软件工程系列教材

Software Engineering:
Best Practice

软件工程
应用实践教程

吴洁明　袁山龙　编著

清华大学出版社

软件工程应用实践教程

吴洁明　袁山龙　编著

清华大学出版社
北　京

内 容 简 介

本书全面系统地阐述软件工程所涉及到的各种概念、方法和新技术，重点突出了软件工程在实践环节中的应用。

书中介绍了面向过程软件工程方法在中小项目中的应用，重点强化软件工程开发中面向对象的技术，特别是面向对象的需求获取、系统分析和设计以及实现，并针对每个环节给出了具体的活动过程和产品规范。书中提供大量的来自实际项目开发过程中的经验性内容，非常实用；并讲述了用户方在软件项目中的权利和义务，如何监督项目的正常实施，如何保护用户的利益，这些内容对软件项目的用户方自我保护很有借鉴价值。

本书既注重知识的系统性，同时注重软件工程的实践性和选材的先进性，可作为高等院校“软件工程”本科或研究生的教材或教学参考书，也可供软件开发人员、项目管理人员和软件项目客户阅读参考。

图书在版编目（CIP）数据

软件工程应用实践教程/吴洁明，袁山龙编著.—北京：清华大学出版社，2003
ISBN 7-302-06672-8

I. 软… II.①吴…②袁… 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（2003）第 039549 号

出版者：清华大学出版社（北京清华大学学研大厦，邮编 100084）
http://www.tup.com.cn
印刷者：北京市耀华印刷有限公司
发行者：新华书店总店北京发行所
开 本：正 16 开 印张：22 字数：535 千字
版 次：2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷
书 号：ISBN 7-302-06672-8/TP・4994
印 数：0001~5000
定 价：29.00 元

前　言

回顾 20 世纪的技术进展，大家一致认为信息技术是发展最快的技术之一，特别是信息技术应用的渗透性，几乎在各个领域中都可以看到它的身影。软件作为信息技术的灵魂，更是扮演了极其重要的角色。软件产业在全球经济中所占的地位越来越重要，为了加速软件的工程化生产，自 20 世纪中期以来，大家对软件工程学的研究越来越深入和广泛，新的学科、新的技术、方法和工具不断涌现。

本书从介绍软件工程的基本原理、概念、方法和技术开始，重点讲述了软件工程在实践环节的应用。由于目前软件开发已经普遍使用面向对象的技术，因此，本书加强了这方面的内容，特别是面向对象的需求获取、面向对象的系统分析和设计、面向对象的实现，每个环节都给出了具体的活动和产品。

由于面向过程的软件工程属于比较经典的内容，其中的结构化分析和设计方法至今仍然适用于一些中小型项目的开发，因此，本书对结构化方法也做了比较详细的讲述。

全书共分为 13 章，各章内容简介如下：

- 第 1 章是软件工程的概述，主要介绍了软件工程的发展和一些相关的概念，使读者对软件工程有一个总体的了解。
- 第 2 章详细讲述了软件工程中最重要的环节“需求工程”。本章从需求获取、需求分析、需求文档的编写、需求审查到需求管理，进行了全面的论述。详细介绍了每个环节的方法、步骤。并且给出了一系列实用的模板，读者可以根据模板进行需求调研。
- 第 3 章讲述了结构化的软件设计方法。详细讲述了从需求分析的数据流程图到软件结构图的转化，给出了软件设计的具体原则和影响设计的因素，并且分别讲述了软件结构设计、数据设计、接口设计和过程设计的方法和步骤。
- 第 4 章讲述了结构化编程的若干实用条例和规则，目的是指导读者编写出规范的程序代码，养成良好的编程习惯。
- 第 5 章讲述了结构化程序的测试方法和测试策略，介绍了一些辅助测试工具。
- 第 6 章介绍软件维护的概念以及提高软件可维护性的方法。
- 第 7 章介绍了面向对象技术，对几种流行的面向对象软件工程方法进行了比较，最后详细介绍了统一建模语言。
- 第 8 章是面向对象的需求获取方法介绍，详细讲述了基于用例图、活动图和状态图进行需求获取的过程和步骤。
- 第 9 章是面向对象的系统分析。首先介绍了面向对象系统分析所涉及的对象交互

图、类图和包图，然后给出了面向对象系统分析过程的具体步骤，以及在本阶段应该提交的产品。

- 第 10 章是面向对象的设计，软件系统的质量主要取决于软件设计，因此，本章首先介绍了面向对象的设计原则，以及设计阶段所涉及的配置图和中间件等内容，然后给出了面向对象设计的具体过程、面向对象设计阶段应提交的产品。
- 第 11 章介绍了软件复用技术。软件工程的一个重要目标就是要实现软件的复用，而达到软件复用需要规范的框架和一致的接口规范，本章除了讲述软件复用的概念、方法之外，还给出了一个实例研究，详细地介绍软件复用技术的应用。
- 第 12 章重点在于面向对象编程实现，本章介绍了面向对象编程语言的特点和选择编程语言的几个参考因素。介绍了面向对象程序设计的风格。本章重点讲述了实现阶段的分工和工作步骤。
- 第 13 章讲述面向对象的测试特点和策略，重点介绍面向对象测试过程的步骤和测试阶段应该提交的产品。

书中第 1~10、12、13 章由吴洁明编写，第 11 章由袁山龙编写。本书在编写过程中得到了许多老师和研究生的帮助，在此对他们表示衷心的感谢。

本书内容新颖、实用，讲述通俗易懂，所给出的各类模型非常实用，读者阅读本书后便可以模仿，进行实际操作。但是，由于水平有限，时间紧张，书中错误在所难免，敬请各位读者批评指正。同时也欢迎读者与我们一起探讨软件工程相关的技术和方法，我们的 E-mail 地址是：wjm@ncut.edu.cn 和 yuanshanlong@163.com 。

目　录

第1章　软件工程概述

自从有了软件以来，软件人员就希望有一天软件的生产能够像工厂的产品生产一样，实现规范化的设计、流水化的生产和标准化的检验过程。经过几十年的摸索研究，今天的软件生产确实在这方面取得了可喜的成就，软件工程化生产已经初露萌芽。

本章对软件的特点、软件的发展、软件危机的出现和软件工程学科的形成、软件工程过程等问题和基本概念进行详细的叙述。

1.1　软件和软件危机

20世纪80年代初，软件还只是在高等院校和科研机构中被认识，那时提到软件时多指源程序。

1983年IEEE为软件下的定义是：计算机程序、方法、规则和相关的文档资料以及在计算机上运行时所必需的数据。目前对软件比较公认的解释为：软件是程序、支持程序运行的数据以及与程序有关的文档资料的完整集合。其中文档是与程序开发、维护和使用有关的图文资料。软件的特点如下：

- 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上、内存、磁盘、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，才能了解它的功能和性能。
- 因为软件是一种逻辑实体，所以软件在使用过程中，没有磨损、老化的问题。软件在使用过程中不会因为磨损而老化，但为了适应硬件、系统环境以及需求的变化可能要不断修改，这些修改不可避免地会引入错误，导致软件失效率升高，从而使得软件可靠性下降。当修改的成本变得难以接受时，软件就被抛弃。
- 软件一旦研制开发成功，其生产过程就变成复制过程。不像其他工程产品那样有明显的生产制造过程。由于软件复制非常容易，因此就出现了软件产品的版权保护问题和打击盗版的问题。
- 软件对硬件和环境有着不同程度的依赖性，这导致了软件升级和移植的问题。随着计算机技术的发展，计算机硬件和支撑环境不断升级，为了适应运行环境的变化，软件也需要不断维护，并且维护成本通常比开发成本高许多。
- 软件生产至今尚未摆脱手工方式。随着软件开发环境的改善，市场上出现了一些辅助开发工具，可以辅助生成代码框架、生成开发文档等等。但是最终的核心代码仍然要程序员手工编写和组织。并且，随着科学技术的进步，人们对计算机的依赖程度越来越高，因此对软件的需求数量和规模也更大，导致软件开发人员的工作压力

也越来越大。

- 软件开发的手工行为产生了一个致命的问题，就是为应用“量身订做”软件。其他工程领域有产品标准，所有生产厂家按照标准生产产品，客户买来就可使用。例如，不管哪个厂家生产的灯泡，只要瓦数、电压、电流、接口几个指标符合要求，用户买来装上就可以使用。而长期以来，软件给人的感觉是修改几条指令很简单，因此，客户总是强调软件要适应自己的业务需求。因此，软件产品大多是为客户“定做”的，通用性差。近几年来，组件的研究开发取得了重大的进展，单一的软件程序逐渐演变为组件式程序。这样，遇到不同的应用需求时，就可以考虑应用已有的组件搭建一个大规模的系统。当组件不能满足要求时，可以购买其他厂家生产的组件或自己开发部分组件。采用这种方法解决了所有软件全部从头开发的情况。
- 软件涉及人类社会的各行各业，常常涉及其他领域的专门知识，这对软件工程师提出了很高的要求。软件开发实际上应该有比较明确的分工，例如，业务（咨询）专家负责规范、描述用户的业务流程；系统分析员负责将用户需求转换为系统功能和性能需求；系统设计人员负责规划系统框架、构建系统模型，设计系统蓝图；软件工程师负责实施设计方案；测试工程师负责软件测试，发现软件缺陷；质量工程师负责制定软件质量标准、监督软件开发过程、评估软件质量等等。实际上软件开发过程中不同的工作应该有不同的人员专门负责实施，然而，实际工作中，我们却经常看到一个项目中几个软件程序员从头到尾，既做系统分析又做系统设计，最后还要编写代码。
- 软件不仅是一种在市场上推销的工业产品，它往往又是与文学艺术作品相似的精神作品，与体力劳动相比，精神活动过程的特点是“不可见性”，这大大增加了组织管理上的困难。

由于软件的这些特点，以及长期以来始终没有发明一种高效的开发方法，导致软件生产效率非常低，交付期一拖再拖，最终交付的软件产品在质量上很难保障。特别是软件对开发者的依赖性非常强，开发队伍中一旦走掉一个主力，有可能使整个产品都处于停止状态。这种现象早在20世纪60年代就被定义为“软件危机”，它的具体表现如下：

- 对软件开发成本的估计不准确，造成开发成本超出预算。
- 开发进度不能保证，交付时间一再拖延。
- “已完成”的软件不能满足用户的需求。
- 软件产品的质量没有保证，运算结果出错、操作死机等现象屡屡出现。
- 软件通常没有适当的文档资料或文档与最终交付的软件产品不符，软件的可维护程度非常低。
- 软件开发生产率的提高赶不上硬件的发展和人们需求的增长。

软件的特点是导致软件危机的客观因素，而软件开发和维护过程中使用的不正确方法是主观因素。其根本原因主要是软件开发过程的不成熟，主要表现为忽视软件开发前期的调研和分析工作；没有统一的、规范的方法论指导，文档资料不齐全，忽视人与人的交流；忽视测试阶段的工作；轻视软件的维护。

1.2　软件的分类

软件的分类有许多方法，不同的人员由于不同的目的可能有不同的划分原则。通常的划分方法是按软件的功能划分和按软件规模划分。此外，软件还可以按重要性划分为关键软件和非关键性软件。按处理方式可划分为实时软件、交互软件、批处理软件。按销售市场可划分为项目软件、产品软件。按使用频率可划分为高使用频率和低使用频率软件等。不同类型的软件的开发要求不同，所遵循的开发标准也不同。例如，关键软件是指关系到人的生命安全、国家财产、国家机密的软件，这类软件必须具有高可靠性。

1.2.1　按软件功能划分

按照软件的功能可以将软件划分为系统软件、支撑软件、应用软件。

系统软件通常是与计算机硬件密切相关的那些比较底层的支持软件。这些软件的规模通常比较大，并且与硬件的结构和性能密切相关。例如，操作系统、设备驱动软件、网络通信软件等。它们的作用是保障计算机各个部件能够正常运行，使相关的软件和数据协调、高效地工作。这部分软件在任何应用中都是必不可少的，也是首先要确定的软件。只有确定了系统软件的类型和版本后，才能够考虑支撑软件和应用软件。

支撑软件是支持软件开发和运行的工具性软件。其中包括数据库管理系统、软件开发环境、软件辅助设计工具、软件辅助测试工具、中间件、程序库等。这类软件非常多，分类也更加细致。

应用软件是为特定应用目的而开发、提供某些特定服务的软件。应用软件可谓是规模各异，种类繁多。不同的领域有不同的应用软件，有大规模的应用软件，例如字处理软件、计算机辅助设计与制造软件、军事指挥系统、导弹防御系统，也有微型软件，例如，只有几条指令的微型控制软件。

1.2.2　按软件规模划分

根据软件开发所投入的人力和时间等资源，以及软件交付的文档和源程序的数量。软件可划分为微型软件、小型软件、中型软件、大型软件、超大型软件和巨型软件。具体划分标准见表1-1。

表1-1　软件的规模

软件规模	投入人数	开发时间	源程序行数	特征
微型	1	1~4周	500	不必有严格的设计和测试文档
小型	1~2	1~6月	2000	通常没有与其他程序的接口
中型	3~5	1~2年	0.5~5万	需要有严格的文档和设计规范
大型	5~20	2~3年	5~10万	需要按照软件工程方法进行管理
超大型	100~1000	4~5年	100万左右	必须按照软件工程开发，有严格的质量管理措施
巨型	2000~5000	5~10年	1000万左右	同上

通常规模大、投入人员多、开发周期长的软件，其开发工作必须严格按照软件工程方法进行。而规模小的软件由于投入有限，不可能完全按照软件工程方法实施，需要开发人员根据具体情况调整软件工程的部分条款。

1.3 软件工程概念

在上一节中，我们已提到了软件工程，那么到底什么是软件工程呢？软件工程是一门旨在生产无故障的、及时交付的、在预算之内的和满足用户需求的软件的学科。实质上，软件工程就是采用工程的概念、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方法和最先进的软件开发技术结合起来，应用到软件开发和维护过程中，来解决软件危机问题。1993年，IEEE为软件工程下的定义是：

> “软件工程是将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护过程，即将工程化应用于软件中的方法的研究。”

软件工程研究所依据的基础理论极为丰富，主要有数学、计算机科学、经济学、工程学、管理学和心理学等其他学科。

软件工程是一种层次化技术，它的最底层是质量保证层，中间是过程层和方法层，最上层是工具层。如图1-1所示。

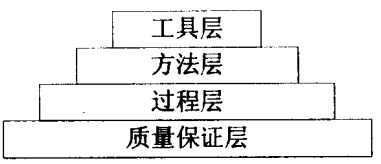

图1-1 软件工程层次图

软件工程的最底层是质量保证层，全面的质量管理和质量需求是推动软件过程不断改进的动力，正是这种改进的动力导致了更加成熟的软件工程方法不断涌现。

过程层与技术层结合在一起，它定义了一组关键过程域框架，目的是保证软件工程技术被有效地应用，使得软件能够被及时地、高质量和合理地开发出来。

方法层提供了软件开发的各种方法。包括如何进行软件需求分析和设计、如何实现设计、如何测试和维护等方法。

工具层为软件工程方法和过程提供了自动或半自动的支撑环境。目前市场上已经有许多不错的软件工程工具，例如，Rational公司最近推出的一系列软件工程工具广为业界赞誉，应用效果也很不错。使用软件工程工具可以有效地改善软件开发过程，提高软件开发的效率，降低开发成本。

软件工程强调规范化和文档化，规范化的目的是使众多的开发者遵守相同的规范，使软件生产摆脱个人生产方式，进入标准化、工程化的生产方式。文档化是将软件的设计思想、设计过程和实现过程完整地记录下来，以便于后人的使用和维护。在开发过程中各类相关的人员借助文档进行交流和沟通。另外，在开发过程中产生的各类文档使得软件的生