



普通高等教育“十一五”国家级规划教材
高等学校计算机系列

软件测试

佟伟光 主编



人民邮电出版社
POSTS & TELECOM PRESS

TP311.5/231

2008

普通高等教育“十一五”国家级规划教材

高等学校计算机系列

软件测试

佟伟光 主编

 人民邮电出版社
POSTS & TELECOM PRESS

北 京

图书在版编目 (CIP) 数据

软件测试 / 佟伟光主编. —北京: 人民邮电出版社, 2008.5
普通高等教育“十一五”国家级规划教材. 高等学校计算机系列
ISBN 978-7-115-17775-9

I. 软… II. 佟… III. 软件—测试—高等学校—教材
IV. TP311.5

中国版本图书馆 CIP 数据核字 (2008) 第 030072 号

内 容 提 要

本书为普通高等教育“十一五”国家级规划教材。全书共 10 章, 第 1~9 章系统地介绍软件测试的基本概念和基本知识, 详细讲解软件测试的基本技术、测试过程、测试用例设计、测试文档、测试工具等内容, 并从软件工程的角度介绍了如何规划测试工作, 如何报告软件缺陷, 如何评估测试和测试项目管理等内容, 第 10 章通过一个实际软件项目的测试案例, 来加深学生对软件测试技术和软件测试过程的理解, 尽力使理论的阐述更清晰、更形象。

本书内容全面、深入浅出, 理论简明, 实用性强, 利于学生较好地学习和掌握软件测试的基本知识和基本技术。

本书为高等学校计算机相关专业软件测试课程的教材, 也可作为软件测试培训班的教材和软件测试技术人员的自学参考书。

普通高等教育“十一五”国家级规划教材

高等学校计算机系列

软 件 测 试

◆ 主 编 佟伟光

策划编辑 潘春燕

责任编辑 李 凯

◆ 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>

北京铭成印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 17.5

字数: 424 千字

2008 年 5 月第 1 版

印数: 1—3 000 册

2008 年 5 月北京第 1 次印刷

ISBN 978-7-115-17775-9/TP

定价: 29.00 元

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

反盗版热线: (010)67171154

前 言

随着软件规模的不断增大和软件复杂性的日益增加，如何保证软件质量已成为软件开发过程中越来越重要的问题。软件测试是保证软件质量的重要手段，近年来，软件测试工作受到人们越来越多的重视，软件行业对进行专业化、高效率软件测试的要求也越来越严格。

软件测试是一项专业性较强的工作，它包含许多理论知识并要求软件测试人员具有一定的工程实践经验，缺少这些知识和经验，测试的深度和广度就不够，测试工作的质量也就无法保证。因此，软件测试人员需要接受专门的培训并在实践中不断积累经验。随着我国软件产业的不断发展，社会对软件测试人员的需求迅速增长。为适应软件产业发展的需要，各高等院校的计算机软件相关专业都相继开设了软件测试课程。为了满足教学的需要，我们组织高等院校教师、国内大型软件公司的项目负责人、软件测试工程师共同编写了这本教材。在教材编写过程中，我们将工程师丰富的软件测试经验和教师多年的授课经验及体会融入其中，增加了教材的专业性与实用性。

本教材内容全面，涵盖了软件测试的各项基本技术和知识。在编写教材时，我们注意保持教材内容的先进性，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，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。

本教材在编写过程中，特别注重突出教材的应用性、实践性，理论联系实际，把对学生应用能力的培养融汇于教材之中。本教材提供了丰富的实例和习题，并在最后一章重点介绍了一个完整的实际软件项目的测试案例，这些将有助于提高学生软件测试技术的应用能力。

本教材在内容组织结构方面作了精心安排，第1章较详细地介绍软件测试的基本概念和基础知识；第2章详细地介绍如何制订测试计划；第3章讲述软件测试的基本技术；第4章详细介绍软件测试过程；第5章讲述测试用例设计的相关知识和设计实例；第6章详细讲述如何报告发现的软件缺陷，并介绍测试评估等

技术；第7章介绍测试项目管理，包括测试文档、测试的组织与人员管理、测试风险管理、测试成本管理以及测试过程的控制和管理等内容；第8章介绍面向对象软件测试技术；第9章讲述软件测试自动化和软件测试工具等相关知识；第10章是一个完整的实际软件项目的测试案例，详细介绍软件测试项目从制订测试计划、测试实施、测试实现直到报告软件缺陷和测试评测的全过程，并介绍在该项目测试过程中采用的主要技术。

本书为普通高等教育“十一五”国家级规划教材，适合作为高等学校计算机相关专业“软件测试”课程的教材，也可作为软件测试培训机构的教材和软件测试技术人员的自学参考书。

本书由佟伟光主编，参加本书编写的还有栾好利、谢爽爽、宋喜莲、赵忠诚、张欣、杨胜等。李卓铃教授对本书的编写大纲和初稿做了全面、仔细的审定，提出了宝贵的修改意见，谨此表示衷心的感谢。

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

编 者
2008年3月

目 录

第 1 章 软件测试概述	1
1.1 软件、软件危机和软件工程	1
1.1.1 软件、软件危机和软件工程的基本概念	1
1.1.2 软件工程的目标及其一般开发过程	3
1.1.3 软件过程模型	4
1.2 软件缺陷与软件故障	8
1.3 软件质量与质量模型	10
1.4 软件测试	13
1.4.1 软件测试的概念	13
1.4.2 软件测试的原则	15
1.4.3 软件测试过程模型	16
1.4.4 软件测试的分类	19
1.4.5 软件测试流程	21
1.4.6 软件测试发展历程和发展趋势	25
1.5 软件测试人员的基本素质	26
习题 1	27
第 2 章 软件测试计划	28
2.1 软件测试计划的作用	28
2.2 制订测试计划的原则	30
2.3 如何制订软件测试计划	30
2.4 制订测试计划时面对的问题	32
2.5 衡量测试计划的标准	32
2.6 制订测试计划	33
习题 2	40
第 3 章 软件测试基本技术	41
3.1 软件测试技术概述	41

3.2	白盒测试技术	42
3.2.1	静态测试	43
3.2.2	程序插桩	46
3.2.3	逻辑覆盖	48
3.2.4	基本路径测试	53
3.2.5	其他白盒测试方法	59
3.2.6	白盒测试应用策略	61
3.3	黑盒测试技术	62
3.3.1	功能测试	62
3.3.2	非功能测试	73
3.3.3	黑盒测试策略	77
	习题 3	77
第 4 章	软件测试过程	80
4.1	软件测试过程概述	80
4.2	单元测试	81
4.2.1	单元测试的定义	81
4.2.2	单元测试的重要性与单元测试原则	82
4.2.3	单元测试的主要任务	83
4.2.4	单元测试环境的建立	85
4.2.5	单元测试主要技术和单元测试数据	86
4.2.6	单元测试工具简介	89
4.2.7	单元测试人员	90
4.3	集成测试	90
4.3.1	集成测试的定义	90
4.3.2	集成测试的主要任务	91
4.3.3	集成测试遵循的原则	91
4.3.4	集成测试实施方案	91
4.3.5	集成测试的测试技术与集成测试数据	96
4.3.6	集成测试人员	97
4.4	系统测试	98
4.4.1	系统测试的定义	98
4.4.2	系统测试前的准备工作	98
4.4.3	系统测试的测试技术和系统测试数据	99
4.4.4	系统测试人员	100
4.5	验收测试	101
4.5.1	验收测试的定义	101
4.5.2	验收测试的主要内容	102

4.5.3	验收测试的测试技术和验收测试数据	105
4.5.4	α 、 β 测试	106
4.5.5	验收测试人员	107
4.6	回归测试	107
4.6.1	回归测试的测试技术和回归测试的数据	108
4.6.2	回归测试的范围	108
4.6.3	回归测试人员	109
4.7	系统排错	109
习题 4		111
第 5 章	测试用例设计	113
5.1	测试用例的基本概念	113
5.2	测试用例的设计	114
5.2.1	测试设计说明	114
5.2.2	测试用例的编写标准	115
5.2.3	测试用例设计应考虑的因素	116
5.2.4	测试用例的分类	118
5.3	测试用例设计实例	119
5.4	测试用例的执行与跟踪	131
5.5	测试用例管理	133
习题 5		136
第 6 章	测试报告与测试评测	137
6.1	软件缺陷和软件缺陷种类	137
6.1.1	软件缺陷的定义和描述	137
6.1.2	软件缺陷的种类	138
6.1.3	软件缺陷的属性	141
6.2	软件缺陷的生命周期	144
6.3	分离和再现软件缺陷	146
6.4	正确面对软件缺陷	147
6.5	报告软件缺陷	148
6.5.1	报告软件缺陷的基本原则	148
6.5.2	IEEE 软件缺陷报告模板	150
6.6	软件缺陷的跟踪管理	152
6.6.1	软件缺陷跟踪管理系统	152
6.6.2	手工报告和跟踪软件缺陷	154
6.7	软件测试的评测	155
6.7.1	覆盖评测	155

6.7.2 质量评测	158
6.7.3 性能评测	162
6.8 测试总结报告	163
习题 6	165
第 7 章 软件测试项目管理	166
7.1 软件测试项目管理概述	166
7.1.1 软件测试项目与软件测试项目管理	166
7.1.2 软件测试项目的范围管理	169
7.2 软件测试文档	169
7.2.1 软件测试文档的作用	170
7.2.2 软件测试文档的类型	171
7.2.3 主要软件测试文档	171
7.3 软件测试的组织与人员管理	175
7.3.1 软件测试的组织与人员管理概述	175
7.3.2 软件测试人员的组织结构	176
7.3.3 软件测试人员	178
7.3.4 软件测试人员的通讯方式	178
7.3.5 软件测试人员管理的激励机制	178
7.3.6 软件测试人员的培训	179
7.3.7 软件测试的组织与人员管理中的风险管理	180
7.4 软件测试过程管理	180
7.4.1 软件项目的跟踪与质量控制	180
7.4.2 软件测试项目的过程管理	181
7.5 软件测试的配置管理	182
7.6 软件测试风险管理	184
7.7 软件测试的成本管理	188
7.7.1 软件测试成本管理概述	188
7.7.2 软件测试成本管理的一些基本概念	188
7.7.3 软件测试成本管理的基本原则和措施	192
习题 7	193
第 8 章 面向对象软件测试	195
8.1 面向对象软件的特点及其对测试的影响	195
8.2 面向对象软件测试的不同层次及其特点	198
8.3 面向对象软件测试模型	206
习题 8	210

第 9 章 软件测试自动化	211
9.1 软件测试自动化基础	211
9.2 软件测试自动化的作用和优势	212
9.3 软件测试自动化的引入条件	217
9.4 软件测试自动化的实施过程	219
9.5 软件测试工具分类	220
9.6 几种常用软件测试工具	223
9.6.1 QACenter	223
9.6.2 WinRunner	225
9.6.3 LoadRunner	226
9.6.4 全球测试管理系统	228
9.6.5 GUI 接口自动化测试工具	229
9.6.6 BoundsChecker	231
9.6.7 Jtest	231
9.6.8 JUnit	232
9.6.9 JCheck	232
9.6.10 CodeReview	232
9.6.11 SmartCheck	233
9.6.12 TrueTime	234
9.6.13 TrueCoverage	234
9.6.14 FailSafe	235
习题 9	235
第 10 章 测试实践——一个实际软件项目的测试案例	236
10.1 被测试软件项目介绍	237
10.1.1 被测试软件项目的背景	237
10.1.2 门诊挂号管理子系统介绍	238
10.1.3 门诊挂号管理子系统的功能需求分析	239
10.1.4 门诊挂号管理子系统的性能及可用性要求	241
10.2 测试计划	242
10.2.1 概述	242
10.2.2 定义	243
10.2.3 质量风险摘要	243
10.2.4 测试进度计划	244
10.2.5 进入标准	244
10.2.6 退出标准	244
10.2.7 测试配置和环境	245
10.2.8 测试开发	245

10.2.9 预算 245

10.2.10 关键参与者 245

10.2.11 参考文档 245

10.3 HIS 测试过程概述 246

10.3.1 单元测试 246

10.3.2 集成测试 247

10.3.3 系统测试 248

10.3.4 验收测试 248

10.4 测试用例设计 248

10.4.1 挂号管理子系统测试大纲 249

10.4.2 其他可用性测试检查标准 250

10.4.3 功能测试用例 250

10.4.4 性能测试用例 259

10.5 缺陷报告 259

10.5.1 建立缺陷报告数据库 259

10.5.2 编写缺陷报告 260

10.6 测试结果总结分析 261

10.6.1 测试总结报告 261

10.6.2 测试用例分析 262

10.6.3 软件测试结果统计分析 263

10.7 软件测试自动化工具 266

10.8 文档测试 268

习题 10 269

参考文献 270

第 1 章 软件测试概述

本章导读

软件测试是软件工程中的重要部分，是确保软件质量的重要手段。最近几年，由于软件的复杂度不断增强、软件产业的不断发展，软件测试得到越来越广泛的重视。本章概括地介绍了软件测试的基本概念，包括软件测试的原则、分类和工作流程等基本知识。

本章主要知识点

- 软件、软件危机和软件工程的基本概念
- 软件缺陷和软件故障
- 软件质量与质量模型
- 软件测试的基本概念
- 软件测试人员的基本素质

1.1 软件、软件危机和软件工程

1.1.1 软件、软件危机和软件工程的基本概念

计算机系统分为硬件系统和软件系统两大部分。在过去的 50 多年里，随着微电子技术的发展和进步，计算机硬件技术以令人惊讶的速度发展，现在已经达到相当成熟的状态。

计算机软件是在计算机系统中与硬件相互依存的另一部分，它是包括程序、数据及其相关文档的完整集合。程序是指按特定的功能和性能要求而设计的能够执行的指令序列；数据是指程序能正常操纵、处理的信息及其数据结构；文档是指与程序设计开发、维护和使用有关的图文材料。

进入 20 世纪 60 年代，随着计算机技术的进步，软件功能日益复杂，人们对软件的需求急剧增加。软件开发从早期以个人活动为主的手工作坊方式，逐步转到以程序员小组为代表的集体开发方式。在这一转换过程中，国外的软件开发人员在开发一些大型软件系统时遇到了许多困难，有些系统最终彻底失败了；有些系统虽然完成了，但比原计划推迟了好几年，而且费用大大超过了预算；有些系统未能完全地满足用户的期望；有些系统则无法被修改和维护。例如，

美国 IBM 公司的 OS/360 系统和美国空军某后勤系统都耗费了几千人·年的工作量，历尽艰辛，但结果却令人失望。在计算机软件的开发和维护过程中产生了一系列严重问题，软件生产与市场需求出现了严重对立——这就是软件危机。软件危机主要表现在以下几个方面。

(1) 软件生产不能满足日益增长的软件需求，软件生产率远低于硬件生产率和计算机应用的增长率，社会出现了软件供不应求的局面。更为严重的是，软件生产效率随软件生产规模的增加和软件复杂性的提高而急剧下降。

(2) 软件生产率随软件规模与复杂性提高而下降，智力密集型行业的人力成本不断增加，这些都导致软件成本在计算机系统成本构成中的比例急剧上升。

(3) 软件开发的进度与成本失控。人们很难估计软件开发的成本与进度，通常情况是预算成倍突破，项目计划一再延期。软件开发单位为了赶进度、节约成本，往往只有降低软件质量。软件开发陷入成本居高不下、质量无保证、用户不满意、开发单位信誉降低的怪圈中。

(4) 软件系统实现的功能与实际需求不符。软件开发人员对用户需求缺乏深入的理解，往往急于编写程序，闭门造车，最后完成的软件与用户需求相距太远。

(5) 软件难以维护。程序中的错误很难改正，要想使软件适应新的运行环境几乎不可能，软件在使用过程中不能增加用户需要的新功能，大量的软件开发人员在重复开发基本类似的软件。

(6) 软件文档配置没有受到足够的重视。软件文档包括开发过程各阶段的说明书、数据词典、程序清单、软件使用手册、维护手册、软件测试报告和测试用例等。这些软件文档的不规范、不健全是造成软件开发的进度、成本不可控制和软件的维护、管理困难的重要原因。

软件危机实际上是软件开发与维护中存在的具有共性的种种问题的汇总。近 30 年来，为解决这些问题，计算机科学家和软件产业从业者已经做出了巨大的努力。

软件危机产生的原因可以从两个方面加以认识，一是软件产品的固有特性；二是软件专业人员自身的缺陷。

软件的不可预见性是软件产品的固有特点之一。与硬件产品不同，软件是计算机系统逻辑部件。在程序代码运行之前，开发工作的质量、进度难以度量。软件产品最终的使用价值是在软件运行过程中体现出来的。软件产品的故障隐蔽性强，可靠性难以度量，对原有故障的修改可能导致新的错误。

软件产品的固有特点之二是软件的规模较大并且逻辑较复杂。现代的软件产品往往规模庞大，功能多种多样、逻辑结构十分复杂。从软件开发管理的角度看，软件生产率常随软件规模和复杂性的增加而下降。当多人合作完成一个系统时，作为一个工程项目，参与人员的组织与信息交流、工作质量与进度控制等更是非常复杂的问题。就目前的软件技术水平而言，软件开发的工作量随软件规模的增大而呈几何级数上升。

软件开发人员存在的问题主要是没有掌握正确的软件开发方法，对软件的开发与维护存在许多模糊、错误的认识，不可避免地导致许多软件错误。软件管理人员存在的问题主要是软件管理技术落后，缺乏软件质量管理。

从宏观上、从整个社会对软件的需求来看，软件危机的实质是软件产品的供应赶不上需求的增长；从微观上看简单地说，软件危机就是开发的软件有错误，软件质量达不到要求，软件项目无法按时完成，软件项目的花费超预算。

为了解决软件危机，既要有技术措施，又要有必要的组织管理措施。软件工程正是从技

术和管理两方面研究如何更好地开发和维护计算机软件的一门学科。

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。通俗来说, 软件工程是一套实现一个大型程序的原则方法, 是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。

为了很好地理解软件工程, 可以将软件工程分解为人 (People)、过程 (Process)、项目 (Project) 和产品 (Product) 4 个部分, 即所谓的 4P 问题。4P 基本上给了软件工程一个大的轮廓, 其主要内涵如下。

- 人员是关于软件工程由谁来完成的问题。

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

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

- 产品指所实现的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。

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是过程, 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是影响项目成功与否的关键因素。

1.1.2 软件工程的目标及其一般开发过程

从狭义上说, 软件工程的目标是生产出满足预算、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。从广义上说, 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。

软件工程强调使用生存周期方法学, 人类在解决复杂问题时, 普遍采用的一个策略就是对问题进行分解, 然后再分别解决各个子问题。软件工程采用的生存周期方法学就是从时间角度对软件开发和维护的复杂问题进行分解, 把软件生存的漫长周期依次划分为若干个阶段, 每个阶段有相对独立的任务, 最后逐步完成每个阶段的任务。一个软件产品从形成概念开始, 经过开发、测试、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。软件工程采用的生存周期方法, 对软件产品的质量保障以及组织好软件开发工具有重要意义。首先, 由于能够把整个开发工作明确地划分成若干个开发步骤, 这样就能把复杂的问题按阶段分别加以解决, 使得对问题的认识与分析、解决的方案与采用的方法以及如何具体实现等工作在各个阶段都有着明确的目标。其次, 把软件开发划分成阶段, 就为中间产品提供了检验的依据。一般软件生存周期包括软件定义、软件开发、软件测试、软件使用与维护等几个阶段。

1. 软件定义

软件定义可分为软件系统的可行性研究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。

可行性研究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作详细的调查,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。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书、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是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,确定软件需求通常需要开发人员与用户多次反复沟通、讨论才能确认,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书的要求由抽象到具体、逐步完成软件开发的过程。软件开发一般由设计和实现等几个阶段组成,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书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。实现也称为编码,就是将软件设计的结果转换成计算机可运行的程序代码。在程序编码中,开发人员必须要制订统一、符合标准的编写规范,以保证程序的可读性和易维护性,提高程序的运行效率。

3. 软件测试

软件需经过严密的测试才能发现软件在整个设计过程中存在的问题并加以纠正。整个测试过程分单元测试、集成测试、系统测试以及验收测试 4 个阶段进行。测试的方法主要有白盒测试和黑盒测试。在测试过程中需要建立详细的测试计划并严格按照测试计划进行测试,以减少测试的随意性。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,有时软件测试的成本甚至可高达软件工程其他步骤成本总和的 3~5 倍。

4. 软件使用和维护

软件的使用是指在软件通过测试后,将软件安装在用户确定的运行环境中移交给用户使用。软件的维护是指对软件系统进行修改或对软件需求变化作出反映的过程。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。软件维护的成功与否直接影响软件的应用效果和软件的生存周期。软件的可维护性与软件的设计密切相关,因此在软件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,软件停止使用。

1.1.3 软件过程模型

软件开发过程包含各种复杂的风险因素,为了解决由这些因素带来的种种问题,软件开

发人员经过多年的摸索，总结出了多种软件工程的实现方式——软件过程模型，如瀑布过程模型、螺旋过程模型、增量过程模型、快速原型过程模型等。这些软件过程模型是软件开发的指导思想和全局性框架，它们的提出和发展反映了人们对软件过程的认识观，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早期人们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。

瀑布过程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，各个阶段的主要工作内容如下。

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

- 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，由软件人员和用户共同讨论决定哪些需求是可以满足的，并且给予确切的描述；写出软件需求规格说明书等文档，提交管理机构审查。

- 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都是功能明确的模块，每个模块都能体现相应需求，这一步骤称为总体设计。然后进行详细设计，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。

- 编码阶段的工作是编写各种层次的代码，包括高级可视化开发系统生成的代码和用第四代程序设计语言编写的代码等。

- 软件测试主要是为了发现程序中的错误。

- 运行维护主要是对软件进行修复和改动，使它能够持续发挥作用。

在实际软件开发过程中，并不是严格按照图 1.1 中所示各阶段顺序执行的，因此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。软件开发人员很少像图 1.1 所示的那样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软件开发人员以前做过的项目类似，这主要是因为绝大多数软件都是复杂的、非线性的。

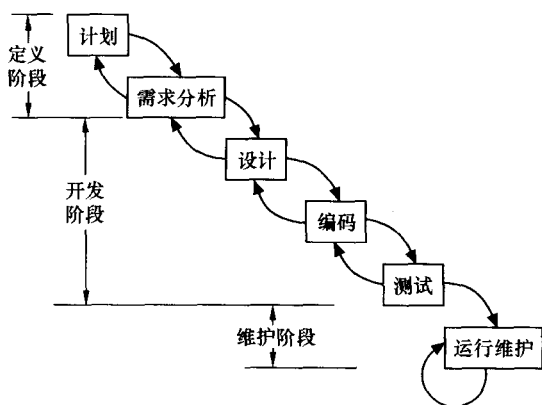


图 1.1 瀑布开发过程

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、测试这组顺序活动。这样做有多种原因，其中主要的原因是规避风险；另一个原因是在早期构造软件的局部版本时即交给客户以获得反馈；还有一个原因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。

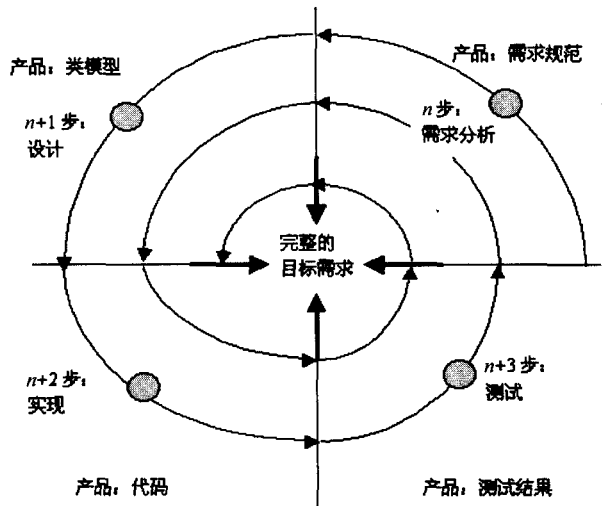


图 1.2 螺旋开发过程

螺旋过程模型的一个额外的优点就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耗费的时间，依此即可以改进后续设计和实现所耗费时间的估计方法，这对于没有任何历史数据的开发组织尤其具有价值。

螺旋过程模型符合典型软件项目的发展特点，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。这主要由于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，为了提高开发小组的生产效率，往往会在前一个迭代结束之前就开始一次新的迭代，这为协调文档的一致性增加了一定难度。

螺旋开发过程需要多少次迭代？这取决于具体的情况。例如，由 3 人组成的开发小组、耗时 4 个月的项目大概需要 2 次或者 3 次迭代，而项目若采用 5 次迭代，则所需要的管理费用通常会超过新增迭代所创造的价值。

3. 增量过程模型

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，这种迭代过程就是增量开发过程。

增量过程模型是用一种几乎连续的过程小幅度地推进项目，如图 1.3 所示。增量过程模型在项目的后期尤其适用，比如当项目处于维护阶段，或者立项的产品与原先开发出来的