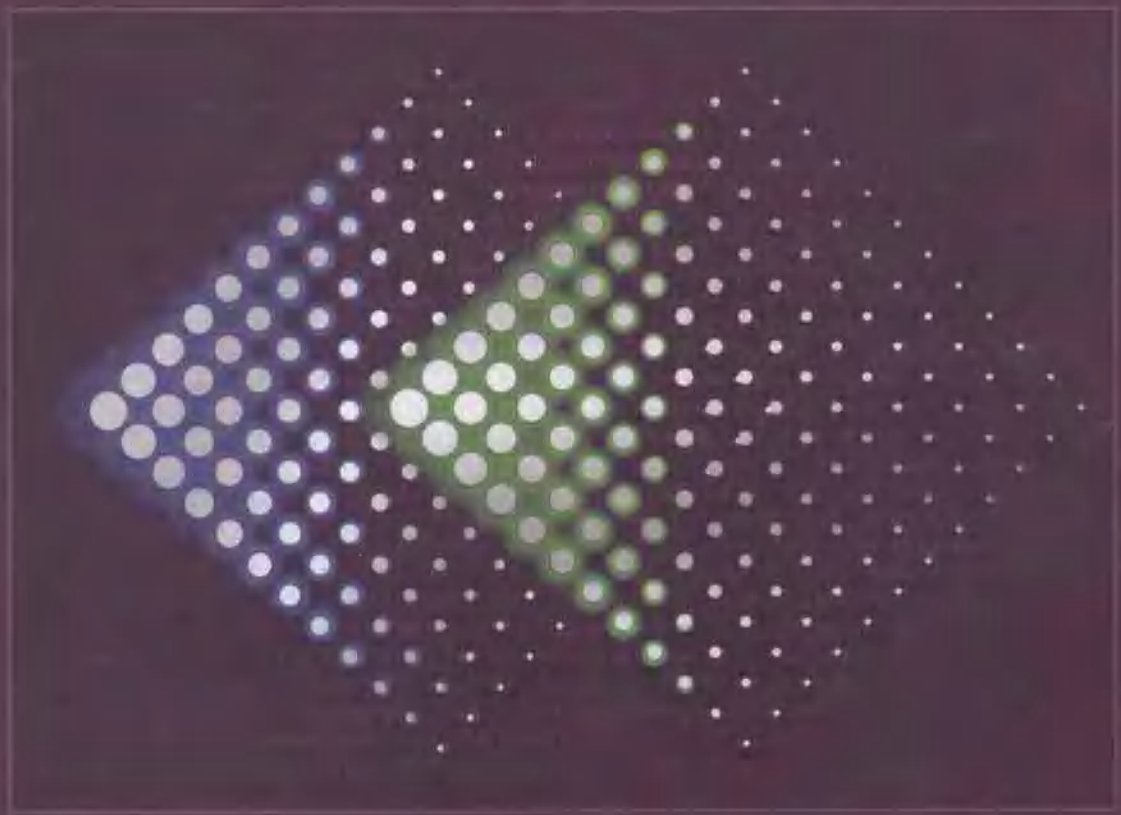
新编计算机类本科规划教材

新编软件工程实用教程

周丽娟　王华　编著



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

新编计算机类本科规划教材

新编软件工程实用教程

周丽娟　王华　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容提要

本书从方法学的角度出发，概述了软件生存周期模型和各种开发方法，介绍了结构化的设计方法；在传统软件开发方法的基础上，结合当前软件工程的理论和实践，以面向对象技术和 UML 语言为主线，详细介绍了软件工程的技术方法和实践原则；讨论了软件维护和软件工程管理技术。本书既有理论深度，又有具体的操作方法和案例分析，不仅介绍了软件工程的概念、原理、方法和技术，而且强调了方法和技术的实际应用。本书突出实际技能的培养，用一个综合性的实例贯穿始终，逐步实现一个完整的系统设计，可使读者真正做到学以致用。

本书可作为高等院校计算机及相关专业软件工程课程的教材或参考书，也可供软件工程师、软件项目管理者和应用软件的开发人员阅读参考。

图书在版编目(CIP)数据

新编软件工程实用教程 / 周丽娟等编著. —北京：电子工业出版社，2008.6
（新编计算机类本科规划教材）
ISBN 978-7-121-06450-0

Ⅰ. 新… Ⅱ. 周… 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(2008)第 057074 号

责任编辑：谭海平　　特约编辑：张荣琴
印　　刷：北京季蜂印刷有限公司
装　　订：三河市万和装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1092　1/16　印张：17　字数：435 千字
印　　次：2008 年 6 月第 1 次印刷
定　　价：26.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

随着信息技术的不断发展，软件已成为信息技术的核心。软件工程对软件产业的发展起到了重要的技术保证和促进作用。

软件工程首先是为了解决软件危机而提出的，现在已成为计算机科学技术的一个重要分支。20世纪90年代以来，软件工程不仅从方法学的角度为管理人员和开发人员提供了可见的结构和有序的思考，而且大量的成功软件总结出的设计经验，使软件开发人员可以充分利用设计模式、框架和部件等。软件工程正在逐步发展为一门成熟的专业学科。

本书的特点在于理论、方法与实践相结合，对传统的软件工程知识点进行了适当的删减，以面向对象方法学为技术主线，从实践角度介绍了软件工程的基本概念、基本原理、实用的开发方法和技术。

本书利用案例分析贯穿全书，对软件的分析、设计、实现、测试到维护过程进行全面的讲述，力求使读者在学习基本理论和技术过程中掌握软件工程的方法以解决应用问题。

本书作者十余年来一直从事软件工程的教学和科研工作，积累了丰富的教学经验和实践开发经验。在本书的编写中，作者注重内容的新颖性，结构的条理性，力图反映软件工程领域的最新发展，并从实用性出发，对各章节均结合实例进行讲解，深入浅出，使读者易于理解和掌握。希望能使读者通过本书的学习，对软件工程理论有一个较全面的理解，并对实际的软件工程活动有所帮助。

全书共有13章。第1章概述了软件工程的基本概念和发展历史，介绍了软件危机的原因和解决途径。第2章介绍了软件过程的基本活动、软件开发模型和开发方法。第3章介绍了软件工程的需求分析过程。第4章、第5章介绍了传统软件工程的分析、设计、实现的基本理论和方法。第6章、第7章、第8章以面向对象技术为核心内容，贯穿一个实例，全面、系统地介绍了软件开发各阶段的方法和技术。第9章介绍了软件测试方法，包括传统软件测试和面向对象软件测试方法。第10章介绍了软件维护的概念和方法。第11章介绍了软件项目管理、过程管理、质量管理等方法。第12章介绍了基于构件的软件开发方法。第13章介绍了Web工程，包括Web分析、设计、测试和项目管理等方面的内容。

本书第3章、第6章、第7章、第8章、第9章、第13章由周丽娟编写，第1章、第2章、第4章、第5章、第10章、第11章、第12章由王华编写。

目前，国内外有关软件工程技术与设计方面的资料很多，新理论、新技术层出不穷。因时间和水平有限，书中存在不周到和不准确之处，恳请专家学者提出宝贵意见，以便进一步完善。

作者

目　录

第1章 概 述

计算机技术把人类社会带入了一个崭新的"信息时代",给人们的工作和生活带来了巨大变化。然而,作为信息化基础的软件技术发展还不成熟,至今仍然受到软件危机的困扰。人们开发优质软件的能力大大落后于计算机硬件日新月异的进展和社会对计算机软件不断增长的需求。为了摆脱软件危机的困扰,一门研究软件开发与维护的原理和技术的工程学科——软件工程学,从20世纪60年代末逐步发展起来。目前,软件工程已经成为高等院校计算机专业的必修课程。

1.1 软件的概念和特征

软件产品和服务变得越来越不可少,大多数行业的业务越来越多地依赖于软件,开发并利用软件强大的能力已成为新经济中各国竞争的要素。

1.1.1 软件的概念

1. 软件

软件这个概念,从它出现之时,就带有一层神秘的色彩。其高度的抽象性使人们无法从物理实体上感知它、认识它。那么什么是软件呢?早期人们对软件的定义是:用来完成某些任务的程序和数据的集合。程序是指令的序列,指令是能在计算机硬件上执行的动作。而数据是指令操作的对象。随着技术的发展,人们对软件的认识也在加深。今天,从软件工程的角度看,软件的定义应当是,软件是完成某类问题求解的程序和数据,以及为维护程序必须提供的一系列文档组成的集合。用简洁的公式可表示为

$$\text{软件} = \text{程序} + \text{数据} + \text{系列文档}$$

(1) 软件的内部性质是,软件具有高度的抽象性和严密的逻辑性。

软件是问题求解方法的信息表达形式,而问题求解方法(计算机中称为算法)是高度抽象的思维活动。高度的抽象性是软件与生俱来的本性,人们无法直接感知软件,必须通过认识、理解、判断、推理等一系列复杂的思维过程才能感知它、认识它、理解它,从而得到它。

软件是大量逻辑元素的复杂组合。这些逻辑元素可以是变量、数组、记录、文件、标号、常数等数据结构,也可以是循环、转移、条件、顺序、推理、赋值等控制机制,甚至还可以是环境、人、其他软件、硬件等外部元素。显然软件中涉及的逻辑量比硬件系统要多出10~100倍。为完成一个复杂的大型软件,常常需要建立一个庞大的逻辑体系。严密的逻辑性是指这个复杂的逻辑体系中,各种逻辑元素之间的联系必须是一致的、无矛盾的,体系结构及其表示必须是统一的。任何逻辑联系上的失误都将导致软件的错误,严重时将导致软件的失败。

(2) 软件的外部性质是指,软件是一种逻辑的信息产品,是用文字、符号表达的智力产物。软件的内部性质决定了软件的外部形式只能是逻辑的、思维的结果,其外在表达形式只能通过抽象的符号表达。人们要认识、理解或编制软件,只能通过建立相应的符号体系进行。程序设计语言是人们为了表达软件而设计的符号体系。

2. 软件的特征

由于软件本身具有的特殊性质，使得软件产品具有以下特征。

(1) 软件是一种逻辑实体，具有抽象性。这个特点使它与其他工程对象有着明显的差异。人们可以把它记录在纸、内存、磁盘和光盘上，但却无法看到软件本身的形态，必须通过观察、分析、思考和判断，才能了解它的功能、性能等特性。

(2) 软件没有明显的制造过程。软件一旦研制开发成功，就可以大量复制。为此，软件的质量控制重点必须放在软件开发方面。

(3) 软件在使用过程中没有磨损、老化的问题。软件在生存后期不会因为磨损而老化，但会为了适应硬件、运行环境及需求的变化而进行修改，这些修改又不可避免地引入错误，从而导致软件失效率升高，使得软件退化。当修改的成本变得难以接受时，软件就会被抛弃。

(4) 软件对硬件和运行环境有着不同程度的依赖性，因而可移植性差。

(5) 软件的开发至今尚未完全摆脱手工作坊式的开发方式，生产效率低。

(6) 软件是复杂的，而且以后会更加复杂。软件是有史以来人类生产的复杂度最高的工业产品。软件涉及人类社会的各行各业和方方面面，软件开发常常涉及其他领域的专业知识，这对软件工程师提出了很高的要求。

(7) 软件的成本相当昂贵。软件开发需要投入大量高强度的脑力劳动，成本非常高，风险也很大。软件的开销已大大超过了硬件的开销。

(8) 软件开发工作牵涉到很多社会因素，如机构设置、体制和管理方式，以及人们的观念和心理等。这些因素常常会成为软件开发的难点，并直接影响到项目的成败。

1.1.2 软件的分类

目前对软件进行分类还找不到一个统一的分类标准，但可以从不同的角度去考察软件。

1. 基于软件功能的划分

软件按其功能可划分为以下 3 种。

(1) 系统软件。系统软件是能与计算机硬件紧密配合，使计算机内各个部件、相关软件和数据协调、高效工作的软件，如操作系统、设备驱动程序等。

(2) 支撑软件。支撑软件是协助用户开发软件的工具性软件，其中包括帮助程序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，如编辑程序、程序库、图形软件包等。

(3) 应用软件。应用软件是在特定的领域内开发、为特定目的服务的一类软件，如工程科学计算软件、CAD/CAM 软件、CAI 软件、信息管理系统等。

2. 基于软件工作方式的划分

软件按其工作方式可划分为以下 4 种。

(1) 实时处理软件。实时处理软件是指在事件或数据产生时，立即处理并及时反馈信号，控制需要监测控制过程的软件。

(2) 分时处理软件。分时处理软件是允许多个用户同时使用计算机的软件,系统把处理时间分配给各个联机用户。

(3) 交互式软件。交互式软件是实现人机通信的软件。

(4) 批处理软件。批处理软件是把一组作业或一批数据以成批处理的方式一次运行,按顺序处理的软件。

3. 基于软件规模的划分

如表1.1所示,按照开发软件所需的人力、时间以及完成的源程序代码行数进行划分,可分为6种不同规模的软件:微型、小型、中型、大型、甚大型和极大型。需要说明的是,随着软件产品规模的不断增大,类别的指标也会发生变化。无论哪一种规模的软件项目,都需要有软件工程的知识作为指导,并遵循一定的开发规范;即其基本原则是一样的,只是对软件工程技术依赖的程度不同而已。

表1.1 软件规模分类

类 别	参加人数	研制期限	产品规模(源代码行)
微型	1	1~4周	500
小型	1	1~6月	1000~2000
中型	2~5	1~2年	5000~50 000
大型	5~20	2~3年	5000~5000 000
甚大型	100~1000	4~5年	1000 000
极大型	2000~5000	5~10年	1 000 000~10 000 000

4. 基于软件失效的影响进行划分

工作在不同领域的软件,在运行中对可靠性也有不同的要求。事实上,随着计算机进入国民经济各个重要领域,软件的可靠性越来越重要。人们一般称这类软件为关键软件,这类软件具有以下特点。

(1) 可靠性质量要求高。

(2) 常与完成重要功能的大系统的处理部件相连。

(3) 含有的程序可能对人员、公众、设备或设施的安全造成影响,还可能影响到环境的质量和国家的安全。

5. 基于软件服务对象的范围进行划分

软件工程项目完成后可以有如下两种结果。

(1) 项目软件。项目软件是受某个特定客户(或少数客户)的委托,由一个或多个软件开发机构在合同的约束下开发出来的软件。

(2) 产品软件。产品软件是由软件开发机构开发出来直接提供给市场的软件,或是为众多用户服务的软件。

1.1.3 软件的发展

1. 程序设计阶段

计算机发展的早期阶段(20世纪50年代初期至60年代中期)为程序设计阶段。在这个阶段硬件已经通用化,而软件的生产却是个体化的。这时,由于程序规模小,几乎没有什么系统化的方法可以遵循。因此对软件的开发没有任何管理方法,一旦计划推迟了或者成本提高了,程序员才开始弥补。在通用硬件已经非常普遍的时候,软件产品还处在初级阶段,对每一类应

用均需自行再设计，使得应用范围很有限。设计往往仅是人们头脑中的一种模糊想法，而文档根本不存在。

2. 程序系统阶段

计算机系统发展的第二阶段(20 世纪 60 年代中期到 70 年代末期)为程序系统阶段。多道程序设计、多用户系统引入了人机交互的新概念。交互技术打开了计算机应用的新世界，使硬件和软件的配合达到了一个新层次，实时系统和第一代数据库管理系统的出现就是这种配合的体现。这个阶段还有一个特点，即软件产品的使用和"软件作坊"的出现。被开发的软件可以在较宽广的范围中应用。主机和微机上的程序能够有数百甚至上千的用户。

在软件的使用过程中，当发现程序错误、用户需求和硬件环境发生变化时都需要修改软件，这些活动统称为软件维护。在软件维护上的花费正以惊人的速度增长。更为严重的是，许多程序的一体化特性使得它们根本不能维护，从而出现了"软件危机"。

3. 软件工程阶段

计算机系统发展的第三阶段(20 世纪 70 年代中期至 20 世纪 90 年代)为软件工程阶段。在这一阶段，以软件的产品化、系列化、工程化、标准化为特征的软件产业发展起来，打破了软件生产的个体化特征，有了可以遵循的软件工程化的设计原则、方法和标准。在分布式系统中，各台计算机同时执行某些功能，并与其他计算机通信，极大地提高了计算机系统的复杂性。广域网、局域网、高带宽数字通信以及对"即时"数据访问需求的增加都对软件开发者提出了更高的要求。

纵观软件的发展，可以看出：软件的数量越来越多，规模越来越大，需求越来越广，成本越来越高。

4. 软件发展的第四阶段

Internet 技术的迅速发展使软件系统从封闭走向开放，Web 应用成为人们在 Internet 上最主要的应用模式，异构环境下分布式软件的开发成为一种主流需求，软件复用和构件技术成为技术热点。与此同时，需求工程、软件过程、软件体系结构等方面的研究也取得了新的进展。

进入 21 世纪，Internet 正在向智能网络时代发展，以网格计算(Grid Computing)和 Web 服务(Web Services)为代表的分布式计算日趋成熟，从而实现信息充分共享和服务无处不在的应用环境；高信度计算(Trustworthy Computing)普遍引起了人们的高度重视，从而创造一个安全、可靠、值得信赖的环境创造了条件。

表 1.2 给出了软件发展的 4 个时期及其特点。

表 1.2 计算机软件发展的 4 个时期及其特点

时间 / 特点	程序设计 (20 世纪 50～60 年代)	程序系统 (20 世纪 60～70 年代)	软件工程 (20 世纪 70～80 年代)	第四阶段
软件所指	程序	程序及说明书	程序、文档、数据	软件工程定义
主要程序设计语言	汇编及机器语言	高级语言	软件语言	软件语言及辅助工具
软件工作范围	程序编写	包括设计和测试	软件生存期	软件生存周期
需求者	程序设计者本人	少数用户	市场用户	市场用户
开发软件的组织	个人	开发小组	开发小组及大中型软件开发机构	开发小组及大中型软件开发机构
软件规模	小型	中小型	大中小型	大中小型

（续表）

时间 / 特点	程序设计（20世纪50～60年代）	程序系统（20世纪60～70年代）	软件工程（20世纪70～80年代）	第四阶段
决定质量的因素	个人程序技术	小组技术水平	管理水平	管理水平
开发技术和手段	子程序 程序库	结构化程序设计	数据库、开发工具、开发环境、工程化开发方法、标准和规范、网络及分布式开发、面向对象技术、嵌入式技术	Web应用、异构环境下分布式开发、软件复用和构件技术、智能技术应用、信息共享和服务
维护责任者	程序设计者	开发小组	专职维护者	维护小组
硬件特征	价格高 存储容量小 工作可靠性差	价格下降、速度、容量及工作可靠性有明显提高	向超高速、大容量、微型化及网络化方向发展	微型化、网络计算机
软件特征	完全不受重视	软件技术的发展不能满足需要，出现软件危机	开发技术有进步，但未获突破性进展，价格高，未完全摆脱软件危机	先进的软件开发和管理工具不断出现，软件过程控制及管理、CMM、CMMI

1.2 软件危机

1.2.1 软件危机的主要表现

1. 软件危机的含义

软件危机指的是软件开发和维护过程中遇到的一系列严重问题。软件危机包含下述两方面的问题：如何开发软件，怎样满足对软件的日益增长的需求；如何维护数量不断膨胀的已有软件。

2. 软件危机的表现

软件危机主要有下列表现。

(1) 软件产品不符合用户的实际需要。由于软件开发人员对用户需求没有深入、准确的了解，甚至对所要解决的问题还没有正确的认识，就着手编写代码，而且软件开发人员和用户之间的信息交流又很不充分，从而导致用户对软件产品不满意的情况时有发生。

(2) 软件开发生产率提高的速度远远不能满足客观需要。软件的生产率远远低于硬件的生产率和计算机应用的增长率，使人们不能充分利用现代计算机硬件提供的巨大潜力。

(3) 软件产品的质量差。软件可靠性和质量保证的定量概念刚刚出现不久，软件质量保证技术（审查、复审及测试）没有贯穿到软件开发的全部过程中，这些都将导致软件产品发生质量问题。

(4) 对软件开发成本和进度的估计常常不准确。实际成本比估计成本有可能偏高，实际进度比预期进度推迟。这种现象降低了软件开发者的信誉级别。为了赶进度和节约成本所采取的一些权宜之计又往往降低了软件产品的质量，从而不可避免地引起用户的不满。

(5) 软件的可维护性差。程序中的很多错误是难以改正的，这些程序实际上不能适应硬件环境的改变，也不能根据用户的需要在原有程序中增加一些新的功能。软件的不可重用性，造成了重复开发功能类似的软件这种结果。

(6) 软件文档资料通常不完整、不合格。计算机软件不应仅有程序，还应该包括一整套文档资料。这些文档资料应该是在软件开发过程中产生出来的，而且应该和程序代码完全一致。软件开发的管理人员可以用文档资料来管理和评价软件开发过程的进展状况，如软件开发人员可以利用它们作为通信工具，在软件开发过程中准确地交流信息。对于软件维护人员而言，文档资料是至关重要的。

(7) 软件的价格昂贵。软件成本在计算机系统总成本中所占的比例逐年上升。由于微电子学技术的进步和生产自动化程度的不断提高，导致了硬件成本逐年下降，然而软件开发则需要大量人力，从而使软件成本上升。

通过对以上软件危机表现的分析，可以看出，在软件开发和维护过程中存在严重的问题。这些问题一方面与软件本身的特点有关，另一方面也和软件开发与维护的方法有关。

以下是两个软件危机的实例。

① IBM OS/360 系统耗资超过 4000 万美元，用工 5000 人·年，当投入运行仅一个月时便发现了 2000 个以上的错误。该系统负责人 F.D.Brooks 描述道："就像一头巨兽在泥潭中垂死挣扎，挣扎得越猛，下陷得便越快"。

② 1981 年 10 月美国航天飞机(哥伦比亚号)首次发射前 20 分钟，因突然发现软件错误而导致发射推迟。

1.2.2 产生软件危机的原因

根据软件危机的种种表现，分析其产生的原因，大致为以下几方面。

(1) 软件不同于硬件，它是计算机系统中的逻辑部件而不是物理部件。由于软件缺乏"可见性"，在写出程序代码并在计算机上试运行之前，软件开发过程的进展情况较难衡量，很难检验开发的正确性，而且软件开发的质量也难以评价。因此，管理和控制软件开发过程相当困难。此外，如果在软件运行过程中发现错误，它很可能是在开发时期引入的错误，而在测试阶段没能检测出来。

软件不同于一般程序，它的一个显著特点是规模庞大。如何保证系统中每个开发人员完成的工作整合起来确实构成一个高质量的大型软件系统，更是一个极端复杂、困难的问题。它不仅涉及许多技术问题，还涉及严格、科学的管理问题，其开发和维护必然相当困难。

(2) 目前相当多的软件从业人员对软件开发和维护还有不少错误的观念，在实践过程中没有采用工程化的方法，这是产生软件危机的主要原因。软件本身独有的特点确实给开发和维护带来了一些客观困难，但是人们在开发和使用计算机系统的长期实践中也确实积累了许多成功的经验。如果坚持不懈地使用实践证明正确的方法，许多困难是完全可以克服的。

(3) 开发和管理人员只重视开发而轻视问题的定义，使软件产品无法满足用户的需求。对用户要求没有完整准确的认识就匆忙着手编写程序，是许多软件开发工程失败的主要原因之一。事实上，只有用户才真正了解他们自己的需要。软件开发人员需要做大量深入细致的调查研究工作，反复多次地与用户交流信息，才能真正全面、准确、具体地了解用户的需求。软件开发的基本过程应该是先从软件开发最初的问题定义