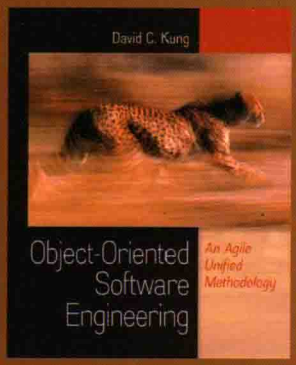

清华计算机图书·译丛

Object-Oriented Software Engineering

David C. Kung 著 牟永敏 邢颖 译

面向对象软件工程

清华大学出版社

清华计算机图书·译丛

Object-Oriented Software Engineering

David C. Kung 著 牟永敏 邢颖 译

面向对象软件工程

清华大学出版社

David C. Kung
Object-Oriented Software Engineering, First Edition
ISBN: 9780073376257

图书在版编目（CIP）数据

面向对象软件工程 /（美）David C.Kung 著；牟永敏，邢颖译. —北京：清华大学出版社，2017
（清华计算机图书译丛）
书名原文：Object-Oriented Software Engineering

ISBN 978-7-302-46094-7

Ⅰ. ①面…　Ⅱ. ①D…　②牟…　③邢…　Ⅲ. ①面向对象语言-程序设计　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2016）第 312621 号

责任编辑： 袁勤勇
封面设计： 傅瑞学
责任校对： 白　蕾
责任印制： 杨　艳

出版发行： 清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62795954
印 刷 者： 北京富博印刷有限公司
装 订 者： 北京市密云县京文制本装订厂
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　　张：** 35.5　　**字　　数：** 840 千字
版　　次： 2017 年 1 月第 1 版　　**印　　次：** 2017 年 1 月第 1 次印刷
印　　数： 1～2000
定　　价： 98.00 元

产品编号：057631-01

译 者 序

计算机已经渗透到社会生活的方方面面。计算机之所以能够被广泛使用，其背后的推动力就是市场经济。但是，实际上是软件在指挥着计算机按照人们想要的方式在工作。软件或者说计算机程序由成千上万条指令构成，这些指令指挥着计算机进行复杂的运算并且控制计算机硬件设备的运行。近些年对于计算机软件的需求快速增长。为了能够满足软件开发工程师或系统分析师的工作需求，学习软件工程的相关知识是非常必要的。

软件工程关注三个环节（软件开发过程、软件质量保证和软件项目管理）的行为，这些行为贯穿于软件生命周期中，并同时发生；而面向对象的软件工程（object-oriented software engineering，OOSE）是软件工程的一个专门学科。OOSE 将世界和各种系统看作是由相互联系和相互作用的对象构成。

在 20 世纪 80 年代，C++ 的迅速传播激起了对引导 OO 软件开发工作的开发方法的需求。提出了三个有影响的、在软件业内被广泛使用的 OO 软件开发方法，即 Booch 框图、对象建模技术（Object Modeling Technique，OMT）和用例工程。业界很快发现使用不同的方法将设计和实现的系统集成起来是一个非常大的挑战，原因就是不同的方法使用不同的建模概念和标记。为了解决这个问题，对象管理组织（Object Management Group，OMG）采用统一建模语言（Unified Modeling Language，UML）作为 OMG 标准。UML 是一个框图的集合，这些框图用于给一个 OO 系统的方方面面进行建模和设计。UML 框图用于需求分析阶段来帮助开发团队理解线性系统的业务流程，也作为设计说明的一部分用于设计阶段。本书大量展示了 UML 框图。

感谢德克萨斯大学阿灵顿分校的 David C. Kung 先生。敏捷过程、设计模式以及测试驱动的开发（test-driven development，TDD）激发了他写本书的巨大兴趣。敏捷过程强调团队合作、为改变而设计、对软件系统的每个小增量进行快速部署，以及与客户和用户共同开发。设计模式是对常见设计问题的有效解决方案，设计模式促进了软件重用并增进了团队沟通。总之，本文主要关注将 UML、设计模式、OO 软件测试、TDD 集成起来的敏捷统一方法学。

软件是你的作品，需要用心血去创作。我们可以用编程语言来准确反映我们的思想，我们应该做得更好，也可以做得更好。只要遵循科学的方法学，一定可以创作出好的软件作品。

本书由邢颖翻译，牟永敏统稿。由于译者水平有限，译文中的不当之处在所难免，真诚希望广大读者和同行不吝赐教，我们将不胜感激。

译 者

2016 年 11 月

序　言

背景

计算机被广泛应用于人类社会的方方面面，它与基于其运行的应用软件一起发挥了很多不同的作用。因此，软件工程师的岗位需求也在井喷式增长。2006 年 3 月的 *Money* 杂志将软件工程师评为美国最好的 50 个工作岗位的第一名。根据美国劳工统计局(Bureau of Labor Statistics，BLS)2010—2020 预测，应用程序开发工程师岗位将从 520 800 增长到 664 500(27.6%)，而系统分析师的岗位将从 544 400 增长到 664 800(22.1%)。为了能够满足对应用程序开发工程师或系统分析师的要求，接受软件工程方面的教育是非常有必要的。但是根据 BLS 发布的数据，2006 年在软件工程领域只授予了 160 个学士学位和 600 个硕士学位，而在计算机科学领域则授予了 10 289 个学士学位和 4512 个硕士学位。因此，在软件工程人才的需求和供给之间有巨大的鸿沟，尤其是研究生。

很多人都不清楚软件工程的研究范围以及它到底有什么用途，这门学科也经常被误解。许多媒体似乎将软件工程定义为写 Java 程序。有的学生认为软件工程包括与软件相关的一切。另外一些人则认为软件工程就是画 UML 框图，这个我们后面会提到。多年前，在第一节面向对象的软件工程(object-oriented software engineering，OOSE)课后，一位学生对我说："教授，你应该知道这门课对于我来说很容易，因为我们以前都画过太多的 UML 图了。"学期末，那个学生又来找我，他说："教授，我想说我们很努力地学习，但我们学的是 OO 设计。这门课并不是我想象的画 UML 图。"所以到底什么是软件工程呢？作为一门学科，它包含为了明显提高软件的生产率和质量，同时降低成本和投放到市场的时间所需的工程过程、方法、质量保证和项目管理研究、教育及应用。OOSE 是软件工程的一个分支，其特点是将世界和系统看成是相关联系和相互作用的独特视角。20 世纪 80 年代 C++ 语言的出现标志着 OOSE 时代的到来。从那时起，软件产品开始了其前所未有的世界范围的增长，并因统一建模语言(unified modeling language，UML)和统一过程(unified process，UP)的产生以及世界范围的应用而进一步加速增长。严格来说，软件过程描述了一些阶段以及在各个阶段应该做什么。它不会(详细)定义在每个阶段如何实施各种行为。类似 UML 这样的建模语言会定义用于交流和记录分析及设计思路的标记、语法和语义。UML 和 UP 都很好，也很必需，但是还不够。这是因为还没有介绍如何产生画 UML 图的分析和设计思路。

动机

为了填补上文提到的鸿沟，我们需要一个方法学，或者称之为"食谱"。与过程不同，方法学是对于一些步骤的详细描述，或者是如何实施一些行为使得初学者也可以学习从而生成并部署所需的软件系统。没有方法学，一名刚入门的软件工程师可能会花费数年在职的训练才能学会 OO 设计、实现和测试技巧。

写本书的动机也包括作者被敏捷过程、设计模式以及测试驱动的开发(test-driven development，TDD)所激发的巨大兴趣。敏捷过程强调团队合作、为改变而设计、对软件系

统的每个小增量进行快速部署，以及与客户和用户共同开发。设计模式是对常见设计问题的有效解决方案。设计模式促进了软件重用并增进了团队沟通。TDD 鼓励可测试的软件，并需要在软件实现前产生测试脚本从而使得软件可以立刻被测试，经常被测试。

这类似开发一个游乐场的思路。整个过程包括如下阶段：计划、公共支持、分析设计、筹措资金、画施工图纸、施工、采购设备、安装设备、试运行、剪彩。但是，光了解整个过程是不够的。开发团队必须知道如何实施各阶段的行为。例如，计划行为包括产生初步概念、可行性分析、总体规划。主题公园团队必须知道如何实施这些行为。分析设计行为包括从利益相关者那里获取需求、实地调查、公园布局设计、公园不同区域主题设计、建模从而研究布局设计和主题、进行总体规划。与描述不同阶段行为的过程不同，一套方法学详细描述了一些步骤，或者是如何实施这些行为。

游乐场的开发是一个数年的项目，且花费达数十亿美元。投资者希望公园越早产生收益越好，但是如果按照上述方法开发，投资者必须一直等到公园完工。由于传统过程所施加的约束，总体规划一旦结束就不能被轻易修改。而一旦公园完工，如果公园不能满足利益相关者的期待，则其改变的代价很大。

敏捷开发旨在解决这些问题。使用敏捷开发，主题公园初步需求被迅速获取，并允许在开发过程中进行演化。然后从中提取出游乐和娱乐设施的需求，并将其认真归类。生成计划来在相对短的时期内开发和部署这些分类设施，即小增量的快速部署。因此，不同于一刀切的总体规划，开发过程每次只设计和部署一类设施。在这些设施部署好并投入运营后，就寻求反馈，并与利益相关者制订出关于开发计划、预算和时间表的需求变化，即联合开发。另外，应用架构设计模式来提高公园对改变的适应能力和质量，也即为改变而设计。强调团队合作是因为团队之间以及团队成员之间有效的协作能够确保这些设施按时无缝地开发和部署。敏捷过程有一些优点：投资者能较早获得收益，因为这些设施能按照计划投入运营而且是可行的；由于少量的设施是同时开发和部署的，可以容易改正错误并进行改变。

总之，本文主要关注将 UML、设计模式、OO 软件测试、TDD 集成起来的敏捷统一方法学。本书中的方法学叫做“统一方法学”，因为它使用 UML 作为建模语言并遵循敏捷统一过程。当然这并不意味着对于所有项目都将其他方法统一起来或被当成一个“统一”方法使用。

读者

本书的读者为计算机科学或软件工程的学生，以及软件开发的专门人士。本书尤其适合作为高年级本科生的基础教材，以及研究生课程和 IT 业专业培训课程的入门教材。本书包含过去十年间的很多内容，既有在美国以及全球的大学和公司里讲授的课程，也有行业内外软件工程项目的应用素材。这些素材让我能够近距离观察学生和软件工程师如何应用 UP、UML、设计模式和 TDD，以及他们所面对的困难。他们的反馈让我能够持续更新本书的内容。

组织

本书有 24 章，被分为以下 8 个部分：

第 1 部分　引言与系统工程。这部分由前 3 章构成，它提供了观察软件生命周期行为的视角。尤其是，它覆盖了软件过程模型、方法学的概念、过程与方法的区别以及系统工程。

第 2 部分　分析和体系结构设计。这部分介绍了计划阶段的行为，包括需求提取、领域建模和架构设计。

第 3 部分　交互式系统的建模和设计。这部分主要处理建模与交互系统设计问题，共包括 6 章。这 6 章介绍了如何根据需求识别出用例、如何对参与者-系统交互以及对象交互行为进行建模和设计、如何应用职责分配模式、如何得到作为设计蓝本的设计类框图，以及如何设计用户交互界面。

第 4 部分　其他类型系统的建模和设计。这部分包括 3 章，每章介绍一种类型系统的建模与设计。具体来说，第 13 章介绍事件驱动系统的建模与设计；第 14 章介绍迁移系统的建模与设计；第 15 章介绍基于业务规则的系统的建模与设计。

第 5 部分　应用情景特定模式。这部分由两章构成，主要介绍如何应用情景特定模式。其中使用了一个案例分析，即状态图编辑器的设计来帮助理解这个过程。

第 6 部分　实现和质量保证。这部分包括 3 章，包括实现时要考虑的问题、软件质量保证的概念和行为以及软件测试。

第 7 部分　维护和配置管理。这部分包括两章，内容覆盖了软件维护与软件配置管理。

第 8 部分　项目管理和软件安全。本书的最后一部分由两章构成，一章介绍软件项目管理，另一章介绍软件安全，包括涉及安全软件系统的建模与设计的生命周期行为。

本书素材可以满足若干软件工程课程的需求，例如：

1. 第 1～3 部分与第 6～8 部分的一些主题对于面向对象的软件工程(OOSE)或软件工程课程的入门，都是一个很好的组合。既可以作为大学本科的课程，也可以作为入门的研究生课程。

2. 第 2 部分和第 5 部分以及其他部分的部分章节可以构成软件设计模式的课程。建议将上述 OOSE 课程作为这门课的先修课。但是可能有些国外的学生没学过 OOSE 这门课，在这种情况下，建议先使用 2～4 周的时间大概了解一下第 2 和第 3 部分介绍的方法，因为这些方法为应用模式提供了框架。

3. 第 6 和第 7 这两部分可以以很多种方式讲授。它们可以形成一门课——质量保证、测试和维护。也可以当成两门课来教，分别是软件质量保证，以及软件测试和维护。当然也可以作为三门课来教，分别是软件质量保证、软件测试和软件维护。

4. 第 13～15 章和第 19～20 章以及其他章的部分模式可以形成关于建模、设计、复杂系统的验证与确认这门课。

5. 第 1 和第 6～8 这几部分以及其他部分的部分章节可以形成软件项目管理这门课。

6. 最后，第 1 部分和第 2 部分以及第 24 章加上其他章节的部分模式和主题可以构成软件安全概论这门课程。教员可以增加素材使得课程内容更加丰富。

在网站 http://www.mhhe.com/kung 上可以找到教学辅助素材，包括 PowerPoint 教学幻灯片、随堂测试题和测试生成软件、测试问题的数据库、课程描述及要点、实验手册，以及软件工具。尤其对于没上过这门课的新教师来说，这些教学工具可以帮助减少备课的时间和工作量。

致谢

我想感谢我的众多学生，他们的提问、反馈、热情，以及将课程内容应用于实际项目的努

力都在不断激励我。学生们也阅读和使用本书以及教学素材。他们提供给我宝贵的反馈和改进的建议。有些人实地参加到了本书中某些方法的设计和实现中。有些人参加到了评估本书方法效果的实验中。很多人毕业后继续将本书方法应用于行业内，并和我分享他们的宝贵经历。我想感谢我的女儿 Jacquelyn Kung 帮我编辑草稿的部分章节。感谢 McGraw-Hill 环球出版社的工程和计算机科学部的 Raghu Srinivasan 提供给我宝贵的改进建议和对出版流程的指导。感谢评论家们提出的意见和建议。这些意见和建议对于本书的结构组织、内容显示以及很多方面都有极大的帮助。在漫长的写作过程中，我的妻子 Cindy Kung 和很多学术界及业内的同事都给予我持续的鼓励和无私的帮助，在此也对他们表示感谢。

目　录

第1部分　引言与系统工程

第 2 部分 分析和体系结构设计

第 6 部分 实现和质量保证