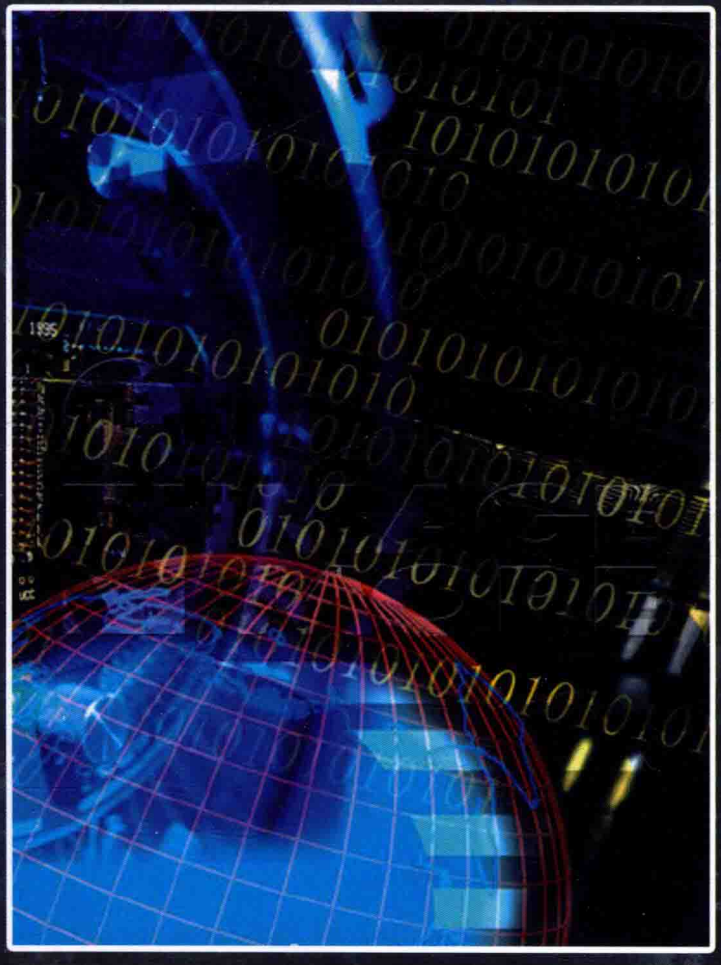
普通高等教育“十三五”规划教材

软件工程

Software Engineering

陈 永 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等教育“十三五”规划教材

软件工程

陈 永 主 编
张 薇 杨 磊 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

软件工程是一门讲授如何采用工程学的原理、技术和方法开发与维护软件的课程。本书结合国内外软件工程领域最新的进展，以软件设计方法、工具应用为主线，通过基础篇、高级篇和案例篇三部分，从实践、实用的角度，通过丰富的实例和热点问题，系统地介绍了软件工程的基本思想、方法。本书内容全面，涉及大数据、云计算、可穿戴计算、面向服务计算等新技术内容，此外，对软件职业素质和职业道德也进行了阐述。

本书取材新颖，深入浅出地介绍了软件工程基本理论和前沿技术。通过典型项目案例和应用实例，使读者能够快速掌握软件工程理论和设计方法。

本书适合作为计算机科学与技术、软件工程、信息管理与信息系统、信息与计算科学等专业本科生的教材，也可作为研究生及相关学科领域软件技术研究人员的参考用书。

图书在版编目（CIP）数据

软件工程/陈永主编. —北京：中国铁道出版社，2017.1

普通高等教育“十三五”规划教材

ISBN 978-7-113-19717-9

Ⅰ.①软… Ⅱ.①陈… 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（2017）第 005124 号

书　　名：软件工程
作　　者：陈　永　主编

策　　划：周海燕　　**读者热线：**（010）63550836
责任编辑：周海燕　冯彩茹
封面设计：刘　颖
封面制作：白　雪
责任校对：张玉华
责任印制：郭向伟

出版发行：中国铁道出版社（100054，北京市西城区右安门西街 8 号）
网　　址：http://www.51eds.com
印　　刷：北京市燕鑫印刷有限公司
版　　次：2017 年 1 月第 1 版　　2017 年 1 月第 1 次印刷
开　　本：787 mm×1 092 mm　1/16　**印张：**24.75　**字数：**601 千
书　　号：ISBN 978-7-113-19717-9
定　　价：58.00 元

版权所有　侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社教材图书营销部联系调换。电话：（010）63550836

打击盗版举报电话：（010）51873659

前言

软件工程采用工程学的概念、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，研究和应用如何以系统性的、规范化的、可定量的过程化方法开发和维护软件的学科。随着信息处理技术的不断发展，软件的作用越来越广泛，对软件的开发方法、开发理念、开发工具提出了更高的要求。

软件工程是高等学校软件工程学科和计算机科学与技术学科专业的一门重要专业基础课程。本书针对软件开发过程中的理论体系进行讲解，通过基础理论、高级软件开发技术、项目案例实战，从经典软件工程基本方法到形式化方法、面向服务软件工程、软件质量管理、合同、职业素质与职业道德、软件标准化文档等问题进行了系统的阐述。

在本书的编写过程中，编者结合多年的授课讲义，充分吸取了国内外经典软件工程著作内容，并增加了大量新技术和新方法。在本书知识结构及教学内容上进行了精心推敲和认真规划，其主要特点如下：

（1）在保证软件工程理论体系完整的同时，突出软件工程新技术和新方法，对大数据、云计算、面向服务计算、极限编程、领域工程等问题有所阐述，该部分内容体现了软件工程知识的新颖性。

（2）软件设计方法与设计工具并重，合理安排教学内容，通过全新实例和项目案例突出设计方法和设计工具的使用。如基于二维码食品安全追溯、人体运动捕捉系统、虚拟心脏系统、按图搜索网上订餐系统等内容，科学地丰富了课程内容。

（3）遵循教材编写规律，注重教材写作技巧，尤其是对一些软件工程新技术、新思想，采用由浅入深、实例引导的方式，有效提高了教材的可读性和易理解性。

本书正文共三部分，分为17章，选材上内容新颖，案例翔实，不仅能体现经典软件工程基本理论的知识性，还能体现实践性、前瞻性。本书通过理论、设计、应用的主线，做到了理论和设计分析相结合，同时，本书注重各专业、学科间知识的融合交叉性，内容涉及管理学、认知心理学、数学、人机工程学、经济学、职业素质和职业道德等学科内容。

第一部分为基础篇，包括第1章至第10章：第1章概述了软件工程基本概念和软件工程人员职业素质和职业道德；第2章主要对各种软件过程和开发模型进行了介绍；第3章讲解了软件规划和可行性分析；第4章介绍了软件需求分析方法、任务和各种分析工具；第5章介绍了软件总体设计及食品安全追溯案例分析，重点讲解了面向数据流的设计方法；第6章介绍了人

机交互设计、设计原则、可穿戴计算；第7章介绍了详细设计工具和方法，并对程序复杂度定量计算进行了讲解；第8章介绍了软件编码风格和语言选择；第9章介绍了软件各种测试方法和压力测试、容量测试等内容；第10章介绍了软件维护、软件再工程、软件复用等技术。

第二部分为高级篇，包括第11章至16章。第11章介绍了软件形式化方法在软件设计中的应用，并给出案例说明；第12章对软件设计模式技术进行了讲解；第13章介绍了极限编程思想及过程；第14章介绍了大数据与面向服务的软件工程；第15章介绍了软件项目管理与质量控制；第16章介绍了合同的相关知识。

第三部分为案例篇，包括第17章，通过对项目案例“网上订餐系统”的设计与实现，对软件工程各个设计阶段的思想和设计方法、工具实现过程进行了详细讲解。

本书在编写过程中，通过总结编者多年的授课经验，力图做到概念清晰、内容新颖、层次分明、通俗、易懂、理论联系实际，以对读者富有启发性。

本书由陈永（兰州交通大学）任主编，张薇（兰州交通大学）、杨磊（甘肃省教育考试院）任副主编。其中，第 1 章到第 7 章及附录 D 由陈永编写；第 8 章到第 15 章由张薇编写；其余章节及附录 A、B、C 由杨磊编写。硕士研究生张甜甜、贺红参与了本书的校对工作，优秀本科毕业生安书山、张海钰、陈振花为本书提供了部分翔实的案例素材。

在本书的编写过程中，参考了大量的互联网电子资源、图书文献资料，在此向相关作者致以谢意。本书也得到了中国铁道出版社、兰州交通大学、甘肃省教育考试院有关部门的帮助和支持，在此一并表示感谢。

鉴于编者水平有限，加之时间仓促，书中难免存在疏漏及不足之处，恳请读者指正。

编　者

2016 年 12 月

此为试读，需要完整PDF请访问：www.ertongbook.com

目 录

基 础 篇

高　级　篇

案 例 篇

此为试读，需要完整PDF请访问：www.ertongbook.com

基础篇

第1章 概　论

软件（Software）是一系列按照特定顺序组织的计算机数据和指令的集合。软件一般被划分为系统软件、应用软件和介于这两者之间的中间件软件。软件并不是只包括可以在计算机（这里的计算机是指广义的计算机）上运行的程序，与这些程序相关的文档、数据等元素一般也被认为是软件的一部分。

随着计算机技术的迅速发展和广泛应用，社会对软件的需求也与日俱增，软件在计算机系统中的比重不断增大。现代社会已经离不开软件。国家基础设施和公共建设，工业制造、金融、交通等行业，软件已经成为必不可少的一部分。软件可以将劳动生产率水平进一步提高，促进经济全球化、经济增长集约化、环保经济绿色化、军事技术信息化，甚至影响和改变着人类的生活方式。软件从最初的计算机硬件的附属品，仅仅作为计算机硬件的运行和做一些简单的计算与数据处理的程序，发展到今天大规模的封闭或开放式的系统软件和应用软件。有的软件的源代码甚至超过千万行。例如，美国阿波罗计划的软件长达 1 000 万行，航天飞机计划的软件更是长达 4 000 万行，桌面操作系统为千万级量级规模。如今，物联网技术、云计算、大数据、移动互联网融合发展，为生产生活、社会管理带来深刻变化。现代软件技术结合物联网、大数据、云计算和移动互联网、虚拟现实、大规模并行计算等一系列技术让“智慧城市”与“智慧交通”的美好画卷正在变成现实。

软件是抽象的，是人类逻辑思维的产物，它不受物质材料的限制，也不受物理定律或加工过程的制约，这一特性使软件工程得以简化，因为软件的潜能不受物理因素的限制；另外一方面，由于缺乏自然约束，软件系统的实现在实施过程中，容易变得极为复杂，理解它会很困难、改变它付出的代价更加高昂。软件规模的增长，使其复杂度也随之大大增加，而高复杂度和高可靠性的不相容性，使得软件可靠性随着其规模的增长而降低，质量难以保证，维护愈加困难，投资预算很难控制，传统的软件研制开发方法已无法适应大规模软件的开发需求。

为了解决在软件开发和维护过程中遇到的一系列软件危机的严重问题，1968 年，北大西洋公约组织（NATO）的科学家和官员们在原德意志联邦共和国召开的国际会议上讨论并首次提出了软件开发要工程化。当时，单个的程序开发技术已经不能扩展并应用到大型的、复杂

的软件系统中。软件项目有时甚至要推迟几年才能完成，不仅比预计的费用高且难以维护。软件工作者开始认真研究消除软件危机的途径，从而逐渐形成了一门新兴的工程学科——计算机软件工程学（Software Engineering），简称软件工程。软件工程是一门工程学科，涉及软件生产过程中的各个方面，从最初的问题提出一直到投入使用后的系统维护，都属于其学科研究范畴。

1.1 软 件 危 机

1.1.1 摩尔定律和超越摩尔

1965 年，Intel 联合创始人戈登 · 摩尔提出了著名的理论：半导体芯片上可集成的元器件的数目每 12 个月便会增加一倍。也就是说，同样规格的芯片的成本，每 12 个月便会降低一半。1965 年每个芯片可以容纳 50 个晶体管，摩尔预测到 1970 年，每个芯片将能够容纳 1 000 个元器件，每个晶体管的价格会降低 90%。

经过简化，这个发现被归纳为“摩尔定律”：每个芯片上晶体管的数目每 12 个月将会增加一倍。戈登 · 摩尔的发现不基于任何特定的科学或工程理论，只是真实情况的映射总结。硅芯片行业注意到了这个定律，没有简单地把它当作一个描述的、预言性质的观察，而是作为一个说明性的、重要的规则、整个行业努力的目标。

除此之外，还有一个与摩尔定律相对的洛克定律（Rock's law），强调了生产中的成本因素。通过观察可知，芯片制造厂商的成本每 4 年便会增加一倍。技术的进步以不断为芯片上晶体管数量的增加铺平道路，但是芯片生产设施的建造会十分昂贵，而更小、更便宜的处理器的使用还在不断增加。

硬件技术在不断发展，但现在，这种发展轨迹要告一段落了。由于同样小的空间里集成越来越多的硅电路，产生的热量也越来越大，这种原本两年处理能力加倍的速度已经慢慢下滑，原本的摩尔定律在逐步失效。目前，行业研究规划蓝图新的战略是“超越摩尔”（More than Moore）：与以往首先改善芯片、软件随后跟上的发展趋势不同，以后半导体行业的发展将首先看软件——从手机到超级计算机再到云端的数据中心——然后反过来看要支持软件和应用的运行需要什么处理能力的芯片来支持。

这种局势的转变使得人们更加强调软件的重要性。计算机的应用日益广泛、深入，然而硬件的进步只是为计算机系统提供了潜在的能力，如果没有软件来驾驭和开发这种能力，人类并不能有效地使用计算机，因此，软件已成为限制计算机系统发展的关键因素。

计算机软件是一个逻辑的而非物理的系统，它具有与硬件显著的不同特点。它的主要工作集中在定义、开发、维护等纯智力活动方面。随着软件需求的剧增，软件规模不断增大，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修改程序；硬件或操作系统更新时，通常需要修改程序以适应新的环境。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。

1.1.2 软件危机的介绍

软件危机（Software Crisis）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的问题，也可以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。

广义上讲，所谓软件危机包含两方面问题：如何开发软件，以满足对软件日益增长的需求；如何维护数量不断膨胀的已有软件。

狭义上讲，所谓软件危机主要有以下一些典型表现：

（1）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降低了软件产品的质量，从而不可避免地会引起用户的不满。

（2）开发人员和用户之间很难沟通，矛盾很难统一。往往是软件开发人员不能真正了解用户的需求，而用户又不了解计算机求解问题的模式和能力，双方无法用共同熟悉的语言进行交流和描述。在双方互不充分了解的情况下，就仓促上阵设计系统、匆忙着手编写程序，这种“闭门造车”的开发方式必然导致最终的产品不符合用户的实际需要。

（3）大型软件项目需要组织一定的研发人力共同完成。软件项目管理人员缺乏开发大型软件系统的经验及软件开发各类人员的信息交流不及时、不准确，有时还会产生误解，这些都会导致软件质量无法得到保证。

（4）软件系统中的错误难以消除。软件是逻辑产品，质量问题很难以统一的标准度量，因而造成质量控制困难。软件产品并不是没有错误，而是盲目检测很难发现错误，而隐藏下来的错误往往是造成重大事故的隐患，这些都会导致软件产品出现质量问题。

（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求在原有程序中增加一些新的功能。“可重用的软件”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。错误的观点经常认为：软件就是程序。程序代码写完软件也就设计完了。实际上软件不仅仅是程序，还应该有一整套文档资料。这些文档资料应该是软件开发过程中产生出来的，而且应该是和程序代码完全一致的。软件开发过程中，基线是软件文档和源代码的一个稳定版本，它是进一步开发的基础。软件开发组织的管理人员可以使用这些文档资料作为“里程碑”，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成本中所占的比例逐年上升。随着互联网时代的到来，电子商务、移动互联网兴起，软件经济已经影响到社会经济生活中的方方面面。硬件成本逐年下降，然而软件开发需要大量人力，软件成本随着软件规模和数量的不断扩大而持续上升。

（8）软件开发生产率跟不上计算机应用系统迅速普及深入的速度。软件开发是一种高强度的脑力劳动，理论性和实践性都很强，软件开发人员的生产效率也对开发的周期和质量有

很大影响。特别是软件工程，对软件开发的成功（按质按量，按期完成）有决定性作用。

（9）软件产品的特殊性和人类智力的局限性，导致人类无力处理“复杂问题”。“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开发效率和能力，新的、更大的、更复杂的问题又摆在人们的面前，所以“复杂问题”的解决需要诸多学科知识及技术的协同发展。

以上举例的仅仅是软件危机的一些典型表现，与软件开发和维护有关的问题远不止这些。

1.1.3 产生软件危机的原因

开发软件系统需要投入大量的人力和物力，但软件系统的质量却难以保证，也就是说，开发软件所需的高成本同产品的低质量之间有着尖锐的矛盾，这种现象就是所谓的“软件危机”。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，而另一方面的主要原因是软件开发和维护的方法不正确有关。

软件开发不同于一般的加工制造业、机械工业以及一般的加工业，这些行业都已经有了上百年的历史，产品的生产流程及工厂、车间、工种等的机构设置和角色分工都有了成熟的模式。但是，软件企业及软件产品的生产，历史不长，加之软件本身的智力劳动的特性，软件作为产品的生产流程及其相应的管理活动，还远远没有一个成熟的模式。此外，软件不同于一般程序，它的一个显著特点是规模庞大，而程序复杂性将随着程序规模的增加而呈指数上升。为了在预定时间内开发出规模庞大的软件，必须由许多人分工合作。然而，如何保证每个人完成的工作合在一起确实能够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，这不仅涉及许多技术问题，如分析方法、设计方法、形式说明方法、版本控制等，更重要的是必须有严格而科学的管理。

与软件开发和维护有关的许多错误认识和做法的形成，可归因于在计算机系统发展的早期阶段软件开发的个体化特点。错误的认识和做法主要表现为忽视软件需求分析的重要性，认为软件开发就是写程序并设法使之运行，轻视软件维护等。另外，软件开发过程中如果缺乏有力的方法学和工具方面的支持会产生软件危机。由于软件开发不同于大多数其他工业产品，其开发过程是复杂的逻辑思维过程，其产品极大程度地依赖于开发人员高度的智力投入。由于过分地依靠程序设计人员在软件开发过程中的技巧和创造性，加剧软件开发产品的个性化，也是发生软件开发危机的一个重要原因。

软件项目管理（Software Project Management）的对象是软件工程项目。它所涉及的范围覆盖了整个软件工程过程。为使软件项目开发获得成功，关键问题是必须对软件项目的工作范围、可能风险、需要资源（人、硬件、软件）、要实现的任务、经历的里程碑、花费工作量（成本）、进度安排等做到心中有数。这种管理在技术工作开始之前就应开始，在软件从概念到实现的过程中继续进行，当软件工程过程最后结束时才终止。

软件项目管理是为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对人员（People）、产品（Product）、过程（Process）和项目（Project）进行分析和管理的活动。软件项目管理的根本目的是为了让软件项目尤其是大型项目的整个软件生命周期（从分析、设计、编码到测试、维护全过程）都能在管理者的控制之下，以预定成本按期、按质地完成软件交付用户使用。研究软件项目管理要从已有的成功或失败的案例中总结出能够指导今后开发的

通用原则、方法，同时避免前人的失误。软件工程学的一个重要目标就是结合软件开发技术和先进管理技术，以提高软件的可维护性，减少软件维护的代价。

1.1.4 消除软件危机的途径

软件工程作为一个新兴的工程学科，主要研究软件生产的客观规律性，建立与系统化软件生产有关的概念、原则、方法、技术和工具，指导和支持软件系统的生产活动，以期达到降低软件生产成本、改进软件产品质量、提高软件生产率水平的目标。软件工程学从硬件工程和其他人类工程中吸收了许多成功的经验，明确提出了软件生命周期的模型，发展了许多软件开发与维护阶段适用的技术和方法，并应用于软件工程实践，取得了良好的效果。

为了消除软件危机，首先应该对计算机软件有一个正确的认识。软件设计者应该彻底消除在计算机系统早期发展阶段形成的“软件就是程序”的错误观念。一个软件必须由一个完整的配置组成，事实上，软件是程序、数据及相关文档的完整集合。其中，程序是能够完成预定功能和性能的可执行的指令序列；数据是使程序能够适当处理信息的数据结构；文档是开发、使用和维护程序所需要的图文资料。1983 年 IEEE 为软件下的定义是：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程序时所必需的数据。

更重要的是，必须充分认识到软件开发不是某种个体劳动的神秘技巧，而应该是一种组织良好、管理严格、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，特别要吸取几十年来人类从事计算机硬件研究和开发的经验教训。

在软件开发过程中人们开始研制和使用软件工具，用以辅助进行软件项目管理与技术生产，人们还将软件生命周期各阶段使用的软件工具有机地集合成为一个整体，形成能够连续支持软件开发与维护全过程的集成化软件支持环境，以期从管理和技术两方面解决软件危机问题。应该推广使用在实践中总结出来的开发软件的成功的技术和方法，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。

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烦琐重复的工作需要做，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结合成一个整体，支持软件开发的全过程，则称为软件工程支撑环境。

此外，人工智能与软件工程的结合成为 20 世纪 80 年代末期活跃的研究领域。基于程序变换、自动生成和可重用软件等软件新技术研究也已取得一定的进展，把程序设计自动化的进程向前推进一步。软件标准化与可重用性得到了工业界的高度重视，在避免重用劳动、缓解软件危机方面起到了重要作用。

软件开发的风险之所以大，是由于软件过程能力低，其中最关键的问题在于软件开发组织不能很好地管理其软件过程，从而使一些好的开发方法和技术起不到预期的作用。而且项目的成功也是通过工作组的共同努力，所以仅仅建立在可得到特定人员上的成功不能为全组织的生产和质量的长期提高打下基础，必须在建立有效的软件如管理工程实践和管理实践的基础设施方面，坚持不懈地努力，才能不断改进，才能持续地成功。

软件质量，乃至于任何产品质量，都是一个很复杂的事物性质和行为。产品质量，包括软件质量，是人们实践产物的属性和行为，是可以认识、可以科学地描述的。还可以通过一些方法和人类活动，来改进质量。针对以上问题，可以在软件开发过程中实施能力成熟度模

此为试读，需要完整PDF请访问：www.ertongbook.com