

计算机应用与软件技术专业领域技能型紧缺人才培养培训系列教材

软件工程实用技术

主编 尹毅峰

高职高专
适用



高等教育出版社
HIGHER EDUCATION PRESS

计算机应用与软件技术专业领域技能型紧缺人才培养培训系列教材

软件工程实用技术

主编 尹毅峰

高等教育出版社

内 容 提 要

本书是“计算机应用与软件技术专业领域技能型紧缺人才培养培训系列”教材之一,是根据教育部“技能型紧缺人才培养培训项目”的精神,结合人才市场需求,以实用性为原则编写的。

本书以具体软件项目开发过程为主线,融入软件工程的基本原理、方法,力求理论联系实际。全书共五章,内容由三部分组成:1. 软件工程的理论部分,包含软件工程概述、软件项目准备阶段、软件项目设计阶段、代码实现、软件项目的后期阶段等知识点;2. Rose 软件工具的建模技术,UML 知识的初步应用;3. 结合具体的计算机软件项目指导开发文档的编制。

本书适合作为各类高等职业技术学校、部分普通高等院校计算机应用与软件技术专业应用型人才培养用教材,也可供广大计算机爱好者学习使用。

图书在版编目(CIP)数据

软件工程实用技术/尹毅峰主编. —北京:高等教育出版社,2005.6
ISBN 7-04-016898-7

I. 软... II. 尹... III. 软件工程—高等学校:技术学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2005)第 062758 号

责任编辑 古 锋 张永琳 封面设计 吴 昊 责任印制 潘文瑞

出版发行	高等教育出版社	购书热线	010—58581118
社 址	北京市西城区德外大街 4 号		021—56964871
邮政编码	100011	免费咨询	800—810—0598
总 机	010—82028899	网 址	http://www.hep.edu.cn
传 真	021—56965341		http://www.hep.com.cn
			http://www.hepsh.com
排 版	南京理工出版信息技术有限公司		
印 刷	宜兴市德胜印刷有限公司		
开 本	787×1092 1/16	版 次	2005 年 7 月第 1 版
印 张	14.75	印 次	2005 年 7 月第 1 次
字 数	368 000	定 价	20.50 元

凡购买高等教育出版社图书,如有缺页、倒页、脱页等质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

出版说明

为实现党的十六大提出的全面建设小康社会的奋斗目标,落实《国务院关于大力推进职业教育改革与发展的决定》,促进职业教育更好地适应社会主义现代化建设对生产、服务第一线技能型人才的需要,缓解劳动力市场上制造业和现代服务业技能型人才紧缺状况,教育部、劳动和社会保障部、国防科工委、信息产业部、交通部、卫生部决定组织实施“职业院校制造业和现代服务业技能型紧缺人才培养培训工程”(教职成[2003]5号,以下简称《工程》)。《工程》的目标是:“根据劳动力市场技能型人才的紧缺状况和相关行业人力资源需求预测,在数控技术应用、计算机应用与软件技术、汽车运用与维修、护理等四个专业领域,全国选择确定500多所职业院校作为技能型紧缺人才示范性培养培训基地;建立校企合作进行人才培养的新模式,有效加强相关职业院校与企事业单位的合作,不断加强基地建设,扩大基地培养培训能力,缓解劳动力市场上技能型人才的紧缺状况;发挥技能型紧缺人才培养培训基地在探索新的培养培训模式、优化教学与训练过程等方面的示范作用,提高职业教育对社会和企业需求的反应能力,促进整个职业教育事业的改革与发展。”

《工程》实施启动以来,各有关职业院校在职业教育人才培养目标、人才培养模式以及专业设置、课程改革等方面做了大量的研究、探索和实践,取得了不少成果。为使这些研究成果能够得以固化并更好地推广,从而总体上提高职业教育人才培养的质量,我们组织了有关职业院校进行了多次研讨,根据“教育部办公厅、信息产业部办公厅关于确定职业院校开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的通知”(教职成厅[2003]5号)中的两年制高等职业教育计算机应用与软件技术专业领域技能型紧缺人才培养指导方案,确立了“以就业为导向,以企业需求为依据”的宗旨,“以综合职业素质为基础,以能力为本位”的思路,“适应行业技术发展,以应用为目的”的体系,“以学生为主体,体现教学组织的科学性和灵活性”的风格,组织编写了一批“计算机应用与软件技术专业领域技能型紧缺人才培养培训系列教材”。这些教材结合《工程》的指导思想与目标任务,反映了最新的教学改革方向,很值得广大职业院校借鉴。

此系列教材出版后,我们还将不定期地举行相关课程的研讨与培训活动,并联合一些软件企业共同探讨人才培养目标、人才培养模式以及专业设置、课程改革,为各院校提供一个加强校企合作、交流的互动平台。

“计算机应用与软件技术专业领域技能型紧缺人才培养培训系列教材”适合高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院、继续教育学院和民办高校使用。

高等教育出版社

2004年6月

前 言

本书是根据教育部等部委组织实施的“职业院校制造业和现代服务业技能型紧缺人才培养培训工程”中有关计算机应用与软件技术专业领域技能型紧缺人才培养指导方案的精神,按照高等职业技术教育技能型人才的培养目标和基本要求编写的“计算机应用与软件技术专业领域技能型紧缺人才培养培训系列教材”之一。

目前,国内有关软件工程的教材大多数以理论讲授为主,且高职高专所用的教材大多沿袭了本科“软件工程”课程的教学模式,理论性强,缺乏实践环节。因而高职高专的软件工程教学急需能结合具体的软件设计项目、提高高职高专软件技术专业学生实践能力的教材。本书从当前软件行业的需求出发,培养学生软件文档的编制能力,并使能够较熟练地掌握 Rose 这样的软件工程工具进行建模,目的是培养高职高专学生的软件工程的思想。本书的特点是在内容的组织上尽可能贴近实际项目开发,教材主要内容包含四个部分,采用了 3+1 的编写形式,即在软件工程理论中融入具体项目开发实践、结合 Rose 软件的使用、附加一整套计算机软件开发文档书(以具体项目为实例)。

本书充分考虑了高职高专学生在软件开发中所扮演的角色。一方面,作为软件项目的辅助人员或测试人员,本教材为学生提供了文档的编制模板,将软件工程的思想融于文档编制中,教授工具软件 Rose 的使用;另一方面,对于有潜力的学生,从本教材所包含的具体项目开发任务的实践中能够较系统地掌握软件工程在软件开发中的作用,通过软件开发全过程的实践与项目管理实践,了解软件开发/测试过程,学会文档编写及管理内容。

建议授课课时数为:第 1 章 2 课时,第 2 章 6 课时,第 3 章 10 课时,第 4 章 6 课时,第 5 章 12 课时,实验课时根据具体的情况安排。

本书由尹毅峰担任主编,李小遐任副主编,编写分工为:第 1 章由尹毅峰编写,第 2 章由李小遐编写,第 3 章由刘慧梅、陈巧莉、尹毅峰编写,第 4 章由魏晓艳、刘龙江编写,第 5 章由张帆编写,附录由王勃编写。

在本书的编写大纲讨论过程中,陕西国防工业职业技术学院贾宝勤提出了很多宝贵建议,在此表示诚挚的感谢。此外,在本书的编写过程中参考了大量文献,在此对这些文献的作者表示诚挚的谢意!

由于编者水平有限,加之时间仓促,书中难免有错漏之处,敬请广大读者批评指正。编者的电子邮箱为 yinyifeng@yeah.net。

编 者

2005 年 3 月

目 录

第 1 章 软件工程概述	1
1.1 软件的概念、特点和分类	1
1.2 软件的发展和软件危机	3
1.3 软件生存周期与软件生存周期模型	5
本章小结	10
习题	10
第 2 章 项目准备阶段	12
2.1 总体规划阶段	12
2.2 需求分析	17
2.3 项目立项	24
本章小结	25
习题	25
第 3 章 项目设计阶段	27
3.1 系统分析	27
3.2 系统设计	39
3.3 面向对象的分析和设计方法	53
3.4 代码实现	84
本章小结	95
习题	96
第 4 章 项目后期阶段	98
4.1 项目测试	98
4.2 文档编制	115
4.3 项目验收及发布	119
4.4 软件维护	120
4.5 项目后期工程管理	130

目 录

本章小结.....	142
习题.....	143
第 5 章 Rose 软件建模技术	144
5.1 Rose 简介	144
5.2 Rose 的安装与启动	144
5.3 使用 Rose/C++ 建立系统模型.....	152
5.4 关系操作	160
5.5 交互图	165
5.6 状态图	168
5.7 生成代码及逆向工程	169
本章小结.....	172
习题.....	172
附录 计算机软件开发文档编写指南(以具体项目为实例).....	173
附录 1 可行性分析报告	173
附录 2 项目开发计划	179
附录 3 软件规格需求说明书	182
附录 4 概要设计说明书	187
附录 5 详细设计说明书	196
附录 6 用户操作手册	201
附录 7 测试分析报告	206
附录 8 开发进度月报	209
附录 9 项目开发总结报告	211
附录 10 程序维护手册	215
附录 11 软件问题报告	219
附录 12 软件修改报告	222
参考文献.....	226

第 1 章 软件工程概述

本章从软件的概念和特点开始,介绍软件危机的产生,由此引出软件工程的理论,并且详细描述软件生存周期的各个阶段以及软件生存周期模型的相关概念。

1.1 软件的概念、特点和分类

1.1.1 软件的概念

“软件(software)”一词最早出现于 20 世纪 60 年代,根据 GB/T 11457—89 中的定义,软件是指与计算机系统的操作相关的计算机程序、规程、规则以及任何与之有关的文件。简而言之,软件包括程序(program)和文档(document)两部分。在软件的定义中,“程序”是指对计算任务的处理对象和处理规则的描述,计算机程序包括源程序和目标程序。同一程序的源代码和目标代码应当视为同一作品。源程序是指用高级语言或汇编语言编写的程序,目标代码是指源程序经编译或解释加工以后可以由计算机直接执行的程序。“文档”是与计算机程序功能、设计、编制、使用相关的文字或图形资料,是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。软件与硬件一起构成完整的计算机系统,它们是相互依存的,缺一不可的。

1.1.2 软件的特点

软件是一种特殊的产品,它具有下列一些特点:

① 软件的逻辑性和抽象性。计算机的逻辑性特点是通过软件的执行体现出来的,软件具有很强的逻辑性。这个特性使软件与其他工程对象有着明显的差异,但软件用户却无法看到软件本身的形态。同一个软件可以记录在打孔的纸带上、内存芯片内、磁盘或者光盘上,必须通过观察、分析、思考、判断,才能了解它的功能、性能等特性。

② 软件没有明显的制造过程。软件一旦被开发成功,利用拷贝技术可以用很低的成本大量产生该软件的副本,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 在使用过程中无磨损、老化的问题。软件的数字化特性使其在生存周期后期不会因为磨损而老化,但会为了适应硬件、环境以及需求的变化而进行修改,而这些修改有不可避免的引入错误,导致软件失效率升高,从而造成软件退化。当修改的成本变得难以接受时,软件就被抛弃。

④ 软件对硬件和环境有着不同程度的依赖性。最常见的用汇编语言编制的软件,根据机器型号的不同,程序代码相差很大。

⑤ 软件的开发至今尚未完全摆脱手工作坊式的开发方式,生产效率低。

⑥ 软件是复杂的,而且以后会更加复杂。软件是人类有史以来生产的复杂度最高的工业产品。软件涉及人类社会的各行各业、方方面面,软件开发常常涉及其他领域的专门知识,从而对软件工程师提出了很高的要求。

⑦ 软件的成本相当昂贵。软件开发需要投入大量的、高强度的脑力劳动,成本非常高,风险也大。现在软件的开销已大大超过了硬件的开销。

⑧ 大多数软件是自定的,而不是通过已有的组件组装而来的。

⑨ 软件工作牵涉很多社会因素。许多软件的开发和运行涉及机构、体制和管理方式等问题,还会涉及人们的观念和心理。这些人的因素常常成为软件开发的困难所在,直接影响到项目的成败。

1.1.3 软件 的 分 类

人们从各种不同的角度对计算机软件提出了许多分类的方法。按软件的功能分为系统软件、支撑软件、应用软件;按软件的规模分为微型、小型、中型、大型、超大型软件;按软件的工作方式分为实时、分时、交互、批处理软件;按软件服务对象的范围分为项目软件、产品软件。

很难兼顾多个不同类型的对象和范围对软件给出一个科学、通用的分类,因为各种分类方法之间的相互渗透,难以找出软件之间一般的差异性。因此,本书从应用领域的角度简单地把软件做以下分类。

(1) 系统软件

系统软件是与计算机硬件紧密结合,构成用户在某一方面使用计算机的基础平台,具有良好的用户界面。操作系统是最常见的系统软件,是对计算机全部软、硬件资源进行控制和管理的大型程序,是直接用于裸机上的最基本的系统软件,其他软件必须在操作系统的支持下才能运行,因而是软件系统的核心。此种软件的开发需要高素质的程序设计人员和大规模的软件集团公司(例如微软)。

(2) 实时软件

计算机的高速处理能力使得应用计算机对事件和数据进行实时处理成为可能,例如工业过程控制、卫星导弹的运行控制、管理信息处理等。这些完成处理、反馈、控制过程的软件称为实时软件。这些软件的特征是,对事件的响应时间有严格限定,主要包括数据收集、实时分析和控制输出3个部分。实时软件既可以应用于信息处理,也广泛应用于过程控制。

(3) 嵌入式软件

随着数字化产品的不断普及,微型处理器在消费产品和工业产品中的应用越来越广泛。这些用于提供控制和专职功能的软件称之为嵌入式软件。众所周知,“一次编程、到处使用”的Java软件概念原本就是针对网上嵌入式设备提出的。目前,SUN公司已经推出了针对信息家电的Java新版本——J2ME(Java 2 Platform Micro Edition),其技术日趋成熟,并开始投入使用,这对网络化的数字化家电设备十分有利。嵌入式软件一般为某一单独的应用专门设计,驻留在只读存储器中,执行有限的专职功能(如空调等简单器具的操作及控制)或提供重要的功能及复杂的控制能力(如数控机床等控制系统)。另外,它还可以用来进行信

息处理(如手机、电子词典、PDA 等)。

(4) 基于 Web 的软件

互联网的普及给计算机的应用提供了更广泛的空间,Internet 提供了无所不包、使用方便的软件资源。为了浏览和检索,基于 Web 的可执行指令(如 CGI、HTML、Perl 或 Java)以及数据(如超文本以及可视、音频等格式的数据)应用于浏览器的软件已成为当今的又一热点。

(5) 实用软件

实用软件是针对计算机在某一领域或特定工作性质中应用的具有一定通用性质的软件。它通常可分为应用软件和支撑软件两大类。应用软件是在操作系统的基础上为某一特定领域应用而开发的软件,例如商业处理软件、科学计算软件、计算机辅助设计软件、人工智能软件等;支撑软件一般用来辅助和支持开发人员开发和维护软件,例如需求分析工具、设计工具、编码工具、测试工具、维护和管理工具等。

1.2 软件的发展和软件危机

1.2.1 软件的发展

如图 1.2.1 所示,软件的发展经历了 4 个阶段。

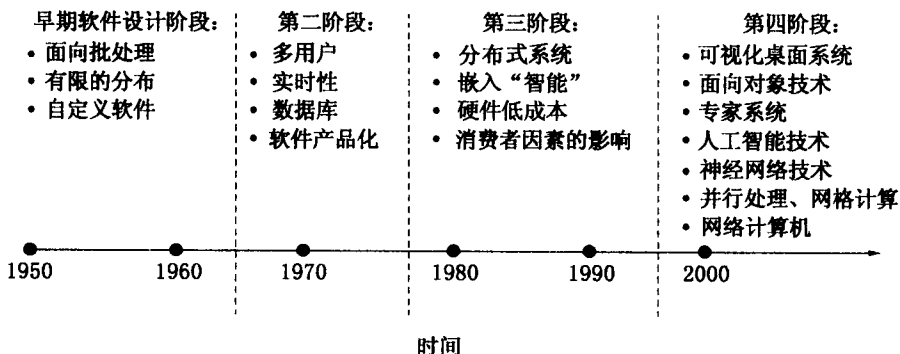


图 1.2.1 软件的发展阶段

(1) 早期软件设计阶段

这个阶段也称为程序设计时代。由于计算机硬件和存储设备条件的限制,软件的生产方式主要采用个体手工劳动,在软件设计方法上追求编程技巧和程序的运行效率,使用的编程工具是机器语言和汇编语言,造成程序难阅读、难修改、设计效率低。

(2) 第二阶段

此阶段也称为程序系统时代。这一阶段出现了 BASIC 和 FORTRAN 等高级语言编程工具,软件的生产方式是小作坊式的合作开发,没有形成规模化生产;程序设计方法仍然依

靠个人技巧,但开始提出了结构化方法;硬件特征是,速度、容量及工作可靠性有明显提高,价格降低,销售有爆炸性增长。软件特征是,程序员数量猛增,其他行业人员大量进入这个行业,由于缺乏训练,开发人员素质较低。在这一阶段,软件开发的重要性已经被人们所重视,大量软件开发的需求已提出,但开发技术没有新的突破,开发人员的素质和落后的开发技术不适应规模大、结构复杂的软件开发,因此产生了尖锐的矛盾,导致软件危机的产生。

(3) 第三阶段

此阶段的显著特点是具有成熟的结构化程序设计思想,随着硬件的低成本化以及软件产品化所带来的消费者因素的影响,软件产品的开发周期大幅度缩短。

(4) 第四阶段

此阶段也称为软件工程时代。在这一阶段,软件产品的生产方式是工程化的,利用数据库、开发工具、开发环境、网络、分布式、面向对象等多种技术来开发软件;硬件特征是向超高速、大容量、微型化以及网络化方向发展;软件特征是开发技术有很大进步,但是未能获得突破性进展,软件价格不断上升,没有完全摆脱软件危机。

1.2.2 软件危机

1. 软件危机的概念

软件危机指的是,在计算机软件的开发和维护过程中所遇到的一系列严重问题。“软件危机”(software crisis)这个词最早是1968年北大西洋公约组织的计算机科学家在联邦德国召开的国际学术会议上提出的。软件危机包含两方面问题:首先,如何开发软件,以满足不断增长、日趋复杂的需求;其次,如何维护数量不断膨胀的软件产品。具体而言,软件危机主要有7个方面的表现:

- ① 不能准确地估计软件开发成本和进度,开发成本经常超出预算,实际进度比预定计划一再拖延的现象并不罕见。
- ② 经常发生用户对“已交付”的系统不满意的现象。
- ③ 软件产品的质量无法保证,故障出现频繁。
- ④ 软件的可维护程度非常低。
- ⑤ 软件通常缺乏文档资料的支持,代码可读性差。
- ⑥ 软件的开发成本不断提高。
- ⑦ 软件开发生产率的提高赶不上硬件的发展和人们需求的增长。

2. 软件危机的起因

软件危机的原因,一方面与软件本身的特点有关,另一方面与软件开发和维护的方法不正确有关。本章前面已经简单介绍了软件的特点。软件开发和维护的不正确方法主要表现为忽视软件开发前期的需求分析;开发过程没有统一的、规范的方法论的指导,文档资料不齐全,忽视人与人的交流;忽视测试阶段的工作,提交用户的软件质量差;轻视软件的维护。这些问题大多数都是由于软件开发过程管理上的原因造成的。

软件危机的一个典型案例是 OS360 操作系统,这个操作系统主要用于 IBM360 系列主机。该操作系统作为一个极度复杂的软件项目经历了数十年,甚至产生了一套不包括在原始设计方案之中的工作系统。作为第一个超大型的软件项目,OS360 操作系统动用的程序员的数量达到了 1 000 人左右。Fred Brooks 在随后他的大作《人月神话》(*The Mythical Man-Month*)中曾经承认,在他管理这个项目的时候,他犯了一个价值数百万美元的错误。

软件危机的直接后果是:

① 财产损失。由于软件的差错可能导致巨大的财产损失,人们所知道的最为惨痛的教训之一就是阿里亚娜火箭的爆炸。

② 人员伤亡。由于计算机软件被广泛应用于包括医疗等与人的生命息息相关的行业,这也使得软件的错误导致人员伤亡成为可能。

1.2.3 软件工程的概念

软件工程的兴起源于 20 世纪 60~80 年代的软件危机。在那个时代,很多软件最后都得到了一个悲惨的结局,很多软件项目的开发时间大大超出了规划的时间表,一些项目导致了财产的流失,甚至某些软件导致了人员伤亡。同时,软件开发人员也发现软件开发的难度越来越大。

软件项目的开发过程不是一个简单的编程过程,而是一个包含了从规划、分析、设计、编码、测试到维护的一系列工程活动的复杂过程。本章将从软件生存周期和软件生存周期模型两方面来介绍软件工程的概念。

1.3 软件生存周期与软件生存周期模型

1.3.1 软件生存周期

1. 自顶向下的分析方法

随着管理信息系统(MIS)规模的扩大和对开发经验的总结与归纳,自顶向下的系统分析方法论逐步得到了发展和完善。自顶向下法要求开发者首先制定系统的总体规划,然后逐步分离出高度结构化的子系统,从上至下实现整个系统。运用这类方法可以为企业或机构 MIS 的中期或长期发展规划奠定基础,同时支持 MIS 的整体性,为系统的总体规划、子系统的协调和通信提供保障。但它同样也存在缺点:对系统分析、设计人员要求较高。在大系统中,对下层系统的实施往往缺乏约束力,开发的周期长,系统复杂,成本较高。

由于软件的生产过程实现了工程化,为了更好地管理软件的生产、控制软件的进度,借用工程中产品生存周期而得到软件生存周期的概念。引入软件生存周期概念,对于软件生产有着非常重要的意义,可使软件生产具有相应的模式、相应的流程、相应的工序和步骤。

2. 软件生存周期

软件生存周期是指一个软件从提出开发要求开始直到该软件退役为止的整个时期。在 GB 8567 中规定了软件的生存周期,如图 1.3.1 所示。软件的生存周期可以划分为若干阶段,每个阶段有明确的任务,使得规模大、结构复杂和管理复杂的软件开发变得容易控制和管理。通常,软件生存周期包括可行性分析和项目开发计划、需求分析、概要设计、详细设计、编码、测试、软件维护几个阶段。

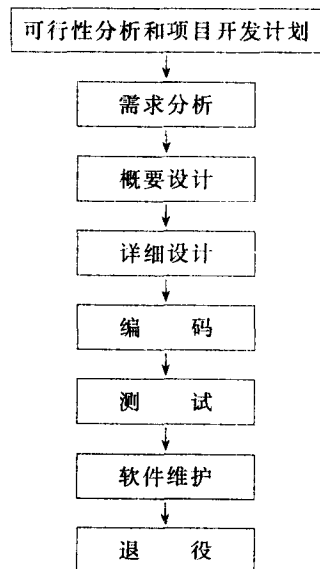


图 1.3.1 软件生存周期

软件生存周期的各阶段有不同的划分方法。软件规模、种类、开发方式、开发环境以及开发使用的方法都影响软件生存周期的划分。在划分软件生存周期的阶段时,应遵循的基本原则是,各阶段的任务尽可能相对独立,同一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程度,简化不同阶段之间的联系,有利于软件项目开发的组织和管理。

(1) 可行性分析和项目开发计划

可行性分析和项目开发计划阶段必须要回答的问题包括:要解决的问题是什么?该问题是否有解决方法?若有解决问题的办法,则需要多少人力、物力和财力?需要多少时间?要回答这些问题,就要进行问题定义、可行性分析,制定项目开发计划。

当用户提出一个软件开发需求后,系统分析员首先要明确解决该软件项目的性质:是数据处理问题还是实时控制问题,是科学计算问题还是人工智能问题等。还要明确该项目的目标是什么,该项目的规模和预算如何等。

系统分析员通过对用户和使用部门负责人的访问和调查、开会讨论等一系列方法来解决上述问题。在清楚了问题的性质、目标、规模后,还要确定该问题有没有行得通的解决办法。

系统分析员要进行压缩和简化的需求分析和设计,也就是在高层次上进行分析和设计,探索这个问题是否值得去解决,是否有可行的解决办法。最后要提交可行性研究报告。经过可行性分析后,确定该问题值得去解决,然后制定项目开发计划。根据开发项目的目标、功能、性能及规模,估计项目需要的资源,即需要的计算机硬件资源,需要的软件开发工具和应用软件包,需要的开发人员数目及层次。还要对软件开发费用做出估算,对开发进度做出估计,制定完成开发任务的实施计划。最后,将项目开发预算和可行性分析报告等文档、资料一起提交管理部门审查。

(2) 需求分析

需求分析阶段的任务不是具体地解决问题,而是确定软件系统必须做什么,软件系统必须具备哪些功能。需求分析的主要目的是软件设计人员通过与用户广泛交流得出所要完成的目标系统必须具备哪些功能,应该为用户完成些什么工作。需求分析相当于在用户和软

件工程人员之间架设了一座桥梁,软件工程人员通过需求分析得到用户的需求,成为软件编制所实现的目标。用户了解他们所面对的问题,知道必须做什么,但是通常不能完整、准确地表达出来,也不知道怎样用计算机解决他们的问题。而软件开发人员虽然知道怎样用软件完成人们提出的各种功能要求,但是对用户的具体业务和需求不完全清楚,这就是需求分析阶段的困难所在。系统分析员要和用户密切配合,充分交流,充分理解用户的业务流程,完整、全面地收集、分析用户业务中的信息和处理流程,从中分析出用户要求的功能和性能,并完整、准确地表达出来。这一阶段要给出软件需求规格说明书。

需求分析的质量直接关系到软件的成败,是软件生命周期中的关键环节。由于用户和计算机工程人员之间的交流有阻碍,使计算机工程人员不能很好地理解问题域,用户又对目标系统存在很多不清楚的地方。传统的数据流分析法、功能分析法等对这个问题并不能有效地解决。面向对象的方法的出现,正好为此问题提供了一个较好的解决方案。因为用“对象”的观点或方法来认识问题是人类的自然趋向,分析问题以及解决问题,用基于“对象”的概念模型来建立问题域模型自然成为系统分析员与用户交流的有效工具。

用面向对象的方法进行需求分析,其根本要点在于,利用“对象”的概念模型建立一个针对问题域的模型,用户和软件工程师通过该模型进行交流,并在这么一个基于“对象”的问题域模型的基础上形成需求规格说明书。

(3) 概要设计

“概括地说,应该如何解决这个问题”,是概要设计阶段必须回答的关键问题,应该考虑几种可能的解决方案。首先,低成本的解决方案,系统只要求完成必要的工作,不必做额外的工作。其次,中等成本的解决方案,这样的系统不仅能够很好地完成预定的任务,使用起来也很方便,而且可能还具有用户没有具体指定的某些功能和特点。例如,目标系统的一些主要功能是用计算机自动完成还是用人工完成;如果使用计算机,那么是使用批处理方式还是人机交互方式;信息存储使用传统的文件系统还是数据库。通常,至少应该考虑下述几类可能的方案:虽然用户没有提出这些具体要求,但是系统分析员根据自己的知识和经验断定,这些附加的功能在实践中将证明是很有价值的。最后,高成本的“十全十美”的解决方案,这样的系统具有用户可能希望有的所有功能和特点。系统分析员应该使用系统流程图或其他工具描述每种可能的系统,估计每种方案的成本和效益,还应该在充分权衡各种方案利弊的基础上,推荐一个较好的系统(最佳方案),并且制定实现所推荐系统的详细计划。如果用户接受分析员推荐的系统,则可以着手完成本阶段的另一项主要工作——体系结构设计。

在概要设计阶段,开发人员要把确定的各项功能需求转换成体系结构,在该体系结构中,每个成分都是意义明确的模块,即每个模块都和某些功能需求相对应。

因此,概要设计就是设计软件的结构:该结构由哪些模块组成,这些模块的层次结构是怎样的,这些模块的调用关系是怎样的,每个模块的功能是什么。同时还要设计该项目的应用系统的总体数据结构和数据库结构,即应用系统要存储什么数据,这些数据是什么样的结构,它们之间有什么关系等。

(4) 详细设计

详细设计阶段就是为每个模块完整的功能进行具体描述,要把功能描述转变为精确的、

结构化的过程描述。即该模块的控制结构是怎样的,先做什么,后做什么,有什么样的条件判定,有什么重复处理等,并用相应的表示工具把这些控制结构表示出来。总体设计阶段以比较抽象概括的方式提出了解决问题的办法,详细设计阶段的任务就是把解法具体化,也就是回答“应该怎样具体地实现这个系统”这个关键问题。这个阶段的任务还不是编写程序,而是设计出程序的详细规格说明。这种规格说明的作用很类似于其他工程领域中工程师经常使用的工程蓝图,它们应该包含必要的细节,程序员可以根据它们写出实际的程序代码。通常用 HIPO 图(层次图加输入/处理/输出图)或 PDL 语言(过程设计语言)描述详细设计的结果。

(5) 编码

编码阶段就是把每个模块的控制结构转换成计算机可接受的程序代码,即写成以某特定程序设计语言表示的“源程序清单”。当然,写出的程序应结构好,清晰易读,并且与设计相一致。

(6) 测试

测试是保证软件质量的重要手段,其主要方式是在设计测试用例的基础上检验软件的各个组成部分。测试分为模块测试、组装测试、确认测试。模块测试是查找各模块在功能和结构上存在的问题。组装测试是将各模块按一定顺序组装起来进行的测试,主要是查找各模块之间接口上存在的问题。确认测试是按说明书上的功能逐项进行的,发现不满足用户需求的问题,决定开发的软件是否合格,能否交付用户使用等。

(7) 软件维护

作为软件生存周期中时间最长的阶段,软件维护阶段以已交付的软件投入正式使用为起点,此阶段可以持续几年甚至几十年。由于各方面的原因,可能要对使用过程中的软件进行修改,这些可能的原因包括:在运行中发现了软件隐含的错误;软件工作环境发生了变化;用户业务发生变化而需要扩充和增强软件的功能等。

软件维护阶段的关键任务是,通过各种必要的维护活动使系统持久地满足用户的需要。通常有 4 类维护活动:改正性维护,也就是诊断和改正在使用过程中发现的软件错误;适应性维护,即修改软件以适应环境的变化;完善性维护,即根据用户的要求改进或扩充软件,使它更完善;预防性维护,即修改软件为将来的维护活动预先做准备。

虽然没有把维护阶段进一步划分成更小的阶段,但是实际上每一项维护活动都应该经过提出维护要求(或报告问题),分析维护要求,提出维护方案,审批维护方案,确定维护计划,修改软件设计,修改程序,测试程序,复检验收等一系列步骤,因此实质上是经历了一次压缩和简化了的软件定义和开发的全过程。

上面的工作确定了解决问题的策略以及目标系统需要哪些程序。但是,怎样设计这些程序呢?结构设计的一条基本原理就是程序应该模块化,也就是一个大程序应该由许多规模适中的模块按合理的层次结构组织而成。总体设计阶段的第二项主要任务就是设计软件的结构,也就是确定程序由哪些模块组成以及模块间的关系。通常用层次图或结构图描绘软件的结构。

有些文献将生存周期划分为 5 个阶段,即要求定义、设计、编码、测试及维护。其中,要求定义阶段包括可行性研究和项目开发计划、需求分析,设计阶段包括概要设计和详细设计。

1.3.2 软件生存周期模型

模型是为了理解事物而对事物做出的一种抽象,它忽略了不必要的细节,是事物的一种抽象形式、一个规划、一个程式。软件生存周期模型是描述软件开发过程中各种活动如何执行的模型。

一个强有力的软件生存周期模型对软件开发提供了强有力的支持,为软件开发过程中所有活动提供了统一的政策保证,为参与软件开发的所有成员提供帮助和指导。它揭示了如何演绎软件过程的思想,软件生存周期模型化技术的基础,也是建立软件开发环境的核心。

软件生存周期模型确立了软件开发和演绎中各阶段的次序限制以及各阶段活动的准则,确立开发过程所遵守的规定和限制,便于各种活动的协调以及各种人员的有效通信,有利于活动重用和活动管理。

软件生存周期模型能表示各种活动的实际工作方式,各种活动间的同步和制约关系,以及活动的动态特性。生存周期模型应该容易为软件开发过程中的各类人员所理解,它应该适应不同的软件项目,具有较强的灵活性,以及支持软件开发环境的建立。

目前,有若干种软件生存周期模型,常见的有瀑布模型、增量模型、螺旋模型、喷泉模型、变换模型和基于知识的模型等。本章主要介绍瀑布模型这个典型的软件生存周期模型。如图 1.3.2 所示,瀑布模型是将软件生存周期各项活动规定为依线性顺序连接的若干阶段的模型。它包括需求分析、概要设计、详细设计、编码、软件测试和软件维护这 6 个重要阶段。其中,在需求分析阶段,要完成可行性分析报告、项目开发计划报告和需求分析报告的文档

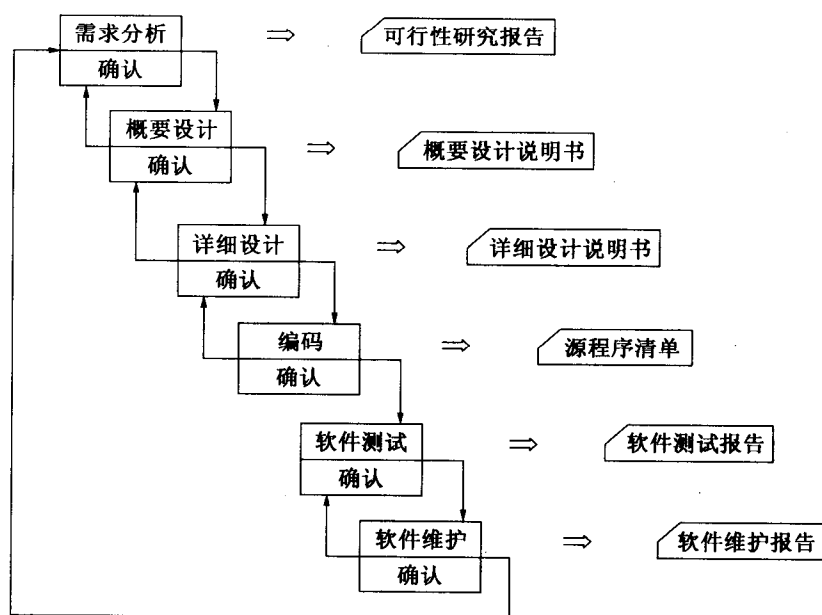


图 1.3.2 瀑布模型

编制工作。在此模型中,各个阶段由前至后、相互衔接,形成了固定的次序,这种次序如同瀑布,逐级下落。瀑布模型为 MIS 系统开发过程提供了一种有效的管理模型。根据这一模型制定开发计划,进行成本预算,组织开发力量,以项目的阶段评审和文档控制为手段,可以有效地对整个开发过程进行指导。因此,它是以文档作为驱动,适合于需求很明确的软件项目开发的模型。

由于软件项目的开发工程是动态的,有很多情况是无法预测的,在此提供软件工程的4条基本原则:

- ① 采取适宜的开发模型,控制易变的需求。
- ② 采用合适的设计方法,要做到软件模块化、抽象与信息隐藏、局部化、一致性以及适应性等,要有合适的设计方法的支持。
- ③ 提供高质量的工程支持、软件工具和环境对软件过程的支持。
- ④ 重视开发过程的管理,有效利用可用的资源,生产满足目标的软件产品,提高软件组织的生产能力。

本章小结

软件工程概念的引入主要是用于解决软件危机。本章针对软件项目的特点和不同分类,分析了软件的发展过程,从而针对软件危机产生的原因,阐述了软件的生存周期,对可行性分析和项目开发计划、需求分析、概要设计、详细设计、编码、测试、软件维护,这些软件工程的阶段进行了简要的描述。为了介绍对软件开发的规划,本章从众多的软件生存周期模型中选择了瀑布模型进行描述,目的是让读者能够了解软件生存周期模型对软件开发规划所起到的指导作用,了解软件工程的概念。

习 题

1. 软件需求分析,不应该包括()。
A) 问题分析
B) 信息域分析
C) 结构化程序设计
D) 确定逻辑模型
2. 在概要设计阶段,面向数据流的设计方法要解决的任务是()。
A) 取得软件系统的结构
B) 划分事务流与变换流
C) 进行变换分析
D) 进行事务分析
3. 将每个模块的控制结构转换成计算机可接受的程序代码是()阶段的任务。
A) 编码
B) 需求分析
C) 详细设计
D) 测试
4. 软件生存周期中时间最长的阶段是()。
A) 概要设计
B) 详细设计
C) 软件维护
D) 需求分析
5. 使用软件工具根据软件设计者的说明自动生成源代码,这种软件开发模型叫