面向21世纪课程教材

Textbook Series for 21st Century

软件工程

（第二版）

齐治昌　谭庆平　宁　洪

高等教育出版社

内 容 提 要

软件工程第二版覆盖 CC2001 对软件工程课程的基本要求，内容包括：软件开发模型、软件项目管理、需求分析、软件设计、软件编码与测试、软件维护、配置管理、软件工程管理、软件工程工具和环境。本书结合目前软件工程教学的需要，特别介绍了统一建模语言（UML），并以此为基础讨论面向对象的需求分析与软件设计方法；介绍集成化 CASE 环境，Java 与 Internet 环境下的软件开发技术；介绍支持软件过程改进的“软件能力成熟度模型”（CMM）。书中含有丰富的例题与习题，便于教学和自学。

本书强调软件工程的理论与实践相结合、技术与管理相结合、方法与 CASE 工具相结合。教材思路清晰、语言简练，具有实用性和可操作性，可作为高等院校计算机专业或信息类相关专业高年级本科生或研究生教材，也可作为软件开发人员的参考书。

第一版序言

软件工程是计算机学科中一个年轻并且充满活力的研究领域。自20世纪60年代末期以来,人们为克服“软件危机”在这一领域做了大量工作,逐渐形成了系统的软件开发理论、技术和方法,它们在软件开发实践中发挥了重要作用。今天,现代科学技术将人类带入了信息社会,计算机软件扮演着十分重要的角色,软件工程已成为信息社会高技术竞争的关键领域之一。

“软件工程”是高等院校计算机教学计划中的一门核心课程,主要内容包括支持软件开发和维护的理论、方法、技术、标准以及计算机辅助工具和环境。这些内容对于软件研制人员、软件项目管理人员都是必需的。本书比较全面、系统地反映了软件工程课程的全貌,既兼顾了传统的、实用的软件开发方法,又介绍了软件工程领域比较新颖的技术和方法,包括面向对象的需求分析与软件设计方法,实时系统的分析与设计技术,软件重用技术,快速原型技术和集成化软件开发环境。对计算机辅助软件工程(CASE)技术的介绍贯穿全书的始终,对近年来兴起的主要研究热点,如Java与Internet环境下的软件开发、对象链接与嵌入(OLE)技术、通用对象请求代理机制(CORBA)与分布计算技术等,都做了专门介绍,并附有较完整的参考文献,以供有兴趣的读者进一步学习和研究。本书的另一重要特点是理论与实践相结合,软件工程的技术、方法与CASE工具相结合。全书内容的选材强调实用价值和可操作性,强调CASE工具和环境对软件开发全过程的支持。此外,作者在本书的写作过程中深感软件工程的内容较易流于琐碎、冗繁,往往给读者造成软件工程缺乏深度、缺乏内在逻辑关联等印象。所以,本书力求语言的精炼,注重内容的条理性、系统性和逻辑性。当然,作者的这一初衷是否已经实现还取决于读者的评价。

全书共二十章,大致分为五个部分。前三章构成本书的第一部分,内容包括:软件工程的基本概念,软件开发模型,软件的度量与估算,开发过程管理与质量控制,计算机系统工程。第四至七章构成本书的需求分析部分,内容包括:需求分析的任务与原则,面向数据流、面向对象与面向数据的需求分析方法,形式化方法,需求规格说明与评审。第八至十二章构成本书的软件设计部分,内容包括:软件设计过程与一般性技术,面向数据流、面向对象与面向数据的设计方法,人机界面设计技术,设计规格说明与评审。第十三至十六章构成本书的软件实现与维护部分,内容包括程序设计语言与编码,软件测试与排错,软件维护,逆向工程与重构工程,软件配置管理。第十七至二十章构成本书的最后一部分,专门介绍比较新颖、颇具发展潜力的软件开发技术,包括:软件重用技术、快速原型技术、集成化CASE环境以及新近出现的热点技术。

第二版序言

近年来,由于计算机硬件、软件、网络的飞速发展和广泛应用,人们迎来了信息社会和知识经济。我国进入了以信息化带动工业化、以工业化促进信息化的新的历史时期。而软件产业肩负着发展信息产业、对传统产业进行信息化改造的历史任务。为适应这一形势发展的需要,我国加快软件人才培养的步伐,扩大软件人才培养规模,深化软件教学改革,加强计算机应用教育,扩大对外交流,短短几年取得了长足进步。

自 1997 年本书问世以来,软件工程领域涌现了许多新的技术和方法,其中尤以软件能力成熟度模型(CMM)、统一建模语言(UML)以及基于 UML 的面向对象软件开发方法的影响最为深远。与此同时,"软件工程"课程在高等院校计算机教学计划中的重要性也与日俱增,ACM、IEEE - CS 联合推出的 CC2001 将软件工程作为计算机学科的 14 个知识领域之一,还专门制定了软件工程学科方向的教学计划。

为了适应高等院校软件工程教学工作的需要,我们对"软件工程"教材进行了修订。作者在保持第一版基本结构和风格的前提下,增加了软件能力成熟度模型(CMM)的有关内容,以统一建模语言(UML)为基础全部改写了面向对象的需求分析与软件设计两章,并对初版中发现的不妥之处进行了全面修订。希望这些措施能够促进"软件工程"课程的教学内容改革,提高学生的软件开发能力和项目管理能力。

本书主要供计算机及信息领域相关专业高年级本科生及硕士研究生作为软件工程课程的教材使用,同时,也适合软件开发人员与软件项目管理人员用作技术参考书。在教学计划中,如果讲授 60 学时,建议采用第一至第二十章的自然顺序讲授,其中带"*"的内容可酌情处理。实习以 12 至 20 学时为宜。如果以 40 学时讲授本书,对于高年级本科生,建议的教学内容及次序为:第一、二章→第四至六章→第八至十章→第十二至十五章→第十七、十八章;对于已在本科阶段学过软件工程的硕士研究生,建议的教学内容及次序为:第二、三章→第四、六章及第七章中的形式化方法→第十章→第十六章→第十七至十九章→*第二十章。根据培养目标和学生的实际情况,实习以 10 至 20 学时为宜。

"软件工程"第一版问世以来,得到了很多高等院校计算机专业老师和同学的大力支持,2000 年曾荣获教育部科技进步(教材)二等奖,并被教育部研究生工作办公室推荐为"研究生教学用书",2003 年被列入"高等教育百门精品课程教材建设计划"。借此机会,我们再次向为本书付出辛勤劳动的高等教育出版社的领导和编辑、所有使用本书进行教学的老师和同学,以及对本书提出建议和意见的专家和读者表示诚挚的谢意。欢迎大家继续对本书的新版给予支持和指导。

作　者

2004 年 2 月

本书主要供计算机专业高年级本科生和低年级硕士研究生作为软件工程课程的教材使用,其内容满足国家教育委员会颁布的计算机专业软件工程课程教学基本要求。同时,本书也适合于软件开发人员与软件项目管理人员作为技术参考书使用。在教学计划中,如果安排60学时,建议采用第一至第二十章的自然顺序讲授,其中带"*"的内容可酌情处理。实习以8至12学时为宜。如果以40学时讲授本书,对于高年级本科生,建议的教学内容及次序为:第一、二章→第四至六章→第八至十章→第十二至十五章→第十七、十八章;对于已在本科阶段学过软件工程的硕士研究生,建议的教学内容及次序为:第二、三章→第四、六章及第七章中的形式化方法→第十六章→第十七至十九章→*第二十章。实习以6至10学时为宜。

齐治昌教授组织了本书的编写工作,并撰写了前三章。第四至第七章及第十七至二十章由谭庆平博士撰写,第八至十六章由宁洪副教授撰写。本书内容曾在国防科技大学计算机专业本科生和硕士研究生的教学中讲授过。

西北大学郝克刚教授认真审阅了本书的全部初稿,并提出了许多中肯的修改意见。国防科技大学计算机系陈怀义教授、殷建平博士也对书稿进行了认真细致的初审,并提出了很好的意见和建议。陈火旺教授在本书的成书过程中给予了多方面指导。此外,国防科学技术大学计算机系领导对本书的编写工作给予了有力支持,计算机系很多教师和学生的鼓励和建议也对本书的出版起了推动作用。在此,作者向所有对本书编写工作给予支持和帮助的人表示衷心的感谢。最后,作者还要特别感谢施伯乐教授、许卓群教授以及国家教育委员会计算机科学教学指导委员会的其他专家们,没有他们的信任、鼓励和支持,本书是不可能问世的。

最后,诚恳欢迎各位读者对本书的缺点、错误提出批评。

作　者

1997年4月

目　录

第一章! 软件与软件工程

计算机的发展一直与微电子技术的进步紧密地联系在一起。自"#$%年计算机诞生以来,它已走过了电子管、晶体管、集成电路、大规模集成电路的时代,如今,超大规模集成电路在计算机工业界已得到广泛的应用。据统计,计算机的性能平均每"&个月提高一倍,呈几何级数增长。如果这样的趋势再持续'(年,计算机的性能将比现在提高"((((倍。然而,随着计算机性能的提高,计算机的体积、功耗和价格却不断下降。今天的微处理器和个人计算机已广泛应用于办公自动化、工业自动控制、商业信息处理和家用电器控制等领域,大型机、巨型机正成功地担负着数值天气预报、石油勘探、地震数据处理、航天飞机发射等领域的任务。计算机和电视、汽车、电话一样,已成为人们工作和生活中不可缺少的工具。

随着计算机技术、电子技术的惊人进步,计算机与全球互连网络)*+,-*,+相连接,使今天的社会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、处理、交流和决策都需要大量高质量的计算机软件,这样就促使人们对计算机软件的品种、数量、功能、质量、成本和开发时间等提出越来越高的要求。为了使世界上丰富的软件资源为人类共享,人们越来越重视软件、软件开发及运行环境的标准化。计算机的各类程序设计语言和多媒体人机交互工具已被越来越多的人所掌握,成为世界性的文化现象。

然而,不幸的是,要想使软件功能越强、使用越方便,开发出来的软件就越复杂、越庞大,人们的软件开发能力越显得力不从心,以致软件开发计划一拖再拖,成本失去控制,软件质量得不到保证。为了扭转这种被动局面,自'(世纪%(年代末期以来,人们十分重视软件开发方法、工具和环境的研究,并在这些领域取得了重要的成果。

这一章将介绍软件和软件工程的基本概念,包括软件、软件工程、软件开发过程与模型、软件工具与环境,等等。

"."! 软! ! 件

本节讨论软件的定义、组成、特点和软件的分类,介绍软件的发展过程与软件危机。

!"!"!# È¡¼þÓëÈ¡¼þµÄ×é³É

计算机软件是与计算机系统操作有关的程序、规程、规则及任何与之有关的文档和数

据。它由两部分组成：一是机器可执行的程序及有关数据；二是机器不可执行的、与软件开发、运行、维护、使用和培训有关的文档。

程序(program)是用程序设计语言描述的、适合于计算机处理的语句序列。它是软件开发人员根据用户需求开发出来的。程序设计语言编译器可以将程序翻译成一组机器可执行的指令。这组指令亦称机器语言程序，它将根据用户的需求，控制计算机硬件的运行，处理用户提供的或机器运行过程中产生的各类数据并输出结果。为了对程序设计语言进行机器自动翻译，人们不得不限制程序设计语言的词汇范围(如字符集、关键字等)，并用良好的形式规则精确地定义程序设计语言的语法和语义。目前的程序设计语言有三种类型：依赖于具体计算机的机器语言、汇编语言，独立于机器的面向过程的语言，以及独立于机器的面向问题的语言。机器语言是用中央处理器(CPU)指令集表示的符号语言，优秀的软件开发人员使用机器语言可以开发出时空开销较小的高质量程序。但用机器语言编写程序时，工作效率低，程序难以阅读和调试，不利于软件的维护，也难以在不同 CPU 系统中推广使用。高级语言与机器无关，其表达能力强，容易阅读和修改，大大提高了软件开发效率。高级语言的编译器或解释器是依赖于具体机器的，它把高级语言程序转换为机器语言程序，然后再运行。今天，世界上的程序设计语言有几百种，但广泛使用的高级语言不过十余种，如用于科学计算的 FORTRAN 语言，用于事务处理的 COBOL 语言，支持结构化程序设计的 Pascal 语言，支持现代软件开发的 C 语言、Ada 语言，还有支持面向对象设计方法的 C++语言、Java 语言等。机器语言、汇编语言和高级语言经常被称为前三代的计算机语言或面向过程的语言。用这些语言进行程序设计，程序员必须指明信息的结构和程序的控制流程。面向问题的语言不需要程序员指明程序实现的过程，只需给出问题和输入数据，并指出输出的形式，就可以得到所需结果。数据库查询语言、报表语言、机床控制专用语言和电路设计专用语言等都是面向问题的语言，也称为非过程式语言或四代语言(4GL)。

文档(document)是一种数据媒体和其上所记录的数据。文档记录软件开发的活动和阶段成果，它具有永久性并能供人或机器阅读。它不仅用于专业人员和用户之间的通信和交流，而且还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，提高软件产品的质量，许多国家对软件文档都制定了详尽、具体的规定，颁布了各种规范和标准。我国国家标准局也从 1988 年开始陆续颁布了《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》，等等。

1.1.2

软件是逻辑产品而不是物理产品，因此，软件在开发、生产、维护和使用等方面与硬件相比均存在明显的差异。

软件开发与硬件开发相比，更依赖于开发人员的业务素质、智力以及人员的组织、合作

进行处理并在规定的时间内做出反应的软件，称之为实时软件。实时软件依赖于处理机系统的物理特性，如计算速度和精度、I/O 信息处理与中断响应方式、数据传输效率等。支持实时软件的操作系统称为实时操作系统。实时软件使用的计算机语言有汇编语言、Ada 语言等。实时系统的服务经常是连续的，系统在规定的时间内必须处于能够响应的状态，因此，实时软件和计算机系统必须有很高的可靠性和安全性。

（3）嵌入式软件。嵌入式计算机系统将计算机嵌入在某一系统之中，使之成为该系统的重要组成部分，控制该系统的运行，进而实现一个特定的物理过程。用于嵌入式计算机系统的软件称为嵌入式软件。大型的嵌入式计算机系统软件可用于航空航天系统、指挥控制系统和武器系统等。小型的嵌入式计算机系统软件可用于工业的智能化产品之中，这时，嵌入式软件驻留在只读存储器内，为该产品提供各种控制功能和仪表的数字或图形显示功能等。例如，汽车的刹车控制，空调机、洗衣机的自动控制，等等。嵌入式计算机系统一般都要和各种仪器、仪表、传感器连接在一起，因此，嵌入式软件必须具有实时的采集、处理和输出数据的能力。这样的系统称之为实时嵌入式系统。它广泛应用于连续的动力学系统的控制与仿真。

（4）科学和工程计算软件。它们以数值算法为基础，对数值量进行处理和计算，主要用于科学和工程计算，例如数值天气预报、弹道计算、石油勘探、地震数据处理、计算机系统仿真和计算机辅助设计（CAD）等。这类软件大多数用 FORTRAN 语言描述，近年来有的也用 C 语言或 Ada 语言描述。它是使用最早、最广泛、最为成熟的一类软件。从 20 世纪 50 年代起，有经验的程序员就把许多常用算法用程序设计语言编制成标准程序，如今已经积累了大量的科学和工程计算软件。人们将各种软件按学科或应用领域分类，开发了各种程序库、软件包和软件系统。这些软件具有质量高、使用方便等特点，为计算机在科学和工程上的应用做出了重要贡献。

（5）事务处理软件。用于处理事务信息，特别是商务信息的计算机软件。事务信息处理是软件最大的应用领域，它已由初期零散的、小规模的软件系统，如工资管理系统、人事档案管理系统等，发展成为管理信息系统（MIS），如世界范围内的飞机订票系统、旅馆管理系统、作战指挥系统，等等。事务处理软件需要访问、查询、存放有关事务信息的一个或几个数据库，经常按某种方式和要求重构存放在数据库中的数据，能有效地按一定的格式要求生成各种报表。有些管理信息系统还带有一定的演绎、判断和决策能力。它们往往具有良好的人机界面环境，在大多数场合采用交互工作方式。它们需要交互式操作系统、计算机网络、数据库、文字/表格处理系统的支持。常用的语言有 COBOL、四代语言等。

（6）人工智能软件。支持计算机系统产生人类某些智能的软件。它们求解复杂问题时，不是采用传统的计算或分析方法，而是采用诸如基于规则的演绎推理技术和算法，在很多场合还需要知识库的支持。人工智能软件常用的计算机语言有 LISP 和 Prolog 等。迄今

为止,在专家系统、模式识别、自然语言理解、人工神经网络、程序验证、自动程序设计、机器人学等领域开发了许多人工智能应用软件,用于诊断疾病、产品检测、自动定理证明、图像和语音的自动识别、语言翻译等。

(7) 个人计算机软件。个人计算机上使用的软件也可包括系统软件和应用软件两类。近 20 年来,个人计算机的处理能力已提高了三个数量级,以前在中小型计算机上运行的系统软件和应用软件,如今已经大量移植到个人计算机上。在个人计算机上开发了大量的文字处理软件、图形处理软件、报表处理软件、个人和商业上的财务处理软件、数据库管理软件、网络软件、多媒体信息处理软件,等等。个人计算机软件的最大特色是人机界面采用了多窗口技术、多媒体技术,使个人计算机具有了用文字、图形、图像、声音进行人机交互的能力,为个人计算机的普及创造了必要条件。人们还将个人计算机与计算机网络连接在一起,进行通信和共享网络资源,加速了人类社会信息化的进程。随着社会的进步,个人计算机及其软件的发展、普及和应用前景将更加广阔。

(8) CASE 工具软件。计算机辅助软件工程是指软件开发和管理人员在软件工具的帮助下进行软件产品的开发、维护以及开发过程的管理。CASE 工具软件一般为支撑软件生存周期中不同活动而研制,包括项目管理工具、需求分析工具、编程环境(集编辑器、编译器、链接器和调试器于一体)、软件测试工具,等等。为了方便工具之间交换信息,它们一般并不独立存在,而是按某种方式集成为一个环境。关于集成化 CASE 环境的讨论请见第十九章。

1.1.4 软件的发展

除去像 Ada Augusta Byron(1816—1851)早期的程序设计工作外,从 20 世纪 40 年代世界上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 50 年代初期至 20 世纪 60 年代初期的十余年,是计算机系统开发的初期阶段。这时的通用计算机由于价格昂贵、体积大、功耗高、机器不稳定和需要专人维护等原因,只能放在公共的实验室内供大家使用。计算机系统中的多数软件是用户自己设计、自己使用、自己维护。软件规模小,文档不完全。开发过程的中间结果经常保留在程序员的脑子里,软件开发过多地依靠程序员个人的‘ 技艺 ”。软件开发不规范,生产率低,开发过程很难管理,开发计划难以贯彻,时间一拖再拖,成本不断增加并难以估算和控制。软件质量得不到保证。计算机软件开发很少涉及计算机系统工程的方法。然而,人们也必须承认,这个时期依靠个人或几个人的聪明才智和程序设计技巧仍然开发了一批非常好的软件,例如,航空公司的飞机订票系统,用于国防的实时应用系统,等等 在系统软件方面也开发出了一批好的程序设计语言编译系统、操作系统和标准子程序库等。初期开发的大多数计算机系统采用批处理技术,提高了计算机的使用效率,其缺点是不利于程序的设计、调试和修改。今天,这种工作方式已很少采用。

第二阶段是20世纪60年代中期至20世纪70年代末期,引进了多用户、多道程序和人机交互等新概念。实时系统可以从多路信号源上采集、处理、分析和转换数据,并在相当短的时间内输出计算结果。用先进的在线(on - line)存储技术开发出第一代数据库管理系统。这一时期出现的"软件车间"生产并推销他们的软件产品。软件规模可达数万行代码,一个软件产品可卖给成百上千个顾客,从而改变了软件由用户自己开发、自己使用的被动局面。由于软件是逻辑产品,每个软件都有自己的个性,准确理解软件内部结构是一件十分困难的工作,因此,软件维护问题的矛盾加剧了。20世纪60年代末,"软件危机"变得十分严重。

第三阶段是20世纪70年代中期至20世纪80年代末期。在这一阶段,分布式系统、计算机网络、嵌入式计算机系统有了很大发展。微处理器、个人计算机、高性能的桌面工作站具有相当高的性能/价格比,广泛应用于人们工作、生活的各个领域。以前在大型机、中小型机上才能工作的软件逐步被移植到个人计算机和工作站上。个人计算机和工作站的发展推动了各种类型软件公司的发展。这时的软件产品可以拷贝几万份甚至几十万份向公众出售。社会上各行各业甚至家庭为购买软件而投资。软件产品销售额不断增加,软件产品的品种和质量有了很大的提高。

第四阶段是从20世纪80年代末期开始的。在很多应用领域,人们开始采用面向对象的技术,专家系统、人工智能软件开始走向实际应用。人工神经网络软件开始用于模式识别,多媒体技术将计算机的图形、图像、文字、声音处理集成在一起,提高了计算机的信息处理能力,扩大了计算机的应用领域,为人们使用计算机提供了方便。由于元器件极限速度的限制,高性能计算机的体系结构发生了很大变化。人们开始用成百上千个微处理器芯片互连成多种形式的多处理机系统。它们有巨大的处理能力,向传统的单指令流单数据流(SISD)和单指令流多数据流(SIMD)计算机系统提出了强有力的挑战。多指令流多数据流(MIMD)计算机系统并行算法研究、并行软件开发、传统的应用软件如何有效地移植到并行处理机系统上,已成为当前计算机软件领域的重要课题。20世纪90年代以来,由于Internet的高速发展和社会对分布计算的需求不断增长,Internet环境下的软件开发技术、分布计算环境下的软件互操作技术引起了广泛的研究兴趣。目前,人们已在这些热点领域取得了一批重要成果,例如Java语言与CORBA(通用对象请求代理结构)标准。

1.1.5　软件危机

1. 危机的表现

20世纪60年代末至20世纪70年代初,"软件危机"一词在计算机界广为流传。事实上,"软件危机"几乎从计算机诞生的那一天起就出现了,只不过到了1968年在原西德加米施(Garmish)召开的国际软件工程会议上才被人们普遍认识到。当时,训练有素的程序员大