汇集作者多年的教学研究与工程项目开发经验

软件工程

基础实践教程

吴洁明 编著

- 从软件的特点入手介绍软件工程基础知识；
- 介绍软件生命周期各个阶段软件的存在形式、评价标准以及软件文档的写作、管理；
- 通过案例实践，讲述可行性研究，结构化分析和设计的方法、工具和步骤；
- 以相同案例结合UML讲述面向对象分析和设计的方法、工具和步骤；
- 详述软件测试概念、方法、策略和步骤。

清华大学出版社

软件工程基础实践教程

吴洁明 编著

清华大学出版社
北 京

内 容 简 介

本书是作者多年的教学研究与工程项目开发经验编写的软件工程实践教程。

全书以5条主线介绍软件工程的原理、方法和过程。第1条，从软件的特点引出软件危机的概念，介绍软件危机的解决方法，即本书的核心内容软件工程；第2条从软件的生存周期入手，介绍软件生命周期各个阶段软件的存在形式和评价标准，详细介绍软件文档的写作及管理；第3条从软件工程实践出发，介绍可行性研究结构化分析和设计方法，以及相应的工具和步骤；第4条从面向对象的概念入手，结合UML讲述面向对象分析和设计的方法、工具和步骤；第5条详细讲述软件测试的概念、方法、策略和步骤。

本书以实例驱动理论讲解。在结构化方法和面向对象方法的讲述中使用相同的示例，目的是让读者从中体会两种方法各自的特点，特别是面向对象方法的优点。

本书适用于普通高等院校“软件工程”课程的教材或参考书。书中给出了大量的实用模板和表格，可供软件工程师在实际项目开发中参考应用。

图书在版编目（CIP）数据

软件工程基础实践教程/吴洁明编著. —北京：清华大学出版社，2007.11
ISBN 978-7-302-16342-8
Ⅰ.软…　Ⅱ.吴…　Ⅲ.软件工程-教材　Ⅳ.TP311.5
中国版本图书馆CIP数据核字（2007）第163524号

责任编辑：夏非彼　杨　洋
责任校对：贾淑媛
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　**邮　　编**：100084
c—service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：清华大学印刷厂
装 订 者：三河市金元印装有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：24.75　**字　数**：605千字
版　　次：2007年11月第1版　　**印　　次**：2007年11月第1次印刷
印　　数：1～4000
定　　价：35.00元

序

从提出软件工程的概念至今已经过去了四十多年，这期间计算机技术发生了巨大的变化。人们经过软件危机之后已经充分认识到软件工程的重要性，并且在不断地探索软件工程的新方法、新工具和有效的过程。

软件工程的本质是研究如何用工程化的方法解决软件开发和管理过程中的问题。软件开发和管理中究竟存在哪些主要问题呢？我们首先想到的是软件质量难以保证，软件开发成本无法控制，软件生产不易量化，软件开发过程管理缺乏标准等。这些问题引起了人们对软件工程研究的兴趣，因此人们从方法、过程、工具和质量 4 个层次上展开了对软件工程的研究。

本书将沿着以下几条主线向读者介绍软件工程的原理、过程和方法。

第 1 条主线是从软件的特点入手，引出软件危机的概念，介绍软件危机的解决方法，即本书的核心内容——软件工程。

第 2 条主线是从软件的生存期入手，介绍软件生命周期各个阶段软件的存在形式和评价标准，详细介绍软件文档写作及管理。

第 3 条主线是从软件工程实践出发，引出可行性研究，结构化分析和设计的方法、工具与步骤，最后带领大家以一个图书馆信息管理系统为例进行演练。

第 4 条主线是从面向对象的概念入手，结合 UML 讲述面向对象分析和设计的方法、工具和步骤，最后也是以图书馆信息管理系统为例进行实际演练。

第 5 条主线是详细讲述软件测试的概念、方法、策略和步骤。

本书的写作方法是以实例驱动内容，结构化方法和面向对象方法使用相同的实例，目的是让读者从中体会两种方法各自的特点，特别是面向对象方法的优点。

全书分为 14 章，第 1 章“软件工程概述”，主要目的是解决为什么要研究软件工程，软件工程研究的主要内容和方法，它的发展过程，以及当前研究的前沿内容是什么。第 2 章“可行性研究”，介绍软件工程可行性研究的目的和方法，具体介绍软件成本估算模型，读者可参考这部分内容估算软件开发的工作量，因为软件工程研究的一个重要内容就是“量化”。第 3 章“需求工程”，详细讲述需求的获取方法和描述方法，目的是研究获得高质量需求的方法和过程。这部分给出了大量实用的需求调研表格和文档模板。第 4 章“结构化分析”，以图书馆信息管理系统为例讲述软件结构化分析方法的具体实现过程和相关的各种图形分析工具。第 5 章“结构化软件设计”，介绍软件设计的原则和设计概念。在结构化分析的基础上，以实例详细讲述结构化方法的概要设计和详细设计过程。第 6 章“软件测试”，介绍了软件测试的概念、测试策略和一些常用的测试方法。第 7 章“编写程序”，主要讲述编程规范和良好的编程风格。第 8 章“面向对象的需求分析”，介绍基于 UML 的面向对象分析方法，结合图书馆信息管理系统的实例，分步骤讲述面向对象分析方法的过程。第 9 章“面向对象设计”，介绍基于 UML 的面向对象设计方法，结合实例详细讲述了面向对象设计的几个重要原则，简单介绍了面向对象设计过程中常用的 XML 技术，结合前一章图

书馆信息管理系统的分析结果讲述面向对象的设计过程。第 10 章“面向对象的实现”，介绍面向对象程序设计的特点和风格，指导读者编写出可复用性、可扩展性更高的程序。第 11 章“面向对象的测试”，主要介绍面向对象软件测试的特点和测试策略，讲述了面向对象软件测试的具体过程。第 12 章“软件开发文档的写作和管理”，主要介绍软件文档编写过程中应该注意的一些问题，同时介绍一些国内外著名的软件工程标准和规范。第 13 章“系统维护”，介绍软件维护的类型，分析影响软件可维护性的主要因素，讲述了软件维护的过程和具体维护步骤。第 14 章“软件复用”，介绍软件复用的概念和方法，以实例讲述了软件复用的具体步骤和方法。

本书适合用作普通高等院校“软件工程”课程的教材和参考书，每章最后的练习对学习者掌握教材内容和扩充知识有很大帮助。

书中给出了大量的实用模板和表格，它们都是众多软件开发项目的结晶，读者可参考并把它们用于实际项目，也可以通过实际项目不断地改进，逐步提炼出一套适合自己的模板和表格。

本书的第 14 章由袁龙山老师编写。本书在成稿过程中得到许多老师和研究生的帮助，在此对他们表示衷心的感谢。特别要感谢本书的编辑周烈强老师，她认真阅读了本书，并对本书作了大量细致的文字工作，她的工作态度令人敬佩。

由于水平有限，书中一定存在许多问题和不足，希望读者提出宝贵的意见和建议，本人将不胜感激。联系方式 E-MAIL：wujieming@263.net，booksaga@126.com。

2007 年 8 月于北京

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产品能够像工厂生产其他产品一样，实现规范化设计、流水化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些方面取得了可喜的成就，软件工程已经成为软件产业健康发展的基础。软件工程化生产已经不再是人们的奢望。

本章讲述软件工程的基本概念，包括软件的特点、软件危机、软件工程的发展、软件工程过程等内容。

1.1 软件和软件危机

1983年，IEEE（国际电气与电子工程协会）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：

> 软件是计算机系统中与硬件相互依存的另一部分，它包括程序、相关数据及其说明文档。

其中程序是按照事先设计的功能和性能要求执行的指令序列；数据是程序运行过程中处理的对象；文档是与程序开发、维护和使用有关的各种图文资料。

1.1.1 软件的特点

要理解软件工程，首先要对软件的特点有所了解，软件的特点如下。

（1）软件是一种逻辑实体，具有抽象性。这个特点使它与对应的硬件产品有着明显的差异。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。

（2）由于软件是一种逻辑实体，所以软件在使用过程中没有磨损的问题。软件在使用过程中不会因为磨损而老化，但为了适应硬件和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。

（3）软件一旦研究开发成功，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。

（4）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。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

护的成本通常比开发成本高许多。

（5）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，对软件的需求数量和规模也更大，所以软件开发人员的工作压力也越来越大。

（6）软件开发的手工行为造就了一个致命的问题，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，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“订做”的，通用性差。近几年来，组件软件的研究开发取得了重大进展——将单一的软件程序改为组件式程序。这样，遇到不同的应用需求时，就可以考虑应用已有的软件搭建一个大规模系统。当组件不能满足要求时，可以购买其他厂家生产的软件组件或自己开发部分组件。采用这种方法基本解决了所有软件全部从头开发的情况。

（7）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功能需求和性能需求；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由不同的人员专门负责。然而，实际工作中却经常看到一个项目的几个程序员从头做到尾，既做系统分析又做系统设计，最后还要做编程和测试。

（8）软件不仅是一种在市场上推销的工业产品，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。

（9）在软件开发的不同时期进行修改需要付出的代价有很大不同。在早期引入变动，涉及的因素较少，因而代价也比较低；而在开发的中期，软件配置的许多成分已经完成，引入一个变动要对所有已完成的配置成分都作相应的修改，不仅工作量大，而且逻辑上也更加复杂，因此付出的代价剧增；在软件完成后引入变动，需要付出的代价更高。

1.1.2 软件危机

由于软件具有上述这些特点，长期以来一直没有发明一种高效的开发方法，从而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程度非常紧密，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滞状态。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。
- 开发进度不能保障，交付时间一再拖延。
- “已完成”的软件不满足用户的需求。

- 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。其根本原因是软件开发过程不成熟，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。

说明：成本、进度、质量是软件生产过程中最具核心竞争力的因素。

1.1.3 软件错误导致的失败实例

1962 年 6 月，美国飞向金星的第一个空间探测器水手 I 号，因其飞行舱的计算机导航程序中一条语句的逻辑错误，其结果完全不同于程序员所期待的，致使这个探测器偏离航线而失败。

美国阿波罗 8 号太空飞船的一个软件错误，造成了存储器部分信息丢失；而阿波罗 14 号在飞行的 10 天中，出现了 18 个软件错误。

这些项目投资巨大、设计精细，但这么重大的项目其软件存在错误却是事实。

1.2 软件工程发展简史

1968 年在德国格密斯（Garmish）举行的学术会议上，NATO（北大西洋公约组织）正式提出了“软件工程”这一术语。软件工程作为工程学科家族中的新成员，对软件产业的形成和发展起了决定性作用。它指导人们科学地开发软件，有效地管理软件项目，以保证软件开发能够在预算的范围内高质量地按时完成。

在 20 世纪 70 年代已经基本形成了软件工程的概念、框架、方法和手段，我们称之为第一代软件工程，即传统软件工程。

80 年代出现的 Smalltalk 80 标志着面向对象程序设计进入了实用阶段，从 80 年代中到 90 年代中，研究的重点转移到面向对象的分析和设计上来，从而演化成软件工程的第二代，称之为对象工程。

90 年代后期，软件工程的一个重要进展就是基于构件的开发方法。为了提高软件生产力，避免草率地开发应用程序，尽可能地利用可复用构件来组装成新的应用软件系统。到目前为止，构件技术的研究和发展形成了新一代软件工程，即第三代软件工程，也有不少人称之为构件工程。

软件工程至今还在不断发展，无论是构件工程、传统工程还是对象工程都是如此，即使是传统软件工程的一些基本概念、框架，也随着技术的进步在发生变化。总之，软件工程代与代之间并没有鸿沟，它们不仅交叉重叠，也携手并进。这种划分方法只是为了阐述软件工程的迅速发展，分别从不同角度来研究和实践而已，并没有严格的含义。

1.3 软件工程的定义和目标

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，运用到软件开发和维护过程中，来解决软件危机。

1983 年 IEEE 的软件工程术语汇编中，把软件工程定义为："对软件开发、运行、维护、消亡的系统研究方法"。1993 年新版的 IEEE 软件工程术语汇编重新定义了软件工程。

> "软件工程是：① 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，也就是将工程化应用于软件开发和管理之中；② 对①中所选方法的研究"。

从以上的定义中可以看出人们对软件工程的理解是逐步深入的。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。其中：数学和计算机科学用于构造模型、分析算法；经济学和工程学用于评估成本、制定规范和标准；管理学和心理学用于进度、资源、环境、质量、成本等的分析和管理。

软件工程研究的主要内容是软件开发技术和软件开发管理两个方面。在软件开发技术方面，主要研究软件开发方法、软件开发过程、软件开发工具和环境。在软件开发管理方面，主要研究软件管理学、软件经济学、软件心理学。

最近几年，人们发现软件工程具有层次化结构，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。见图 1-1 所示。

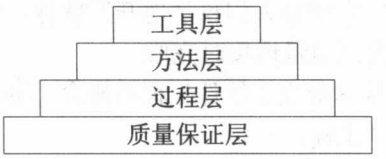

图 1-1 软件工程层次结构图

全面的质量管理和质量需求是推动软件工程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。过程层与方法层结合在一起，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、高质量地和合理地开发出来。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，应用效果良好。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，降低开发和管理成本。

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人员借助于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