



软件工程

——统一过程的理论与案例

主编 梁颖红



北京交通大学出版社
<http://press.bjtu.edu.cn>

软 件 工 程

——统一过程的理论与案例

主 编 梁颖红
副主编 张春祥 王 芳
李 莉 陆公正

北京交通大学出版社

· 北京 ·

内 容 简 介

软件工程是讲述如何采用工程化的方法来开发软件,对学生提高软件开发能力有重要作用的一门学科。本书以统一过程的面向对象软件开发方法为主体,从理论方法和案例应用两大部分全面讲述了软件工程的理论、技术和方法。内容包括软件工程中常见问题、生命周期模型、统一过程理论、重用性与移植性等理论内容,还包括采用统一过程方法进行软件项目开发的完整实现过程。

本书在理论部分对重点内容进行了知识点总结;通过古董买卖软件项目讲述了采用统一过程方法开发软件的全过程,从需求分析、设计、实现和测试的全过程完整展现开发全貌,生成文档资料完整齐全。在讲述统一过程方法的同时兼顾了与面向过程方法的比较。全书知识点和案例过程醒目、完整。本书可作为高等院校计算机及相关专业本科、专科、高职及成人教育的软件工程课程的教材,也可作为软件开发人员的参考用书。

版权所有,侵权必究。

图书在版编目(CIP)数据

软件工程:统一过程的理论与案例/梁颖红主编. — 北京:北京交通大学出版社,2010.6
ISBN 978-7-5121-0133-3

I. ①软… II. ①梁… III. ①软件工程-高等学校:技术学校-教材 IV. ①TP311.5

中国版本图书馆CIP数据核字(2010)第104581号

责任编辑:张慧蓉

出版发行:北京交通大学出版社

电话:010-51686414

北京市海淀区高粱桥斜街44号

邮编:100044

印刷者:北京泽宇印刷有限公司

经 销:全国新华书店

开 本:185×260 印张:17.5 字数:427千字

版 次:2010年6月第1版 2010年6月第1次印刷

书 号:ISBN 978-7-5121-0133-3/TP·595

印 数:1~3 000册 定价:30.00元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

前 言

软件工程是计算机学科中非常重要的一个研究领域,自 20 世纪 60 年代以来,人们在软件理论、技术和方法方面进行了不懈的探索,取得了一些系统的成果,并且在软件开发实践中发挥了重要作用。然而,计算机和软件的发展十分迅速,软件工程理论和技术也在不断发展,这就要求软件工程教材也必须跟上这一发展要求,不断更新教材内容,反映最新的成熟软件开发理论和技术。

本书比较系统全面地涵盖了软件工程课程的理论和技术。既兼顾传统的、实用的软件开发方法,又包含了软件工程领域比较新颖的技术和方法。其特点是:

一,本书的理论部分,以知识点的形式对重点内容进行了总结和详细讲解,使枯燥的理论内容变得醒目和易于理解;

二,重点讲述了基于统一过程的面向对象软件开发方法,使用统一过程讲解了一个实例,并给读者安排了一个学期项目,使读者成为统一过程的理论学习和实践者;

三,本书包含了一个贯穿全文的极具代表性的案例,借助案例完整展示了软件开发的全过程,使读者从中学会了软件开发各环节的技术;

四,本书每章后增加了软件水平考试真题解析,对增强读者的应试水平很有帮助,部分章节给读者布置了学期项目,以配合教师的实例讲解,使读者能把学到的理论知识应用到实践中;

五,配有电子教案,教师可以免费获得该教案。

全书共分两篇,第 1 篇是基础理论及方法篇,主要讲解软件工程理论,共有 8 章,内容包括软件工程概述、软件生命周期模型、统一过程、面向对象的概念和建模、测试、项目管理和计划、用户界面的设计、软件的重用和配置管理。第 2 篇是综合案例及应用篇,结合古董买卖管理系统讲解了软件开发的全过程,内容包括需求、分析、设计、编码、测试和软件维护。

本书由多所学校的从事软件工程一线教学多年的教师亲自编写,第 1~3 章由江苏省现代企业信息化应用支撑软件工程技术研究开发中心的梁颖红编写,第 4.1~4.4 节、第 6、8 章由苏州市职业大学的王芳编写,第 4.5~4.10 节、第 5、7 章由苏州市职业大学的陆公正编写,第 9~11 章由哈尔滨理工大学的张春祥编写,第 12~14 章由东北林业大学的李莉编写。学期项目由李莉收集和整理。梁颖红组织了本书的编写,负责全书的策划、编审和统稿。

虽然我们经过精心的准备和调研,也多次修改手稿,但书中还难免存在不足和错误,希望使用本教材的教师和学生提出批评和建议,以便改进我们的工作,提高教材的质量。

主编联系方式:liangyh7036@126.com

编 者
2010 年 5 月

目 录

第1篇 基础理论及方法篇

第1章 软件工程概述	(3)
1.1 软件工程中的常见问题	(3)
1.1.1 什么是软件	(3)
1.1.2 什么是软件工程	(4)
1.1.3 软件工程涉及的人员有哪些	(6)
1.1.4 软件文档有哪些	(7)
1.1.5 什么是CASE	(9)
1.1.6 什么是好的软件	(11)
1.1.7 软件工程方法有哪些	(12)
1.2 软件工程的思维和方法	(15)
第2章 软件生命周期模型	(17)
2.1 软件过程和软件生命周期模型	(17)
2.2 瀑布模型	(19)
2.3 快速原型模型	(20)
2.4 螺旋模型	(22)
2.5 迭代—递增模型	(23)
2.6 同步—稳定模型	(25)
2.7 极限编程	(25)
第3章 统一过程	(29)
3.1 一维与二维生命周期模型	(29)
3.2 统一过程的5个核心 workflow	(30)
3.2.1 需求流	(30)
3.2.2 分析流	(31)
3.2.3 设计流	(31)
3.2.4 实现流	(32)
3.2.5 测试流	(32)
3.3 统一过程的4个阶段	(34)
3.3.1 开始阶段	(34)
3.3.2 细化阶段	(35)

3.3.3 构建阶段	(35)
3.3.4 转换阶段	(35)
3.4 软件能力成熟度模型	(36)
第4章 面向对象的概念和建模	(42)
4.1 模块	(42)
4.2 模块设计原则	(44)
4.2.1 内聚	(44)
4.2.2 耦合	(47)
4.2.3 抽象	(50)
4.2.4 数据封装	(50)
4.2.5 信息隐藏	(50)
4.3 对象和类	(51)
4.3.1 对象	(51)
4.3.2 类	(51)
4.4 类的基本关系	(52)
4.4.1 继承	(52)
4.4.2 泛化	(52)
4.4.3 聚合	(52)
4.4.4 关联	(52)
4.5 UML 简介	(53)
4.5.1 UML 语言的基本结构	(53)
4.5.2 UML 语言中的图形	(53)
4.5.3 UML 语言的扩展机制	(55)
4.6 用例图	(55)
4.6.1 用例图	(56)
4.6.2 识别参与者	(57)
4.6.3 识别用例	(57)
4.7 类图	(59)
4.7.1 类的 UML 表示	(59)
4.7.2 类间关系的 UML 表示	(60)
4.8 交互图	(62)
4.8.1 顺序图	(62)
4.8.2 协作图	(64)
4.9 状态图和活动图	(64)
4.9.1 状态机	(64)
4.9.2 状态图	(65)
4.9.3 活动图	(66)
4.10 实现图	(68)

4.10.1	组件图	(68)
4.10.2	部署图	(69)
第5章	测试	(72)
5.1	软件测试概述	(72)
5.1.1	软件错误和缺陷	(72)
5.1.2	软件质量	(74)
5.1.3	软件测试定义	(76)
5.1.4	软件测试分类	(77)
5.1.5	软件测试工具	(79)
5.2	软件测试过程	(80)
5.2.1	确定测试方法	(81)
5.2.2	测试的计划	(81)
5.2.3	测试的设计	(81)
5.2.4	测试的执行	(83)
5.2.5	测试停止判断	(83)
5.3	正确性证明和测试	(87)
5.4	静态测试	(88)
5.4.1	代码走查	(88)
5.4.2	代码审查	(88)
5.5	测试指标	(89)
5.5.1	正确性	(90)
5.5.2	可靠性	(90)
5.5.3	效率	(90)
5.5.4	健壮性	(90)
5.5.5	实用性	(90)
第6章	项目管理和计划	(92)
6.1	软件规模估算	(92)
6.1.1	代码行的估算	(92)
6.1.2	功能点的估算	(93)
6.2	成本估算	(95)
6.2.1	静态单变量模型	(95)
6.2.2	动态多变量模型	(96)
6.2.3	CoCoMo 模型	(96)
6.3	进度管理	(98)
6.3.1	项目进度管理过程	(98)
6.3.2	进度估算	(99)
6.3.3	进度计划	(100)

6.4	风险管理	(104)
6.4.1	风险	(104)
6.4.2	风险识别	(105)
6.4.3	风险评估	(105)
6.4.4	风险规划	(106)
6.4.5	风险管理过程	(107)
6.5	软件度量	(107)
6.5.1	软件质量度量	(108)
6.5.2	软件复杂性度量	(111)
6.5.3	软件可靠性度量	(113)
6.6	软件质量保证	(114)
6.6.1	概念	(114)
6.6.2	评审	(114)
6.6.3	测试	(116)
6.7	项目团队管理	(116)
6.7.1	组织结构	(116)
6.7.2	程序设计小组的组织形式	(117)
6.7.3	项目经理	(117)
6.7.4	项目组成员	(118)
6.8	项目管理计划	(118)
6.8.1	项目管理计划	(119)
6.8.2	计划文档	(119)
6.8.3	项目管理计划综合案例——古董买卖估价管理软件项目开发计划	(121)
第7章	用户界面的设计	(126)
7.1	用户界面设计的规则	(126)
7.1.1	从用户的角度出发	(126)
7.1.2	尽量给用户提供更多的方便	(127)
7.1.3	保持界面一致	(127)
7.2	用户界面的分析和设计	(128)
7.2.1	用户界面分析和设计模型	(128)
7.2.2	用户界面分析和设计过程	(129)
7.3	用户界面分析	(130)
7.3.1	用户分析	(130)
7.3.2	任务分析和建模	(131)
7.3.3	显示内容分析	(132)
7.3.4	工作环境分析	(133)
7.4	用户界面设计	(133)
7.4.1	应用界面设计步骤	(134)

7.4.2	用户界面设计模式	(135)
7.4.3	界面设计中的问题	(137)
7.4.4	设计评估	(138)
第8章	软件的重用和配置管理	(141)
8.1	什么叫重用	(141)
8.1.1	重用的概念	(141)
8.1.2	重用的发展历史	(141)
8.1.3	可重用的软件成分	(142)
8.1.4	重用过程	(142)
8.1.5	重用的效益	(143)
8.2	重用的障碍	(144)
8.3	构件与重用	(145)
8.3.1	构件的概念	(145)
8.3.2	构件软件的标准	(146)
8.4	什么叫可移植性	(148)
8.5	实现可移植性的技术	(149)
8.6	配置管理	(149)
8.6.1	软件配置管理基本概念	(149)
8.6.2	软件配置管理过程	(150)

第2篇 综合案例及应用篇

第9章	需求	(157)
9.1	需求概述	(157)
9.2	应用域理解	(158)
9.3	建立商业模型	(159)
9.3.1	调查问卷	(159)
9.3.2	访谈	(159)
9.3.3	观察手工操作及数据处理	(160)
9.3.4	用例	(160)
9.4	初始需求	(161)
9.5	古董买卖管理软件开发项目	(161)
9.5.1	古董买卖领域的初始理解	(162)
9.5.2	古董交易的商业模型	(163)
9.5.3	古董交易的初始需求	(164)
9.5.4	古董交易的需求继续	(166)
9.5.5	测试阶段	(169)
9.6	需求流的 CASE 工具	(170)
9.7	需求阶段的注意事项	(171)

第 10 章 分析	(173)
10.1 结构化分析	(173)
10.1.1 结构化分析方法的表示	(173)
10.1.2 结构化分析的步骤	(176)
10.2 面向对象分析	(177)
10.2.1 实体类模型	(178)
10.2.2 动态模型	(179)
10.2.3 功能模型	(179)
10.2.4 抽象边界类和控制类	(180)
10.2.5 面向对象分析中的相关问题	(180)
10.3 面向对象分析：古董商买卖管理系统案例	(181)
10.3.1 古董商问题的功能模型	(181)
10.3.2 古董商问题的实体类模型	(182)
10.3.3 古董商问题的动态模型	(185)
10.3.4 古董商问题的抽象边界类和控制类	(187)
10.3.5 古董商问题的分析测试	(187)
10.4 面向对象分析的 CASE 工具	(187)
10.5 面向对象分析阶段注意事项	(187)
第 11 章 设计	(190)
11.1 结构化设计	(190)
11.1.1 数据流图的类型	(190)
11.1.2 古董商问题的结构化设计	(192)
11.2 面向对象设计	(195)
11.3 设计阶段的测试	(200)
11.4 设计阶段的 CASE 工具	(201)
11.5 面向对象设计阶段的注意事项	(201)
第 12 章 编码	(204)
12.1 程序设计语言	(205)
12.1.1 程序设计语言的分类	(205)
12.1.2 程序设计语言的选择	(206)
12.2 应用编程标准的意义	(207)
12.3 编程的指导方针	(208)
12.3.1 使用合理的控制结构	(208)
12.3.2 设计适合的算法	(209)
12.3.3 选择恰当的数据结构	(210)
12.3.4 一般指导原则	(210)
12.4 编码中的文档问题	(212)

12.5	古董买卖管理软件案例	(214)
第13章	测试	(221)
13.1	系统前期阶段的测试	(222)
13.2	系统实现阶段的测试	(223)
13.2.1	单元测试	(223)
13.2.2	集成测试	(225)
13.2.3	系统测试	(228)
13.3	古董买卖管理软件案例	(231)
13.4	白盒测试实例分析	(243)
第14章	软件维护	(249)
14.1	系统的类型	(249)
14.2	软件维护的任务及特点	(251)
14.3	软件维护的类型	(252)
14.4	软件维护的过程	(253)
14.5	软件的可维护性	(256)
14.6	软件再工程	(257)
14.6.1	源代码转换	(257)
14.6.2	反向工程	(258)
14.6.3	程序结构改善	(258)
14.6.4	程序模块化	(259)
14.6.5	数据再工程	(259)
14.7	古董买卖管理软件维护案例	(261)
参考文献		(265)

第 1 篇

基础理论及方法篇

本书共由两篇组成，第 1 篇是基础理论及方法篇，主要讲述软件工程学科所涉及的理论知识和方法，对重点内容进行了知识点的总结；第 2 篇是综合案例及应用篇，通过古董买卖管理软件案例讲述软件开发的各个阶段。

第 1 篇共由 8 章组成，分别是：

第 1 章 软件工程概述，主要论述软件工程的基本概念、原理及相关知识，能帮助读者对软件工程的原理和方法有一个概括的、本质的认识；

第 2 章 软件生命周期模型，对瀑布模型、原型模型等传统模型和统一过程模型、敏捷过程等新的模型都进行了介绍，总结了各个模型的优缺点和适用场合，为软件开发人员选择模型提供了理论依据；

第 3 章 统一过程，讲述了统一过程的四个阶段和五个工作流，使读者对统一过程这一方法有了深入的理解，为后续的案例和应用部分奠定了基础；

第 4 章 面向对象的概念和建模，对内聚和耦合的类型采用实例进行了讲解，对对象之间的关系类型和 UML 也做了重点阐述；

第 5 章 测试，主要讲述与测试有关的理论知识，测试的具体技术在第二部分中通过案例进行阐述；

第 6 章 项目管理和计划，对项目计划的制定和管理方面涉及的知识进行了介绍，使学生重视项目的管理和掌握管理的技能；

第 7 章 用户界面的设计，通过实例讲解软件界面设计中应注意的事项，使学生掌握软件设计的技术和理论；

第 8 章 软件的重用和配置管理，讲解可移植性和重用性，运用实例讲解了重用和移植的技术。

第 1 章

软件工程概述

本章重点

软件工程的概念和作用、软件开发的方法、人员和过程。

本章用途

本章从总体上论述软件工程的基本概念、原理及相关知识，能帮助读者对软件工程方法在软件开发中的应用有总体的认识。

软件工程（Software Engineering, SE）是指导计算机软件开发和维护的工程学科。“软件工程”这一概念是 1968 年北大西洋公约组织（NATO）在联邦德国召开的为了解决“软件危机”的国际会议上首次提出的，是伴随着软件的发展，为了解决软件危机而诞生的一门新兴学科。随着计算机在各行各业的普及应用，计算机软件的需求不断增加，作为指导计算机软件开发软件工程学科也变得越来越重要。

1.1 软件工程中的常见问题

软件工程是一门综合性的学科，它涵盖了数据库、编程语言、算法设计等相关学科，涉及的范围比较广，为了使读者对软件工程学科有更好的认识，本节将对软件工程的常见问题进行解释。

1.1.1 什么是软件

可能有些人认为软件（Software）等同于计算机程序，其实这种理解是不全面的。软件是程序和所有使程序正确运行所需要的相关文档和配置信息，即软件 = 程序 + 文档。一个软件系统通常包含大量独立的程序、用于设置这些程序的配置文件、描述系统结构的系统文档和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 站点。

程序（Program）是用程序设计语言描述的、适合于计算机处理的语句序列。它能控制计算机硬件的运行、处理用户提供的或机器运行过程中产生的各类数据并输出结果。

文档（Document）是一种数据媒体和其中所记录的技术数据和信息，包括计算机列表和打印输出。文档中记录了计算机软件的要求、设计或细节，说明软件的能力和限制条件，还记录软件开发活动和阶段成果，供人进行查阅。文档是专业开发人士和用户沟通的手段，还可以用于软件开发过程的管理和维护。

软件具有如下特点。

(1) 软件是一种逻辑实体，不是具体的物理实体。

(2) 软件产品的生产主要是研制。软件不存在磨损和老化问题，但存在退化问题。

(3) 软件成本昂贵，其开发方式目前尚未完全摆脱手工生产方式。

(4) 软件是一种创造性的思维活动，可用不同的方法来开发，也可产生不同结果的软件产品。

(5) 软件具有“复杂性”，其开发和运行常受到计算机系统的限制。

目前，软件应用于社会生活的各个方面，如科学研究、教育、工农业生产等。软件产品常用的分类有以下几种。

1. 按服务对象

(1) 通用软件产品。这类软件产品由软件开发机构制作，在市场上公开销售，可以独立使用。这类软件产品有：数据库软件、字处理软件、绘图软件以及工程管理工作等。

(2) 定制软件产品。这类软件产品受特定的客户委托，由软件承包商专门为这类客户开发。这类软件有：电子设备的控制系统、特定的业务处理系统和空中交通管制系统等。

这两类产品的一个重要区别在于：在通用软件中，软件描述由开发者自己完成，而定制软件产品的软件描述通常是由客户给出，开发者必须按客户要求开发。

然而，这两类产品之间的界线正在变得越来越模糊。现在更多的公司从一个通用软件产品开始进行定制处理，来满足特别客户的具体要求。企业资源规划（ERP）系统，如 SAP 系统，就是这种方法的一个最好见证。像这样的一个庞大而复杂的系统，需要通过嵌入一系列信息，比如说业务和操作规则以及各种报表等，以适应一个新企业的需求。

2. 按功能

(1) 系统软件。管理、控制和维护计算机系统内的各种软硬件资源，使其充分发挥作用，提高计算机的工作效率。操作系统是典型的系统软件。

(2) 应用软件。为计算机的特定应用提供特定功能，为解决实际问题而编写的程序。如信息系统，Office 软件等。

3. 按规模

(1) 小型程序。程序的长度不超过 2 000 行，与其他程序没什么联系，如学校课程设计中学生自己编写的程序，数据操作人员使用的小型商业程序。在开发这类程序时，应贯彻软件工程中的技术标准和表示方法（Notations），按标准编写文档，并系统地进行复审。当然，上述工作的正规程度不必像开发大程序时那样严格。

(2) 中型程序。这些程序可能与其他程序有少量联系，如小型管理信息系统、仓库系统等。在开发中如能系统地应用软件工程的原理，对改进软件质量、提高程序员生产率和满足用户需求，都将有很大好处。

(3) 大型程序。程序编码长度可达 5 ~ 10 万行，常与别的程序或软件系统有种种联系，如大型编译程序、某些图形软件。毫无疑问，所有大型以上软件的生产必须从头至尾坚持软件工程的方法，严格遵守标准文档格式和正规的复审制度。

1.1.2 什么是软件工程

早期软件开发中，多数软件项目要推迟几年才能完成，而且比预计的费用高，不可靠，

难以维护。结果是硬件成本在下降而软件成本却呈快速增长之势，使软件开发陷入危机。

建筑业采用工程化方法来建造系统，在建造复杂系统之前必须进行结构设计。

20 世纪 70 年代初，自“软件工程”这一概念提出以来，提倡采用工程化方法来进行软件系统开发，在设计一个系统之前，软件工程师必须了解它的需求。这就意味着必须做需求分析，而且必须用某种建模语言说明。必须用抽象和“分治”的方法对系统进行模块化。在程序员编写代码之前，对每个模块都要进行详细说明，并且必须定义与其他模块的接口。

软件工程是一类工程。IEEE [IEEE93] 给出的软件工程定义如下：“软件工程是将系统化的、规范化的、可度量的方法应用于软件开发、运行和维护的过程，即将工程化应用于软件中的方法的研究。”

工程是将理论和知识应用于实践的科学。它借鉴了传统工程的原则和方法，以求高效地开发高质量软件。其中应用了计算机科学、数学和管理科学。计算机科学和数学用于构造模型与算法，工程科学用于制定规范、设计范型、评估成本及确定权衡，管理科学用于计划资源、对质量和成本进行管理。

软件工程的框架可概括为：目标、过程和原则。

1. 软件工程的目标

软件工程的目标是生产具有正确性、可用性且开销合宜的产品。一个成功的软件系统需达到以下 8 个条件：

- (1) 付出的开发成本较低；
- (2) 达到了要求的软件功能；
- (3) 取得了较好的软件性能；
- (4) 开发的软件易于移植；
- (5) 需要较低的维护费用；
- (6) 能按时完成开发任务；
- (7) 及时交付产品；
- (8) 开发的软件可靠性高。

2. 软件工程的过程

软件工程的过程是生产一个最终能满足需求且达到工程目标的软件产品所需要的步骤。软件工程过程主要包括开发过程、运作过程、维护过程。包括需求、设计、实现、确认以及维护等活动。

3. 软件工程的原则

软件工程的原则是指围绕工程设计、工程支持以及工程管理在软件开发过程中必须遵循的原则。有以下四项基本原则。

1) 选取适宜的开发范型

须认识需求定义的易变性，采用适宜的开发范型予以控制，以保证软件产品满足用户的要求。

2) 采用合适的设计方法

在软件设计中，通常要考虑软件的模块化、抽象性与信息隐蔽性、局部化、一致性以及适应性等特征。合适的设计方法有助于这些特征的实现，以达到软件工程的目标。

3) 提供高质量的工程支持

在软件工程中，软件工具与环境对软件过程的支持颇为重要。好的软件工具与环境可以帮助软件开发人员获得高质量的软件产品。

4) 重视开发过程的管理

管理是生产的保障，因此，仅当软件过程得以有效管理时，才能实现有效的软件工程。

软件工程技术的两个明显特点是规范化和文档化。美国工业界和政府部门开始认识到，在软件开发中，最关键的问题是软件开发组织不能很好地定义和管理其软件过程，从而使一些好的开发方法和技术起不到所期望的作用。目前的软件开发过程没有明确规定，文档不完整，也不规范，软件项目的成功往往归功于软件开发组的一些杰出个人或小组的努力。只有坚持软件工程的四条基本原则，既重视软件技术的应用，又重视软件工程的支持和管理，并在实践中贯彻实施，才能高效地开发出高质量的软件。

1.1.3 软件工程涉及的人员有哪些

软件开发的一个关键部分是客户与开发人员之间的交流，如果交流失败，那么系统也将失败。必须在构建帮助解决问题的系统之前，理解客户想要什么以及他们需要什么。要做到这一点，必须知道软件开发所涉及的人员。

通常情况下，参与项目的人员分为三类：客户、用户和开发人员。客户（Customer）是为将要开发的软件系统支付费用的公司、组织或个人。开发人员（Developer）是为客户构建软件系统的公司、组织或个人，其中包含任何协调并指导程序员和测试人员的管理人员。用户（User）是将实际使用系统的人，包括坐在终端前的人、提交数据的人和阅读输出的人。尽管就某些项目而言，客户、用户、开发人员是同一个人或同一组人，多数情况下，他们各不相同。参与项目人员及其关系详见知识点 1-1。

知识点 1-1：参与项目的三类人员及其关系

☞ 知识点 1-1：参与项目的三类人员及其关系

知识点简述：通常，参与项目的人员分为三类：顾客、用户和开发人员。

顾客：支付软件系统开发费用的公司、组织或个人。

用户：实际使用开发系统的人。

开发人员：为顾客构建软件系统的公司、组织或个人。

关系：顾客控制资金，进行合同谈判并在验收报告上签字，然而，有时候顾客并不是用户。

案例 1：公司甲与公司乙签订了一份合同，为公司建立财务管理系统

案例简述：公司甲的总经理可以向公司乙的代表说明软件系统的要求并签订合同。然而，总经理并不直接使用这套系统，使用的人将是会计或记账的职员。

案例分析和答案：顾客：公司甲总经理；

用户：公司甲会计或记账的职员；

开发人员：公司乙。