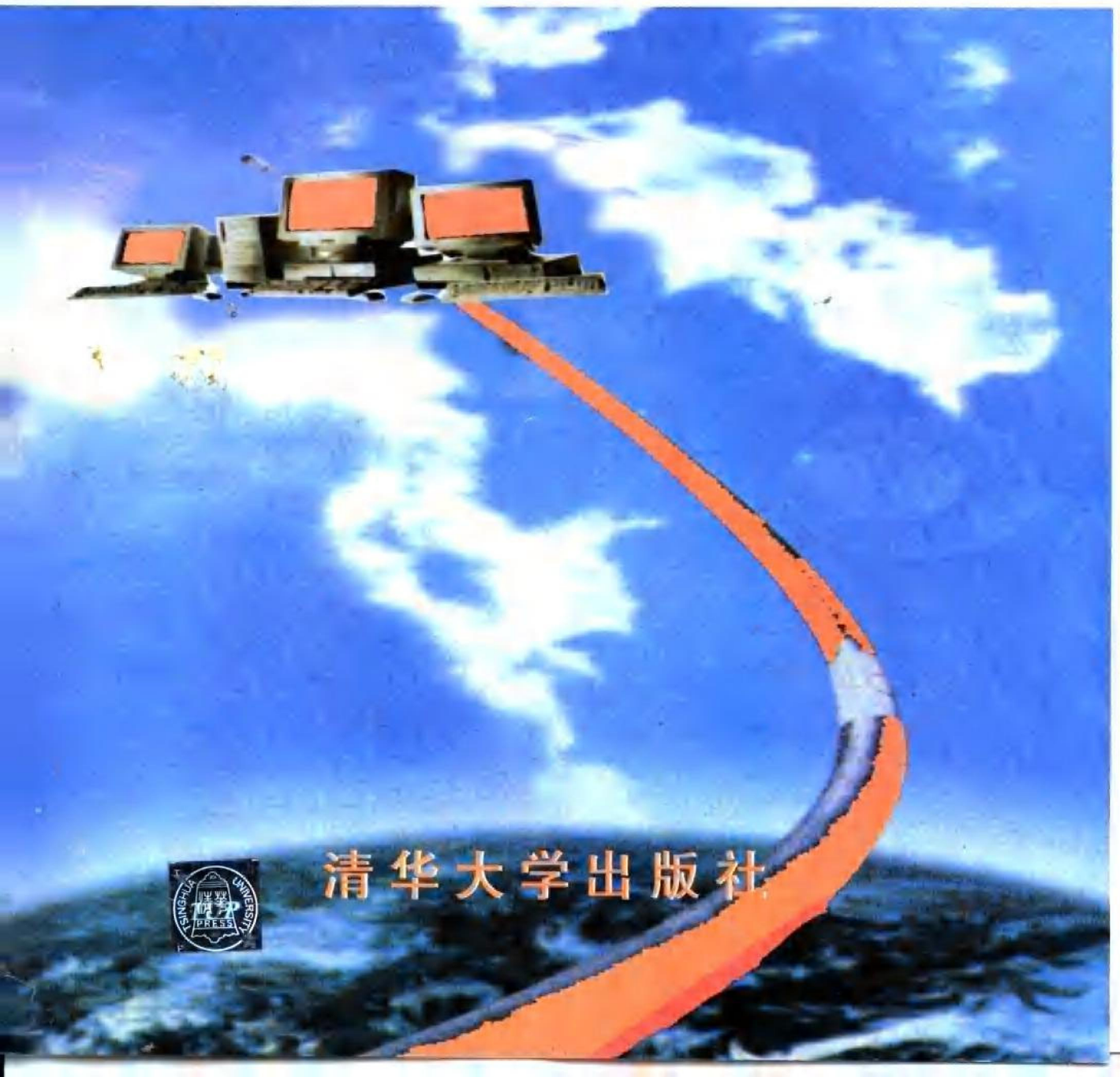
计算机专业大专系列教材

软件工程概论

郑人杰　殷人昆　编著



清华大学出版社

计算机专业大专系列教材

软件工程概论

郑人杰　殷人昆　编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

软件工程是 20 世纪 60 年代开始发展起来的新学科。随着计算机的普及作为其核心部分的软件已深入到社会生产活动和生活的各个领域。软件的开发和维护都需要软件工程知识,因此有人称它为软件产业的支柱。本书是作者根据在清华大学多年教学的讲义改编的。内容包括:软件工程概述;软件需求分析;软件设计;详细设计描述的工具;程序编码;面向对象技术;软件测试;软件维护;软件工程标准化与软件文档。书中适当介绍了软件管理和软件工程标准化问题。掌握这些知识将有助于读者在软件工程项目中体现工程化和标准化。内容通俗易懂,图文并茂,原理、方法与实例结合。

本书适于作大专院校中计算机或软件专业的教材,也可供计算机软件人员和计算机用户阅读。

图书在版编目(CIP)数据

软件工程概论/郑人杰,殷人昆编著.—北京:清华大学出版社,1998

计算机专业大专系列教材

ISBN 7-302-02909-1

Ⅰ.软… Ⅱ.①郑…②殷… Ⅲ.软件工程 Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字(98)第 07165 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

http://www.tup.tsinghua.edu.cn

印刷者:北京昌平环球印刷厂

发行者:新华书店总店北京发行所

开 本:787×1092 1/16 **印张**:18.75 **字数**:441 千字

版 次:1998 年 4 月第 1 版 1999 年 8 月第 3 次印刷

书 号:ISBN 7-302-02909-1/TP·1539

印 数:16001～24000

定 价:19.80 元

序　言

为什么要组织编写这套计算机专业大专使用的教材？根据什么来组织这套教材？这套教材的特点是什么？它能起到哪些作用？这就是我们在这篇序言中，要回答，也必须回答的问题。

计算机专业大专教育发展非常迅速，它满足了社会对这个层次人才的需求。在数量上已经超过了对本科人才的需求。大专这个层次有自己的特殊性，时间只有三年，要学习的内容很多，怎样精选教学内容，就成为十分重要的问题；他们又不同于中专层次，要求既有相当坚实的理论基础，又要运用理论解决实际问题，因此，如何处理好理论和实践的关系就十分重要。

大专这个层次人才的重要性是不言而喻的。但是，在培养大专这个层次人才的过程中，突出的矛盾之一，就是缺乏合适的大专教材。目前，不是用本科教材代用，就是很难及时获得所需教材。这就是组织这套专为大专使用的教材的起因。

那么，我们组织编写这套教材以什么为依据呢？中国计算机学会教育委员会与全国高等学校计算机研究会联合推荐的《计算机学科教学计划 1993》(以下简称"93 计划")是我们这次组织编写这套教材的主要依据。

"93 计划"所提供的指导思想和学科内容不仅适合大学本科，也适合于大专的需要。

"93 计划"明确规定了计划实施的目标：1. 要为"计算机学科"的毕业生提供一个广泛坚实的基础；2. 在培养人才的过程中，必须反映培养目标的差异；3. 要为学生毕业后，进一步学习新的知识和迎接新的工作挑战，做好理论和实践上的准备；4. 要学生能够把在校学到的知识，用到解决实际问题的过程中去。

在学科内容方面"93 计划"概括了九个科目领域。九个科目领域组成《计算机学科》的主科目。每个科目领域都有重要的理论基础、重要的抽象(实验科学)、重要的设计和实现的成就。

这九个科目领域作为教学计划的公共要求，它们是：算法与数据结构、计算机体系结构、人工智能与机器人学、数据库与信息检索、人-机通信、数值与符号计算、操作系统、程序设计语言、软件方法学和工程。

我们根据上述指导思想和学科要求精选了十三门课，作为大专用的主干教材。它们是：《数据结构》、《数字电路逻辑设计》、《计算机组成原理》、《微机原理与应用》、《微机接口技术》、《计算机网络》、《数据库原理及应用》、《操作系统基础》、《汇编语言程序设计》、《C 程序设计》、《软件工程概论》、《微机系统应用基础》和《离散数学》。

这十三种教材大体上反映了除人工智能与机器人学和数值与符号计算之外的全部要求，足以满足大专主干课程教学的需求。

这套教材我们都是聘请大专院校有丰富教学实践经验的、工作在第一线的专家、教授编写。在编写过程中，充分考虑了大专的特点，在选材上贯彻少而精的原则，在处理上贯彻理论密切联系实际的原则，力求深入浅出，便于教学。并且在主要章节后均附有适量的习题。

这套教材适合于计算机专业大专生使用，也可作为非计算机专业的本科生使用。

主编　李大友

1996.6

前　言

软件工程是一门实用性很强的年青学科。尽管其中也包含了某些理论的内容，但它具有一个显著特点是实践性。软件工程学科的实践性不仅体现在，它的形成和发展得益于软件工程项目的推动，或者说，是人们在软件开发的实践中碰壁之后为寻求“软件危机”的出路而总结出的原则和方法；而且它的实践性还体现在对于软件开发项目的实际指导作用。许多人感到，理解和掌握这一学科的知识并不难，然而，常常发生的问题是不能坚持按照它所提供的原则和方法去做。例如，有些规模不小的软件项目因为一开始就忽视了按软件工程的要求开发，致使开发后期，或是在维护阶段步入了十分被动的境地。由于结构性、清晰性和可扩充性差，导致整体的可维护性差，运行中发现了问题，多次修改形成“补丁上加补丁”之后，难于再行修补。加上文档编制得不够理想，以致没有人愿意承担维护工作。但若将其放弃重新开发，从时间、资源多方面考虑，又是不可能的。这类情况一再发生表明，其教训并没有被人们认真吸取。一个中型以上的软件开发项目成功与否很大程度上取决于管理工作，这一点已逐渐成为人们的共识。为防止类似事件的重演，建议从两个方面着手，即：

1. 加强软件过程的管理，不断改进已为人们习惯了的传统开发过程。这就要克服轻视项目管理和文档工作、程序编写任意性等传统习惯。这一点正是本书编入最后两章内容的初衷。

2. 初学者在一接触本学科时，即强调应用，强调实践。希望初学者在一开始就打下良好的软件工程观念的基础，并在今后的软件开发工作中得到贯彻。

本书主要针对大专以上的读者，如感到其中材料不足，可参阅作者的另一本教材《实用软件工程》(第二版)，清华大学出版社出版。

特别感谢殷人昆副教授，他在繁忙的教学科研工作中抽空帮我最后脱稿付梓，才使本书与读者见面。

郑人杰

1997 年 9 月 22 日于清华园

目　录

第1章　软件工程概述

在近代技术发展的历史上，工程学科的进步一直是产业发展的巨大动力。传统的工程学科走过的道路已为人们所熟知。水利工程、建筑工程、机械工程、电力工程等对工农业、商业、交通业的影响是极为明显的。近年来人们开始对气象工程、生物工程、计算机工程等有了新的认识。然而对工程学科家族的另一新成员——软件工程却不很熟悉。事实上，软件工程的地位非常重要，它对软件产业的形成和发展起着决定性的推动作用。在计算机的发展和应用中至关重要，在人类进入信息化社会时成为新兴信息产业的支柱。

本章将对软件的地位和作用、软件的特点、软件的发展和软件危机、软件工程学科的形成、软件生存期及软件工程过程等方面的问题和基本概念给出简要的介绍，以便使读者在进入软件工程专题以前，对总体框架获得一般性的理解。

1.1　软件的概念、特点和分类

1.1.1　软件的概念与特点

“软件”这一名词20世纪60年代初从国外传来。英文software一词确是soft和ware两字组合而成。有人译为“软制品”，也有人译为“软体”。现在人们统称它为软件。对于它的一种公认的解释为，软件是计算机系统中与硬件相互依存的另一部分，它是包括程序、数据及其相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料。

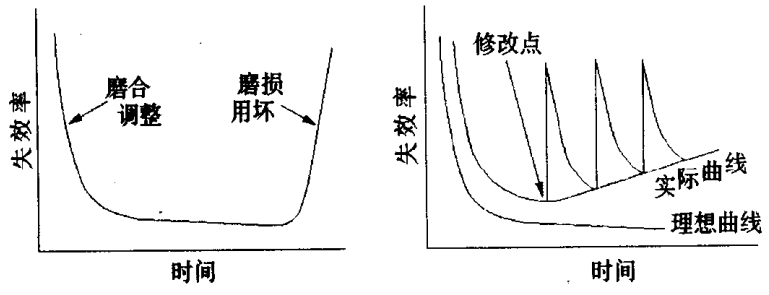
为了能全面、正确地理解计算机和软件，必须了解软件的特点。

1）软件是一种逻辑实体，不是具体的物理实体。它具有抽象性。这个特点使它和计算机硬件，或是其它工程对象有着明显的差别。人们可以把它记录在纸面上，保存在计算机的存储器内部，也可以保留在磁盘、磁带和光盘上，但却无法看到软件本身的形态，必须通过观察、分析、思考、判断，去了解它的功能、性能及其它特性。

2）软件的生产与硬件不同，在它的开发过程中没有明显的制造过程。也不像硬件那样，一旦研制成功，可以重复制造，在制造过程中进行质量控制。软件是通过人们的智力活动，把知识与技术转化成信息的一种产品。一旦某一软件项目研制成功，以后就可以大量地复制同一内容的副本。所以对软件的质量控制，必须着重在软件开发方面下功夫。由于软件的复制非常容易，因此出现了软件产品的保护问题。

3）在软件的运行和使用期间，没有硬件那样的机械磨损、老化问题。任何机械、电子设备在使用过程中，其失效率大都遵循如图1.1(a)所示的U型曲线（即浴盆曲线）。因为在刚投入使用时，各部件尚未做到配合良好、运转灵活，容易出现问题。经过一段时间运行，就可以稳定下来。而当设备经历了相当长时间的运转，就会出现磨损、老化，使失效率

越来越大。当失效率达到一定程度,就到达了寿命的终点。而软件的情况与此不同,它没有U型曲线的右半翼,因为它不存在磨损和老化问题。然而它存在退化问题。在软件的生存期中,为了使它能够克服以前没有发现的故障、使它能够适应硬件、软件环境的变化以及用户新的要求,必须要多次修改(维护)软件,而每次修改不可避免地引入新的错误,导致软件失效率升高,如图1.1(b)所示,从而使得软件退化。



(a) 硬件失效率曲线　　(b) 软件失效率曲线

图1.1 失效率曲线

4) 软件的开发和运行常常受到计算机系统的限制,对计算机系统有着不同程度的依赖性。软件不能完全摆脱硬件单独活动。有的软件这种依赖性大些,常常为某个型号的计算机所专用。有的软件依赖于某个操作系统。为了解除这种依赖性,在软件开发中提出了软件移植的问题。

5) 软件的开发至今尚未完全摆脱手工艺的开发方式。软件产品大多是"定做"的,很少能做到利用现成的部件组装成所需的软件。近年来软件技术虽然取得了不少进展,提出了许多新的开发方法,例如利用现成软件的复用技术、自动生成技术,也研制了一些有效的软件开发工具或软件开发环境。但在软件项目中采用的比率仍然很低。由于传统的手工艺开发方式仍然占据统治地位,开发的效率自然受到很大限制。

6) 软件本身是复杂的。软件的复杂性可能来自它所反映的实际问题的复杂性,例如,它所反映的自然规律,或是人类社会的事务,都具有一定的复杂性;另一方面,也可能来自程序逻辑结构的复杂性,软件开发,特别是应用软件的开发常常涉及到其它领域的专门知识,这对软件人员提出了很高的要求。软件的复杂性与软件技术的发展不相适应的状况越来越明显。图1.2显示出软件技术的发展落后于复杂的软件需求。

7) 软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动,它的成本比较高。问题不仅于此,值得注意的是硬件软件的成本近30年来发生了戏剧性的变化。无论研制也好,向厂家购买也好,在20世纪50年代末,软件的开销大约占总开销的百分之十几,大部分成本要花在硬件上;但80年代这个比例完全颠倒过来,软件的开销大大超过硬件的开销,如图1.3所示。今天的情况更是这样。美国每年投入软件开发的经费要有几百亿美元。然而,也并非在所有软件开发上的花费都能获得成果。

8) 相当多的软件工作涉及到社会因素。许多软件的开发和运行涉及机构、体制及管理方式等问题,甚至涉及到人的观念和人们的心理。对于这些人的因素重视得不够,常常是软件工作遇到的问题之一。例如,由于主管部门对正在开发的软件不够理解,因而软件

开发得不到应有的重视和必要的支持，造成人力和资金上的困难，它直接影响到项目的成败。

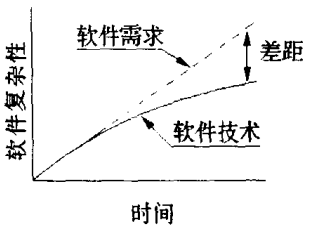


图 1.2 软件技术的发展落后于需求

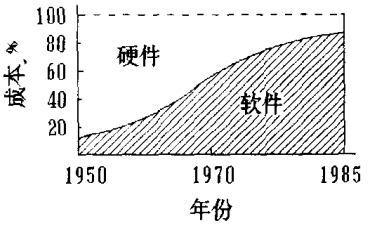


图 1.3 计算机系统硬、软件成本比例的变化

1.1.2 软件的分类

以上讨论的是区别于计算机硬件或其它工程对象的各种软件的共同特点。下面讨论软件类型。事实上，要给软件做出科学的分类很难，但鉴于不同类型的工程对象，对其进行开发和维护有着不同的要求和处理方法，因此仍需要对软件类型进行必要的划分。

1）按软件的功能进行划分

·系统软件：能与计算机硬件紧密配合在一起，使计算机系统各个部件、相关的软件和数据协调、高效地工作的软件。例如，操作系统、设备驱动程序以及通信处理程序等。系统软件的工作通常伴随着：频繁地与硬件交往、资源的共享与复杂的进程管理，以及复杂数据结构的处理。系统软件是计算机系统必不可少的一个部分。

·支撑软件：是协助用户开发软件的工具软件，其中包括帮助程序人员开发软件产品的工具，也包括帮助管理人员控制开发进程的工具。表 1.1 给出了一些支撑软件的实例。

·应用软件：是在特定领域内开发，为特定目的服务的一类软件。现在几乎所有领域都使用了计算机，为这些应用领域服务的应用软件种类繁多。例如商业数据处理软件、工程与科学计算软件、计算机辅助设计/制造（CAD/CAM）软件、系统仿真软件、智能产品嵌入软件（如汽车油耗控制、仪表盘数字显示、刹车系统），以及人工智能软件（如专家系统、模式识别）等。而在事务管理、办公自动化方面的软件也在企事业机关迅速推广，中文信息处理、计算机辅助教学（CAI）等软件使得计算机向家庭普及，甚至连娃娃也能在计算机上学习和游戏。

2）按软件规模进行划分

按开发软件所需的人力、时间以及完成的源程序行数，可确定 6 种不同规模的软件，如表 1.2 所示。

·微型：只是一个人，甚至是半日工作，在几天内完成的软件。写出的程序不到 5 百行语句，仅供个人专用。通常这种小题目无需做严格的分析，也不必要有一套完整的设计、测试资料。但事实说明，即使这样小的题目，如果经过一定的分析、系统设计、结构化编码以及有步骤地测试，肯定也是非常有益的。

表 1.1 支撑软件举例

一般类型	支持需求分析
文本编辑程序	PSL/PSA 问题描述语言、问题描述分析程序
文件格式化程序	关系数据库系统
磁盘向磁带向数据传输的程序	一致性检验程序
程序库系统	CARA 计算机辅助需求分析程序
支持设计	支持实现
图形软件包	编辑程序
结构化流程图绘图程序	交叉编辑程序
设计分析程序	预编译程序
程序结构图编辑程序	连接编辑程序
支持测试	支持管理
静态分析程序	PERT 进度计划评审方法绘图程序
符号执行程序	标准检验程序
模拟程序	库管理程序
测试覆盖检验程序	

表 1.2 软件规模的分类

类别	参加人员数	研制期限	产品规模(源程序行数)
微型	1	1~4 周	0.5k
小型	1	1~6 月	1k~2k
中型	2~5	1~2 年	5k~50k
大型	5~20	2~3 年	50k~100k
甚大型	100~1000	4~5 年	1M(=1000k)
极大型	2000~5000	5~10 年	1M~10M

·小型:一个人半年内完成的 2 千行以内的程序。例如,数值计算问题或是数据处理问题就是这种规模的课题。这种程序通常没有与其它程序的接口。但需要按一定的标准化技术、正规的资料书写以及定期的系统审查。只是没有大题目那样严格。

·中型:5 个人以内在一年多时间里完成的 5 千到 5 万行的程序。这种课题出现了软件人员之间、软件人员与用户之间的联系、协调的配合关系问题。因而计划、资料书写以及技术审查需要比较严格地进行。这类软件课题比较普遍,许多应用程序和系统程序就是这样的规模。在开发中使用系统的软件工程方法是完全必要的。

·大型:5 至 10 个人在两年多的时间里完成的 5 万到 10 万行的程序。例如编译程序、小型分时系统、应用软件包、实时控制系统等。对于这样规模的软件,采用统一的标准,实行严格的审查是绝对必要的。由于软件的规模庞大以及问题的复杂性,往往会在开发的过程中出现一些事先难于做出估计的不测事件。