21世纪高等学校计算机规划教材

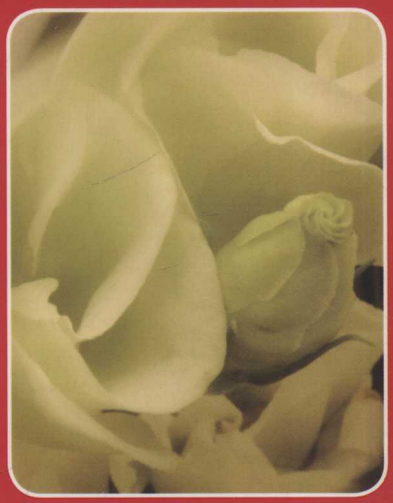
21st Century University Planned Textbooks of Computer Science

软件工程——理论与实践

Theory and Practice of Software Engineering

吕云翔 王昕鹏 邱玉龙 编著

- 完善的理论和充分实践的有机结合
- 实际的案例贯穿软件开发的各过程
- 丰富的软件工程课程设计可选题目



精品系列

人民邮电出版社
POSTS & TELECOM PRESS

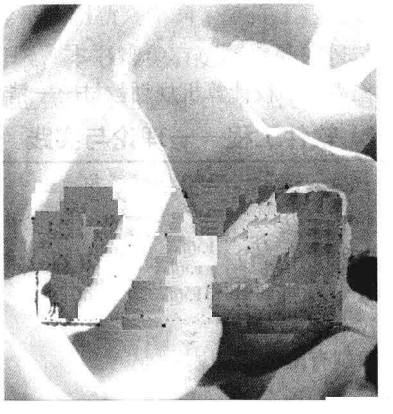
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

软件工程——理论与实践

Theory and Practice of Software Engineering

吕云翔 王昕鹏 邱玉龙 编著



精品系列

人民邮电出版社

北京

图书在版编目（CIP）数据

软件工程 ：理论与实践 / 吕云翔，王昕鹏，邱玉龙编著. -- 北京 ：人民邮电出版社，2012.8
21世纪高等学校计算机规划教材. 精品系列
ISBN 978-7-115-28228-6

Ⅰ. ①软… Ⅱ. ①吕… ②王… ③邱… Ⅲ. ①软件工程－高等学校－教材 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字(2012)第106704号

内 容 提 要

本书从结构化方法和面向对象方法两方面介绍软件工程的基本概念、原理和方法，并用一个案例贯穿每一章的实践部分，让读者在认识软件工程原理的基础上，能进一步利用相关的工具对所学内容进行实践，从而实际掌握进行软件开发的各种技能。本书理论与实践相结合，内容翔实，可操作性强。

本书是高等院校计算机科学、软件工程及相关专业“软件工程”课程的理想教材。

21 世纪高等学校计算机规划教材——精品系列

软件工程——理论与实践

◆ 编　　著　吕云翔　王昕鹏　邱玉龙
责任编辑　武恩玉

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：18.5　　2012 年 8 月第 1 版
字数：489 千字　　2012 年 8 月北京第 1 次印刷

ISBN 978-7-115-28228-6

定价：36.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

软件工程是应用计算机科学技术、数学、管理学的原理，运用工程科学的理论、方法和技术，研究和指导软件开发和演化的一门交叉学科。随着科技的发展，软件工程已成为计算机科学及其相关专业的一门重要的必修课。软件工程的教学目的在于使学生掌握其基本概念和原理，培养学生使用工程化的方法高效开发高质量软件的能力以及进行项目管理的能力。

软件工程是一门理论与实践并重的课程。本书在讲述软件工程的基本概念、原理和方法的基础上，详细而全面地介绍可以实际用于软件开发实践的各种技能，旨在使学生通过有限课时的学习后，不仅能对软件工程的原理有所了解，而且能具备实际开发软件的各种技能，比如熟练使用各种软件工程工具，按照标准和规范编写文档等。

本书共分为7章，内容涉及软件工程的基本原理和概念、软件开发生命周期的各个阶段、项目管理的相关内容、使用各种自动化工具来辅助软件开发的过程以及课程设计和一些课程设计题目。每章分为理论部分和实践部分。理论部分从理论学习的角度阐述软件工程的基本概念、原理和方法，在内容的安排上详略得当，使读者在有限的时间内能领会软件工程的精髓。实践部分配合理论部分的学习内容，指导读者利用相关的工具对所学内容进行运用。理论与实践的紧密结合，不仅有利于读者巩固和掌握知识，还能提高实践能力。此外，本书使用一个案例（网上书店系统）贯穿于各章的实践部分，使读者能通过实例对软件开发过程有一个系统的了解。

本书总结了我们多年软件工程教学与实践的经验，在编写的过程中得到了王明华、张航、陈捷、于承东和曹科的支持，在此对他们表示感谢。

由于软件工程是一门新兴学科，软件工程的教学方法本身还在探索之中，加之我们的水平和能力有限，本书难免有疏漏之处，恳请各位同仁和广大读者批评指正，也希望各位能将实践过程中的经验和心得与我们交流（邮箱：yunxianglu@hotmail.com）。

吕云翔　王昕鹏　邱玉龙
2012年4月于北京航空航天大学软件学院

目 录

第 1 章 软件工程概述

本章目标

- 了解软件的概念，理解软件的特点。
- 了解软件危机产生的原因和表现，明确软件工程的重要性。
- 掌握软件工程的基本内容。
- 掌握软件生命周期的概念。
- 掌握各种软件开发模型的特征及适用范围。
- 熟悉与软件开发项目相关的常用自动化工具。
- 掌握 Microsoft Office Visio 的功能及使用方法。
- 掌握 Rational Rose 的功能及使用方法。

1.1 软　　件

1.1.1 软件的概念

软件是计算机系统的“思维中枢”，在计算机系统中起着举足轻重的作用。它与计算机硬件相互作用，相互配合，从而实现了特定的系统功能。计算机软件的概念是随着计算机技术的发展而发展的。

在计算机发展初期，软件就是指程序，即计算机可以识别的源代码或机器可直接执行的代码。当时，软件的作用并没有得到足够的重视。使用者一般需要直接操纵计算机硬件，程序是为某个特定问题而专门设计的。

随着计算机技术的发展，人们越来越充分认识到高质量的软件会使计算机系统的功能和效率大大地提高。高质量、多功能的软件使得计算机的应用从单一的科学计算扩展到多个领域，比如数据处理、实时控制等。随着计算机应用的日益普及，软件日益复杂，规模日益增大，人们开始意识到软件并不仅仅等于程序。

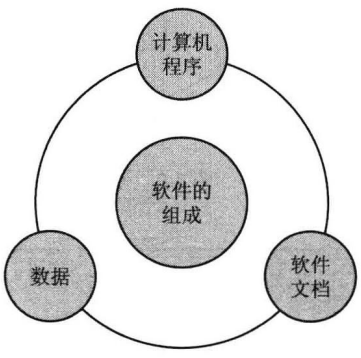


图 1-1　软件的组成

全面地讲，软件由图 1-1 所示的 3 部分组成。具体包括：

（1）计算机程序，即人们为了完成特定的功能而编制的一组指令集。

（2）数据，即程序能处理的具有一定数据结构的信息。

（3）软件文档，它是与程序的开发、维护和使用有关的图文资料，如软件开发计划书、需求规格说明书、设计说明书、测试分析报告和用户手册等。

1.1.2 软件的特点

任何事物都有自己的特点，这是区别于其他事物的根本。理解事物的特点有利于人们更加深刻、更加准确地认识事物的本质。作为计算机系统的重要组成部分，计算机软件的功能依赖于计算机硬件的支持。与硬件相比，计算机软件具有以下特点。

（1）计算机硬件是实物产品，是有形的设备，具有明显的可见性。但是，人们却无法直接观察计算机软件的物理形态，只能通过观察它的实际运行情况来了解它的功能、特性和质量等。

（2）人们在分析、设计、开发、测试软件产品以及在软件开发项目的管理过程中，渗透了大量的脑力劳动。可以说，人类的逻辑思维、智能活动和技术水平是生产软件产品的关键。而传统意义上的硬件制造，除了人类的脑力劳动外，还需要大量的体力劳动。

> 程序员，就像诗人一样，几乎是仅仅工作在单纯的思考中。
>
> ——《人月神话》

（3）在计算机系统运行的过程中，计算机硬件存在着磨损和老化的现象，这也是一切物理器件都存在的普遍现象。但是，硬件设备磨损后，人们可以简单地用另一个硬件设备替换。对于计算机软件而言，不存在像硬件一样的磨损和老化现象，因为它不会受到引起硬件磨损的环境因素（比如温度、振动、灰尘和阳光等）的影响。但是软件却存在着缺陷维护和技术更新的问题。人们在对软件的缺陷进行维护或者对技术进行更新的时候，往往要对软件的设计和编码进行改动，这个过程会比简单的硬件替换复杂得多。如图 1-2 和图 1-3 所示分别展示了硬件的失效率与使用时间的关系以及软件的失效率与使用时间的关系。

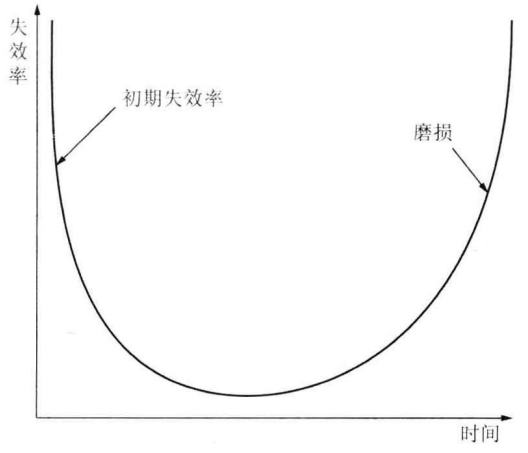


图 1-2　硬件失效曲线图

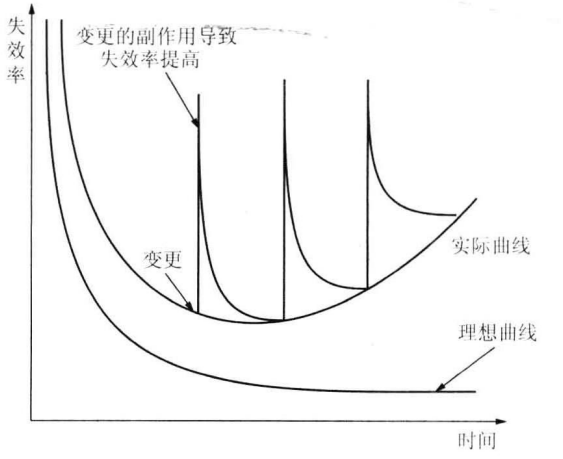


图 1-3　软件失效曲线图

> 计算机系统的硬件维护包括了 3 项活动：更换损坏的器件，清洁和润滑及修改设计上的缺陷。软件维护不包括清洁、润滑和对损坏器件的修复，它主要包含对设计缺陷的修复。
>
> ——《人月神话》

（4）软件的开发和运行必须依赖于特定的计算机系统环境，比如硬件、网络配置和支撑软件等。软件对运行环境的这种依赖性是一般产品所没有的。为了减少这种依赖性，在软件开发的过程中提出了软件的可移植性。

（5）软件具有可复用性。软件一旦被开发出来，便可以很容易地被大量复制，从而形成多个副本，而硬件产品必须经过完整的生产周期才能得到。

1.1.3　软件的分类

随着计算机软件复杂性的增加，在某种程度上很难对软件给出一个通用的分类，但是可以从不同的角度，按照特定的方法对软件进行归类。

按照功能的不同，可以把软件划分为*系统软件*、*支撑软件*和*应用软件*。

（1）*系统软件*是与计算机硬件结合最紧密的软件，它在计算机系统中必不可少，可以协调各个物理部件的工作，同时服务于其他上层软件。操作系统就是最典型的系统软件，它负责管理系统的资源，并为上层软件的运行提供了必备的接口和条件。

（2）*支撑软件*是工具性软件，它一方面可以协调用户进行软件开发，另一方面还能对应用软件进行维护。我们常用的文本编辑器、绘图软件、数据库管理系统和 CASE 工具系统等都属于支撑软件。

（3）*应用软件*是为特定的领域或服务开发的针对性较强的软件。它的种类极其繁多，应用范围最为广泛，是直接服务于用户的软件。比如，地理信息系统软件、航空售票软件、教务管理系统软件和信息管理系统等。

按照软件服务对象的不同，软件还可以分为*通用软件*和*定制软件*。

（1）*通用软件*是由特定的软件开发机构开发，面向市场公开销售的独立运行的软件系统，如操作系统、文档处理系统和图片处理系统等。

（2）*定制软件*通常是面向特定的用户需求，由软件开发机构在合同的约束下开发的软件，如为企业定制的办公系统、交通管理系统和飞机导航系统等。

按照软件产品规模的不同，计算机软件还可以划分为*小型软件*、*中型软件*和*大型软件*。按照工作方式的不同，计算机软件还可以划分为*实时软件*、*分时软件*、*交互式软件*和*批处理软件*。

软件的分类示意图如图 1-4 所示。

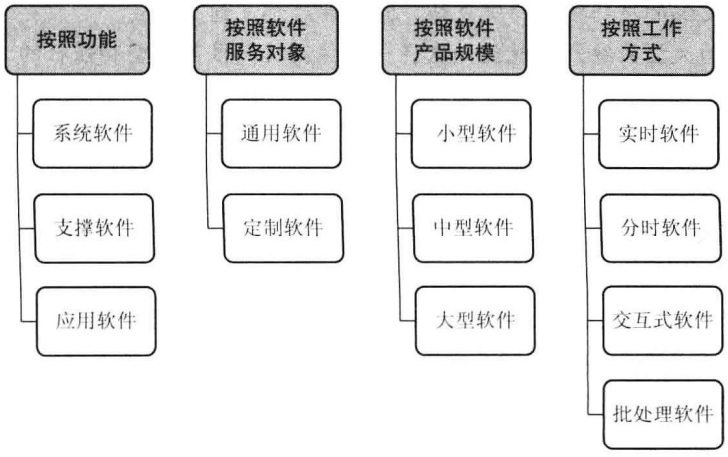


图 1-4　软件的分类

1.2 软件危机

1.2.1 软件危机的表现

软件危机是指落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现一系列严重问题的现象。软件危机爆发于 20 世纪 60 年代中期，随着软件规模的扩大、复杂性的增加及功能的增强，高质量的软件开发变得越来越困难。在软件开发的过程中经常会出现不能按时完成任务，产品质量得不到保证，工作效率低下和开发经费严重超支等情况。这些情况逐渐使人们意识到软件危机的存在及其重要性。具体来讲，软件危机的主要表现如下。

（1）开发人员开发的软件产品不能完全满足用户的需求，即产品的功能或特性与需求不符。这主要是因为开发人员与用户之间的交流不够充分，使得开发人员理解的用户需求与实际的用户需求之间存在差异。

（2）软件产品的质量难以得到保证。开发团队缺少完善的软件质量评审体系以及科学的软件测试规程，使得最终的软件产品存在诸多缺陷。

（3）软件产品的开发周期、开发经费和维护费用很难被准确地估计，从而给项目的管理带来很多麻烦。很多情况下，软件产品的开发周期或经费会大大超出预期。

（4）随着技术的更新和用户需求的扩大，已有的软件产品往往不能灵活地适应环境的改变。再加上软件生产观念的落后和高素质软件开发人员的匮乏等因素，软件产品的可维护性、可扩展性和可复用性往往不能满足市场的要求。

（5）软件文档不完备，并且存在文档内容与软件产品不符的情况。软件文档是计算机软件的重要组成部分，它为在软件开发人员之间以及开发人员与用户之间信息的共享提供了重要的平台。软件文档的不完整和不一致会给软件的开发和维护等工作带来很多麻烦。

> 在史前史中，没有别的场景比巨兽们在焦油坑中垂死挣扎的场面更令人震撼。上帝见证着恐龙、猛犸象、剑齿虎在焦油中挣扎。它们挣扎得越猛烈，焦油纠缠得就越紧。没有任何猛兽足够强壮或具有足够的技巧能够挣脱束缚，它们最后都沉到了坑底。
>
> 过去几十年的大型系统开发就犹如这样一个焦油坑，很多大型和强壮的“动物”在其中剧烈地挣扎。他们中大多数开发出了可运行的系统——但只有极少数项目满足了目标、进度和预算的要求。各种团队，大型的和小型的，庞杂的和精干的，一个接一个地淹没在了“焦油坑”中。表面上看起来好像没有哪一个问题会导致开发困难，每个问题都能得到解决，但是当它们互相纠缠和累积在一起的时候，团队的行动就变得越来越慢。对问题的麻烦程度，每个人似乎都会感到惊讶，却很难看清问题的本质。
>
> ——《人月神话》

软件开发过程中出现的这些问题严重影响了软件产业健康快速地发展。为了形象地描述软件危机，OS 360 经常被作为一个典型的案例。这是一个超大型的软件项目，动用了近千名程序员。经历了数十年的开发之后，极度复杂的软件项目甚至产生了一套不包括在原始设计方案之中的工作系统。Frederic kP. Brooks 是这个项目的管理者，他在自己的著作《人月神话》中曾经承认，自

己犯了一个价值数百万美元的错误。

Frederick P. Brooks, Jr.

Frederick P. Brooks 是 1999 年的图灵奖得主。他在 29 岁时就主持和领导了被称为人类从原子能时代进入信息时代标志的 IBM/360 系列计算机的开发工作，取得辉煌成功，从而名噪一时，与 Bob Evans 并称为“IBM/360 之父”。IBM/360 成功以后，他离开 IBM 回到其故乡，为北卡罗来纳大学（UNC）创建了计算机科学系，担任该系系主任长达 20 年（1964～1984 年）。卸任以后仍在该系任教至今。

他作为硬件和软件的双重专家和出色的教育家，始终活跃在计算机舞台上，在计算机技术的诸多领域中都做出了巨大的贡献。1987 年，Brooks 当选为美国工程院院士，他同时也是英国皇家学会和荷兰皇家科学与艺术院的外籍院士。

Brooks 的著作不多，但影响都很大，最为著名的是 1975 年出版的《人月神话》（The Mythical Man-Month: Essays on Software Engineering）。本书是他领导 IBM/360 软件开发经验的结晶，内容丰富而生动，堪称软件工程方面的经典之作。

1.2.2　软件危机的原因

软件危机的出现及其日益严重的趋势充分暴露了软件产业在早期发展过程中存在的各种各样的问题。可以说，人们对软件产品认识的不足以及对软件开发的内在规律的理解偏差是软件危机出现的本质原因。具体来说，软件危机出现的原因可以概括为以下几点。

（1）软件开发是一项复杂的工程，需要用科学的工程化的思想来组织和指导软件开发的各个阶段。而这种工程学的视角正是很多软件开发人员所没有的，他们往往简单地认为软件开发就是程序设计。

（2）没有完善的质量保证体系。建立完善的质量保证体系需要有严格的评审制度，同时还需要有科学的软件测试技术及质量维护技术。软件的质量得不到保证，开发出来的软件产品往往不能满足需求，同时还可能需要花费大量的时间、资金和精力去修复软件的缺陷，从而导致软件质量下降和开发预算超支等后果。

（3）软件文档的重要性没有得到软件开发人员和用户的足够重视。软件文档是软件开发团队成员之间交流和沟通的重要平台，也是软件开发项目管理的重要工具。如果不能充分重视软件文档的价值，势必会给软件开发带来很多不便。

（4）从事软件开发的专业人员对这个产业认识不充分，缺乏经验。软件产业相对于其他工业产业而言是一个比较年轻、发展不太成熟的产业，人们对它的认识缺乏深刻性。

（5）软件独有的特点也给软件的开发和维护带来困难。软件的抽象性和复杂性使得软件在开发之前很难对开发过程的进展进行估计。再加上软件错误的隐蔽性和改正错误的复杂性，都使得软件开发和维护在客观上比较困难。

为了解决软件危机，人们逐渐认识了软件的特性以及软件产品开发的内在规律，并尝试用工程化的思想去指导软件开发，于是软件工程诞生了。

1.3 软 件 工 程

1.3.1 软件工程的概念

1968 年，在北大西洋公约组织举行的一次学术会议上，该组织的科学委员们在开会讨论软件的可靠性与软件危机的问题时，首次提出了“软件工程”的概念，并将其定义为“为了经济地获得可靠的和能在实际机器上高效运行的软件而建立和使用的健全的工程规则”。这个定义肯定了工程化的思想在软件工程中的重要性，但是并没有提到软件产品的特殊性。

经过 40 多年的发展，软件工程已经成为一门独立的学科，人们对软件工程也逐渐有了更全面、更科学的认识。在现代，软件工程是指应用计算机科学技术、数学和管理学的原理，运用工程科学的理论、方法和技术，研究和指导软件开发和演化的一门交叉学科。它强调按照软件产品的特殊性质，采用工程化的思想来指导软件开发，在高效的软件生产和科学的项目管理的基础上得到高质量的软件产品。

可以说，软件工程的提出是为了解决软件危机所带来的各种弊端。具体地讲，软件工程的目标主要包括以下几点：

（1）使软件开发的成本能够控制在预计的合理范围内；

（2）使软件产品的各项功能和性能能够满足用户需求；

（3）提高软件产品的质量；

（4）提高软件产品的可靠性；

（5）使生产出来的软件产品易于移植、维护、升级和使用；

（6）使软件产品的开发周期能够控制在预计的合理时间范围内。

> 开发人员交付的是用户满意度，而不仅仅是有形的产品。
>
> ——Cosgrove

实际上，我们可以把上述各个目标概括为开发正确、可用和经济的软件产品。当然，在实际的软件开发过程中，软件开发团队很难同时兼顾所有的目标。通常，人们会根据实际项目的情况，对各个目标做优先级排序。

1.3.2 软件工程的基本内容

相对于其他学科而言，软件工程是一门比较年轻的学科，它的思想体系和理论基础还有待进一步修正和完善。软件工程学科包含的内容有软件工程原理、软件工程过程、软件工程方法、软件工程模型、软件工程管理、软件工程度量、软件工程环境和软件工程应用等，如图 1-5 所示。

软件工程学科							
软件工程原理	软件工程过程	软件工程方法	软件工程模型	软件工程管理	软件工程度量	软件工程环境	软件工程应用

图 1-5 软件工程学科

软件工程原理就是指软件工程学科在发展过程中所遵循的基本原理和普遍规律。实际的软件开发项目只有在一定的软件工程原理的约束下，才能够有效地贯彻软件工程的思想。经过长期的开发实践和理论研究，著名软件工程专家 B. W. Boehm 提出了以下几

项软件工程的基本原则。

（1）将软件的生命周期划分为多个阶段，对各个阶段实行严格的项目管理。软件开发是一个漫长的过程，人们可以根据工作的特点或目标，把整个软件的开发周期划分为多个阶段，并为每个阶段制定分阶段的计划及验收标准，这样有利于对整个软件开发过程进行管理。在传统的软件工程中，软件开发的生命周期可以划分为可行性研究、需求分析、软件设计、软件实现、软件测试、产品验收和交付等阶段。

（2）坚持阶段评审制度，以确保软件产品的质量。严格地贯彻与实施阶段评审制度可以帮助软件开发人员及时地发现并改正错误。在软件开发的过程中，错误发现得越晚，修复错误所要付出的代价就会越大。实施阶段评审，只有在本阶段的工作通过评审后，才能进入下一阶段的工作。

（3）实施严格的产品控制，以适应软件规格的变更。在软件开发的过程中，用户需求很可能不断发生变化。有些时候，即使用户需求没有改变，软件开发人员受到经验的限制以及与用户交流不充分的影响，也很难做到一次性获得全部的、正确的需求。可见，需求分析工作应该贯穿整个软件开发的生命周期。在软件开发的整个过程中，需求的改变是不可避免的。当需求变更时，为了保证软件各个配置项的一致性，实施严格的版本控制是非常有必要的。

> 不变只是愿望，变化才是永恒。
>
> ——《人月神话》

（4）采用现代程序设计技术。这是提高软件开发和维护效率的关键。现代的程序设计技术，比如面向对象，可以使开发出来的软件产品更易于维护和修改，同时还能缩短开发的时间，并且更符合人们的思维逻辑。

（5）开发出来的软件产品应该能够清楚地被审查。虽然软件产品的可见性比较差，但是它的功能和质量应该能够被准确地审查和度量，从而有利于有效的项目管理。一般软件产品包括可以执行的源代码、一系列相应的文档和资源数据等。

（6）合理地安排软件开发小组的人员，并且开发小组的人员要少而精。开发小组人员的数量少，有利于组内成员充分的交流，这是高效团队管理的重要因素。而高素质的开发小组成员是影响软件产品的质量和开发效率的重要因素。

（7）不断地改进软件工程实践。随着计算机科学技术的发展，软件从业人员只有不断地总结经验并且主动学习新的软件技术，才能不落后于时代。

*软件工程过程*是指在软件的生命周期内，为了实现特定目标而进行的一系列相关活动。每个活动都有其确定的实现步骤。过程也可以划分为多种类型，如开发过程、维护过程、支持过程和管理过程等。

*软件工程方法*包含软件开发方法、软件度量方法、软件管理方法和软件环境方法，但是通常把软件工程方法等同于软件开发方法。目前，常用的软件开发方法有面向过程的开发方法、面向对象的开发方法、面向数据的开发方法和形式化方法。在本书的后续章节中，我们将详细介绍面向过程的开发方法和面向对象的开发方法。

面向数据的开发方法是指面向元数据的方法，它与关系数据库管理系统紧密地联系在一起。从 20 世纪 80 年代开始流行的关系数据库管理系统和关系数据库程序设计语言促进了面向数据的开发方法的发展。这种方法适合于信息系统中数据层的开发，在程序执行的过程中，可以根据数据流动和处理的需要，选择由程序控制还是由用户控制。

形式化方法建立在严格的数学逻辑的基础上，以逻辑推理为出发点，其内容包括有限状态机、程序的正确性证明、净室软件工程、Petri 网和通信系统演算等。形式化方法多用在对安全性和可靠性要求较高的系统中，如航空航天系统和铁路运输系统等。

软件工程模型有领域模型、需求模型、设计模型、实现模型和测试模型。每种模型都反映了一定的工作特点，它们可以由建模语言（如 UML）来描述。

除了以上这些内容以外，软件工程还包括管理、度量、环境和应用等领域的内容。总的来说，软件工程是一门交叉学科，涉及的范围很广泛。

1.3.3 软件工程的知识体系

虽然“软件工程”这个概念对我们来说并不陌生，它在 20 世纪 60 年代就被提出，但是，软件工程作为一个合理的工程学科和一个被认可的职业状态的时间却不长。IEEE 计算机学会主持了一个名为“软件工程知识体系指南”（SWEBOK: Guide to the Software Engineering Body of Knowledge 2004 Version）的项目，它清晰地描述了“软件工程知识体系”的内容以及体系中哪些部分已经被普遍接受。概括来讲，IEEE 计算机学会建立“软件工程知识体系指南”的目的主要有以下几点。

（1）促进世界范围内对软件工程的一致观点。

（2）阐明软件工程相对于其他学科的位置，并确立它们的分界。“软件工程知识体系指南”认为与软件工程相关的学科有 8 个，分别是

① 计算机工程（computer engineering）；
② 计算机科学（computer science）；
③ 管理（management）；
④ 数学（mathematics）；
⑤ 项目管理（project management）；
⑥ 质量管理（quality management）；
⑦ 软件人类工程学（software ergonomics）；
⑧ 系统工程（systems engineering）。

（3）刻画软件工程学科的内容。

（4）提供使用知识体系的主题。

（5）为开发课程表、个人认证和许可材料提供基础。

知识小卡片

电气电子工程师学会（Institute of Electrical and Electronics Engineers， IEEE）是一个成立于 1963 年 1 月 1 日的国际性电子技术与电子工程师协会，亦是世界上最大的专业技术组织之一，拥有来自 175 个国家的 36 万会员。除设立于美国纽约市的总部以外，其在全球 150 多个国家亦拥有分会，并且还有 35 个专业学会及两个联合会。其每年均会出版多种杂志、学报、书籍，亦举办至少 300 次的专业会议。目前 IEEE 在工业界所定义的标准有着极大的影响。

IEEE 定位在“科学和教育，并直接面向电子电气工程、通信、计算机工程、计算机科学理论和原理研究的组织以及相关工程分支的艺术和科学”。为了实现这一目标，IEEE 承担着多个科学期刊和会议组织者的角色。它也是一个广泛的工业标准开发者，主要领域包括电力、能源、生物技术和保健、信息技术、信息安全、通信、消费电子、运输、航天技术和纳米技术。

在 SWEBOK 中，“软件工程”被定义为

（1）应用系统化、学科化和定量化的方法来开发、运行和维护软件，即将工程应用到软件中；

（2）对（1）中各种方法的研究。

SWEBOK 将“软件工程”组织为 11 个知识域（knowledge areas，KA），如图 1-6 所示。

图 1-6　软件工程的知识域

软件需求是为解决特定问题而必须由被开发的软件展示的特性。软件需求知识域涉及软件需求的获取、分析、规格说明和确认，它还可以进一步分解为 7 个子领域，如图 1-7 所示。

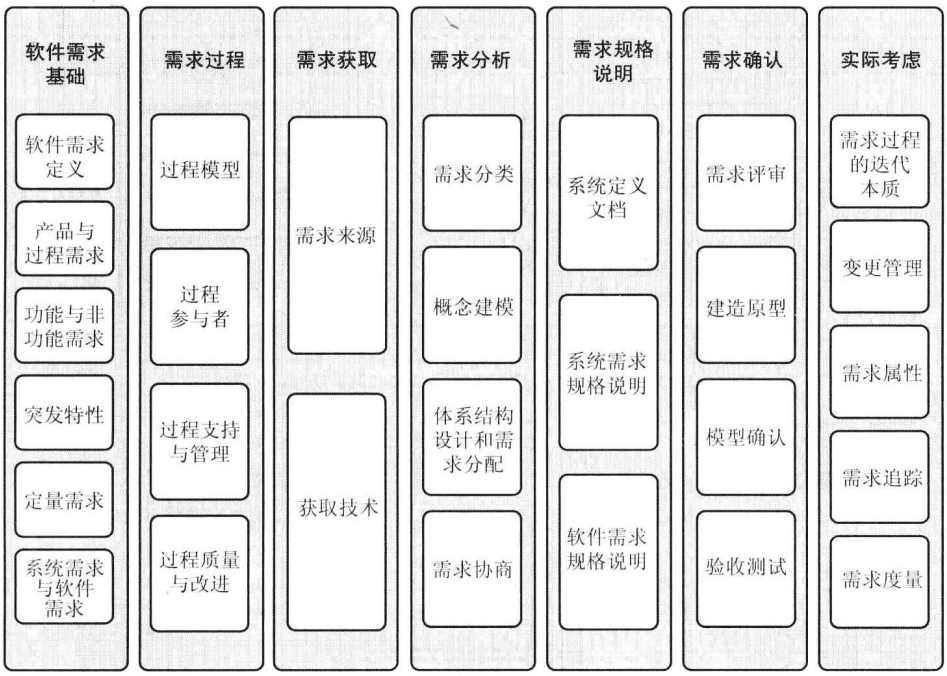


图 1-7　软件需求知识域的结构分解

定义一个系统或组件的体系结构、组件、接口和其他特征的过程以及这个过程的结果被称为软件设计。软件设计包含 6 个子领域，如图 1-8 所示。

软件构造是指通过编码、验证、单元测试、集成测试和调试的组合，详细地创建可工作的和有意义的软件的过程。软件构造知识域包含 3 个子领域，如图 1-9 所示。

软件测试是为评价和改进产品的质量，标识产品的缺陷和问题而进行的活动，软件测试知识域的结构分解如图 1-10 所示。