



编写有效用例

Writing Effective Use Cases

[美] Alistair Cockburn 著

王雷 张莉 译

Writing Effective
Use Cases



Alistair Cockburn



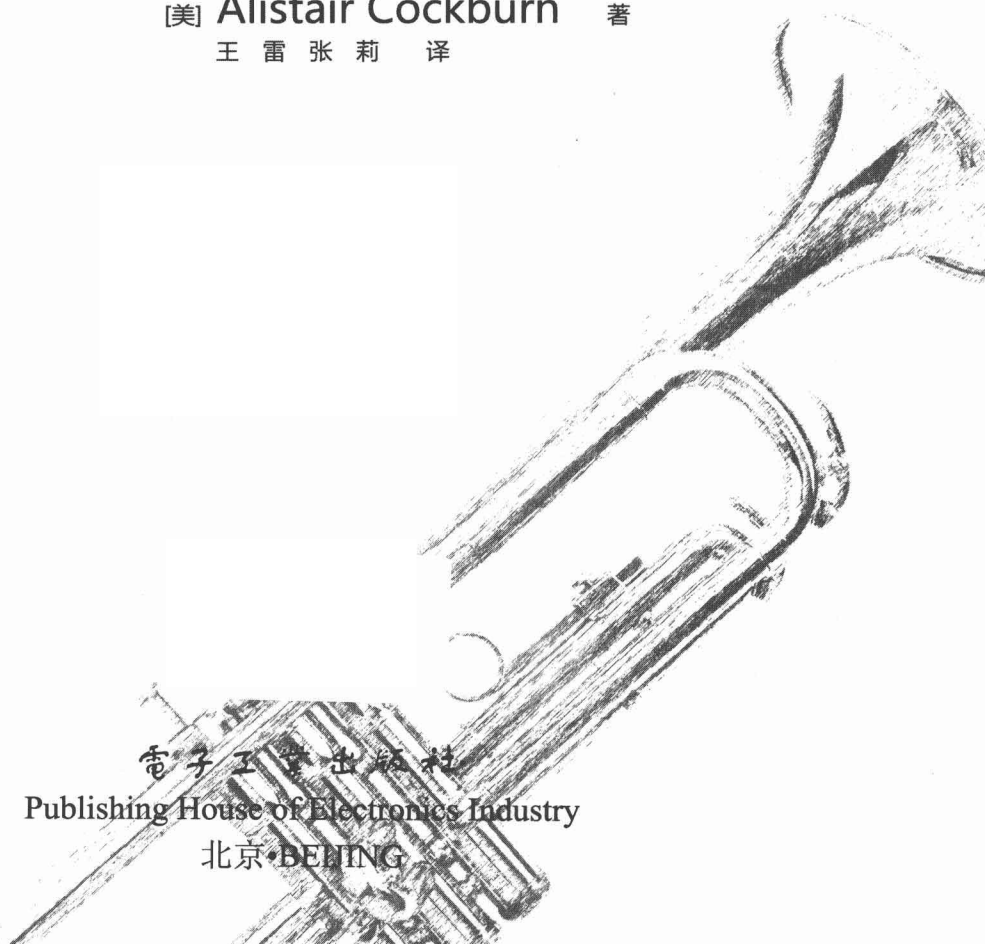
Jolt 大奖精选丛书

编写有效用例

Writing Effective Use Cases

[美] Alistair Cockburn 著
王雷 张莉 译

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING



内 容 简 介

Jolt 大奖素有“软件业之奥斯卡”的美称,本丛书精选自 Jolt 历届获奖图书,以植根于开发实践中的独到工程思想与杰出方法论为主要甄选方向。本书作者 Alistair Cockburn,凭借自己在面向对象领域的丰富经验,并参考其他专家的建议,扩展了典型的用例处理方法,为软件开发人员编写用例提供了一种“基本、具体和实用的”指南。本书完整地叙述了有关用例的初、中、高级概念,并提供了大量的、正反两方面的用例编写实例,是一本概念清晰、结构完整、内容丰富的专业图书。

本书荣获 2001 年 Jolt 世界图书大奖,适用于不同知识层次的软件工作、研究人员和用例编写人员。

Authorized translation from the English language edition, entitled WRITING EFFECTIVE USE CASES, 1E, 9780201702255 by COCKBURN, ALISTAIR, published by Pearson Education, Inc, publishing as Addison-Wesley Professional, Copyright © 2001 Addison-Wesley.

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 PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2012

本书简体中文版专有出版权由 Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2011-4222

图书在版编目(CIP)数据

编写有效用例 / (美)科伯恩(Cockburn,A.)著;王雷,张莉译. —北京:电子工业出版社,2012.5
(Jolt 大奖精选丛书)

书名原文: Writing Effective Use Cases

ISBN 978-7-121-16672-3

I. ①编… II. ①科… ②王… ③张… III. ①软件开发 IV. ①TP311.52

中国版本图书馆 CIP 数据核字(2012)第 056838 号

策划编辑:张春雨 符隆美

责任编辑:付 睿

印 刷:北京东光印刷厂

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×980 1/16 印张:22 字数:563 千字

印 次:2012 年 5 月第 1 次印刷

印 数:3500 册 定价:69.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

经久不息的回荡

今时的读书人，不复有无书可读之苦，却时有品种繁多而无从择优之感，甚而专业度颇高的技术书领域，亦日趋遭逢乱花迷眼的境地。此时，若得觅权威书评，抑或有公信力的排行榜，可按图索骥，大大增加选中好书的命中率。然而，如此良助，不可多得，纵观中外也唯见一枝独秀——素有“软件业奥斯卡”之美誉的 Jolt 奖！

震撼世界者为谁

在计算设备已经成为企业生产和日常生活之必备工具的今天，专业和大众用户对于软件的功能、性能和用户体验的要求都在不断提高。在这样的背景下，如何能够发挥出软件开发的最高效率和最大效能，已经是摆在每一个从业者面前的重大课题，而这也正是 Jolt 大奖横空出世的初衷及坚持数年的宗旨。

Jolt 大奖历时 20 余年，在图书及软件业知名度极高，广受推崇。奖如其名，为引领计算机科学与工程发展主流，Jolt 坚持将每年的奖项只颁给那些给整个 IT 业界带来震撼结果的图书、工具、产品及理念等，因一流的眼光及超高的专业度而得以闻名遐迩，声名远播。

除图书外，Jolt 针对软件产品设有诸多奖项分类，如配置管理、协作工具、数据库引擎/数据库工具、设计工具/建模、开发环境、企业工具、库/框架、移动开发工具等。但图书历来是 Jolt 大奖中最受瞩目且传播最广的一个奖项分支。Jolt 曾设有通用类图书、技术类图书等分类，每个分类又设有“卓越奖”（Jolt Award，一般为一个）和“生产力奖”（Productivity Award，一般为 2 或 3 个）。

获奖技术图书一经公布，即打上经典烙印，可谓一举“震撼全世界”（赞助商 Jolt 可乐的广告词）。

作为计算机技术图书的厚爱者，我们总在追问——是谁在震撼世界，是谁在照亮明天？Jolt 大奖恰似摆在眼前的橱窗，让我们可以近距离观看潮流在舞蹈，倾听震撼在轰鸣！

朝花夕拾为哪般

Jolt 像是一年一度的承诺，在茫茫书海中为我们淘砾出一批批经得起岁月冲刷的杰作，头顶桂冠的佳作也因而得以一批批引进中国，为国人开阔了眼界，滋补了技术养分。然而，或因技术差距造就的生不逢时、水土不服，或因翻译、制作的不如人意，抑或是疏于宣传等诸多原因，这些经典著作在国内出版后，尽管不乏如获至宝的拥趸，却仍不为诸多人所知，从而与大量本应从中获益的读者擦肩而过。既然这生生错失的遗憾本不该发生，则更不应延续。为此，我们邀国外出版同行、国内技术专家一道，踏上朝花夕拾之路，竭力为广大读者筛选出历久弥新、震撼依旧的 Jolt 图书精品。

Jolt 获奖图书皆由业界专家一致评出，并得到软件从业人员的高度认可，虽然这些书今天读来，不再能看到上世纪史诗时代那般日新月异的理论突破，以及依赖于高深繁复的科学研究所取得的系统化成果，更多是在日复一日的开发实践中总结和提炼出来的工程思想和方法论。重新选材之所以有所取舍，从 Jolt 多年来的评奖规律中可窥端倪——

一万小时真理见

凡是在工程思想领域取得革命性、颠覆性突破的图书，就被归于“震撼”获奖分类。比如，从基于过程的程序设计模型过渡到面向对象的全新模型，就是软件开发思想上的一次带来巨大震撼的革命；再比如，打破传统的瀑布模型而转向持续集成的软件交付模型，这也是一场业界的重大思想转变。像这样的重大思想突破，可以说是数年甚至数十年一遇的，而荣获 Jolt 大奖的图书中更为常见的，则是基于最佳实践的“生产效率”获奖者。获得此类殊荣的图书，都是作者们从平凡的、重复的，甚至用一般人的眼光看来不怎么起眼的日常开

发实践中，以独具的慧眼、过人的耐心和大胆的创新，闯开一条不平常道路的心血与经验总结。

这些图书所涉及的主题，都是普通的软件开发人员每天要面对的工作——代码阅读、撰写测试用例、修复软件问题……但就是这样貌似平淡无奇的工作，是否能每一天、每一个项目都做好，着实拉开了软件开发人员素质的差距，也决定了软件企业开发出来的产品和服务的质量。我们中国有一句古话，叫做熟能生巧；某位著名企业家也说过一句家喻户晓的名言：“把简单的事千百万次地做好，就是不简单的。”这些朴素而实际的真理，同样也是本套丛书最能彰显的所谓程序员精神。它建立在脚踏实地的实践基础之上，也充满了对于自由和创新的向往。

名作可堪比名曲

就不因岁月流逝而褪色来说，与这些 Jolt 名作相媲美者，只有那些百年响彻、震撼古今的经典名曲。希望本丛书带给大家的每部著作，也如百听不厌的乐曲，掩卷良久方余音绕梁，真知存心。仔细想来，软件开发与古典音乐岂非有异曲同工之妙？既是人类心智索问精确科学的探究，亦是寻觅美学享受的追求。工程是艺术的根基，而艺术是工程的极致。衷心地希望各位读者能够认真阅读本丛书的本本珍品，并切实地用于自己的日常工作中，在充分享受大师魅力的同时，为中国的软件事业谱写更多、更震撼的乐章。

电子工业出版社博文视点

二零一二年春

译者序

如何准确地描述用户需求，一直是软件开发领域研究的热点。Ivar Jacobson 首先提出了“用例”的概念，用例在描述系统行为需求、软件系统或业务流程等方面被广泛使用。但是，用例应该描述什么？描述到什么程度？这些问题，却非常难以回答。《编写有效用例》一书很好地回答了这些问题，并深受读者欢迎。非常感谢电子工业出版社张春雨和符隆美编辑选择再版本书。

本书荣获了 2001 年度《美国软件开发杂志》的“Jolt and Productivity Award”，这个奖项奖励本年度全球软件业最杰出的技术和书籍，共分 6 大类，其中有一类是图书。2001 年全球只有 4 本书获得了“Jolt and Productivity Award”，最优秀的图书荣获“Jolt Award”，其余 3 本书共享“Productivity Award”。本书获得的就是 2001 年度“Productivity Award”。

本书通过大量来自实际项目的用例，总结出了用例编写和培训中的一些指导性原则，完整地介绍了用例的基本概念。本书一个重要的特点在于，提出了**项目相关人员和利益模型**，并且提供了大量的正反两方面实例。本书共分为“引言”、“用例体部分”、“经常讨论的主题”和“对忙于编写用例人的提示”等 4 个部分。

“引言部分”介绍了与用例相关的重要概念，包括：用例的定义、用例中 3 个最基本的概念、用例的基本组成部分、不同项目组的不同用例编写风格等。

“用例体部分”，由 10 个章节组成，详细地阐述了执行者、项目相关人员和目标层次等编写用例需要掌握的重要概念，并且提供了大量用例编写的模板。具体章节为“用例是规范行为的契约”、“范围”、“项目相关人员和执行者”、“三个命名的目标层次”、“前置条件、触发事件和保证”、“场景和步骤”、“扩展”、“技术和数据的变化”、“连接用例”和“用例格式”。通过该部分的阅读，读者可以完整地了解用例各个部分的编写原则。

“经常讨论的主题”部分，由 8 个章节组成，详细说明了编写用例过程中可能反复出现的问题，例如用例什么时候才算完成、如何从一个用例扩展到多

个用例、如何编写基于数据库的小用例、如何对业务过程进行建模、哪些是可能被遗漏的需求及用例在整个过程中的作用等问题。该部分详细解答了用例编写过程中所遇到的问题，有很高的参考价值。

“对忙于编写用例的人的提示”部分，由3个章节组成，包含了针对用例关键概念的一组提示。这部分分为：“对每个用例的提示”、“对用例集的提示”和“处理用例的提示”。通过阅读这部分，读者可以回顾一下在编写用例过程中可能遇到的关键问题。

本书由浅入深，每一部分都包括概念、示例、提示和练习（以及部分答案），非常适合软件开发专业人员自学。

本书作者 Alistair Cockburn 是一位面向对象领域的国际著名专家，是软件项目开发方面的资深顾问。在他建立的网站 <http://alistair.cockburn.us/> 中，相信读者可以找到所需要的资料。

再版过程中，我们对翻译的内容进行了重新校对，限于我们的翻译水平，书中不可避免有疏漏和错误的地方，望读者指正。参加校对工作的还有聂坤明、路红、王欢、王刚和李康，在此对他们表示感谢！

王雷

2011 年 12 月于北京

前言

为了描述行为需求、软件系统或业务流程，用例正在被越来越多的人所使用。初听起来，编写用例似乎是件很容易的事——只需要写清楚如何使用系统就可以了。但是，当提笔编写时，首先碰到的问题就是：“到底应该写些什么？多写一些，还是少写一些，详细程度如何？”这些问题确实很难回答。编写用例就像写散文一样，全部困难在于既要采用一般的写作方式，又要具有完美的表达能力。很难说什么样的用例是好用例，但是我们却可以去考虑：如何去写，才能写出好用例。

本书给出了我在用例编写和培训中总结出的一些指导性原则：应该如何思考、观察需要的内容，最终写出更好的用例和用例集。

本书还给出了大量的例子，既有好的用例，也有不好的用例，以及在不同情况下的编写方式。最重要的是要记住，我们需要的是有用的用例，而不必是一个最好的用例。有时即使是很平常的用例也很有用，甚至远胜于编写许多需求文档。所以无须太紧张，只要写出的东西可读，就说明你已经做出了贡献。

本书的读者

本书主要是针对工业界专业人员自学而编写的，因此在组织方式上也有所侧重，就像一本自学指南。本书由浅入深，每一部分都包括概念、示例、提示和练习（部分答案）。

编写用例的培训教师应给出适当的解释和示例，课程设计人员可以围绕本书设计课程资料，在必要时作为指定读物发布（由于我对许多练习都给出了答案，因此他们必须编写属于自己的考试材料）。

本书的组织

本书首先概要地介绍了用例的概念，之后对用例体部分进行了详细阐述，

然后是常见的问题，对忙于编写用例的人的提示，以及最后的注意事项。

“引言”部分对重要的概念进行了介绍，依次讨论了以下问题：“用例是什么样的？”、“何时编写用例？”和“什么样的用例是合法的？”。所有问题的答案都是根据具体情况（如时间、地点、人员及编写的理由等）来确定的。关于这几个问题的讨论虽从本部分开始，但实际上贯穿本书。

“第一部分，用例体部分”，由多个章节组成，分别阐述了需要掌握的重要概念，以及应编写的模板。具体章节为“用例是规范行为的契约”、“范围”、“项目相关人员和执行者”、“三个命名的目标层次”、“前置条件、触发事件和保证”、“场景和步骤”、“扩展”、“技术和数据的变化”、“连接用例”和“用例格式”。

“第二部分，经常讨论的主题”，列举了反复出现的特殊问题：“什么时候才算完成”、“扩展到多个用例”、“CRUD 和参数化用例”、“业务过程建模”、“遗漏的需求”、“用例在整个过程中的作用”、“用例概述和极端编程”和“错误改正”。

“第三部分，对忙于编写用例的人的提示”，包含了一组提示，针对那些已经阅读完本书，或者已经了解这些资料，想要回顾一下关键概念的人们。这部分结构如下：“对每个用例的提示”、“对用例集的提示”和“处理用例的提示”。

本书有 4 个附录：附录 A 讨论了“UML 中的用例”，附录 B 包括“部分习题的答案”。最后是附录 C（术语表）和附录 D（参考文献），即撰写本书时使用的资料列表。

中文版书中订口处的“”表示英文版原书页码，便于读者与原书对照阅读，本书的索引所列页码为原英文版页码。

本书概念的来源

在 20 世纪 60 年代后期，Ivar Jacobson 在爱立信公司电话系统工作时提出了后来成为众所周知的“用例”。在 20 世纪 80 年代后期，他将用例引入了面向对象编程领域，在这里人们认识到用例可以填补需求分析过程中一个明显的空白。我在 20 世纪 90 年代初参加了 Jacobson 的课程，他和他的小组没用我所

使用的目标（goal）和目标失败（goal failure）这两个词，但最后我终于知道他们曾经使用过类似的概念。经过几次比较，他和我都发现我们两个模型之间没有显著的区别。我慢慢地根据最近的想法扩展了他的模型。

在 1994 年我为 IBM 顾问组编写用例指南时，建立了执行者（actor）和目标（goal）概念模型。从而把用例中一些难以理解的事情解释清楚了，并且为如何构造和编写用例提供了指导。从 1995 年开始，执行者和目标模型就在 <http://members.aol.com/acockburn> 及后来的 www.usecases.org 中正式使用，最后出现在 1997 年 *Journal of Object Oriented Programming* 上我写的一篇文章中，题为“构造带目标用例”。

从 1994 到 1999 年，尽管在理论上还有几个松散的分支，但用例模型的概念基本保持稳定。在教授和指导过程中，我终于明白了人们为什么在简单的概念上花费了大量时间（我竟然从来没想到，我第一次尝试时也犯过很多同样的错误！）。这些想法，加上“执行者和目标”模型中一些不合理之处，产生了本书及项目相关人员（stakeholder）和利益（interest）模型，该模型是本书提出的一个新概念。

统一建模语言（UML）对这些概念没什么影响，反之亦然。Jacobson 以前的一位同事 Gunnar Overgaard 编写了大量 UML 用例的资料，并且保持了 Jacobson 的风格。然而 UML 标准开发组受画图工具的影响很大，以至于用例的文本特征在标准中消失了。Gunnar Overgaard 和 Ivar Jacobson 讨论了我提出的概念，并向我保证这些关于用例的大部分概念都适合放在 UML 的一个椭圆图中，因此既不会影响 UML，也不会被 UML 标准所提出的概念影响。这意味着你可以使用本书的概念，并与 UML 1.3 用例标准兼容。另一方面，如果你只读了 UML 标准，由于它根本没有讨论用例的内容及如何去编写，那么你也知道用例到底是什么、如何使用，并且还可能产生一个危险的想法，即用例由图形而不是文本构成的。本书的目的是告诉你如何编写有效的用例，而标准很少谈及这些，因此我把对 UML 的评述单独放在附录 A 中。

本书所用的实例

本书编写的实例尽可能取自实际项目，并且有些实例看起来可能不太完

美。我所要说明的是，这些实例对项目组的需求来说已经足够了，并且用例编写过程中的那些不完美之处是在误差和经济允许的范围內。

Addison-Wesley 的编辑们说服我把这些实例从原有形式中整理出来，以强调它们正确的形式，而不是它们实际的和可用的形式。希望你通过阅读这些实例发现一些有用之处，并了解项目中的实际编写过程。你可以应用我在这些实例中采用的一些原则，并找到改进它们的方法。这种事情经常发生，因为改进一个人编写的实例是一件永无止境的事情，所以我接受这个挑战和任何批评意见。

《软件专业人员集锦》中的用例

《软件专业人员集锦》(*The Crystal Collection for Software Professionals*)只是图书选集之一，强调了一直被忽略的、发挥人能动性的软件开发技术。其中一些书讨论了某种技术，一些书讨论了项目中的角色，一些书讨论了开发组协作的观点。

“集锦”遵循下面两个原则。

- 软件开发是创造和交流相互作用的游戏。当我们提高开发人员的个人技术和开发组协作效率时，它得到了改进。
- 不同的项目有不同的需求。系统有不同的特征，由大小不同的开发组研制，并且开发组成员有着不同的价值观，优先考虑的事情也不同。因此不可能定义一个最好的软件开发方法。

“集锦”中的基础读物——*Software Development as a Cooperative Game*，详细描述了这样一些观点：软件开发是一种合作游戏，方法学是一种合作文化，以及方法学系列化。该书分别对方法学的不同侧面、技术与活动、工作产品和标准进行了讨论。用例所需要讨论的精髓出现在本书的 1.2 节“你的用例不能作为我的用例”。

本书是一本技术指南，描述了用例编写过程。尽管你可以在几乎所有项目中使用这些技术，但是必须根据每个项目的实际需要选择模板和编写标准。

致谢

在此我要感谢很多人。感谢那些阅读了本书草稿，并要求对那些可能引起他们的客户、同事及学生混淆的主题予以澄清的人们。特别感谢 Russell Walters 的鼓励和具体的反馈意见，他是一个经验丰富、对开发组的方向和实际需求有敏锐目光的人。感谢 FirePond & Fireman 基金保险公司提供了生动的用例实例。Pete McBreen 第一个尝试了项目相关人员和利益模型，并加入他的合理想法、实践结果和改进建议。感谢硅谷的 Patterns Group 仔细阅读了早期的几版草稿，以及他们对不同文稿和概念所做的富有教益的评述。感谢 Fort Union Bean & Brew 的 Mike Jones 为子系统用例想出了螺钉图标。

应该特别提一下 Susan Lilly，她严格地阅读本书，并改正能想到的任何疏忽：顺序、内容、格式，甚至用例实例。她所做的大量工作都体现在得到大为改进的最终版本上。

还有其他一些评阅者提出了详细的意见并给予了鼓励，他们是 Paul Ramney, Andy Pols, Martin Fowler, Karl Waclawek, Alan Williams, Brian Henderson-Sellers, Larry Constantine, Russell Gold. Addison-Wesley 出版公司的编辑们很好地完成了他们的工作，清除了我不雅观的句型及经常出现的拼写错误。

感谢听我讲课的人们，他们帮助我找出了书中的错误概念。

再次感谢我的家人, Deanna, Cameron, Sean 和 Kieran; 还要感谢 Fort Union Beans & Brew 的人员提供了许多咖啡和营造了一种欢乐的气氛。

在我维护的网站 members.aol.com/acockburn 和 www.usecases.org 上可以找到更多关于用例的材料。为了避免我们将来见面时的尴尬，我名字的发音是 Co-burn, o 是长元音。

提示

编写有可读性的用例。

不正式但具有可读性的用例还是有用的，而没有可读性的用例则一点儿用都没有。

从低精度到高精度，按广度优先开展工作：

精度级别 1：主执行者的名字和目标

精度级别 2：用例简介或主成功场景

精度级别 3：扩展条件

精度级别 4：扩展的处理步骤

对于每一步：

显示一个成功的目标

捕获执行者的意图，而不是用户接口的细节

有一个传递信息、检测条件或者改变状态的执行者

在步骤之间编写注释，指明步骤的顺序（如果需要）

通过问“为什么”来找到更高层次的目标

对于数据描述（只有精度级别 1 放在用例文本中）：

精度级别 1：数据别名

精度级别 2：与别名相关联的数据域

精度级别 3：域的类型、长度和确认条件

图标

设计范围		目标级别	
	组织（黑盒）		很高的概要
	组织（白盒）		概要
	系统（黑盒）		用户目标
	系统（白盒）		子功能
	组件		太低

对于目标级别，也可以采用用例名字加下列字符后缀的方式来表示。

在概要用例名字后加上“+”。

在用户目标用例名字后加上“!”或什么也不加。

在子功能用例名字后加上“-”。

编写过程

1. 定义系统范围和边界。

追踪带有输入/输出列表的初始语境图的变化。

2. 集中讨论，并列出主执行者。

在系统整个生命周期内，找出每个人或非人的主执行者。

3. 集中讨论，并尽可能列出系统所有的用户目标。

产生初始的执行者-目标列表。

4. 捕获最外部的概要用例，看看谁最值得关注。

对每个主执行者，检查最外部的概要用例。

5. 再次考虑和修改概要用例。增加、减少或者合并用户目标。

对基于时间的触发条件及系统边界上的其他事件进行双重检查。

6. 选择一个用例进行扩展。

考虑编写一个叙述文档来了解已有的材料。

7. 捕获项目相关人员和利益、前提条件和保证。

系统确保前提条件和利益得到满足。

8. 编写主成功场景（Main Success Scenario, MSS）

用 3~9 个步骤满足所有利益和保证。

9. 集中讨论，并尽可能列出所有扩展条件。

包括系统所能检测到，并且必须处理的所有条件。

10. 编写扩展处理的步骤。

每个处理结束于：返回主成功场景，一个单独的成功结束条件，或者失败。

11. 抽取复杂的处理流程作为子用例，合并琐碎的子用例。

抽取子用例是容易的，但这会增加项目的开销。

12. 重新调整用例集：进行必要的增加、减少与合并。

检查可读性、完备性及是否满足项目相关人员的利益。

目录

第1章 引言	1
1.1 用例是什么（梗概）	1
用例1 通过网络购买股票	3
用例2 汽车交通事故索赔	5
用例3 对运到的包装箱进行登记	6
1.2 你的用例不能作为我的用例	7
用例4 买东西（非正式版本）	10
用例5 买东西（完整正式版本）	10
◆ Steve Adolph：在新领域中“发掘”需求	14
1.3 需求和用例	15
图 1-1 “轮轴和轮辐”需求模型	17
用例作为项目连接结构	18
1.4 用例的增值点	18
1.5 合理安排你的精力	19
1.6 先用使用叙述做热身	21
1.7 练习	22

第1部分 用例体部分

第2章 用例是规范行为的契约	27
2.1 具有目标的执行者之间的交互	27
执行者具有目标	27
图 2-1 一个具有目标的执行者请求另一个执行者履行职责	28
目标可能失败	29
交互是复杂的	30
用例聚集场景	33