普通高等教育“十一五”精品规划教材

实用软件工程学

SHIYONG RUANJIAN GONGCHENGXUE

主编　庄晋林　杨志宏

中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十一五”精品规划教材

实用软件工程学

主编　庄晋林　杨志宏
参编　董海祥　张晓红　杨雅军
　　石　燕　唐静静

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书从实用的角度，力求点面兼顾、深入浅出地介绍软件工程学的基本概念、方法和技术。同时，在注重基本概念、基本原理的基础上，强调了实践的重要性和必要性，部分章节给出了应用案例，可作为案例教学的素材，并在附录中编排了课程设计内容，供读者进行实践教学。

全书分为10章，主要内容包括软件工程学概述、软件需求分析、软件设计（包括概要设计与详细设计）、编码与语言选择、软件测试、软件维护、面向对象方法、软件复用、软件项目管理与软件质量保证简介、软件工程环境。

本书的适用范围主要是作为计算机科学与技术及相关专业应用型本科或专科学生的教材，也可作为相关专业的教师或工程技术人员的参考书。

图书在版编目（CIP）数据

实用软件工程学/庄晋林，杨志宏主编．—北京：中国水利水电出版社，2009

普通高等教育“十一五”精品规划教材

ISBN 978-7-5084-6439-8

Ⅰ．实… Ⅱ．①庄…②杨… 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（2009）第050974号

书　　名	普通高等教育“十一五”精品规划教材 **实用软件工程学**
作　　者	主编　庄晋林　杨志宏
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路1号D座　100038） 网址：www.waterpub.com.cn E-mail：sales@waterpub.com.cn 电话：（010）68367658（营销中心）
经　　售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643 全国各地新华书店和相关出版物销售网点
排　　版	中国水利水电出版社微机排版中心
印　　刷	北京纪元彩艺印刷有限公司
规　　格	184mm×260mm　16开本　22印张　522千字
版　　次	2009年6月第1版　2009年6月第1次印刷
印　　数	0001—3000册
定　　价	**38.00**元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

版权所有·侵权必究

普通高等教育“十一五”精品规划教材

实用软件工程学

主　编　庄晋林　杨志宏

参　编　董海祥　张晓红　杨雅军

石　燕　唐静静

前言

在信息技术广泛普及的今天，作为信息化重要基础的软件产业已经成为国际竞争的焦点和各国竞相发展的战略性先导产业。国家先后出台了一系列政策、法规推动和鼓励软件产业的发展。软件产业是指从事软件开发、销售及相关服务等业务的企业组成的集合体。其中软件开发的技术水平是决定整个软件产业发展的关键因素。软件工程学是计算机科学与技术学科中研究软件开发的一个重要分支学科。

在20世纪60年代，伴随着西方国家现代化进程的不断加快，计算机软件技术在各行各业迅速普及，人们对软件的需求日益复杂化，对软件质量的要求也越来越高，软件开发面临着日益严峻的挑战。在“软件危机”的强烈冲击下，软件工程学应运而生，经过40余年的发展，如今已经成为包含一整套的理论、方法、技术和工具的较为完善的学科体系。但是，正如著名软件工程学者B. W. Boehm总结软件工程的基本原理时提出的，要“承认不断改进软件工程实践的必要性”，从而表明在当今软件工程技术日趋成熟的条件下，仍然需要不断改进、发展，朝着最终彻底消除“软件危机”的目标前进。

本书的适用范围主要是作为计算机科学与技术及相关专业应用型本科或专科学生的教材或教学参考书。书中从实用的角度，力求点面兼顾、深入浅出地介绍软件工程学的基本概念、方法和技术。希望读者通过阅读和学习，能够了解软件开发过程中的主要活动，初步认识当今主流的软件开发方法及其关键技术。在深入理解基本概念、基本原理的同时，动手实践是掌握软件工程方法和技术的唯一途径。为此，在部分章节中给出了应用案例，可用于教学中案例分析的素材。并在附录中给出了课程设计内容，读者可以通过多人合作参与课程设计或小型项目的开发，逐步培养团队精神和工程化意识，为今后从事软件开发工作奠定良好的基础。

本书分为10章，可以按照60学时左右安排理论教学。各章主要内容是软件工程学概述、软件需求分析、软件设计（包括概要设计与详细设计）、编码与语言选择、软件测试、软件维护、面向对象方法、软件复用、软件项目管

此为试读，需要完整PDF请访问：www.ertongbook.com

理与软件质量保证简介、软件工程环境。附录 A 为课程设计，附录 B 为软件工程文档规范。

全书编写工作分工如下：庄晋林编写第 7 章 7.1～7.4 节和 7.8 节及附录 A；杨志宏编写第 4、5 章；董海祥编写第 1、8、10 章；张晓红编写第 3 章；杨雅军编写第 2 章及附录 B；石燕编写第 7 章 7.5～7.7 节；唐静静编写第 6、9 章。由庄晋林、杨志宏负责统稿。

由于时间仓促及水平所限，书中难免存在一些错误和偏颇，恳请读者批评指正。

编 者

2009 年 2 月

目录

第1章 软件工程学概述

1.1 软件发展史与软件危机

自世界上第一台电子计算机问世以来，计算机技术日趋成熟，运算速度不断提高，存储容量不断扩大，从单机运行已经发展到网络环境。计算机硬件技术的每次突破，都为软件技术的发展提供了更加广阔的空间，开拓了更新、更广阔的应用领域。随着新的电子元器件的出现，计算机硬件的性能和质量逐年提高，并且价格大幅度下降，使得计算机的应用几乎遍及社会的各个领域，成为人们日常工作和生活中不可缺少的工具。人们对软件的需求也急剧增加，软件系统也从简单发展到复杂，从小型发展到大型，由封闭系统发展到开放系统。在软件应用需求发展的过程中，软件开发方法也从注意技巧发展为注重管理，力图在可接受的性价比条件下，不断改进个人和软件开发组织的开发过程，强调在各自条件下追求软件过程的改进。

1.1.1 软件的特点

要知道什么是软件工程，首先应明确什么是软件。软件是计算机系统中与硬件相互依存的另一部分，它是包括程序、数据及相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料。

软件具有以下特点：

（1）软件是一种逻辑实体，而不是具体的物理实体，因而它具有抽象性。这个特点使它与计算机硬件或其他工程对象有着明显的差别。人们可以把它记录在介质上，但却无法看到软件的形态，而必须通过测试、分析、思考、判断等行为去了解它的功能、性能及其他特性。

（2）软件与硬件的生产方式不同，它没有明显的制造过程。对软件的质量控制，必须着重在软件开发方面下功夫。一旦某一软件项目研制成功，以后就可以大量地复制同一内容的副本，即其研制成本远远大于其生产成本。软件故障往往是在开发时产生而在测试时没有被发现的问题。所以，要保证软件的质量，必须着重于软件开发过程，加强管理和减少故障。

（3）在软件的运行和使用期间，没有像硬件那样的机械磨损、老化问题。

（4）软件的开发和运行常常受到计算机系统的限制，对计算机系统有着不同程度的依赖性。在软件的开发和运行中必须以硬件提供的条件为基础。为了消除这种依赖关系，在软件开发中提出了软件移植的问题，并且把软件的可移植性作为衡量软件质量的因素之一。

（5）软件的开发尚未完全摆脱手工的开发方式。由于传统的手工开发方式仍然是主要方式，软件开发的效率受到很大的限制。因此，应促进软件技术发展，提出和采用新的开发方法。例如，近年来出现的充分利用现有软件的复用技术、自动生成技术和其他一些有效的软件开发工具或软件开发环境，既方便了软件开发的质量控制，也提高了软件的开发效率。

（6）软件本身是复杂的。软件的复杂性可能来自它所反映的实际问题的复杂性，也可能来自程序逻辑结构的复杂性。

（7）软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动，它的成本是比较高的。

（8）相当多的软件工作涉及社会因素。许多软件的开发和运行涉及机构、体制及管理方式等问题，甚至涉及人的观念和人们的心理。它直接影响到项目的成败。

注意：软件不仅包括程序，还包括文档。所以，软件的开发不仅仅是编写程序，还包括编写相关的文档。首先来看一下软件发展的历史和一些相关的数据。

1.1.2　软件发展简史

1. 个体手工方式时期

在计算机系统发展的早期（20 世纪 60 年代中期以前），人们最关心的是计算机能否可靠、持续地运行等问题。当时，硬件通用化程度已经比较高了，软件却是为每项具体应用而专门编写的。这时的软件通常是规模较小的程序，编写者和使用者往往是同一个（或同一组）人。几乎没有什么系统化的标准方法可遵循。对软件的开发没有任何管理方法，一旦任务超时或者成本提高，程序员才开始弥补。这种个体化的软件开发方式，使得程序的开发结果，只有程序框图和源程序清单可以保留下来。同时由于硬件体积大、存储容量小、运算速度慢，因此特别讲究编程技巧，以解决计算机内存容量不足和运算速度太慢的矛盾。由于过分追求编程技巧，开发出的程序除程序作者之外，其他人很难读懂。所以，这个时期也称为个体手工方式时期。

2. 软件作坊时期

计算机系统发展的第二个时期（20 世纪 60 年代中期到 70 年代末期）为软件作坊时期。此时，多道程序设计、多用户系统引入了人机交互的新概念。人机交互技术打开了计算机应用的新世界以及硬件和软件配合的新层次，出现了实时系统和第一代数据库管理系统。这个时期，软件规模相当大，软件产业已经萌芽，软件产品广泛销售，软件数量急剧增加。

该时期软件开发的主要特征表现如下：

（1）因程序的规模增大，程序的开发需要多人分工协作，软件开发的方式由“个体生产”发展到“软件作坊”。

（2）程序的运行、维护、使用已不再由一个人承担。

（3）程序已不再是硬件的附属品，而是计算机系统中与硬件相互依存、共同发挥作用的不可缺少的部分。但是，“软件作坊”基本上仍然沿用早期形成的个体化软件开发方法。随着计算机应用的日益普及，软件数量急剧膨胀。在程序运行时发现的错误必须设法改正；用户有了新的需求时必须相应地增删或修改程序；硬件或操作系统更新时，通常需要

此为试读，需要完整PDF请访问：www.ertongbook.com

修改程序以适应新的操作环境。上述种种软件维护工作，以令人吃惊的比例耗费资源。更严重的是，许多程序的个体化特性使得它们最终成为不可维护的。“软件危机”就这样开始出现了！

3. 软件工程时期

计算机系统发展的第三个时期始于20世纪70年代末期并跨越了近20年。在这一时期，以软件的产品化、系列化、工程化、标准化为特征的软件产业发展起来，打破了软件生产的个体化特征，有了软件工程化的设计原则、方法、标准可以遵循。软件开发的成功率大大提高，软件的质量在一定程度上也有了很大的保证。在分布式系统中，各台计算机同时执行某些功能，并与其他计算机之间的通信，极大地提高了计算机系统的复杂性。广域网、局域网、高带宽数字通信以及对即时数据访问需求的增加都对软件开发提出了更高的要求。

4. 现代软件工程时期

随着互联网的应用与普及，软件开发已经不再着重于单台计算机系统和程序，而是面向计算机和软件的综合影响。由复杂的操作系统控制的强大的桌面机、广域网络和局域网络，配以先进的软件应用已成为标准。计算机体系结构迅速地从集中的主机环境转变为分布的客户机/服务器环境。世界范围的信息网提供了一个基本结构，信息高速公路和网际空间连通已成为令人关注的热点问题。事实上，Internet可以看作是能够被单个用户访问的软件，计算机发展正朝着社会信息化和软件产业化方向发展，从技术的软件工程阶段过渡到社会信息化的计算机系统。随着第四个时期的进展，一些新技术开始出现。面向对象技术将在许多领域中迅速取代传统软件开发方法。

1.1.3 软件危机

软件危机是指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。这些问题绝不仅仅是不能正常运行的软件才具有的，实际上，几乎所有软件都不同程度地存在这些问题。这是因为软件本身是一个逻辑实体，开发过程是一个渐进的思考的过程，很难进行管理。开发者根据自己的喜好和习惯来编写程序，没有统一的标准和规范可以遵循。

随着软件规模越来越大，人们在实践中发现随心所欲编写的程序给维护带来了很大的麻烦。程序晦涩难懂，不同的程序员、不同时期编写的模块接口不统一等问题使得软件开发变得非常困难，往往投入很大，收获甚微。开发的软件漏洞百出，或无人使用。

1. 软件危机主要的表现

(1) 软件的复杂度越来越高，传统的软件开发方式已无法满足要求。计算机软件系统的规模和复杂程度，已经不是程序员各自为战的手工作坊式的生产方式可以解决的。

(2) 对软件开发成本和进度的估计常常很不准确。实际成本比估计成本有可能高出一个数量级，实际进度比预期进度拖延几个月甚至几年的现象并不罕见。这种现象降低了软件开发组织的信誉。而为了赶进度和节约成本所采取的一些权宜之计又往往影响了软件产品的质量，从而不可避免地会引起用户的不满。

(3) 用户对软件的满意度差。软件开发人员常常在对用户要求只有模糊的了解，甚至对所要解决的问题还没有明确认识的情况下，就匆忙着手编写程序。软件开发人员和用户之间的信息交流往往很不充分，闭门造车必然导致最终的产品不符合用户的实际需要。

（4）软件产品的质量往往靠不住。软件可靠性和质量保证的确切的定量概念刚刚出现不久，软件质量保证技术（审查、复审和测试）还没有坚持不懈地应用到软件开发的全过程中，这些都导致软件产品发生质量问题。

（5）软件常常是不可维护的。很多程序中的错误是非常难改正的，实际上不可能使这些程序适应新的硬件环境，也不能根据用户的需要在原有程序中增加一些新的功能。可重用（或复用）的软件还是一个没有完全做到的、正在努力追求的目标，人们仍然在重复开发类似的或基本类似的软件。

（6）软件通常没有适当的文档资料。计算机软件不仅仅是程序，还应该有一整套文档资料。这些文档资料应该是在软件开发过程中产生出来的，而且应该是最新式的（即和程序代码完全一致的）。软件开发组织的管理人员可以使用这些文档资料作为参照，来管理和评价软件开发工程的进展状况；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，这些文档资料更是必不可少的。缺乏必要的文档资料或者文档资料不合格，必然给软件开发和维护带来许多严重的困难和问题。

（7）软件成本在计算机系统总成本中所占的比例持续上升。由于微电子学技术的进步和生产自动化程度不断提高，硬件成本逐年下降，然而软件开发需要大量人力，软件成本随着通货膨胀以及软件规模和数量的不断扩大而持续上升。美国在 1985 年软件成本大约已占计算机系统总成本的 90%。

（8）软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的趋势。软件产品供不应求的现象使人类不能充分利用现代计算机硬件提供的巨大潜力。

例如，20 世纪末的“千年虫”问题，由于设计的问题，“千年虫”如同一颗定时炸弹，在 2000 年到来的前几年，引起了恐慌，各种补救、测试手段蜂拥而至。另一个软件危机经典案例是美国 IBM 公司在 1963～1966 年开发的 IBM 360 操作系统，这一项目花了 5000 人一年的工作量，最多时有 1000 人投入开发工作，写出了 100 万行源程序，包含 4000 多个模块，耗资达数亿美元。该系统投入使用后发现了 2000 多个错误。OS/360 系统的负责人 F. D. BRooms 曾这样形象地描述了开发过程中的困难和混乱：“像巨兽在泥潭中作垂死挣扎，挣扎的越猛，泥浆就沾得越多，最后没有一个野兽能逃脱淹没在泥潭中的厄运。程序设计就像是这样一个泥潭，一批批程序员在泥潭中挣扎……没人料到问题竟会这样棘手……”

2. 软件危机产生的原因

在软件开发和维护的过程中存在这么多严重问题，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不正确有关。

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；软件样品即是产品，试制过程也就是生产过程；软件不会因使用时间过长而“老化”或“用坏”；软件具有可运行的行为特性，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，软件质量也较难评价，因此管理和控制软件开发过程十分困难；软件质量不是根据大量制造的相同实体的质量来度量，而是与每一个组成部分的不同实体的质量紧密相关，因此，在运行时所出现的软件错误几乎都是在开发时期就存在而一直未被发现的，改正这类错误通常意味着改正或修改原来的设计，这就在客观上使得软件维护远比硬件维

护困难；软件是一种信息产品，具有可延展性，属于柔性生产，与通用性强的硬件相比，软件更具有多样化的特点，更加接近人们的应用问题。随着计算机应用领域的不断扩大，99%的软件应用需求已不再是定义良好的数值计算问题，而是难以精确描述且富于变化的非数值型应用问题。因此，当人们的应用需求变化发展时，往往要求通过改变软件来使计算机系统满足新的需求，维护用户业务的延续性。

危机原因来自于软件开发人员的以下弱点：其一，软件产品是人的思维结果，因此软件生产水平最终在很大程度上取决于软件人员的教育、训练和经验的积累；其二，对于大型软件往往需要许多人合作开发，甚至要求软件开发人员深入应用领域的问题研究，这样就需要在用户与软件人员之间以及软件开发人员之间相互通信，在此过程中难免发生理解的差异，从而导致后续错误的设计或实现，而要消除这些误解和错误往往需要付出巨大的代价；其三，由于计算机技术和应用发展迅速，知识更新周期加快，软件开发人员经常处在变化之中，不仅需要适应硬件更新的变化，而且还要涉及日益扩大的应用领域问题研究；软件开发人员所进行的每一项软件开发几乎都必须调整自身的知识结构以适应新的问题求解的需要，而这种调整是人所固有的学习行为，难以用工具来代替。

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，也确实积累和总结出了许多成功的经验。如果坚持不懈地使用经过实践考验证明是正确的方法，许多问题是完全可以避免的，过去也确实有过一些成功的范例。但是，目前相当多的软件专业人员对软件开发和维护工作还有不少糊涂观念，在实践过程中或多或少地采用了错误的方法和技术，这可能是使软件问题发展成软件危机的主要原因。

3. 消除软件危机的途径

(1) 采用工程化方法和途径来开发与维护软件。软件开发是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目。必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法。应该推广使用在实践中总结出来的开发软件的成功技术和方法，并且研究探索更好、更有效的技术和方法，尽快消除在计算机系统早期发展阶段形成的一些错误概念和做法。将软件的生成问题在时间上分成若干阶段以便于分步而有计划地分工合作，在结构上简化若干逻辑模块。把软件作为工程产品来处理，按计划、分析、设计、实现、测试、维护的周期来进行生产。

(2) 应该开发和使用更好的软件工具。在软件开发的每个阶段都有许多繁琐重复的工作要做，在适当的软件工具辅助下，开发人员可以把这类工作做得既快又好。如果把各个阶段使用的软件工具有机地集合成一个整体，支持软件开发的全过程，则称为软件工程支撑环境。

(3) 采取必要的管理措施。软件产品是把思维、概念、算法、组织、流程、效率、质量等多方面问题融为一体的产品。但它本身是无形的，所以有不同于一般工程项目的管理。它必须通过人员组织管理、项目计划管理、配置管理等来保证软件按时高质量完成。

总之，为了解决软件危机，既要有技术措施（包括方法和工具），又要有必要的组织管理措施。软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门学科。

1.2　软 件 工 程 学 的 概 念

1.2.1　软件工程的定义

软件工程是指导计算机软件开发和维护的一门工程学科。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技术和当前能够得到的最好的技术方法结合起来，这就是软件工程。

Boehm 为软件工程下的定义："运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料。"

Fritz Bauer 曾经为软件工程下了定义："软件工程是为了经济地获得能够在实际机器上有效运行的可靠软件而建立和使用的一系列完善的工程化原则。"

1983 年 IEEE 给出的软件工程的定义为："软件工程是开发、运行、维护和修复软件的系统方法"，其中，"软件"的定义为：计算机程序、方法、规则、相关的文档资料以及在计算机上运行时所必需的数据。

后来尽管又有一些人提出了许多更为完善的定义，但其主要思想都是强调在软件开发过程中需要应用工程化原则的重要性。

软件工程方法学包括 3 个要素：方法、工具和过程。

软件工程方法学为软件开发提供了"怎样做"的技术。它包括了多方面的任务，如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法过程的设计、编码、测试及维护等。

1.2.2　软件工程的基本策略

软件的开发是一个包含了比较、分类、推理、归纳、演绎、综合、分析的思考过程。人们就是在这种反复进行的自下而上的分解、演绎中，不断地认识客观世界、描述客观世界，并最终将其转化为计算机世界中的程序代码。软件开发中的基本策略，对于确保软件工程项目的顺利进行、确保软件目标系统的质量是十分重要的。这些策略也是软件开发中众多具体方法的指导思想。

（1）抽象与模型策略：简言之，抽象就是抽出事物的本质特性或者具有共同特性的信息而暂时不考虑它们的细节，并将这些特性用各种概念精确地加以描述，模型是描述抽象结果的一种简化的结构。对软件系统进行分析和设计时，可以有不同的抽象层次，既能够把握问题的整体，又能深入细节。在软件工程的生命周期中，从问题定义到系统实现，每进展一步都可以看作是对软件解决方法的抽象过程的一次细化。

（2）复用策略：复用指利用现成的东西。将具有一定集成度并可以重复使用的软件组成单元称为软构件。软件复用可以表述为：构建新的软件系统可以不必每次都从零做起，直接使用已有的软构件，即可组装（或加以合理修改）成新的系统。

（3）分解策略：分解是指把一个复杂的问题分解成若干个简单的问题，然后逐个解决。软件人员在执行分解时，应该着重考虑：复杂问题分解后，每个问题能否用程序实现？所用程序最终能否集成为一个软件系统并有效解决原始的复杂问题。

（4）优化策略：软件的优化是指优化软件的各个质量要素，如提高运行速度、提高对

内存资源的利用率、使用户界面更加友好、使三维图形的真实感更强等。优化策略的复杂之处是很多目标存在千丝万缕的关系，不可能使所有的目标都得到优化，只能通过“折衷”来协调各个质量要素，实现整体质量的最优。

1.2.3 软件工程应遵循的原则

1968年在联邦德国召开的国际会议上正式提出并使用了“软件工程”这个术语，运用工程学的基本原理和方法来组织和管理软件生产。后来还发展了与软件有关的心理学、生理学和经济学等方面的学科。在这期间，研究软件工程的专家、学者们陆续提出了100多条关于软件工程的准则。这100多条软件工程准则可以概括为下述7条基本原则。

1. 用分阶段的生命周期计划严格管理

经统计发现，在不成功的软件项目中有一半左右是由于计划不周全造成的，可见把建立完善的计划作为第一条基本原理是吸取了前人的教训而提出来的。

一个软件从定义、开发、使用和维护，直到最终被废弃，要经历一个漫长的时期，通常把软件经历的这个漫长的时期称为生命周期。在软件开发与维护的漫长过程中，需要完成许多不同性质的工作，所以应把软件生命周期划分为若干个阶段，并相应地制定出可行的计划，然后按照这个计划对软件的开发与维护工作进行管理。不同层次的管理人员都必须严格按照计划，各尽其职地管理软件开发与维护工作，绝不能受客户或上级人员的影响而擅自背离预定计划。

2. 坚持进行阶段评审

软件的质量保证工作不能等到编码阶段结束之后再进行。这样说至少有两个理由：第一，大部分错误是在编码之前造成的，例如，根据Boehm等人的统计，设计错误占软件错误的63%，编码错误仅占37%；第二，错误发现与改正得越晚，所需付出的代价也越高（见图1-1）。因此，在每个阶段都要进行严格的评审，以便尽早发现在软件开发过程中所犯的错误，是一条必须遵循的重要原则。

3. 实行严格的产品控制

在软件开发过程中不应随意改变需求，因为改变一项需求往往需要付出较高的代价。但是，在软件开发过程中改变需求又是难免的，由于外部环境的变化，相应地改变用户需求是一种客观需要，显然不能硬性禁止客户提出改变需求的要求，而只能依靠科学的产品控制技术来顺应这种要求。也就是说，当改变需求时，为了保持软件各个配置成分的一致性，必须实行严格的产品控制，其中主要是实行基准配置管理。所谓基准配置又称为基线配置，它们是经过阶段评审后的软件配置成分（各个阶段产生的文档或程序代码）。基准配置管理也称为变动控制：一切有关修改软件的建议，特别是涉及对基准配置的修改建议，都必须按照严格的规程进行评审，获得批准以后才能实施修改，绝对不能随意进行修改。

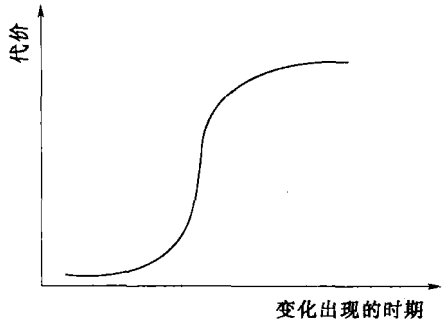

图1-1 同一变动所付出的代价随时间变化的趋势

4. 采用现代程序设计技术

从提出软件工程的概念开始，人们一直把主要精力用于研究各种新的程序设计技术，并进一步研究各种先进的软件开发与维护技术。实践表明，采用先进的技术不仅可以提高软件开发和维护的效率，而且可以提高软件产品的质量。

5. 结果应能清楚地审查

软件产品不同于一般的物理产品，它是看不见摸不着的逻辑产品。软件开发人员（或开发小组）的工作进展情况可见性差，难以准确度量，从而使得软件产品的开发过程比一般产品的开发过程更难以评价和管理。为了提高软件开发过程的可见性，更好地进行管理，应该根据软件开发项目的总目标及完成期限，规定开发组织的责任和产品标准，从而使得所得到的结果能够清楚地审查。

6. 开发小组的人员应该少而精

软件开发小组的人员合理构成的原则是应该少而精，即小组的组成人员的素质应该好，而人数不应过多。高素质的人员会大大提高软件的开发效率，且明显减少软件中的错误。此外，随着开发小组人员数目的增加，因交流问题和讨论情况而造成的通信开销也急剧增加，所以要保证软件开发小组人员少而精。

7. 承认不断改进软件工程实践的必要性

遵循上述 6 条基本原理，就能够按照当代软件工程基本原理实现软件的工程化生产，但是，仅有上述 6 条原理并不能保证软件开发与维护的过程能赶上时代前进的步伐，能跟上技术的不断进步。因此，Boehm 提出应把承认不断改进软件工程实践的必要性作为软件工程的第 7 条基本原理。按照这条原理，不仅要积极主动地采纳新的软件技术，而且要注意不断总结经验。

1.3 软件生命周期

软件工程采用的生命周期方法就是从时间角度对软件的开发与维护这个复杂问题进行分解，将软件生命漫长的时期分为若干阶段，每个阶段都有其相对独立的任务，然后逐步完成各个阶段的任务。概括地说，软件生命周期由软件定义、软件开发和软件维护（也称为运行维护）3 个时期组成，每个时期又进一步划分成若干个阶段。

1.3.1 软件定义

软件定义时期的任务是：确定软件开发工程必须完成的总目标；确定工程的可行性；导出实现工程目标应该采用的策略及系统必须完成的功能；估计完成该项工程需要的资源和成本，并且制定工程进度表。这个时期的工作通常又称为系统分析，由系统分析员负责完成。软件定义时期通常进一步划分成 3 个阶段，即问题定义、可行性研究和需求分析。

1. 问题定义

软件定义阶段必须考虑的问题是“做什么”。正确理解用户的真正需求，是系统开发成功的必要条件。通过对客户的访问调查，系统分析员扼要地写出关于问题性质、工程目标和工程规模的书面报告，经过讨论和必要的修改之后这份报告应该得到客户的确认。问题定义阶段是软件生命周期中最短的阶段。

此为试读，需要完整PDF请访问：www.ertongbook.com