普通高等教育“十五”国家级规划教材

软件工程
——理论、方法与实践

孙家广　主编
刘　强　编著

高等教育出版社

普通高等教育“十五”国家级规划教材

软件工程——理论、方法与实践

孙家广　主编
刘　强　编著

高等教育出版社

内 容 提 要

本书为普通高等教育“十五”国家级规划教材。由作者结合多年软件开发实践和近年讲授软件工程课程的教学经验编写而成，强调理论与实践的有机结合。全书共11章，第1章概括介绍软件工程的历史发展和基本原理，讨论IEEE和ACM最新提出的软件工程知识体系和软件工程职业道德规范；第2章讨论软件过程的基本思想和活动，介绍常见的软件过程模型和微软开发过程的实际案例；第3章讨论软件工程的管理技术，主要涉及人员管理、沟通管理、项目规划和风险管理等内容；第4章、第6章至第10章以面向对象技术为核心，全面、深入、系统地介绍软件开发各个阶段的任务、过程、方法和工具；第5章介绍软件工程中的形式化方法，包括时序逻辑、Z语言、Petri网等；第11章介绍软件进化的概念和方法，包括软件进化的特性、软件维护活动、逆向工程与再工程等内容。

全书注重内容的新颖性、条理性、系统性和实用性，始终以大量的开发实例贯穿全书，可作为计算机专业有关高年级本科生和低年级研究生学习软件工程课程的教材，也可供软件从业人员参考使用。

图书在版编目（CIP）数据

软件工程：理论、方法与实践 / 孙家广主编；刘强编著. — 北京：高等教育出版社，2005.7
ISBN 7-04-016308-X

Ⅰ.软... Ⅱ.①孙...②刘... Ⅲ.软件工程
Ⅳ.TP311.5

中国版本图书馆CIP数据核字（2005）第073134号

策划编辑 倪文慧　　**责任编辑** 倪文慧
封面设计 于文燕　　**责任印制** 韩　刚

出版发行	高等教育出版社	**购书热线**	010-58581118
社　　址	北京市西城区德外大街4号	**免费咨询**	800-810-0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经　　销	北京蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印　　刷	北京市鑫霸印务有限公司		
开　　本	787×1092　1/16	**版　　次**	2005年7月第1版
印　　张	17.5	**印　　次**	2005年7月第1次印刷
字　　数	370 000	**定　　价**	25.00元

物料号　16308-00

前　言

随着信息技术的不断发展，软件产业已经成为信息产业的核心和国民经济信息化的基础。在软件产业走向工程化和规模化发展的过程中，软件工程集成了软件开发的过程、方法和工具，以解决软件生产的质量和效率问题为宗旨，对软件产业的发展起到了重要的技术保障和促进作用，最终实现软件的工业化生产。

从“软件工程”概念提出至今，有关软件的概念、思想、方法和技术层出不穷。特别是20世纪90年代以来，软件工程不仅从方法论的角度为管理人员和开发人员提供可见的结构和有序的思考方式，而且从大量软件开发的成功经验总结出设计模式、框架、部件库等，软件工程正在逐步发展为一门成熟的专业学科。

软件工程学科涉及的内容十分广泛，包括理论、方法、技术、标准、工具和管理等诸多方面。本书结合当前软件工程的理论和实践，以当前普遍流行的面向对象技术和UML语言为核心，介绍软件工程的基本概念、技术方法和实践原则。在过去的软件工程教学中，学生通常孤立地学习一些编程技术和软件工程技术，而缺乏贯穿软件开发整个过程的系统性认识和实践性应用。本书以IEEE最新发布的软件工程知识体系为基础构建内容框架，采用IEEE给出的一系列软件工程文档标准，从“可实践”软件工程的角度描述需求分析、软件设计、软件实现、软件测试以及软件开发的管理，力求使读者在学习基本理论和方法的过程中学会运用软件工程的思想解决实际问题。

本书反映了作者十余年在软件开发方面的实践经验和近几年在软件工程方面的教学成果，强调理论与实践的有机结合，始终以大量的开发实例贯穿全书，并注重内容的新颖性、条理性、系统性和实用性，期待所有的读者能够从本书中得到有价值的收获，并在自己的软件工程实践中提升自己解决问题的能力。

目标读者

本书适合作为计算机类专业高年级本科生和低年级研究生学习软件工程课程的教材，软件从业人员同样可以将其作为参考书来充实自己在软件工程方面的知识。

本书要求读者具备计算机专业的基础知识，掌握程序设计基础、数据结构、操作系统、数据库原理等知识，具有一定的编程能力。

本书的组织结构

全书共有11章，可以作为一个学期的课程进行教学。

- 第1章概括介绍软件工程的历史发展和基本原理，讨论IEEE和ACM最新提出的软件工程知识体系和软件工程职业道德规范；

- 第2章讨论软件过程的基本思想和活动，介绍常见的软件过程模型和微软开发过程的实际案例；
- 第3章讨论软件工程的管理技术，主要涉及人员管理、沟通管理、项目规划和风险管理等内容；
- 第4章、第6章至第10章以面向对象技术为核心，全面、深入、系统地介绍软件开发各个阶段的任务、过程、方法和工具；
- 第5章介绍软件工程中的形式化方法，包括时序逻辑、Z语言、Petri网等；
- 第11章介绍软件演化的概念和方法，包括软件进化的特性、软件维护活动、逆向工程与再工程等内容。

本书的编写

孙家广院士组织了本书的编写工作，刘强副教授编写了全书的第1章至第4章以及第6章至第11章，刘璘副教授编写了本书的第5章，研究生刘旸和万欣分别协助第6章、第9章和第11章的资料准备和编写工作。

本书内容在清华大学计算机系和软件学院的本科生课程教学中使用，书中教学案例MiniLibrary系统是课程实践实例。

感谢

清华大学软件学院和高等教育出版社对于本书的编写过程给予了很大的支持，很多教师和学生的鼓励和建议也对本书的出版起到了促进作用。在此，作者向所有对本书编写给予支持和帮助的人表示诚挚的谢意。最后，还要特别感谢高等教育出版的刘建元先生和倪文慧女士，由于他们的信任、鼓励和支持，本书得以顺利完成。

作 者

2005年7月

目　录

第1章 概 述

软件是人类思维创造的杰作，并成为人类现代生活的催化剂。今天，软件遍布整个世界，在生物工程、现代通信、宇宙探索、商务处理、工业控制等方面发挥出巨大的威力，推动了商业、科学和工程领域的跨越式发展，对整个社会的经济和文化产生了深远的影响。

在计算机诞生的初期，软件仅仅是计算机硬件的附属品，其作用和成本微乎其微。如今，软件以各种形式嵌入在越来越多的产品中，不仅成为影响系统功能和性能的关键因素，而且在整个系统的成本中占据着越来越大的比重。因此，如何以经济有效的方法开发高质量的软件是人们长期以来一直努力研究的主要问题。

软件工程是为了解决开发成本效益和软件质量的问题而产生的。从 1968 年 NATO (North Atlantic Treaty Organization，北大西洋公约组织)会议首次提出“软件工程”概念至今，虽然人们并没有彻底解决软件危机的问题，然而正是软件工程的发展促使软件取得了如此令人瞩目的成就。三十多年以来，人们更好地认识了软件开发过程，在软件的需求、设计、实现、测试和维护等方面提出了许多有效的方法，新的开发方法和开发工具在大型复杂软件系统的开发过程中起到了事半功倍的作用。如果没有这些复杂的软件，人们就不可能探索宇宙空间，也不可能拥有网络和现代化的通信技术，更不可能揭开人类基因的奥秘。

当前，软件工程仍然是一个正在迅速兴起的年轻学科，尚未形成完整的理论知识体系，需要大量的理论研究和工程实践。我们相信，随着该学科的日益成熟，软件工程必将对未来的软件开发产生更大的推动力。

1.1 软 件

在软件的发展过程中，软件从个性化的程序演变为工程化的产品，人们对软件的看法发生了根本性的变化。“软件 = 程序”显然不能涵盖软件的完整内容，除了程序之外，软件还包括与之相关的文档和配置数据，以保证这些程序的正确运行。

《IEEE Standard Glossary of Software Engineering Terminology》给出了有关软件的如下定义：

软件是计算机程序、规程以及运行计算机系统可能需要的相关文档和数据。

其中：① 计算机程序是计算机设备可以接受的一系列指令和说明，为计算机执行提供所需的功能和性能；② 数据是事实、概念或指令的结构化表示，能够被计算机设备接收、理解或

处理;③ 文档是描述程序研制过程、方法及使用的图文材料。

然而,软件的真正含义却不是一个形式的定义所能体现的。从软件的内容来看,软件更像是一种嵌入式的数字化知识,其形成是一个通过交互对话和抽象理解而不断演化的过程。

软件的应用领域十分广泛,呈现形式也是多种多样的,在某种程度上很难对软件的类型给出一个通用的界定。根据软件服务对象的范围不同,一般可以将软件划分为通用软件和定制软件两种类型。

1. 通用软件(Generic Software)

通用软件是由软件开发组织开发,面向市场用户公开销售的独立运行系统,像操作系统、数据库系统、字处理软件、绘图软件包和项目管理工具等都属于这种类型。有时,通用软件也被称为套装软件(Packaged Software)。

2. 定制软件(Customised Software)

定制软件由某个特定客户委托,软件开发组织在合同的约束下开发的软件,像企业 ERP 系统、卫星控制系统和空中交通指挥系统等都属于这种类型。

通用软件由开发组织根据市场调研自主提出产品需求,并以此进行设计和开发。其最大的优势在于,它可以通过近乎零成本的复制来分摊最初投入的一次性开发成本,最终使原本昂贵的软件产品的价格降至众多用户可接受的程度,从而有效地提高市场份额和利润。但是,为了分摊最初的开发投入,通用软件必须面对足够大的市场空间,其功能设计也只能面向大规模用户普遍存在的共性需求。

对于不同的用户来说,除了共性需求之外还存在着个性化的需求,而这些个性化需求对于很多用户来说,恰恰是应用的关键所在。定制软件完全是订单开发,即按照单个客户的个性化要求,以软件项目的方式为其提交个性化的解决方案,从而更好地满足客户的需求。

1.1.1　软件的特性

计算机在使社会生产力得到迅速解放、社会高度自动化和信息化的同时,却没有使计算机本身的软件生产得到类似的巨大进步。软件开发依然面临着过分依赖人工、软件难以重用、开发大量重复和生产率低下等问题,而导致这些问题的关键在于软件本身的特性。

1. 软件是复杂的

软件是人类思维和智能的一种延伸和在异体上的再现,远比任何以往人类的创造物都要复杂得多。在大型软件系统中,无数种数据、状态和逻辑关系的可能组合以及人类思维的复杂性和不确定性导致的理解歧义和差异,使整个系统的复杂性急剧增加,也使软件的设计、实现和测试都变得相当困难。

在著名的《没有银弹:软件工程中的根本和次要问题》一文中,Fred Brooks 先生认为正是软件固有的复杂性造成了软件开发的诸多问题。由于复杂性,人们难以全面理解问题,团队

成员之间的沟通也变得非常困难，从而导致了产品缺陷、成本超支和进度拖延；由于复杂性，描述和理解软件系统所有可能的状态是极其困难的，影响了产品的可靠性；由于软件结构及其依赖关系的复杂性，软件的任何更改和扩充都有可能带来灾难性的后果，形成所谓的“雪崩效应”。

2. 软件是不可见的

一般情况下，人们可以通过几何抽象模型准确地描述有形的物体，例如建筑师可以用平面图描述建筑物的结构，硬件工程师可以用电路图描述计算机的系统结构，甚至化学家也可以用分子模型描述客观事物的微观结构。但是，软件是客观世界空间和计算机空间之间的一种逻辑实体，不具有物理的形体特征。人们一直试图用不同的图形技术来描述软件结构，即便是现在流行的面向对象技术，仍然无法给出准确的、完整的描述。

由于软件的不可见性，定义“需要做什么”成为软件开发的根本问题。过去，几乎所有的技术、工具和过程都致力于解决“怎么做”这个次要问题，并且取得了很大的进展，但是在“做什么”的问题上，几十年来情况似乎并没有太大的改善。因此，我们需要知道描述哪些部分、忽略哪些部分，这就是软件的本质问题。

3. 软件是不断变化的

软件是纯粹思维活动的产物，它不会像硬件一样发生磨损，而是需要随着应用、硬件、用户和社会等各种因素的变化而不断地被修改和扩展。由于软件是人类思维和智能的一种延伸，因此当软件被真正应用之后，人们往往希望超越原有的应用边界进行软件功能的提升或扩展；另外，由于软件必须依附于硬件平台，因此需要随着硬件设备的更新和接口的不同而变化。

人们总是认为软件是很容易修改的，通常忽视了修改带来的副作用，即引入新的错误，造成故障率的升高。图 1.1 显示了软件修改对其质量带来的冲击，不断的修改最终导致软件的退化，从而结束其生命周期。

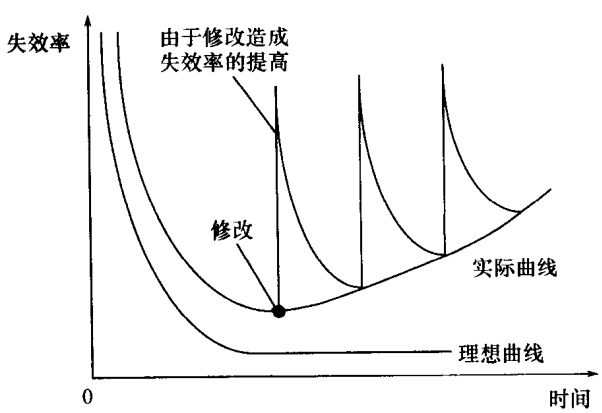

图 1.1　软件的失效率曲线

成功的软件都会发生演化，没有任何变化的软件一定是没有用的。虽然软件的易变性给软件开发带来了许多难题，但是这种固有的特性也给软件带来了异常的生命力。因此不能回避软件演化的问题，需要采用积极的态度和有效的方法将变更在可控的情况下予以实施。

4. 大多数软件仍然是定制的，而不是通过已有构件组装而成的

在软件的发展历程中，曾经涌现出许多开发技术和开发工具，当前流行的面向对象开发技术也日趋成熟，但是手工作坊式的软件开发方式仍占主导地位。随着数字化、网络化、智能化成为信息产业的发展趋势，软件复用和软件构件技术受到广泛的关注，并成为一种社会化的开发方法，有助于软件工程化、工厂化生产的实现。

1.1.2　软件的发展

伴随着第一台计算机的问世，计算机程序就出现了。在以后几十年的发展过程中，人们逐步认识了软件的本质特性，发明了许多有意义的开发技术与开发工具，同时软件的规模在不断扩大，其应用几乎渗透到各个领域。纵观整个软件的发展过程，大致可以将其分成以下4个重要的阶段。

1. 第一阶段：20世纪50－60年代

在计算机发展的早期阶段，计算机的主要应用是快速计算，出现了以Algol、Fortune等编程语言为标志的算法技术。在这一时期，程序设计被认为是一种任人发挥创造才能的活动，不存在什么系统化的方法和开发管理，程序的质量完全依赖于程序员个人的技巧。随着软件规模的扩大，20世纪60年代末期出现了“软件危机”。

2. 第二阶段：20世纪70年代

计算机应用开始涉及到各种以非数值计算为特征的商业事务领域，交互技术、多用户操作系统、数据库系统等随之发展起来，出现了以Pascal、Cobol等编程语言和关系数据库管理系统为标志的结构化软件技术。在这一时期，软件的概念不再仅仅是程序，还包括开发、使用、维护程序需求的所有文档，软件的工作范围从只考虑程序的编写扩展到从定义、编码、测试到使用、维护等整个软件生命周期，瀑布模型被普遍使用。

3. 第三阶段：20世纪80年代

微处理器的出现与应用使计算机真正成为大众化的东西，而软件系统的规模、复杂性以及在关键领域的广泛应用，促进了软件开发过程的管理及工程化开发。在这一时期，软件工程开发环境CASE及其相应的集成工具大量涌现，软件开发技术中的度量问题受到重视，出现了著名的软件工作量估计COCOMO模型、软件过程改进模型CMM等。20世纪80年代后期，以Smalltalk、C++等为代表的面向对象技术重新崛起，传统的结构化技术受到了严峻的考验。

4. 第四阶段:20 世纪 90 年代至今

Internet 技术的迅速发展使软件系统从封闭走向开放,Web 应用成为人们在 Internet 上最主要的应用模式,异构环境下分布式软件的开发成为一种主流需求,软件复用和构件技术成为技术热点,出现了以 Sun 公司的 EJB/J2EE、Microsoft 的 COM +/DNA 和 OMG 的 CORBA/OMA 为代表的 3 个分支。与此同时,需求工程、软件过程、软件体系结构等方面的研究也取得了有影响的成果。

进入 21 世纪,Internet 正在向智能网络时代发展,以网格技术(Grid Computing)和网络服务(Web Services)为代表的分布式计算日趋成熟,从而实现信息充分共享和服务无处不在的应用环境;高信度计算(Trustworthy Computing)普遍引起高度重视,从而为人们创造一个安全、可靠、值得信赖的环境。

1.1.3 软件危机

所谓软件危机,是指在计算机软件的开发和维护过程中遇到的一系列严重问题。软件危机在 20 世纪 60 年代末全面爆发,至今 40 年过去了,虽然软件开发的新工具和新方法层出不穷,但是软件危机依然没有消除。

1. 软件开发的成本和进度难以准确估计,延迟交付甚至取消项目的现象屡见不鲜

1995 年,美国 Standish 咨询集团公布了题为"混沌"的研究报告,如图 1.2 所示的研究数据表明:在 20 世纪 90 年代初期,软件项目的平均成功率只有 16.2%,在这里,成功的含义是指在计划的时间和预算内实现项目目标;仅 1995 年的一年间,有 31% 的项目在完工之前就被取消了;其余 53.8% 的项目由于各种原因在开发过程中遇到了这样或那样的问题,在开发成本、交付时间、产品功能或性能等方面没有实现预期的目标。近几年,有关统计资料显示:软件项目的平均成功率上升到 26%,但仍有 46% 的项目超出预算和最后期限,另有 28% 的项目没有成功。

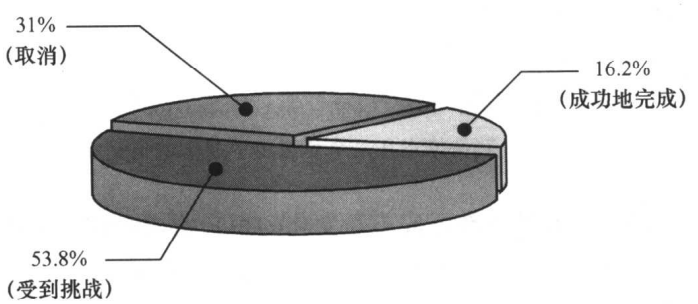

图 1.2　软件项目的成功率

2. 软件存在着错误多、性能低、不可靠、不安全等质量问题

投入极大努力开发出来的软件常常出现人们无法预料的错误，甚至造成严重的后果。1996年6月，Ariane 5火箭在发射37秒之后突然发生爆炸，其错误原因是浮点数转换成整数时发生溢出，系统对此也没有提供相关的异常处理程序。就这样，一个简单的疏漏造成了这支耗资70亿美元的火箭发射失败。另一个例子是在1991年的海湾战争期间，美国对抗伊拉克的飞毛腿导弹曾发生过几次对抗失利，其中一枚导弹在沙特阿拉伯的多哈误击了28名美国士兵，而问题的症结在于导弹的软件存在一个累加计时的故障。一个很小的系统时钟错误积累起来就可能产生14个小时的误差，从而造成跟踪系统失去准确度。今天，随着计算机网络应用的不断普及，计算机病毒的传播、拒绝服务的攻击、网络信息的安全等问题日益突出，软件的质量保证成为人们关注的焦点。

3. 软件成本在计算机系统的整个成本中所占比例越来越大

在过去的40多年里，硬件性能至少跨越了8个重要的阶段，其成本也随着硬件技术的飞速发展而大幅度地下降。但令人遗憾的是，软件开发的能力却未能与硬件的发展保持同步，伴随着软件规模和复杂性的不断增长，软件成本在整个系统成本中所占的比例也持续上升，图1.3显示了软件成本的上升趋势。

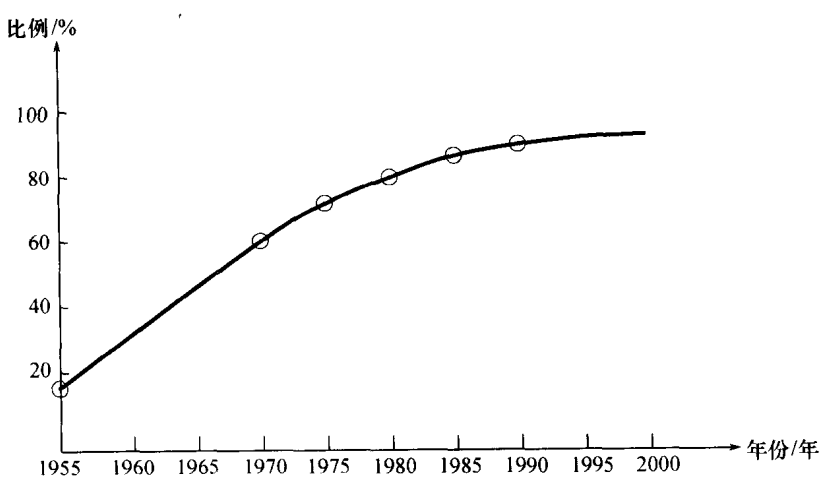

图1.3　软件成本在系统总成本中所占的比例

4. 软件维护极其困难，而且很难适应不断变化的用户需求和使用环境

在软件交付使用的初期，需要识别和纠正软件的错误，改正软件性能上的缺陷，避免实施中的错误使用。即使软件进入了正常的使用期，由于计算机新技术的出现和用户新需求的提出，也需要修改和改进软件。然而，软件维护依然是一件非常困难的工作，常常出现诸如错误难以修改或者修改又带来新的错误等现象，长期不断的修改也引起了软件的退化。

另外，由于软件开发本身的缺陷和软件维护技术的不成熟，软件维护需要消耗大量的工作量，从而造成维护成本相当昂贵。统计数据表明，软件维护费用占软件整个生存周期总费用的55% ~70%。

1.2 软件工程

软件工程是采用工程的概念、原理、技术和方法来开发与维护软件，将经过时间考验而证明正确的管理技术与当前能够得到的最好的技术方法结合起来，其目的在于提高软件的质量与生产率，最终实现软件的工业化生产。

1.2.1 软件工程的概念

1968年10月，NATO科学委员会在德国的加尔密斯(Garmisch，Germany)开会讨论软件可靠性与软件危机的问题，Fritz Bauer首次提出了“软件工程”的概念，他认为：

软件工程是为了经济地获得能够在实际机器上高效运行的可靠软件而建立和使用的一系列好的工程化原则。

后来，人们曾经多次给出了有关软件工程的定义。这里引用《IEEE Standard Glossary of Software Engineering Terminology》给出的一个更为全面的定义：

软件工程是①将系统性的、规范化的、可定量的方法应用于软件的开发、运行和维护，即将工程化应用到软件上；②对①中所述方法的研究。

从上述定义中可以看出，软件工程包括以下两方面的内容：

(1) 软件工程是工程概念在软件领域里的一个特定应用。与其他工程一样，软件工程是在环境不确定和资源受约束的条件下，采用系统性的、规范化的、可定量的方法进行有关原则的实施和应用，这些原则一般是以往经验的积累和提炼，经过时间检验并证明是正确的。因此，软件工程师需要选择和应用适当的理论、方法和工具，同时还要不断探索新的理论和方法解决新的问题。

(2) 软件工程涉及软件产品的所有环节。人们往往偏重于软件开发技术，忽视软件项目管理的重要性。统计数据表明，导致软件项目失败的主要原因几乎与技术和工具没有任何关系，更多的是由于不适当的管理造成的。

1.2.2 软件工程的三要素

软件工程以关注软件质量为目标，由过程、方法和工具三个要素组成，如图1.4所示。

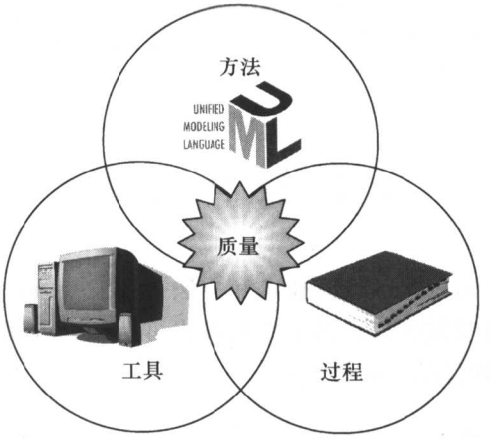

图 1.4　软件工程的三要素

软件工程的方法为软件开发提供了“如何做”的技术，通常包括某种语言或图形的模型表示方法、良好的设计实践以及质量保证标准等，其中使用最广泛的两种方法是传统的软件开发方法和当前流行的面向对象方法。

软件工程的过程是管理和控制产品质量的关键，它定义了技术方法的采用、工程产品（包括模型、文档、数据、报告、表格等）的产生、里程碑的建立、质量的保证和变更的管理，从而将人员、技术、组织与管理有机地结合在一起，实现在规定的时间和预算内开发高质量软件的目标。

软件工具为软件工程方法提供了自动的或半自动的软件支撑环境，辅助软件开发任务的完成。现有的软件工具覆盖了需求分析、系统建模、代码生成、程序调试和软件测试等多个方面，形成了集成化的软件工程开发环境 CASE（Computer Aided Software Engineering，计算机辅助软件工程），以便提高开发效率和软件质量，降低开发成本。

1.2.3　软件质量的特性

软件工程的一个重要目标是“开发出高质量的软件”，那么如何看待“软件质量”的含义呢？简单地说，软件质量是软件产品与明确的和隐含的需求相一致的程度，它通常由一系列的质量特性来描述。例如，除了要求软件正确运行之外，人们可能还希望软件运行的响应时间符合要求、软件使用方便快捷、程序代码易于理解等，而“程序代码易于理解”往往是一种用户没有明确提出的需求，但却是影响软件质量的重要因素。

需要强调的是，软件质量并不取决于开发人员的观点，它通常与客户、用户、维护人员等提出的要求密切相关，图 1.5 显示了他们各自对于软件质量的不同视角。