TURING 图灵计算机科学丛书

PEARSON

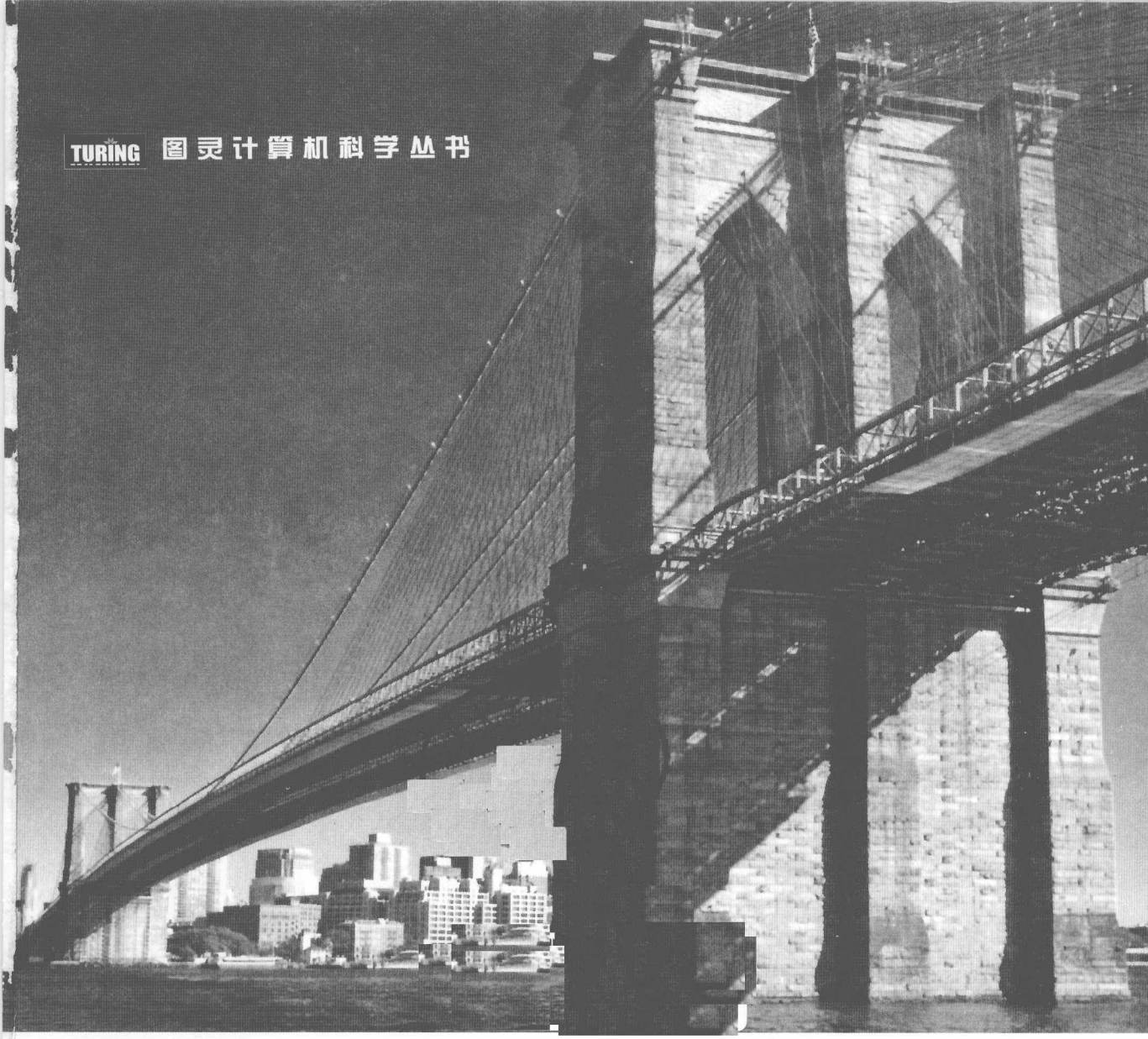
Software Engineering Theory and Practice (Fourth Edition)

软件工程（第4版）

[美] Shari Lawrence Pfleeger
[加] Joanne M. Atlee 著

杨卫东 译

人民邮电出版社
POSTS & TELECOM PRESS



Software Engineering Theory and Practice (Fourth Edition)

软件工程（第4版）

[美] Shari Lawrence Pfleeger
[加] Joanne M. Atlee 著

杨卫东 译

人民邮电出版社
北 京

图书在版编目（CIP）数据

软件工程 : 第4版 / (美) 弗里格 (Pfleeger, S. L.), (加) 阿特利 (Atlee, J. M.) 著 ; 杨卫东译. -- 北京: 人民邮电出版社, 2010.5
(图灵计算机科学丛书)
书名原文: Software Engineering:Theory and Practice, Fourth Edition
ISBN 978-7-115-20551-3

Ⅰ. ①软… Ⅱ. ①弗… ②阿… ③杨… Ⅲ. ①软件工程 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字(2010)第056678号

内 容 提 要

本书是软件工程领域的经典著作，国际上众多名校均采用本书作为教材。本书分为3个部分。第一部分解释为什么软件工程知识对实践者和研究者同样重要，还讨论了理解过程模型问题的必要性以及敏捷方法和精细地进行项目计划的必要性；第二部分论述开发和维护的主要步骤；第三部分主要讲述软件评估和改进。

本书适合作为计算机相关专业软件工程课程的本科教材，也适用于介绍软件工程的概念与实践的研究生课程，期望进一步学习该领域相关知识的专业人员也可以阅读本书。

图灵计算机科学丛书

软件工程（第4版）

◆ 著 [美] Shari Lawrence Pfleeger [加] Joanne M. Atlee
译 杨卫东
责任编辑 杨海玲

◆ 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 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：787×1092 1/16
印张：35.25
字数：1067千字 2010年5月第1版
印数：1-3 000册 2010年5月河北第1次印刷

著作权合同登记号 图字：01-2009-5712号

ISBN 978-7-115-20551-3

定价：79.00元

读者服务热线：(010)51095186 印装质量热线：(010)67129223
反盗版热线：(010)67171154

版权声明

前　言

跨越研究与实践之间的鸿沟

在1968年的北约会议上，首次使用了“软件工程”这一术语。时至今日，软件工程已经走过了很长的一段路，软件本身也已经以各种形式融入了我们的生活。就算在10年前，估计也没人会预料到软件会有这么大的影响力。因此，要懂得如何开发好的软件以及如何评估软件在日常生活中面临的风险和机遇，坚实的软件工程理论和实践基础是不可或缺的。本书体现了当前软件工程领域实践者阵营和研究者阵营之间相互融合的现状：实践者主要关注构造高质量产品完成某些功能，而研究者则努力寻找各种方法改进产品质量以及提高开发人员生产效率。Edsgar Dykstra不断提醒我们：研究与实践之间的紧张关系将检验我们对软件工程的理解，并帮助我们改进思维方式、方法，进而最终改进我们的产品。

正是本着这种精神，我们对本书进行了增订，为这种不断的探究和改进构造一个基础框架。尤其是，第4版纳入了更广泛的素材，来说明如何抽象出一个问题并对它建立模型，以及如何使用各种模型、设计原则、设计模式和设计策略，来产生适当的解决方案。软件工程师绝不止像程序员那样按照说明书来编写程序，就像厨师不止是遵循菜谱来烹饪。构造优秀的软件是一门艺术，这体现在如何抽象出问题的要素并建模，再使用这些抽象设计出解决方案。经常能听到优秀开发人员谈论“优雅的”解决方案，说明这样的解决方案抓住了问题的核心，得到的软件不仅能够解决当前的问题，而且当问题随着时间演化时，软件也能够很容易地进行修改。这样，学生就能够学会融合研究与实践、艺术与科学，构造出坚实的软件。

科学总是以事实为基础的。本书是为本科生软件工程课程而设计的，注重软件工程研究与实践的实际效果层面，使学生能够直接将所学知识应用于要解决的现实问题。书中所举的例子针对的是经验有限的学生，但是，这些例子清楚地阐明了大型软件开发项目是如何从需求到计划，再进而成为现实的过程。例子所描述的许多情形，读者未来都很可能经历：大型项目与小型项目、“敏捷”方法与高度结构化方法、面向对象与面向过程的方法、实时处理与事务处理、开发情形与维护情形。

本书也适用于介绍软件工程的概念与实践的研究生课程，还适合于那些期望进一步学习该领域相关知识的专业人员。尤其是最后3章给出了一些引人思考的资料，旨在引起研究生对当前研究主题的兴趣。

主要特色

下面是本书区别于其他书的主要特色。

- 其他软件工程图书将度量和建模作为单独的问题分开考虑，本书则将二者与更全面的软件工程论述结合起来。也就是说，把度量和建模看作是软件工程策略不可分割的部分，而不作为一个单独的分支。这样，学生能够学会如何抽象与建模，以及如何在日常开发活动中进行定量评估和改进。他们可以利用他们的模型理解所要解决的问题的要素，了解可供选择的其他解决方案；他们可以利用度量对个人、小组和项目的进度进行评估。
- 类似地，诸如复用、风险管理和质量工程之类的概念都糅合到在它们影响之下的软件工程活动中，而不是作为单独的问题来讲述。

- 这一版介绍了敏捷方法的使用，包括极限编程。它论述了给予开发人员更多自主权所带来的利益和遭遇的风险，并将敏捷方法与传统的软件开发方法进行了比较。
- 每一章都将相应的概念应用于两个贯穿全书的例子：一个例子是典型的信息系统，另一个是实时系统。两个例子都基于实际的项目。信息系统的例子描述了一个大型英国电视公司确定广告时间价格的软件。实时系统的例子是阿丽亚娜5型火箭的控制软件，我们将考查该软件见诸报端的一些问题，并探讨软件工程技术如何能够帮助找出并避免其中的问题。学生能够随着两个典型项目的进展，领会本书所描述的各种实践方法如何融入到构造系统的技术之中。
- 每章结尾的结论都用三种形式表述：这一章的内容对开发团队的意义，对单个开发人员的意义，以及对研究者的意义。学生可以很容易地复习每章的要点，并了解每一章与研究以及实践的相关性。
- 本书有配套网站，网址是http://www.prenhall.com/pfleeger。该网站包含了各种文献中的最新例子，以及真实项目中的实际工件（artifact）的例子。该网页还提供了相关工具和方法厂商的网页的链接。学生可以在这里找到实际的需求文档、设计方案、代码、测试计划等等内容。那些想寻找更多、更深入信息的学生，可以通过该网页找到其他值得信赖的、容易理解的出版物及网站。这些网页会定期更新，以及时补充书中的内容。它还包括一个工具，读者可用来向作者和出版社提供反馈。
- 学生学习指南可从Pearson的销售代表处获得。
- PPT及完整的习题答案手册可以从本书配套网站的教师资源中心获得。要获得访问权限，请与Pearson的销售代表联系。
- 本书从各种文献中引用了大量案例研究和例子。书中以补充材料形式给出的很多只有一页左右篇幅的案例研究在网页上都有详细描述。据此，学生能够了解本书中的理论概念是如何应用于现实情形的。
- 每章最后都列出了引人思考的、与软件工程有关的法律和道德问题。学生可以根据软件工程所处的社会和政治背景来理解软件工程。与其他的学科一样，软件工程的结果会影响到人，必须据此来考虑软件工程决策。
- 每章都强调过程化和面向对象的开发。另外，第6章是专门讲述面向对象开发的，解释了面向对象开发过程的步骤。我们会讨论若干设计原则，并用面向对象的例子展示如何结合这些原则来改善设计。
- 本书还有一个带有注解的参考文献列表，其中不少是软件工程中开创性的论文。另外，本书的网页有进一步的链接指向注释的参考文献和特定领域讨论组（诸如软件可靠性、容错性、计算机安全等）。
- 每章都会提到同一个学期项目，是开发一个住房抵押处理系统软件，教师可以把这个学期项目稍加改动，作为课程作业。
- 每章的结尾都列出了本章概念的主要参考文献，学生能够借此进一步深入研究本章中讨论的特定工具和方法。
- 这一版还包含了强调计算机安全性的例子。尤其是，我们强调了设计时的安全性，而不是在编码或测试的时候加入安全性。

本书的内容和组织结构

本书分为3部分。第1部分力图激发读者学习软件工程的兴趣，解释为什么软件工程知识对实践者和研究者同样重要。第1部分还讨论理解过程问题的必要性，确定开发人员具有多大程度“敏捷性”

的必要性和精细地进行项目计划的必要性。第2部分论述开发和维护的主要步骤，这些步骤与构造软件所使用的过程模型关系不大，都是先引出需求、对需求建模、检查需求，然后设计问题的解决方案，编写和测试代码，最后将软件交付给客户。第3部分主要讲述软件评估和改进，这一部分着眼于如何评价过程和产品的质量，以及如何改进质量。

第 1 章：软件工程概述

这一章介绍软件工程的发展历程，以激发读者的学习兴趣，并简要介绍在后面的章节中要研究的某些关键问题。尤其是，讨论Wasserman用来帮助定义软件工程的关键因素：抽象、分析与设计方法、表示法、模块化与体系结构、软件生命周期与过程、复用、度量、工具与集成环境，以及用户界面与原型化。我们将讨论计算机科学和软件工程的差异，解释可能遇到的一些主要问题，并为本书的其余部分奠定基础。还探讨采用系统的方法构造软件的必要性，并介绍每一章都会用到的两个公共的例子。学期项目的背景也是在这里介绍的。

第 2 章：过程和生命周期的建模

这一章概要介绍不同类型的过程和生命周期模型，包括瀑布模型、V模型、螺旋模型和各种原型化模型。论述敏捷方法的必要性，与传统的软件开发过程相比，在敏捷方法中开发人员拥有许多自主权。还描述几种建模技术和工具，包括系统动态建模以及其他常用方法。两个公共的例子都用到了这里介绍的一些建模技术。

第 3 章：计划和管理项目

在这一章，我们着眼于项目计划和进度安排。介绍诸如活动、里程碑、工作分解结构、活动图、风险管理、成本以及成本估算等概念。用估算模型估算两个公共例子的成本和进度。我们重点关注实际案例研究，包括F-16飞机和DEC的alpha AXP程序的软件开发管理。

第 4 章：获取需求

这一章强调在优秀的软件工程中抽象和建模的关键作用。尤其是，我们使用模型澄清需求中容易误解和遗漏的细节，并使用模型与其他人员进行沟通。本章中探讨了多种不同的建模范型；针对每一种范型，研究表示法实例；讨论什么时候使用哪种范型，并对如何做出特定建模和抽象决策给出建议。讨论了需求的不同来源和类型（功能性需求、质量需求与设计约束）。解释如何编写可测试的需求，并描述如何解决其中的冲突。其他讨论的主题还有需求引发、需求文档、需求评审、需求质量、如何测量需求，并介绍了一个如何选择规格说明方法的例子。本章最后将其中几个方法应用于两个公共的例子。

第 5 章：设计体系结构

在本书的第4版中，完全改写了软件体系结构这一章。在这一章的开始，论述了体系结构在软件设计过程中以及在较大型的开发过程中的作用。我们详细讨论了产生软件体系结构的相关步骤，从建模、分析、文档化和评审，到最后产生软件体系结构文档；程序设计人员用软件体系结构文档来描述模块和接口。我们讨论了如何对问题进行分解，以及如何使用不同的视图来检查问题的若干方面，以便找到合适的解决方案。接着，我们集中讨论了如何使用一种或多种体系结构风格来建模解决方案，包括管道-过滤器、对等网络、客户/服务器、发布-订阅、信息库和分层体系结构。我们还讨论了如何组合体系结构风格并使用它们来达到质量目标，如可修改性、性能、安全性、可靠性、健壮性、易使用性。

一旦完成了初始的体系结构，就可以对其进行评估和改进。在这一章，我们说明了如何测量设

计质量，以及如何使用评估技术在故障分析、安全性分析、权衡分析以及成本效益分析中选择满足客户需求的体系结构。我们强调了把设计理念记入文档、确认和验证设计是否匹配需求、创建满足客户产品需求的体系结构这三方面的重要性。在本章快结束的时候，我们详细论述了如何构建一个产品线体系结构，以允许软件提供者在一族相似的产品中复用设计。最后，讨论了信息系统和实时系统这两个例子的体系结构分析。

第 6 章：设计模块

这一版中，第6章做了全面的修订，讨论了如何从系统体系结构的描述转向单个模块设计的描述。本章从讨论设计过程开始，然后介绍6个关键的设计原则来指导我们由体系结构到模块的细化设计：模块化、接口、信息隐藏、增量式开发、抽象和通用性。接着，进一步探讨面向对象的设计以及面向对象设计是如何支持这6个设计原则的。这一章使用统一建模语言的各种表示法来说明如何表示模块功能性和交互的多个侧面，这样，我们就能够构造健壮的、可维护的设计。本章还论述了一组设计模式，其中每一个设计模式都具有明确的目的，并说明了如何使用它们来强化设计原理。接下来，本章讨论了一些全局的问题，如数据管理、异常处理、用户界面和框架，我们从中可以了解到一致而清晰地使用方法可以获得更有效的设计。

本章详细探讨面向对象度量，并将一些常用的面向对象度量方法应用于一个服务站的例子。我们可以从中看到，设计中发生变化后，度量中的值是如何变化的，这有助于我们决定如何分配资源和如何查找故障。最后，我们把面向对象的概念应用到信息系统和实时系统的例子中。

第 7 章：编写程序

本章论述代码层的设计决策，以及实现一个设计以产生高质量代码所涉及的问题。我们讨论各种标准和过程，并给出一些简单的编程准则。提供了用多种语言编写的例子，包括面向对象语言和过程语言。讨论了程序文档和错误处理策略的必要性。最后，将一些概念应用于两个公共的例子。

第 8 章：测试程序

这一章探讨一些测试程序方面的问题。我们把传统测试方法和净室方法区分开，并着眼于如何测试各种系统。给出软件问题的定义和分类，讨论怎样利用正交缺陷分类使数据汇集和分析更加有效。随后，解释单元测试与集成测试之间的区别。在介绍几种自动化测试工具和技术之后，解释测试生命周期的必要性，以及如何将这些工具集成到生命周期中。最后，把这些概念应用于两个公共的例子。

第 9 章：测试系统

这一章首先介绍系统测试的原理，包括测试包和数据的复用，并讨论精细地进行配置管理的必要性。介绍的概念包括功能测试、性能测试、验收测试和安装测试。讨论测试面向对象系统的特殊需要。描述几种测试工具，并讨论测试小组成员的角色。然后，向读者介绍软件可靠性建模，并讨论可靠性、可维护性和可用性问题。读者将学会如何使用测试结果来评估交付的产品可能具有的特征。本章还介绍几种测试文档，最后描述两个公共例子的测试策略。

第 10 章：交付系统

这一章讨论培训和文档的必要性，并给出信息系统和实时系统例子中可能使用的几个培训和文档的例子。

第 11 章：维护系统

这一章强调系统变化的结果。解释在系统生命周期的过程中变化是如何发生的，以及系统设计、

编码、测试过程、文档如何必须与这些变化保持一致。讨论典型的维护问题，以及精细进行配置管理的必要性。全面讨论使用度量预测可能的变化，并评估变化所产生的影响。还讨论在使遗留系统再生的大背景下的再工程和重组技术。最后，根据变化的可能性对两个公共的例子进行评估。

第 12 章：评估产品、过程和资源

由于许多软件工程的决策涉及合并和集成现有组件，这一章讨论评估过程和产品的方法。讨论经验性评估的必要性，并给出若干例子以说明如何使用度量建立质量和生产率的基线。我们着眼于几个质量模型，如何评估系统的复用性，如何执行事后分析，以及如何理解信息技术的投资回报。把这些概念应用于两个公共的例子。

第 13 章：改进预测、产品、过程和资源

这一章建立在第11章的基础之上，说明如何完成预测、产品、过程和资源改进。包含几个深入的案例研究，以说明如何通过多种调查技术来理解和改进预测模型、审查技术及软件工程的其他方面。本章最后给出一组指导原则，用于评估当前情形并识别改进的机会。

第 14 章：软件工程的未来

在最后这一章，讨论软件工程中的若干悬而未决的难题。我们重新回顾Wasserman的概念，来审视软件工程这一学科的发展状况。研究技术转移和决策制定中的若干问题，以确定在把重要的思想从研究应用于实践方面，我们做得是否出色。最后，我们研究一些有争议的问题，比如给软件工程师发放职业证书的问题，以及向更特定领域解决方案和方法发展的趋势。

致谢

由于朋友及家人所给予的技术上的支持和情感上的鼓励，本书才得以完成。我们不可能列举出所有在本书的编写和修订期间帮助过我们的人，若有遗漏，在此提前表示歉意。我们很感谢本书早期版本的读者，他们详细审读了本书，提出了不少良好建议。我们已经尽力将所有这些建议融入这一版本。我们一如既往地感谢来自读者的反馈，不管是正面的还是负面的。

Carolyn Seaman（马里兰大学巴尔的摩校区）是本书第1版非常杰出的审稿人。她提出许多方法使内容更简洁明了，使得本书更紧凑、更易于理解。她还整理好大部分的练习解答，并帮助我们建立本书早期版本的网站。我要感谢她的友谊和帮助。Yiqing Liang和Carla Valle更新了该网站，并为第2版增加了重要的新资料；Patsy Ann Zimmer（滑铁卢大学）修订了本书第3版的网站，特别是关于建模表示法和敏捷方法。

我们万分感谢Forrest Shull（马里兰大学法朗霍夫中心）以及Roseanne Tesoriero（华盛顿学院），他们编写了本书最初的学习指导；感谢Maria Vieira Nelson（巴西米纳斯吉拉斯州天主教大学），他对第3版的解答手册及学习指导进行了修订；感谢Eduardo S. Barrenechea（滑铁卢大学）对第4版中素材的更新。还要感谢Hossein Saiedian（堪萨斯大学）制作了第3版和第4版的PowerPoint演示。我们要特别感谢Guilherme Travassos（里约热内卢联邦大学），本书使用了他与Pfleeger在马里兰大学期间共同编写的材料，而且他在以后的课程教学中，又对这些材料进行了大量扩充。

对我们有所帮助且颇有创见的本书所有4个版本的审稿人有：Barbara Kitchenham（英国基尔大学）、Bernard Woolfolk（朗讯公司）、Ana Regina Cavalcanti da Rocha（里约热内卢联邦大学）、Frances Uku（加利福尼亚大学伯克利分校）、Lee Scott Ehrhart（MITRE公司）、Laurie Werth（得克萨斯大学）、Vickie Almstrum（得克萨斯大学）、Lionel Briand（挪威Simula研究室）、Steve Thibaut（佛罗里达大学）、Lee Wittenberg（新泽西基恩学院）、Philip Johnson（夏威夷大学）、Daniel Berry（加拿大滑铁

卢大学）、Nancy Day（滑铁卢大学）、Jianwei Niu（滑铁卢大学）、Chris Gorringe（英国东英吉利大学）、Ivan Aaen（奥尔堡大学）、Damla Turget（中佛罗里达大学）、Laurie Williams（北卡罗来纳州立大学）、Ernest Sibert（锡拉丘兹大学）、Allen Holliday（加州大学，福乐顿市）、David Rine（乔治梅森大学）、Anthony Sullivan（得克萨斯大学，达拉斯）、David Chesney（密西根大学，安娜堡）、Ye Duan（密苏里大学）、Rammohan K. Ragade（肯塔基大学）以及Prentice Hall提供的一些不具名的评阅人。与下列人员的讨论使本书得到了许多改进和加强：Greg Hislop（德雷克赛尔大学）、John Favaro（意大利Intecs Sistemi公司）、Filippo Lanubile（意大利巴里大学）、John d'Ambra（澳大利亚新南威尔士大学）、Chuck Howell（MITRE公司）、Tim Vieregge（美国军方计算机应急反应组）以及James Robertson和 Suzanne Robertson（英国大西洋系统行业协会）。

感谢Toni Holm以及Alan Apt，他们使本书第3版的创作显得有趣并相对轻松。也感谢James Robertson和Suzanne Robertson让我们使用皮卡地里例子，还要感谢Norman Fenton同意使用我们软件度量一书中的资料。我们非常感谢Tracy Dunkelberger，她在我们编纂本书第4版的时候给予我们鼓励；我们也感激她的耐心和专业精神。还要感谢Jane Bonnell和Pavithra Jayapaul的完美工作。

这里，我们非常感谢一些出版商允许我们引用其中的一些图和例子。来自于*Complete Systems Analysis*（Robertson and Robertson 1994）和*Mastering the Requirements Process*（Robertson and Robertson 1999）的资料是从网站www.dorsethouse.com上抽取的，得到了Dorset House出版社的使用许可。练习1-1中的文章是在美联社的许可下从《华盛顿邮报》中引用的。图2-15及图2-16是在John Wiley & Sons公司的许可下，从Barghouti等人的著作（Barghouti et al. 1995）中引用的。图12-14和图12-15是在John Wiley & Sons公司的许可下，从参考文献（Rout 1995）中引用的。

在电气和电子工程师学会（IEEE）的许可下，我们引用了第2、3、4、5、9、11、12和14章中标注有IEEE版权的图和表。类似地，第14章中标注有ACM版权的三个表，是在计算机协会（ACM）的许可下引用的。表2-1和图2-11是在软件生产力联合技术发展中心的许可下，引用于参考文献（Lai 1991）。来自参考文献（Graham 1996a）的图8-16、图8-17是在Dorothy R. Graham的许可下引用的。在公众利益科学中心（华盛顿特区西北部康涅狄格大街1875号）的许可下，图12-11和表12-2改编自参考文献（Liebman 1994）。表8-2、表8-3、表8-5和表8-6是在McGraw-Hill公司的许可下引用的。来自于参考文献（Shaw and Garlan 1996）、（Card and Glass 1990）、（Grady 1997）以及（Lee and Tepfenhart 1997）的图和例子是在Prentice Hall的许可下引用的。

经过Norman Fenton的许可，表9-3、表9-4、表9-6、表9-7、表13-1、表13-2、表13-3、表13-4，以及图1-15、图9-7、图9-8、图9-9、图9-14、图13-1、图13-2、图13-3、图13-4、图13-5、图13-6和图13-7（部分或全部）引用或改编于Fenton和Pfleeger的著作（Fenton and Pfleeger 1997）。在Norman Fenton的善意许可下，图3-16、图5-19和图5-20引用或改编于Norman Fenton的课程注解。

我们特别感谢我们的雇主，兰德公司以及滑铁卢大学，他们分别给我们以鼓励。[①]我们感谢朋友及家人的关心、支持和耐心，写作本书占用了我们本应共度的时光。尤其是，Shari Lawrence Pfleeger要感谢皇家服务站的经理Manny Lawrence，也感谢他的薄记员Bea Lawrence，不仅因为他们与她及她的学生在皇家系统规格说明上的合作，而且因为他们作为父母所给予Shari的关爱和指导。Jo Atlee特别感谢她的父母，Nancy和Gary Atlee，他们对她所做的或尝试要做的任何事情都给予支持和鼓励；也感谢她的同事及学生，他们在本书的主要创作期间，愉快地承担了许多的额外工作。我们最要特别感谢的是Charles Pfleeger和Ken Salem，他们是我们获得赏识和持续不断的支持、鼓励和好心情的源泉。

Shari Lawrence Pfleeger
Joanne M. Atlee

① 请注意，本书并不是RAND公司的产品，也未经RAND的质量保证过程。本书的相关工作仅代表作者个人，并不代表我们所在的机构。

目　录

第1章

软件工程概述

本章讨论以下内容：

- 软件工程的含义；
- 软件工程的发展历程；
- “好的软件”的含义；
- 为什么系统的方法是重要的；
- 自20世纪70年代以来，软件工程是如何变革的。

软件在生活中随处可见，我们时常认为，软件理所当然地应该使我们的生活更加舒适、高效和有效。例如，准备早餐面包这样一个简单的任务中，烤箱中的代码控制面包颜色的深浅，以及何时将烤好的面包弹出；住宅的电力供应由程序控制和调节；能源使用的账单均由软件记录输出。实际上，我们可以使用自动化程序支付电费账单、订购更多食品，甚至购买一台新烤箱！现在，几乎在我们生活的各个方面，包括影响我们健康和福利的关键系统，软件都发挥着作用。正因为如此，软件工程比以往任何时候都更加重要。好的软件工程实践必须确保软件在我们的生活中发挥积极的作用。

本书强调软件工程中的关键问题，描述我们所了解的技术和工具，以及它们如何影响我们构建和使用的最终软件产品。我们将从理论和实践两个方面说明我们所了解的软件工程以及如何将其应用于通常的软件开发或维护项目中。我们也将研究那些我们还不了解但有助于使产品更加可靠、安全、有用、易理解的理论和实践。

首先，我们探讨如何分析问题以及寻求解决方案。然后，研究计算机科学问题与工程问题之间的区别。我们的最终目标是，生产出高质量软件，进而找到解决方案，并考虑那些对质量有影响的特性。

我们还将探讨软件系统的开发人员已经取得了多大的成功。通过对几个软件失效的例子进行研究，可以了解在掌握高质量软件开发的艺术方面，软件系统的开发人员已经取得的进展以及还需要在哪些方面做出努力。

接着，探讨软件开发涉及的人员。在描述客户、用户和开发人员的角色和责任之后，会转而研究系统本身。我们看到，可以把一个系统看作是与活动相关的一组对象，并且处于某个边界之内。或者，我们从工程师的角度考虑系统：可以像盖房子一样开发一个系统。在定义了构造系统的步骤以后，将讨论每个步骤中开发团队的角色。

最后，讨论影响我们实践软件工程方式的一些变化，给出Wasserman的将实践融为一体的8个概念。

1.1 什么是软件工程

软件工程师要利用与计算机和计算相关的知识来解决问题。通常情况下，我们要处理的问题是与计算机或现有的计算机系统相关的。但是，也有时候，解决问题的潜在困难与计算机无关。因此，首先要理解问题的本质，这是至关重要的。尤其是，我们必须十分谨慎，不要把计算机器或技术按我们的意愿强加给遇到的每个问题。我们必须首先解决这个问题。然后，如果需要的话，再把技术