



普通高等教育“十一五”国家级规划教材

# 软件工程——原理、 方法与应用 (第3版)

Software Engineering: Principles, Methods  
and Applications (3rd Edition)

史济民 顾春华 郑红 编著



高等教育出版社  
Higher Education Press

普通高等教育“十一五”国家级规划教材

# 软件工程

## ——原理、方法与应用（第3版）

史济民 顾春华 郑红 编著

高等教育出版社

## 内容提要

作为一本注重能力培养的实践型教材,第3版继承并保持了“注重实践”的风格,将全书分编为绪论和上、中、下3篇,共14章。内容分别为:上篇为传统软件工程,包括软件生存周期与软件过程、结构化分析与设计;中篇为面向对象软件工程,包括面向对象与UML、需求工程与需求分析、面向对象分析、面向对象设计、编码与测试;下篇为软件工程的近期进展、管理与环境,包括软件维护、软件复用、软件工程管理、软件质量管理、软件工程环境、软件工程高级课题。与第2版相比,本版将“并行介绍传统的和面向对象的软件工程”改变为“重点介绍面向对象的软件工程”,并在“面向对象分析”和“面向对象设计”两章分别给出案例;通过“软件工程高级课题”,对Web工程及基于体系结构的软件开发等热门课题集中进行了简要的讲解。

本书结构合理、文字通俗、例题丰富、可读性强,主要面向计算机及相关专业本科生,亦可供软件开发人员作为参考。

### 图书在版编目(CIP)数据

软件工程:原理、方法与应用 / 史济民,顾春华,郑红  
编著. —3版. —北京:高等教育出版社,2009.3

ISBN 978-7-04-026146-2

I. 软… II. ①史…②顾…③郑… 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(2009)第010893号

策划编辑 倪文慧 责任编辑 张海波 封面设计 张志奇 责任绘图 尹莉  
版式设计 范晓红 责任校对 杨凤玲 责任印制 宋克学

出版发行 高等教育出版社  
社 址 北京市西城区德外大街4号  
邮政编码 100120  
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司  
印 刷 高等教育出版社印刷厂

开 本 787×1092 1/16  
印 张 24.25  
字 数 540 000

购书热线 010-58581118  
免费咨询 800-810-0598  
网 址 <http://www.hep.edu.cn>  
<http://www.hep.com.cn>  
网上订购 <http://www.landaco.com>  
<http://www.landaco.com.cn>  
畅想教育 <http://www.widedu.com>

版 次 1995年3月第1版  
2009年3月第3版  
印 次 2009年3月第1次印刷  
定 价 28.00元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 26146-00

## 第3版前言

本书第2版自2002年12月首印,迄今已5年有半了。许多高等院校的计算机专业和软件学院相关专业采用该版书作为本科生“软件工程”课程的教材。为了更好地满足读者需要,我们决定编写第3版,并确定了如下的编写方针:

### 一、继续保持“注重实践”的风格

软件工程具有很强的实践性,但早期的软件工程教材往往偏重于理论,介绍原理有余,联系应用不足。由于缺乏可供借鉴的示例,使读者在具体开发时常有不知从何处入手的感觉。本书取名《软件工程——原理、方法与应用》,就是希望从“应用”出发,兼顾“原理”与“方法”两个方面:讲解方法时精选例题,方便模仿;上升到原理时提纲挈领,画龙点睛。目的是让读者一方面掌握软件工程的常用方法及其具体操作,另一方面又提升到以原理为指导,不致被具体方法中繁琐的细节所淹没。从本书第1版起,编者就参照国际知名教材《软件工程:实践者的方法》(Pressman 著)的做法,广举例题,注重实践,因而取得第1版连续发行12年近12万册,第2版在5年半中重印12次、累计发行14.1万册的好成绩。本版保持了上述风格,力求使之名副其实地成为理论结合实际、注重能力培养的实践型教材。

### 二、重点讲解面向对象软件工程

从20世纪80年代中期以来,面向对象软件工程发展迅速,主要技术已基本成熟,成为软件开发的主流范型。因此在本版编写中,将第2版“并行介绍传统的和面向对象的软件工程”,改变为“重点介绍面向对象的软件工程”。一方面,删去了Jackson与LCP等现已基本不用的方法,着重讲述以“结构化分析与设计”为代表的“结构化软件工程”技术;另一方面,把面向对象软件工程的内容由第2版的第6、7两章扩充为“面向对象与UML”、“面向对象分析”(OOA)和“面向对象设计”(OOD)等3章,并在“面向对象分析”和“面向对象设计”两章分别给出案例。为了更好反映软件工程的近期发展,除“软件复用”仍在第10章设置专章讨论外,还增编了第14章“软件工程高级课题”,对Web工程及基于体系结构的软件开发等当前的热门课题集中进行简要的讲解。

### 三、着重介绍软件工程技术,兼顾管理与环境

作为软件工程的综合性入门教材,本版同前两版一样主要介绍软件工程技术,兼讲一点管理与环境方面的知识,包括项目管理、项目度量、风险预测、I-CASE环境等。

#### 四、定位与主要读者对象

经过 40 年的实践,软件工程的基本原则现已被产业界广泛接受。高等院校计算机专业普遍开设了“软件工程”课程,有些院校在非计算机专业中也设置了相关的课程。本书主要定位于计算机及相关学科各专业的本科学生,亦可供软件学院研究生及软件开发人员作为参考。全书结构合理、文字通俗、例题丰富、可读性强。但由于软件工程日新月异,新技术层出不穷,编者水平有限,疏漏难免,诚恳欢迎读者和专家批评指正。

本版由史济民主编并统稿,顾春华博士和郑红博士参与策划和编写。全书 14 章,除新编的 3 章由史济民(第 3 章)、顾春华(第 5 章)、郑红(第 14 章)分别编写外,其余 11 章也由 3 人分工修改,其中史济民改写了第 1 章,顾春华改写了第 2、4、6、7、8 等 5 章,郑红改写了第 9~13 等 5 章。杨晶、郑姜和罗珍等 3 位同志认真阅读和校对了书稿,参加了部分习题的讨论,并且从学生的角度提出不少有益的建议。

浙江大学软件学院副院长、国家级精品课程主讲人陈越教授在百忙中审查了本版书稿,并提出了宝贵的意见和建议。借此机会,表示由衷的感谢!

编 者

2008 年 6 月于华东理工大学

## 第2版前言

本书第一版自1990年问世以来,已经过了12年。在此期间,软件工程从第一代传统的软件工程发展为第二代面向对象的软件工程,现正向基于软件复用的第三代软件工程发展。在高等学校中,软件工程不仅已成为计算机专业学生的必修课,而且在非计算机专业的“软件技术基础”等公共课中也上升为必学的内容。

作为把软件工程推广到高校非计算机专业的一种尝试,本书第一版曾被收入高等教育出版社的“高校计算机基础教育系列教材”,但实际上多数高校仍把它用作计算机专业的教材。为了适应国内软件工程当前教学的需要,这次改版将本书仍定位为计算机专业的本科教材,并且确定了如下的编写方针。

### 一、继续保持“注重实践”的风格

软件工程具有很强的实践性,但由于例题难选,多数教材往往偏重于理论。本书第一版参考国际知名教材 Pressman 著的《软件工程:实践者的方法》的做法,广举实例,注重实践,因而受到读者的欢迎。第一版连续发行12年,累计印数近12万册,并于1995年荣获上海市高校优秀教材二等奖。本版保持了上述风格,使之名副其实地成为原理、方法与应用紧密结合的教材。

### 二、平行讲解第一、二两代软件工程

经过20世纪90年代的发展,面向对象软件工程已渐趋成熟,在许多应用领域取代了传统的软件工程。但是,传统的、基于结构化程序设计的软件工程并未退出历史舞台,其基本原理与方法有不少仍在新一代软件工程中继续应用。为此,本书将两类软件工程平行讲解,既方便读者对照,又便于展示两者“你中有我、我中有你”的关系。

### 三、充分反映软件工程近十几年的发展

本次改版增加了许多新方法、新模型的介绍,例如面向对象软件工程的UML语言、软件复用、质量认证标准、净室工程和软件容错技术等。对于代表新的发展方向的软件构件工程、软件过程工程等新技术,均设置专章或专节进行讲解,以突出其重要性。但鉴于本书读者对象主要是高校的本、专科学生,对于还处于形成阶段的有些新技术,本教材着重阐明它们的基本概念、产生背景和主要作用,不展开细节的讨论。

### 四、重点讲解软件工程技术,兼顾管理与环境

作为软件工程的综合性入门教材,本版同第一版一样主要讲解软件工程技术,兼讲一点管理与环境方面的知识。后者的重点放在质量管理与I-CASE环境上,以便读者对软件工程

的整体情况与近期发展有一比较全面的理解。

本书由史济民、顾春华共同策划。第一章由史济民改写；第二、三、四、十一、十二、十五章由顾春华改写；第五、八、九、十三章由李昌武改写。新增加的4章，分别由顾春华（第六、七章及第14.5、14.6节）、苑荣（第十章）和史济民（第十四章其余各节）编写。全书由史济民统一修改定稿。华东理工大学宋国新教授十分支持本次改版，除承担书稿审阅外，还在经费和编写人员时间安排上给予帮助。对此编者表示由衷的感谢。

由于软件工程覆盖面宽，发展又很迅速，编者水平有限，疏漏难免，诚恳希望读者不吝指正。

2002年5月于上海

## 郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

**反盗版举报电话：**(010) 58581897/58581896/58581879

**反盗版举报传真：**(010) 82086060

**E-mail：**dd@hep.com.cn

**通信地址：**北京市西城区德外大街4号

高等教育出版社打击盗版办公室

**邮 编：**100120

**购书请拨打电话：**(010) 58581118



# 目 录

第 1 章 绪论	1	1.3 软件工程的发展	6
1.1 软件和软件危机	1	1.3.1 3 种编程范型	7
1.1.1 软件的定义	1	1.3.2 3 代软件工程	9
1.1.2 软件的特征	2	1.4 软件工程的应用	10
1.1.3 软件危机	2	1.4.1 在各种规模软件 开发中的应用	10
1.2 软件工程学的范畴	4	1.4.2 软件工程的成就与 发展展望	12
1.2.1 软件开发方法学	5	1.5 软件工程的教學: 本书导读	13
1.2.2 软件工具	5	小结	15
1.2.3 软件工程环境	6	习题	15
1.2.4 软件工程管理	6		

## 上篇 传统软件工程

第 2 章 软件生存周期与软件过程	19	2.5.1 统一过程	31
2.1 软件生存周期	19	2.5.2 敏捷过程	32
2.1.1 软件生存周期的主要活动	19	2.5.3 极限编程	33
2.1.2 生存周期与软件过程的关系	20	2.6 软件可行性研究	35
2.2 传统的软件过程	21	2.6.1 可行性研究的内容与步骤	35
2.2.1 瀑布模型	21	2.6.2 软件风险分析	36
2.2.2 快速原型模型	23	2.6.3 项目实施计划	39
2.3 软件演化模型	24	小结	40
2.3.1 增量模型	24	习题	40
2.3.2 螺旋模型	25		
2.3.3 构件集成模型	27	第 3 章 结构化分析与设计	41
2.4 形式化方法模型	28	3.1 概述	41
2.4.1 转换模型	28	3.1.1 结构化分析与设计的由来	41
2.4.2 净室模型	29	3.1.2 SA 模型的组成与描述	43
2.5 统一过程和敏捷过程	31	3.1.3 SD 模型的组成与描述	50
		3.2 结构化系统分析	51

3.2.1	画分层数据流图	51	3.3.5	优化初始 SC 图的指导规则	62
3.2.2	确定数据定义与加工策略	54	3.3.6	教材购销系统的总体结构	63
3.2.3	需求分析的复审	55	3.4	模块设计	65
3.3	结构化系统设计	55	3.4.1	目的与任务	65
3.3.1	SD 概述	55	3.4.2	模块设计的原则与方法	66
3.3.2	SD 的步骤: 从 DFD 图 到 SC 图	57	3.4.3	常用的表达工具	68
3.3.3	变换映射	58	小结		70
3.3.4	事务映射	60	习题		71

## 中篇 面向对象软件工程

第 4 章	面向对象与 UML	75	小结		104
4.1	面向对象概述	75	习题		104
4.1.1	对象和类	75	第 5 章	需求工程与需求分析	105
4.1.2	面向对象的基本特征	76	5.1	软件需求工程	105
4.1.3	面向对象开发的优点	78	5.1.1	软件需求的定义	105
4.2	UML 简介	78	5.1.2	软件需求的特性	106
4.2.1	UML 的组成	79	5.1.3	需求工程的由来	107
4.2.2	UML 的特点	82	5.2	需求分析与建模	108
4.2.3	UML 的应用	83	5.2.1	需求分析的步骤	108
4.3	静态建模	83	5.2.2	需求分析是迭代过程	109
4.3.1	用例图与用例模型	84	5.3	需求获取的常用方法	110
4.3.2	类图和对象图	85	5.3.1	常规的需求获取方法	110
4.3.3	包	92	5.3.2	用快速原型法获取需求	111
4.4	动态建模	92	5.4	需求模型	112
4.4.1	消息	92	5.4.1	需求模型概述	112
4.4.2	状态图	93	5.4.2	面向对象的需求建模	114
4.4.3	时序图和协作图	95	5.5	软件需求描述	123
4.4.4	活动图	98	5.6	需求管理	123
4.5	物理架构建模	99	5.6.1	需求管理的内容	124
4.5.1	物理架构	99	5.6.2	需求变更控制	125
4.5.2	构件图和部署图	100	5.6.3	需求管理工具	128
4.6	UML 工具	101	5.7	需求建模示例	128
4.6.1	Rational Rose	101	5.7.1	问题陈述	129
4.6.2	StarUML	103	5.7.2	用例模型	130

5.7.3 补充规约 .....	134	7.3.3 任务管理策略 .....	179
5.7.4 术语表 .....	135	7.3.4 分布式实现机制 .....	182
小结 .....	136	7.3.5 数据存储设计 .....	185
习题 .....	136	7.3.6 人机交互设计 .....	188
<b>第 6 章 面向对象分析</b> .....	<b>138</b>	<b>7.4 系统元素设计</b> .....	<b>189</b>
6.1 软件分析概述 .....	138	7.4.1 子系统设计 .....	189
6.1.1 面向对象软件分析 .....	138	7.4.2 分包设计 .....	192
6.1.2 面向对象分析模型 .....	140	7.4.3 类/对象设计 .....	195
6.2 面向对象分析建模 .....	141	<b>7.5 面向对象设计示例</b> .....	<b>204</b>
6.2.1 识别与确定分析类 .....	141	7.5.1 关联关系的具体化 .....	205
6.2.2 建立对象-行为模型 .....	144	7.5.2 网上购物系统的 架构设计 .....	206
6.2.3 建立对象-关系模型 .....	147	7.5.3 网上购物系统的类/ 对象设计 .....	207
6.3 面向对象分析示例 .....	149	小结 .....	210
6.3.1 注册 .....	149	习题 .....	211
6.3.2 维护个人信息 .....	151	<b>第 8 章 编码与测试</b> .....	<b>212</b>
6.3.3 维护购物车 .....	153	8.1 编码概述 .....	212
6.3.4 生成订单 .....	157	8.1.1 编码的目的 .....	212
6.3.5 管理订单 .....	159	8.1.2 编码的风格 .....	213
小结 .....	164	8.2 编码语言与编码工具 .....	216
习题 .....	164	8.2.1 编码语言的发展 .....	216
<b>第 7 章 面向对象设计</b> .....	<b>165</b>	8.2.2 常用的编程语言 .....	218
7.1 软件设计概述 .....	165	8.2.3 编码语言的选择 .....	220
7.1.1 软件设计的概念 .....	165	8.2.4 编码工具 .....	222
7.1.2 软件设计的任务 .....	167	8.3 编码示例 .....	222
7.1.3 模块化设计 .....	167	8.3.1 注册功能编码实现 .....	222
7.2 面向对象设计建模 .....	173	8.3.2 维护购物车功能编码实现 .....	228
7.2.1 面向对象设计模型 .....	173	8.4 测试的基本概念 .....	233
7.2.2 面向对象设计的任务 .....	174	8.4.1 目的与任务 .....	233
7.2.3 模式的应用 .....	175	8.4.2 测试的特性 .....	234
7.3 系统架构设计 .....	176	8.4.3 测试的种类 .....	235
7.3.1 系统高层结构设计 .....	176	8.4.4 测试的文档 .....	235
7.3.2 确定设计元素 .....	177	8.4.5 软件测试过程 .....	236

8.5 黑盒测试和白盒测试	236	8.7.3 集成测试	256
8.5.1 黑盒测试	237	8.7.4 确认测试	259
8.5.2 白盒测试	240	8.7.5 系统测试	260
8.6 测试用例设计	248	8.7.6 终止测试的标准	260
8.6.1 黑盒测试用例设计	248	8.8 面向对象系统的测试	261
8.6.2 白盒测试用例设计	250	8.8.1 OO 软件的测试策略	261
8.7 多模块程序的测试策略	254	8.8.2 OO 软件测试用例设计	262
8.7.1 测试的层次性	254	小结	265
8.7.2 单元测试	255	习题	265
<b>下篇 软件工程的近期进展、管理与环境</b>			
<b>第 9 章 软件维护</b>	271	10.4 面向对象与软件复用	292
9.1 软件维护的种类	271	10.4.1 OO 方法对软件 复用的支持	292
9.2 软件可维护性	272	10.4.2 复用技术对 OO 方法的支持	293
9.3 软件维护的实施	274	10.4.3 基于构件软件开发的 现状与问题	295
9.4 软件维护的管理	275	小结	295
9.5 软件配置管理	277	习题	296
9.6 软件再工程	279	<b>第 11 章 软件工程管理</b>	297
小结	281	11.1 管理的目的与内容	297
习题	282	11.2 软件估算模型	298
<b>第 10 章 软件复用</b>	283	11.2.1 资源估算模型	298
10.1 软件复用的基本概念	283	11.2.2 COCOMO 模型	300
10.1.1 软件复用的定义	283	11.3 软件成本估计	302
10.1.2 软件复用的措施	284	11.4 人员的分配与组织	308
10.1.3 软件复用的粒度	285	11.5 项目进度安排	311
10.2 领域工程	286	小结	314
10.2.1 横向复用和纵向复用	286	习题	315
10.2.2 实施领域分析	287	<b>第 12 章 软件质量管理</b>	316
10.2.3 开发可复用构件	287	12.1 从质量保证到质量认证	316
10.2.4 建立可复用构件库	289	12.2 质量保证	317
10.3 基于构件的软件开发	290		
10.3.1 构件集成模型	291		
10.3.2 应用系统工程	291		

12.2.1 软件的质量属性 .....	317	13.2.1 CASE 的组成构件 .....	342
12.2.2 质量保证的活动内容 .....	318	13.2.2 CASE 的一般结构 .....	343
12.3 软件可靠性 .....	319	13.3 CASE 环境实例 .....	345
12.3.1 可靠性的定义和分级 .....	319	13.3.1 Rational SUITE Enterprise Studio .....	346
12.3.2 可靠性模型 .....	321	13.3.2 青鸟系统 .....	351
12.3.3 软件容错技术 .....	323	小结 .....	351
12.4 程序正确性证明 .....	326	习题 .....	352
12.5 CMM 软件能力成熟度模型 .....	328	<b>第 14 章 软件工程高级课题 .....</b>	<b>353</b>
12.5.1 CMM 的基本概念 .....	328	14.1 Web 工程 .....	353
12.5.2 软件能力成熟度等级 .....	329	14.1.1 Web 工程与软件工程 .....	354
12.5.3 CMM 的应用 .....	330	14.1.2 Web 开发 .....	354
12.5.4 CMM 评估的实施 .....	331	14.2 基于体系结构的软件开发 .....	358
12.5.5 软件过程评估的 SPICE 国际标准 .....	331	14.2.1 应用软件的体系结构 .....	359
12.6 ISO 9000 国际标准 .....	332	14.2.2 编程范型对体系 结构的影响 .....	360
12.6.1 ISO 9001 和 ISO 9000-3 .....	332	14.2.3 编程范型对复用 粒度的影响 .....	360
12.6.2 ISO 9000 标准对软件 企业的重要性 .....	333	14.2.4 软件体系结构技术 仍在发展 .....	360
12.6.3 在软件企业中实施 ISO 9000 标准 .....	334	14.3 面向方面的软件开发 .....	361
12.7 软件度量 .....	334	14.3.1 面向方面编程 .....	361
12.7.1 项目度量 .....	334	14.3.2 AOP 语言规范 .....	362
12.7.2 过程度量 .....	336	14.3.3 AOP 与 OOP 比较 .....	363
小结 .....	337	14.3.4 面向方面软件开发 .....	364
习题 .....	338	14.4 形式化的软件开发 .....	364
<b>第 13 章 软件工程环境 .....</b>	<b>339</b>	14.4.1 形式化方法的定义 .....	365
13.1 什么是软件工程环境 .....	339	14.4.2 形式化的软件开发 .....	366
13.1.1 软件开发环境的特点 .....	339	小结 .....	368
13.1.2 理想环境的模型 .....	341	习题 .....	369
13.1.3 CASE 环境 .....	341		
13.2 CASE 环境的组成与结构 .....	342		
附录 缩略语中英文对照表 .....	370		
主要参考文献 .....	372		

# 第1章 绪 论

1969年，美国IBM公司首次宣布除操作系统继续随计算机配送外，其余软件一律计价出售，由此开创了软件成为独立商品的先河。短短30多年间，计算机软件的重要性与日俱增。从PC到笔记本式计算机，从Internet到移动电话，从先进的武器到现代家电，计算机软件几乎无处不在，无时不在。世界上最大的软件公司Microsoft及其创始人，已分别成为全球知名度最高的企业和人物之一。在很多发达国家，软件产业已成为社会的支柱产业，软件工程师也成为最受青睐的一种职业。

随着软件产业的发展，计算机应用逐步渗透到社会生活的各个角落，使各行各业都发生了很大的变化。这同时也促使人们对软件的品种、数量、功能和质量等提出了越来越高的要求。然而，软件的规模越大、越复杂，软件开发越显得力不从心。于是，业界开始重视软件开发过程、方法、工具和环境的研究，软件工程应运而生。

本章介绍软件和软件工程的基本概念，包括软件、软件危机、软件工程学、三代软件工程及其应用等。章末讨论与软件工程教学有关的几种观点，也可视为本书的导读。

## 1.1 软件和软件危机

和计算机硬件一样，20世纪60年代以来，软件也在规模、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，它有哪些主要特征呢？

### 1.1.1 软件的定义

众所周知，计算机硬件的发展基本上遵循了摩尔定律，即每18个月芯片的性能与速度均提高一倍。软件的发展也十分惊人，例如就体系结构而言，它经历了从主机结构到文件服务器结构，从客户/服务器系统到基于Internet的服务器/浏览器结构的体系结构等变化；从编码语言来讲，它经历了从机器代码到汇编代码，从高级程序设计语言到人工智能语言等变化；从开发工具来看，它经历了从分离的开发工具（如代码编辑器、中间代码生成器和连接器）到集成的可视化开发系统，从简单的命令行调试器到方便的多功能调试器等变化。

但是在过去40余年中，软件的基本定义却并未改变。有些初学者认为软件就是程序，这个理解是不完全的。这里引用著名的美国软件工程教材作者R. S. Pressman的定义：“软件

是能够完成预定功能和性能的可执行的计算机程序，包括使程序正常执行所需要的数据，以及有关描述程序操作和使用的文档。”简而言之，可以表述为“软件 = 程序 + 文档”。

程序是为了解决某个特定问题而用程序设计语言描述的适合计算机处理的语句序列。它是由软件开发人员设计和编码的，通常要经过编译程序，才能编译成可在计算机上执行的机器语言指令序列。程序执行时一般要输入一定的数据，同时也会输出运行的结果。而文档则是软件开发活动的记录，主要供人们阅读，既可用于专业人员和用户之间的通信和交流，也可以用于软件开发过程的管理和运行阶段的维护。为了提高软件开发的效率和方便软件产品的维护，现在软件开发人员越来越重视文档的作用及其标准化工作。我国国家标准局已参照国际标准，陆续颁布了《计算机软件开发规范》、《计算机软件需求说明编制指南》、《计算机软件测试文件编制规范》、《计算机软件配置管理计划规范》等文档规范。

### 1.1.2 软件的特征

要对软件有一个全面的理解，首先要了解软件的特征。当制造硬件时，生产的结果能转换成物理的形式。如果建造一台新的计算机，从设计图纸、生产部件（VLSI 芯片、线路板、面板等）到装配原型，每一步都将演化成物理的产品。而软件却是逻辑的而不是物理的，在开发、生产、维护和使用等方面，都同硬件具有完全不同的特征。

1. 软件开发不同于硬件设计  
与硬件设计相比，软件更依赖于开发人员的业务素质、智力，以及人员的组织、合作和管理，而硬件设计与人的关系相对小一些。对硬件而言，设计成本往往只占整个产品成本的一小部分，而软件开发的成本很难估算，通常占整个产品成本的大部分，这意味着对软件开发项目不能像硬件设计项目那样来管理。

2. 软件生产不同于硬件制造  
硬件设计完成后就投入批量制造，制造也是一个复杂的过程，其间仍可能引入质量问题；而软件成为产品之后，其制造只是简单的复制而已。

3. 软件维护不同于硬件维修  
硬件在运行初期有较高的故障率（主要来源于设计或制造的缺陷），在缺陷修正后的一段时间中，故障率将下降到一个较低和稳定的水平上。随着时间的推移，故障率会再次升高，这是因为硬件将受到磨损等损害，达到一定程度后就应该报废。软件是逻辑的而不是物理的，虽然不会磨损和老化，但在使用过程中的维护却比硬件复杂得多。如果软件内部的逻辑关系比较复杂或规模比较大，在维护过程中很可能产生新的错误。

### 1.1.3 软件危机

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 1 000 万代码行，航天飞机软件长达 4 000 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，那么开发一个数万以至数百万行的软件，其复杂度将大大上升，即使是富有经验的开发人员，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期，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 1.1）。即在每 100 元用于计算机投资总额中，软件将花费 90 元。这一预期早已为实践所证实。

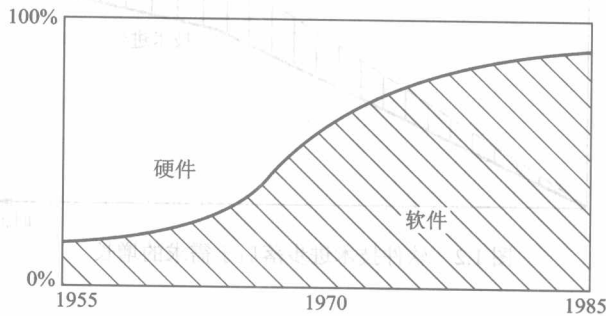


图 1.1 硬件/软件成本变化趋势

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了巨大的威胁。面对这种严峻的形势，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。

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进行修改，即所谓“完善性维护”和“适应性维护”（详见第 9 章）。不言而喻，软件的规模愈大，以上各种维护的成本必然愈高。

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许多软件人员，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），将其视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，是加剧软件危机的重要原因。

有人统计，硬件的性能价格比在过去 40 余年中增长了  $10^6$ 。一种新器件的出现，其性能较旧器件提高，价格反而有所下降，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（参阅图 1.2）；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，从而促使了软件工程的诞生。

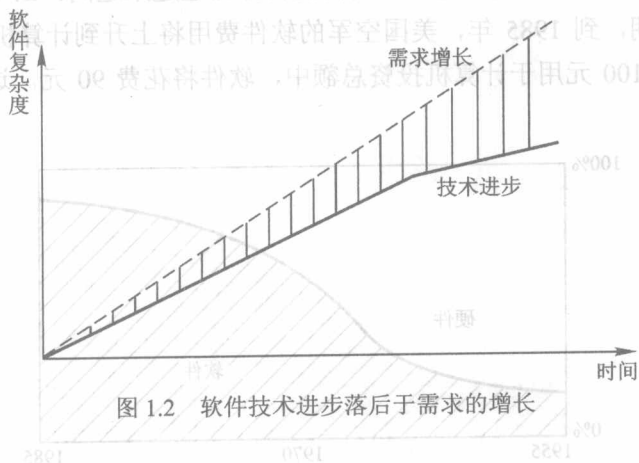


图 1.2 软件技术进步落后于需求的增长

## 1.2 软件工程的范畴

“软件工程”一词，首先是 1968 年北大西洋公约组织（NATO）在联邦德国召开的一次会议上提出的。它反映了软件人员认识到软件危机的出现，以及为谋求解决这一危机而做的一种努力。

人们曾从不同的角度，给软件工程下过各种定义。但是不论有多少种说法，它的中心思想都是把软件当作一种工业产品，要求“采用工程化的原理与方法对软件进行计划、开发和维护”。这样做的目的，不仅是为了实现按预期的进度和经费完成软件生产计划，也是为了提高软件的生产率与可靠性。

40 年来，人们围绕着实现软件优质高产这个目标，从技术到管理做了大量的努力，形成了“软件工程”这一计算机新学科。图 1.3 列举了它所包含的主要内容。

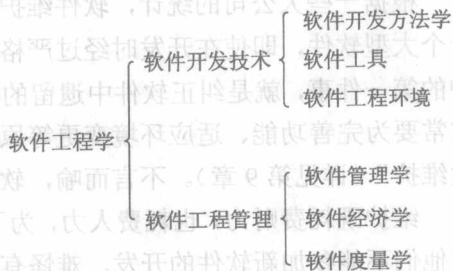


图 1.3 软件工程的范畴