

21世纪高等学校规划教材 | 软件工程



实用软件测试教程 (第2版)

王法胜 李绪成 温艳冬 编著



清华大学出版社

21世纪高等学校规划教材 | 软件工程



实用软件测试教程 (第2版)

王法胜 李绪成 温艳冬 编著

清华大学出版社
北京

内 容 简 介

本书主要介绍软件测试基础理论和测试技术,以及自动化测试工具的使用方法。全书共分为 11 章。第 1 章为测试基础部分,主要讲解软件测试的基础理论。第 2 章和第 3 章讲解测试用例的设计方法,较全面地介绍了常用的黑盒、白盒测试用例设计方法,以及 IBM Rational Purify 测试工具的使用方法。第 4 章介绍了单元测试的相关内容,以及常用的测试框架 JUnit、HttpUnit、HtmlUnit 等。第 5 章介绍了集成测试的理论及方法。第 6 章和第 7 章分别系统地介绍了功能测试和性能测试,还介绍了功能及性能自动化测试工具的使用方法,对主流的 HP 系列、IBM Rational 系列自动化测试工具进行了较为详细的讲解。第 8 章和第 9 章分别介绍了本地化测试和网络互联与测试的相关内容。第 10 章和第 11 章分别介绍测试环境搭建技术以及软件测试管理理论,还简要介绍了 QC 的使用方法。

本书可作为高等院校、高职高专院校、示范性软件学院的软件工程、软件测试、软件技术以及计算机相关专业的学生软件测试课程的教材,也可作为从事软件开发和软件测试工作的专业技术人员学习软件测试的参考书和培训教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

实用软件测试教程/王法胜,李绪成,温艳冬编著.--2 版.--北京:清华大学出版社,2014

21 世纪高等学校规划教材·软件工程

ISBN 978-7-302-33364-7

I. ①实… II. ①王… ②李… ③温… III. ①软件—测试—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2013)第 181010 号

责任编辑:魏江江 王冰飞

封面设计:傅瑞学

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.75 字 数:458 千字

版 次:2011 年 6 月第 1 版 2014 年 4 月第 2 版 印 次:2014 年 4 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人: 魏江江

E-mail: weijj@tup.tsinghua.edu.cn

前言

软件测试正在成为 IT 产业生命力的重要保障,越来越多的高校开设了相关课程并设置了与软件测试相关的专业方向以培养测试方面的人才,旺盛的需求在高校培养体系中不断得以体现。与此相得益彰的是软件测试相关的书籍的不断出现。《实用软件测试教程》(第 1 版)自问世以来,得到了读者较为广泛的关注,让作为作者的我们受宠若惊,部分高校的同行人陆续通过电子邮件给我们提出了许多宝贵的修改意见。汇总了同行们的意见后,我们决定对第 1 版进行修订,使得这本书能够在保留原有风格的同时,更适合学生们学习参考。

全书共分为 11 章。第 1 章介绍了软件测试的基础理论,从软件开发过程开始,由浅入深,用较为简洁的语言概述了软件测试的发展历程,从定义、软件缺陷、分类、过程模型、测试的原则、测试的误区等几个方面,让读者初步了解软件测试框架。第 2 章介绍了黑盒测试用例设计方法,对常用的等价类划分法、边界值分析法、决策表法、场景法、正交实验法等进行了解。尤其是对日益受到重视的场景法,本书从理论到实践,进行了系统讲解。第 3 章介绍了白盒测试用例设计方法,主要讲解逻辑覆盖、基本路径、循环测试、代码审查等内容,并在最后一部分介绍了 IBM Rational Purify 测试工具的实际运用。第 4 章介绍了单元测试及常用的单元测试框架 JUnit、HttpUnit 和 HtmlUnit。第 5 章介绍了集成测试的基本理论与方法。第 6 章和第 7 章主要介绍了与功能及性能测试的基本理论及实践相关的内容,并着重介绍了 HP QuickTest Professional、IBM Rational Robot、IBM Rational Functional Tester 等功能自动化测试工具,HP LoadRunner、IBM Rational Performance Tester 等性能自动化测试工具的实际应用。第 8 章介绍了本地化测试的相关内容,包括基本概念、简体中文本地化规范、本地化测试工程师等内容。第 9 章以计算机网络技术为基础,讲述了网络互联与测试的基础知识,包括 OSI 的七层模型、TCP/IP 协议族、IP 地址分类、ping 命令、tracert 命令、ipconfig 命令、arp 命令、ftp 命令和网络故障分析。第 10 章介绍了测试环境搭建的内容,包括测试环境的概述、Windows 系统及 Linux 系统环境下的典型测试环境搭建以及常见问题的解决方法。第 11 章介绍了软件测试管理的相关内容,包括缺陷管理、团队管理、风险管理、过程管理,最后简单介绍了 Quality Center 测试管理工具。本书每一章的最后都有小结部分,供读者在学习过程中进行阶段性总结。

本书是在总结了作者多年教学软件测试课程建设和教学经验,以及作者在软件公司从事测试工作的经验的基础上进行编写的。在编写过程中,作者参考了大量的国内外参考文献资料,不断进行充实和总结,最终完成了本书的编写工作。本书原稿及第 1 版经过了多轮的试用,经过了实践的检验。本书适用于高等院校、高职高专院校、示范性软件学院的软件工程、软件测试、软件技术等专业,以及计算机相关专业的学生使用,可作为软件测试课程的教材;本书也适用于从事软件开发和软件测试工作的专业技术人员,作为学习软件测试的

参考书和培训教材。

在编写过程中,作者得到了多位老师和前辈的帮助,在此对他们表示感谢。同时,我们要感谢清华大学出版社的编辑,他们在本书出版的过程中提出了认真细致的修改意见,使得本书在最大程度上避免了错误的出现。

对于书中存在的不足之处,恳请读者批评指正。

编 者

2013 年 12 月

第 1 章 软件测试基础	1
1.1 软件开发过程	1
1.1.1 线性顺序模型	2
1.1.2 原型实现模型	3
1.1.3 螺旋模型	4
1.2 软件测试概述	6
1.2.1 软件缺陷的案例	6
1.2.2 软件测试的发展	8
1.2.3 软件测试的定义	9
1.2.4 软件缺陷的定义	10
1.2.5 软件测试的分类	11
1.3 软件测试过程模型	13
1.3.1 V 模型	13
1.3.2 W 模型	14
1.3.3 H 模型	15
1.4 软件测试的原则与误区	16
1.4.1 软件测试的原则	16
1.4.2 软件测试的误区	18
本章小结	19
第 2 章 黑盒测试用例设计方法	20
2.1 测试用例设计概述	20
2.1.1 测试用例的定义及表示	21
2.1.2 测试用例与需求	21
2.1.3 设计测试用例的步骤	22
2.2 等价类划分法	22
2.2.1 确定等价类的原则	23
2.2.2 设计测试用例的步骤	23
2.2.3 等价类划分法的应用实例	24
2.2.4 实践体会	25
2.3 边界值分析法	25
2.3.1 边界值分析法的原则	26

2.3.2	内部边界值	27
2.3.3	外部边界值	27
2.3.4	边界值分析法应用实例	28
2.4	因果图法	29
2.4.1	因果图的定义	29
2.4.2	因果图法的基本步骤	30
2.4.3	因果图法的应用实例	31
2.5	决策表法	32
2.5.1	决策表的结构	32
2.5.2	决策表的化简	33
2.6	场景法	35
2.6.1	用例规约	35
2.6.2	场景法的应用	37
2.7	正交实验法	38
2.7.1	正交实验表	38
2.7.2	用例设计的基本步骤	39
	本章小结	40

第3章 白盒测试用例设计方法

3.1	逻辑覆盖测试	41
3.1.1	语句覆盖	41
3.1.2	判定覆盖	43
3.1.3	条件覆盖	43
3.1.4	判定/条件覆盖	44
3.1.5	组合条件覆盖	45
3.1.6	路径覆盖	47
3.2	基本路径测试	47
3.2.1	控制流图	47
3.2.2	环形复杂度	50
3.2.3	独立路径	51
3.2.4	基本路径法的应用	51
3.3	循环测试	53
3.4	代码检查	54
3.4.1	代码审查	54
3.4.2	代码走查	55
3.4.3	桌面检查	56
3.5	Rational Purify 应用	56
3.5.1	Purify 概述	56
3.5.2	Purify 实际运用	57

本章小结	60
第 4 章 单元测试	62
4.1 单元测试概述	62
4.1.1 单元测试的定义	62
4.1.2 单元测试的内容	63
4.2 单元测试的过程	65
4.2.1 计划单元测试	65
4.2.2 设计单元测试	66
4.2.3 执行单元测试	68
4.2.4 评估单元测试	68
4.3 JUnit	69
4.3.1 JUnit 框架的组成	69
4.3.2 JUnit 应用示例	70
4.3.3 SetUp 和 TearDown 方法	72
4.4 HtmlUnit	74
4.5 HttpUnit	76
4.5.1 工作原理	76
4.5.2 特征	76
4.5.3 HttpUnit 实践	76
本章小结	79
第 5 章 集成测试	80
5.1 集成测试概述	80
5.1.1 集成测试的定义	80
5.1.2 集成测试的层次	80
5.1.3 集成测试的原则	81
5.2 集成测试的策略	81
5.2.1 非增量式集成策略	82
5.2.2 增量式集成策略	82
5.3 集成测试的步骤	85
5.3.1 制定集成测试计划	86
5.3.2 集成测试分析和设计	86
5.3.3 集成测试的实施	87
5.3.4 集成测试的执行	87
本章小结	87
第 6 章 功能测试	89
6.1 系统测试概述	89

6.2	功能测试概述	90
6.3	功能测试的策略	90
6.4	功能测试的内容	92
6.5	功能测试的方法	93
6.6	HP QuickTest Professional	95
6.6.1	QTP 工作流程	96
6.6.2	测试脚本的录制与执行	96
6.6.3	测试脚本的分析	98
6.6.4	测试结果的分析	99
6.6.5	检查点技术	100
6.6.6	参数化技术	108
6.6.7	输出值	109
6.6.8	动作切分	110
6.6.9	测试对象库的创建和使用	112
6.7	IBM Rational Robot	114
6.7.1	使用 Robot 录制 GUI 脚本	115
6.7.2	回放脚本	116
6.7.3	验证点的使用	116
6.7.4	数据池的使用	126
6.8	IBM Rational Functional Tester	129
6.8.1	录制脚本	129
6.8.2	脚本的回放	132
6.8.3	验证点	133
6.8.4	测试对象映射	135
6.8.5	RFT 数据池	137
	本章小结	140
第 7 章	性能测试	141
7.1	性能测试基础	141
7.1.1	性能测试的分类	142
7.1.2	性能计数器	142
7.2	性能测试流程	146
7.2.1	性能测试的计划	146
7.2.2	性能测试的设计	146
7.2.3	性能测试的执行	147
7.2.4	测试结果的分析	147
7.3	LoadRunner 应用	147
7.3.1	脚本录制与开发	148
7.3.2	场景设计与运行	164

7.3.3 结果分析	170
7.4 Rational Performance Tester 应用	181
7.4.1 工具的基本使用方法	182
7.4.2 测试验证点的设置	185
7.4.3 数据池的应用	187
7.5 性能测试结果分析	189
7.5.1 性能分析原则	189
7.5.2 常见瓶颈症状	190
本章小结	191
第 8 章 本地化测试	192
8.1 本地化测试概述	192
8.1.1 本地化测试的定义	192
8.1.2 本地化测试与其他测试的区别	193
8.1.3 本地化测试的类型	193
8.1.4 本地化软件缺陷的分类	196
8.1.5 本地化测试的流程	198
8.2 简体中文本地化翻译语言文字规范	200
8.2.1 本地化语言翻译技巧	200
8.2.2 一般翻译规则	203
8.3 本地化测试工程师	204
8.3.1 基础技能	205
8.3.2 专业基础知识	205
8.3.3 职业素质	206
本章小结	206
第 9 章 网络互联与测试	207
9.1 网络基础知识	207
9.1.1 OSI 的七层模型	207
9.1.2 TCP/IP 协议族	209
9.1.3 IP 地址分类	210
9.2 常用网络测试命令	211
9.2.1 ping 命令	211
9.2.2 tracert 命令	214
9.2.3 ipconfig 命令	216
9.2.4 arp 命令	218
9.2.5 ftp 命令	219
9.3 常见网络故障的分析与处理	219
9.3.1 物理层故障分析	220

9.3.2	数据链路层故障分析	221
9.3.3	网络层故障分析	221
9.3.4	传输层及高层故障分析	222
9.3.5	用 ping 命令检测网络故障	222
本章小结		223
第 10 章 测试环境搭建		224
10.1	测试环境概述	224
10.1.1	测试环境的定义	224
10.1.2	测试环境的组成	224
10.1.3	测试环境的管理	225
10.1.4	测试环境的备份与恢复	225
10.2	Windows 环境下 Web 测试环境的搭建	226
10.2.1	操作系统的配置与安装	226
10.2.2	JDK+Tomcat+MySQL 环境的搭建	226
10.2.3	JUnit 单元测试环境的搭建	230
10.2.4	常见问题分析及对策	231
10.3	Linux 环境下测试环境的搭建	232
10.3.1	Linux 下 JDK 的安装与配置	232
10.3.2	Linux 下 Tomcat 的安装与配置	234
10.3.3	Linux 下 MySQL 的安装与配置	236
10.3.4	Linux 下的其他测试工具	239
10.3.5	常见问题分析及对策	240
10.4	自动化测试环境的搭建	242
10.4.1	Rational 自动化测试软件的安装	242
10.4.2	HP LoadRunner 的安装	242
10.4.3	HP QuickTest Professional 的安装	243
10.4.4	常见问题分析与对策	243
10.5	测试管理工具 Quality Center 的安装	243
本章小结		253
第 11 章 软件测试管理		255
11.1	缺陷管理	255
11.1.1	缺陷生命周期	255
11.1.2	缺陷报告的编写	257
11.2	团队管理	257
11.2.1	组织结构	258
11.2.2	素质要求	259
11.2.3	激励方法	260

11.3	风险管理	261
11.4	过程管理	262
11.4.1	测试需求分析	262
11.4.2	测试计划	264
11.4.3	测试设计	265
11.4.4	测试开发	267
11.4.5	测试执行	268
11.5	Quality Center 测试管理工具	269
11.5.1	Quality Center 简介	269
11.5.2	测试流程管理	269
	本章小结	273
附录 A	代码检查	274
附录 B	测试用例模板	276
附录 C	自动化测试工具简表	278
附录 D	软件测试英语词汇	284
	参考文献	286

第1章

软件测试基础

学习目标

- 软件和软件开发过程模型
- 软件测试的发展过程
- 软件测试的定义和分类
- 软件测试的过程模型
- 软件测试的原则

软件测试与软件开发是密不可分的。本章首先简要介绍主要的软件开发过程模型,其次介绍软件测试的基本概念以及测试过程模型,最后介绍软件测试工程师应遵循的软件测试原则。

1.1 软件开发过程

软件在当今的信息社会中占有重要的地位,软件产业是信息社会的支柱产业之一。随着软件应用范围的日益广泛、软件规模的日益扩大,人们开发、使用、维护软件必须采用工程化的方法,才能经济有效地解决软件问题。

1983年国际电子电气工程师协会(Institute of Electrical and Electronics Engineers, IEEE)将软件定义为:计算机程序、方法、规则和相关文档资料以及在计算机上运行时所必需的数据。目前对软件比较公认的解释是:程序、支持程序运行的数据以及与程序有关的文档资料的完整集合。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,数据是使程序能正常操作信息的数据结构,文档是与程序开发、维护和使用有关的图文材料。

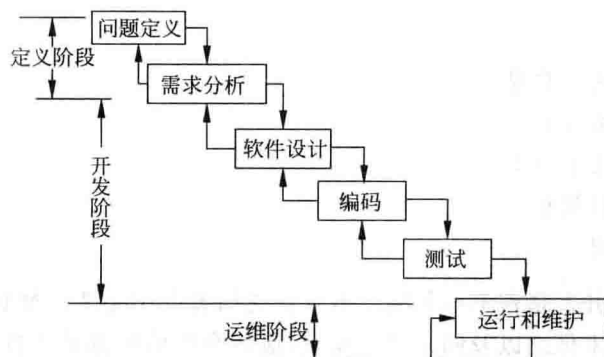
软件开发应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目,需要充分吸收和借鉴人类长期以来从事各种工程项目所积累的行之有效的管理、概念、技术和方法,特别要吸取几十年来人类从事计算机硬件研究和开发的经验教训。几十年的软件开发实践证明:按工程化的原则和方法组织管理软件开发工作是有有效的,是摆脱软件危机的一个主要方法。为了解决软件危机,既要有技术措施(方法和工具),又要有必要的组织管理措施(例如软件质量管理和配置管理)。

软件生命周期模型是从软件需求定义直至软件交付使用后报废为止,在这整个生存期中的系统开发、运行和维护所实施的全部过程、活动和任务的结构框架。到目前为止已经提出了多种模型,主要有线性顺序模型,即传统的瀑布模型(Waterfall Model)、原型模型

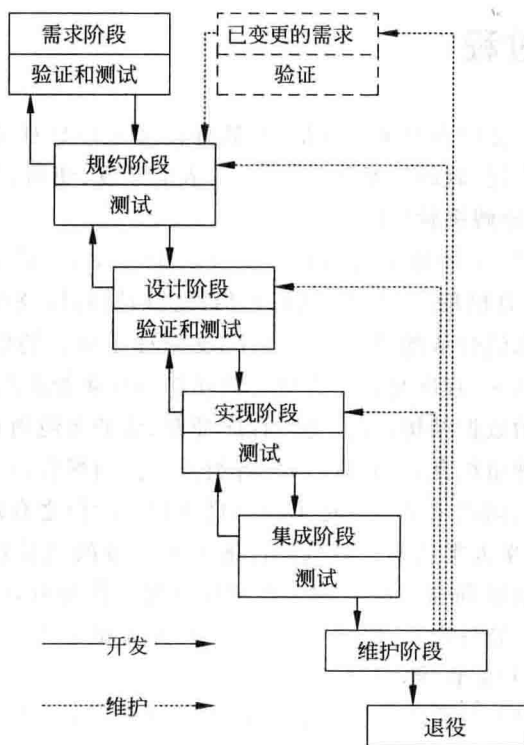
(Prototype Model)、螺旋模型(Spiral Model)等。模型的选择是基于软件的特点和应用的领域。下面介绍一些典型的模型。

1.1.1 线性顺序模型

线性顺序模型提出了系统地按顺序开发软件的方法,从系统级开始分析、设计、编码、测试和支持。从传统工程周期的角度,线性顺序模型包含以下活动:问题定义、需求分析、软件设计、编码、测试、运行和维护,如图 1-1 所示。各项活动以自上而下、相互衔接,如同瀑布流水、逐级下落,体现了不可逆转性。



(a) 线性顺序模型表示一



(b) 线性顺序模型表示二

图 1-1 线性顺序模型的表示

软件开发的实践表明,上述各项开发活动之间并非完全是自上而下的线性模式。实际上,要对每项活动实施的工作进行评审,如果该项活动得到确认则继续进行下一项活动,在图 1-1 中用向下的箭头表示;否则返回前面的活动进行返工,在图 1-1 中用向上的箭头表示。

线性顺序模型的优点表现在它强调开发的阶段性、强调早期计划和需求调查以及强调产品测试。但是在使用时有时会遇到如下一些问题。

(1) 实际项目很少按照该模型给出的顺序进行。虽然线性模型允许迭代,但却是间接的,在项目开发过程中可能会引起混乱。

(2) 客户常常难以清楚地给出所有需求,而该模型却要求必须如此,所以它不能接受在许多项目的开始阶段自然存在的不确定性。

(3) 客户必须有耐性。一直要等到项目开发周期的后期才能得到程序的运行版本,此时若发现大的错误,其后果可能是灾难性的。

(4) 过分依赖于早期进行的需求调查,不能适应需求的变化;由于是单一流程,开发中的经验教训不能反馈应用于本产品的过程;风险往往迟至后期的开发阶段才会显露出来,因而失去了及早纠正的机会,项目开发往往会失去控制。

尽管存在以上问题,传统的生存周期模型在软件工程中仍然占有非常重要的位置。它提供了一个模板,使得分析、设计、编码、测试和支持的方法可以在此指导下执行。

1.1.2 原型实现模型

原型实现模型的基本思想是:原型实现模型从“需求的采集和细化”开始,如图 1-2 所示,然后是“快速设计”,集中于软件中那些对用户或客户可见的部分的表示(如输入方式和输出格式),并最终导致原型的创建。这个过程是迭代的。原型由用户或客户评估,并进一步精化待开发软件的需求,通过逐步调整以满足用户要求,同时也使开发者对将要做的事情有一个更好的理解。

原型实现模型的主要价值是可视化、强化沟通、降低风险、节省后期变更成本、提高项目成功率。一般来说,采用原型实现模型后可以改进需求质量;虽然先期投入了较多的时间,但可以显著减少后期变更的时间;原型投入的人力成本代价并不大,但可以节省后期成本;对于较大型的软件来说,原型系统可以成为开发团队的蓝图。另外,原型通过充分与客户交流,还可以提高客户的满意度。

根据运用原型的目的和方式不同,原型的类型有以下两种。

(1) 抛弃型或丢弃型。先构造一个功能简单而且质量要求不高的模型系统,针对这个模型系统反复进行分析修改,形成比较好的设计思想,据此设计出更加完整、准确、一致、可靠的最终系统。系统构造完成后,原来的模型系统就被丢弃不用。这种类型通常是以对系

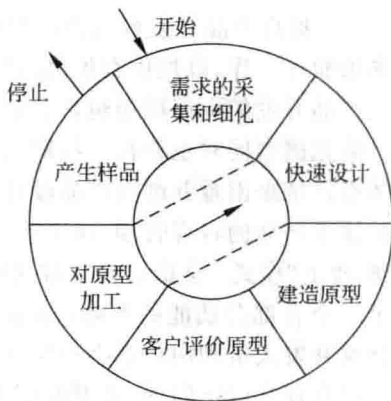


图 1-2 原型实现模型