



高职高专计算机技能型紧缺人才培养规划教材

计算机软件技术专业

软件测试技术

佟伟光 主编

免费提供

教学相关资料

 人民邮电出版社
POSTS & TELECOM PRESS

高职高专计算机技能型紧缺人才培养规划教材
计算机软件技术专业

软件测试技术

佟伟光 主编 ←

人民邮电出版社

图书在版编目 (CIP) 数据

软件测试技术 / 佟伟光主编. —北京: 人民邮电出版社, 2005.7
ISBN 7-115-13321-2

I. 软... II. 佟... III. 软件—测试 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2005) 第 075782 号

内 容 提 要

本书共有 8 章。主要内容包括: 软件测试的基本概念和基础知识; 软件测试的基本技术和测试用例设计的相关知识; 软件测试过程; 如何报告发现的软件缺陷, 及测试评价等技术; 测试项目管理, 包括测试文档、制定测试计划、测试的组织与人员管理, 以及测试过程的控制和管理等内容; 软件测试自动化和软件测试工具等相关知识。最后一章是一个完整的实际软件项目的测试示例, 给出了软件测试项目从制定测试计划、测试实施、测试实现, 到报告软件缺陷和测试评测的全过程, 并介绍了在该项目测试过程中采用的主要技术。

本书为高职高专计算机专业的教材, 也可作为软件测试人员的参考书。

高职高专计算机技能型紧缺人才培养规划教材

计算机软件技术专业

软件测试技术

-
- ◆ 主 编 佟伟光
责任编辑 潘春燕
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 14
字数: 329 千字 2005 年 7 月第 1 版
印数: 3 001 - 4 000 册 2006 年 7 月北京第 2 次印刷

ISBN 7-115-13321-2/TP · 4619

定价: 20.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

高职高专计算机技能型紧缺人才培养

规划教材编委会

主 任 武马群

副主任 王泰峰 徐民鹰 王晓丹

编 委 (以姓氏笔画为序)

马 伟 安志远 向 伟 刘 兵 吴卫祖 吴宏雷
余明辉 张晓蕾 张基宏 贺 平 柳 青 赵英杰
施晓秋 姜 锐 耿 壮 郭 勇 曹 炜 蒋方纯
潘春燕

丛书出版前言

目前,人才问题是制约我国软件产业发展的关键。为加大软件人才培养力度和提高软件人才培养质量,教育部继在2003年确定北京信息职业技术学院等35所高职院校试办示范性软件职业技术学院后,又同时根据《教育部等六部门关于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的通知》(教职成[2003]5号)的要求,组织制定了《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》。示范性软件职业技术学院与计算机应用与软件技术专业领域技能型紧缺人才培养工作,均要求在较短的时间内培养出符合企业需要、具有核心技能的软件技术人才,因此,对目前高等职业教育的办学模式和人才培养方案等做较大的改进和全新的探索已经成为学校的当务之急。

据此,我们认为做一套符合上述一系列要求的切合学校实际的教学方案尤为重要。遵照教育部提出的以就业为导向,高等职业教育从专业本位向职业岗位和就业为本转变的指导思想,根据目前高等职业教育院校日益重视学生将来的就业岗位,注重培养毕业生的职业能力的现状,我们联合北京信息职业技术学院等几十所高职院校和普拉内特计算机技术(北京)有限公司、福建星网锐捷网络有限公司、北京索浪计算机有限公司等软件企业共同组建了计算机应用与软件技术专业领域技能型紧缺人才培养教学方案研究小组(以下简称研究小组)。研究小组对承担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79所院校的专业设置情况做了细致的调研,并调查了几十所高职院校计算机相关专业的学生就业情况以及目前软件企业的人才市场需求状况,确定首批开发目前在高职院校开设比较普遍的计算机软件技术、计算机网络技术、计算机多媒体技术和计算机应用技术等4个专业方向的教学方案。

同时,为贯彻教育部提出的要与软件企业合作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精神,使高等职业教育培养出的软件技术人才符合企业的需求,研究小组与许多软件企业的专家们进行了反复研讨,了解到目前高职院校的毕业生的实际动手能力和综合应用知识方面较弱,他们和企业需求的软件人才有着较大的差距,到企业后不能很快独当一面,企业需要投入一定的成本和时间进行项目培训。针对这种情况,研究小组在教学方案中增加了“综合项目实训”模块,以求强化学生的实际动手能力和综合应用前期所学知识的能力,探索将企业的岗前培训内容前移到学校的教学中的实验之路,以此增强毕业生的就业竞争力。

在上述工作的基础上,研究小组于2004年多次组织召开了包括企业专家、教育专家、学校任课教师在内的各种研讨会和方案论证会,对各个专业按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”一步步进行了认真的分析和研讨:

- 列出各专业的岗位群及核心技能。针对教育部提出的以就业为导向,根据目前高职高专院校日益关心学生将来的就业岗位的现状,在前期大量调研的基础上,首先提炼各个专业的岗位群。如对某专业的岗位群进行研究时,首先罗列此专业的各个岗位,以便能正确了解

每个岗位的职业能力，再根据职业能力进行有意义的合并，形成各个专业的岗位群，再对每个岗位群总结和归纳出其核心技能。

- 根据岗位群及核心技能做出教学方案。在岗位群及核心技能明确的前提下，列出此岗位应该掌握的知识点，再依据这些知识点推出应该学习的课程、学时数、课程之间的联系、开课顺序并进行必要的整合，最终形成一套科学完整的教学方案。

为配合学校对技能型紧缺人才的培养工作，在研究小组开发上述4个专业的教学方案的基础上，我们组织编写了这套包含计算机软件技术、计算机网络技术、计算机多媒体技术及计算机应用技术等4个专业的教材。本套教材具有以下特点：

- 注重专业整体策划的内涵。对各专业系列教材按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”的思路组织开发教材。

- 按照“理论够用为度”的原则，对各个专业的基础课进行了按需重新整合。

- 各专业教材突出了实训的比例，注重案例教学。每本教材都配备了实验、实训的内容，部分专业的教材配备了综合项目实训，使学生通过模拟具体的软件开发项目了解软件企业的运行环境，体验软件的规范化、标准化、专业化和规模化的开发流程。

为了方便教学，我们免费为选用本套教材的老师提供部分专业的整体教学方案及教学相关资料。

- 所有教材的电子教案。
- 部分教材的习题答案。
- 部分教材中实例制作过程中用到的素材。
- 部分教材中实例的制作效果以及一些源程序代码。

本套教材以各个专业的岗位群为出发点，注重专业整体策划，试图通过对系列教材的整体构架，探索一条培养技能型紧缺人才的有效途径。

经过近两年的艰苦探索和工作，本套教材终于正式出版了，我们衷心希望，各位关心高等职业教育的读者能够对本套教材的不当之处给予批评指正，提出修改意见，也热切盼望从事高等职业教育的教师以及软件企业的技术专家和我们联系，共同探讨计算机应用与软件技术专业的教学方案和教材编写等相关问题。来信请发至 panchunyan@ptpress.com.cn。

编者的话

随着软件的规模及复杂性的日益增加，软件质量已成为越来越重要的问题。软件测试是保证软件质量的重要手段，对于专业化、高效的软件测试的要求也越来越严格。

软件测试是一项需要专业技能的工作，它包含许多理论和实践。缺少这些知识和经验，测试的深度和广度就不够，测试的质量也就无法保证，因此，它需要专门的培训和实践。随着我国软件产业的不断发展和规范化，对软件测试人员的需求迅速地增长，软件测试技术提供了令人感兴趣且富有挑战性的工作任务和职业。为适应软件产业发展的需要，各高等职业院校的计算机软件相关专业都相继开设了软件测试等课程。为了满足教学的需要，我们联合国内大型软件公司的软件项目负责人及软件测试工程师一起合作，共同编写了此教材。

本教材内容丰富，涵盖了软件测试的各项基本技术和知识。在编写教材时，注意教材内容的先进性，将软件测试的新概念、新技术、新方法编入教材中。在内容的安排上注意由易到难、深入浅出、简明扼要、通俗易懂，使得学生能够较好地掌握软件测试的基本知识和基本技术。

本教材的编写特别注重突出应用性、实践性，理论联系实际，把学生应用能力培养融汇于教材之中。教材提供了丰富的实例和习题，并专门编写了测试实践一章，在该章给出了一个完整的实际软件项目的测试示例，将有助于提高学生软件测试技术的应用能力。

本书由佟伟光、郑秀影、宋喜莲、赵忠诚、张伟共同编写，全书由佟伟光修改定稿。李卓铃教授对本书的大纲及书稿做了全面、仔细的审定，提出了宝贵的修改意见，在此表示衷心的感谢。

由于编者水平有限，加之时间仓促，书中难免存在错误和不妥之处，请读者不吝指正。

编者
2005年6月

目 录

第 1 章	软件工程与软件测试	1
1.1	软件工程	1
1.1.1	软件工程的目標及其一般开发过程	1
1.1.2	软件过程模型	3
1.2	软件质量	5
1.2.1	质量与质量模型	5
1.2.2	软件质量保证	7
1.2.3	ISO 9000 和 CMM	9
1.3	软件测试	10
1.3.1	软件测试的定义及目的	11
1.3.2	软件测试信息流	12
1.3.3	软件测试与软件开发过程的关系	13
1.3.4	软件测试与质量保证的区别	14
1.3.5	软件测试的发展历程及趋势	15
1.4	软件测试人员的基本素质	16
	习题	17
第 2 章	软件测试的基本知识	18
2.1	软件测试贯穿于整个的软件开发生命周期	18
2.1.1	软件测试中使用的各种术语	18
2.1.2	软件测试贯穿于整个的软件开发生命周期	19
2.1.3	软件测试的手段	21
2.2	测试模型	23
2.3	软件测试的分类	24
2.4	软件测试的原则	26
2.5	软件测试策略	27
2.6	软件测试流程	30
2.6.1	制定测试计划	31
2.6.2	设计测试	32
2.6.3	实施测试	34
2.6.4	执行测试	41
2.6.5	评估测试	42
2.7	测试的成功经验	43

习题.....	45
第 3 章 软件测试的方法和技术	46
3.1 软件测试方法概述.....	46
3.2 白盒测试.....	47
3.2.1 程序插桩.....	48
3.2.2 逻辑覆盖.....	51
3.2.3 基本路径测试.....	56
3.2.4 程序的静态测试.....	58
3.2.5 其他白盒测试方法简介.....	61
3.3 黑盒测试.....	63
3.3.1 等价类划分法.....	64
3.3.2 边界值分析法.....	65
3.3.3 错误推测法.....	65
3.3.4 因果图法.....	66
3.3.5 场景法.....	67
3.4 测试用例设计.....	72
3.4.1 测试用例的基本概念.....	72
3.4.2 测试用例的设计步骤.....	73
3.4.3 测试用例的编写.....	76
3.4.4 测试用例设计实例.....	79
3.4.5 测试用例的管理.....	83
习题.....	84
第 4 章 软件测试过程	86
4.1 软件测试过程概述.....	86
4.2 单元测试.....	87
4.2.1 单元测试的主要任务.....	87
4.2.2 单元测试的执行过程.....	89
4.2.3 单元测试技术和测试数据.....	90
4.2.4 单元测试人员.....	90
4.3 集成测试.....	91
4.3.1 集成测试的主要任务.....	91
4.3.2 集成测试方法.....	91
4.3.3 集成测试技术和测试数据.....	94
4.3.4 集成测试遵循的原则.....	95
4.3.5 集成测试人员.....	95
4.4 系统测试.....	95
4.4.1 系统测试的任务.....	96

4.4.2	系统测试技术和测试数据	96
4.4.3	系统测试人员	97
4.5	验收测试	97
4.5.1	验收测试的主要任务	97
4.5.2	验收测试技术和测试数据	98
4.5.3	验收测试人员	99
4.5.4	α 、 β 测试	99
4.6	回归测试	99
4.6.1	回归测试技术和测试数据	100
4.6.2	回归测试的范围	101
4.6.3	回归测试人员	101
4.7	系统排错	101
	习题	103
第 5 章	软件测试报告与测试评价	104
5.1	软件缺陷的概念和种类	104
5.2	正确面对软件缺陷	107
5.3	软件缺陷的生命周期	108
5.4	软件缺陷的严重性和优先级	109
5.5	报告软件缺陷	110
5.5.1	报告软件缺陷的基本原则	110
5.5.2	IEEE 软件缺陷报告模板	112
5.5.3	软件缺陷数据库跟踪系统	114
5.5.4	手工报告和跟踪软件缺陷	115
5.6	分离和再现软件缺陷	116
5.7	测试总结报告	117
5.8	测试的评测	118
5.8.1	覆盖评测	119
5.8.2	质量评测	121
5.8.3	性能评测	124
	习题	125
第 6 章	软件测试项目管理	127
6.1	测试项目管理概述	127
6.1.1	测试项目与测试项目管理	127
6.1.2	测试项目的范围管理	128
6.2	测试文档	129
6.2.1	测试文档的作用	129
6.2.2	测试文档的类型	130

6.2.3	主要软件测试文档	131
6.3	软件测试计划	135
6.3.1	制定测试计划的目的	135
6.3.2	制定测试计划的原则	136
6.3.3	制定测试计划时面对的问题	136
6.3.4	制定测试计划	137
6.4	测试的组织与人员管理	143
6.4.1	测试的组织与人员管理概述	143
6.4.2	测试人员的组织结构	144
6.4.3	测试人员	146
6.4.4	人员的通信方式	146
6.4.5	测试人员管理的激励机制	146
6.4.6	测试人员的培训	147
6.4.7	测试的组织与人员管理中的风险管理	148
6.5	软件测试过程管理	148
6.5.1	测试项目的跟踪与监控	148
6.5.2	测试的配置管理	149
6.5.3	软件测试风险管理	150
6.5.4	软件测试的成本管理	153
	习题	157
第7章	软件测试自动化与软件测试工具	158
7.1	软件自动化测试基础	158
7.2	自动化测试的作用和优势	159
7.3	软件测试工具分类	165
7.4	几种常用软件测试工具	168
	习题	176
第8章	软件测试实例	177
8.1	被测试软件项目介绍	177
8.1.1	软件背景	177
8.1.2	门诊挂号管理子系统介绍	178
8.1.3	门诊挂号管理子系统的功能需求分析	180
8.1.4	门诊挂号管理子系统性能及可用性要求	182
8.2	HIS 测试过程概述	183
8.2.1	单元测试	183
8.2.2	集成测试	184
8.2.3	系统测试	184
8.2.4	验收测试	185

8.3 测试计划	185
8.3.1 概述	185
8.3.2 定义	185
8.3.3 质量风险摘要	186
8.3.4 测试进度计划	186
8.3.5 进入标准	187
8.3.6 退出标准	187
8.3.7 测试配置和环境	187
8.3.8 测试开发	187
8.3.9 关键参与者	187
8.3.10 预算	188
8.3.11 参考文档	188
8.4 测试用例	188
8.4.1 挂号管理子系统测试大纲	189
8.4.2 其他可用性测试检查标准	190
8.4.3 功能测试用例	191
8.4.4 性能测试用例	198
8.5 缺陷报告	199
8.5.1 建立缺陷报告数据库	199
8.5.2 编写缺陷报告	200
8.6 测试结果总结分析	201
8.6.1 测试总结报告	201
8.6.2 测试用例分析	202
8.6.3 软件测试结果统计分析	202
8.7 应用测试工具	206
8.8 文档测试	207
习题	209
参考文献	210

什么是软件工程？严格地说，软件工程是应用计算机科学、数学及管理科学等原理开发软件的工程。通俗地说，软件工程是实现一个大型程序的一套原则方法，是指将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，即按工程化的原则和方法组织软件开发工作。

软件测试是软件工程的一个重要环节，相当于工程领域中的质量检验部分，是确保软件工程质量的重要手段。近几年来，由于软件工程的复杂度不断增强，以及软件的工业化发展趋势，软件测试受到了广泛的重视。

1.1 软 件 工 程

1.1.1 软件工程的目标及其一般开发过程

狭义上说，软件工程的目标在于生产出满足预算、按期交付、用户满意的无缺陷的软件，进而当用户需求改变时，所生产的软件必须易于修改。广义上说，软件工程的目标就是提高软件的质量与生产率，最终实现软件的工业化生产。

为了很好地理解软件工程，可以将软件工程分解为人员（People）、过程（Process）、项目（Project）和产品（Product）4个部分，即所谓的4P问题。

① 人员是关于软件工程由谁来完成的问题。

② 过程指明了软件工程的实现方式，一般包括可行性与需求分析、系统设计、程序设计、测试和维护。

③ 项目是指软件工程的实现过程，它包括和客户的交流、撰写文档、设计、编码和对产品做测试等一系列流程。

④ 产品指所实现的应用软件及其相关的应用工件，所谓应用工件包括需求分析所得到的软件需求说明书、详细设计的最终结果（设计模型）、具体实现的源码和目标码以及测试过程和测试用例。

这4P之间的关系可表达为：每个项目的目标是产生一个软件产品，设计和实现一个有效产品就是一个过程，而项目小组中人员的组织、协调、管理，以及人与人之间的信息交流则是影响项目成功与否的关键因素。

4P基本上给出了软件工程的一个大轮廓。从事软件开发工作的人员，考虑更多的是开发活动。过去，人们常把软件开发简单理解为“编程序”。实际上，软件是指程序以及开发、使用和维护程序所需的所有文档，在软件开发过程中，及时地按一定准则产生各种文档是软件

开发的重要组成部分，必须充分重视。因此，软件开发已不仅仅是“编程序”了，它的含义已经扩充，因此有必要介绍“软件生存周期”这一概念。

一个软件产品从形成概念开始，经过开发、使用和维护，直到最后退出使用的全过程称为软件生存周期。软件生存周期根据软件所处的状态，以及软件开发活动的目的和任务，可划分为若干个阶段。一般软件生存周期包括软件定义、软件开发以及软件使用与维护 3 个部分。

1. 软件定义

软件定义可分为软件系统的可行性分析和需求分析两个阶段，其基本任务是确定软件系统的工程需求。

可行性分析的任务是了解用户的要求及实现环境，从技术、经济和社会等几个方面研究并论证软件系统的可行性。参与软件系统开发的人员应在用户的配合下对用户的要求及系统的实现环境做详细的调查，并在调查研究的基础上撰写调查报告，然后根据调查报告及其他相关资料进行可行性论证。可行性论证一般包括技术可行性、操作可行性和经济可行性 3 个部分的论证。在可行性论证的基础上制定初步的项目开发计划，大致确定软件系统开发所用的时间、资金和人力。

需求分析的任务是确定所要开发软件的功能需求、性能需求和运行环境约束，编制软件需求规格说明、软件系统的确认测试准则。软件的功能需求应给出软件必须完成的功能。软件的性能需求包括软件的适应性、安全性、可靠性、可维护性错误处理等。软件系统在运行环境方面的约束指所开发的软件系统必须满足的运行环境方面的要求。软件需求不仅是软件开发的依据，也是软件验收的标准。因此，确定软件需求是软件开发的关键和难点，通常需要开发人员与用户反复沟通、讨论，在需求分析阶段完成需求分析工作是一项十分艰巨的任务。

2. 软件开发

软件开发是按照需求规格说明的要求，由抽象到具体，逐步生成软件的过程。软件开发一般由设计、实现和测试等阶段组成。其中设计可分为概要设计和详细设计，主要是根据软件需求规格说明建立软件系统的结构，算法、数据结构和各程序模块之间的接口信息，规定设计约束，为编写源代码提供必要的说明，并制定组装测试计划。实现也称为编码。软件测试是为了发现程序中的错误。大量统计表明，软件测试的工作量往往占软件开发总工作量的 40% 以上，有时甚至可高达软件工程其他步骤工作量总和的 3~5 倍。

3. 软件使用和维护

软件的使用是在软件通过测试后，将软件安装在用户确定的运行环境中移交给用户使用。软件的维护是对软件系统进行修改或对软件需求变化做出反应的过程。当发现软件中的潜伏错误，或用户对软件需求进行修改，或软件运行环境发生变化时都需要对软件进行维护。软件的维护直接影响软件的应用和软件的生存期。软件的可维护性是与设计密切相关的，因此软件在开发过程中应该重视对软件可维护性的支持。

软件生存周期的最后一个阶段是终止对软件系统的支持，停止软件的使用。

在 20 世纪 80 年代以后，分布式系统、计算机网络、嵌入式计算机系统有了很大的发展，特别是 Internet 的高速发展，使得各种应用软件的数量呈爆炸性增长。另外，在很多应用领域，人们开始采用面向对象的设计技术，人工智能软件开始走向实际应用，多媒体技术将计算机的图形、图像、文字、声音处理集成在一起，提高了计算机的信息处理能力，扩大了计算机的应用领域。尽管显现了这些新的变化，但构建一个软件系统所采用的上述基本活动还

是保持不变的。

1.1.2 软件过程模型

软件开发过程中存在各种复杂因素，为了解决由此而带来的种种问题，软件开发者们经过多年的摸索，给出了多种实现软件工程的方式——软件过程模型，如瀑布过程模型、螺旋过程模型和增量过程模型等。这些软件过程模型是软件开发的指导思想和全局性框架，它的提出和发展反映了人们对软件过程的某种认识观，体现了人们对软件过程认识的提高和飞跃。软件开发所遵循的软件过程是保证高质量的软件开发的一个至关重要的因素。

1. 瀑布过程模型

瀑布过程模型反映了人们早期对软件工程的认识水平，是人们所熟悉的一种线性思维的体现。

瀑布过程模型强调阶段的划分及其顺序性、各阶段工作及其文档的完备性，是一种严格线性的、按阶段顺序的、逐步细化的开发模式，如图 1-1 所示。

该模型主要由顺序的活动（或者阶段）组成，其开发阶段包括计划、需求分析、设计、编码、测试和运行维护等。

① 计划阶段是确定软件开发的总目标；给出软件的功能、性能、可靠性以及接口等方面的设想。研究完成该项软件任务的可行性，探讨问题解决的方案；对可供开发使用的资源（如计算机硬件、软件、人力等）、成本、可取得的效益和开发的进度做出估计；制定完成开发任务的实施计划。

② 需求分析是指收集产品的需求，对开发的软件进行详细的定义，由软件开发人员和用户共同讨论决定，确定哪些需求是可以满足的，并且给予确切的描述；写出软件需求说明书等文档，提交管理机构审查。

③ 设计是软件过程的技术核心。在设计阶段应把已确定的各项需求，转换成相应的体系结构，在结构中，每一组成部分是功能明确的模块，每个模块都能体现相应的需求。这一步称为总体设计。然后进行详细设计，即对某个模块要完成的工作进行具体的描述，以便为程序编写打下基础。

④ 编码阶段是编写各种层次的代码，包括由高级的可视化开发系统生成的代码和用第 4 代语言编写的代码等。

⑤ 软件测试是为了发现程序中的错误而执行的过程。

⑥ 运行维护主要是对应用进行修复和改动，使它能够持续发挥作用。

实际上，图 1-1 所示各阶段并不是严格按照顺序执行的，因为过程中的各部分之间都有某种程度的重叠。造成这种重叠的原因是，绝大多数应用系统都是复杂的，非线性的，上述任何一个阶段都不可能在下一阶段开始之前完全结束。我们很少使用单纯的瀑布过程模型，除非是很小的项目或者是开发的产品与自己以前做过的项目类似。

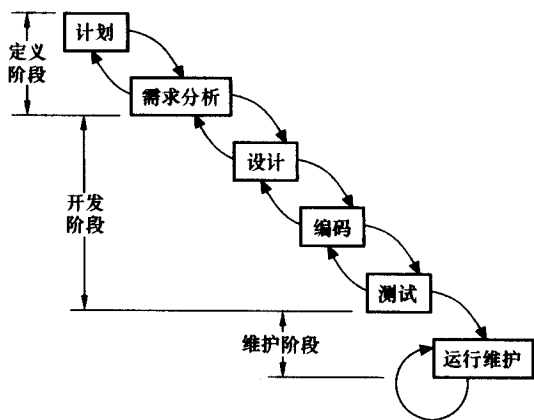


图 1-1 瀑布过程模型

2. 螺旋过程模型

螺旋过程模型需要经历多次需求分析、设计、实现、测试。这样做有多种原因。其中主要的原因是基于规避风险的需要；另一个原因是需要在早期构造产品的局部版本时即交给客户以获得反馈；最后是为了避免像瀑布过程模型一样一次集成大量的代码。

螺旋过程模型的基本思路是，依据前一个版本的结果构造新的版本，这个不断重复迭代的过程形成了一个螺旋上升的路径，如图 1-2 所示。

螺旋过程模型的一个主要优点是，能够在每次迭代中都收集到过程中产生的各种度量数据。例如，在第一次迭代时记录下小组进行设计和实现所花费的时间，依此即可改进后续设计，实现耗费时间的估计。这对于没有任何历史数据的开发组织尤其有用。

螺旋过程模型符合典型项目的发展特点，但是跟简单的瀑布过程模型相比，它需要投入更多的精力来更细致地管理其过程。一个最直接的原因是，每次迭代完成之后都必须保证文档的一致性。特别是代码应该实现文档中描述的设计并且满足文档中记录的需求。此外为了提高小组的生产效率，往往会在前一个迭代结束之前就开始新的一次迭代，这为协调文档的一致性增加了难度。

螺旋开发过程需要多少次迭代取决于具体情况。一个典型的工作量为 3 人/月、耗时 4 个月的项目大概需要两次或者 3 次迭代。项目若采用 5 次迭代，则所需要的管理费用通常会耗掉新增迭代所创造的价值。

当迭代的速度加快，每次迭代只是在前一次的基础上增加少量功能的时候，螺旋过程模型就演变为增量过程模型。

3. 增量过程模型

有些时候可能会用一种几乎连续的过程小幅度地推进项目，这就是增量过程模型，如图 1-3 所示。这种过程模型在项目的后期尤其适用，例如，当项目处于维护阶段，或者立项的

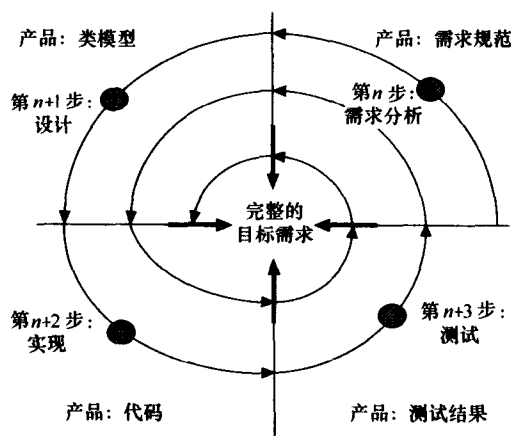


图 1-2 螺旋过程模型

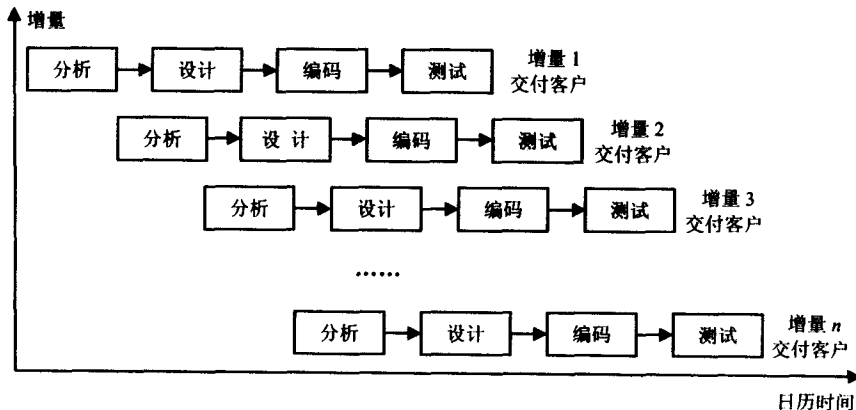


图 1-3 增量过程模型

产品与原先开发出来的产品结构极为相似的时候。

以上3种只是众多软件过程模型中较为典型的，除此之外还有快速原型法、喷泉模型、统一软件开发过程模型等。

软件机构形成一套完整而成熟的软件过程不是一蹴而就的，它需要一个从无序到有序、从特殊到一般、从定性到定量、从静态到动态的历程。或者说，软件机构在形成成熟的软件过程之前，必须经历一系列的成熟阶段。因此有必要建立一个软件能力成熟度模型来对过程做出一个客观、公正的评价，以促进软件开发组织改进软件过程。

1.2 软件质量

软件质量是软件的生命，它直接影响软件的使用与维护。软件开发人员和用户都十分重视软件的质量问题。因此，软件质量问题一直都是软件工程的核心问题。什么是软件质量？现在还没有一个准确的定义。随着计算机软件、硬件技术的发展，人们对软件质量的理解不断深化，软件质量的标准也在不断改变。通常软件质量由以下几方面进行评价。

- ① 软件需求是衡量软件质量的基础，不符合需求的软件就不具备质量。设计的软件应在功能、性能等方面都符合要求，并能可靠地运行。
- ② 软件结构良好，易读、易于理解，并易于修改、维护。
- ③ 软件系统具有友好的用户界面，便于用户使用。
- ④ 软件生存周期中各阶段文档齐全、规范，便于配置、管理。

当前，软件的开发准则都已经有了规范化的标准，用以指导软件开发人员用工程化的方法来开发软件。在整个软件生存周期中应按照标准的开发准则进行软件开发。2001年，软件“产品质量”国际标准 ISO 9126 定义的软件质量包括“内部质量”、“外部质量”和“使用质量”3部分。也就是说，“软件满足规定或潜在用户需求的能力”要从软件在内部、外部和使用中的表现来衡量。所谓内部质量，是从内部观点出发的软件产品特性的总体。内部质量是针对内部质量需求而测量和评价的质量。而外部质量是从外部观点出发的软件产品特性的总体。它是当软件执行时，在模拟环境中，用模拟数据测试时所测量和评价的质量。使用质量是从用户观点出发来看待软件产品用于特定环境和条件下的质量。它测量用户在特定环境中达到其任务目标的程度，而不是测量软件自身的性质。

软件质量是各种因素的复杂组合，它随着应用的不同而不同，随着用户提出的质量要求不同而不同。我们所要了解的是影响软件质量的因素都有哪些，评价软件质量的标准是怎样的。换句话说，满足哪些标准才能保证软件产品具有好的质量。

1.2.1 质量与质量模型

对于没有开发经验的人来说，似乎使自己开发的软件能够正确运行就可以了。但软件工程的结果是软件产品，一个产品的质量要求是多方面的，能正确实现产品所要求的功能只是其存在的基本因素。因为对于软件产品来说，“运行正确”的软件不见得就是高质量的软件。这个软件也许运行速度很低并且浪费内存；也许无论是结构还是代码都写得一塌糊涂，除了开发者本人谁也看不懂，难以进行诊断和修改。

实际的情况是，正确性只是反映软件质量的一个因素而已。软件的质量因素很多，如正