21世纪

计算机应用技术系列规划教材

软件工程实用教程

◎ 郭宁 杨一平 编著 ◎

人民邮电出版社
POSTS & TELECOM PRESS

21世纪计算机应用技术系列规划教材

软件工程实用教程

郭　宁　杨一平　编著

人民邮电出版社

图书在版编目（CIP）数据

软件工程实用教程 / 郭宁，杨一平编著. —北京：人民邮电出版社，2006.3
（21 世纪计算机应用技术系列规划教材）
ISBN 7-115-14534-2

Ⅰ. 软… Ⅱ. ①郭…②杨… Ⅲ. 软件工程－教材 Ⅳ. TP311.5

中国版本图书馆 CIP 数据核字（2006）第 010005 号

内 容 提 要

本书根据软件工程的最新发展，结合目前软件工程教学的需要，以传统的软件工程和面向对象的软件工程为主线，遵循软件开发“工程化”思想，结合大量的应用案例，系统地介绍软件工程学的理论、方法以及应用技术。内容包括：软件开发模型、需求分析、软件设计、软件测试、软件维护、质量管理、文档技术、软件项目管理、软件工程工具和环境等。

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，语言简练，通俗易懂，采用案例教学方法，注重培养实际开发能力和文档的写作能力，具有很强的实用性和可操作性。书中含有丰富的例题与习题，便于教学和自学。

本书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。

21 世纪计算机应用技术系列规划教材

软件工程实用教程

◆ 编　著　郭　宁　杨一平
责任编辑　邹文波

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
人民邮电出版社河北印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：18.75
字数：449 千字　2006 年 3 月第 1 版
印数：1－3 000 册　2006 年 3 月河北第 1 次印刷

ISBN 7-115-14534-2/TP・5255

定价：26.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223

前　言

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。软件工程学将计算机科学理论与现代工程方法论相结合，着重研究软件开发理论、软件设计方法、工程开发技术和工具，是指导软件生产和管理的一门新兴的、综合性的应用科学。随着计算机科学和软件产业的迅猛发展，作为一个异常活跃的研究领域，软件工程学正在不断涌现新方法、新技术，蓬蓬勃勃地发展着。

软件工程学作为一门软件开发的工程方法学，它在软件开发中的指导意义与基础地位已经越来越多地得到整个 IT 业界的高度重视。由于软件工程在软件开发实践中发挥的重要作用，也使它成为高等院校计算机教学中一门核心课程。软件工程已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，目前，正在向基于软件复用和软件构件的第三代软件工程发展。为了适应信息技术迅速发展和教学的需要，我们特意编写了本书。

软件工程学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，快速原型技术和集成化软件开发环境等。本书不仅覆盖了软件工程的主要内容，而且强调了在软件这个特殊领域，理论与实践相结合的重要性。本书的另一个重要特点是实用性强，全书内容的选材强调实用价值和可操作性。采用案例形式来描述软件开发的全过程，案例贯穿全书始终，通过案例说明，帮助读者消化和理解所学内容，迅速提高实践能力。

全书共分 13 章。第 1～2 章为软件工程导论，重点介绍了软件工程的一些重要概念、软件工程学研究范畴、软件开发过程和软件过程模型等。第 3～6 章以传统的软件工程和面向对象的软件工程为主线，系统介绍软件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；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，人机界面设计技术以及应用实例剖析。第 7～9 章主要介绍软件测试、质量管理、软件维护与配置管理。第 10～12 章重点介绍了软件工程标准与文档、项目管理和集成化 CASE 环境以及工具等内容。第 13 章为本书的最后一部分，针对本学科实践性较强的特点，专门撰写了课程设计的目的、要求、实验指导与实例研究等内容，以利于读者全面、深入地掌握软件工程学的知识与方法。

本书以系统性、科学性和实用性为原则，并以结构严谨、布局合理、概念清新、内容适宜，能够更好地适应软件工程课程教学的需要为创作目标。本书编写有特色，应用指导性强，每章开头有章节导言、本章学习重点与难点，结束有小结和练习题。对开发过程中的重点方法和技术给出系列化应用示例，便于理解和学习。在教学计划中，如果安排 60 学时，建议采用第 1～12 章的自然顺序讲授。实习以 20～30 学时为宜。

本书适用面广，可以作为高校计算机专业高年级本科生和低年级硕士研究生的教材，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。

本书也是对作者多年来进行软件工程实践的一次经验总结。教材中诸多软件问题或应用实例，有许多就取材于作者的软件开发实践，并都按照教学的需要进行了模型简化。显然，这些源于实践的工程问题，对提高软件工程教学的实践性与实用性，将具有很好的示范效应。

首都经济贸易大学信息学院的杨一平教授组织了本书的编写工作，并在成书的过程中给予了多方面的指导，提出了很好的意见和建议。信息学院的赵玮、侯晶晶参与了本书的编写工作。此外，田建勇工程师对本书的编写工作给予了有力的支持。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。

在本书的编写过程中，我们参阅了大量的资料，在此对所有编著者表示衷心的感谢。由于编者时间仓促，水平有限，书中难免存在不足之处，敬请读者批评指正。

本书为教师配有电子教案，希望利用此电子教案，可以减轻教师负担，提高教学质量。

编　者

2006 年 1 月

目 录

第 1 章
软件工程引论

21 世纪是高度依赖计算机信息系统的时代，面对大量的计算机应用需求，怎样才能更有效地开发出各种不同类型的软件，是软件开发技术与软件工程所要解决的问题。软件工程是应用计算机科学、数学及现代工程的原则、方法来创建软件的学科，它对指导软件开发、质量控制以及开发过程的管理起着重要的作用。本章将概括地介绍软件的基本概念与特点，软件工程学科的诞生背景与形成，软件工程学研究的内容与对象，以及软件工程的方法等，使读者对软件工程与软件开发技术有所认识。

本章学习目标：

1. 掌握软件的定义与特点
2. 了解什么是软件危机以及软件危机产生的原因
3. 掌握软件工程的定义、目标和原则
4. 了解软件工程的研究内容与对象
5. 掌握软件工程的基本原则
6. 明确学习软件工程的意义

1.1 软件及软件危机

随着计算机技术的发展和应用领域的扩大，软件规模也越来越大，复杂程度不断增加，使得软件生产的质量、周期、成本难以预测和控制，出现了软件危机。软件工程正是为了解决软件危机而提出的，其目的是改善软件生产的质量，提高软件的生产效率。经过几十年的实践与探索，软件工程正在逐步发展成为一门成熟的专业学科，在软件产业的发展中起到重要的技术保障和促进作用。

1.1.1 软件及其特性

软件是计算机系统的思维中枢，是软件产业的核心。作为信息技术的灵魂——计算机软件，在现代社会中起着极其重要的作用。

1. 软件

计算机软件是由计算机程序的发展而形成的一个概念。它是与计算机系统操作有关的程

序、规程、规则及其文档和数据的统称。软件由两部分组成：一是机器可执行的程序和有关的数据；二是与软件开发、运行、维护、使用和培训有关的文档。

程序是按事先设计的功能和性能要求执行的语句序列。数据是程序所处理信息的数据结构。文档则是与程序开发、维护和使用相关的各种图文资料，例如，各种规格说明书、设计说明书、用户手册等。在文档中记录着软件开发的活动和阶段成果。

2. 软件的特点

软件是一种逻辑产品而不是实物产品，软件功能的发挥依赖于硬件和软件的运行环境，没有计算机相关硬件的支持，软件毫无实用价值。若要对软件有一个全面而正确的理解，我们应从软件的本质、软件的生产等方面剖析软件的特征。

（1）软件固有的特性

- 复杂性。软件是一个庞大的逻辑系统。一方面在软件中要客观地体现人类社会的事务，反映业务流程的自然规律，另一方面在软件中还要集成多种多样的功能，以满足用户在激烈的竞争中对大量信息及时处理、传输、存储等方面的需求，这就使得软件变得十分复杂。
- 抽象性。软件是人们经过大脑思维后加工出来的产品，一般寄生在内存、磁盘、光盘等载体上，我们无法观察到它的具体形态，这就导致了软件开发不仅工作量难以估计，进度难以控制，而且质量也难以把握。
- 依赖性。软件必须和运行软件的机器（硬件）保持一致，软件的开发和运行往往受到计算机硬件的限制，对计算机系统有着不同程度的依赖性。软件与计算机硬件的这种密切相关性与依赖性，是一般产品所没有的特性。为了减少这种依赖性，在软件开发中提出了软件的可移植性问题。
- 软件使用特性。软件的价值在于应用。软件产品不会因多次反复使用而磨损老化，一个久经考验的优质软件，可以长期使用。由于用户在选择新机型时，通常提出兼容性要求，所以一个成熟的软件可以在不同型号的计算机上运行。

（2）软件生产特性

- 软件开发特性。由于软件固有的特性，使得软件的开发不仅具有技术复杂性，还有管理复杂性。技术复杂性体现在软件提供的功能比一般硬件产品提供的功能多，而且功能的实现具有多样性，需要在各种实现中做出选择，更有实现算法上的优化带来的不同，而实现上的差异会带来使用上的差别。管理上的复杂性表现在：第一，软件产品的能见度低（包括如何使用文档表示的概念能见度），要看到软件开发进度比看到有形产品的进度困难得多；第二，软件结构的合理性差，结构不合理使软件管理复杂性随软件规模增大而呈指数增长。因此，领导一个庞大人员的项目组织进行规模化生产并非易事，软件开发比硬件开发更依赖于开发人员的团队精神、智力和对开发人员的组织与管理。
- 软件产品形式的特性。软件产品的设计成本高昂而生产成本极低。硬件产品试制成功之后，批量生产需要建设生产线，投入大量的人力、物力和资金，生产过程中还要对产品进行质量控制，对每件产品进行严格的检验。然而，软件是把人的知识与技术转化为信息的逻辑产品，开发成功之后，只需对原版软件进行复制即可，大量人力、物力、资金的投入和质量控制、软件产品检验都是在软件开发中进行的。由于软件的复制非常容易，软件的知识产权保护就显得极为重要。
- 软件维护特性。软件在运行过程中的维护工作比硬件复杂得多。首先，软件投入运行

后，总会存在缺陷甚至暴露出潜伏的错误，需要进行“纠错性维护”。其次，用户可能要求完善软件性能，对软件产品进行修改，进行“完善性维护”。当支撑软件产品运行的硬件或软件环境改变时，也需要对软件产品进行修改，进行“适应性维护”。软件的缺陷或错误属于逻辑性的，因此不需要更换某种备件，而是修改程序，纠正逻辑缺陷，改正错误，提高性能，增加适应性。当软件产品规模庞大、内部的逻辑关系复杂时，经常会发生纠正一个错误而产生新的错误的情况，因此，软件产品的维护要比硬件产品的维护工作量大而且复杂。

1.1.2　软件危机

20 世纪 60～70 年代，由于软件规模的扩大、功能的增强和复杂性的增加，使得在一定时间内仅依靠少数人开发一个软件变得越来越困难。在软件开发中经常会出现时间延迟、预算超支、质量得不到保证、移植性差等问题，甚至有的项目在耗费了大量人力、财力后，由于离目标相差甚远而宣布失败。这种情况使人们认识到“软件危机”的存在。

1. 软件危机的突出表现

（1）软件生产率低。软件生产率提高的速度远远跟不上计算机应用迅速普及和深入的趋势。落后的生产方式与开发人员的匮乏，使得软件产品的供需差距不断扩大。由于缺乏系统有效的方法，现有的开发知识、经验和相关数据难以积累与复用，另外，低水平的重复开发过程浪费了大量的人力、物力、财力和时间。人们为不能充分发挥计算机硬件提供的巨大潜力而苦恼。

（2）软件产品常常与用户要求不一致。开发人员与用户之间的信息交流往往不充分，经常存在二义性、遗漏，甚至是错误。由于开发人员对用户需求的理解与用户的本意有所差异，以致造成开发中后期需求与现实间的矛盾的集中暴露。

（3）软件规模的增长，带来了复杂度的增加。由于缺乏有效的软件开发方法和工具的支持，过分依靠程序设计人员在软件开发过程中的技巧和创造性，所以，软件的可靠性往往随着软件规模的增长而下降，质量保障越来越困难。

（4）不可维护性突出。软件的局限性和欠灵活性，不仅使错误非常难改正，而且不能适应新的硬件环境，也很难根据需要增加一些新的功能，整个软件维护过程除了程序之外，没有适当的文档资料可供参考。

（5）软件文档不完整、不一致。软件文档是计算机软件的重要组成部分，在开发过程中，管理人员需要使用这些文档资料来管理软件项目；技术人员则需要利用文档资料进行信息交流；用户也需要通过文档来认识软件，对软件进行验收。但是，由于软件项目管理工作的不规范，软件文档往往不完整、不一致，这给软件的开发、交流、管理、维护等都带来了困难。

2. 产生软件危机的原因

软件危机是指计算机软件的开发和维护过程中所遇到的一系列严重问题。产生软件危机的原因主要有以下几点。

（1）软件独有的特点给开发和维护带来困难。由于软件的抽象性、复杂性与不可预见性，使得软件在运行之前，开发过程的进展情况较难衡量，软件的错误发现较晚，软件的质量也较难评价，因此，管理和控制软件开发过程相当困难。此外，软件错误具有隐蔽性，往往在很长时间里软件仍可能需要改错。这在客观上使得软件维护较为困难。

（2）软件人员的错误认识。相当多的软件专业人员对软件开发和维护还有不少的错误观念。例如，软件开发就是编写程序，忽视软件需求分析的重要性，轻视文档的作用，轻视软件维护等。这些错误认识加重了软件危机的影响。

（3）软件开发工具自动化程度低。尽管软件开发工具比 30 年前已经有了很大的进步，但直到今天，软件开发仍然离不开工程人员的个人创造与手工操作，软件生产仍不可能向硬件设备的生产那样，达到高度自动化。这样不仅浪费了大量的财力、物力和宝贵的人力资源，无法避免低水平的重复性劳动，而且软件的质量也难以保证。此外，软件生产工程化管理程度低，致使软件项目管理混乱，难以保障软件项目成本、开发进度按计划执行。

1.2 软件工程

为了克服“软件危机”，1968 年在北大西洋公约组织（NATO）召开的计算机科学会议上，Fritz Bauer 首先提出“软件工程”的概念，试图用工程的方法和管理手段，将软件开发纳入工程化的轨道，以便开发出成本低、功能强、可靠性高的软件产品。几十年来，人们一直在努力探索克服软件危机的途径。

1.2.1 软件工程的形成与发展

自 1968 年 NATO 会议上提出软件工程这一概念以来，人们一直在寻求更先进的软件开发的方法与技术。当出现一种先进的方法与技术时，就会使软件危机得到一定程度的缓解。然而，这种进步又促使人们把更多、更复杂的问题交给计算机去解决，于是又需要探索更先进的方法与技术。几十年来，软件工程研究的范围和内容也随着软件技术的发展不断变化和拓展。软件工程的发展经历了以下 3 个阶段。

第一阶段：20 世纪 70 年代，为了解决软件项目失败率高、错误率高以及软件维护任务重等问题，人们提出软件生产工程化的思想，希望使软件生产走上正规化的道路，并努力克服软件危机。人们发现将传统工程学的原理、技术和方法应用于软件开发，可以起到使软件生产规范化的作用。它有利于组织软件生产，提高开发质量，降低成本和控制进度。随后，人们又提出软件生命周期的概念，将软件开发过程划分为不同阶段（需求分析、概要与详细设计、编程、测试和维护等），以适应更加复杂的应用。人们还将计算机科学和数学用于构造模型与算法上，围绕软件项目开展了有关开发模型、方法以及支持工具的研究，并提出了多种开发模型、方法与多种软件开发工具（编辑、编译、跟踪、排错、源程序分析、反汇编、反编译等），并围绕项目管理提出了费用估算、文档评审等一些管理方法和工具，基本形成了软件工程的概念、框架、方法和手段，成为软件工程的第一代——传统软件工程时代。

第二阶段：20 世纪 80 年代，面向对象的方法与技术受到了广泛的重视，Smalltalk-80 的出现标志着面向对象的程序设计进入了实用和成熟阶段。20 世纪 80 年代末逐步发展起来的面向对象的分析与设计方法，形成了完整的面向对象技术体系，使系统的生命周期更长，适应更大规模、更广泛的应用。这时，进一步提高软件生产率、保证软件质量就成为软件工程追求的更高目标。软件生产开始进入以过程为中心的第二阶段。这个时期人们认识到，应从