

21世纪

高等院校计算机系列教材

软件工程

曹哲主编
高诚 车进辉 吴燕 副主编
朱敦名 主审



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校计算机系列教材

软件工程

曹 哲 主 编

高 诚 车进辉 吴 燕 副主编

朱敦名 主 审

中国水利水电出版社

内 容 提 要

本书从实用的角度出发系统地介绍了软件工程的基本理论、方法、技术、工具和环境。全书共 17 章。内容主要包括：概述，软件项目管理，计算机系统工程，需求分析，面向数据流的分析方法，面向数据的分析方法与形式化方法，面向对象的需求分析，软件设计基础，面向数据流的设计方法，面向数据的设计方法，面向对象的设计方法，人机界面设计，程序设计语言与编码，软件测试、软件维护、软件配置管理以及软件开发新技术简介等。

本书结构合理，在选材上注重了实用性，以期达到理论与实践相结合、学以致用为目的，本书对计算机辅助软件工程（CASE）工具的介绍几乎贯穿全书，概念清楚、通俗易懂、内容翔实、实例丰富、习题思考题与内容配合紧密。

本书既可作为高等院校“软件工程”课程的教材或教学参考书，也可作为软件开发人员的参考书。

图书在版编目（CIP）数据

软件工程 / 曹哲主编. —北京：中国水利水电出版社，2004

（21 世纪高等院校计算机系列教材）

ISBN 7-5084-1555-8

I. 软… II. 曹… III. 软件工程—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字（2004）第 058766 号

书 名	软件工程
主 编	曹 哲
副 主 编	高 诚 车进辉 吴 燕
主 审	朱敦名
出版 发行	中国水利水电出版社（北京市三里河路 6 号 100044） 网址：www.waterpub.com.cn E-mail: mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 16.75 印张 375 千字
版 次	2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷
印 数	0001—5000 册
定 价	24.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

版权所有·侵权必究

前 言

软件工程是计算机学科中一个非常有价值并具有广阔发展空间的研究领域。多年来,随着计算机硬件技术的迅猛发展,人们开发优质软件的能力远远落后于社会各个领域对计算机软件的需求,也就是说,时至今日,仍然经受着“软件危机”的困扰。为了克服“软件危机”,自20世纪60年代末期以来,人们在这一领域做了大量的研究与实践工作,积累了大量的软件开发技术和方法,进而逐渐形成了系统的软件项目开发与管理理论。于是,一门新兴的学科——软件工程学(简称软件工程)诞生了。软件工程所研究的范围十分广泛,主要包括软件项目开发和软件维护的有关理论、技术、方法、标准、计算机辅助工具和环境以及软件项目管理等诸多方面。软件工程领域的研究成果为缓解软件危机发挥了关键性作用。

“软件工程”课程是高等学校计算机学科教学计划中的一门主干课程。本书正是为普通高校计算机学科“软件工程”课程而编写的教材。本书共17章,其中第1章~第3章介绍软件工程的基本概念、软件项目管理、计算机系统工程;第4章~第7章介绍需求分析的有关内容,包括需求分析概念、面向数据流、面向数据、形式化、面向对象等需求分析方法、需求规格说明与评审等;第8章~第12章介绍软件设计的有关内容,包括软件设计的概念、设计过程和一般性技术,面向数据流、面向数据、面向对象以及人机界面的设计方法与技术、设计规格说明与评审等;第13章~第16章介绍软件的实现、测试与维护,主要包括程序设计语言与编码、软件测试与调试、软件维护、软件配置管理等;第17章简单介绍近期出现的与软件开发有关的新技术,包括软件重用技术、Internet与Java、对象链接与嵌入(OLE)技术以及分布计算技术等。

本书具有以下几个特点:

- (1) 结构合理,系统地介绍了软件工程的基本原理、概念、方法和工具。
- (2) 在选材上注重了实用性,以期达到理论与实践相结合、学以致用目的。
- (3) 对计算机辅助软件工程(CASE)工具的介绍几乎贯穿全书。
- (4) 概念清楚、通俗易懂、内容翔实、实例丰富,习题思考题与内容配合紧密。

本书可以作为高等院校“软件工程”课程的教材或教学参考书,也可以作为软件开发人员的参考书。

本书由曹哲主编,由高诚、车进辉、吴燕任副主编,具体分工为:第1章、第2章和第14章由曹哲编写;第5章、第6章、第8章、第9章、第10章、第12章、第16章由高诚编写,第7章、第11章、第13章、第15章、第17章由车进辉编写,第3章和第4章由吴燕编写。

本书由朱敦名教授主审。在本书的编写过程中,北华大学计算机科学技术学院计算机

实验室的全体同志给予了多方面的支持和帮助。赵津燕、刘艳、李益民、胡晓宏、尹健慧、葛建梅、王双立等同志也参与了大纲讨论，并提出了很多宝贵意见。在此，编者向他们表示衷心地感谢。

由于时间仓促及编者水平有限，书中难免存在疏漏和不妥之处，恳请广大读者批评指正。

编者联系方式：zxqiu3@tom.com。

编 者

2004 年 5 月

目 录

前言

第 1 章 概述	1
1.1 软件	1
1.1.1 软件	1
1.1.2 软件危机	3
1.2 软件工程的概念	4
1.2.1 软件工程的定义	4
1.2.2 软件工程的基本原理	5
1.2.3 软件工程的目标	6
1.2.4 软件工程的原则	7
1.3 软件生存周期	8
1.3.1 软件定义	8
1.3.2 软件开发	9
1.3.3 软件的使用与维护及退役	11
1.4 软件开发模型	12
1.4.1 瀑布模型	12
1.4.2 原型模型	13
1.4.3 螺旋模型	14
1.4.4 喷泉模型	16
1.4.5 变换模型	16
1.4.6 基于四代技术的模型	17
1.4.7 基于知识的智能模型	17
1.5 软件开发方法、工具及环境	18
1.5.1 软件开发方法	18
1.5.2 软件开发工具与环境	18
1.5.3 计算机辅助软件工程	18
习题	19
第 2 章 软件项目管理	20
2.1 软件度量	20
2.1.1 软件度量的基本概念	20
2.1.2 面向规模的度量	22
2.1.3 面向功能的度量	23

2.2	软件项目估算	25
2.2.1	软件项目的估算方法.....	25
2.2.2	代码行和功能点的估算.....	27
2.2.3	软件项目的经验估算模型.....	28
2.3	软件质量度量	33
2.3.1	软件质量的定义.....	33
2.3.2	软件质量的度量模型.....	33
2.4	软件复杂性度量	37
2.4.1	软件复杂性的概念及度量原则.....	37
2.4.2	McCabe 度量模型	38
2.4.3	Halstead 度量模型.....	39
2.5	软件可靠性度量	40
2.5.1	软件可靠性的有关概念.....	40
2.5.2	软件可靠性的估算.....	42
2.6	软件开发过程的管理	43
2.6.1	软件开发项目管理过程.....	44
2.6.2	风险分析	44
2.6.3	进度安排	48
2.6.4	软件质量保证.....	51
2.6.5	软件项目组织的建立与人员分工.....	53
2.6.6	软件项目的跟踪与控制.....	56
2.6.7	软件开发标准.....	56
	习题	57
第3章	计算机系统工程	59
3.1	计算机系统的概念	59
3.1.1	硬件工程	60
3.1.2	软件工程	60
3.1.3	人机工程	63
3.1.4	数据库工程	63
3.2	可行性研究	64
3.2.1	可行性研究的任务及步骤.....	64
3.2.2	经济可行性研究.....	65
3.2.3	技术可行性研究.....	67
3.2.4	选择方案	67
3.3	系统模型	68
3.4	系统规格说明与评审	71
3.4.1	系统规格说明.....	71

3.4.2 系统规格说明的评审.....	71
习题.....	72
第 4 章 需求分析.....	73
4.1 需求分析的任务.....	73
4.2 需求分析的一般性技术.....	74
4.2.1 初步需求获取技术.....	74
4.2.2 需求建模技术.....	76
4.2.3 快速原型技术.....	76
4.2.4 问题分解与抽象、多视点分析技术.....	77
4.3 需求规格说明与评审.....	78
4.3.1 需求规格说明书的作用与内容.....	78
4.3.2 需求评审.....	79
习题.....	80
第 5 章 面向数据流的分析方法.....	82
5.1 数据流图与数据字典.....	82
5.2 实体-关系图.....	85
5.2.1 数据对象、属性与关系.....	85
5.2.2 实体-关系图.....	86
5.3 基于数据流的分析方法.....	86
5.3.1 创建数据流模型.....	86
5.3.2 过程规格说明.....	88
5.4 基于数据流图的需求分析 CASE 工具.....	88
5.4.1 核心思想.....	89
5.4.2 基于 CASE 工具的需求分析.....	90
习题.....	91
第 6 章 面向数据的分析方法与形式化方法.....	92
6.1 面向数据结构的系统开发方法.....	92
6.1.1 Warnier 图.....	92
6.1.2 DSSD 方法.....	93
6.2 Jackson 系统开发方法.....	96
6.2.1 标识实体与行为.....	96
6.2.2 生成实体结构图.....	96
6.2.3 创建软件系统模型.....	97
6.3 形式化方法简介.....	99
6.3.1 形式方法的引入.....	99
6.3.2 形式方法的主要思想.....	100
6.3.3 形式化方法的分类.....	101

6.3.4	软件形式开发方法.....	101
6.3.5	形式方法的优缺点.....	101
	习题.....	102
第 7 章	面向对象的需求分析.....	103
7.1	面向对象的概念.....	103
7.1.1	对象 (Object).....	103
7.1.2	类 (Class).....	104
7.1.3	继承 (Inheritance).....	104
7.1.4	聚合 (Aggregation).....	104
7.1.5	消息 (Message).....	104
7.2	面向对象的分析方法.....	105
7.2.1	OMT 方法.....	105
7.2.2	Booch 方法.....	106
7.2.3	OOSE 方法.....	106
7.2.4	Coad/Yourdon 的 OOA/OOD 方法.....	107
7.3	面向对象的需求分析 CASE 工具.....	113
7.3.1	视.....	114
7.3.2	UML 的图.....	114
7.3.3	银行系统的需求分析.....	115
7.4	Rational 统一过程.....	117
7.4.1	时间轴.....	117
7.4.2	workflow.....	118
7.4.3	微过程的划分.....	119
	习题.....	119
第 8 章	软件设计基础.....	120
8.1	软件设计基本概念.....	120
8.1.1	软件设计过程.....	120
8.1.2	抽象与逐步求精.....	121
8.1.3	模块化与信息隐藏.....	123
8.1.4	软件总体结构设计.....	126
8.1.5	数据结构设计.....	127
8.1.6	软件过程设计.....	127
8.2	软件过程设计技术和工具.....	128
8.2.1	结构化程序设计.....	128
8.2.2	图形表示法.....	129
8.2.3	判定表与判定树.....	130
8.2.4	过程设计语言 (PDL).....	132

8.3 设计规格说明与评审	133
习题	136
第 9 章 面向数据流的设计方法	138
9.1 SD 方法的设计过程及有关概念	138
9.2 变换分析	139
9.3 事务分析	145
9.4 设计优化及原则	148
9.4.1 启发式设计策略	148
9.4.2 设计优化原则	150
习题	151
第 10 章 面向数据的设计方法	153
10.1 面向数据设计的概念	153
10.2 Jackson 系统开发方法	154
10.2.1 JSD 分析技术回顾	154
10.2.2 扩充功能性过程	154
10.2.3 施加时间约束	159
10.2.4 实现与 JSP 技术	159
10.2.5 产生过程表示	161
10.3 基于结构化数据的系统开发 (DSSD) 方法	162
10.3.1 DSSD 设计步骤	162
10.3.2 推导输出数据的逻辑结构	163
10.3.3 推导处理过程的逻辑结构	164
10.3.4 复杂过程逻辑的描述	165
习题	166
第 11 章 面向对象的设计方法	168
11.1 面向对象设计的概念	168
11.1.1 对象、操作和消息	168
11.1.2 类、实例和继承	168
11.2 面向对象的设计方法	169
11.2.1 问题域部件 (PDC) 的设计	169
11.2.2 用户界面部件 (HIC) 的设计	172
11.2.3 任务管理部件 (TMC) 的设计	174
11.2.4 数据管理部件 (DMC) 的设计	175
习题	177
第 12 章 人机界面设计	178
12.1 人的因素	178
12.1.1 人类感知基础	178

12.1.2 用户的技能	179
12.1.3 任务与用户的特殊要求.....	179
12.2 人机界面风格	180
12.3 人机界面设计过程	180
12.3.1 界面设计的有关模型.....	181
12.3.2 任务分析与建模.....	181
12.3.3 界面设计的一般问题.....	182
12.3.4 实现工具	182
12.4 人机界面实现的原则与标准	183
12.4.1 一般可交互性.....	183
12.4.2 信息显示	183
12.4.3 数据输入	184
12.4.4 人机界面标准.....	184
习题	184
第 13 章 程序设计语言与编码	186
13.1 程序设计语言	186
13.1.1 程序设计语言的特点.....	186
13.1.2 程序设计语言的发展和分类.....	187
13.1.3 程序设计语言的选择标准.....	188
13.2 编程质量与程序设计语言	189
13.3 编程标准	190
13.4 编程风格	192
13.5 程序设计支持环境 (PSE)	193
习题	195
第 14 章 软件测试	196
14.1 软件测试的基本知识	196
14.1.1 软件测试的目标与原则.....	196
14.1.2 软件测试的常用方法.....	197
14.1.3 测试阶段的信息流.....	198
14.1.4 软件测试的步骤.....	199
14.1.5 软件测试中常见的错误类型.....	200
14.2 软件测试技术	201
14.2.1 白盒测试	201
14.2.2 黑盒测试	207
14.3 软件测试过程	210
14.3.1 单元测试	210
14.3.2 集成测试	212

14.3.3	验收测试	215
14.3.4	系统测试	217
14.4	调试	218
14.4.1	调试的概念	218
14.4.2	调试策略	218
14.5	测试工具	219
14.5.1	自动测试工具	219
14.5.2	调试工具	220
习题		220
第 15 章	软件维护	221
15.1	软件维护的基本内容和分类	221
15.2	软件维护的特点	222
15.2.1	结构化维护与非结构化维护	222
15.2.2	维护成本	223
15.2.3	维护可能存在的问题	224
15.3	软件维护过程	224
15.3.1	维护组织	224
15.3.2	维护的报告与审核	225
15.3.3	维护过程的事件流	225
15.3.4	保存维护记录	227
15.3.5	评价维护活动	227
15.4	维护的副作用	228
15.5	软件的可维护性	229
15.5.1	影响可维护性的因素	229
15.5.2	可维护性的度量	229
15.5.3	可维护性复审	230
习题		230
第 16 章	软件配置管理	231
16.1	软件配置管理的概念	231
16.1.1	基线技术	231
16.1.2	软件配置项	232
16.2	软件配置管理的任务及标准	234
16.2.1	标识配置对象	234
16.2.2	版本控制	235
16.2.3	修改控制	235
16.2.4	配置审计	237
16.2.5	配置状况报告	237

16.2.6 软件配置管理标准.....	238
16.3 配置管理的 CASE 工具	238
习题	239
第 17 章 软件开发新技术简介	240
17.1 软件重用技术简介	240
17.1.1 软件重用的概念和过程.....	240
17.1.2 构造软部件库.....	240
17.1.3 软部件的重用.....	245
17.1.4 软件重用项目的管理.....	246
17.2 Internet 与 Java 简介	246
17.2.1 Internet 简介	246
17.2.2 Java 技术简介	247
17.3 对象链接与嵌入技术简介	250
17.3.1 复合文档的概念.....	250
17.3.2 实现复合文档的方法——链接和嵌入.....	251
17.3.3 COM 是实现链接和嵌入的技术基础.....	252
17.3.4 ActiveX.....	252
17.4 分布式计算技术简介	253
17.4.1 分布式技术概述.....	253
17.4.2 主流分布式计算技术.....	254
习题	255
参考文献	256

第1章 概述

随着微电子技术和计算机技术的迅速发展,计算机硬件的性能和质量也在迅速提高,而其体积、功耗、成本却在不断下降。据统计,计算机的性能平均每18个月就可以提高一倍。与此同时,随着计算机应用的日益普及和深化、计算机与 Internet 互联网的连接,人类已经进入了以计算机为核心的信息社会。在信息社会中,信息的获取、加工处理和使用都需要大量高质量的软件。多年来,和计算机硬件相比,计算机软件的开发效率却远远跟不上普及计算机应用的需求,软件成本在逐年上升,而质量却难以得到可靠的保证,计算机软件已经成为限制计算机系统发展的关键因素。特别是对于大型软件的开发,人们往往显得力不从心,致使进度一拖再拖、成本失去控制、软件质量得不到保证、所开发的软件难以维护。为了扭转这种被动局面,自20世纪60年代末以来,人们开始重视软件开发方法、工具和环境的研究,从而逐步形成了“计算机软件工程学”这一计算机科学技术领域中的新兴学科,通常简称为“软件工程”。

本章主要介绍软件和软件工程的基本概念,包括软件、软件工程、软件生命周期、软件开发模型、软件开发方法、工具和环境等。

1.1 软件

1.1.1 软件

计算机软件是指与计算机系统操作有关的程序、数据以及任何与之有关的文档资料。也就是说,软件由两部分组成:其一是机器可执行的程序和数据;其二是与软件开发、运行、维护、使用 and 培训等有关的文档资料。

1. 软件的特点

硬件属于物理产品,而软件则属于逻辑产品。软件的开发、使用和维护与硬件相比存在很大的差别。

① 软件开发与硬件研制相比,更依赖于开发人员的业务素质和智力,以及开发人员的组织、合作和管理。在大多数场合,软件的开发几乎都是从头开始的,而且开发的成本、进度很难估计。

② 一般大型软件(比如 Windows 操作系统)在提交使用之前,尽管经过了严格的测试和试用,但仍然存在着潜伏的错误。而硬件产品经过严格的测试、试验、试用后,设计过程中的错误一般都是可以排除的。

③ 软件产品开发成功后,只需对原版软件进行复制,即可生产出任意多的同样的产品;而硬件产品试制成功后,要批量生产还需建厂房和生产线,投入大量的人力、物力和财力,并且需要对每件产品的质量进行控制和检验。

④ 软件在使用过程中的维护工作比硬件要复杂得多。为了修正软件中潜藏的错误，提高软件的可维护性和可靠性，完善软件的性能以适应新的软硬件环境，就需要对软件不断进行修改。由于软件的内部逻辑关系复杂，很难维护，而且在维护的过程中还可能产生新的错误，因此，软件产品的维护工作远比硬件产品的维护复杂。

⑤ 由于软件不是物理产品，所以它不会磨损和老化。一个高质量的软件，只要需要，就可以长期使用下去。

软件的上述特性已经构成了一种特殊的文化，即“软件文化”。

2. 软件的发展

自从 20 世纪 40 年代第一台电子计算机问世以来，软件的发展可以划分为四个阶段。

第一阶段是 20 世纪 60 年代中期以前，这是一个计算机系统开发的初期阶段。在这一阶段，硬件经历了由电子管计算机到晶体管计算机的变革，而软件还没有系统化的开发方法。计算机系统软件多数是用户自己设计、自己使用、自己维护，软件开发处于个体化生产状态。由于计算机的硬件价格昂贵、体积大、运算速度低、内存容量小，程序设计的目标主要集中在如何提高时空效率上。程序员为减少一条指令、少占一个内存单元而大动脑筋。程序设计过多地依靠程序员的“技巧”，所开发的“软件”文档不全、开发过程的中间成果大都保留在程序员的记忆中，他人无法维护。初期阶段的大多数计算机系统采用批处理方式进行人机之间的联系。

第二阶段是从 20 世纪 60 年代中期到 20 世纪 70 年代中期。软件开发已进入了作坊式的生产方式，即出现了“软件车间”。硬件经历了由晶体管计算机到中小规模集成电路计算机的变革，由于 CPU 速度和内存容量的提高，为了有效地利用计算机系统资源，引进了多用户、多道程序和人机交互等工作方式，出现了实时系统。由于辅助存储设备的发展，出现了第一代数据库管理系统。软件开发开始形成产品。由于软件是逻辑产品，其开发、管理和维护十分复杂，致使供求矛盾加剧，到 20 世纪 60 年代末期，“软件危机”变得十分严重。

第三阶段是从 20 世纪 70 年代中期到 20 世纪 80 年代末期。软件开发进入了产业化生产，即出现了众多大型的“软件公司”。在这一阶段，硬件经历了由中小规模计算机到大规模计算机的变革，致使微处理器、个人计算机、工作站具有相当高的性能价格比，并广泛应用于生产、生活的各个领域。与此同时，分布式系统、计算机网络、嵌入式计算机系统有了很大发展。所有这些都推动着软件产业的发展，软件开发开始采用了“工程”的方法，软件产品急剧增加，质量也有了很大的提高。

第四阶段是从 20 世纪 80 年代末期开始的。这是一个软件产业大发展的时期。在这一时期，人工智能、专家系统、人工神经网络软件开始走向实际应用。多媒体技术的发展使计算机可以同时处理文字、声音、图形和图像，从而提高了计算机的信息处理能力，扩大了计算机的应用领域。多处理机系统的发展促进了多指令流多数据流（MIMD）计算机系统并行算法的研究和并行软件的开发。Internet 的高速发展推动了在 Internet 环境下的软件开发技术和分布计算环境下的软件互操作技术的发展，从而涌现出了如 Java、ASP.NET 等新的编程语言和 CORBA（通用对象请求代理结构）标准等。这一阶段也是软件工程大发展的时期，人们开始采用面向对象的技术和可视化的集成开发环境。

1.1.2 软件危机

软件危机是指在计算机软件开发、使用与维护过程中遇到的一系列严重问题和难题。

1. 软件危机的表现

应该说,自从计算机诞生以来,软件危机就一直存在,软件危机主要表现在以下几个方面:

(1) 人们对软件开发的成本和进度的估计常常不够准确。由于软件的特殊性,不同类型的软件其开发所需的工作量、成本往往差别很大。因此,常常出现实际成本比估算成本高出一个数量级,实际进度比计划进度拖延几个月甚至几年的现象,这大大降低了开发商的信誉,也引起用户不满。

(2) 用户对已完成的软件不满意的现象时有发生。这主要是由于软件开发者和用户没有进行充分的交流和沟通,开发者与用户对软件功能的理解有偏差,从而开发出不实用或不能完全符合用户需求的软件系统。

(3) 软件产品的质量往往是靠不住的。由于软件可靠性和软件质量的定量计算还没有确切可信的方法,加上开发成本和进度的限制,软件质量保证技术(复审、测试等)很难完全贯彻到软件开发的全过程中,因此,软件的质量难以保证。

(4) 软件常常是不可维护的。许多软件在开发时就没有考虑到将来的修改。因此,很多软件中的错误是非常难改正的。实际上,不可能使这些软件适应新的硬件环境,也无法根据用户的需要在原有的软件中增加一些新的功能,软件的维护工作是相当困难的工作。

(5) 软件通常没有适当的文档资料。软件开发一般没有严格的文档制度,软件开发结束后,保留下来的有效的文档资料很少,特别是与软件版本相匹配的文档资料就更少。需要指出,软件文档资料是开发人员自己及其与用户之间交流信息的依据,也是软件维护人员进行维护的依据,更是软件管理人员对软件开发工程进行管理和评价的根据。文档资料不全或不合格,必将给软件开发和维护工作带来许多难以想象的困难和难以解决的问题。

(6) 软件成本在计算机系统总成本中所占的比例逐年上升。随着微电子技术的迅速发展,计算机硬件成本逐年下降。与此相反,随着对软件规模和数量需求的不断增长,软件开发需要大量的人力、物力和财力,从而使软件成本占计算机系统总成本的比例逐年上升。特别是软件维护成本迅速增加,已经占软硬件总成本的40%~75%,如图1-1所示。

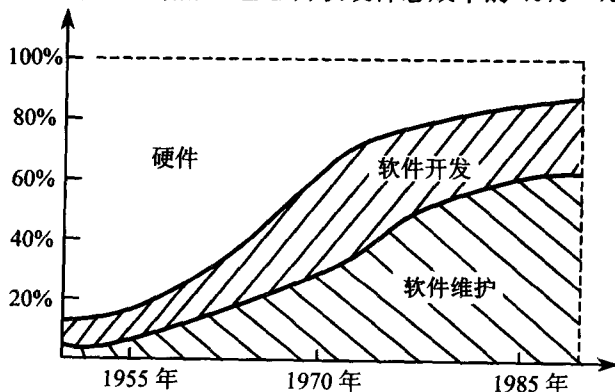


图 1-1 软件、硬件成本变化趋势

(7) 软件开发生产率提高的速度远跟不上日益增长的软件需求。

2. 软件危机的产生原因

造成软件危机的原因很多, 其中主要有以下几个:

(1) 用户对软件需求的描述不精确。在软件开发的初期, 用户对目标软件的功能、性能很可能不十分了解, 因此对软件需求的描述可能有遗漏、二义性或错误, 到了软件开发中期, 用户才提出新的要求, 这时的变动将面临巨大的工作量和相当复杂的逻辑关系, 付出的代价巨大。如果到软件完成了再提出修改, 其代价比初期要高出 2~3 个数量级。

(2) 软件开发人员对用户需求的理解有偏差。软件开发人员很可能并不十分熟悉用户的业务, 因此对用户需求的理解有偏差, 这将导致软件产品与用户的需求不一致。

(3) 缺乏处理大型软件项目的经验。开发大型软件项目需要组织众多人员共同完成。一般来说, 多数管理人员缺乏大型软件的开发经验, 而多数软件开发人员又缺乏大型软件项目的管理经验, 致使各类人员的信息交流不及时、不准确、容易产生误解。

(4) 开发大型软件易产生疏漏和错误。大型软件逻辑关系复杂、开发周期长、涉及人员多, 因此大型软件产品中很可能存在疏漏和潜伏错误。

(5) 缺乏有力的方法学的指导和有效的开发工具的支持, 导致软件开发过多地依靠程序员的“技巧”, 从而加剧了软件产品的个性化。

(6) 面对日益增长的软件需求, 人们显得力不从心。目前, 需要开发的软件越来越大, 逻辑关系越来越复杂, 致使人们无力解决“复杂问题”。从某种意义上说, 解决供求矛盾将是一个永恒的主题。

3. 缓解软件危机的途径

到了 20 世纪 60 年代末期, 软件危机已相当严重。这促使计算机科学家们开始探索缓解软件危机的方法。他们提出了“软件工程”的概念, 即用现代工程的原理、技术和方法进行软件的开发、管理、维护和更新。于是, 开创了计算机科学技术的一个新的研究领域——软件工程。几十年来, 软件工程在软件开发方法、工具和管理等方面的研究与应用已经取得了很多成果, 并已大大缓解了软件危机所带来的影响。当然, 要进一步解决软件危机, 还有相当多的工作要做。

1.2 软件工程的概

1.2.1 软件工程的定义

1968 年, 北大西洋公约组织在原西德召开计算机科学会议, 在此次会议上 Fritz Bauer 首次提出了“软件工程”的概念。

软件工程是指用工程、科学和数学的原则与方法开发、维护计算机软件的有关技术和管理方法。软件工程由方法、工具和过程三部分组成, 称为软件工程的三要素。软件工程中的各种方法是完成软件工程项目的手段, 它们支持软件工程的各个阶段。软件工程使用的软件工具能够自动或半自动地支持软件的开发、管理和文档的生成。软件工程中的