

经 典 原 版 书 库

编写有效用例

(英文版)

Writing Effective Use Cases



The Agile Software Development Series

Cockburn • Highsmith
Series Editors



Alistair Cockburn

(美) Alistair Cockburn 著



机械工业出版社
China Machine Press



Pearson Education
培生教育出版集团

经 典 原 版 书 库

编写有效用例

(英文版)

Writing Effective Use Cases

(美) Alistair Cockburn 著



机械工业出版社
China Machine Press



Pearson Education
培生教育出版集团

English reprint edition copyright © 2002 by PEARSON
EDUCATION NORTH ASIA LTD and China Machine Press.

Original English language title: Writing Effective Use Cases 1st
Edition by Alistair Cockburn, Copyright © 2001.

All Rights Reserved.

Published by arrangement with the original publisher, Pearson
Education, Inc., publishing as Addison Wesley Longman, Inc.

This edition is authorized for sale only in People's Republic of China
(excluding the Special Administrative Region of Hong Kong and Macau).

本书英文影印版由Pearson Education North Asia Ltd. 授权机械工
业出版社在中国大陆境内独家出版发行, 未经出版者许可, 不得以任
何方式抄袭、复制或节录本书中的任何部分。

本书封面贴有Pearson Education培生教育出版集团激光防伪标
签, 无标签者不得销售。

版权所有, 侵权必究。

本书版权登记号: 图字: 01-2002-1839

图书在版编目 (CIP) 数据

编写有效用例/ (美) 科伯恩 (Cockburn, A.) 著. -北京: 机械工
业出版社, 2002.7

(经典原版书库)

书名原文: Writing Effective Use Cases

ISBN 7-111-10537-0

I. 编… II. 科… III. 软件工程 - 英文 IV. TP311.5

中国版本图书馆CIP数据核字 (2002) 第046250号

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: 华 章

北京昌平奔腾印刷厂印刷 · 新华书店北京发行所发行

2002年7月第1版第1次印刷

850mm × 1168mm 1/32 · 9.5 印张

印数: 0 001-3 000 册

定价: 25.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

Reminders

Write something readable.

Casual, readable use cases are still useful, whereas unreadable use cases won't get read.

Work breadth-first, from lower precision to higher precision.

Precision Level 1: Primary actor's name and goal

Precision Level 2: The use case brief, or the main success scenario

Precision Level 3: The extension conditions

Precision Level 4: The extension handling steps

For each step:

Show a goal succeeding.

Capture the actor's intention, not the user interface details.

Have an actor pass information, validate a condition, or update state.

Write between-step commentary to indicate step sequencing (or lack of).

Ask "why" to find a next-higher level goal.











For data descriptions (only put Precision Level 1 into the use case text):

Precision Level 1: Data nickname

Precision Level 2: Data fields associated with the nickname

Precision Level 3: Field types, lengths, and validations

Icons

Design Scope		Goal Level	
 Organization (black-box)		 Very high summary	
 Organization (white-box)		 Summary	
 System (black box)		 User-goal	
 System (white box)		 Subfunction	
 Component		 Too low	

For Goal Level, alternatively, append one of these characters to the use case name:

Append "+" to summary use case names.

Append "!" or nothing to user-goal use case names.

Append "-" to subfunction use case names.

The Writing Process

1. Name the system scope and boundaries.
Track changes to this initial context diagram with the in/out list.
2. Brainstorm and list the primary actors.
Find every human and non-human primary actor, over the life of the system.
3. Brainstorm and exhaustively list user goals for the system.
The initial Actor-Goal List is now available.
4. Capture the outermost summary use cases to see who really cares.
Check for an outermost use case for each primary actor.
5. Reconsider and revise the summary use cases. Add, subtract, or merge goals.
Double-check for time-based triggers and other events at the system boundary.
6. Select one use case to expand.
Consider writing a narrative to learn the material.
7. Capture stakeholders and interests, preconditions and guarantees.
The system will ensure the preconditions and guarantee the interests.
8. Write the main success scenario (MSS).
Use 3 to 9 steps to meet all interests and guarantees.
9. Brainstorm and exhaustively list the extension conditions.
Include all that the system can detect and must handle.
10. Write the extension-handling steps.
Each will end back in the MSS, at a separate success exit, or in failure.
11. Extract complex flows to sub use cases; merge trivial sub use cases.
Extracting a sub use case is easy, but it adds cost to the project.
12. Readjust the set: add, subtract, merge, as needed.
Check for readability, completeness, and meeting stakeholders' interests.

出版者的话

文艺复兴以降，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中取得了垄断性的优势；也正是这样的传统，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，不仅擘划了研究的范畴，还揭橥了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray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、研究及收藏。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襄助，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，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“经典原版书库”是响应教育部提出的使用原版国外教材的号召，为国内高校的计算机教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这30多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。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，我们的联系方法如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995265

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范 明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅 宏	程 旭	程时端	谢希仁
裘宗燕	戴 葵			

Preface

More and more people are writing use cases, for behavioral requirements, for software systems or to describe business processes. It all seems easy enough—just write about using the system. But, faced with writing, one suddenly confronts the question, “Exactly what am I supposed to write—how much, how little, what details?” That turns out to be a difficult question to answer. The problem is that writing use cases is fundamentally an exercise in writing prose essays, with all the difficulties in articulating *good* that comes with prose writing in general. It is hard enough to say what a good use case looks like, but we really want to know something harder: how to write them so they will come out being good.

These pages contain the guidelines I use in my use case writing and in coaching: how a person might think, what he or she might observe, to end up with a better use case and use case set.

I include examples of good and bad use cases, plausible ways of writing differently, and, best of all, the good news that a use case need not be the *best* to be *useful*. Even mediocre use cases are useful, more so than are many of the competing requirements files being written. So relax, write something readable, and you will have done your organization a service.

Audience

This book is predominantly aimed at industry professionals who read and study alone. and is therefore organized as a self-study guide. It contains introductory through advanced material: concepts, examples, reminders, and exercises (some with answers, some without).

Writing coaches should find suitable explanations and samples to show their teams. Course designers should be able to build course material around the book, issuing

reading assignments as needed. (However, as I include answers to many exercises, they will have to construct their own exam material. :-)

Organization

The book is organized as a general introduction to use cases followed by a close description of the use case body parts, frequently asked questions, reminders for the busy, and end notes.

The Introduction contains an initial presentation of key notions, to get the discussion rolling: “What does a use case look like?” “When do I write one?” and “What variations are legal?” The brief answer is that they look different depending on when, where, with whom, and why you are writing them. That discussion begins in this early chapter, and continues throughout the book

Part 1, *The Use Case Body Parts*, contains chapters for each of the major concepts that need to be mastered, and parts of the template that should be written. These include “The Use Case as a Contract for Behavior,” “Scope,” “Stakeholders and Actors,” “Three Named Goal Levels,” “Preconditions, Triggers, and Guarantees,” “Scenarios and Steps,” “Extensions,” “Technology and Data Variations,” “Linking Use Cases,” and “Use Case Formats.”

Part 2, *Frequently Discussed Topics*, addresses particular topics that come up repeatedly: “When Are We Done?,” “Scaling Up to Many Use Cases,” “CRUD and Parameterized Use Cases,” “Business Process Modeling,” “The Missing Requirements,” “Use Cases in the Overall Process,” “Use Case Briefs and eXtreme Programming,” and “Mistakes Fixed.”

Part 3, *Reminders for the Busy*, contains a set of reminders for those who have finished reading the book, or already know this material and want to refer back to key ideas. The chapters are organized as “Reminders for Each Use Case,” “Reminders for the Use Case Set,” and “Reminders for Working on the Use Cases.”

There are four appendices: Appendix A discusses “Use Cases in UML” and Appendix B contains “Answers to (Some) Exercises.” The book concludes with Appendix C. Glossary; and a list of materials used while writing, Appendix D. Readings.

Heritage of the Ideas

In the late 1960s, Ivar Jacobson invented what later became known as use cases while working on telephony systems at Ericsson. In the late 1980s, he introduced them to the object-oriented programming community, where they were recognized as filling a significant gap in the requirements process. I took Jacobson’s course in the early 1990s. While neither he nor his team used my phrases *goal* and *goal failure*, it eventually became clear to me that they had been using these notions. In several comparisons, he

and I have found no significant contradictions between his and my models. I have slowly extended his model to accommodate recent insights.

I constructed the Actors and Goals conceptual model in 1994 while writing use case guides for the IBM Consulting Group. It explained away much of the mystery of use cases and provided guidance as to how to structure and write them. The Actors and Goals model has circulated informally since 1995 at <http://members.aol.com/acockburn> and later at www.usecases.org, and finally appeared in the *Journal of Object-Oriented Programming* in 1997, in an article I authored entitled “Structuring Use Cases with Goals.”

From 1994 to 1999, the ideas stayed stable, even though there were a few loose ends in the theory. Finally, while teaching and coaching, I saw why people were having such a hard time with such a simple idea (never mind that I made many of the same mistakes in my first tries!). These insights, plus a few objections to the Actors and Goals model, led to the explanations in this book and to the Stakeholders and Interests model, which is a new idea presented here.

The Unified Modeling Language (UML) has had little impact on these ideas—and vice versa. Gunnar Overgaard, a former colleague of Jacobson’s, wrote most of the UML use case material and kept Jacobson’s heritage. However, the UML standards group has a strong drawing-tools influence, with the effect that the textual nature of use cases has been lost in the standard. Gunnar Overgaard and Ivar Jacobson discussed my ideas and assured me that most of what I have to say about a use case fits *within one of the UML ellipses*, and hence neither affects nor is affected by what the UML standard has to say. That means that you can use the ideas in this book quite compatibly with the UML 1.3 use case standard. On the other hand, if you only read the UML standard, which does not discuss the content or writing of a use case, you will not understand what a use case is or how to use it, and you will be led in the dangerous direction of thinking that use cases are a graphical, as opposed to a textual, construction. Since the goal of this book is to show you how to write effective use cases and the standard has little to say in that regard, I have isolated my remarks about UML to Appendix A.

Samples Used

The writing samples in this book were taken from live projects as much as possible, and they may seem slightly imperfect in some instances. I intend to show that they were sufficient to the needs of the project teams that wrote them, and that those imperfections are within the variations and economics permissible in use case writing.

The Addison-Wesley editing crew convinced me to tidy them up more than I originally intended, to emphasize correct appearance over the actual and adequate appearance. I hope you will find it useful to see these examples and recognize the

writing that happens on projects. You may apply some of my rules to these samples and find ways to improve them. That sort of thing happens all the time. Since improving one's writing is a never-ending task, I accept the challenge and any criticism.

Use Cases in The Crystal Collection

This is just one in a collection of books, *The Crystal Collection for Software Professionals*, that highlights lightweight, human-powered software development techniques. Some books discuss a single technique, some discuss a single role on a project, and some discuss team collaboration issues.

Crystal works from two basic principles:

- ◆ Software development is a cooperative game of invention and communication. It improves as we develop people's personal skills and increase the team's collaboration effectiveness.
- ◆ Different projects have different needs. Systems have different characteristics and are built by teams of differing sizes, with members having differing values and priorities. It is impossible to name one, best way of producing software.

The foundation book for the Crystal Collection, *Software Development as a Cooperative Game*, elaborates the ideas of software development as a cooperative game, of methodology as a coordination of culture, and of methodology families. That book separates the different aspects of methodologies, techniques and activities, work products and standards. The essence of the discussion, as needed for use cases, appears in this book in Section 1.2, *Your Use Case Is Not My Use Case* on page 7.

Writing Effective Use Cases is a technique guide, describing the nuts-and-bolts of use case writing. Although you can use the techniques on almost any project, the templates and writing standards must be selected according to each project's needs.

Acknowledgments

Thanks to lots of people. Thanks to the people who reviewed this book in draft form and asked for clarification on topics that were causing their clients, colleagues, and students confusion. Special thanks to Russell Walters, a practiced person with a sharp eye for the direct and practical needs of the team, for his encouragement and very specific feedback. Thanks to FirePond and Fireman's Fund Insurance Company for the live use case samples. Pete McBreen, the first to try out the Stakeholders and Interests model, added his usual common sense, practiced eye, and suggestions for improvement. Thanks to the Silicon Valley Patterns Group for their careful reading of early drafts and their educated commentary on various papers and ideas. Mike Jones at the Fort Union Beans & Brew thought up the bolt icon for subsystem use cases.

Susan Lilly deserves special mention for the exact reading she did, correcting everything imaginable: sequencing, content, formatting, and even use case samples. The huge amount of work she contributed is reflected in the much improved final copy.

Other reviewers who contributed detailed comments and encouragement include Paul Ramney, Andy Pols, Martin Fowler, Karl Wacławek, Alan Williams, Brian Henderson-Sellers, Larry Constantine, and Russell Gold. The editors at Addison-Wesley did a good job of cleaning up my usual ungainly sentences and frequent typos.

Thanks to the people in my classes for helping me debug the ideas in the book.

Thanks again to my family, Deanna, Cameron, Sean, and Kieran, and to the people at the Fort Union Beans & Brew who once again provided lots of caffeine and a convivial atmosphere.













More on use cases is at the web sites I maintain: *members.aol.com/acockburn* and *www.usecases.org*. Just to save us some future embarrassment, my name is pronounced Cö-burn, with a long o.














Contents

Preface *xix*
Acknowledgments *xxiii*

Chapter 1 Introduction **1**

1.1 What is a Use Case (More or Less)? 1
 Use Case 1 ■ *Buy Stocks over the Web* ✂ 4
 Use Case 2 ▲ *Get Paid for Car Accident* ρ 5
 Use Case 3 ■ *Register Arrival of a Box* ✂ 6
1.2 Your Use Case Is Not My Use Case 7
 Use Case 4 ☞ *Buy Something (Casual Version)* ρ 9
 Use Case 5 ☞ *Buy Something (Fully Dressed Version)* ρ 9
 ◆ Steve Adolph: "Discovering" Requirements in New Territory 12
1.3 Requirements and Use Cases 13
 Use Cases as Project-Linking Structure 14
 Figure 1.1 The "Hub-and-Spoke" model of requirements 15
1.4 When Use Cases Add Value 15
1.5 Manage Your Energy 16
1.6 Warm Up with a Usage Narrative 17
 ◆ Usage Narrative: Getting "Fast Cash" 18
1.7 Exercises 19

Part 1 The Use Case Body Parts	21
Chapter 2 The Use Case as a Contract for Behavior	23
2.1 Interactions between Actors with Goals	23
Actors Have Goals	23
Figure 2.1 An actor with a goal calls on the responsibilities of another	24
Goals Can Fail	25
Interactions Are Compound	25
A Use Case Collects Scenarios	27
Figure 2.2 Striped trousers: Scenarios succeed or fail	28
Figure 2.3 The striped trousers showing subgoals	29
2.2 Contract between Stakeholders with Interests	29
Figure 2.4 The SuD serves the primary actor, protecting offstage stakeholders	30
2.3 The Graphical Model	31
Figure 2.5 Actors and stakeholders	32
Figure 2.6 Behavior	32
Figure 2.7 Use Case as responsibility invocation	33
Figure 2.8 Interactions as composite	33
Chapter 3 Scope	35
Table 3.1 A Sample In/Out List	36
3.1 Functional Scope	36
The Actor-Goal List	36
Table 3.2 A Sample Actor-Goal List	37
The Use Case Briefs	37
Table 3.3 Sample Use Case Briefs	38
3.2 Design Scope	38
Figure 3.1 Design scope can be any size	40
Using Graphical Icons to Highlight the Design Scope	40
Design Scope Examples	41
Enterprise-to-System Examples	41
Use Case 6  Add New Service (Enterprise) 	42
Use Case 7  Add New Service (Acura) 	42
Many Computers to One Application	43
Use Case 8  Enter and Update Requests (Joint System) 	43
Use Case 9  Add New Service (into Acura) 	44
Use Case 10  Note New Service Request (in BSSO) 	44
Use Case 11  Update Service Request (in BSSO) 	44

Use Case 12	■ <i>Note Updated Request (in Acura)</i> 	44
Figure 3.2	Use case diagrams for Acura-BSSO..	45
Figure 3.3	A combined use case diagram for Acura-BSSO..	45
Nuts and Bolts Use Cases.		46
Use Case 13	 <i>Serialize Access to a Resource</i> 	46
Use Case 14	 <i>Apply a Lock Conversion Policy</i> 	47
Use Case 15	 <i>Apply an Access Compatibility Policy</i> 	48
Use Case 16	 <i>Apply an Access Selection Policy</i> 	48
Use Case 17	 <i>Make Service Client Wait for Resource Access</i> 	49
3.3	The Outermost Use Cases.	49
3.4	Using the Scope-Defining Work Products.	51
3.5	Exercises.	51
Chapter 4 Stakeholders and Actors		53
4.1	Stakeholders.	53
4.2	The Primary Actor.	54
	Why Primary Actors Are Unimportant (and Important).	55
	Actors versus Roles.	57
	Characterizing the Primary Actors.	58
	Table 4.1 A Sample Actor Profile Table.	56
4.3	Supporting Actors.	59
4.4	The System Under Discussion.	59
4.5	Internal Actors and White-Box Use Cases.	59
4.6	Exercises.	60
Chapter 5 Three Named Goal Levels		61
	Figure 5.1 Use case levels.	62
5.1	User Goals (Blue, Sea-Level).	62
	Two Levels of Blue.	63
5.2	Summary Level (White, Cloud/ Kite).	64
	Use Case 18  <i>Operate an Insurance Policy</i> + 	65
	The Outermost Use Cases Revisited.	65
5.3	Subfunctions (Indigo/Black, Underwater/Clam).	66
	Summarizing Goal Levels.	66
5.4	Using Graphical Icons to Highlight Goal Levels.	67
5.5	Finding the Right Goal Level.	68
	Finding the User's Goal.	68

Raising and Lowering Goal Levels 69

Figure 5.2 Ask “why” to shift levels 69

5.6 A Longer Writing Sample: “Handle a Claim” at Several Levels 70

 Use Case 19 *Handle a Claim (Business)* 71

 Use Case 20 *Evaluate Work Comp Claim* 72

 Use Case 21 *Handle a Claim (Systems)* + 73

 Use Case 22 *Register a Loss* 75

 Use Case 23 *Find a Whatever (Problem Statement)* 79

5.7 Exercises 79

Chapter 6 *Preconditions, Triggers, and Guarantees* **81**

6.1 Preconditions 81

6.2 Minimal Guarantees 83

6.3 Success Guarantee 84

6.4 Triggers 84

6.5 Exercises 85

Chapter 7 *Scenarios and Steps* **87**

7.1 The Main Success Scenario 87

 The Common Surrounding Structure 87

 The Scenario Body 89

7.2 Action Steps 90

 Guidelines 90

 Guideline 1: Use Simple Grammar 90

 Guideline 2: Show Clearly “Who Has the Ball” 90

 Guideline 3: Write from a Bird’s Eye View 91

 Guideline 4: Show the Process Moving Forward 91

 Guideline 5: Show the Actor’s Intent, Not the Movements 92

 Guideline 6: Include a “Reasonable” Set of Actions 93

 Figure 7.1 A transaction has four parts 93

 Guideline 7: “Validate,” Don’t “Check Whether” 95

 Guideline 8: Optionally Mention the Timing 95

 Guideline 9: Idiom: “User Has System A Kick System B” 96

 Guideline 10: Idiom: “Do Steps x–y until Condition” 96

 To Number or Not to Number 97

7.3 Exercises 98