



广东外语外贸大学国际服务外包人才培养系列教材

*An Intermediate English Reading and Writing Course
On Software Development For IT Professionals*

软件开发中级英语阅读与写作教程

许罗迈 主编



暨南大学出版社
JINAN UNIVERSITY PRESS



广东外语外贸大学国际服务外包人才培养系列教材

*An Intermediate English Reading and Writing Course
On Software Development For IT Professionals*

软件开发中级英语阅读与写作教程

许罗迈 主编



暨南大学出版社
JINAN UNIVERSITY PRESS

中国·广州

图书在版编目 (CIP) 数据

软件开发中级英语阅读与写作教程/许罗迈主编. —广州: 暨南大学出版社, 2014. 1

(广东外语外贸大学国际服务外包人才培养系列教材)

ISBN 978 - 7 - 5668 - 0778 - 6

I. ①软… II. ①许… III. ①软件开发—英语—阅读—教学—高等学校—教材 ②软件开发—英语—写作—高等学校—教材 IV. ①H31

中国版本图书馆 CIP 数据核字(2013)第 228399 号

出版发行: 暨南大学出版社

地 址: 中国广州暨南大学

电 话: 总编室 (8620) 85221601

营销部 (8620) 85225284 85228291 85228292 (邮购)

传 真: (8620) 85221583 (办公室) 85223774 (营销部)

邮 编: 510630

网 址: <http://www.jnupress.com> <http://press.jnu.edu.cn>

排 版: 广州市天河星辰文化发展部照排中心

印 刷: 湛江日报社印刷厂

开 本: 787mm × 1092mm 1/16

印 张: 8.75

字 数: 154 千

版 次: 2014 年 1 月第 1 版

印 次: 2014 年 1 月第 1 次

印 数: 1—2000 册

定 价: 20.00 元

(暨大版图书如有印装质量问题, 请与出版社总编室联系调换)

“广东外语外贸大学国际服务外包人才 培训系列教材”编委会

主任：隋广军

副主任：顾也力 郑建荣

委员：（按姓氏笔画排序）

毕惠阳 李铁立 林吉双 姜灵敏 黄立军

黄永智 蒋吉频 曾 增 熊海涛 魏 青

总 序

自 21 世纪以来，我国承接美欧日等国家和地区的国际服务外包呈加速发展之势。2012 年，我国承接国际服务外包执行金额为 336.4 亿美元，现已成为全球第二大服务外包接包国。伴随着服务外包产业的迅速发展，我国能熟练从事国际服务外包业务中高端人才的短缺问题日益突显出来。因此，尽快培养国际服务外包产业所需的中高端人才，已成为促进我国服务外包产业持续、快速和健康发展的当务之急。

广东外语外贸大学国际服务外包研究院和国际服务外包人才培养基地是全国普通高等院校中最早成立的有关国际服务外包研究和人才培养的专门机构。2009 年 10 月以来，国际服务外包研究院承接国际服务外包的理论研究和政府咨询等课题 40 余项，发表论文 200 余篇。目前，广东外语外贸大学国际服务外包研究院已成为华南地区国际服务外包理论研究中心、政府决策咨询智库。四年来，广东外语外贸大学国际服务外包人才培养基地共培训软件架构师、软件测试工程师和网络工程师等 IT 类高校“双师型”教师 150 余人；培养和培训 ITO、BPO、KPO 等适用型大学毕业生 2 000 余人；为 IBM、西艾、从兴等服务外包企业定制培训服务外包商务英语和相关业务流程专业人才 500 余人；培训服务外包企业和政府中高层管理人员近 7 000 人。经过几年来对服务外包人才培养模式与实践的有益探索，广东外语外贸大学国际服务外包人才培养基地已成为广东省服务外包“双师型”教师资源库、大学毕业生适用型人才交付中心、企业和政府管理人员短期进修中心。

广东外语外贸大学作为广东省国际服务外包高端人才培养基地，为更好地发挥其在国际化人才培养上的优势，进一步提高国际服务外包人才培养的质量，特组织专家学者编写了本套教材。本套教材包括《服务外包企业战略管理》、《服务外包项目管理》、《服务外包客户关系管理》、《商务沟通英语》、《商务会谈技巧英语》、《商务谈判日语》、《商务交际日语》、《软件开发中级英语阅读与写作教程》和《软件测试中级英语阅读与写作教程》，共 9 本。

培训服务外包产业所需的中高端人才是一项系统工程，其中，编写出能

够既反映服务外包发展理论，又符合服务外包发展实践的教材就尤其重要。我们希望本套教材的出版能够为服务外包人才的培养尽一份力量；同时，我们也真诚地欢迎各位读者对本套教材的不足之处提出修改的意见和建议，以期进一步提高我们教材编写的质量。

广东外语外贸大学国际服务外包人才培养系列教材编委会

2013年5月

前 言

对于中国的软件企业来说，提高竞争力、积极参与国际软件外包服务市场的竞争是一条合适的道路，因此企业员工必须具备一定的英文写作能力。

本书作者通过对广东若干具有代表性的国际软件外包服务公司做调研，发现这些企业的软件工程师都只接受过大学（公共）英语教育，英语水平有限，但是他们却要在编写程序的同时承担英文技术文件的撰写工作。调研也发现，他们面临的最大困难是写出符合英语表达习惯的规范文章。根据目前国内软件工程师的实际英语水平，面向广大学习群体，本教材定位为中级英语阅读与写作教材。教材包括8个单元，按2周学完1个单元计算，适合一个学期使用。

本教材的教学目标：

- 掌握英语最常用的2 000个单词以及1 000个左右的常用软件开发文档词汇。
- 能够读懂、使用以常用表达方式表述的各类英文软件开发文件。
- 能够借助词典等工具，使用规范英语语法写出若干种简单语法结构，如名词短语、动词短语、介词短语、简单句等。能够在填写软件开发文件的各个栏目时使用这些简单语法结构。

本教材特点：

- 阅读内容采用各类较新的软件开发专业文件，并进行必要的改编，以适合教学对象的水平。
- 侧重选择具有软件开发模板性质的文本，以便学习者在实际工作中模仿套用。
- 根据上下文提供常用词汇表，相关词汇只采用相应的专业释义。
- 提供词汇使用规则说明。
- 针对教学对象的英语水平编写各种阅读理解练习。题目指示用双语，



问题用中文，以方便学习者完全理解阅读材料。

- 为各篇范文编写写作练习。遵循从易到难的原则，提供从客观判断题到主观写作题等各种练习题型。
- 写作练习尽量模拟软件开发实际工作中常见的形式。
- 注意培养学生使用词典等工具查找并选择合适的英文表达的能力。
- 为各类练习提供参考答案。

教材内容的选择及处理方法：

- 本书阅读材料和写作材料全部选自真实的软件开发文件。
- 为了符合中级英语学习者的阅读和写作水平，作者对原始材料全部进行了简化。
- 简化语法，确保 95% 的课文都仅使用名词短语、动词短语和简单句。
- 只教授两种常用复合句句型，且专注于通用性高、易掌握的用法。
- 用易懂的方式讲解语法，避免使用专用术语。
- 简化词汇，用常用词替换了低频词。
- 词汇表中只列入与文本内容相关的释义，易于阅读理解。
- 将部分涉及语法的知识转化为词汇学习形式，易于中级学习者习得。
- 简化过程中确保原文意思不变。
- 简化的课文向学习者表明，可以用简单语法和常用词汇撰写各种软件开发文件。
- 学习者可以模仿教材内容进行软件开发文件的写作。

Contents

(目 录)

总 序	1
前 言	1
Unit 1 Use Case (用例)	1
1. Reading (阅读)	1
2. Writing (写作)	13
Unit 2 Software Requirements Specification (软件需求规约)	19
1. Reading (阅读)	19
2. Writing (写作)	32
Unit 3 Software Design Specification 1 (软件设计规约一)	35
1. Reading (阅读)	35
2. Writing (写作)	53
Unit 4 Software Design Specification 2 (软件设计规约二)	56
1. Reading (阅读)	56
2. Writing (写作)	66
Unit 5 User Interface Design (用户界面设计)	69
1. Reading (阅读)	69
2. Writing (写作)	75
Unit 6 Source Code Comment (源代码注释)	77





1. Reading (阅读)	77
2. Writing (写作)	91
Unit 7 Project Status Report (项目状态报告)	94
1. Reading (阅读)	94
2. Writing (写作)	101
Unit 8 User Manual (用户手册)	103
1. Reading (阅读)	103
2. Writing (写作)	115
Keys (参考答案)	118



Unit 1 Use Case

(用 例)

1. Reading (阅读)

1. 1 Read the texts on use cases(阅读以下各个用例文本)

Text 1 What Is a Use Case

- Scenario-based requirements process
- Focus on functional requirements
 - ◆ “what” the system will do instead of “how”
 - ◆ describe the interactions between the system and its users

1. Terminology of Use Case

- Actors
The people or external computer systems that will communicate with your system.
- Goals
The things that the Actors want to achieve.
- Use Case Title
Each Goal turns into a Use Case Title. The title is a verb phrase and describes what the Actor wants to do.
- Use Case Body
A bunch of things that describe the “joint requirements” of the system and its Actors: main sunny-day scenario, preconditions, postconditions, frequency, performance, business rules and notes.

• Non-use-case Requirements

Use Cases are usually only one-third of the volume of requirements. You also need business rules, computational algorithms, constraints, user interface guidelines and prototypes, and non-functional requirements.

2. Forms of Use Cases

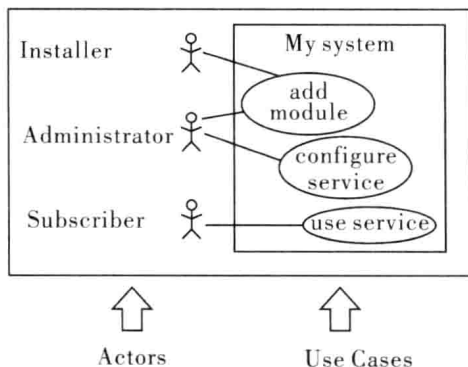


Fig. 1 - 1 A Use Case Diagram

Use Case Title: Withdraw Cash

Main Scenario:

1. User inserts his/her ATM card
2. System reads and validates the card information
3. User selects transaction and enters transaction details
4. System validates transaction
5. User collects cash and withdraws card
6. System updates the account and resets the system

Fig. 1 - 2 A Simple Text Use Case

3. Format of Use Case

(1) For each use case, write a “sunny-day scenario” (a scenario where everything works well).

(2) You can add failure branches later:

- Scenarios describing how to recover from failing steps
- Scenarios describing how to clean up when a scenario can not reach its goal

But do not spend a lot of time on failures until you write and review the success scenarios.

Use case writing is an iterative process...

(3) Make it clear “who has the ball” at each step of a scenario.

- Use active voice in each step.
- The “subject” of each sentence is either the name of an actor or “the system”.

4. An Example of a Use Case

- Goal: Buy a book on the web
- Who wants to do it? A Customer.
- What do they need? A web connection and a credit card.
- How do they start? Go to <http://www.amazon.com>.
- What is the normal sequence of events?

(1) Customer clicks on “search”, and types in a title.

(2) System displays a list of books that match.

(3) Customer chooses a book from the list, and clicks on “buy this book”.

(4) System prompts the user for credit card and delivery details.

(5) Customer provides the requested information.

(6) System checks if the credit card is valid.

(7) System displays the finished order, sends the order number to the Customer.

5. Notes on Use Case Writing

- Use cases are the easiest way to describe functional requirements.
- Write them in a language that makes sense to developers and users.

6. What Is a “Failure Scenario”?

- Description of how the system fails.
 - ◆ One of the steps of the main scenario fails, the system performs a smooth shutdown.
 - ◆ One of the steps fails, the system “recovers”.
 - ◆ Failure might be caused by a user (invalid input).
 - ◆ Failure could be resource exhaustion (ATM runs out of cash).

- ◆ Failure could be a real external failure (network failure, virus, etc.).

7. Important Rules for Reviewing Use Cases

- There are two important kinds of reviewers:
 - ◆ Users/stakeholders; they look for things that might be missing in the requirements.
 - ◆ Technical users of the requirements (designers, coders, and testers) : they want the requirements to be complete and unambiguous.
- Pretend to be one of the actors—can you write the first chapter of the user manual from the use cases?

Text 2 What Is a Use Case Scenario?

When a cardholder tries to withdraw cash from an ATM, things do not always happen the same way. Sometimes he gets his money, sometimes he might have insufficient funds, or the ATM may be out of cash. These can be examples of use case scenarios.

1. Capturing Use Case Scenarios with Descriptions

Once we understand the actor and the goal for a use case, and the key use case scenarios, we can begin some high-level interaction design. Actors interact with the system by pressing buttons, typing into text boxes and clicking on icons to achieve the goal.

A classic mistake at this early stage of design lies in the technical detail and a specific user interface design. This is the wrong time to make these low-level decisions. We first need to understand the business logic, so we can focus on the business goal. Instead of saying the user presses the “enter” button, we say the user confirms his choice.

A good way to write use case is to split the actions into two columns, one for actors and the other for the system. Then we can see the order of events in a scenario, and who is doing what.

These use case descriptions will form the basis of our high-level object oriented design, the UI design, system test design, and user documentation.

Cardholder	ATM
<ul style="list-style-type: none"> • select withdrawal option • specify amount • take card • take cash 	<ul style="list-style-type: none"> • display withdrawal options • check whether the card has sufficient funds • eject card • prompt cardholder to take card • dispense amount • prompt cardholder to take cash • debit cardholder's account • thank cardholder • display welcome and await next cardholder

Fig. 1 –3 A Use Case: The Order of Events and the Responsibilities of the Actor(s) and System in a Scenario

WARNING!

99% of teams are unaware that use case descriptions above are not system requirements documents. These are high-level interaction designs.

2. Describe Use Case Scenarios with Sequence Diagrams

If you are familiar with UML sequence diagrams, then you can describe a use case scenario above with a sequence diagram. This also shows the order of events, the interactions, and who is doing what.

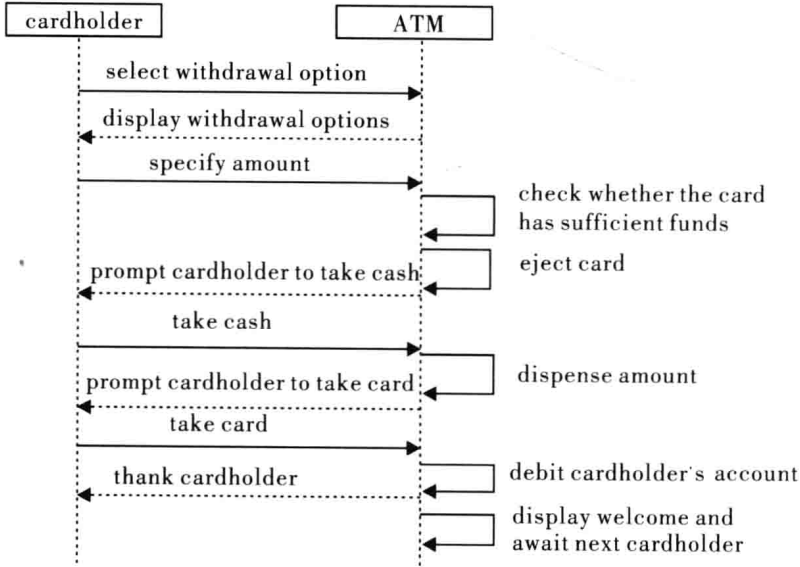


Fig. 1 - 4 A Use Case Scenario with a Sequence Diagram

1.2 High Frequency Vocabulary(常用词汇)

use case		用例	scenario	n.	场景, 剧情
based	adj.	基于……的	requirement	n.	需求
process	n/v.	过程; 处理	focus on		关注
functional	adj.	功能的	what	pron.	什么
is	v.	是	system	n.	系统
will	v.	会, 将要	do	v.	做
not	adv.	不, 非	how	adv.	如何
describe	v.	描述	interaction	n.	交互
between	prep.	在……之间	and	conj.	和, 与
its	adj.	它的	user	n.	使用者, 用户
one	num.	一个, 一	her	det.	她的
more	adj.	更多的, 更加	actor	n.	参与者
terminology	n.	术语	people	n.	人, 人们
computer	n.	计算机	communicate	v.	交流
with	prep.	与, 用	goal	n.	目标
thing	n.	事情, 事物	want	v.	要

achieve	<i>v.</i>	达到,取得	title	<i>n.</i>	标题,标签, 书名
each	<i>pron.</i>	每个	turn into		变成
verb phrase		动词短语	body	<i>n.</i>	主体
a bunch of		一组,一群	joint	<i>adj.</i>	联合的, 共同的
main	<i>adj.</i>	主要的	sunny day		艳阳天
precondition	<i>n.</i>	前提条件	postcondition	<i>n.</i>	后续条件
frequency	<i>n.</i>	频率	performance	<i>n.</i>	执行,性能
business	<i>n.</i>	业务,商务	rule	<i>n.</i>	规则
note	<i>n.</i>	注解,注释	non-	<i>prefix.</i>	非
are	<i>v.</i>	是	usually	<i>adv.</i>	通常
only	<i>adv.</i>	仅仅	one-third	<i>n.</i>	三分之一
you	<i>pron.</i>	你	volume	<i>n.</i>	量
also	<i>adv.</i>	也	need	<i>v.</i>	需要
computational	<i>adj.</i>	计算的	algorithm	<i>n.</i>	算法
interface	<i>n.</i>	界面	guideline	<i>n.</i>	指引,指导
prototype	<i>n.</i>	原型,框架	some	<i>det.</i>	一些
basic	<i>adj.</i>	基本的	concept	<i>n.</i>	概念
diagram	<i>n.</i>	图,图表	simple	<i>adj.</i>	简单的
text	<i>n.</i>	文本,文字	add	<i>v.</i>	添加
withdraw	<i>v.</i>	提取,取回	cash	<i>n.</i>	现金
his	<i>det.</i>	他的	card	<i>n.</i>	卡
read	<i>v.</i>	读	information	<i>n.</i>	信息
enter	<i>v.</i>	输入	detail	<i>n.</i>	详情,细节
important	<i>adj.</i>	重要的	format	<i>n.</i>	格式
documentation	<i>n.</i>	文件编撰, 记录	everything	<i>pron.</i>	事事,每件事
work	<i>v.</i>	运作,工作	well	<i>adv.</i>	好
can	<i>v.</i>	能	failure	<i>n.</i>	失败
where	<i>adv. /</i>	其中;	branch	<i>n.</i>	分支
	<i>conj.</i>	在……地方			
later	<i>adv.</i>	之后	recover	<i>v.</i>	恢复,复原
failing	<i>adj.</i>	失败的	step	<i>n.</i>	步骤