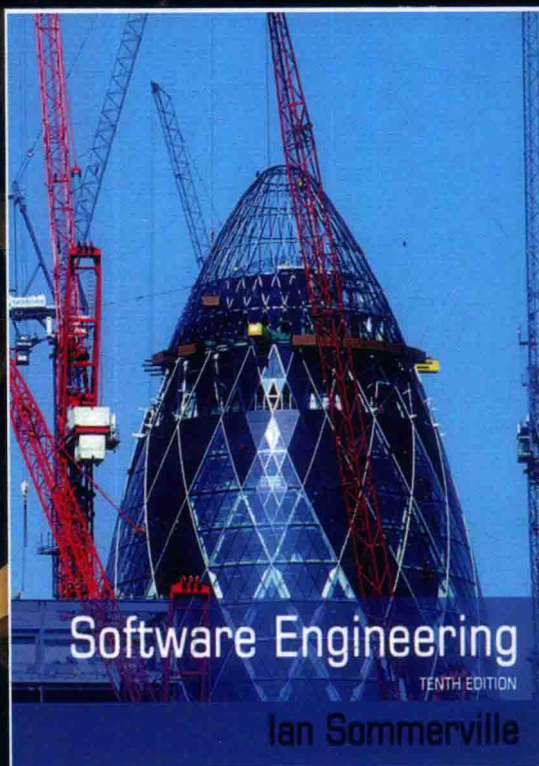


软件工程

[英] 伊恩·萨默维尔 (Ian Sommerville) 著

彭鑫 赵文耘 等译

Software Engineering
Tenth Edition



计 算 机 科 学 丛 书

原书第10版

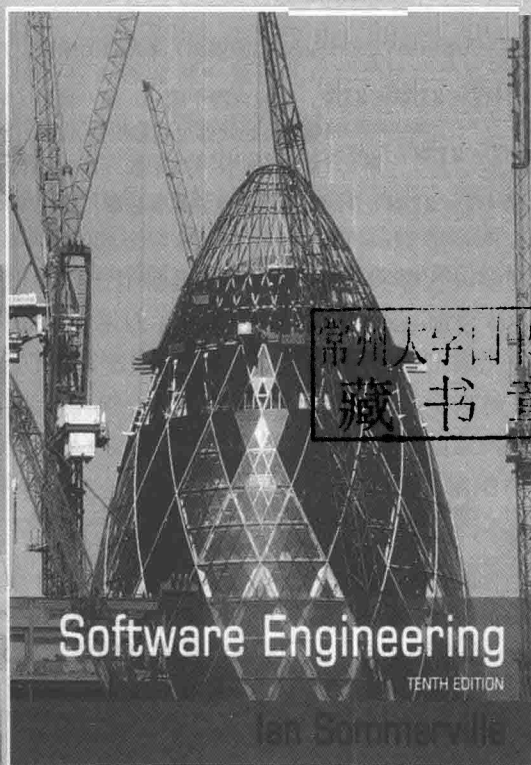
软件工程

[英] 伊恩·萨默维尔 (Ian Sommerville) 著

彭鑫 赵文耘 等译

Software Engineering

Tenth Edition



机械工业出版社
China Machine Press

图书在版编目(CIP)数据

软件工程(原书第10版)/(英)伊恩·萨默维尔(Ian Sommerville)著;彭鑫等译. —北京:机械工业出版社,2018.1

(计算机科学丛书)

书名原文:Software Engineering, Tenth Edition

ISBN 978-7-111-58910-5

I. 软… II. ①伊… ②彭… III. 软件工程—高等学校—教材 IV. TP311.5

中国版本图书馆CIP数据核字(2018)第003918号

本书版权登记号:图字 01-2015-2931

Authorized translation from the English language edition, entitled *Software Engineering, Tenth Edition*, 978-0-13-394303-0 by Ian Sommerville, published by Pearson Education, Inc., Copyright © 2016, 2011, 2006.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press Copyright © 2018.

本书中文简体字版由Pearson Education(培生教育出版集团)授权机械工业出版社在中华人民共和国境内(不包括香港、澳门特别行政区及台湾地区)独家出版发行。未经出版者书面许可,不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

本书是系统介绍软件工程理论的经典教材,共包含四个部分:第一部分(第1~9章)是对软件工程的一般性介绍,介绍了软件工程的一些重要概念(如软件过程和敏捷方法),描述了基本的软件开发活动(从需求规格说明一直到系统演化);第二部分(第10~14章)关注软件系统可依赖性和信息安全等重要话题;第三部分(第15~21章)介绍更高级的软件工程话题;第四部分(第22~25章)关注技术管理问题。

本书适合作为软件和系统工程专业本科生或研究生教材,同时也是软件工程师难得的参考书。

出版发行:机械工业出版社(北京市西城区百万庄大街22号 邮政编码:100037)

责任编辑:迟振春

责任校对:李秋荣

印刷:中国电影出版社印刷厂

版次:2018年2月第1版第1次印刷

开本:185mm×260mm 1/16

印张:33.25

书号:ISBN 978-7-111-58910-5

定价:89.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991 88361066

投稿热线:(010) 88379604

购书热线:(010) 68326294 88379649 68995259

读者信箱:hjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问:北京大成律师事务所 韩光/邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brain W. Kernighan, Dennis Ritchie, Jim Gray, Afred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

很荣幸能从北京理工大学程成老师手中接过本书的翻译任务。程成老师从第6版开始承担本书的翻译任务，直至第9版，为我们这次第10版的翻译打下了良好的基础。

作为一本经典的软件工程教材，本书的内容非常全面。其中，第一部分覆盖了基本的软件开发生命周期，可以作为本科软件工程课程教学的基本内容。第二部分针对高可信软件系统对于可依赖性和信息安全的高要求，介绍了相应的软件工程方法和技术，体现了软件工程对于大规模复杂软件系统的质量保障作用。第三和第四部分则分别针对软件工程中的一些高级技术问题和开发管理问题进行了介绍。后三个部分的内容为面向高年级本科生和研究生的高级软件工程课程以及软件开发管理、软件可靠性工程等其他更深入的软件工程专业课程提供了教学内容。

难能可贵的是，本书每一次改版都能根据软件工程方法、技术及产业实践的最新发展更新并调整相关内容。本版在第9版的基础上，更新了关于敏捷软件工程的相关内容，增加了RESTful 服务、系统之系统等反映软件开发技术和复杂系统发展趋势的新内容，重新组织了与高可信软件系统密切相关的多个章节，同时将面向方面的软件工程以及过程改进等不太重要的章节移到了网站上。

本书系统反映了工程化软件开发所具有的严谨性和规范性，但同时并不教条。例如，虽然系统性地介绍了UML 建模方法，但在谈到UML 在软件设计过程中的作用时，作者指出非正式的表达法在软件设计过程中可能更有用。因此，我们在学习本书所介绍的软件工程方法和规范的同时，也不要忘了与软件开发实践相结合，在做中学，在实践中领会。

本书主要由彭鑫、赵文耘翻译。参与本书翻译工作的还有复旦大学计算机科学技术学院研究生刘汶涑、周翔、姜清涛、陈驰、黄凯峰等。此外，机械工业出版社华章公司的朱劼、迟振春两位编辑还认真审校了译稿，在此一并表示感谢。

由于时间仓促以及译者自身知识和水平有限，译稿中难免存在错误和遗漏，在此向广大读者表示歉意，敬请批评指正！

软件工程在过去 50 年之中取得了令人瞩目的进展。我们的社会已经无法在缺少大型专业化软件系统的情况下正常运转了。国家的公用事业和基础设施（如能源、通信、交通）全都依赖于复杂且可靠的计算机系统。软件使得我们可以探索空间，创造万维网这一人类历史上最重要的信息系统。智能手机和平板电脑无处不在，而为这些设备开发软件的整个“应用开发产业”已经在过去几年中悄然形成。

人类现在正面临着一系列迫切的挑战——气候变化和极端天气、自然资源的减少、需要为更多的人口提供食物和住房、国际恐怖主义的威胁，以及为老年人提供令人满意的生活。我们需要新技术来帮助我们应对这些挑战，可以肯定的是软件将在这些技术中扮演核心角色。因此，软件工程对于我们在这个星球上的未来极其重要。我们必须继续培养软件工程师并推动软件工程学科的持续发展，从而满足开发更多的软件系统以及创造我们所需要的越来越复杂的未来系统的需要。

当然，软件项目还存在很多问题。系统仍然有时会延迟交付并且成本超支。我们正在创造越来越复杂的软件系统之系统（software systems of systems），在这条道路上遇到各种困难也是不足为奇的。然而，我们不应该让这些问题掩盖软件工程领域已经取得的巨大成就，以及所形成的各种令人印象深刻的软件工程方法和技术。

本书的不同版本已经有超过 30 年的历史，而这一版同样遵循了本书第 1 版中所建立的基本原则：

1. 按照工业界实践介绍软件工程，不对任何特定的方法（例如敏捷开发、形式化方法）持倾向性态度。在现实中，工业界往往将各种技术（例如敏捷以及基于计划的开发）混合在一起使用，这一点也在本书中有所反映。

2. 根据我所知道的以及所理解的知识介绍软件工程。很多人建议我增加并详细介绍其他相关话题，例如开源软件开发、UML 的使用以及移动软件工程等，但是我对这些领域的了解并不多，我个人的工作主要是在系统可靠性和系统工程方面，这一点在我为本书所选择的高级专题中有所反映。

我认为现代软件工程的关键问题是管理复杂性，将敏捷和其他方法结合起来，并确保我们的系统安全以及有韧性。这些问题是我在这一版中修改和新增内容的主要因素。

对第 9 版的修改

这一版相比第 9 版的更新和新增内容汇总如下：

- 全面更新了关于敏捷软件工程的章节，增加了关于 Scrum 的新内容。此外还根据需要对其他章节进行了更新，以反映敏捷方法在软件工程中日益增长的应用。
- 增加了关于韧性工程、系统工程、系统之系统的新章节。
- 对于涉及可靠性、安全、信息安全的 3 章进行了彻底的重新组织。
- 在第 18 章“面向服务的软件工程”中增加了关于 RESTful 服务的新内容。
- 更新和修改了关于配置管理的章节，增加了关于分布式版本控制系统的新内容。
- 将关于面向方面的软件工程以及过程改进的章节移到了本书的配套网站上。
- 在网站上新增了补充材料，包括一系列支持视频。我在视频中对于一些关键话题进

行了解释,并且推荐了相关的 YouTube 视频。

这一版保留了此前版本中的四部分结构,但我对其中每个部分都进行了大量的修改。

1. 在第一部分软件工程导论中,我彻底重写了第 3 章(敏捷方法)并对其进行了更新,以反映 Scrum 方法在实践中日益增长的使用。第 1 章增加了一个关于数字化学习环境的案例研究,这个案例在其他几个章节中也会用到。第 9 章更加详细地介绍了遗留系统。这一部分的其他章节也都进行了少量的修改和更新。

2. 第二部分介绍系统可依赖性。这一部分进行了修改和重新组织,不再按照面向活动的方式进行组织,而是将安全、信息安全、可靠性分散在多个章节中。这使得相关内容(例如信息安全)可以更加方便地作为独立的专题在更加综合性的课程中使用。我增加了关于韧性工程的一章,其中涉及网络安全、组织韧性以及韧性系统设计。

3. 第三部分增加了关于系统工程、系统之系统的新章节,并对与面向服务的系统工程相关的内容进行了全面修改,以反映 RESTful 服务的使用日益增长的趋势。与面向方面的软件工程相关的章节移到了本书的配套网站上。

4. 第四部分对配置管理的内容进行了更新,以反映分布式版本控制工具(如 Git)的使用日益增长的趋势。过程改进相关的章节移到了本书的配套网站上。

本书补充材料中的一个重要变化是为每个章节增加了视频推荐。我制作了关于一系列主题的 40 多段视频,放在我的 YouTube 频道上并可从本书的网页上链接过去。对于没有制作视频的地方,我推荐了一些可能有用的 YouTube 视频。

我在下面这个视频中解释了这一版中所做修改背后的原因。

<http://software-engineering-book/videos/10th-edition-changes>

读者对象

本书主要面向各大学和学院正在学习软件和系统工程初高级课程的学生。我假设读者对于编程基础和基本数据结构都已有所理解。

工业界的软件工程师也会发现这是一本很好的读物,能帮助他们在软件复用、体系结构设计、可依赖性和信息安全性以及系统工程等方面获得新的知识。

教学建议

针对三种不同类型的软件工程课程,我对本书进行了如下设计:

- 软件工程一般导论课程。本书的第一部分专门用于一个学期的软件工程导论课程。这部分包括 9 章,涵盖了软件工程领域的基础内容。如果你的课程中包含实践性教学环节,那么可以选讲第四部分中关于管理的章节。
- 软件工程特定主题的导论课程或进阶课程。通过使用本书第二~四部分的内容,可以创建一系列更高级的课程。例如,我采用第二部分的各章加上关于系统工程和质量管理的两章来讲授以关键系统为主题的课程。而对于讨论软件密集型系统的课程,我选择的章节涉及系统工程、需求工程、系统之系统、分布式软件工程、嵌入式软件、项目管理和项目计划。
- 软件工程特定主题的更高阶课程。对于这类课程,本书的各章可以构成课程的基础,然后辅之以更多的阅读以便进一步探索某个主题。例如,关于软件复用的课程就可以基于第 15 ~ 18 章的内容。

用书教师可以访问培生网站获取相关教辅资源,网址为 <http://www.pearsonhighered.com/>

sommerville^①。部分资源为加密内容,可在网站上通过注册来获取密码。教辅资源包括:

- 部分章末练习题的答案。
- 每章的测验题目及答案。

配套网站

本书采用印刷和在线网站内容相结合的方式,其中印刷版中的核心信息可以链接到网站上的补充材料。有些章节包含特别编写的在线段落以提供更多的信息。在线网站上还有 6 个在线章节,介绍了若干我在本书的印刷版中未介绍的主题。

读者可以从本书的网站(software-engineering-book.com)上下载丰富的补充材料,包括:

- 本书所有章节的 PPT;
- 我所录制的针对一系列软件工程主题的视频,我还推荐了一些有助于学习本书内容的 YouTube 视频;
- 针对课程教师的指南,其中给出了在教授不同课程时如何使用本书的建议;
- 关于本书中案例研究(胰岛素泵、心理健康保健系统、野外气象站系统、数字化学习系统)的附加信息,以及其他一些案例研究(例如阿丽亚娜 5 型运载火箭失效);
- 6 个在线章节,介绍了过程改进、形式化方法、交互设计、应用体系结构、文档化以及面向方面的开发;
- 为每个章节提供补充内容的在线段落,这些在线段落可以通过每一章中用方框突出显示的链接来访问;
- 附加的涉及一系列系统工程主题的 PPT。

应本书读者的要求,我已经在本书的网站上发布了其中一个系统案例研究的完整需求规格说明。对于学生而言,获得这种文档并理解其结构和复杂性有些困难。为了避免保密问题,我对这个来自于真实系统的需求文档进行了再工程,因此使用该文档没有任何限制。

联系信息

网站: software-engineering-book.com

邮件: 名字, software.engineering.book@gmail.com

博客: iansommerville.com/systems-software-and-technology

YouTube: youtube.com/user/SoftwareEngBook

Facebook: facebook.com/sommerville.software.engineering

Twitter: @SoftwareEngBook 或者 @iansommerville (获取更广泛的推文)

请在 Twitter 或 Facebook 上关注我,以便及时获得关于软件和系统工程的新材料和新评论。

致谢

多年以来许多人都对本书的演进做出了贡献,在此我想感谢每一位曾经评论过本书此前版本并且给出了建设性意见的人(审阅人、学生、读者)。我想特别感谢我的家人 Anne、Ali 和 Jane,感谢他们在我编写本书(以及此前所有版本)过程中对我的爱、帮助和支持。

Ian Sommerville

2014 年 9 月

① 关于本书教辅资源,只有使用本书作为教材的教师才可以申请,需要的教师请联系机械工业出版社华章公司,电话 136 0115 6823,邮箱 wanguang@hzbook.com。——编辑注

出版者的话

译者序

前言

第一部分 软件工程导论

第 1 章 概述 2

1.1 专业化软件开发 3

1.1.1 软件工程 5

1.1.2 软件工程的多样性 6

1.1.3 互联网软件工程 8

1.2 软件工程职业道德 9

1.3 案例研究 11

1.3.1 胰岛素泵控制系统 12

1.3.2 心理健康治疗病人信息系统 13

1.3.3 野外气象站 14

1.3.4 学校数字化学习环境 15

要点 16

阅读推荐 17

网站 17

练习 17

参考文献 18

第 2 章 软件过程 19

2.1 软件过程模型 20

2.1.1 瀑布模型 21

2.1.2 增量式开发 23

2.1.3 集成与配置 25

2.2 过程活动 26

2.2.1 软件规格说明 26

2.2.2 软件设计和实现 27

2.2.3 软件确认 29

2.2.4 软件演化 30

2.3 应对变化 31

2.3.1 原型 32

2.3.2 增量式交付 33

2.4 过程改进 34

要点 36

阅读推荐 36

网站 37

练习 37

参考文献 37

第 3 章 敏捷软件开发 39

3.1 敏捷方法 40

3.2 敏捷开发技术 42

3.2.1 用户故事 43

3.2.2 重构 44

3.2.3 测试先行的开发 45

3.2.4 结对编程 46

3.3 敏捷项目管理 47

3.4 敏捷方法的伸缩 50

3.4.1 敏捷方法的实践问题 51

3.4.2 敏捷和计划驱动的方法 52

3.4.3 面向大型系统的敏捷方法 54

3.4.4 面向整个组织的敏捷方法 56

要点 57

阅读推荐 57

网站 58

练习 58

参考文献 58

第 4 章 需求工程 60

4.1 功能性需求和非功能性需求 62

4.1.1 功能性需求 63

4.1.2 非功能性需求 64

4.2 需求工程过程 66

4.3 需求抽取 67

4.3.1 需求抽取技术 69

4.3.2 故事和场景	71	6.3.1 分层体系结构	111
4.4 需求规格说明	73	6.3.2 知识库体系结构	112
4.4.1 自然语言规格说明	74	6.3.3 客户-服务器体系结构	113
4.4.2 结构化规格说明	75	6.3.4 管道和过滤器体系结构	115
4.4.3 用况	76	6.4 应用体系结构	116
4.4.4 软件需求文档	77	6.4.1 事务处理系统	117
4.5 需求确认	79	6.4.2 信息系统	118
4.6 需求变更	80	6.4.3 语言处理系统	119
4.6.1 需求管理计划	81	要点	121
4.6.2 需求变更管理	82	阅读推荐	122
要点	83	网站	122
阅读推荐	83	练习	122
网站	84	参考文献	123
练习	84		
参考文献	85		
第5章 系统建模	86	第7章 设计和实现	125
5.1 上下文模型	87	7.1 使用UML的面向对象设计	126
5.2 交互模型	89	7.1.1 系统上下文和交互	126
5.2.1 用况建模	90	7.1.2 体系结构设计	128
5.2.2 顺序图	91	7.1.3 对象类识别	129
5.3 结构模型	93	7.1.4 设计模型	130
5.3.1 类图	93	7.1.5 接口规格说明	133
5.3.2 泛化	95	7.2 设计模式	134
5.3.3 聚集	95	7.3 实现问题	136
5.4 行为模型	96	7.3.1 复用	137
5.4.1 数据驱动的建模	96	7.3.2 配置管理	138
5.4.2 事件驱动的建模	97	7.3.3 宿主机-目标机开发	139
5.4.3 模型驱动的工程	99	7.4 开源开发	141
5.5 模型驱动的体系结构	100	7.4.1 开源许可证	142
要点	102	要点	143
阅读推荐	103	阅读推荐	143
网站	103	网站	144
练习	103	练习	144
参考文献	104	参考文献	145
第6章 体系结构设计	105	第8章 软件测试	146
6.1 体系结构设计决策	107	8.1 开发测试	149
6.2 体系结构视图	109	8.1.1 单元测试	150
6.3 体系结构模式	110	8.1.2 选择单元测试用例	151
		8.1.3 构件测试	153
		8.1.4 系统测试	155

8.2 测试驱动的开发	157	第 11 章 可靠性工程	200
8.3 发布测试	159	11.1 可用性和可靠性	202
8.3.1 基于需求的测试	159	11.2 可靠性需求	203
8.3.2 场景测试	160	11.2.1 可靠性度量	204
8.3.3 性能测试	161	11.2.2 非功能性可靠性需求	205
8.4 用户测试	161	11.2.3 功能性可靠性规格说明	207
要点	163	11.3 容错体系结构	207
阅读推荐	164	11.3.1 保护性系统	208
网站	164	11.3.2 自监控系统体系结构	208
练习	164	11.3.3 <i>N</i> 版本编程	210
参考文献	165	11.3.4 软件多样性	211
第 9 章 软件演化	166	11.4 可靠性编程	212
9.1 演化过程	168	11.5 可靠性度量	216
9.2 遗留系统	170	11.5.1 运行概况	218
9.2.1 遗留系统管理	173	要点	218
9.3 软件维护	176	阅读推荐	219
9.3.1 维护预测	178	网站	219
9.3.2 软件再工程	179	练习	219
9.3.3 软件重构	181	参考文献	220
要点	182	第 12 章 安全工程	221
阅读推荐	182	12.1 安全关键系统	222
网站	183	12.2 安全需求	224
练习	183	12.2.1 危险识别	225
参考文献	183	12.2.2 危险评估	226
第二部分 系统可依赖性和信息安全		12.2.3 危险分析	227
第 10 章 可依赖系统	186	12.2.4 风险降低	229
10.1 可依赖性属性	187	12.3 安全工程过程	229
10.2 社会技术系统	189	12.3.1 安全保证过程	230
10.2.1 规章与守约	191	12.3.2 形式化验证	232
10.3 冗余和多样性	192	12.3.3 模型检测	233
10.4 可依赖的过程	193	12.3.4 静态程序分析	234
10.5 形式化方法与可依赖性	195	12.4 安全案例	235
要点	197	12.4.1 结构化论证	237
阅读推荐	198	12.4.2 软件的安全论证	238
网站	198	要点	240
练习	198	阅读推荐	241
参考文献	199	网站	241
		练习	241
		参考文献	242

第 13 章 信息安全工程	244
13.1 信息安全和可依赖性	245
13.2 信息安全和组织	248
13.2.1 信息安全风险评估	249
13.3 信息安全需求	249
13.3.1 滥用案例	252
13.4 信息安全系统设计	253
13.4.1 设计风险评估	254
13.4.2 体系结构设计	257
13.4.3 设计准则	259
13.4.4 信息安全系统编程	263
13.5 信息安全测试和保证	263
要点	265
阅读推荐	265
网站	265
练习	266
参考文献	266

第 14 章 韧性工程	268
14.1 网络安全	270
14.2 社会技术韧性	273
14.2.1 人为错误	275
14.2.2 运行和管理过程	277
14.3 韧性系统设计	279
要点	284
阅读推荐	284
网站	285
练习	285
参考文献	285

第三部分 高级软件工程

第 15 章 软件复用	288
15.1 复用概览	290
15.2 应用框架	292
15.3 软件产品线	294
15.4 应用系统复用	298
15.4.1 可配置的应用系统	299
15.4.2 集成的应用系统	301
要点	303

阅读推荐	303
网站	304
练习	304
参考文献	305

第 16 章 基于构件的软件工程	306
16.1 构件和构件模型	307
16.1.1 构件模型	310
16.2 CBSE 过程	311
16.2.1 面向复用的 CBSE	312
16.2.2 基于复用的 CBSE	314
16.3 构件组装	316
要点	321
阅读推荐	321
网站	322
练习	322
参考文献	322

第 17 章 分布式软件工程	324
17.1 分布式系统	325
17.1.1 交互模型	327
17.1.2 中间件	329
17.2 客户-服务器计算	330
17.3 分布式系统的体系结构模式	331
17.3.1 主从体系结构	332
17.3.2 两层客户-服务器体系结构	332
17.3.3 多层客户-服务器体系结构	334
17.3.4 分布式构件体系结构	335
17.3.5 对等体系结构	338
17.4 软件即服务	340
要点	342
阅读推荐	343
网站	343
练习	343
参考文献	344

第 18 章 面向服务的软件工程	345
18.1 面向服务的体系结构	348

18.1.1 SOA 中的服务构件	349	要点	404
18.2 RESTful 服务	351	阅读推荐	405
18.3 服务工程	354	网站	405
18.3.1 可选服务识别	354	练习	405
18.3.2 服务接口设计	356	参考文献	406
18.3.3 服务实现和部署	358		
18.4 服务组合	359	第 21 章 实时软件工程	407
18.4.1 workflow 设计与实现	361	21.1 嵌入式系统设计	408
18.4.2 服务组合测试	363	21.1.1 实时系统建模	411
要点	364	21.1.2 实时编程	413
阅读推荐	364	21.2 实时软件体系结构模式	413
网站	365	21.2.1 观察和反应模式	414
练习	365	21.2.2 环境控制模式	415
参考文献	366	21.2.3 处理管道模式	417
第 19 章 系统工程	367	21.3 时序分析	418
19.1 社会技术系统	370	21.4 实时操作系统	421
19.1.1 涌现特性	372	21.4.1 进程管理	422
19.1.2 不确定性	373	要点	423
19.1.3 成功准则	374	阅读推荐	424
19.2 概念设计	375	网站	424
19.3 系统采购	377	练习	424
19.4 系统开发	380	参考文献	425
19.5 系统运行和演化	383		
19.5.1 系统演化	384	第四部分 软件管理	
要点	385	第 22 章 项目管理	428
阅读推荐	385	22.1 风险管理	430
网站	385	22.1.1 风险识别	431
练习	386	22.1.2 风险分析	432
参考文献	386	22.1.3 风险计划	433
第 20 章 系统之系统	388	22.1.4 风险监控	434
20.1 系统复杂度	390	22.2 人员管理	435
20.2 系统之系统的分类	393	22.2.1 激励人员	435
20.3 还原论和复杂系统	395	22.3 团队协作	437
20.4 系统之系统工程	397	22.3.1 成员的挑选	439
20.4.1 接口开发	398	22.3.2 小组的结构	440
20.4.2 集成和部署	399	22.3.3 小组的沟通	442
20.5 系统之系统的体系结构	400	要点	443
20.5.1 系统之系统的体系结构模式	402	阅读推荐	443
		网站	443

练习	444	24.2.1 ISO 9001 标准框架	473
参考文献	444	24.3 评审与审查	475
第 23 章 项目计划	445	24.3.1 评审过程	475
23.1 软件报价	446	24.3.2 程序审查	476
23.2 计划驱动的开发	447	24.4 质量管理与敏捷开发	478
23.2.1 项目计划	448	24.5 软件度量	479
23.2.2 计划过程	449	24.5.1 产品量度	481
23.3 项目进度安排	450	24.5.2 软件构件分析	483
23.3.1 进度安排表示方法	451	24.5.3 度量歧义	484
23.4 敏捷计划	453	24.5.4 软件解析	485
23.5 估算技术	455	要点	486
23.5.1 算法成本建模	456	阅读推荐	486
23.6 COCOMO 成本建模	458	网站	487
23.6.1 应用组合模型	459	练习	487
23.6.2 早期设计模型	460	参考文献	487
23.6.3 复用模型	460	第 25 章 配置管理	489
23.6.4 后体系结构模型	462	25.1 版本管理	492
23.6.5 项目的工期和人员配备	464	25.2 系统构建	495
要点	464	25.3 变更管理	499
阅读推荐	465	25.4 发布版本管理	503
网站	465	要点	505
练习	465	阅读推荐	505
参考文献	467	网站	506
第 24 章 质量管理	468	练习	506
24.1 软件质量	469	参考文献	506
24.2 软件标准	471	术语表	507

软件工程导论

这一部分的目标是对软件工程做一般性概述，各章可以用于一个学期的软件工程导论性质的课程。这部分介绍软件工程的一些重要概念（如软件过程和敏捷方法），描述基本的软件开发活动（从需求规格说明一直到系统演化）。

第1章是一个概述，介绍了专业软件工程并定义了一些软件工程的概
念。这一章还就软件工程职业道德方面的问题进行了简要的讨论。软件工程师认真思考所从事的工作的深远影响是很重要的。这一章还介绍了本书中所使用的4个案例研究，分别是：管理接受心理健康问题治疗的病人记录的信息系统（Mentcare），便携式胰岛素泵控制系统，野外气象站嵌入式系统，数字化学习环境（iLearn）。

第2章和第3章分别介绍软件工程过程和敏捷开发。第2章介绍软件过程模型，例如瀑布模型，以及这些过程中的基本活动。第3章补充了一个关于软件工程的敏捷开发方法的讨论，并在上一版的基础上进行了全面的修改，主要关注使用Scrum的敏捷开发以及对于敏捷实践的讨论（例如，用于需求定义的故事以及测试驱动开发）。

这部分的其他各章是对第2章中所介绍的软件过程活动的深入阐述。第4章介绍需求工程这一十分重要的话题，其中定义了系统应该做什么的需求。第5章介绍如何使用UML对系统进行建模，主要关注使用用例图、类图、顺序图 and 状态图来建模软件系统。第6章对软件体系结构的重要性进行了讨论，并介绍了如何在软件设计中使用体系结构模式。

第7章介绍面向对象的设计和模式的使用，还介绍了重要的实现问题——复用、配置管理、宿主-目标机开发，并且讨论了开源软件开发。第8章主要介绍软件测试，从系统开发时的单元测试到软件的发布测试。此外，第8章还讨论了测试驱动开发的使用，这是在敏捷方法中率先使用的一种方法，现在已经得到广泛应用。最后，第9章简要介绍了软件演化问题，包括演化过程、软件维护和遗留系统管理。

概述

目标

本章的目标是介绍软件工程的概
念，并为理解本书其他部分内容提供一个框架。阅读完本章后，你将：

- 理解软件工程是什么，为什么它很重要；
- 理解开发不同类型的软件系统可能需要不同的软件工程技术；
- 理解对于软件工程师很重要的道德和职业问题；
- 了解4个不同类型的软件系统，全书都将以它们为例。

软件工程对于政府、社会、国内和国际企业、机构的正常运转都是至关重要的。现在的世界离开软件就无法运转了。国家基础设施和公共服务都是由基于计算机的系统控制的，大多数电子产品都包括计算机和控制软件。工业制造和分销已经完全计算机化了，金融系统也是这样。娱乐业，包括音乐产业、计算机游戏产业、电影和电视产业，也是一个软件密集型的产业。全世界75%以上的人口都有一台软件控制的移动电话，到2016年其中的大多数都能够连接互联网。

软件是抽象的、不可触摸的，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：一方面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；另一方面，由于缺乏自然约束，软件系统也就很容易变得极为复杂，理解它会很困难，改变它价格高昂。

有很多不同类型的软件系统，从简单的嵌入式系统到复杂的全球范围的信息系统。没有放之四海而皆准的软件工程表示法、方法或技术，因为不同类型的软件需要不同的方法。开发一个组织的信息系统和开发一个科学仪器的控制器是完全不同的，而这些系统都跟图形密集型的计算机游戏没有太多的共同点。所有这些应用都需要软件工程，但并不都是需要相同的软件工程方法和技术。

现在仍有许多有关软件项目出问题和“软件失效”的报道。软件工程因不能充分支持现代软件的开发而遭受批评。然而，在我看来，这些所谓的软件失效很多都源于以下两方面的因素：

1. 不断增长的系统复杂性。随着新的软件工程技术可以帮助我们构建更大、更复杂的系统，要求在发生变化。系统必须更快地构建和交付；需要更大甚至更复杂的系统；系统必须具备在以前看来不可能实现的功能。必须不断发展新的软件工程技术来迎接交付更复杂的软件的新挑战。

2. 未有效采用软件工程方法。不采用软件工程的方法和技术去编写计算机程序是容易的。许多公司都是随着他们的产品和服务的逐步发展而逐渐陷入软件开发中的。他们在日常工作中不使用软件工程方法，结果导致他们的软件经常比预计的费用高而可靠性低于预期。我们需要更好的软件工程教育和训练来应对这一问题。

软件工程师应该为自己所取得的成就而感到自豪。虽然我们在开发复杂软件时还存在一些问题,但如果没有软件工程,我们就不能探索太空,无法使用互联网和现代的电信服务,而各种形式的旅行则会更加危险和昂贵。人类在 21 世纪所面临的挑战是气候变化、自然资源不足、人口结构变化以及世界人口膨胀。我们将依赖软件工程来开发应对这些问题所需的系统。



软件工程的历史

“软件工程”这一概念是在 1968 年召开的一个讨论所谓的“软件危机”问题的会议上首次被提出的 (Naur and Randell 1969)。当时,人们认识到单一的程序开发技术已经不能适应大型复杂软件系统的开发。这些软件项目经常不可靠、费用高于预期,并且延迟交付。

20 世纪七八十年代,各种新的软件工程技术和方法陆续出现,例如结构化编程、信息隐藏和面向对象开发。各种工具和标准的表示法陆续出现,并逐渐成为现代软件工程的基础。

<http://software-engineering-book.com/web/history/>

1.1 专业化软件开发

许多人都在编写程序。业务人员编写电子表格程序来简化工作;科学家和工程师编写程序来处理实验数据;业余爱好者出于自己的兴趣和爱好而编写程序。然而,大多数软件开发都是一种专业化的活动,软件的开发是为了满足某种业务目标,为了嵌入其他设备中,或者作为软件产品(例如信息系统、计算机辅助设计系统)。专业化开发与个人化开发的关键区别在于,专业化软件除了开发者之外还有其他用户会使用,而且专业化软件通常都是由团队而非个人开发的。专业化软件在其生命周期内要不断维护和修改。

软件工程的目的是支持专业化的软件开发,而非个人编程。它包括支持程序规格说明、设计和演化的相关技术,而这些通常都与个人化的软件开发无关。为了帮助你全面理解软件工程的含义,图 1-1 对一些常见的问题及其回答进行了总结。

很多人将软件等同于计算机程序,其实这种理解是很狭隘的。在我们讨论软件工程时,软件不仅包括程序本身,而且还包括所有使程序能够正常使用的相关文档、库、支持网站、配置数据等。一个专业化开发的软件系统通常远不止一个程序。系统可能包含多个程序以及用于设置这些程序的配置文件。系统还可能包括描述系统结构的系统文档、解释如何使用该系统的用户文档,以及告知用户下载最新产品信息的 Web 网站。

这是专业软件开发与业余软件开发的一个重要区别。如果你只是自己编写一个程序,而且除你自己之外没有别的用户的话,你就不用费心编写程序指南和设计文档等。然而,如果你的软件有其他用户并且其他工程师会进行修改的话,你就必须在程序源代码之外提供其他附加信息。

软件工程师关心的是软件产品(即能卖给客户的软件)的开发。软件产品有以下两类。