



主编 罗嗣卿 李 莉 吴 頔

RUANJIAN GONGCHENG
DAOLUN

软件工程导论



东北林业大学出版社

软 件 工 程 导 论

主编 罗嗣卿 李莉 吴頔

东北林业大学出版社

图书在版编目(CIP)数据

软件工程导论 / 罗嗣卿, 李莉, 吴頔主编. -- 哈尔滨: 东北林业大学出版社, 2011. 7
ISBN 978-7-81131-871-5

I. ①软… II. ①罗… ②李… ③吴… III. ①软件工程 IV. ①TP311.5

● 中国版本图书馆CIP数据核字(2011)第136363号

责任编辑: 杨秋华

封面设计: 彭宇



软件工程导论

Ruanjian Gongcheng Daolun

主编 罗嗣卿 李莉 吴頔

东北林业大学出版社出版发行

(哈尔滨市和兴路26号)

东北林业大学印刷厂印装

开本 787 × 960 1/16 印张 18.25 字数 350 千字

2011年7月第1版 2011年7月第1次印刷

印数 1—1 000 册

ISBN 978-7-81131-871-5

定价: 36.50 元

前 言

随着计算机软件的发展,软件工程理论和技术也在不断发展,这就亟须软件工程教材必须跟上这一发展要求,不断更新教材内容以反映最新的、成熟的软件开发理论和技术。本书比较系统全面地阐述了软件工领导论课程的理论和技术,既兼顾传统的、实用的软件开发方法,又涵盖了软件工程领域最新的技术和方法,将当前的软件工程理论与方法进行了很好的融合;借助贯穿于全书的极具代表性的实例,细致而巧妙地讲解了各种相关概念,使读者从中学会软件开发各个环节的技术。这是本书区别于其他传统软件工程领域教材之处,也使其更加适应软件工程相关专业的人才培养要求。

本书共11章,主要包括软件工程概述、软件生命周期模型、统一过程、需求、分析、编码、测试、软件维护、软件项目管理和CMM等。

本书由东北林业大学和哈尔滨职业技术学院从事软件工程一线教学的教师编写。第1章、第2章、第3.2节、第3.3节、第3.4节、第8.3节、第11.4节、第11.5节、第11.6节、第11.7节由东北林业大学罗嗣卿编写,第4章、第8.2.2节、第8.2.3节由东北林业大学李琰编写,第5章由东北林业大学赵玉茗编写,第6章、第10章由东北林业大学吴顿编写,第7章、第8.1节、第8.2.1节、第9章由东北林业大学李莉编写,第11.1节、第11.2节、第11.3节由哈尔滨职业技术学院李晓会编写。金蝶国际软件集团有限公司宣伟编写了第3.1节,同时对本书中的案例实践部分提出了很多宝贵的意见。罗嗣卿和李莉组织了本书的编写,负责全书的策划和统稿。

由于编者水平有限,本书难免还存在一些不足和缺点,希望读者批评和指正。

编 者

2011年4月

目 录

1 软件工程概述	(1)
1.1 软件工程中的基本概念	(1)
1.2 软件工程的思维和方法	(19)
2 软件生命周期模型	(22)
2.1 软件工程过程和软件生命周期模型	(22)
2.2 瀑布模型	(25)
2.3 快速原型模型	(27)
2.4 螺旋模型	(29)
2.5 迭代递增模型	(31)
2.6 同步 - 稳定模型	(33)
2.7 极限编程	(34)
3 统一过程	(38)
3.1 Rational 统一过程(RUP)	(38)
3.2 RUP 统一过程的核心 workflow	(40)
3.3 RUP 统一过程的四个阶段	(45)
3.4 软件能力成熟度模型	(47)
4 需求	(54)
4.1 需求概述	(54)
4.2 需求工程过程	(56)
4.3 需求获取	(57)
4.4 需求描述	(65)
4.5 运输公司业务管理系统开发项目案例	(69)
4.6 需求有效性验证	(82)
4.7 需求管理	(82)
4.8 需求 CASE 工具	(83)
5 分析	(86)
5.1 结构化分析	(86)
5.2 面向对象分析	(98)
5.3 面向对象分析:运输公司业务管理系统案例	(104)

2 软件工程导论

5.4	面向对象分析的 CASE 工具	(126)
5.5	面向对象分析阶段注意事项	(127)
6	设计	(130)
6.1	设计的基本原则	(130)
6.2	结构化设计	(138)
6.3	面向对象设计	(148)
6.4	设计文档	(153)
6.5	设计阶段的 CASE 工具	(157)
7	编码	(159)
7.1	程序设计语言	(160)
7.2	应用编程标准的意义	(162)
7.3	编程的指导方针	(164)
7.4	编码中的文档问题	(169)
7.5	运输公司业务管理系统案例	(172)
8	测试	(177)
8.1	系统前期阶段的测试	(178)
8.2	系统实现阶段的测试	(180)
8.3	运输公司业务管理系统案例	(192)
9	软件维护	(213)
9.1	系统的类型	(213)
9.2	软件维护的任务及特点	(216)
9.3	软件维护的类型	(217)
9.4	软件维护的过程	(219)
9.5	软件的可维护性	(222)
9.6	软件再工程	(223)
9.7	运输公司业务管理系统维护案例	(228)
10	软件项目管理	(233)
10.1	项目评估	(233)
10.2	项目计划	(240)
10.3	人员管理	(244)
10.4	风险管理	(247)
11	CMM 及其应用	(251)
11.1	CMM 发展概述	(252)
11.2	CMM 的必要性	(255)

11.3	CMM 体系结构	(256)
11.4	CMM 实施	(265)
11.5	CMM 在国内的现状	(272)
11.6	CMM 与 ISO9001 的比较	(273)
11.7	CMMI 综述	(275)
参考文献	(281)

1 软件工程概述

【本章重点】 软件工程的基本概念，软件的发展、定义及特征，软件危机的表现及解决途径，软件工程的基本原理及目标，软件开发的思维和开发方法。

【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，能帮助读者了解软件工程的发展历史，并对软件工程方法在软件开发中的应用有总体的认识。

随着计算机应用的深入发展，人们对软件的需求量急剧增加，但此时计算机软件的开发技术却远远没有跟上硬件技术的发展，同时早期所形成的一些错误概念和做法曾严重地阻碍了计算机软件的开发，这些导致了20世纪60年代软件危机的发生，软件开发技术已经成为影响计算机系统发展的“瓶颈”。20世纪60年代后期，西方的计算机科学家开始认真研究解决软件危机的方法，提出借鉴工程界严密完整的工程设计思想来指导软件的开发与维护，并取得了可喜成果，从而一门新的学科——软件工程学（Software Engineering）诞生了。

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，这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 软件工程中的基本概念

计算机系统包括计算机硬件和软件两部分，硬件在软件的控制下进行逻辑处理和数学运算，实现特定的功能。早期的软件就是程序，开发人员的组织形式是个体，程序的质量完全取决于个人的编程技巧。随着计算机技术的发展，软件的规模和复杂程度也逐渐增加，软件的开发过程逐步产业化，参与软件开发的人员越来越多，软件的开发面临新的问题，如软件开发的规范

化管理、软件开发技术和方法、软件的质量管理等。软件工程就是为适应软件产业化发展的需要，逐步发展起来的有关软件项目开发的工程方法学。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，涉及的范围比较广泛。

1.1.1 什么是软件

计算机软件不仅仅是程序，软件（software）是计算机系统中与硬件相互依存的另一部分，是包括程序、数据及其相关文档的完整集合，即软件 = 程序 + 文档。

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。

数据（data）是进行通信、解释和处理而使用的信息的形式化表现形式。

文档（document）是与程序开发、维护和使用有关的图文材料。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。文档是专业开发人员和用户沟通的手段，还可以用于软件开发过程的管理和维护。

为了能全面、正确地理解计算机和软件，必须了解软件的特点。与硬件相比，软件主要有以下特点。

①软件是一种逻辑实体，不是具体的物理实体。硬件是有形有色、看得见摸得着的，而软件是无形无色、看不见摸不着的。软件正确与否，是好是坏，一直要到程序在机器上运行才能知道，这给设计、生产和管理带来了许多困难。

②软件与硬件的生产方式不同。软件在开发过程中没有明显的制造过程，也不像硬件那样一旦研制成功就可以重复制造，并且可在制造过程中进行质量控制以保证产品的质量。软件是通过人们的智力活动，把知识与技术转化成信息的一种产品。某一软件项目一旦研制成功，就可以大量地复制同一内容的副本，应用到更多的地方。因为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，所以对软件的质量控制，必须着重在软件开发方面下工夫。

③软件产品的生产主要是研制。硬件是有损耗的，会产生磨损和老化而使故障率增加甚至损坏。其解决的办法是换上一个相同的硬件。而软件不存在磨损和老化的问题，但却存在退化的问题。在软件的生存期中，为了使它

能够克服以前没有发现的故障、适应硬件和软件环境的变化以及满足用户新的要求，必须要多次修改软件，而每次修改都不可避免地引入新的错误，随着一次次地修改，导致软件失效率升高，从而使软件退化。

④软件是复杂的。有人认为，人类能够创造的最复杂的产物是计算机软件。软件的复杂性一方面来自它所反映的实际问题的复杂性；另一方面，也来自程序结构的复杂性。软件技术的发展明显落后于复杂的软件需求，并且随着时间的推移，这个差距日益加大。

⑤软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动，研制成本是比较高的。在20世纪50年代末，软件的成本大约占计算机系统总成本的10%，大部分成本花在硬件上；但今天，这个比例完全颠倒过来，软件的成本大大超过硬件的成本。同时，其开发方式目前尚未完全摆脱手工生产方式。

在现代社会中，软件应用于多个方面。典型的软件有电子邮件、嵌入式系统、人机界面、办公套件、操作系统、编译器、数据库、游戏等；各个行业几乎都有计算机软件的应用，如工业、农业、银行、航空、政府部门等。这些应用促进了经济和社会的发展，使得人们的工作更加高效，同时提高了生活质量。软件产品的分类有以下几种。

①按服务对象分类。

A. 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具等。

B. 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。

②按功能分类。

A. 系统软件：管理、控制和维护计算机系统中的各种软硬件资源，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。

B. 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按规模分类。

A. 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等。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。

B. 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方面都将有很大的好处。

C. 大型程序：程序编码长度为 5 万~10 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 软件危机

早在 20 世纪 60 年代，人们已经意识到编程的技术落后于软件的发展。没有合理的开发计划，许多程序员没有接受正规的培训，组织和管理方法不科学，造成软件不能按时交付使用；程序也不像使用者期望的那样工作，而且在使用的过程中又发现很多的错误。

软件危机是指在计算机软件的开发、使用和维护过程中遇到的一系列严重问题。

1968 年，北大西洋公约组织（NATO）的计算机科学家在联邦德国召开的国际学术会议上第一次提出了“软件危机”这个名词。

①软件危机的表现。应该说，自从计算机诞生以来，软件危机就一直存在。软件危机主要表现在以下几个方面。

A. 人们对软件开发的成本和进度的估计常常不够准确。盲目制订开发计划，进度无法执行，造成计划进度拖延，经费预算超支。

B. 在开发的初期，软件需求不够明确，开发过程中又未能和用户及时交换意见，致使开发出的软件不能满足用户的需求，甚至无法使用。

C. 在软件开发过程中，没有统一、公认的方法和规范进行指导，且设计和实现过程的资料很不完整，这使得软件出现问题后很难维护。

D. 不重视软件的测试工作, 提交给用户的软件质量差, 在运行中暴露出大量问题, 软件产品的质量往往不可靠, 有可能导致系统不能正常运行或产生重大的事故。

E. 软件开发生产率提高的速度远远跟不上日益增长的软件需求, 满足不了社会发展的需要。

②软件危机产生的原因。

A. 所编写的程序代码在计算机上运行之前, 很难检验其正确性, 且质量也难以评价, 因此管理和控制软件的开发过程相当困难。同时, 软件不同于一般的程序, 它的显著特点就是庞大, 因此要保证其质量, 不仅仅涉及技术问题, 还涉及严格的科学管理和组织。

B. 目前相当多的软件从业人员利用长期积累开发和使用计算机系统的经验进行软件的设计和开发, 缺少采用规范的工程化的方法, 这是产生软件危机的主要原因。

C. 开发和管理人员对用户的要求没有完整、准确的认识就进行程序的设计, 这也是导致软件开发工程失败的主要原因之一。事实上, 软件开发和管理人员不仅要重视开发还要重视问题的定义, 使产品满足用户的需求。只有软件开发人员和用户多做沟通, 才能全面、准确、具体地了解用户的需求。

软件开发的基本过程应该是从软件开发出现的问题定义开始, 来明确要解决什么问题; 然后进行可行性分析, 确定该问题是否存在解决的方法; 再进行需求分析, 也就是具体深入了解用户的需求, 使所开发的软件的系统功能和用户的需求达成一致。经过软件定义阶段的准备工作以后, 才能进入软件的开发时期。

D. 软件的管理技术不能满足现代软件的开发需要, 缺少软件质量管理的规范, 软件的文档缺乏统一性和一致性, 导致缺少管理的依据。同时, 人们在软件的开发和维护关系的问题上存在错误概念, 侧重软件的开发, 忽视了定义时期的工作, 没有正确全面理解用户的需求, 可能造成软件使用后期出现问题进行修改, 而产生较大的代价。如果这些问题能及早发现, 就可以避免错误的出现或者对其问题在开发阶段进行改正。

③缓解软件危机的途径。到 20 世纪 60 年代末期, 软件危机已相当严重。要想解决软件危机, 必须做好以下几方面的工作。

A. 加强软件开发过程的管理, 做到组织有序、各类人员协同配合, 共同保证工程项目完成, 避免软件开发过程中个人单干的现象。

B. 推广使用开发软件的成功技术和方法, 并且不断探索更好的技术和

方法，消除一些错误的概念和做法。

C. 开发和利用好软件工具，支持软件开发的全过程。

如何做好上述三方面的工作来缓解软件危机呢？计算机科学家们经过多年的研究，提出了“软件工程”的概念，即应用现代工程的原理、技术和方法进行软件的开发、管理、维护和更新，于是，开创了计算机科学技术的一个新的研究领域——软件工程。几十年来，软件工程在软件开发方法、工具和管理等方面的研究与应用已经取得了很多成果，并已大大缓解了软件危机所带来的影响。

1.1.3 什么是软件工程

软件工程学是一门指导软件开发和维护的工程学科，是为了经济地获得能够在实际机器上有效运行的可靠软件而建立和使用的一系列完善的工程化原则。它应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则、方法来生产软件，以达到提高质量、降低成本的目的。1983年，国际权威机构 IEEE 给出的软件工程定义为：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。

软件工程包括 3 个要素：方法、工具和过程。

软件工程方法为软件开发提供了“如何做”的技术，是指导研制软件的某种标准规范。

软件工程方法包括多方面的任务，如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试以及维护等。软件工程方法常采用某种特殊的语言或图形的表达方法及一套质量保证标准。

软件工程的框架可概括为目标、过程和原则。

① 软件工程目标是生产具有正确性、可用性以及合理成本的产品。从技术和管理上采取多项措施以后，组织实施软件工程项目最终目的是保证项目成功，即达到以下几个主要目标：

- A. 付出的开发成本较低；
- B. 达到了要求的软件功能；
- C. 取得了较好的软件性能；
- D. 开发的软件易于移植；
- E. 需要较低的维护费用；
- F. 能按时完成开发任务；

G. 及时交付产品；

H. 开发的软件可靠性高。

②软件工程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(需求、设计、实现、确认以及维护等活动)。

③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：

A. 选取适宜的开发范型。必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。

B. 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、抽象与信息隐蔽、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。

C. 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。

D. 重视开发过程的管理。管理是生产的保障，因此，只有当软件过程得以有效管理时，才能实现有效的软件工程。

软件工程技术两个明显特点就是规范化和文档化。美国工业界和政府部门认识到，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都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4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。

1.1.4 软件工程的基本原理

著名的软件工程专家 B. W. Boehm 于 1983 年综合了软件工程专家、学者们的意见，并总结了开发软件的经验，提出了软件工程的 7 条基本原理。这 7 条原理被认为是确保软件产品质量和开发效率的最小集合，又是相互独立、缺一不可、相当完备的最小集合。下面简单介绍软件工程的这 7 条基本原理。

①用分阶段的生存周期计划严格管理。统计发现开发不成功的软件，大部分是由于计划不周密造成的，严格按照计划对软件开发与维护进行管理是从失败的经验教训中总结出来的，根据这条基本原理可以把软件生存周期划

分成若干个阶段，并相应地制订出切实可行的计划。Boehm认为，需要制订的计划有项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划和运行维护计划6类。各级管理人员必须严格按照计划对软件开发和维护工作进行管理。

②坚持进行阶段评审。软件的质量保障工作不能等软件完成后再进行。因为，在编码阶段之前的错误约占63%，而编码错误仅占37%，另外，错误的发现及改正得越晚，所付出的代价越高。坚持在每个阶段结束前进行严格的评审，就可以尽早发现错误。这是一条必须坚持的重要原理。

③实行严格的产品控制。在软件的开发过程中不应随意改变需求，因为改变一项需求都会付出较高的代价。但是，由于外部环境的变化，在软件开发的过程中改变需求是难免的，但绝不能随意改变需求，只能依靠科学的产品控制技术来满足用户提出的改变需求的要求。为了保持软件各个配置成分的一致性，必须实行严格的产品控制，其中主要是实行基准配置管理（又称为变动控制），即凡是修改软件的建议，尤其是涉及基本配置的修改建议，都必须按规定进行严格的评审，评审通过后才能实施。这里的“基准配置”是指经过阶段评审后的软件配置成分，即各阶段产生的文档或程序代码等。

④采用现代程序设计技术。从提出软件工程的观念开始，人们一直致力于研究新的“程序设计技术”如20世纪60年代末提出的结构化程序设计技术，以及后来提出的面向对象的分析（OOA）和面向对象的设计（OOD）技术等。实践表明，采用先进的程序设计技术既可以提高软件开发与维护的效率，又可以提高软件的质量。

⑤结果应能清楚地审查。软件不同于一般的物理产品，它是看不见、摸不着的逻辑产品。因此，软件开发小组的工作进展情况可见性差，难以准确度量、评价和管理。为了更好地进行评价和管理，应根据软件开发的总目标和完成期限，尽量明确地规定出软件开发小组的责任和产品标准，从而能清楚地审查所得到的结果。

⑥开发小组的人员应少而精。软件开发小组人员的素质和数量是影响软件质量和开发效率的重要因素。实践表明，素质高的人员与素质低的人员相比，其软件开发的效率可能高出几倍至几十倍，而且所开发的软件中的错误也要少得多。所以，开发小组的人数不宜过多，因为随着人数的增加，人员之间交流情况、讨论问题的通信开销将急剧增加，这不但不能提高生产效率，反而由于误解等原因可能会增加出错的概率。因此，组成少而精的开发小组是软件工程的一条基本原理。

⑦承认不断改进软件工程实践的必要性。遵循上述6条基本原理,就能较好地实现软件的工程化生产。但是,软件工程不能停留在已有的技术水平上,应积极主动地采纳或创造新的软件技术,要注意不断总结经验,收集工作量、进度和成本等数据,并进行出错类型和问题报告的统计。这些数据既可用于评估新的软件技术的效果,又可用于指明应优先进行研究的软件工具和技术。

1.1.5 软件工程涉及的人员

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。必须在构建有助于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,我们把讨论的重点转向软件开发所涉及的人员。

通常情况下,参与项目的人员分为3类:客户、用户及开发人员。

客户(customer)是指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)是指为客户构建软件系统的公司、组织或个人,其中包括协调并指导程序员和测试人员的管理人员。

用户(user)是指实际使用系统的人,包括坐在终端前的人、提交数据的人或阅读输出的人。

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,但多数情况下,他们还是各不相同的。

客户控制着资金,常常进行合同谈判并在验收文件上签字。但是,有时候客户并不是用户。例如,假定 Wittenberg Water Works 与 Gentle Systems, Inc. 签订了一份合同,要为公司建立计算机化的会计系统。Wittenberg 的总裁可以向 Gentle Systems, Inc. 的代表明确地描述他需要什么,并且他将要签订这份合同。但是,总裁并不直接使用这个会计系统,用户将是簿记员或会计职员。因此,开发人员确切地理解客户和用户想要什么以及需要什么是非常重要的。

另外,假定 Wittenberg Water Works 是一家大型公司,它有自己的计算机系统开发部门。该部门可能决定它需要一个自动化的工具来记录其项目的成本和进度,并且决定自己构建这个工具。此时,该部门既是用户、客户,同时也是开发人员。

近几年,客户、用户和开发人员之间的这种简单的区别变得越来越复杂。客户和用户以各种方式介入到开发过程中。客户可能决定购买商业现货

(Commercial Off - The - Shelf, COTS) 软件, 并集成到开发人员将要提供和支持的最终产品中。当出现这种情况时, 客户会介入到系统体系结构的决策中, 并且对开发过程有很多约束。类似地, 开发人员可能决定使用额外的开发人员, 后者称为分包商 (subcontractor), 分包商构建子系统并把它交付给开发人员, 这些子系统包含在最终产品中。分包商可以与主开发人员在同一地点工作, 他们也可能在不同的地点与主开发人员协调地工作, 并在开发过程后期交付子系统。子系统可能是个交钥匙系统 (turnkey system), 其代码是一个合并的整体 (不需要集成额外的代码), 或者它可能需要一个单独的集成过程将主系统和子系统连接起来。

1.1.6 软件工程开发文档格式

软件文档 (document) 也称文件, 通常指一些记录的数据和数据媒体, 具有固定不变的形式, 可被人和计算机阅读。软件工程文档在整个软件生命周期的各个阶段起到了重要的桥梁作用, 它和计算机程序共同构成了能完成特定功能的计算机软件 (有人把源程序也当做文档的一部分)。从形式上看软件工程文档可分为两类: 一类是开发过程中填写的各种图表, 可称之为工作表格; 另一类是应编制的技术资料或技术管理资料, 可称之为文档或文件。

文档是软件的重要组成部分, 没有文档的软件就不成其为软件。软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量。高质量、高效率地开发、分发、管理和维护文档, 对于转让、变更、修正、扩充和使用文档, 对于充分发挥软件产品的效益有着重要的意义。从某种意义上说, 文档是软件开发规范的体现和指南。按照规范要求生成一整套文档的过程, 就是按照软件开发规范完成一个软件开发的过程。所以, 在使用工程化的原理和方法来指导软件的开发和维护时, 应当充分注意软件文档的编制和管理。好的软件文档在产品的开发生产过程中能起到如下作用:

①提高软件开发过程的能见度。把开发过程中发生的事件以某种可阅读的形式记录在文档中。管理人员可把这些记载下来的材料作为检查软件开发进度和开发质量的依据, 实现对软件开发的工程管理。

②提高开发效率。软件文档的编制使得开发人员对各个阶段的工作都进行周密思考、全盘权衡, 减少返工, 并且可在开发早期发现错误和不一致性, 便于及时加以纠正。

③作为开发人员在一定阶段的工作成果和结束标志。