

高职高专系列教材

软件开发技术

任务式教程

宋贤钧 周立民 主编

中国石化出版社

[HTTP://WWW.SINOPEC-PRESS.COM](http://www.sinopec-press.com)

高职高专系列教材

软件开发技术任务式教程

宋贤钧 周立民 主编

本教材主要有以下特点:

(1) 知识体系完整、层次清晰、内容先进、实用。基于软件项目开发的工作

流程组织内容、设置案例。

(2) 项目形式新颖、任务驱动、以职业能力培养为主线。大项目贯穿全书，小案例并行，关键技能培养。

(3) 教材内容、页岗位对接。从软件需求、页面岗位对接。从软件开发能力实践出发，掌握软件开发的方法和技术，突出编码能力。

本教材共分五章，数据表示与存储、数据组织与处理、算法设计、页面岗位、页面岗位对接。从软件开发各个方面系统地讲解了软件开发过程所涉及的方法和技术，突出编码能力。

本书可作为高职院校、中职学校、成人高校计算机专业及相关专业的教材，也可供从事软件开发工作的工程技术人员参考。

本书可作为软件、数据表示与存储、数据组织与处理、算法设计、页面岗位、页面岗位对接。从软件开发各个方面系统地讲解了软件开发过程所涉及的方法和技术，突出编码能力。

本书可作为软件、数据表示与存储、数据组织与处理、算法设计、页面岗位、页面岗位对接。从软件开发各个方面系统地讲解了软件开发过程所涉及的方法和技术，突出编码能力。

本书由宋贤钧、周立民、张文川、陆建强、任黎明、杜吉梁等老师编写。宋贤钧、周立民、张文川、陆建强、任黎明、杜吉梁等老师对本教材的编写提出了很多建议，杜吉梁老师对本教材的出版给予了很大的帮助，在此一并表示衷心的感谢。

中国石化出版社

内 容 提 要

软件开发技术是电子信息类专业的一门核心课程。本教材从软件开发方法与环境、数据表示与存储、数据组织与处理、算法设计与应用、软件测试与维护技术五个方面系统地讲解了软件开发过程所涉及的基本方法和技能。教材基于软件产品开发的工作过程甄选内容,从典型工作任务出发,与程序员岗位对接,精心设置案例,具有知识体系完整,层次清晰,内容先进、实用,组织形式新颖独特、便于学习等特点。

教材采用任务驱动教学法,深入浅出,注重理论够用,突出实践操作,在基于工作过程的学习中培养学生的程序设计能力、算法构建能力和应用软件实际工程项目开发的能力,主要面向高职高专、成人高校等计算机类专业的学生。

图书在版编目(CIP)数据

软件开发技术任务式教程/宋贤钧,周立民主编.
—北京:中国石化出版社,2013.5
高职高专系列教材
ISBN 978-7-5114-2106-7

I. ①软… II. ①宋… ②周… III. ①软件开发-高等职业教育-教材 IV. ①TP311.52

中国版本图书馆 CIP 数据核字(2013)第 081717 号

未经本社书面授权,本书任何部分不得被复制、抄袭,或者以任何形式或任何方式传播。版权所有,侵权必究。

中国石化出版社出版发行

地址:北京市东城区安定门外大街 58 号

邮编:100011 电话:(010)84271850

读者服务部电话:(010)84289974

<http://www.sinopec-press.com>

E-mail:press@sinopec.com

北京柏力行彩印有限公司印刷

全国各地新华书店经销

*

787×1092 毫米 16 开本 15.5 印张 374 千字

2013 年 7 月第 1 版 2013 年 7 月第 1 次印刷

定价:38.00 元

前言

近年来,随着软件外包业务的飞速发展,处于核心地位的软件产业在国民经济增长中发挥着越来越重要的作用,而如今企业用人的结构逐渐发生变化,要求不断提高,市场需求主要以高素质、实用型、技能型软件专业技术人才为主,即我们通常说的“软件蓝领”。

软件开发技术是电子信息类专业的一门核心课程。本教材采用任务驱动法,深入浅出,注重理论够用,突出实践操作,在基于工作过程的学习中培养学生程序设计能力、算法构建能力和应用软件实际工程项目开发的能力,主要面向高职高专、成人高校等计算机类专业的学生。

本教材主要有以下特点:

(1)知识体系完整、层次清晰、内容先进、实用。基于软件产品开发的工作过程甄选内容、设置案例。

(2)组织形式新颖、独特、便于学习。以职业能力培养为主线,大项目贯穿,小案例并行,突出基础知识、职业素养和关键技能培养。

(3)教材内容从典型工作任务出发,与程序员岗位对接。从软件开发能力实际出发,紧跟软件开发技术的前沿,又兼顾传统的方法和技术,突出编码能力。

本教材共分5部分,从软件开发方法与环境、数据表示与存储、数据组织与处理、算法设计与应用、软件测试与维护技术五个方面系统地讲解了软件开发过程所涉及的基本方法和技能。

本书作为省级精品课程配套教材,相关教学资源(课程标准、PPT教学课件、电子教案、实训案例及书中案例源代码等)可登陆课程网站下载或填写书后所附的《教学资源索取单》依照相关方式索取。

网址: <http://jpkc.lzpcc.edu.cn/12/sxj/default.asp>。

本教材由宋贤钧、周立民主编,张丽景参加编写。张文川负责资料的收集和校对工作。任泰明、文晖、赵睿、彭涓、韩艳、杜吉梁老师对本教材的编写提出了很多建议,杜韦辰老师对本教材的出版给予了很大的帮助,在此一并表

示感谢。

本教材编者都是多年从事教学和软件开发的一线教师，限于作者水平，书中难免会有错误或疏漏之处，真诚地欢迎各位专家和读者批评指正，以帮助我们进一步完善。

编者

“双基”教学是教学工作的基础，也是教学工作的重点。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。

在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。

在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。

在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。

在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。

在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。

在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。

在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。在“双基”教学中，教师应注重基础知识的传授，同时也要注意培养学生的基本能力。

目 录

第1部分 软件开发方法与环境

任务 1.1 提炼 SAGM(教职工津贴发放管理)系统需求	(2)
1.1.1 案例描述	(2)
1.1.2 案例分析	(2)
1.1.3 知识准备	(3)
1.1.4 案例实现	(9)
1.1.5 技能训练	(9)
任务 1.2 确定 SAGM 系统开发模型	(10)
1.2.1 案例描述	(10)
1.2.2 案例分析	(11)
1.2.3 知识准备	(11)
1.2.4 案例实现	(22)
1.2.5 技能训练	(22)
任务 1.3 分析 SAGM 系统开发方法	(23)
1.3.1 案例描述	(23)
1.3.2 案例分析	(23)
1.3.3 知识准备	(24)
1.3.4 案例实现	(31)
1.3.5 技能训练	(34)
任务 1.4 配置 SAGM 系统运行环境	(35)
1.4.1 案例描述	(35)
1.4.2 案例分析	(35)
1.4.3 知识准备	(36)
1.4.4 案例实现	(40)
1.4.5 技能训练	(42)

第2部分 数据表示与存储

任务 2.1 确定 SAGM 数据库的需求	(44)
2.1.1 案例描述	(44)

2.1.2	案例分析	(44)
2.1.3	知识准备	(44)
2.1.4	案例实现	(48)
2.1.5	技能训练	(48)
任务 2.2	建立 SAGM 数据模型	(49)
2.2.1	案例描述	(49)
2.2.2	案例分析	(49)
2.2.3	知识准备	(50)
2.2.4	案例实现	(57)
2.2.5	技能训练	(59)
任务 2.3	创建 SAGM 数据库	(59)
2.3.1	案例描述	(59)
2.3.2	案例分析	(60)
2.3.3	知识准备	(61)
2.3.4	案例实现	(66)
2.3.5	技能训练	(70)
任务 2.4	学习结构化查询语言 (SQL)	(70)
2.4.1	案例描述	(70)
2.4.2	案例分析	(70)
2.4.3	知识准备	(71)
2.4.4	案例实现	(78)
2.4.5	技能训练	(79)

第3部分 数据组织与处理

任务 3.1	顺序表的逆置问题	(81)
3.1.1	案例描述	(81)
3.1.2	案例分析	(81)
3.1.3	知识准备	(81)
3.1.4	案例实现	(95)
3.1.5	技能训练	(98)
任务 3.2	中缀表达式转换为后缀表达式	(99)
3.2.1	案例描述	(99)
3.2.2	案例分析	(100)
3.2.3	知识准备	(100)
3.2.4	案例实现	(106)
3.2.5	技能训练	(108)
任务 3.3	分油问题	(109)
3.3.1	案例描述	(109)

3.3.2	案例分析	(109)
3.3.3	知识准备	(109)
3.3.4	案例实现	(116)
3.3.5	技能训练	(119)
任务 3.4	文本编辑助手	(119)
3.4.1	案例描述	(119)
3.4.2	案例分析	(119)
3.4.3	知识准备	(120)
3.4.4	案例实现	(126)
3.4.5	技能训练	(128)
任务 3.5	稀疏矩阵的转置	(128)
3.5.1	案例描述	(128)
3.5.2	案例分析	(129)
3.5.3	知识准备	(129)
3.5.4	案例实现	(133)
3.5.5	技能训练	(134)
任务 3.6	族谱问题	(135)
3.6.1	案例描述	(135)
3.6.2	案例分析	(135)
3.6.3	知识准备	(135)
3.6.4	案例实现	(142)
3.6.5	技能训练	(144)
任务 3.7	哈夫曼编码问题	(144)
3.7.1	案例描述	(144)
3.7.2	案例分析	(144)
3.7.3	知识准备	(145)
3.7.4	案例实现	(150)
3.7.5	技能训练	(151)
任务 3.8	最小代价通信网问题	(151)
3.8.1	案例描述	(151)
3.8.2	案例分析	(151)
3.8.3	知识准备	(151)
3.8.4	案例实现	(164)
3.8.5	技能训练	(165)
任务 3.9	最短路径问题	(166)
3.9.1	案例描述	(166)
3.9.2	案例分析	(166)
3.9.3	知识准备	(166)
3.9.4	案例实现	(171)

3.9.5 技能训练	(171)
------------------	-------

第4部分 算法设计与应用

任务4.1 猴子吃桃问题	(174)
4.1.1 案例描述	(174)
4.1.2 案例分析	(174)
4.1.3 知识准备	(174)
4.1.4 案例实现	(177)
4.1.5 技能训练	(178)
任务4.2 SAGM 系统津贴数据排序	(179)
4.2.1 案例描述	(179)
4.2.2 案例分析	(179)
4.2.3 知识准备	(179)
4.2.4 案例实现	(187)
4.2.5 技能训练	(188)
任务4.3 卖鸡蛋问题	(188)
4.3.1 案例描述	(188)
4.3.2 案例分析	(189)
4.3.3 知识准备	(189)
4.3.4 案例实现	(191)
4.3.5 技能训练	(192)
任务4.4 新娘和新郎	(192)
4.4.1 案例描述	(192)
4.4.2 案例分析	(193)
4.4.3 知识准备	(193)
4.4.4 案例实现	(198)
4.4.5 技能训练	(199)
任务4.5 马跳棋盘	(199)
4.5.1 案例描述	(199)
4.5.2 案例分析	(199)
4.5.3 知识准备	(200)
4.5.4 案例实现	(202)
4.5.5 技能训练	(206)

第5部分 软件测试与维护

任务5.1 制定 SAGM 的系统测试方案	(209)
5.1.1 案例描述	(209)

5.1.2 案例分析	(209)
5.1.3 知识准备	(210)
5.1.4 案例实现	(215)
5.1.5 技能训练	(220)
任务 5.2 进行部门管理的功能测试	(220)
5.2.1 案例描述	(220)
5.2.2 案例分析	(220)
5.2.3 知识准备	(221)
5.2.4 案例实现	(225)
5.2.5 技能训练	(227)
任务 5.3 完成一个程序段的检查	(227)
5.3.1 案例描述	(227)
5.3.2 案例分析	(227)
5.3.3 知识准备	(227)
5.3.4 案例实现	(231)
5.3.5 技能训练	(232)
任务 5.4 处理 SAGM 系统报表导出异常	(232)
5.4.1 案例描述	(232)
5.4.2 案例分析	(233)
5.4.3 知识准备	(233)
5.4.4 案例实现	(236)
5.4.5 技能训练	(236)
参考文献	(237)

第1部分 软件开发方法与环境

❖ 知识目标:

- * 了解软件开发最基本的思想,熟悉软件工程理论。
- * 知道软件开发生命周期,熟悉软件开发模型。
- * 懂得结构化软件开发原理。
- * 了解 web 体系结构。

❖ 技能目标:

- * 学会提炼系统需求的方法。
- * 能够正确选择系统开发模型。
- * 掌握数据流图和系统功能图的绘制。
- * 掌握 Web 系统环境的配置和部署方法。

资源宝典

有效沟通是一种能力

养过猫和狗的人都知道,“猫、狗是仇家,见面就掐”。其实,猫和狗之所以“见面就掐”,主要是因为沟通上出现了问题。比如:“摇尾摆臀”在狗这里表示友好的意思,在猫那里却成了挑衅;同样,“呼噜呼噜”的鼾声在猫这里是放松、友好和休闲的意思,对于狗看来是“示威”,当然要“掐”了。

在职场中,员工与员工之间因缺乏沟通而导致的冲突和矛盾也并不少见。很多人喜欢独来独往,不喜欢与人交流,更不喜欢与人沟通,结果在执行任务时经常走弯路,有时还会步入歧路,不能完成任务。这些人总是疲于应付工作过程中出现的问题,他们埋怨问题太多、时间太少,总是被问题压得喘不过气来。可是他们却很少思考造成这种状况的原因是什么。在很大程度上,这是他们没有与人进行有效沟通的结果。所以有效沟通是更迅速、更有效解决问题的方法。

任务 1.1 提炼 SAGM(教职工津贴发放管理)系统需求

1.1.1 案例描述

某高校信息化建设步伐加快,为了实现教职工的津贴发放管理与银行业务对接,决定开发一套教职工津贴发放管理系统(Staff Allowances Grant Management,简称 SAGM 系统),下面是一段对话。

财务长:现在我们需要一个教职工的津贴发放管理系统。

业务员:请问您的系统主要是哪些人使用呢?

财务长:系统使用人员主要有财务管理员、学院各二级部门管理员和全院教职工。

业务员:您能简单说一下这些人员在津贴发放事务中的主要工作吗?

财务长:财务管理员设置津贴类型,比如:各类补助、各类津贴、各类奖金等,并按期下发各二级部门津贴总额,将审核通过的津贴分配结果导出为银行报表,直接由银行发放到职工工资卡中。学院各二级部门管理员定期依据财务处下发的津贴总额进行本部门的津贴分配,落实到个人,生成报表,提交财务审核,如果审核通过则打印出来供教职工签字确认;如果审核没有通过,根据反馈信息,重新修改后再次提交,直至审核通过。教职工就是查看自己的津贴信息和个人信息。

业务员:听了您的介绍,我对教职工津贴发放的工作流程有了初步的了解。我简单分析了一下,系统的主要数据包括学院各二级部门信息、教职工信息和各类津贴信息,我会把我的理解绘制一张流程图出来,咱们再深入沟通。你们还有其他的要求吗?比如希望系统在什么环境下运行?

财务长:就在 Windows 操作系统下使用,比较方便嘛!一定要保证系统数据的安全性,以防发生意外情况时造成数据丢失、泄漏等事故,还要有管理权限设置……

1.1.2 案例分析

从案例的描述中,我们得知用户需要开发一个系统,那么这个系统是软件产品吗?软件是由哪几部分组成的呢?案例描述是一段典型的软件开发需求分析的访谈层次对话,彼此在沟通业务需求、进行关键功能的描述和确认。在软件开发领域,人们越来越多地提到需求,这里的需求源自用户的“需要”,也就是准确说明开发什么。为什么我们如此多的关注需求?原因恰恰在于我们无法有效地获取需求,无法准确地表述需求。而需求的变化对于整个项目的开发成本、开发周期影响极大,可谓“一石激起千层浪”。所以,想让项目获得成功,首先要做好需求分析。

软件需求分析是指理解用户需要,使软件功能与客户要求达成一致,评估软件开发风险

和成本代价,最终形成软件开发计划的一个复杂过程。在这个过程中,用户处在主导地位,需求分析工程师和项目经理要负责整理用户需求,为之后的软件设计打下基础。

案例中业务员和财务长进行有效沟通,就是分析软件用户类型有多少,每种用户的需求是什么,系统功能有哪些,软件运行环境是什么等等。如果忽略前期需求分析或需求分析做的不充分,很可能会导致投入大量的人力、物力、财力和时间,开发出的软件产品不符合用户需求,那所有的投入都是徒劳。比如,客户希望 SAGM 系统运行在 Windows 环境下,易用且部署方便,如果业务员在调研时忽略了软件的运行环境,忘了向用户询问这个问题,而想当然地认为是开发 For Linux 的软件,以确保数据相对安全。当开发人员千辛万苦完成系统向用户提交时才发现出了问题,那时候真是欲哭无泪了。

因此,软件需求分析在软件开发过程中具有举足轻重的地位,具有决策性、方向性和指导性的作用。所以,一定要对需求分析引起足够的重视,尤其是在大型软件系统的开发中,它的作用要远远大于程序设计。

1.1.3 知识准备

1.1.3.1 认识软件

1) 软件的概念

计算机系统由硬件和软件两大部分组成。计算机的发展到目前为止按总体发展阶段来划分,可分为三个时期。自 1946 年到 20 世纪 60 年代中期是计算机发展的早期,称为电子计算机时代,系统以硬件为主,软件费用是总费用的 20% 左右,主要用于商业、大学教学和政府机关。从 20 世纪 60 年代中期到 80 年代初期是计算机发展中期,称为微型计算机时代,软件费用迅速上升到总费用的 60%,软件不再只是技巧性和高度专业化的神秘机器代码,以 Microsoft 公司为代表的各种应用软件相继出台,使用计算机的人也多了起来。计算机工业迅速发展,其应用范围渗透于工业控制、自动化仪器与家电、OA、商业、机电与宇航等各个领域。1985 年以直到今天,称为网络与多媒体时代,软件费用已上升到 80% 以上,人们对计算机的需求更加广泛。计算机软件业迅速发展,各种可视化软件开发工具和多媒体技术实现算法(多种图像格式与压缩算法)应运而生,是计算机网络飞速发展的阶段。其主要特征是:计算机网络化,协同计算能力发展以及全球互连网络(Internet)的盛行。计算机的发展已经完全与网络融为一体,体现了“网络就是计算机”的口号。

可以看出,随着计算机技术(硬件技术和软件技术)的发展,软件相对硬件的费用比例在不断提高,软件在计算机系统的比重越来越大,而且这种趋势还在增加。人们感到传统的软件生产方式(个体、互助合作的手工方式^①)已不再适应发展的需要,因此软件生产必须按照工程学的基本原理和方法进行生产,其生产过程即为软件项目开发过程。软件项目开发过程将软件生产分成几个阶段,每个阶段都有严格管理和质量检验,在研制软件设计和生产

^① 个体手工方式:以硬件生产为主,软件处于从属地位。设计过程是在一个人的头脑中完成的,程序的质量完全取决于个人的编程技巧。软件定义要点是程序或机器指令程序;互助合作的手工方式:软硬件均考虑。由多人分工合作完成编程任务,程序的质量无法得到保证。软件定义要点是程序+说明书。

方法中,以书面文件作为共同遵循的依据,这些文档属于软件含义中的内容。

【软件的定义】软件是计算机系统中与硬件相互依存的部分,它包括计算机程序、相关的文档资料以及在计算机上运行程序时所必需的数据。

计算机系统是通过运行程序来实现各种不同的应用。程序是指以算法语言为基础而编制的,能够使计算机做出信息处理并产生一定结果的指令或指令组合。程序包括实现各种不同功能的程序、用户为自己的特定目的编写的程序、检查和诊断机器系统的程序、支持用户应用程序运行的系统程序、管理和控制机器系统资源的程序等。

数据是使程序能够适当地处理信息的数据结构。

文档指用自然语言或者形式化语言所编写的文字资料和图表,用来描述程序的内容、组成、设计、功能规格、开发情况、测试结果及使用方法,如程序设计说明书、流程图、用户手册等。

总之,软件主要包括以下内容:

在运行中能提供所希望的功能和性能的指令集(即程序);

使程序能够正确运行的数据结构;

描述程序研制过程、方法所用的文档。

2) 软件的特点

计算机系统软件与硬件是相互依存的,缺一不可。而软件与其他产品的特点不同。它是一种特殊的产品,具有下列特殊性质:

(1)无形性。软件是一种逻辑产品,它与物质产品有很大区别,它是脑力劳动的结晶。软件产品看不见摸不着,因而具有无形性。它以程序和文档的形式出现,保存在存储介质上,通过计算机运行才能体现它的功能和作用。

(2)研发性。软件产品的生产主要是脑力劳动,还没有完全摆脱手工开发方式,是人的智力的高度发挥,不同于传统意义上的硬件制造,它没有明显的制造过程,其生产主要是研制,对软件的质量控制必须立足于软件开发过程。

(3)维护复杂。软件产品不会用坏,不存在磨损、消耗的问题。但却存在需要更新的问题。因为在软件的生存期中,它一直处于改变(维护)状态,随着针对某些缺陷的维护,可能带来一些新的缺陷,使软件故障率增加,而软件不像硬件一样有备件,它的维护要比硬件复杂得多。

(4)成本高。由于软件产品的生产主要是研制,其成本主要体现在软件的开发和研制上,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。另外,软件是无形的,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、生产和管理带来许多困难,从而使生产成本非常高。

3) 软件的分类

(1)按软件的功能。

系统软件:为了方便地使用机器及其输入输出设备,充分发挥计算机系统的效率,围绕计算机系统本身开发的程序系统叫做系统软件。系统软件是计算机系统的重要组成部分,它

支持应用软件的开发和运行。系统软件包括操作系统(常用的有 DOS^①、Windows^②、Unix^③、NetWare^④、Windows NT^⑤等)、各种语言编译程序(机器语言、汇编语言等)、数据库管理系统(Access、SQL Server、Oracle等)、设备驱动程序等。

支撑软件：协助用户开发应用软件的工具性软件，包括帮助程序员开发软件产品的工具、也包括管理人员控制开发进程的工具。如 IBM 公司的 Web Sphere，微软公司的 Studio. NET 等。

应用软件：是专门为了某种使用目的而编写的程序系统。常用的有文字处理软件，如 WPS 和 Word；专用的财务软件、人事管理软件；计算机辅助软件，如 AutoCAD；绘图软件，如 3DS 等。

(2)按软件工作方式。

实时处理软件：在时间和数据产生时对其立即处理，并及时反馈信号以控制需要检测的过程的软件。实时处理软件主要包括数据采集、分析、输出三个部分。

分时软件：允许多个联机用户同时使用计算机的软件。

交互式软件：实现人-机通信的软件。

批处理软件：把一组输入作业或一批数据以成批处理的方式一次运行，按顺序逐个处理的软件。

(3)按软件规模。按开发软件所需人力、时间以及完成的原程序的行数，可分为 6 种不同规模的软件，如表 1-1 所示。

表 1-1 软件规模的参考分类

类别	参加人员数/人	开发期限	程序长度/行
微型	1	1~4 周	500
小型	1	1~6 月	$(1\sim2)\times10^3$
中型	2~5	1~2 年	$(5\sim50)\times10^3$
大型	5~20	2~3 年	$(50\sim100)\times10^3$
超大型	100~1000	4~5 年	1×10^6
极大型	2000~5000	5~10 年	$(1\sim10)\times10^6$

(4)按软件服务对象的范围。

项目软件：也称定制软件，是为某个特定客户或少数客户开发的软件，如军用防空指挥系统、卫星控制系统等。

产品软件：直接提供给市场，或是为众多用户服务的软件，如文字处理软件、财务处理软件、人事管理软件等。

1.1.3.2 软件需求分析

1) 需求分析的任务

【需求分析的任务】就是解决“做什么”的问题，就是要全面地理解用户的各项要求，并

① DOS：基于字符界面的单用户任务的操作系统。
② Windows：基于图形界面的单用户多任务操作系统。
③ 一个通用的交互式分时操作系统。
④ NetWare：基于文件服务和目录服务的网络操作系统。
⑤ Windows NT：基于图形界面 32 位多任务、对等的网络操作系统。

准确地表达所接受的用户需求。但一般情况下,用户并不能描述清楚自己的需要,这时就需要系统分析人员根据用户的语言描述整理出相关的需求信息再进一步和客户核对。

开发软件系统最为困难的概念性工作便是编写出详细技术需求,这包括所有面向用户、面向机器和其他软件系统的接口。同时这也是一旦做错,将最终会给系统带来极大损害的部分,并且以后再对它进行修改也极为困难。

对于开发人员来说,如果没有得到客户认可的需求文档,我们如何知道项目究竟要干什么?如果我们不知道什么对客户来说是重要的,那软件产品又如何能令客户满意呢?

因此,即便是非商业目的的软件需求也是必需的。在平时的学习训练中,小到做的一次习题、编写的一个小程序,都需要把问题描述搞清楚、分析透彻。当然你可能偶尔不需要文档说明就能与其他人意见较为一致,但更常见的是出现重复返工这种不可避免的后果,而重新编制代码的代价远远超过重写一份需求文档的代价,这些惨痛的教训已在国内的软件开发者身上发生过不止一次。

2) 需求分析的过程

需求分析不像案件侦破那样需从细节线索着手,而是先了解宏观的功能和问题,再进行细节的确认。一个应用软件产品(记为S)的功能范围可能很广,可以按不同的解空间(记为M)分类,每个解空间对应一个软件子系统。

$$S = \{M_1, M_2, M_3, \dots, M_n\}$$

解空间 M_i 由若干个子问题(记为P)组成,每个问题对应于系统中的一个组件。

$$M_i = \{P_1, P_2, P_3, \dots, P_n\}$$

问题 P_j 有若干个功能(记为F),每个功能对应于组件中的实现接口。

$$P_j = \{F_1, F_2, F_3, \dots, F_n\}$$

也就是说,需求分析结果性文档(需求规格说明)既符合那些想了解系统宏观功能的领导要求,又能对详细设计系统功能的开发人员、技术员适用。需求分析过程大体分四部分,即问题获取、提炼分析、编制报告、确认验证,如图1-1所示。

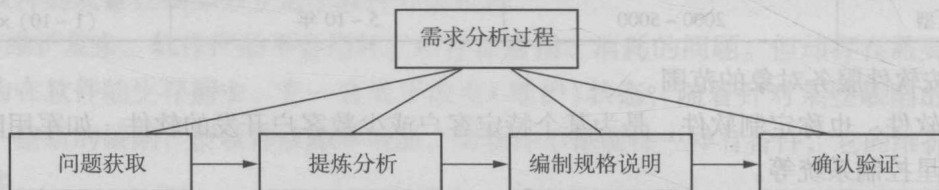


图 1-1 需求分析过程

需求分析的准确性决定了软件项目的复杂性和正确性,因此需求分析的高风险性和重要性不言而喻。实践过程中,造成需求分析工程实施困难主要是以下3个方面:

(1) 用户表述不清。大部分用户对需求只有模糊的感觉,说不清楚具体的要求。例如现阶段很多机构、单位在进行设备基础设施改造和网络升级时,用户方的办公人员大多不清楚计算机网络有什么用,更缺乏IT系统建设方面的专业知识。此时,用户就会依赖开发人员替他们设想需求。工程需求存在一定的主观性,为项目未来建设埋下了潜在的风险。

(2) 需求不断变化。根据经验,随着用户方对信息化建设的认识和自己业务水平的提

高，他们会在不同的阶段和时期对项目提出新的要求或需求变更。事实上，作为项目的实施者，我们必须接受“需求会不断变化”这个事实。因此，在需求分析阶段，我们要尽可能地分析清楚哪些是稳定的需求，哪些是易变的需求，以便在进行系统设计时，将软件的核心建筑在稳定的需求上，同时留出变更空间。

(3)理解失误。用户表达的需求，不同的分析人员可能有不同的理解。如果分析人员理解错了，可能会导致以后的开发工作劳而无功，所以分析人员写好需求说明书后，要请用户方的各个代表验证。如果问题很复杂，双方都不太明白，就有必要请开发人员快速构造软件的原型，双方再次论证需求描述是否正确。由于客户大多不懂软件，他们可能觉得软件是万能的，会提出一些无法实现的需求，有时客户还会曲解软件系统分析人员的建议或答复。

有一个软件人员滔滔不绝地向客户讲解在“信息高速公路上做广告”的种种好处，客户听得津津有味。最后，心动的客户对软件人员说：“好得很，就让我们马上行动起来吧。请您决定广告牌的尺寸和放在哪条高速公路上，我立即派人去做。”

要克服这些困难，需求分析通常包括这样一些活动：

(1)了解用户的相关业务。系统分析员必须首先到用户的工作环境(或者待开发软件的预期使用环境)中去，通过用户的演示、讲解和有关文档了解相关业务，还要和用户进行交流、协商。这是一个关键的活动，如果一个分析员还不能真正地理解用户的相关业务，那么他就无法知道用户到底想要软件帮他们做些什么。用户自然非常清楚他们自己的业务，但是他们往往不能正确、完整地表述他们对于软件的要求。不过如果某个软件实现的并不是他们想要的情况，他们会立刻发现。因此在了解用户的相关业务时，分析员还需要制定一些提问单和表格交给用户填写，以让他们说出那些用户自己无法说出的，但对软件又是重要的信息。

(2)分析用户业务流程。分析员了解到的用户业务也许只是一些离散的业务活动，而业务流程是重要的信息，它往往指出了某些用于指导软件功能组织的信息。分析员将了解到的业务活动加以整理并按照这些活动所固有的次序形成业务流程。

(3)了解用户对于软件的期望值。软件首先需要能正确地处理用户的业务，但是用户对于软件还有一些其他的要求，如使用的便利性、附带接管一些机械的日常活动(如每天的整理业务数据等)和一些性能要求等。用户对软件的这些需求有时虽然并不是直接和业务相关，但是对于用户而言却显得很重要。分析员要尽量启发用户提出更多的附带需求，尽可能地挖掘用户对软件的期望，这样可以避免用户在使用时才发现软件不是他们所希望的，而造成软件无法交工的后果。

(4)整理用户要求。分析员需要独立地完成用户需求的整理并形成规范的需求规格说明文档。如果在整理过程中发现有疑点，就需要立刻和用户协商并解决它；如果有些问题现在根本无法处理，那就只有暂时留在那儿(但是需要通知用户知道)，一旦时机成熟就要立刻解决它。

(5)需求评审。分析员整理得到的需求分析规格说明必须要通过需求评审才能说明需求分析工作可以结束而进入后续工作。

3) 软件需求的类型

软件需求包括四个不同的层次：业务需求、用户需求、功能需求和非功能需求。