软件工程与建模

王长元 赵 莉 王淑蓉 编著

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

软件工程与建模

王长元 赵 涛 王淑蓉 编著

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容简介

本书全面系统地讲述了软件工程的概述、可行性研究、需求工程、软件体系结构基础、软件设计、软件编码、软件测试、面向对象技术、软件系统建模、统一建模语言(UML)等内容。与同类软件工程教材的区别在于本教材把软件系统建模的原理与方法、软件体系结构、设计模式有机地引入教学内容过程中。

本书可作为高等院校“软件工程”课程的教材或教学参考书,也可供有一定实践经验的软件工作人员和需要开发应用软件的广大计算机用户阅读参考。

图书在版编目(CIP)数据

软件工程与建模/王长元,赵莉,王淑蓉编著. —西安:西安交通大学出版社,2010.8
ISBN 978-7-5605-3663-7

Ⅰ.①软… Ⅱ.①王… ②赵… ③王… Ⅲ.①软件工程-系统建模 Ⅳ.①TP311.5

中国版本图书馆 CIP 数据核字(2010)第 144702 号

书　　名　软件工程与建模
编　　著　王长元　赵　莉　王淑蓉
责任编辑　李慧娜　桂　亮

出版发行　西安交通大学出版社
　　(西安市兴庆南路 10 号　邮政编码 710049)
网　　址　http://www.xjtupress.com
电　　话　(029)82668357　82667874(发行中心)
　　(029)82668315　82669096(总编办)
传　　真　(029)82668280
印　　刷　陕西江源印刷科技有限公司

开　　本　787mm×1092mm　1/16　印张 17　字数 405 千字
版次印次　2010 年 8 月第 1 版　2010 年 8 月第 1 次印刷
书　　号　ISBN 978-7-5605-3663-7/TP・535
定　　价　35.00 元

读者购书、书店添货、如发现印装质量问题,请与本社发行中心联系、调换。
订购热线:(029)82665248　(029)82665249
投稿热线:(029)82664954
读者信箱:jdlgy@yahoo.cn

前　言

21世纪是信息社会高速发展的时代，软件作为信息技术的核心起着至关重要的作用。面对计算机日益广泛的应用需求，研究如何更快、更好、更经济地开发出相应的软件，是软件开发技术及软件工程师所面临的问题。为摆脱软件危机，软件工程学从60年代末期开始迅速发展起来，现在已成为计算机科学技术的一个重要分支。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，而且大量成功软件总结出的设计经验，使软件开发人员可以充分利用设计模式、框架和组件等。

软件工程是高等学校计算机教学计划中的一门核心课程，所涉及的范围非常广泛，其主要内容包括支持软件开发和维护的理论、方法、技术、标准等。这些内容对软件研制人员、软件项目管理人员都是十分重要的。本书通过对基本概念、基本原理、基本技术、基本方法的讲解，使读者能很快运用软件工程方法和技术开发软件。

由于设计面向对象软件比较困难，设计可复用的面向对象软件就更困难。而设计模式可使人们更加方便地复用成功的设计和体系结构。设计模式帮助设计者作出有利于系统复用的选择，避免设计损害了系统复用性。鉴于此目的，在面向对象部分，我们编写了一些重要的设计模式。

软件系统的开发方法经历了面向过程的开发方法、面向对象的开发方法、基于组件的开发方法和面向服务架构的开发方法。从软件系统开发的方法可以看出抽象思维的重要性。其中抽象角度与抽象层次是软件建模中的关键思想。软件工程通过建模处理复杂性，即通过在任何特定的时间关注本质细节，而忽视次要方面等细节进行建模。在开发中，软件工程师构造许多系统和应用领域的不同模型。在本教材中，对软件建模的要求与目的，建模的方法与建模应用进行了介绍。体现软件工程是一个建模活动的主题思想。

全书共十一章。第1章是对软件工程的概述，第2章到第7章按照传统软件生命周期的过程对可行性研究、需求分析、总体设计、详细设计、编码和测试技术进行了介绍，并在第4章增加了软件体系结构基础的介绍，第8章到第9章是对面向对象概念及面向对象的分析和设计的介绍，第10章介绍软件系统建模，第11章介绍目前的建模标准——UML技术。与同类软件工程教材的区别在于本教材把软件系统建模的原理与方法、软件体系结构、设计模式有机的引入教学过程中。

本书可供计算机专业的本、专科学生作为软件工程课程的教材使用，同时也适合软件开发人员与软件项目管理人员作为技术参考书使用。

本书由王长元、赵莉、王淑蓉、苏小会、姜虹编写，在编写过程中得到了西安工业大学教务处、计算机科学与工程学院的支持和帮助，在此表示诚挚的感谢。由于时间仓促，水平有限，书中难免存在不足之处，恳请广大读者和专家批评指正。

编　者

2010年6月

目 录

第1章　软件工程概述

随着计算机技术、电子技术的发展，计算机与全球互联网络 Internet 相连接，使今天的社会进入了以计算机为核心的信息社会。在信息社会中，信息的获取、处理、交流和决策都需要大量高质量的计算机软件，这样就促使人们对计算机软件的品种、数量、功能、质量、成本和开发时间等提出越来越高的要求。为了使世界上丰富的软件资源为人类共享，人们越来越重视软件、软件开发及运行环境的标准化。计算机的各类程序设计语言和多媒体人机交互工具已被越来越多的人所掌握，成为世界性的文化现象。

遗憾的是，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发展。然而，要想使软件功能越强、使用越方便，开发出来的软件就越复杂、越庞大，人们的软件开发能力越显得力不从心，以致使软件开发计划一拖再拖，成本失去控制，软件质量得不到保证。为了扭转这种被动局面，自 20 世纪 60 年代末期以来，人们十分重视软件开发方法、工具和环境的研究，并在这些领域取得了重要的成果。

本章要求了解软件的发展历史和软件的主要问题，掌握软件工程的基本概念及其要素，熟悉软件工程过程和软件生存周期，理解软件工程的基本目标和原则。正确理解软件工程的基本概念，探讨软件问题的根本原因，领会软件工程思想的作用和意义。

1.1　软件概论

1.1.1　软件的发展历史

随着计算机硬件性能的极大提高和计算机体系结构的不断变化，计算机软件系统更加成熟且更为复杂，从而促使计算机软件的角色发生了巨大的变化，其发展历史大致可以分为如图 1.1 所示的四个阶段。

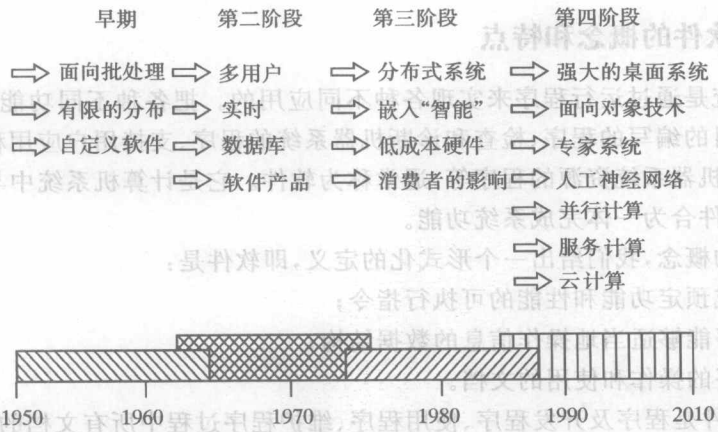

图 1.1　软件的发展阶段

1. 早期阶段

在计算机发展的早期阶段，人们认为计算机的主要用途是快速计算，软件编程简单，不存在什么系统化的方法，开发过程没有任何管理，程序的质量完全依赖于程序员个人的技巧。

2. 第二阶段

计算机软件发展的第二阶段跨越了从60年代中期到70年代末期的十余年，多用户系统引入了人机交互的新概念，实时系统能够从多个源收集、分析和转换数据，从而使得进程的控制和输出的产生是以毫秒而不是分钟来进行，在线存储的发展产生了第一代数据库管理系统。

在这个时期，出现了软件产品和“软件作坊”的概念，设计人员开发程序不再像早期阶段那样只为自己的研究工作需要，而是为了用户更好地使用计算机，人们开始采用“软件工程”的方法来解决“软件危机”问题。

3. 第三阶段

计算机软件发展的第三阶段始于20世纪70年代中期，分布式系统极大地提高了计算机系统的复杂性，网络的发展对软件开发提出了更高的要求，特别是微处理器的出现和广泛应用，孕育了一系列的智能产品。软件开发技术的度量问题受到重视，最著名的有软件工作量估计COCOMO模型、软件过程改进模型CMM等。

4. 第四阶段

计算机软件发展的第四阶段是强大的桌面系统和计算机网络迅速发展的时期，计算机体系结构由中央主机控制方式变为客户机/服务器方式，专家系统和人工智能软件终于走出实验室进入了实际应用，虚拟现实和多媒体系统改变了与最终用户的通讯方式，出现了并行计算和网络计算的研究，面向对象技术在许多领域迅速取代了传统软件开发方法，人们现在又向高层次迈进，开始研究服务计算与云计算的关键技术和方法。

在软件的发展过程中，软件从个性化的程序变为工程化的产品，人们对软件的看法发生了根本性的变化，从“软件＝程序”发展为“软件＝程序＋数据＋文档”及软件变服务。软件的需求成为软件发展的动力，软件的开发从自给自足模式发展为可以在市场中流通以满足广大用户的需要。软件工作的考虑范围也发生了很大变化，人们不再只顾及程序的编写，而是涉及到软件的整个生命周期。

1.1.2 软件的概念和特点

计算机系统是通过运行程序来实现各种不同应用的。把各种不同功能的程序，包括用户为自己的特定目的编写的程序、检查和诊断机器系统的程序、支持用户应用程序运行的系统程序、管理和控制机器系统资源的程序等，通常称为软件。它是计算机系统中与硬件相互依存的另一部分，与硬件合为一体完成系统功能。

关于软件的概念，我们给出一个形式化的定义，即软件是：

- 能够完成预定功能和性能的可执行指令；
- 使得程序能够适当地操作信息的数据结构；
- 描述程序的操作和使用的文档。

简言之，软件是程序及开发程序、使用程序、维护程序过程中所有文档的集合。

然而，真正理解软件的含义需要了解软件的特点，从而明白软件与人类建造的其他事物的

区别。与硬件相比，软件具有以下不同的特点。

①软件是逻辑的，而不是物理的产品。逻辑往往实际只存在于人的头脑当中，软件人员好比“皇帝的新衣”故事中的裁缝，软件的开发过程极难加以控制。

②软件是由开发或工程化形成的，没有明显的制造过程。软件成本集中于“开”上，意味着软件项目不能像硬件制造项目那样来管理。

③如图 1.2 和 1.3 所示，软件在运行和使用期间，不存在硬件那样的磨损和老化问题，但它存在退化问题，开发人员必须维护软件。

图 1.2 是硬件故障率的变化曲线，即硬件在生命初期具有较高的故障率，这些故障主要是由于设计或制造的缺陷造成的。当这些缺陷修正后，故障率在一段时期内会降低到一个稳定的曲线上。随着时间的推移，硬件构件由于种种原因受到不同程度的损害，故障率又升高了。也就是说，硬件已经开始磨损了。

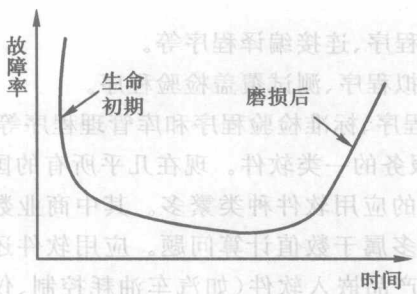

图 1.2　硬件的故障率曲线

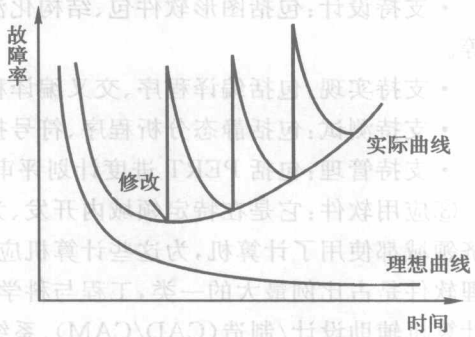

图 1.3　软件的故障率曲线

图 1.3 是软件的故障率曲线，在软件的生命初期隐藏的错误会使程序具有较高的故障率，理想的情况下当这些错误改正后曲线便趋于平稳，但实际情况是随着这些修改有可能引入新的错误，从而使故障率曲线呈现图中所示的锯齿状。于是，软件的退化由于修改而发生了。

④大多数软件是自定的，而不是通过已有构件组装而成的。迄今为止，软件的开发尚未完全摆脱手工的方式。

⑤软件成本相当昂贵。IBM360 操作系统的研制人员最多时可达 1000 多人，从 1963 年到 1966 年共花了 4 年时间才完成，总计耗费了 5000 多人/年，以后又进行了不断的修改和补充。该系统的整个研制费用为 5 亿美元，其中近一半花在软件上。

⑥软件本身是复杂的。软件比任何其他人类制造的结构更复杂，甚至硬件的复杂性和软件相比也是微不足道的。软件本质上的复杂性是软件产品难以理解，影响软件过程的管理，并使维护过程十分复杂。

1.1.3　软件的分类

人们在工作和学习中，经常接触到各式各样的软件。那么，这些数量众多的软件究竟分为哪些类型，这就要考虑对计算机软件进行分类的依据。但事实上，要给出一个科学的、统一的、严格的计算机软件分类标准是不现实的。但对软件的类型进行必要的划分，对于根据不同类型的工程对象采用不同的开发和维护方法是很有价值的。因此，有必要从不同角度对计算机

软件做适当的分类。

1. 基于软件功能的划分

①系统软件：它是与计算机硬件紧密配合以使计算机各个部件与相关软件及数据协调、高效工作的软件。例如，操作系统、数据库管理系统等。系统软件在工作时频繁地与硬件交往以便为用户服务，共享系统资源，在这中间伴随着复杂的进程管理和复杂的数据结构的处理。系统软件是计算机系统必不可少的重要组成部分。

②支撑软件：它是协助用户开发软件的工具性软件，包括帮助程序人员开发软件产品的工具和帮助管理人员控制开发进程的工具。可分为以下几类。

· 一般类型：包括文本编辑程序、文件格式化程序、程序库系统等。

· 支持需求分析：包括 PSL/PSA 问题描述语言、问题描述分析器、关系数据库系统、一致性检验程序等。

· 支持设计：包括图形软件包、结构化流程图绘图程序、设计分析程序、程序结构图编辑程序等。

· 支持实现：包括编译程序、交叉编译程序、预编译程序、连接编译程序等。

· 支持测试：包括静态分析程序、符号执行程序、模拟程序、测试覆盖检验程序。

· 支持管理：包括 PERT 进度计划评审方法、绘图程序、标准检验程序和库管理程序等。

③应用软件：它是在特定领域内开发、为特定目的服务的一类软件。现在几乎所有的国民经济领域都使用了计算机，为这些计算机应用领域服务的应用软件种类繁多。其中商业数据处理软件是占比例最大的一类，工程与科学计算软件大多属于数值计算问题。应用软件还包括计算机辅助设计/制造(CAD/CAM)、系统仿真、智能产品嵌入软件(如汽车油耗控制、仪表盘数字显示、刹车系统)，以及人工智能软件(如专家系统、模式识别)等，此外，在事务管理、办公自动化、中文信息处理、计算机辅助教学(CAI)等方面的软件也得到迅速发展，产生了惊人的生产效率和巨大的经济效益。

2. 基于软件运作方式的划分

①实时处理软件：指在事件或数据产生时，立即处理，并及时反馈信号，控制需要监测和控制过程的软件。主要包括数据采集、分析、输出三部分，其处理时间是严格限定的，如果在任何时间超出了这一限制，都将造成事故。

②分时软件：允许多个联机用户同时使用计算机的软件。系统把处理机时间轮流分配给各联机用户，使各用户都感到只有自己在使用计算机。

③交互式软件：能实现人机通信的软件。这类软件接收用户给出的信息，但在时间上没有严格的限定，这种工作方式给予用户很大的灵活性。

④批处理软件：把一组输入作业或一批数据以成批处理的方式一次运行即按顺序逐个处理的软件。

3. 基于软件规模的划分

根据开发软件所需的人力、时间以及完成的源程序行数，可划分为下述六种不同规模的软件。

①微型软件：指一个人在几天之内完成的、程序不超过 500 行语句且仅供个人专用的软件。通常这类软件没有必要做严格的分析，也不必要有完整的设计、测试资料。

②小型软件：指一个人在半年之内完成的 2000 行以内的程序。这种程序通常没有与其他程序的接口，但需要按一定的标准化技术、正规的资料书写以及定期的系统审查，只是没有大型软件那样严格。

③中型软件：指 5 个人以内在一年多时间里完成的 5000 到 50000 行的程序。中型软件开始出现了软件人员之间、软件人员与用户之间的联系和协调的配合关系问题，因而计划、资料书写以及技术审查需要比较严格地进行。在开发中使用系统的软件工程方法是完全必要的，这对提高软件产品质量和程序开发人员的工作效率起着重要的作用。

④大型软件：指 5～10 个人在两年多的时间里完成的 5 万～10 万行的程序。参加工作的软件人员需要进行二级管理。对于这样规模的软件，采用统一的标准、实行严格的审查是绝对必要的。由于软件的规模庞大以及问题的复杂性，往往在开发过程中会出现一些事先难以估计的不测事件。

⑤甚大型软件：指 100～1000 人在四五年时间里完成的具有 100 万行的程序。这种甚大型项目可能会划分成若干个子项目，每一个子项目都是一个大型软件。子项目之间具有复杂的接口。例如，实时处理系统、远程通信系统、多任务系统、大型操作系统、大型数据库管理系统。很显然，如果这类问题没有软件工程方法的支持，它的开发工作是不可想象的。

⑥极大型软件：指 2000～5000 人在 10 年内完成的具有 1000 万行以内的程序。这类软件很少见，往往用于军事指挥、弹道导弹防御系统等。可以看出，规模大、时间长、多人参加的软件项目，其开发工作必须要有软件工程的知识作指导，而规模小、时间短、参加人员少的软件项目的开发也得遵循软件工程的概念和开发规范，其基本原则是一样的。

4. 基于软件失效的影响进行划分

工作在不同领域的软件，在运行中对可靠性也有不同的要求。事实上，随着计算机不断进入国民经济各个重要领域，软件的可靠性越来越重要。人们一般称这类软件为关键软件，其特点在于：

①可靠性质量要求高。

②常与完成重要功能的大系统的处理部件相连。

③含有的程序可能对人员、公众、设备或设施的安全造成影响，还可能影响到环境的质量和国家的安全。

5. 基于软件服务对象的范围进行划分

软件工程项目完成后可以有两种结果：

①定制软件，是受某个特定客户（或少数客户）的委托，由一个或多个软件开发机构在合同的约束下开发出来的软件。

②产品软件，是由软件开发机构开发出来直接提供给市场，或是为众多用户服务的软件。

1.1.4　软件危机

1. 软件危机的表现

软件危机爆发于 20 世纪 60 年代末期，虽然人们一直致力于发现解决危机的方法，但是软件危机至今依然困扰着我们，并没有一种灵丹妙药可以完全治愈这种病痛。

软件危机的具体表现如下。

①软件开发的进度难以控制，经常出现经费超预算、完成期限一再拖延的现象。

1979 年，美国 US Government Accounting Office 对政府项目进行了调查，其中 9 个软件项目的结果如下：

结果	经费(百万美元)	比例(%)
付了钱，但从不交付	2.0	28.8
交付了，但无法顺利使用	3.2	47.3
大部分重做或放弃后，才使用	1.3	19.2
经过修改后使用	0.2	3
交付后直接就能使用	0.1	1.77

一个复杂的软件系统需要建立庞大的逻辑体系，而这些往往只存在于人们的头脑中，正如一个大项目负责人所说："软件人员太像皇帝新衣故事中的裁缝，当我来检查软件开发工作时，所得到的回答好像对我说：我们正忙于编织这件带有魔法的织物，只要一会儿，你就会看到这件织物是极其美丽的。但是我什么也看不到，什么也摸不到，也说不出任何一个有关的数字，没有任何办法得到一些信息说明事情确实进行得非常顺利，而且我已经知道许多人最终编织了一大堆昂贵的废物，还有不少人最终什么也没有做出来。"

②软件需求在开发初期不明确，导致矛盾在后期集中暴露，从而对整个开发过程带来灾难性的后果。

软件需求的缺陷将给项目成功带来极大风险，如产品的成本过高、产品的功能和质量无法完全满足用户的期望等等。即使一个项目团队的人员和配备都很不错，但不重视需求过程也会付出惨痛的代价。导致需求缺陷的主要原因包括需求的沟通与理解、需求的变化与控制、需求说明的明确与完整。模棱两可的需求所带来的后果便是返工或重做。一些自认为已做好的事情，返工会耗费开发总费用的 40%，而 70%～85%的重做是由于需求方面的错误引起的。

③由于缺乏完整规范的资料，加之软件测试不充分，从而造成软件质量低下，运行中出现大量问题。

在 1985 年到 1987 年之间，至少有两个病人死于 Therac-25 医疗线性加速器的过量辐射，其原因是控制软件中的一个故障。1965 年至 1970 年，美国范登堡基地发射火箭多次失败，绝大部分出于控制系统的故障，一个小小的疏漏往往会造成上千万美元的损失。由此可见，软件错误的后果是十分严重的，医疗软件的错误可能造成病人的死亡，银行系统的错误会使金融混乱，航管系统的错误会造成飞机失事，等等。

2. 危机的原因

从软件危机的种种表现和软件作为逻辑产品的特殊性可以发现软件危机的原因。

①用户对软件需求的描述不精确，可能有遗漏、二义性或错误，甚至在软件开发过程中，用户还提出修改软件功能、界面、支撑环境等方面的要求。

②软件开发人员对用户需求的理解与用户的本来愿望有差异，这种差异必然导致开发出来的软件产品与用户要求不一致。

③大型软件项目需要组织一定的人力共同完成，多数管理人员缺乏开发大型软件系统的

经验，而多数软件开发人员又缺乏管理方面的经验。各类人员的信息交流不及时、不准确、有时还会产生误解。

④软件项目开发人员不能有效地、独立自主地处理大型软件的全部关系和各个分支，因此容易产生疏漏和错误。

⑤缺乏有力的方法和工具方面的支持，过分地依靠程序人员在软件开发过程中的技巧和创造性，加剧软件产品的个性化。

⑥软件产品的特殊性和人智力的局限性，导致人们无力处理“复杂问题”。所谓“复杂问题”的概念是相对的，一旦人们采用先进的组织形式、开发方法和工具提高了软件的开发效率和能力，新的、更大的、更复杂的问题又摆在人们面前。

3. 克服危机的途径

由于认识到软件的设计、实现、维护和传统的工程规则有相同的基础，于是北大西洋公约组织(NATO)于 1967 年首次提出了“软件工程(software engineering)”的概念。关于编制软件与其他工程任务类似的提法，得到了 1968 年在德国召开的 NATO 软件工程会议的认可。委员会的结论是：软件工程应使用已有的工程规则的理论和模式，来解决所谓的“软件危机”。

软件危机至今仍然困扰着我们，这表明软件生产过程在许多方面与传统的工程相似，但却具有独特的属性和问题。

1.2　软件工程与软件过程

软件工程的根本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。在“软件工程”的概念提出后的几十年里，各种有关软件的技术、思想、方法和概念层出不穷，典型的包括结构化的方法、面向对象方法、软件开发模型和软件开发过程等，软件工程逐步发展为一门独立的科学。

1.2.1　软件工程的概念

Fritz Bauer 曾在 NATO 会议上给出软件工程的定义：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。

1983 年，IEEE(institute of electrical & electronic engineers，电气与电子工程师协会)给出了一个更为全面的定义：软件工程是研究和应用如何以系统化的、规范的、可度量的方法去开发、运行和维护软件，即把工程化应用到软件上。

Bauer 的定义给出了软件工程的基线，但忽略了软件质量的技术方面和软件度量的重要性。除以上定义外，软件工程还有许多其他的定义，但其基本思想都是强调在软件开发过程中应用工程化原则，解决软件的整体质量较低、最后期限和费用没有保证等问题。

软件工程是一种层次化的技术，如图 1.4 所示，其中过程、方法和工具是软件工程的三个要素。

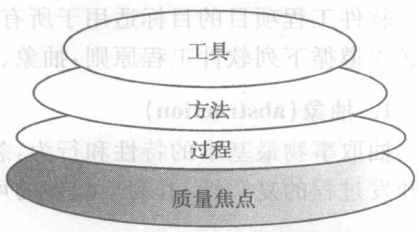

图 1.4　软件工程的层次

①软件工程必须以有组织的质量保证为基础，