



普通高等教育规划教材

实用软件工程

沈 洁 主编

 机械工业出版社
CHINA MACHINE PRESS



普通高等教育规划教材

实用软件工程

主编 沈 洁
主审 刘同明



机械工业出版社

本书侧重于软件工程的“工程”概念、理论和方法，注重介绍软件工程的计划、进度、风险、成本、评估和维护，整体上分为四个部分：第一部分概括地介绍了软件工程的概论，包括：软件工程的基本概念、方法与理论，软件工程的研究内容，软件工程方法的模式，软件工程的工具和环境等；第二部分，主要介绍了结构化软件工程的方法；第三部分，主要介绍了面向对象的软件工程方法；第四部分，介绍了软件工程管理、软件工程经济学、软件的维护以及形式化软件工程和 UML——统一的建模语言等高级软件工程方法。本书的特点是“实用”及充分体现工程特点，介绍了软件工程的最新理念，注重理论结合实际，结合实例介绍方法，内容紧凑，深入浅出。

本书可作为计算机科学与技术专业及其他相关本科专业的“软件工程”课程教材，也可作为从事计算机软件开发人员的参考书或使用手册。

图书在版编目(CIP)数据

实用软件工程/沈洁主编. —北京:机械工业出版社,
2004.8

普通高等教育规划教材

ISBN 7-111-14636-0

I. 实... II. 沈... 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 054243 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:王小东 责任编辑:刘丽敏 版式设计:冉晓华

责任校对:唐海燕 封面设计:饶薇 责任印制:施红

北京铭成印刷有限公司印刷·新华书店北京发行所发行

2004 年 8 月第 1 版第 1 次印刷

787mm × 1092mm $1/_{16}$ · 13 印张 · 320 千字

定价:18.50 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010)68993821、88379646

封面无防伪标均为盗版

普通高等教育应用型人才培养规划教材 编审委员会委员名单

主任：刘国荣 湖南工程学院
副主任：左健民 南京工程学院
陈力华 上海工程技术大学
鲍 泓 北京联合大学
王文斌 机械工业出版社

委员：(按姓氏笔画排序)

刘向东 华北航天工业学院
任淑淳 上海应用技术学院
何一鸣 常州工学院
陈文哲 福建工程学院
陈 峻 扬州大学
苏 群 黑龙江工程学院
娄炳林 湖南工程学院
梁景凯 哈尔滨工业大学(威海)
童幸生 江汉大学

计算机科学与技术专业分委员会委员名单

主任：黄陈蓉 南京工程学院
副主任：吴伟昶 上海应用技术学院
委员：（按姓氏笔画排序）
 汤 惟 江汉大学
 沈 洁 扬州大学
 陈文强 福建工程学院
 肖建华 湖南工程学院
 邵祖华 浙江科技学院
 靳 敏 黑龙江工程学院

序

工程科学技术在推动人类文明的进步中一直起着发动机的作用。随着知识经济时代的到来,科学技术突飞猛进,国际竞争日趋激烈。特别是随着经济全球化发展和我国加入 WTO,世界制造业将逐步向我国转移。有人认为,我国将成为世界的“制造中心”。有鉴于此,工程教育的发展也因此面临着新的机遇和挑战。

迄今为止,我国高等工程教育已为经济战线培养了数百万专门人才,为经济的发展作出了巨大的贡献。但据 IMD1998 年的调查,我国“人才市场上是否有充足的合格工程师”指标排名世界第 36 位,与我国科技人员总数排名世界第一形成很大的反差。这说明符合企业需要的工程技术人员特别是工程应用型技术人才市场供给不足。在此形势下,国家教育部近年来批准组建了一批以培养工程应用型本科人才为主的高等院校,并于 2001、2002 年两次举办了“应用型本科人才培养模式研讨会”,对工程应用型本科教育的办学思想和发展定位作了初步探讨。本系列教材就是在这种形势下组织编写的,以适应经济、社会发展对工程教育的新要求,满足高素质、强能力的工程应用型本科人才培养的需要。

航天工程的先驱、美国加州理工学院的冯·卡门教授有句名言:“科学家研究已有的世界,工程师创造未有的世界。”科学在于探索客观世界中存在的客观规律,所以科学强调分析,强调结论的惟一性。工程是人们综合应用科学(包括自然科学、技术科学和社会科学)理论和技术手段去改造客观世界的实践活动,所以它强调综合,强调方案优缺点的比较并做出论证和判断。这就是科学与工程的主要不同之处。这也就要求我们对工程应用型人才的培养和对科学研究型人才的培养应实施不同的培养方案,采用不同的培养模式,采用具有不同特点的教材。然而,我国目前的工程教育没有注意到这一点,而是:①过分侧重工程科学(分析)方面,轻视了工程实际训练方面,重理论,轻实践,没有足够的工程实践训练,工程教育的“学术化”倾向形成了“课题训练”的偏软现象,导致学生动手能力差。②人才培养模式、规格比较单一,课程结构不合理,知识面过窄,导致知识结构单一,所学知识中有一些内容已陈旧,交叉学科、信息学科的内容知之甚少,人文社会科学知识薄弱,学生创新能力不强。③教材单一,注重工程的科学分析,轻视工程实践能力的培养;注重理论知识的传授,轻视学生个性特别是创新精神的培养;注重教材的系统性和完整性,造成课程方面的相互重复、脱节等现象;缺乏工程应用背景,存在内容陈旧的现象。④老师缺乏工程实践经验,自身缺乏“工程训练”。⑤工程教育在实践中与经济、产业的联系不密切。要使我国工程教育适应经济、社会的发展,培养更多优秀的工程技术人员,我们必须努力改革。

组织编写本套系列教材,目的在于改革传统的高等工程教育教材,建设一套富有特色、有利于应用型人才培养的本科教材,满足工程应用型人才培养的要求。

本套系列教材的建设原则是:

1. 保证基础,确保后劲

科技的发展,要求工程技术人员必须具备终生学习的能力。为此,从内容安排上,保证学生有较厚实的基础,满足本科教学的基本要求,使学生日后具有较强的发展后劲。

2. 突出特色,强化应用

围绕培养目标,以工程应用为背景,通过理论与工程实际相结合,构建工程应用型本科教育系列教材特色。本套系列教材的内容、结构遵循如下9字方针:知识新、结构新、重应用。教材内容的要求概括为:“精”、“新”、“广”、“用”。“精”指在融会贯通教学内容的基础上,挑选出最基本的内容、方法及典型应用;“新”指将本学科前沿的新进展和有关的技术进步新成果、新应用等纳入教学内容,以适应科学技术发展的需要。妥善处理好传统内容的继承与现代内容的引进。用现代的思想、观点和方法重新认识基础内容和引入现代科技的新内容,并将这些按新的教学系统重新组织;“广”指在保持本学科基本体系下,处理好与相邻以及交叉学科的关系;“用”指注重理论与实际融会贯通,特别是注入工程意识,包括经济、质量、环境等诸多因素对工程的影响。

3. 抓住重点,合理配套

工程应用型本科教育系列教材的重点是专业课(专业基础课、专业课)教材的建设,并做好与理论课教材建设同步的实践教材的建设,力争做好与之配套的电子教材的建设。

4. 精选编者,确保质量

遴选一批既具有丰富的工程实践经验,又具有丰富的教学实践经验的教师担任编写任务,以确保教材质量。

我们相信,本套系列教材的出版,对我国工程应用型人才培养质量的提高,必将产生积极作用,会为我国经济建设和社会发展作出一定的贡献。

机械工业出版社颇具魄力和眼光,高瞻远瞩,及时提出并组织编写这套系列教材,他们为编好这套系列教材做了认真细致的工作,并为该套系列教材的出版提供了许多有利的条件,在此深表衷心感谢!

编委会主任 刘国荣教授
湖南工程学院院长

前 言

随着社会的进步、科学发展、计算机技术的应用和普及日新月异，今后，软件将在其中扮演一个十分重要的角色。“实用软件工程”作为计算机专业的一门工程性课程，是计算机专业的后继课程，它整合了计算机学科的各科知识，是通向工程的桥梁，是计算机专业的一门重要课程。

本书从实用和工程角度出发，对软件工程的基本概念、理论以及方法结合实例，进行了深入浅出的介绍。本书注重介绍软件工程的新理念和发展。该书主要作为本科阶段的教材使用，同时也适合于不同层次的计算机专业人员学习或作为软件工程使用手册。

全书共分 12 章，四个部分。第一部分，主要介绍了软件工程的基础知识，让读者了解软件工程的概貌，从整体上把握软件工程；第二部分，主要介绍了软件工程的结构化分析、结构化设计和结构化实施等方法；第三部分，主要介绍了软件工程的面向对象分析、面向对象设计及面向对象实施等方法；第四部分，主要介绍了软件工程管理、软件维护和 UML——统一的建模语言等内容。

本书第一、二章由沈洁编写，第三~五章由冯剑编写，第六~九章由林芳编写，第十~十二章由朱俊武编写。沈洁同志对全书进行了统稿。

在本书的编写过程中，得到了罗建利、许有志和杭月芹同志的帮助，华东船舶工业学院刘同明教授审阅了全部书稿，提出了许多宝贵意见，借此一并表示感谢。

编 者

2004 年 3 月

目 录

| | | | |
|------------------------------|-----|--|--|
| 序 | | | |
| 前言 | | | |
| 第一章 软件与软件危机 | 1 | | |
| 第一节 软件的发展历程 | 1 | | |
| 第二节 软件的概念和特点 | 2 | | |
| 第三节 软件危机 | 4 | | |
| 习题 | 6 | | |
| 第二章 软件工程的概 念 | 7 | | |
| 第一节 软件工程的定义 | 7 | | |
| 第二节 软件工程方法 | 10 | | |
| 第三节 常见的几种软件开发模型 | 12 | | |
| 习题 | 21 | | |
| 第三章 结构化分析 | 22 | | |
| 第一节 结构化方法 | 22 | | |
| 第二节 需求分析 | 23 | | |
| 第三节 结构化分析方法 | 29 | | |
| 第四节 其他具有结构化思想的需求分析方法 | 37 | | |
| 习题 | 39 | | |
| 第四章 结构化设计 | 41 | | |
| 第一节 软件设计过程 | 41 | | |
| 第二节 软件设计的概念和原则 | 42 | | |
| 第三节 结构化设计方法 | 49 | | |
| 第四节 人机界面设计 | 61 | | |
| 第五节 详细设计 | 63 | | |
| 第六节 面向数据结构的设计方法 | 68 | | |
| 习题 | 72 | | |
| 第五章 结构化实现 | 74 | | |
| 第一节 编码 | 74 | | |
| 第二节 软件测试 | 78 | | |
| 第三节 调试 | 99 | | |
| 第四节 软件可靠性 | 101 | | |
| 习题 | 104 | | |
| 第六章 面向对象方法学概述 | 106 | | |
| 第一节 面向对象方法基本概念 | 106 | | |
| 第二节 面向对象的软件工程方法 | 110 | | |
| 习题 | 112 | | |
| 第七章 面向对象分析 | 113 | | |
| 第一节 对象模型的结构 | 113 | | |
| 第二节 对象模型的建立 | 115 | | |
| 第三节 动态模型的建立 | 119 | | |
| 第四节 功能模型的建立 | 123 | | |
| 第五节 定义服务 | 123 | | |
| 习题 | 125 | | |
| 第八章 面向对象设计 | 126 | | |
| 第一节 面向对象设计概述 | 126 | | |
| 第二节 问题域子系统 (PDC) 的设计 | 127 | | |
| 第三节 人一机交互子系统 (HIC) 的设计 | 129 | | |
| 第四节 任务管理子系统 (TMC) 的设计 | 130 | | |
| 第五节 数据管理子系统 (DMC) 的设计 | 131 | | |
| 习题 | 132 | | |
| 第九章 面向对象实现 | 133 | | |
| 第一节 面向对象编程 | 133 | | |
| 第二节 面向对象测试 | 138 | | |
| 习题 | 140 | | |
| 第十章 软件工程管理 | 141 | | |
| 第一节 软件项目管理 | 141 | | |

| | | | |
|------------------------|-----|--------------------------------|-----|
| 第二节 软件计划 | 145 | 习题 | 167 |
| 第三节 软件质量管理 | 147 | | |
| 第四节 软件经济学 | 150 | 第十二章 UML——统一的建模语言 | 168 |
| 习题 | 152 | 第一节 UML 概述 | 168 |
| 第十一章 软件维护 | 153 | 第二节 UML 静态建模机制 | 173 |
| 第一节 软件维护的基本概念 | 153 | 第三节 动态建模机制 | 179 |
| 第二节 软件的可维护性 | 157 | 习题 | 185 |
| 第三节 软件维护的过程分析 | 160 | 附录 软件开发文档编写参考 | 186 |
| 第四节 基于构件复用的软件再工程 | 164 | 参考文献 | 198 |

第一章 软件与软件危机

1946年，人类第一台电子计算机的问世，标志着人类开始从工业社会向信息社会迈进，因此，人类认识自身生存空间的能力逐步提高，同时计算机系统的发展日新月异，其中软件产业的发展尤为引人注目。但是，什么是软件？怎么开发和维护软件？什么是软件危机？怎么解决软件危机等等一系列问题，从20世纪60年代开始一直在困扰着人们，软件工作者们也一直在努力寻求更好的答案、更好的方法。

人们提出了软件工程是解决这些问题的最好方法，那么什么是软件工程？软件工程的理想是什么？软件工程的理念是什么？目的又是什么？本书从软件工程的工程角度，介绍实用的软件工程方法，围绕软件工程的“计划”、“进度”、“成本”、“质量”等工程特点，分别从结构化与面向对象两个方面介绍软件工程。

第一节 软件的发展历程

“软件”是20世纪60年代出现的，尽管自从第一台计算机问世以来就开始了软件的生产，但“软件”的发展经历了“程序设计的原始时代”、“基本软件时代”、“程序设计方法时代”和“软件工程时代”四个时代。

(1) **程序设计的原始时代(20世纪50年代)** 这是软件发展的早期时代，这个时代的软件是为某个具体应用而专门编写的，使用的工具是机器语言和汇编语言，它们是面向机器的；开发方法追求编程技巧、追求程序运行效率，使得程序难读、难懂、难修改；生产方式是个体劳动式的，大多数人认为软件开发是无需计划的事情，编程者和使用者往往是同一个人(同一组人)，程序的规模小，程序编写较容易。对软件开发工作没有管理，这种个体化的软件开发环境，使得软件开发只是在某些人头脑中隐含进行的一个模糊过程，除了程序清单以外没有任何文档资料，是一个不可视的过程。软件特征是只有程序、程序设计概念，不重视程序设计方法。

(2) **基本软件时代(20世纪50年代末~60年代末)** 在这个期间，有了操作系统和编译技术，不再直接用机器码编程了，而使用各种符号语言来编程，如高级语言(BASIC、FORTRAN、ALGOL、COBOL)，高级语言程序设计比较接近人的习惯，人们不必考虑计算机内部的构造和不同机器的特点，只要考虑解题步骤，写出的程序经过编译，翻译成机器能执行的机器指令，计算机就能执行。这个阶段，计算机应用范围大大扩大，计算机大规模进入工商业、银行业等领域。这时编程的目的不是关心计算机硬件的动作，而是要确定程序员定义的动作序列，程序员控制的动作力度可大大增加。它们由高级语言的语句或宏指令、宏语句等表示，面向处理过程，因此解决问题的规模与复杂性大大增加。这个时代的开发方法可以用“功能性程序设计”来表达其开发技术的特点。针对特定的问题，根据所需的功能，制定特

定的方法，甚至要考虑是否需要特定的机器去解决该问题。其编程毫无章法，类似于智力游戏，主要依赖于编程人员的才智与技巧，缺乏软件开发技术与理论。因此，编程的随意性很大，一个人写的程序另一个人很难看懂或理解。这种完全靠个人想象力构造的系统，其中所产生的问题就很难解决，在这个时代产生了“软件危机”。

(3) **程序设计方法时代**(20世纪60年代末~70年代中) 为了解决“软件危机”，不少软件工作者要求提高程序的可读性、可理解性，而不再强调编程技巧。荷兰科学家E.W. Dijkstra于1965年提出：不该简单地只考虑编写程序，就期望产生一个正确的结果，而应考虑如何把软件进行划分及构造。他在编写操作系统时提出了分层次结构、取消GOTO语句以及结构化程序设计方法。1966年，Bohm和Jacopini证明了一个重要的“程序结构定理”，只用“顺序”、“选择”和“循环”三种基本的控制结构就能实现任何单入口单出口的程序。1972年，IBM公司的Mills进一步提出，程序应该只有一个入口和一个出口。结构化程序设计是一种设计程序的技术，它采用自顶向下逐步求精的设计和单入口单出口的控制结构。这就产生了结构化程序设计的方法，即只要用三种控制结构的高级程序设计语言以及只有一个入口和一个出口的程序设计原则的新的程序设计思想、方法和风格，使程序变得清晰、易读、易修改。“程序设计 = 数据结构 + 算法”总结了这个时代程序设计的特征与成就。在这个时代软件开发技术主要有两大发展：第一是从程序中分离出数据结构与算法；第二是把结构化程序设计方法发展成结构化开发方法(包括结构化分析与结构化设计方法)。在这个时代发展了各种程序设计语言，如Pascal(不用GOTO语句)、Smalltalk(面向对象语言)、Prolog(逻辑语言)，形成了系统软件与应用软件的区别，形成了完整的软件系统(不仅是可执行系统，还有独立的数据状态和程序系统的规格说明书)。

(4) **软件工程时代**(20世纪70年代中~今) 面对复杂的、大型的软件开发，不再仅仅是程序设计的事，出现了“软件工程”的思想。如B.W. Boehm在1976年对软件工程定义时所说“在软件(包括程序和文档)设计、实现、测试、运行、维护的各个过程中，建立在科学基础上的一套实用方法”。C.A. Hoare则认为软件工程是“一种导致电子数字计算机具有强大功能的艺术，这种功能表现为人可以方便地使用计算机”。这个时代软件的生产方式是工程化的生产，使用数据库、开发工具、开发环境、网络、分布式、面向对象技术来开发软件。其中最重要的理论成就是关系数据库的理论。在这个时代虽然软件开发技术有很大进步，但是未能获得突破性进展，软件价格不断上升，没有完全摆脱软件危机。

软件不是自然规律的产物，是人类思维的创造物，有待理论的创新和不断实践。

第二节 软件的概念和特点

一、软件定义

“软件”一词随着计算机技术的发展，在不同阶段有不同的认识。在计算机发展的初期，硬件的设计和生是主要问题，那时所谓的“软件”就是“程序”，甚至是“机器指令程序”，它们处于从属地位。后来，人们认识到在机器上增加软件的功能会使计算机系统的能大大提高，于是在研制计算机系统时既要考虑到硬件，又要考虑到软件，而且开始编制

一些大型程序，这时人们认为“软件”是“程序+说明书”。再后来，特别是在软件发展的第三阶段，人们认为“软件”就是“文档+程序”。到了软件工程时代，人们对软件的认识又有了新的进展。

计算机软件可定义为计算机程序、文档和数据的完整集合。在该定义中，“程序”是指当运行时能够提供所要求功能和性能的指令的集合；“文档”是描述有关计算机程序功能、设计、编制、使用的文字或图形资料。数据是指使程序能较好地处理信息的数据结构。

二、软件的特点

软件是一种特殊的产品，它与硬件和人一起构成完整的计算机系统，它们是相互依存的、缺一不可的。它具有下列一些特点：

1) 软件是一种逻辑产品，它与硬件有很大的区别。软件产品是看不见摸不着的，是无形的，是好是坏，直到运行时才知道，这给设计、生产、管理带来了许多困难。它是智力劳动的结晶，以程序和文档的形式出现，保存在计算机的存储设备上，通过计算机的运行才能体现它的功能和作用。

2) 软件生产方式不同于硬件，软件产品的生产主要是研制，软件开发是人的智力的发挥，不是传统意义上的硬件制造。其成本主要体现在软件的开发和研制上，软件开发研究完成后，通过复制就产生了大量软件产品。

3) 软件产品不会用坏，不存在磨损、消耗等问题。

4) 软件产品的生产主要是脑力劳动，还未完全摆脱手工开发方式，大部分产品是“定做”的。软件的研制工作需要投放大量的、复杂的、高强度的脑力劳动，它的成本非常高。

5) 要求不同于硬件，硬件允许误差，软件不允许，美国金星探测器水手1号导航程序的一个语句的语法正确，但语义错了，结果飞行偏离航线，导致试验失败。阿波罗宇宙飞船飞行控制软件，由于粗心把一个逗号写成句号，几乎造成悲剧性的后果。

6) 维护不同于硬件，在理论上，软件是不会用坏的，但实际上软件也会变坏变旧。因为在软件的整个生存期中，一直处于改变(维护)状态，而随着某些缺陷的维护，很可能引入另一些新的缺陷，因而使得软件的故障增加，品质变坏。硬件某一部分变坏，可以使用备件，而软件不行。所以软件的维护要比硬件复杂得多。

三、软件的分类

软件的分类方法很多，如按其功能进行划分，可分为如下几类：

(1) **系统软件** 即能与计算机硬件紧密配合，使计算机系统的各个部件和相关的软件协调、高效地工作的软件。如操作系统。

(2) **支撑软件** 即协助用户开发软件的工具性软件，包括帮助程序人员开发软件产品的工具和帮助管理人员控制开发进程的工具。

(3) **应用软件** 即在特定领域内开发，为特定目的服务的软件。

(4) **交叉软件** 指既具有支撑软件的特征，又具有应用软件的功能，而且随着软件的发展还在不断变化的一类软件。

按软件的规模进行划分，可分为微型、小型、中型、大型、超大型、极大型软件。

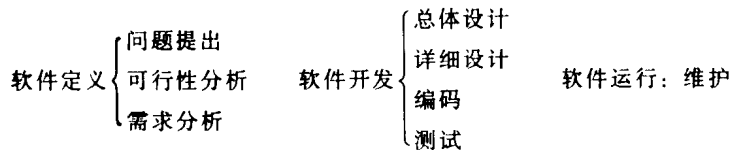
按软件的工作方式进行划分，可分为实时处理软件、分时处理软件、交互式软件、批处

理软件等。

按软件服务对象的范围进行划分,可分为项目软件、产品软件等。

四、软件的生命周期

软件的生产、使用和消亡过程称为软件的生命周期,即是从软件问题的提出,直到软件停止使用的整个过程。软件生命周期由软件定义、软件开发和软件运行三个阶段组成。把整个生存周期划分为若干阶段,使得每个阶段有明确的任务,这样规模大、结构复杂和管理复杂的软件开发变得容易控制和管理。每个阶段根据开发模式的不同又有其不同形式的内容。1988年发布的GB8566《计算机软件开发规范》中软件生存周期阶段的划分是基于“瀑布模型”给出的。它把软件生存期划分为三阶段8个步骤:



在这个过程中,维护是一个很长的阶段,要提高软件的使用价值,就得考虑如何延长软件的使用期。因此,软件的可维护性是提高软件价值的最重要指标。从软件的需求分析开始,就得研究如何提高软件的可维护性。

软件生存周期是借用工程中产品生存周期的概念而得来的。引入软件生存周期概念,对于软件生产的管理、进度控制有着非常重要的意义,可使软件生产有相应的模式、相应的流程、相应的工序和步骤。

第三节 软件危机

一、软件危机

由于计算机硬件技术的进步,计算机的运行速度、容量和可靠性有了显著的提高,生产成本有了显著下降,为计算机的广泛应用创造了条件,因而出现了一些复杂的、大型的软件开发项目,虽然软件开发技术一直在发展,但始终不能满足软件发展的要求。

软件危机是指在计算机软件的开发和维护过程中所出现的一系列严重的问题。这些问题因找不到解决的办法,使之长期积累,形成了尖锐的矛盾,产生了软件危机。

二、软件危机的表现

软件危机表现在很多方面,以下几个方面是比较主要的、明显的表现。

(1) **软件需求增长得不到满足** 由于软件的生产不同于硬件,没有高效率的生产工具,是一种逻辑产品,是人的智力活动的结果,随着计算机的普及应用,软件的社会需求量不断增加,其产品满足不了社会的需求。

(2) **软件生产成本低、价格昂贵** 在计算机系统发展的早期,软件成本在整个计算机系统中所占的比例约为10%~20%,随着软件产业的发展,软件成本日益增长,而计算机硬件的价格不断下降,到20世纪60年代中期,软件成本在计算机系统中所占的比例增长到50%左右。今天在计算机系统中软件成本所占比例已经大于85%了。在计算机中,硬件与

软件的成本比见图 1-1。

(3) **软件生产进度无法控制** 软件是一种逻辑的系统符号, 据估计, 软件设计与硬件设计相比, 使用的逻辑量要多于 10~100 倍。另外, 在软件开发过程中会遇到各种意想不到的情况和困难, 而这些并不是加大资金投入就能解决的。这些都给软件项目的计划和论证带来了极大的困难。

(4) **软件需求定义不准确** 这主要是指一些应用软件, 由于信息的不完全性, 使得对于某些应用软件的需求很难完整或准确地描述(估计)。

(5) **软件质量不易保证** 主要有两个方面的原因: 一方面, 目前提高软件质量的主要手段是测试, 而测试只能发现错误, 不能保证软件正确; 另一方面, 软件设计人员对计算机的了解和认识与用户的想法存在着很大的差距, 用户的需求又在不断地变化。这些都是造成软件质量不易保证的重要因素。

(6) **软件可维护性差** 主要原因是软件开发过程不规范, 软件开发人员按各自的风格工作, 各行其事; 同时开发过程无完整、规范的文档, 发现问题后进行杂乱无章的修改。这样就造成了错误难以改正, 新功能难以增加, “再用性”的软件不能实现等问题。

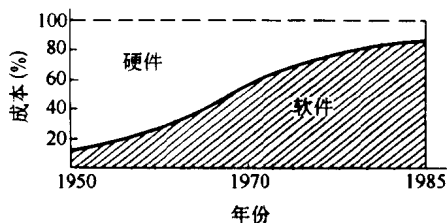


图 1-1 硬件与软件成本比

三、软件危机产生的原因

在软件开发和维护过程中出现的软件危机, 主要原因有两个方面: 一是与软件本身的特点有关, 二是与软件开发和维护的方法不正确有关。具体地有以下几点:

1) 软件是逻辑部件, 不是物理部件。它缺乏“预见性”, 只有在计算机上运行后才能发现是否有缺点。

2) 软件的规模越来越大, 结构越来越复杂。随着计算机应用的日益广泛, 需要开发的软件规模日益庞大, 软件结构也日益复杂。对于庞大、复杂的软件, 其调用关系、接口信息以及数据结构都变得日益复杂, 有时这种复杂程度甚至超过了人所能接受的程度。

3) 软件开发、维护管理困难。一方面, 软件生产是一种智力劳动, 软件的规模较大, 结构复杂, 且具有无形性; 另一方面, 软件开发人员只重视软件开发而轻视问题的定义、开发过程中文档的建设, 造成在软件的开发过程中以及在使用后的维护中, 管理困难, 以至于进度控制困难、质量控制困难、可靠性无法保证。

4) 软件开发技术落后。在计算机系统发展的早期, 人们不注重软件开发技术的研究, 这就造成了在计算机系统的发展中, 软件开发技术不能与时俱进。在实际开发过程中, 用户要求的软件复杂性与软件技术解决复杂性的能力不相适应。

5) 生产方式落后。软件仍然采用个体手工方式开发, 根据个人习惯、爱好工作, 无章可循, 无规范可依据。凭经验不讲优化, 与硬件相比, 很难实现工业化生产。

6) 开发工具落后, 生产率提高缓慢。软件开发工具过于原始, 没有出现高效率的开发工具, 因而软件生产率低下。

四、缓解软件危机的途径

软件危机只有缓解, 不可能解决, 特别是在目前这种硬件结构体系下开发的软件, 其危

机永远存在，即使未来的新一代计算机系统问世后，也会产生新形式的软件危机。辩证地看，正是由于软件危机的存在，才使得软件技术日益发展。所以人类不可能彻底解决软件危机，只有想办法去缓解软件危机，通过缓解软件危机的过程来推动软件的发展。

怎么来缓解软件危机？只有通过从事软件研究的科技人员，不断的勤奋探索，推动软件的发展，改进软件的开发方法、完善软件的开发工具，提高软件的生产率。

针对软件的现状，采取什么措施来缓解软件危机呢？从下列几方面来考虑缓解的途径：

1) 更新观念。软件开发不应当看作某种个体劳动的神秘技巧的结合，而是一种组织良好、管理严密、各类人员协同配合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，把这些融合到软件项目的开发过程中来，或用这种思想来指导、管理软件项目的开发与研制，使软件生产工程化。

2) 吸取系统的理念，结合组织管理与技术管理两面，从软件哲学深层次去考虑软件文化、软件开发的心理学，研究软件开发的模式。要用超越工业化的思想指导软件的开发。

3) 应该不断地去发现、完善、推广、使用在实践中总结出来的开发软件的成功的技术和方法。不断地去研究探索更好、更有效的技术和方法，极力地批评目前正在使用的一些开发计算机软件的错误概念和方法。

4) 要开发更好的软件开发工具，提高软件的生产率。工具的使用可以解放人类的“体力”劳动和“智力”劳动。使人类能摆脱那些繁琐的重复工作，使软件开发人员在软件开发工具的辅助下，把握整个软件开发的过程，控制和管理好整个软件的开发和维护。

综上所述，要缓解软件危机，既要有先进的技术和方法，又需要高水平的组织管理措施，更重要的是要有信息化的管理模式。而软件工程正是综合了管理和技术两方面，研究如何更好地开发软件的一门新兴学科。所以，就目前而言，软件工程是缓解软件危机的最好途径。

习 题

1. 计算机软件的特征是什么？与硬件相比有什么区别？
2. 软件危机产生的背景是什么？为什么会产生软件危机？为什么软件危机只能缓解而不能消除？
3. 什么是软件的生命周期？生命周期是如何划分的？

第二章 软件工程的观念

传统的小作坊式的软件开发方式，显然阻碍了软件的发展，同样工业化的思想和管理模式同样影响了软件的发展。人们试图寻找一种方法使得软件生产超越工业化，用信息化的思想指导制定软件开发和维护的规范，扩大生产的规模，提高软件生产率，对软件的质量和生进度进行控制，建立一种软件开发文化。软件工程是基于项目和产品，追求超越项目、产品、开发，其科学特征是：可观察、公开、可重复。软件工程的基础是客观性，代表软件开发的共性。软件工程的客观性反映在软件工程的一切活动都环绕着事实进行。软件工程重视文档，文档记录了形成开发过程中的事实历史，反映人对项目的理解和项目进展的客观表示，是分析和改进的客观基础。

第一节 软件工程的定义

一、软件工程的定义

软件工程思想是 20 世纪 60 年代末提出的，20 世纪 70 年代以后逐步发展起来的一门指导计算机软件开发和维护的工程科学。采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，这就是软件工程。这门学科的目的是研究如何从管理和技术两方面更好地开发和维护计算机软件。

1968 年在联邦德国召开的国际会议上正式提出并使用了“软件工程”这个术语，运用工程学的基本原理和方法来组织和管理软件生产。后来还发展了与软件有关的心理学、生理学 and 经济学等方面的学科。在这个期间，研究软件工程的专家学者们陆续提出了 100 多条关于软件工程的准则。这 100 多条准则可以概括为下述 6 条基本原则：

(1) **用分阶段的生存周期计划严格管理** 把软件生存期分为若干个阶段，制定出各个阶段的可行计划，然后按这些计划对软件的开发与维护进行管理。

(2) **坚持阶段评审** 软件的质量保证工作不能等到运行时再进行，应该在每个阶段都要进行严格的评审，以便尽早发现在软件开发过程中所犯的错误。

(3) **实行严格的产品控制** 要求软件开发过程中不应当随意改变需求，因为改变一项需求要付出较大的代价。但是，在软件开发过程中改变需求又是不可避免的。由于外部环境的变化，相应地改变用户需求是客观需要的。为保持软件各个配置成份的一致性，必须要实行严格的产品控制。对于一切有关修改软件的建议，都必须按照严格的规程进行评审，获得批准后才能实施修改，绝对不能随意修改。

(4) **采用现代程序设计技术** 实践已经表明，采用先进的技术既可以提高软件开发的效率，又可以提高软件维护的效率。

(5) **能非常清楚地审查结果** 软件产品是一种逻辑产品。软件开发人员的工作进展情况