



国家级骨干高职院校中央财政支持的重点建设课程建设成果
工业和信息化人才培养规划教材

Technical And Vocational Education

高职高专计算机系列

Windows 应用程序 开发教程 (C# 版)

Windows Application
Development

董久敏 ◎ 主编

王韶霞 周阿连 崔蕾 ◎ 副主编 张洪杰 ◎ 主审

综合实训核心课程

提供Visual Studio 2008和Visual Studio 2010两个版本

企业真实案例，完整项目开发流程



人民邮电出版社
POSTS & TELECOM PRESS



CD-ROM



国家级骨干高职院校中央财政支持的重点建设课程建设成果
工业和信息化人才培养规划教材

Technical And Vocational Education

高职高专计算机系列

Windows 应用程序 开发教程 (C# 版)

Windows Application
Development

董久敏 ◎ 主编

王韶霞 周阿连 崔蕾 ◎ 副主编 张洪杰 ◎ 主审

人民邮电出版社

北京

图书在版编目(CIP)数据

Windows应用程序开发教程：C#版 / 董久敏主编

— 北京：人民邮电出版社，2013.12

工业和信息化人才培养规划教材·高职高专计算机系
列

ISBN 978-7-115-32902-8

I. ①W… II. ①董… III. ①

Windows操作系统—程序设计—高等职业教育—教材②

C语言—程序设计—高等职业教育—教材 IV. ①

TP316.7②TP312

中国版本图书馆CIP数据核字(2013)第197566号

内 容 提 要

本书是国家级骨干高职院校中央财政支持的重点建设课程“Windows 应用程序开发(C/S)综合实训”的建设成果，全书以一个完整的已经正式投入运行的超市商品管理系统为案例，以软件工程原理和真实的软件企业的项目实施方案为依据，详细深入地介绍了企业级项目开发的完整过程；在书中重点讲述了项目的需求分析、概要设计、详细设计、代码的实现、项目的测试及打包部署等。

本书可以作为高等院校计算机应用技术、软件技术和计算机网络技术等专业的教材，也可以作为用C#语言进行C/S项目开发的软件工程技术人员的参考书。

◆ 主 编 董久敏

副 主 编 王韶霞 周阿连 崔 蕾

主 审 张洪杰

责任编辑 王 威

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京昌平百善印刷厂印刷

◆ 开本：787×1092 1/16

印张：20.5

2013年12月第1版

字数：524千字

2013年12月北京第1次印刷

定价：49.80元（附光盘）

读者服务热线：(010)81055256 印装质量热线：(010)81055316

反盗版热线：(010)81055315

本书是国家级骨干院校重点课程“Windows 应用程序开发 (C/S) 综合实训”的建设成果,该课程是计算机应用技术、计算机网络技术、软件技术等专业的核心专业课程,是在由高校教授、软件企业系统架构师、项目经理等组成的专业建设理事会全面指导下,引入行业企业技术标准、项目开发规范、管理制度,对软件开发与维护工程师、软件开发工程师等职业岗位要求进行典型工作任务分析、实践专家访谈的基础上设置的专业课程。该课程与北京京胜世纪科技有限公司、东软集团股份有限公司、烟台东方电子海颐软件股份有限公司、华东电子有限公司、中正新技术公司等多家企业合作,基于工作过程项目开发,通过 IT 企业实践专家访谈与研讨,了解软件开发与维护、软件工程师等相关典型工作任务的工作对象、工具材料、工作方法、劳动组织和工作要求。将“面向对象程序设计”、“数据库开发与维护”、“软件工程”、“软件测试”等课程内容深度融合,精心设计综合项目实训,采用当今企业的主流技术和设计理念,根据软件工程的原理,最大限度地结合数据库访问技术、面向对象程序设计技术、事务管理、三层架构(表示层、业务逻辑层、数据访问层)技术等企业项目开发技术。

本书引入企业项目——超市商品管理系统,运行环境为 Windows XP /Windows 7,开发环境为 Visual Studio 2010 (C#语言),数据库为 SQL Server 2008,采用了 LINQ (语言集成查询: Language INtegrated Query) 等最新的技术,对复杂的数据操作有详细、全面的讲解。为了让读者更全面地掌握在 .Net Framework 4.0 框架中开发企业项目的能力,在企业项目中采用两种三层架构技术:传统的三层架构和 LINQ 技术三层架构。对数据库的访问主要由存储过程完成,在项目中广泛应用了 SQL Server 数据库中的事务管理及 .Net Framework 4.0 框架中的事务管理。对项目的实施过程、数据的备份和还原、数据的加密和解密、基于角色的权限管理、数据报表的导出和打印都有完整的设计和实现。

整个企业项目分成 6 个主要功能模块:1. 基于角色的访问控制 (Role-Based Access Control) 实现权限管理的系统管理模块;2. 实现基于 SQL Server 事务管理设计、.Net Framework 4.0 框架事务管理设计的采购管理模块;3. 采用 LINQ 技术、基于 LINQ 事务管理的销售管理模块;4. 实现数据备份和还原功能的数据管理模块;5. 实现数据报表打印和导出、各种信息查询功能的统计查询模块;6. 实现对基础数据管理的资料管理模块等。

全书以企业项目案例为引导,以项目开发为主线,讲解的重点内容全面翔实透彻,涉及企业项目开发用到的各种必备知识、主流技术和标准流程。通过对案例的剖析,培养学生进行面向对象软件开发必须具备的基本能力,使学生在完成项目开发的同时全面掌握软件开发所需要的技术,充分锻炼团队协作能力,满足企业对面向对象软件人才的需求。

随书光盘包含了完整的企业项目代码、项目数据库、项目中的存储过程代码。另提供配套的电子教案、程序完整代码、习题库、习题答案,请读者登录人民邮电出版社教学服务与资源

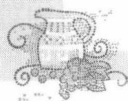
网 (www.ptpedu.com.cn) 下载使用。

本书由董久敏任主编，王韶霞、周阿连、崔蕾任副主编，张洪杰作为本书的企业顾问。
由于编者水平有限，书中难免存在错误或不妥之处，恳请广大读者批评指正。

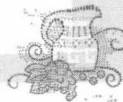
编 者
2013 年夏于烟台

目 录

第 1 章 软件项目开发初步	1	第 4 章 系统的概要设计	24
1.1 软件生存周期	1	4.1 总体设计	24
1.1.1 软件生存周期模型	1	4.1.1 概述	24
1.1.2 软件生存周期过程	2	4.1.2 基本设计	26
1.2 软件开发方法	3	4.2 数据库的设计	29
第 2 章 项目实施过程	6	4.2.1 数据库的概念性设计	29
2.1 项目实施策略	6	4.2.2 数据库的总览	31
2.1.1 项目统一规划	6	第 5 章 系统的详细设计与代码实现	35
2.1.2 模板化构建	6	5.1 项目主框架实现	35
2.1.3 模板化实施	7	5.1.1 系统架构设计	35
2.2 项目实施阶段	7	5.1.2 系统框架类关系图和系统的包结构	39
2.2.1 项目准备阶段	7	5.2 通用类和接口的设计	40
2.2.2 差异需求分析阶段	7	5.2.1 通用类和接口的设计	40
2.2.3 开发阶段	7	5.2.2 FAQ (常见问题解答)	79
2.2.4 测试阶段	9	5.3 系统主窗体的设计	79
2.2.5 现场实施阶段	11	5.3.1 登录窗体的设计	79
2.2.6 验收阶段	14	5.3.2 系统主窗体的设计	81
2.2.7 软件维护阶段	15	5.4 系统管理	90
第 3 章 系统的需求分析	16	5.4.1 权限管理设计方案	90
3.1 需求描述	16	5.4.2 主要界面设计	93
3.1.1 商品进货	16	5.4.3 系统管理模块实现的相关类、接口	94
3.1.2 采购退货	17	5.4.4 实体类层 (模型层) 的主要代码设计	96
3.1.3 商品销售	17	5.4.5 数据访问层中主要的代码设计	97
3.1.4 销售退货	18		
3.1.5 统计查询	19		
3.2 分析设计	21		
3.2.1 需求分析	21		
3.2.2 模块划分	21		



5.4.6 业务逻辑层中主要的代码设计	205
.....	106
5.4.7 控制层中 (对应窗体中) 主要的代码设计	110
5.4.8 主要的存储过程、函数和视图的设计	120
5.5 资料管理	122
5.5.1 任务需求	122
5.5.2 活动图	123
5.5.3 主要界面设计	124
5.5.4 资料管理模块实现的相关类、接口	125
5.5.5 实体类层 (模型层中) 主要的代码设计	126
5.5.6 数据访问层中主要的代码设计	129
5.5.7 业务逻辑层中主要的代码设计	139
5.5.8 控制层中 (对应窗体中) 主要的代码设计	142
5.5.9 主要的存储过程、函数和视图的设计	165
5.6 采购管理	166
5.6.1 任务要求	166
5.6.2 流程图	167
5.6.3 主要界面设计	168
5.6.4 采购管理模块实现的相关类	171
5.6.5 实体类层 (模型层中) 主要的代码设计	172
5.6.6 数据访问层中主要的代码设计	176
5.6.7 业务逻辑层中主要的代码设计	181
5.6.8 控制层中 (对应窗体中) 主要的代码设计	182
5.6.9 主要的工具类设计	204
5.6.10 主要的存储过程、函数和视图的设计	205
5.7 销售管理	209
5.7.1 任务要求	209
5.7.2 流程图	210
5.7.3 主要界面设计	211
5.7.4 销售管理模块实现的相关类 (说明: 本模块由 LINQ 实现)	213
5.7.5 实体类层 (模型层中) 主要的代码设计 (略)	214
5.7.6 数据访问层中主要的代码设计	214
5.7.7 业务逻辑层中主要的代码设计	220
5.7.8 控制层中 (对应窗体中) 主要的代码设计	221
5.7.9 主要的工具类设计	241
5.7.10 主要的存储过程、函数、触发器和视图的设计	243
5.8 统计查询	245
5.8.1 任务需求	245
5.8.2 活动图	246
5.8.3 主要界面设计	247
5.8.4 统计查询模块实现的相关类、接口	249
5.8.5 实体类层 (模型层中) 主要的代码设计	249
5.8.6 数据访问层中主要的代码设计	253
5.8.7 业务逻辑层中主要的代码设计	255
5.8.8 控制层中 (对应窗体中) 主要的代码设计	256
5.8.9 主要的存储过程、函数和视图的设计	261
5.9 数据管理	264
5.9.1 任务要求	264



5.9.2 活动图.....	265	5.10.2 主要界面设计.....	280
5.9.3 主要界面设计.....	266	5.10.3 打印管理模块实现的相关类	280
5.9.4 数据管理模块实现的相关类、 接口	267	5.10.4 打印 Print.dll 动态链接库中 主要的代码设计	281
5.9.5 数据访问层中主要的代码设计	267	第 6 章 软件测试和系统部署	308
5.9.6 业务逻辑层中主要的代码设计	269	6.1 软件测试	308
5.9.7 控制层中（对应窗体中）主要 的代码设计	270	6.1.1 测试的基本概念	308
5.9.8 主要的存储过程、函数和视图 的设计	276	6.1.2 单元测试用例的设计	310
5.10 数据导出与打印管理.....	279	6.2 系统部署	312
5.10.1 任务要求.....	279	6.2.1 任务要求	312
		6.2.2 任务解决方案	312

第1章

软件项目开发初步

1.1 软件生存周期

软件开发是根据用户要求建造出软件系统或者系统中的软件部分的过程。软件开发是一项包括需求捕捉、需求分析、设计、实现和测试的系统工程。软件一般是用某种程序设计语言来实现的，通常采用软件开发工具可以进行开发。软件分为系统软件和应用软件。软件并不只是包括可以在计算机上运行的程序，与这些程序相关的文件一般也被认为是软件的一部分。软件设计的一般过程，包括软件功能的设计、实现算法的设计、软件总体结构的设计和模块设计、编程和调试、程序联调和测试以及提交程序。在软件工程中，重要的概念是软件生存周期模型（又称软件开发模型）以及软件生存周期过程。生存周期划分的原则为：各阶段的任务彼此间尽可能相对独立，同一个阶段各项任务的性质尽可能相同，从而降低每个阶段任务的复杂性，简化不同阶段之间的联系，有利于软件开发过程的组织管理。

1.1.1 软件生存周期模型

软件生存周期模型是软件开发全部过程、活动和任务的结构框架。软件生存周期模型能清晰、直观地表达软件开发全过程，明确规定了开发工作各阶段所要完成的主要活动和任务，以作为软件项目开发工作的基础。

GB8566-1988《计算机软件开发规范》把软件生存周期定义为：一个计算机软件，从构思之日起，经过开发成功投入使用，在使用中不断增补修订，直到最后决定停止使用，并被另一个软件代替之时止，被认为是该软件的一个生存周期。国家标准 GB/T8566-2001《信息技术软件生存周期过程》将软件生存周期划分为以下 8 个阶段：

- 可行性研究与计划
- 需求分析
- 概要设计
- 详细设计
- 实现（包括单元测试）



- 组装测试 (即集成测试)
- 确认测试
- 使用和维护

对于不同的软件系统,可以采用不同的开发方法,使用不同的程序设计语言,有各种不同技能的人员参与工作,运用不同的管理方法和手段,允许采用不同的软件工具和不同的软件工程环境。

1.1.2 软件生存周期过程

GB/T8566-2001《信息技术软件生存周期过程》完善了软件生存周期的定义,即:从概念形成直到退役,并且由获取和供应软件产品及服务的各个过程组成。它提出了软件生存周期过程的思想,进一步把软件工程实践按软件工程实践的各个过程来管理和实施,发展和完善了软件工程方法。从此,软件工程环境进入了过程驱动的时代。

GB/T8566-2007《信息技术软件生存周期过程》(ISO/IEC12207)把软件生存周期中可以开展的活动分为5个基本过程、9个支持过程和7个组织过程。每一生存周期过程划分为一组活动,每一项活动进一步划分为一组任务。软件生存周期过程与角色如图1-1所示。

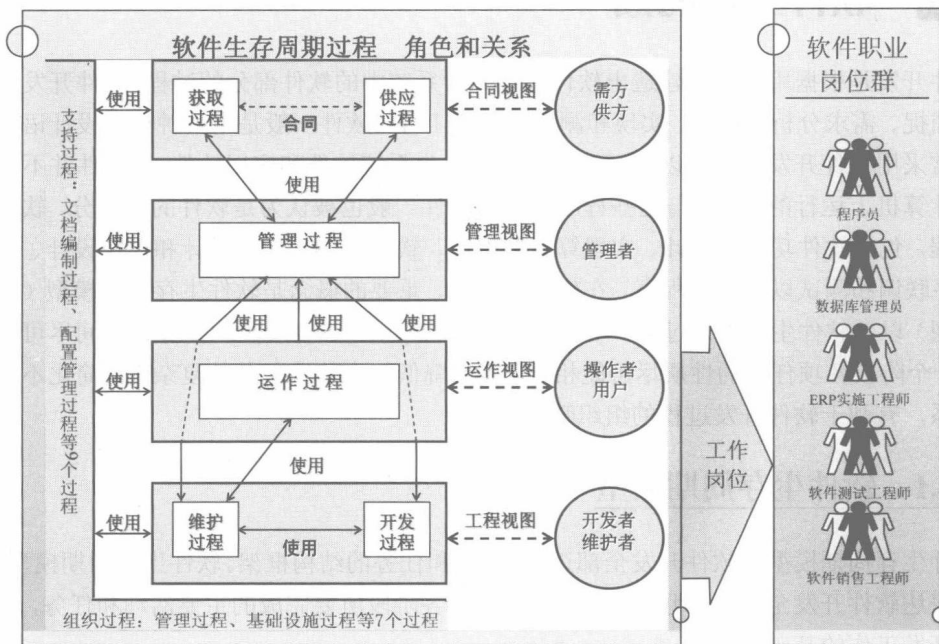


图 1-1 软件生存周期过程与角色

通过对软件生存周期过程的分析,确定软件技术等专业培养的是软件生存周期中开发过程、运作过程和维护过程的开发者、操作者和维护者。对以上软件职业岗位的开发过程、运作过程和维护过程的活动和任务进行分析,参照软件行业职业标准、软件生存周期模型、软件开发流程规范和企业软件项目管理经验,通过反复论证,确定9项典型工作任务并对其进行分析描述。软件开发、运作和维护的典型工作任务如图1-2所示。

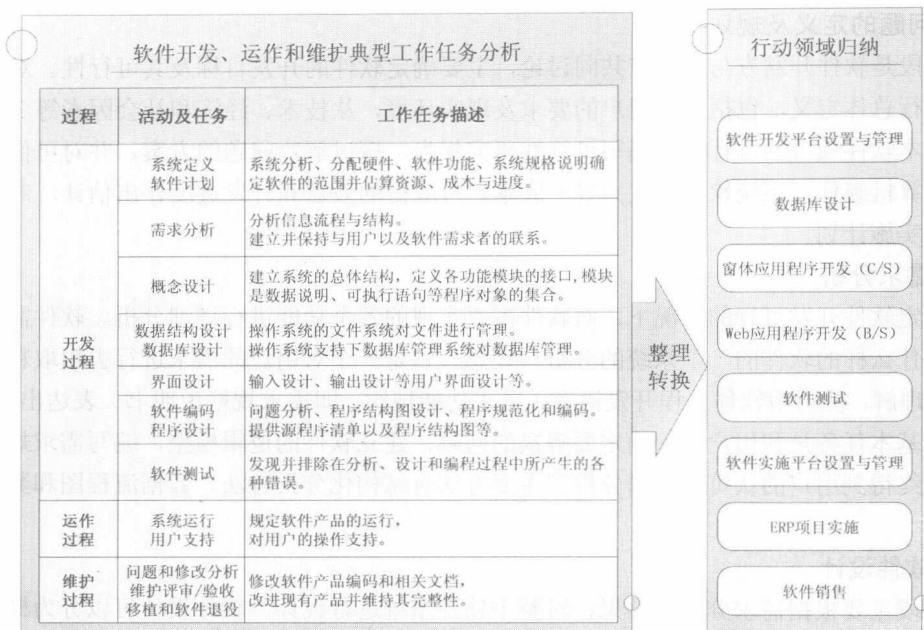
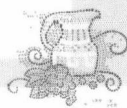


图 1-2 软件开发、运作和维护的典型工作任务

1.2 软件开发方法

目前，软件开发中常用的开发模型和方法主要有如下几种：

- 瀑布模型
- 增量模型
- 演化模型
- 螺旋模型
- 统一软件过程模型
- XP 过程模型

这些开发模型可以单独使用，也可以组合成混合型。开发模型确定后，则根据 GB/T8566-2001 的规定，选择适当的过程、活动和任务而形成软件项目的软件生存周期过程。

其中传统的瀑布模型各阶段的工作自顶向下从抽象到具体顺序进行。瀑布模型意味着在生命周期各阶段间存在着严格的顺序且相互依存。软件周期瀑布模型如图 1-3 所示。

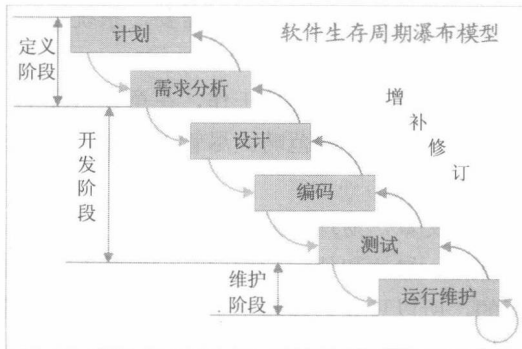


图 1-3 软件周期瀑布模型



1. 问题的定义及规划

此阶段是软件开发方与需求方共同讨论, 主要确定软件的开发目标及其可行性。对所要解决的问题进行总体定义, 包括了解用户的要求及现实环境, 从技术、经济和社会因素等 3 个方面研究并论证本软件项目的可行性, 编写可行性研究报告, 探讨解决问题的方案, 并对可供使用的资源 (如计算机硬件、系统软件、人力等) 成本、可取得的效益和开发进度作出估计。制订完成开发任务的实施计划。

2. 需求分析

在确定软件开发可行的情况下, 对软件需要实现的各个功能进行详细分析。软件需求分析就是对开发什么样的软件的一个系统的分析与设想。它是一个对用户的需求进行去粗取精、去伪存真、正确理解, 然后用软件工程开发语言 (形式功能规约, 即需求规格说明书) 表达出来的过程。本阶段的基本任务是和用户一起确定要解决的问题, 建立软件的逻辑模型, 编写需求规格说明书文档并最终得到用户的认可。需求分析的主要方法有结构化分析方法、数据流程图和数据字典等方法。

3. 软件设计

此阶段主要根据需求分析的结果, 对整个软件系统进行设计。软件设计可以分为概要设计和详细设计两个阶段。实际上软件设计的主要任务就是将软件分解成模块, 即能实现某个功能的数据和程序说明、可执行程序的程序单元。模块可以是一个函数、过程、子程序、一段带有程序说明的独立的程序和数据, 也可以是可组合、可分解和可更换的功能单元。分解成模块之后进行模块设计。概要设计就是结构设计, 其主要目标就是给出软件的模块结构, 用软件结构图表示。详细设计的首要任务就是设计模块的程序流程、算法和数据结构, 次要任务就是设计数据库, 常用方法还是结构化程序设计方法。

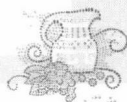
4. 程序编码

程序编码就是把详细设计的结果用计算机能理解的形式——计算机代码来表示, 即使用某种程序设计语言来编写程序。在程序编码中必须要制定统一、符合标准的编写规范, 以保证程序实现以下目标:

可维护性: 一个程序在其运行期间, 会逐步暴露出某些隐含的错误, 因而需要及时排错。同时, 用户还会提出某些新的要求, 这就需要对程序进行修改或扩充, 使其进一步完善。此外, 由于计算机软件与硬件的更新换代, 应用程序也需要做相应的调整或移值。这些工作都属于程序维护的任务。考虑到一般的软件应用系统要运行 3~10 年的时间, 因此, 程序维护的工作量是相当大的。一个程序即使其他方面都十分理想, 但是, 如果它不容易维护, 那么这个程序将不会有任何实际使用价值。所以说, 由于排错、改正、改进的需要, 系统维护是必要的。可维护性是目前程序设计所追求的主要目标之一。

可靠性: 一个程序正常情况下应该能正确地工作, 在意外情况下, 能适当地做出处理, 防止造成严重的损失。这些都属于程序可靠性的范畴。尽管人们不能指望一个程序是完全正确的, 但它应当是十分可靠的。

可理解性: 编写程序如同写文章, 易理解是很重要的。一个逻辑上完全正确但杂乱无章, 无法供人阅读、分析、测试、排错、修改和使用的程序是没有什么价值的。过去人们总是片面地认为程序仅仅是给计算机执行的, 所以只要它逻辑上正确, 计算机能按其逻辑正确地执行就够了, 至于程序是否简洁和清晰则无关紧要。实际上, 对一个大型程序来说, 要求它不仅逻辑上要正确, 计算机能够执行, 而且应当层次清楚, 简洁明了, 便于人们阅读。这是因为程序的维护工作量很



大,程序维护人员经常要维护他人编写的程序,如果一个程序不便于阅读,那么对程序检查与维护工作将会带来极大的困难。要使编写的程序易于理解,就必须有一个结构清晰的程序框架。它是保证程序正确、提高程序可读性与可维护性的基础。

高效率:程序效率是指计算机资源(如时间和空间)能否被有效地使用,即系统运行时应尽量用较少的空间,较快的速度完成规定的功能。由于硬件价格近年来大幅度下降,其性能不断完善和提高,所以效率已经不像以前那样举足轻重了,而程序设计者的工作效率远比程序效率重要,程序设计者工作效率的提高,不仅能减少经费支出,而且程序的出错率也会明显降低,还会减轻程序维护工作的负担。此外,效率与可维护性和可理解性通常是矛盾的,片面追求程序的运行效率不利于程序设计质量的全面提高。在实际编写程序的过程中,人们宁可牺牲一定的时间和空间,也要换取程序的可维护性、可理解性的提高。

健壮性:健壮性是指系统对错误操作、错误数据输入能予以识别与禁止的能力,不会因错误操作、错误数据输入及硬件故障而造成系统崩溃。这是系统长期平稳运行的基本前提。

5. 软件测试

软件在设计完成后要经过严密的测试,以发现软件在整个设计过程中存在的问题并加以纠正。软件测试的目的是以较小的代价发现尽可能多的错误。要实现这个目标的关键在于设计一套出色的测试用例(测试数据与功能和预期的输出结果组成了测试用例)。要设计出一套出色的测试用例,关键在于理解测试方法。不同的测试方法有不同的测试用例设计方法。两种常用的测试方法是白盒法和黑盒法。白盒法的测试对象是源程序,是依据程序内部的逻辑结构来发现软件的编程错误、结构错误和数据错误。结构错误包括逻辑、数据流、初始化等错误。用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果。黑盒法依据的是软件的功能或软件行为描述,发现软件的接口、功能和结构错误。其中接口错误包括内部/外部接口、资源管理、集成化以及系统错误。黑盒法用例设计的关键同样也是以较少的用例覆盖模块的输出和输入接口。

6. 运行维护

软件维护是软件生命周期中持续时间最长的阶段。在软件开发完成并投入使用后,由于多方面的原因,软件不能继续适应用户的要求。要延续软件的使用寿命,就必须对软件进行维护。软件的维护包括纠错性维护和改进性维护两个方面。

维护是指在已完成对软件的研制(分析、设计、编码和测试)工作并交付使用以后,对软件产品所进行的一些软件工程活动,即根据软件运行的情况,对软件进行适当修改以适应新的要求,以及纠正运行中发现的错误,编写软件问题报告、软件修改报告。一个中等规模的软件,如果研制阶段需要一年至二年的时间,在它投入使用以后,其运行或工作时间可能持续五年至十年。那么它的维护阶段也是运行的这五年至十年期间。在这段时间,人们几乎需要着手解决研制阶段所遇到的各种问题,同时还要解决某些维护工作本身特有的问题。做好软件维护工作,不仅能排除障碍,使软件能正常工作,而且还可以使它扩展功能,提高性能,为用户带来明显的经济效益。然而遗憾的是,对软件维护工作的重视往往远不如对软件研制工作的重视。而事实上,和软件研制工作相比,软件维护的工作量和成本都要大得多。在实际开发过程中,软件开发并不是从第一步进行到最后一步,而是在任何阶段,在进入下一阶段前一般都有一步或几步的回溯。在测试过程中的问题可能要求修改设计,用户可能会提出一些需要来修改需求说明书等。

第2章

项目实施过程

2.1 项目实施策略

项目实施成功是甲乙双方精诚合作、共同努力才能达到的结果。按照统一规划、统一标准、统一设计、统一流程、统一开发、统一推广的原则进行规划设计,坚持“总体规划、试点先行、典型设计、全面推广”的工作方针。项目质量管理体系融合 ISO9001 和 CMMI L3 形成统一的质量管理体系进行全面质量监控。项目实施策略如图 2-1 所示。



图 2-1 项目实施策略图

2.1.1 项目统一规划

(1) 成立系统领导组和业务专家组,负责领导、规划、监控、协调整个商品销售技术支持系统项目的实施进展状态及解决实施中的重大问题。

(2) 建立覆盖超市企业公司及下属单位的应用系统统一、数据结构统一、业务流程统一、管理规范统一的信息系统。

(3) 规划和规范业务流程和业务处理,并明确涉及的业务规则等约束条件,以规范和标准促进业务流程和超市管理的规范统一。

(4) 确定岗位职责和权限、时限。

2.1.2 模板化构建

(1) 在项目的统一规划指导下,进行系统的模板化构建。



(2) 在项目准备、差异需求分析、开发、数据转换、测试和验收等不同阶段,超市企业的相关人员统一参与,集中进行,形成模板。

(3) 尽早发现差异,处理差异,使得系统构建真正反映真实的需求,保证版本统一。

2.1.3 模板化实施

(1) 采用先试点,后推广的实施策略。充分调动试点单位的积极性,利用试点单位的积极参与在项目前期为项目建设积累经验,消除系统的主要缺陷,为系统的后期推广应用扫清障碍。

(2) 在实施过程中,融合管理规范体系的工作细节。

2.2 项目实施阶段

在严格的项目质量管理体系监管之下,整个项目实施过程就会透明化,有条不紊地开展。

2.2.1 项目准备阶段

(1) 明确总体目标、系统建设规模和应用范围。

(2) 确定试点单位,界定调研的对象以及进一步推广的步骤。

(3) 确定项目组织结构和人力保障体系,明确职责分工,做到任务到人,责任到人,并建立严格的协作规范和有效的沟通渠道,使得问题能够快速传达和反映,领导的决策能够及时贯彻实施。

(4) 部署项目实施计划安排,明确各阶段的任务,力求各个部门对项目实施各个阶段的任务有总体把握。

(5) 明确系统实施过程中的工作制度,包括计划组织、计划控制和计划变更等有效的项目管理制度,以及项目联络会制度。

(6) 项目实施工作环境筹备和部署。

(7) 明确数据准备方案以及工作方式,部署业务基础资料收集整理工作。

2.2.2 差异需求分析阶段

在统一系统规划、业务规范、数据标准及编码、设计规范的基础上,通过各种面向服务、面向对象、流程化、可定制的技术手段,一定程度满足各地超市管理的个性化差异。

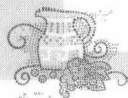
(1) 规范管理体系,根据各地实际业务提炼个性化需求,找出需求差异,并进行差异化处理。例如,报表格式、各种单据等。

(2) 加强需求分析过程中的薄弱环节工作的开展。例如,注重落实第三方接口厂家的洽谈接口协议工作,形成有效的接口方案、数据格式。

(3) 组织开展数据准备工作,将数据分析、数据转换工作提前开展,有利于系统总体进度的提前,同时能够避免今后由于数据质量原因导致系统的使用问题发生。

2.2.3 开发阶段

前期分析和设计过程产生的设计文档(需求分析文档、架构设计文档、概要设计文档、详细设计



文档和数据库设计文档) 以及差异需求分析阶段产生的差异需求文档, 是进行软件编码开发的基础。

1. 制定阶段计划、分配任务

项目经理根据《项目开发计划》和《详细设计说明书》细化编码实现的任务并分配给程序员。

(1) 项目经理根据项目开发计划对实现阶段的任务进行分解和分配。将实现里程碑(milestone)内需要做的任务进行划分并分配给程序员, 分配的基本要素包括任务、承担者、持续时间, 每项任务的持续时间应细化到 3 天之内以便进度控制;

(2) 程序员对分配的任务做出承诺。如果对任务分配有异议, 与项目经理进行协调, 直到每项任务均得到承诺。

2. 功能开发

程序员根据详细设计说明书、数据库设计文档进行程序源代码编写以及程序文档(操作手册、在线帮助文档等)的编写工作。

(1) 程序员按照分配的任务项, 根据详细设计文档编写程序的源代码。编码工作包括: 构建程序框架, 实现设计的接口, 实现具体的算法编码, 创建界面, 创建数据库表, 创建数据库数据, 创建数据库程序(存储过程、函数、触发器等)。

(2) 程序员根据文档模板编写源代码相关部分的文档(或文档片段), 包括操作手册, 在线帮助。

(3) 程序员基于 CVS 进行源代码管理。每天下班前, 程序员将编译通过的源代码以及编写完成的文档上载到配置库的开发库中。每天上班后, 程序员下载最新的源代码到本地。程序员在上载代码前, 必须经过调试, 并经过自查。开发阶段流程如图 2-2 所示。

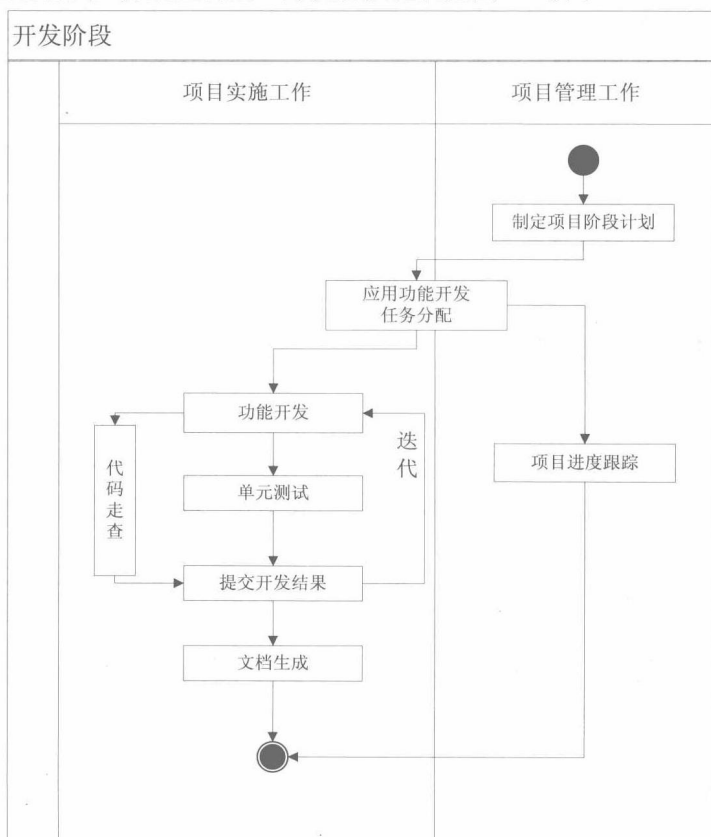


图 2-2 开发阶段流程图



3. 代码走查

项目经理安排项目组内的程序员进行程序源代码走查,发现并解决缺陷。

(1) 制定代码走查计划。

(2) 组织程序员召开代码走查会议,由程序员讲解自己的代码,其他程序员讨论,记录发现的缺陷。

(3) 程序员根据程序源代码中发现的缺陷进行修正。

4. 单元测试

程序员根据详细设计文档对自己所开发的程序编写单元测试用例,并实施单元测试。

(1) 程序员确定要实施单元测试的程序单元。

(2) 程序员依据详细设计文档编写单元测试用例。

(3) 程序员实施单元测试用例。

(4) 程序员就单元测试用例运行中发现的问题进行跟踪和调试。

(5) 填写《单元测试报告》。

(6) 填写《需求跟踪矩阵》。

5. 提交开发结果

项目经理组织系统设计师和程序员对核心部分的程序源代码进行代码评审,评审通过后,程序员提交开发结果。

(1) 制定代码评审计划,准备代码评审会。

(2) 项目经理主持代码评审会议,程序员阐述待评审的程序源代码,评委进行评审,记录缺陷,最终由评委给出是否通过的评审结论。

(3) 程序员根据《代码评审记录》中记录的缺陷进行修改,并提交给项目经理和评委进行验证,直至所有缺陷关闭。

(4) 程序员提交开发结果。

(5) 准备集成环境,应用部署,编写相关文档,开展测试工作。

2.2.4 测试阶段

软件危机曾经是软件界甚至整个计算机界最热门的话题。为了解决这场危机,软件从业人员、专家和学者做出了大量的努力。现在人们已经逐步认识到所谓的软件危机实际上仅是一种状况,那就是软件中有错误,正是这些错误导致了软件开发在成本、进度和质量上的失控。有错是软件属性,而且是无法改变的,因为软件是由人来完成的,所有由人做的工作都不会是完美无缺的。问题在于我们如何去避免错误的产生和消除已经产生的错误,使程序中的错误密度达到尽可能低的程度。由此可见测试是整个软件工程相当重要的一环。

所谓测试就是用已知的输入在已知环境中动态地执行系统(或系统的部件)。

软件测试是软件开发的一个重要环节,同时也是软件质量保证的一个重要环节。我们对超市商品管理系统进行的测试包括:单元测试、集成测试、系统测试和验收测试。

测试的目标:以较少的用例、时间和人力找出软件中潜在的各种错误和缺陷,以确保系统的质量。

测试范围:系统中所有应用软件以及软件集成。