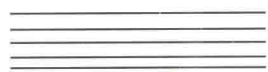




软件工程



项目教程

主 编 余明艳 郭庚麒

副主编 刘武萍 周淦淼 马宏茹

软件工程项目教程

主 编 余明艳 郭庚麒
副主编 刘武萍 周淦森 马宏茹
参 编 张春平 卢 慧 翟嘉伊



东软电子出版社
· 大连 ·

软件工程项目教程/余明艳,郭庚麒主编. —大连:东软电子出版社,2015.2
ISBN 978-7-89436-334-3

策划编辑: 杨光达
光盘开发: 马李昕

责任编辑: 朱 娜
装帧设计: 万点书艺

出版/发行: 东软电子出版社
地 址: 大连市软件园路8号
邮 编: 116023
电话/传真: 0411-84835089
网 址: <http://www.neubooks.com>
电子邮箱: nep@neusoft.edu.cn

出版时间: 2015年2月
印制时间: 2015年2月第1次印制
字 数: 255千字

印 制 者: 吉林庆达光盘科技有限责任公司
抚顺光辉彩色广告印刷有限公司

前 言

从软件工程的观念正式提出至今,软件工程技术取得了大量的理论和应用研究成果,促进了软件产业的飞速发展,同时软件工程学也已成为计算机科学的一个重要分支。软件工程从工程化的原理出发,按照标准化规程以及工程实践开发和管理项目,进行过程改进,它涉及软件定义、开发、使用与维护等软件生命的全过程。

本教材设计结构为由项目分析入手结合相关案例,由浅入深地引导学生掌握软件项目开发的全过程。编写过程中注重以项目为背景,以技能为主线,精心选择相关项目及案例,编排层次清晰,结构严谨,每个项目后面还配有一定数量的习题,还提供了教学课件,可以帮助学生思考、分析和学习。

一、教材特色

- 项目驱动、结合实例

本教材采用项目驱动方式撰写,提出项目实例,分析完成项目,介绍必需、够用的知识和技术,培养解决问题的能力。项目实例教学贯穿在全教材内容中,使阐述内容及学生学习均有很强的目的性,极大地提高学生的学习兴趣和综合能力。

- 循序渐进,通俗易懂

内容简明,案例讲解通俗易懂;步骤分析清晰,实用性强;知识点明确,易于学习。

二、内容介绍

本教材共分6个项目,主要培养学生软件项目开发的能力,具体内容结构如下:

项目1:签订软件开发合同及组建项目组,主要完成学习认识软件、认识软件危机、认识软件的生存周期与开发模型、组建项目组、起草项目开发合同等任务。

项目2:软件需求分析,主要完成认识软件定义过程、需求获取的过程与方法、软件需求分析技术、需求规格说明与评审等任务。

项目3:软件设计,主要完成概要设计、详细设计、面向对象设计、软件设计说明书等任务。

项目4:软件编码,主要完成选择程序设计语言、编码风格、程序调试、面向对象程序设计语言等任务。

项目5:软件测试,主要完成软件测试过程、测试用例设计、程序调试、软件测试计划与测试分析报告等任务。

项目6:软件维护,主要完成软件维护过程、软件维护的实施等任务。

三、读者对象

- 高职高专开设软件工程课程的相关专业学生；
- 应用型本科院校开设软件工程课程相关专业学生；
- 广大软件开发爱好者。

全教材由广东交通职业技术学院余明艳、郭庚麒老师负责编写并统稿，多所高职院校的资深教师都参与了本教材的编写工作。

在本教材编写过程中听取了许多教学第一线教师的意见，在此向他们表示感谢。由于时间仓促，教材中难免有疏漏之处，敬请广大读者在使用中提出宝贵意见和建议，以便我们及时更正。

编者
2015年1月

目 录

项目 1 签订软件开发合同及组建项目组	1
任务 1 认识软件	2
1.1.1 软件的定义	2
1.1.2 软件的主要特点及发展	2
1.1.3 软件分类	4
任务 2 认识软件危机	4
1.2.1 软件危机的概念	5
1.2.2 软件危机产生的原因和解决的途径	5
1.2.3 软件工程的定义	6
1.2.4 软件工程的基本目标和原则	6
任务 3 认识软件的生存周期与开发模型	8
1.3.1 软件生存周期的概念	8
1.3.2 软件生存周期的阶段	8
1.3.3 软件开发模型	10
任务 4 组建项目组	15
1.4.1 项目组织管理	15
1.4.2 软件开发组织机构的管理	17
1.4.3 项目质量管理	19
1.4.4 项目风险管理	22
任务 5 起草项目开发合同	24
1.5.1 技术开发合同的概念	25
1.5.2 技术开发合同的种类	25
1.5.3 技术开发合同的一般条款	25
1.5.4 软件开发合同书	26
习题一	27
项目 2 软件需求分析	28
任务 1 认识软件定义过程	29
2.1.1 问题的定义	29
2.1.2 可行性研究	29
2.1.3 需求分析	30

4 软件工程项目教程

任务 2 需求获取的过程与方法	30
2.2.1 需求获取的过程	31
2.2.2 需求获取的方法	31
任务 3 软件需求分析技术	32
2.3.1 结构化分析方法	32
2.3.2 数据流图	33
2.3.3 数据词典	35
2.3.4 典型案例	37
2.3.5 面向对象的需求分析方法	41
2.3.6 面向对象常用的图形工具	44
2.3.7 ATM 系统案例分析	46
任务 4 需求规格说明与评审	50
2.4.1 需求规格说明的主要内容	50
2.4.2 需求分析的评审	51
2.4.3 学生心理动态跟踪评价系统需求规格说明书	52
习题二	55
项目 3 软件设计	56
任务 1 概要设计	57
3.1.1 概要设计的概念与原则	57
3.1.2 概要设计的任务与步骤	59
3.1.3 设计准则	61
3.1.4 面向数据流的设计方法	62
任务 2 详细设计	68
3.2.1 详细设计的任务与原则	69
3.2.2 详细设计的方法	69
任务 3 面向对象设计	76
3.3.1 面向对象设计(OOD)的概念	76
3.3.2 人机交互子系统的设计	77
3.3.3 问题域子系统的设计	78
3.3.4 任务管理部分的设计	79
3.3.5 数据管理部分的设计	80
3.3.6 OOD 的基本准则	81
任务 4 软件设计说明书	81
3.4.1 概要设计说明书	81
3.4.2 详细设计说明书	82
3.4.3 软件设计说明书	82
习题三	87

项目 4 软件编码	89
任务 1 选择程序设计语言	89
4.1.1 程序设计语言概述	89
4.1.2 程序设计语言的特点	91
4.1.3 程序设计语言的选择	93
任务 2 编码风格	94
4.2.1 源程序文档化	94
4.2.2 数据说明	95
4.2.3 语句结构	95
4.2.4 输入和输出	95
4.2.5 程序效率	96
4.2.6 编码标准	96
任务 3 面向对象程序设计语言	97
4.3.1 面向对象程序设计	97
4.3.2 面向对象语言的技术特点	98
4.3.3 面向对象语言的选择	100
4.3.4 用户登录程序的编码实现	101
习题四	107
项目 5 软件测试	108
任务 1 软件测试过程	109
5.1.1 软件测试概述	109
5.1.2 测试方法	111
5.1.3 测试过程	112
任务 2 测试用例设计	116
5.2.1 测试用例	116
5.2.2 白盒子法测试用例设计	116
5.2.3 黑盒子法测试用例设计	119
5.2.4 面向对象测试	120
5.2.5 典型案例	121
任务 3 程序调试	122
5.3.1 程序调试概述	122
5.3.2 调试过程	123
5.3.3 程序调试的步骤	123
5.3.4 程序调试的主要方法	124
任务 4 软件测试计划与测试分析报告	124
5.4.1 软件测试文档	124
5.4.2 软件测试计划	125
5.4.3 测试分析报告	126
习题五	128

项目 6 软件维护	130
任务 1 软件维护过程	130
6.1.1 软件维护的内容	131
6.1.2 软件维护的特点	132
6.1.3 维护的困难性	132
6.1.4 软件维护的费用	133
任务 2 软件维护的实施	134
6.2.1 维护的组织	134
6.2.2 维护的流程	135
6.2.3 维护技术	136
6.2.4 维护的副作用	136
6.2.5 软件可维护性	138
6.2.6 软件再工程	140
6.2.7 系统维护手册	141
习题六	142
附录 计算机软件开发文档编写指南	143
附录 1 可行性研究报告	143
附录 2 项目开发计划	147
附录 3 需求规格说明书	148
附录 4 概要设计说明书	150
附录 5 详细设计说明书	152
附录 6 测试计划	153
附录 7 测试分析报告	155
附录 8 开发进度月报	156
附录 9 项目开发总结报告	156
参考文献	158

项目 1 签订软件开发合同及组建项目组

【教学目标】

1. 了解软件的定义、特点及分类。
2. 了解软件危机的概念、产生的原因和解决的途径。
3. 了解软件生存周期的概念、各阶段的划分和任务。
4. 了解软件开发模型的概念。
5. 学会组建项目组。
6. 学会签订项目开发合同。

【项目简介】

某软件开发公司应某高校需求开发学生心理动态跟踪评价系统,现该公司拟与客户签订软件开发合同,双方拟订的条款如图 1-1 所示,具体内容详见文中的软件开发合同书。

软件开发合同书	
合同编号:	
甲方:	
乙方:	
第一条	合同项目
第二条	软件价格, 付款方法
第三条	软件开发时间
第四条	软件验收标准
第五条	售后服务条款及时间
第六条	乙方运行软件的硬件设备及软件等条件
第七条	软件维护等问题
第八条	软件使用范围、版权等问题
第九条	其他
第十条	合同一式两份, 甲乙双方各持一份
第十一条	合同生效时间
甲方(盖章):	乙方(盖章):
代表(签字):	代表(签字):
签订时间:	
签订地点:	

图 1-1 软件开发合同条款

任务1 认识软件

【任务分析】

通过完成认识软件这个任务,我们可以通过认识软件的定义、特点来认识项目中拟开发系统学生心理动态跟踪评价系统的组成和主要的特点等。

【知识点分析】

认识软件这个任务所涉及知识点包括软件的定义、软件的主要特点及发展、软件的分类等。

1.1.1 软件的定义

自1946年第一台计算机诞生以来,计算机系统经历了电子管、晶体管、集成电路、大规模集成电路、超大规模集成电路等多个不同的发展时期。由于微电子学技术的进步,计算机硬件性价比平均每十年提高二个数量级,而且质量稳步提高。与此同时,正在运行使用着的计算机软件的数量正以惊人的速度急剧膨胀,计算机软件成本逐年上升,软件开发的生产率也远远跟不上计算机飞速发展的需求,软件已经成为限制计算机系统发展的关键因素。

软件是计算机系统中与硬件相互依存的部分,它是包括程序、数据及相关文档的完整集合。其中,程序是按事先设计的功能和性能要求执行的指令序列;数据是程序所处理信息的数据结构;文档是与程序开发、维护和使用相关的各种图文资料,即软件=程序+数据+文档。常见的应用型软件有:文字处理软件、信息管理软件、辅助设计软件、实时控制软件、教育与娱乐软件等。如项目中所提及的学生心理动态跟踪评价系统即属于应用软件,它是包括学生心理动态跟踪评价系统程序、数据及相关技术性文档的完整集合。

1.1.2 软件的主要特点及发展

1. 软件的主要特点

(1) 软件具有抽象性

这是软件与硬件的根本区别。软件是一种逻辑实体,而不是具体的物理实体,它一般寄生在诸如纸、内存储器、磁带、磁盘或光盘等载体上,人们无法观察到它的具体形态,必须通过对它的运行来分析了解它的功能和特征,因此它具有抽象性。

(2) 软件无明显的制造过程

软件的生产与硬件的生产不同,软件的生产没有明显的制造过程。在硬件的制造过程中必须对每一个制造环节进行质量控制,以保证整个硬件产品的质量,并且每一个硬件的生产都几乎付出与样品一样的生产资料成本。而软件将人类的知识和技术转化成产品,软件产品的开发成本几乎全部用在样品的开发设计上,其制造过程则非常简单,人们可以用很低的成本进行软件产品的复制。因此,基于软件的这个特点也随之产生了软件产品的保护问题,这个问题已引起国际上的普遍重视。为了保护软件开发者的根本利益,除国家在法律上

采取有力的措施之外,开发者在技术上也要采取各种措施,防止他人对软件产品的随意复制。因此,对软件的质量控制,必须立足于软件开发方面。

(3) 软件无磨损、老化的问题

在软件的运行和使用期间,没有像硬件那样的磨损、老化问题。任何机械、电子设备在运行和使用的过程中,其失效率大致遵循如图 1-2(硬件失效率曲线)中所示的 U 型曲线(即浴盆曲线)。软件不存在磨损和老化的问题,但它存在退化的问题,设计人员必须多次修改软件,如图 1-3(软件失效率曲线)所示。

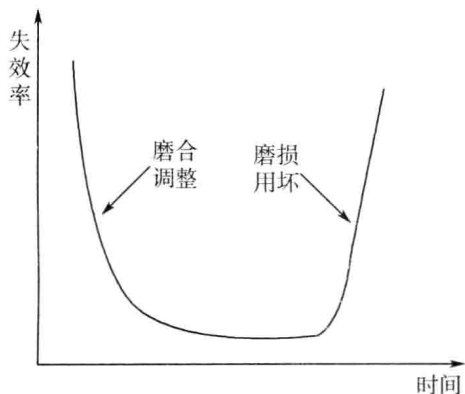


图 1-2 硬件失效率曲线

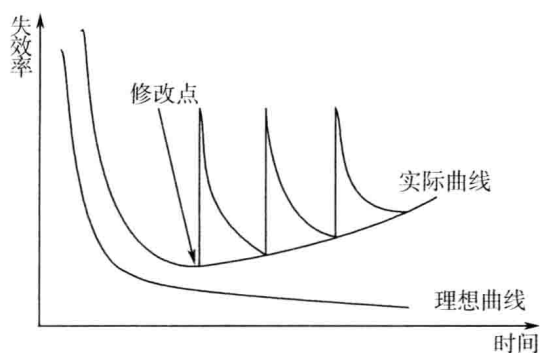


图 1-3 软件失效率曲线

(4) 软件对计算机系统的依赖性

软件的开发和运行往往受到计算机系统的限制,对计算机系统有着不同程度的依赖性。为了减少这种依赖性,在软件开发中提出了软件的可移植问题。

除此之外,软件还具有复杂性、社会性、成本昂贵等特点。如项目中的学生心理动态跟踪评价系统就具备以上所提及的软件的具体特点。

2. 软件的发展

自 20 世纪 40 年代世界上第一台计算机问世以来,软件经历了程序设计、程序系统及软件工程三个阶段的发展。

第一阶段:程序设计阶段(20 世纪 50 年代至 60 年代)。该阶段的软件指程序。程序的开发采用个体工作的方式,开发工作主要依赖于开发人员的个人技能和程序设计的技巧,软件的质量得不到保证,缺乏与程序有关的文档。

第二阶段:程序系统阶段(20 世纪 60 年代至 70 年代)。该阶段的软件包括程序和说明书。软件开发采用开发小组工作的方式,开发工作主要依赖于开发小组的水平,文档资料的不完备性给软件维护带来很大的难度,软件技术的发展不能满足需要。

第三阶段:软件工程阶段(20 世纪 70 年代以后)。该阶段的软件则包括程序、文档和数据。软件开发的任务由开发小组及大中型软件开发机构来承担,开发工作主要依赖于整个机构的管理水平,采用多种开发技术。

目前在很多的应用领域,人们开始采用面向对象的软件开发技术;专家系统、人工智能软件开始走向实际应用。软件技术呈现国际化、网络化、服务化等多种发展趋势。互联网作为 20 世纪最重要的科技成果之一,给人类生活和经济发展都带来了深远的影响,它所展现出的勃勃商机,吸引了众多厂商围绕着它开发形形色色的互联网软件,目前,分布计算、互联

网相关的软件开发成为软件开发产业的热点。此外,自由软件潮流、智能化、简易化、多样化等趋势正极大地拓展软件产业的发展空间,派生出许多具有成长潜力的新兴领域。

1.1.3 软件的分类

尽管人们开发了大量的软件,积累了丰富的软件资源并使之广泛应用于各个领域,但软件的品种、质量和价格方面仍然满足不了人们日益增长的需要。随着软件复杂性和交互性的增加,难以对目前应用的软件进行一个标准化的分类,这里我们简单地介绍其从功能、工作方式、服务范围等方面的划分。

1. 按软件的功能进行划分

可分为系统软件、支撑软件、应用软件等。系统软件是能与计算机硬件系统紧密配合,使计算机系统的各个部件、相关软件和数据协调高效工作的软件。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。如:操作系统、数据库管理程序、设备驱动程序以及通信处理程序等;支撑软件是协助用户开发软件的工具性软件,包括帮助程序人员开发软件产品的工具,也包括帮助管理人员控制开发进程的工具;应用软件是为特定目的服务的软件。拟开发项目学生心理动态跟踪评价系统按软件功能分类属于应用软件。

2. 按软件的工作方式进行划分

可分为实时处理软件、分时软件、交互式软件、批处理软件等。实时处理软件指在事件或数据产生时对其立即处理,并及时反馈信号以控制需要监测的过程的软件。实时处理软件主要包括数据采集、分析、输出3个部分;分时软件允许多个联机用户同时使用计算机;交互式软件是能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。拟开发项目学生心理动态跟踪评价系统软件按工作方式分类属于交互式软件。

3. 按软件服务对象的范围进行划分

可分为项目软件和产品软件。项目软件也称定制软件,是受某个特定客户(或少数客户)的委托,由一个或多个软件开发机构在合同的约束下开发的软件,如军用防空指挥系统、卫星控制系统等;产品软件由软件开发机构开发并直接提供给市场,或是为众多用户服务的软件,如文字处理软件、文本处理软件、财务处理软件、人事管理软件等。项目中拟开发的学生心理动态跟踪评价系统按软件工作方式划分属于交互式软件。

此外,还可以按软件使用的频度、软件的规模、软件失效的影响等方面来进行划分。

任务2 认识软件危机

【任务分析】

通过完成认识软件危机这个任务,我们可以用软件工程化的方法组织软件开发工作,避免在软件开发过程中出现诸多问题。

【知识点分析】

认识软件危机这个任务所涉及的知识点包括软件危机的概念、软件危机产生的原因和解决的途径、软件工程的概念、软件工程的基本目标和原则等。

1.2.1 软件危机的概念

20 世纪 60 年代中期至 20 世纪 70 年代中期，“软件危机”这一词在计算机界广为流传。这个时期的一个重要特征是出现了采用个体化软件开发方法的“软件作坊”，广泛使用产品软件。随着软件数量的急剧膨胀，许多程序的个体化特性使得它们最终成为不可维护的系统。用错误方法开发出来的许多大型软件由于无法维护只好提前报废，造成大量人力、物力的浪费。西方计算机科学家把在软件开发和维护过程中遇到的一系列严重问题统称为“软件危机”。“软件危机”开始出现了！

美国 IBM 公司在 1963~1966 年开发 IBM360 机的操作系统，这一项目花了 5000 人一年的工作量，最多时有 1000 人投入开发工作，定出了近 100 万行源程序。据统计，这个操作系统每次发行的新版本都是从前一版本中找出 1000 个程序错误而修正结果。这个项目的负责人 F. D. Brooks 事后在总结了他在组织开发过程中的沉痛教训时说：“……正像一只逃亡的野兽落到泥潭中做垂死的挣扎，越是挣扎、陷得越深，最后无法逃脱灭顶的灾难……程序设计工作正像这样一个泥潭……一批批程序员被迫在泥潭中拼命挣扎……谁也没有料到问题竟会陷入这样的困难……”IBM 360 操作系统的历史教训成为软件项目开发的经典事例。

1.2.2 软件危机产生的原因和解决的途径

软件危机是指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。产生软件危机的原因主要有以下几点：

1. 软件开发经验的缺乏

由于相关开发人员缺乏软件开发的经验和有关软件开发数据的积累，使得开发工作的计划很难制定，以致经常超出经费预算，进度计划无法遵循，完成开发的期限一再拖延等情况。

2. 软件需求在开发的初期阶段不够明确

由于软件需求在开发的初期阶段不够明确或是未能得到确切的表达，在开发工作开始后，软件人员和用户又未能及时交换意见，造成矛盾在开发后期集中暴露出来。

3. 开发过程没有统一的规范

由于开发过程没有统一、公认的方法论和规范进行指导，参加开发的人员各行其事。另外，设计和实现过程的资料很不完整，或忽视了个人工作与其他人的接口，使得软件很难维护。

4. 没有做好测试阶段的工作

由于未能在测试阶段充分做好检测工作，提交至用户的软件质量差，在运行过程中暴露

出大量的问题。

如果这些障碍不能有效突破,软件的发展是没有出路的。许多计算机软件专家尝试把其他工程领域中行之有效的工程学知识运用到软件开发工作中来。经过不断的实践总结,最后得出结论:按工程化的原则和方法组织软件开发工作是有效的,是解决软件危机的一个重要方法。由此,软件工程成为了计算机科学技术的一个新领域,它从管理和技术两方面研究如何更好地开发和维护计算机软件,有效地缓解了软件危机引发的种种问题。

在20世纪60年代后期人们开始认真研究解决软件危机的方法,从而形成了计算机科学技术领域中的一门新兴的学科——计算机软工工程学,通常简称为软件工程。在以计算机为核心的信息化社会中,信息的获取,处理、交流和决策无一不需要大量高质量的软件,因此软件工程一直是人们研究的焦点。1968年,在联邦德国召开的北大西洋公约组织的国际会议上,计算机科学家们讨论了软件危机的问题。在这次会议上正式提出并使用了“软件工程”这个名词,一门新兴的学科就此诞生了。

1.2.3 软件工程的概念

1. 定义软件工程

软件工程是指应用计算机科学、数学及管理科学等原理,以工程化的原则和方法来解决软件问题,指导计算机软件开发和维护的一门工程学科。

2. 软件工程过程

软件工程过程是为了获得好的软件产品,在软件开发工具的支持下,由软件开发者即软件工程师完成的一系列软件工程活动。软件工程过程通常包含四种基本活动:

(1)P(Plan) 软件规格说明

规定被开发软件的功能及运行时的限制。

(2)D(Do) 软件开发

产生满足规格说明的软件。

(3)C(Check) 软件确认

确认软件能够满足客户提出的要求。

(4)A(Action) 软件演进

为满足客户的变更要求,软件必须在使用的过程中演进。

事实上,软件工程过程是一个软件开发机构针对某类软件产品为自己规定的工作步骤,它应当是科学的、合理的,否则必将影响软件产品的质量。

1.2.4 软件工程的基本目标和原则

软件工程的目的是提高软件生产率,提高软件质量,降低软件成本。组织实施软件项目是为了获得项目的成功,即达到以下几个主要的目标:

(1)付出较低的软件成本。

(2)达到预期的软件功能。

(3)取得较好的软件性能。

- (4)使开发的软件易于移植。
- (5)需要较低的维护费用。
- (6)能按时完成开发工作,及时交付使用。

为了达到这些目标,在软件的开发过程中必须遵循以下的软件工程原则。

1. 抽象

抽取事物最基本的特性和行为,忽略非基本细节。采用分层次抽象,自顶向下、逐层细化的办法控制软件开发过程的复杂性。

2. 信息隐蔽

将模块设计成“黑箱”,实现细节隐藏在模块内部,不让模块的使用者直接访问,这就是所谓信息封装(使用与实现分离)的原则。使用者只能通过模块接口访问模块中封装的数据。

3. 模块化

模块是程序中在逻辑上相对自主的成分,是独立的编程单位,应有良好的接口定义。如C语言程序中的函数过程,C++语言程序中的类。模块化有助于信息隐蔽和抽象,有助于表示复杂的系统。

4. 局部化

在一个物理模块内集中逻辑上相互关联的计算机资源,保证模块之间有松散的耦合,模块内部有较强的内聚,这有助于控制软件的复杂性。

5. 确定性

软件开发过程中所有概念的表达应是确定的、无歧义的、规范的。这样有助于人们在交流时不会产生误解、遗漏,保证整个开发工作的协调一致。

6. 一致性

整个软件系统(包括程序、文档和数据)的各个模块应使用一致的概念、符号和术语;程序内、外部接口应保持一致;软件同硬件、操作系统的接口应保持一致;用于形式化规格说明的公理系统应保持一致。

7. 完备性

软件系统不丢失任何重要成分,可以完全实现系统所要求的功能。为了保证系统的完备性,在软件开发和运行过程中需要严格的技术评审。

8. 可验证性

开发大型的软件时需要对系统自顶向下、逐层分解。系统分解应遵循使系统易于检查、测试、评审的原则,以确保系统的正确性。使用一致性、完备性和可验证性可以帮助开发者设计一个正确的系统。

任务3 认识软件的生存周期与开发模型

【任务分析】

通过完成认识软件的生命周期与开发模型这个任务,我们可以了解软件的开发过程以及可采用的软件的开发模型等,使之可以应用到实际软件的开发中。

【知识点分析】

认识软件的生存周期与开发模型这个任务所涉及知识点包括软件生存周期的概念、软件生存周期各阶段应解决的问题、软件开发的几种主要模型等。

1.3.1 软件生存周期的概念

如同人的一生要经历婴儿期、少年期、青年期、老年期直至死亡这样一个全过程,任何一个软件产品或软件系统也都要经历软件定义、软件开发、软件的使用和维护直至被淘汰这样一个过程,我们把软件这一全过程称之为软件的生存周期。目前对软件生存周期的各阶段的划分尚不统一,有的分得粗些,有的分得细些,无论是哪一种划分方式,都应包括软件定义、软件开发、软件的使用和维护这三个阶段。这里介绍的软件生存周期分为问题定义、可行性研究、需求分析、概要设计、详细设计、编码、测试、运行、维护和退役几个阶段,每个阶段相互独立又彼此有联系,上一阶段的工作结果是下一阶段的工作依据,下一阶段是上一阶段的进化。下面简要介绍一下各个阶段的主要任务和结束标志。

1.3.2 软件生存周期的阶段

1. 软件定义

软件定义又称为系统分析,由系统分析员完成。软件定义阶段通常可细分为三个阶段:问题定义、可行性研究、需求分析,通常将问题定义和可行性研究两个阶段合称为软件项目计划。软件定义的主要任务是确定软件系统必须完成的总目标;确定工程的可行性,分析实现工程目标应采取的技术和软件系统必须完成的功能和性能;估计完成该项工程所需资源和成本,给出开发工程的进度表等。

(1) 问题定义

这个阶段要回答的关键问题:“要解决的问题是什么?”。通过问题定义阶段的工作,系统分析员应提出关于问题的性质、软件系统的目标和规模的书面报告。通过对系统的实际用户和使用部门负责人的访问调查,分析员要扼要地写出他对问题的理解,并在用户和使用部门负责人的会议上认真讨论这份书面报告,澄清含糊不清的地方,改正理解不正确的地方,最后得出一份双方都满意的文档。

(2) 可行性研究

这个阶段要回答的关键问题:“对于上一个阶段所确定的问题有行得通的解决办法吗?”。这个阶段的任务不是具体地解决问题,而是研究问题的范围,探索这个问题是否值得