北京市高等教育精品教材立项项目

软件工程实用教程

Software Engineering

陈 明 编著

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

北京市高等教育精品教材立项项目

软件工程实用教程

Software Engineering

陈　明　编著

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

内 容 简 介

本书介绍软件工程学及其应用，主要内容包括：软件工程概述、可行性分析、需求分析、概要设计、详细设计、编码、测试与维护，软件开发工具与环境，Power Designer，软件项目管理，软件配置管理，软件质量管理，项目管理工具，面向对象的分析与设计方法，UML 方法等。为了保持教材内容的先进性和实用性，本书还包含了面向对象软件工程学方面的内容，并在附录 A 中提供软件文档的书写规范。

本书可作为高等院校计算机及相关专业的教材，也可作为从事软件开发与应用的工程人员的参考书。

图书在版编目（CIP）数据

软件工程实用教程 / 陈明编著. —北京：电子工业出版社，2004.10
北京市高等教育精品教材立项项目
ISBN 7-121-00402-X

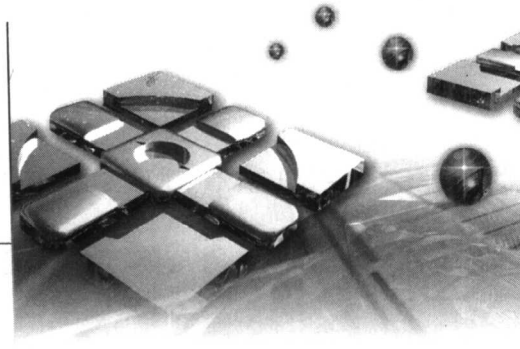
Ⅰ.软… Ⅱ.陈… Ⅲ.软件工程—高等学校—教材 Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字（2004）第 098387 号

责任编辑：冉 哲 特约编辑：陈新中
印 刷：北京牛山世兴印刷厂
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
经 销：各地新华书店
开 本：787×980 1/16 印张：23.5 字数：526 千字
印 次：2004 年 10 月第 1 次印刷
印 数：5000 册 定价：29.80 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。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。

前　言



计算机软件是逻辑产品，而不是物理产品。软件与硬件具有完全不同的特征。软件产业是创新的、充满活力的和成功的产业。

计算机软件现已成为一种新的驱动力，是进行决策的引擎，是现代工程研究和问题解决的基础，在各种类型的应用系统中无所不在、广泛应用。

软件危机是指软件开发和维护过程中遇到的一系列严重问题。例如，如何开发软件，如何满足用户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等一系列问题。

为了克服与摆脱软件危机，诞生了软件工程学。从 1969 年提出软件工程的概念以来，软件工程学历经 30 多年的飞速发展，逐渐成熟，现已成为计算机科学与技术领域中一门重要的学科。软件工程学的目标是以提高软件生产的质量与效率为宗旨，研究一套科学的工程方法，并开发相应的软件工具系统，用来指导和帮助软件的开发与研究工作，起到技术保障与促进作用。

软件开发工具是为了支持软件生存周期中某一阶段的任务实现而使用的计算机程序。软件开发环境是一组相关的软件工具的集合，它们组织在一起支持某种软件开发方法或某种软件开发模型。软件开发工具与环境是软件工程的重要组成部分，对于提高软件生产率，改进软件质量有很大作用。

本书系统地介绍了软件工程学的内容，主要包括：软件工程概述、可行性分析、需求分析、概要设计、详细设计、编码、测试与维护，软件开发工具与环境，Power Designer，软件项目管理，软件配置管理，软件质量管理，项目管理工具，面向对象的分析与设计方法，UML 方法等。

软件工程学是一门实践性极强的实用学科，在学习中，不仅要能掌握其理论原则与方法，更重要的是能学会熟练地应用。计算机科学与技术专业和相近专业的毕业生，有相当一部分人将从事计算机软件开发和应用方面的工作。通过软件工程的理论学习与实践，可以培养学生用软件工程的方法开发软件的习惯和素质，并在软件开发的工作中得

以贯彻。

本书内容系统全面，注重基本概念的解释和方法的说明，各章都附有小结和习题，便于学习总结和练习。

本书在结构上呈积木状，各章内容相对独立，但在逻辑上是一个整体，便于选择性学习。

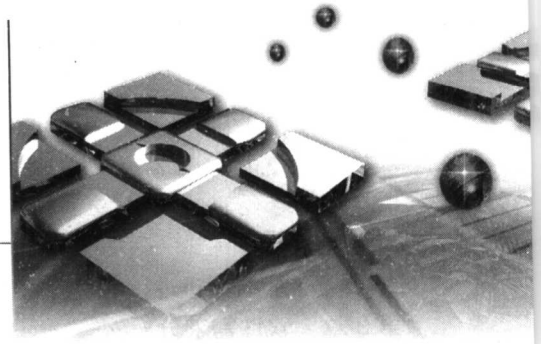
本书在取材上注重实用，避开了过多的理论说明和推演，有助于快速掌握原则和方法。

由于作者水平有限，书中不足之处在所难免，敬请批评指正。

陈　明

2004 年于北京

目　录



第 1 部分　概　　述

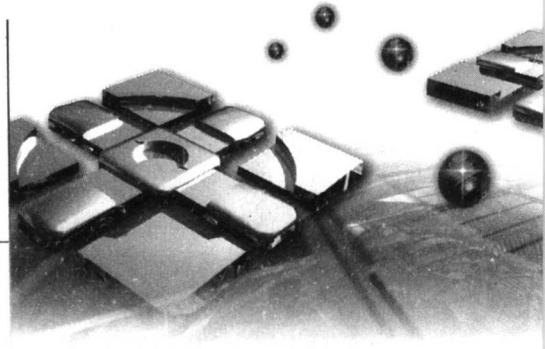
第 2 部分　软件生存周期

第 3 部分 软件开发工具与设计工具

第 4 部分　软 件 管 理

第 5 部分　面向对象方法学

第 1 部分



概　　述

第 1 章　软件工程简介

计算机科学是一门年轻的学科。自 20 世纪 40 年代中期第一台计算机问世以来，这一学科得到了飞速的发展。

计算机早期的应用大致具有以下特点：早期的程序通常较小，一般是由个人单独完成的；它们大都由相关领域的专家开发和使用。程序要解决的问题主要是技术方面的，开发的重点是用某种编程语言高效地表达已知的运算规则。其中典型的情况是：数字数据的输入（从穿孔卡片或者穿孔纸带中读取），以及数据输出（打印在纸上）。如果程序含有错误，程序员就要研究八进制或者十六进制的内存堆。有的时候，程序甚至是在控制台的二进制的可读寄存器上执行的。

计算机现在的应用在许多方面发生了改变。程序通常都很大，一般是由几个团体经过几年的时间合作完成的。程序员通常不是所开发系统的使用者，一般不具备相应领域的专业知识。程序与日常生活的关系越来越密切，如自动银行出纳系统、班机预订系统、工资管理系统、电子商务系统等。

1.1　软件、软件危机和软件工程的定义

1.1.1　软件及其特点

软件（Software）是计算机系统中与硬件（Hardware）相互依存的另一部分，它包括程序（Program）、相关数据（Data）及其说明文档（Document）。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，数据是程序能正常操纵信息的数据结构，文档是与程序开发维护和使用相关的各种图文资料。

软件同传统的工业产品相比，有其独特性：

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。一旦研制开发成功，就可以大量复制。所以软件的质量控制，必须重点在软件开发方面下工夫。

③ 软件在使用过程中没有磨损、老化的问题。软件在生存周期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，而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，软件可移植性差。

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，而直接影响到项目的成败。

1.1.2　软件危机

早在 20 世纪 60 年代，人们已经开始意识到编程技术已经落后于软件的发展。对大多数人而言，编程还是一门艺术而不是手艺；另一个问题是许多程序员没有接受过正规的教育，他们边开发边学习；在组织方面，解决问题的方法通常是增加越来越多的程序员到项目中去。

结果是，软件通常难以按时交付，程序也不能像使用者所期望的那样工作，程序很少能够适应变化的环境，而且在软件交付给客户之后又发现了很多错误。“软件危机”（Software Crisis）就这样开始了。

软件危机指的是在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。

1968 年，北大西洋公约组织（NATO）的计算机科学家在联邦德国召开的国际学术会议上第一次提出了“软件危机”这个名词。

1．软件危机的表现

软件危机主要涉及两方面问题：第一，如何开发软件，以满足对软件的日益增长的需求；第二，如何维护数量不断膨胀的已有软件。具体地讲，软件危机主要有以下表现。

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往往很不充分，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。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。很多程序中的错误是难以改正的，实际上不能使这些程序适

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。没有实现软件的可重用性，造成重复开发功能类似的软件。

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这些文档资料来管理和评价软件开发过程的进展状况；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，这些文档资料更是至关重要的。

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不断提高，导致硬件成本逐年下降，然而软件开发则需要大量人力，使软件成本上升。

以上列举的仅仅是软件危机的一些较明显的表现，与软件开发和维护相关的问题远远不止这些。通过对以上软件危机的表现的分析，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。

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，大致表现在以下几方面。

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。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而在测试阶段没能检测出来的错误。

软件不同于一般程序，它的一个显著特点是规模庞大。因此，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，还涉及到严格科学的管理问题，其开发和维护必然相当困难。

② 虽然软件本身独有的特点确实给开发和维护带来一些客观困难，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经过实践考验证明是正确的方法，许多困难是完全可以克服的。但事实上，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。

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，只有用户真正了解他们自己的需要，同时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、具体地了解用户的要求。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开始，也