质量保证措施	183
9.5	ISO 9000 软件质量体系	183
9.5.1	ISO 9000 国际标准	184
9.5.2	企业软件质量体系的建立和实施	184
9.6	软件配置管理	187
9.7	CMM 软件能力成熟度模型	190
9.8	软件工程标准与软件文档	195
9.8.1	软件工程标准	195
9.8.2	软件文档的编写	196
9.9	小结	199
9.10	习题	200

第 10 章 软件复用与构件技术	201
10.1 软件复用	201
10.2 基于构件的软件开发	203
10.2.1 开发可复用的软件构件	203
10.2.2 软件构件的组织	205
10.2.3 分类和检索软件构件	206
10.3 面向对象的软件重用技术	209
10.4 小结	211
10.5 习题	211
第 11 章 软件工具与集成化环境	213
11.1 软件开发工具	213
11.2 软件工程环境	215
11.3 CASE 技术	217
11.4 小结	219
11.5 习题	219
第 12 章 综合实例——网站新闻管理系统	221
12.1 问题定义	221
12.2 需求分析	221
12.2.1 用例图	221
12.2.2 类图	223
12.3 软件设计	225
12.3.1 系统体系结构	225
12.3.2 数据库设计	225
12.3.3 功能设计	227
12.3.4 界面设计	228
12.4 系统测试	230
第 13 章 实训指导	233
实训 1 初识 Visio 2003	233
实训 2 初识 Rational Rose 2003	239
实训 3 需求分析	245
实训 4 数据库设计	247
实训 5 面向对象分析	252
参考文献	255

初识软件工程

本章要点

- 软件工程的产生;
- 软件的开发方法;
- 软件生存周期;
- 软件开发模型。

学完本章之后你将能够

- 了解软件的特点及软件工程的作用;
- 了解软件的开发方法;
- 了解软件开发模型的特点。

1.1 软件与软件危机

和计算机硬件一样,自 20 世纪 60 年代以来,软件也从规模、功能等方面得到了很大的发展,同时人们对软件质量的要求也越来越高。那么,究竟什么是软件?软件有哪些特征?

1.1.1 理解软件的概念

随着计算机技术的发展,对软件在不同阶段有不同的认识。计算机发展的初期,硬件的设计和生是主要问题,那时的所谓软件就是程序,甚至是机器指令程序,它们处于从属的地位。软件的生产方式是个体手工方式,设计过程是在人的头脑中完成的,程序的质量完全取决于个人的编程技巧。其后,人们认识到在计算机上增加软件的功能会使计算机系统的功能大大提高,在研制计算机体系时既要考虑其硬件,又要考虑与之配套的软件,并且开始编制一些大型程序系统。这时的生产方式为互助合作的手工方式,认为软件就是程序加说明书。

随后社会对计算机提出了更高的要求,有的大型系统的设计和生作的工作量高达几千人/年(一个人工作一年其工作量为一人/年),指令数百万条。现在,软件在计算机系统的比重越来越大,而且这种趋势还在增加。人们感到传统的软件生产方式已经不适

应发展的需要,因此提出把工程学的基本原理和方法引进软件生产中,把软件生产分成几个阶段,每个阶段都有严格管理和质量检验,研制软件设计和生产的方法及工具,并用书面文件作为共同遵循的依据,这时软件的含义就成了文档加程序。

现在对软件的正确理解应该是,软件是计算机系统中与硬件相互依存的部分,它包括程序及其相关文档。程序是计算机程序所需的阐述性资料。

1.1.2 软件的特点

软件包括三个方面:

- ① 一个或多个计算机程序,这些程序执行时能够提供期望的功能和性能;
- ② 一个或多个数据结构,这些数据结构使得程序能够完全操纵信息;
- ③ 一个或多个文档,这些文档描述了程序的分析、设计、实现和维护的细节及使用说明。

软件不同于硬件,它只具备逻辑形式而不具备物理形式。因此,软件具有与硬件不同的特征。

1. 软件的质量是“开发”出来的,不是“制造”出来的

软件和硬件的质量都与“开发”有关,但软件的“制造”只是简单的复制。软件的质量主要取决于软件“开发”,在软件“制造”过程中几乎不会引入新的质量问题,即使存在,也很容易得到改正。而硬件在“制造”过程中有可能会引入新的质量问题。尽管在“开发”和“制造”两种活动中,人的因素都很重要,但两种活动中的参与人员及其所要做的工作却不一样。而且,在两种活动中,虽然目标都是要构造一个“产品”,但是两者采用的方法是不同的。

在 20 世纪 80 年代中期的时候,有人提出了“软件工厂”的概念。但这一概念也不意味着软件“开发”和硬件“制造”能够等同起来。实际上,“软件工厂”这一概念着重于使用自动化工具和环境来进行软件开发。

2. 软件可能会被“废弃”,但不会被“用坏”

软件从被提交给用户使用开始,只存在维护问题,而不存在在使用过程中被“用坏”的问题。硬件则不同,用户购买到硬件并开始使用之后,随着时间的推移,硬件会因为各种原因(如灰尘、震动、高温、磨损、滥用等)造成某些部件失效,从而导致硬件被“用坏”。

在硬件使用的早期,可能会因为设计或制造时的错误或缺陷造成一些问题。当错误或缺陷被改正之后,硬件即可正常运行一段比较长的时期(不同的硬件其时期长短不同),这一时期硬件出错的可能性非常小,称为硬件使用的中期。随着时间的推移,硬件出错的可能性会越来越高,这一时期可以称为晚期,晚期代表着硬件变得越来越不能正常使用了,也就是“用坏”了。

软件的使用过程中出错的可能性与硬件不同。图 1-1 是软件使用过程中随时间变化的错误率曲线的示意图。在软件使用的早期,错误率是比较大的,这主要是因为软件开发时的错误或缺陷造成的。随着时间的推移和错误的改正,软件中潜在的错误越来越少,因而曲线变得越来越平坦。换句话说,软件不会被“用坏”。当然,随着时间的推移,软件最终会由于不能满足用户的需要而被“废弃”,软件需要更新。

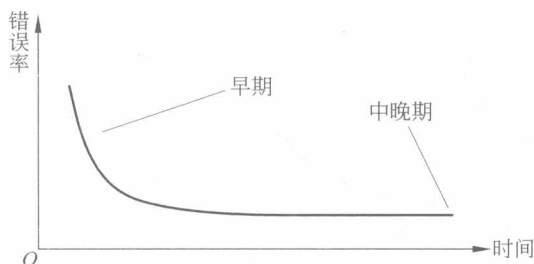


图 1-1 随时间变化的软件错误率曲线

3. 过去的软件大多是“定制”的，而不是“装配”的

现代工业社会中，硬件基本上都是通过集成不同生产厂商的各种零部件，进而“装配”起来的。典型的例子是工厂的流水线作业。硬件设计师的一个主要任务就是要从当前市场上可供选择的零部件中进行选择，通过自己的设计，进而制造出新产品。

软件设计师则没有这么幸运。过去，市场上很少有类似于硬件“零部件”的软件“零部件”。即便有，大多也是具有完整功能的“软件”或独立功能的模块，而不是“零件”或“部件”。因此，软件大多是为用户专门“定制”的，而不是由现有的软件“零部件”“装配”而成的。现在，这种情况正在发生快速变化。面向对象技术和组件开发技术的广泛应用，使得越来越多的软件部件（也称为软件组件、软件构件）被开发出来，并且可以像硬件部件一样做到“即插即用”。当然，软件部件要做到和硬件部件同等程度的“即插即用”，还需要软件工程师进一步地研究和实践。

1.1.3 产生软件危机的原因

随着计算机应用的逐步扩大，软件需求量迅速增加，规模也日益增长。长达数万行、数十万乃至百万行以上的软件，已不鲜见。美国阿波罗登月计划的软件长达 10000 万代码行，航天飞机软件长达 4000 万行，就是两个突出的例子。

软件规模的增长，带来了它的复杂度的增加。如果说编写一个数十到数百行的程序连初学者也不难完成，则开发一个数万乃至数百万行的软件，其复杂度将大大上升，即使是富有经验的程序员，也难免顾此失彼。其结果是，大型软件的开发费用经常超出预算，完成时间也常常脱期。尤其糟糕的是，软件可靠性往往随规模的增长而下降，质量保证也越来越困难。

众所周知，任何计算机系统均由硬件、软件两部分组成。在计算机应用早期，软件仅包含少量规模不大的程序，应用部门花费在软件上的投资（成本）仅占很小的份额。随着应用面的不断扩大，软件的花费越来越大，所占的百分比也越来越高。B. Boehm 在 1973 年发表的一篇文章中预计，到 1985 年，美国空军的软件费用将上升到计算机总费用的 90%（参阅图 1-2）。即在每 100 元用于计算机投资总额中，软件将花费 90 元。这一预计早已为实践所证实。

庞大的软件费用，加上软件质量的下降，对计算机应用的继续扩大构成巨大的威胁。对这种严峻的形势，软件界的有识之士发出了软件危机的警告。

以下再从维护和生产两个方面，进一步说明出现软件危机的原因。

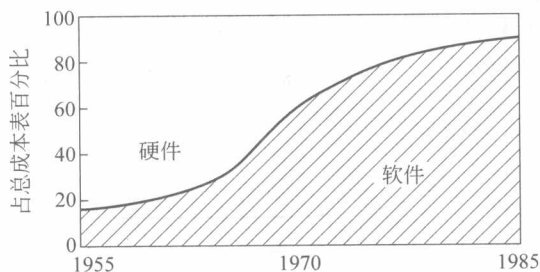


图 1-2 硬件/软件成本变化趋势

1. 软件维护费用急剧上升,直接威胁计算机应用的扩大

根据一些大公司的统计,软件维护费用大约占到软件总花费的 $2/3$,比开发费用高出一倍。一个大型软件,即使在开发时经过严格的测试与纠错,也不能保证运行中不再出现错误。维护的第一件事,就是纠正软件中遗留的错误,称为“纠错性维护”。在此后的运行过程中,还常常要为完善功能、适应环境变更等原因对软件修改,即进行所谓“完善性维护”和“适应性维护”。不言而喻,软件的规模越大,以上各种维护的成本必然越高。

维护既耗费财力,也耗费人力,为了维护,要占用计算机厂家或软件公司大批软件人员,使他们不能参加新软件的开发。难怪有些文献把维护工作中出现的问题比作冰海中横在前进航道上的冰山,或直称之为维护墙(Maintenance Wall),视之为软件生产和维护中难以逾越的障碍。

2. 软件生产技术进步缓慢,是加剧这一软件危机的重要原因

有人统计,在过去 30 年中硬件的性能价格比增长了 10^6 。一种新器件的出现,其性能较旧器件提高,价格反有所下降,这就是微电子技术创造的奇迹。软件则相形见绌。一方面,软件规模与复杂度增长了几个数量级,但生产方式长期未突破手工业的方式,创建新软件的能力提高得十分缓慢(如图 1-3 所示)。另一方面,很多在早期用“自由化”方法开发的、带有很强“个人化”特征的程序,因缺乏文档而根本不能维护,更加剧了供需之间的矛盾。结构化程序设计的出现,使许多产业界人士认识到必须把软件生产从个人化方式改变为工程化方式,从而导致了软件工程的诞生。

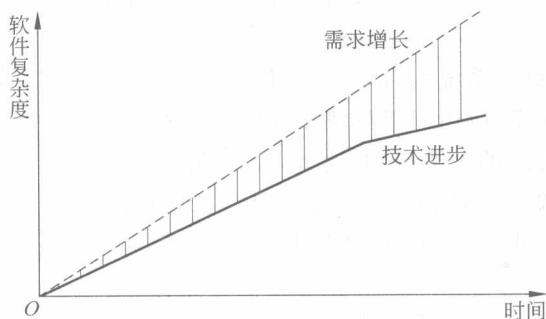


图 1-3 软件技术进步落后于需求的增长