

软件测试 基础教程

曾文 主编

肖政宏 盘茂杰 韩腊萍 副主编

清华大学出版社



软件测试

基础教程

曾文 主编

肖政宏 盘茂杰 韩腊萍 副主编

清华大学出版社
北京

内 容 简 介

本书介绍了软件测试的基本概念、基本测试原理、基本测试方法、基本测试过程等知识,内容包括软件测试与软件开发关系,软件测试的过程模型,白盒测试,黑盒测试,单元测试,集成测试,系统测试和面向对象的测试;还介绍了自动化测试的基本概念,以及自动化测试的两种测试工具(QTP 和 LoadRunner)的使用;介绍了软件测试管理与软件质量保证;最后以一个软件测试案例说明软件测试的过程。

本书是一本实用性较强的教材,适合作为高等院校计算机、软件工程专业高年级本科生、研究生的教材,同时可供软件测试人员、开发人员、广大科技工作者和研究人员参考。

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

软件测试基础教程/曾文主编.--北京:清华大学出版社,2016

ISBN 978-7-302-42915-9

I. ①软… II. ①曾… III. ①软件—测试—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2016)第 030776 号

责任编辑:刘向威 薛 阳

封面设计:文 静

责任校对:李建庄

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:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.25 字 数:477千字

版 次:2016年7月第1版 印 次:2016年7月第1次印刷

印 数:1~2000

定 价:39.00元

前 言

最近几十年,计算机技术突飞猛进,不仅计算机硬件发展迅猛,软件的开发和使用也越来越普及,越来越高端。从早期的数值计算,到现在的大数据、云计算、互联网+、电子商务、4G 数据通信等,涉及各行各业,软件中存在的问题或安全漏洞也越来越多地在各行各业出现,显然,软件的质量保证就越来越重要。而目前我国的软件测试从业人员数目相当缺乏,并且在 IT 行业中重视的程度不够。本书作者从事软件开发及软件测试教学多年,结合自己在测试行业的工作经验,将软件测试内容进行概括和总结,系统地介绍软件测试的基本理论知识,联系实际操作说明测试的具体实施过程。本书内容由浅入深,易学好用。

本书从软件测试的基本内容起步,将软件测试与软件开发的时间关系作为主线,介绍软件开发的基本过程,测试计划,测试用例设计与实施,测试报告的撰写以及测试分析等。在测试各阶段进行单元测试、集成测试、系统测试、发布测试等;在不同阶段中选择不同的测试方法和技术,如静态测试、白盒测试、黑盒测试、灰盒测试等,并分别介绍怎样使用自动化工具对相关软件进行测试,主要介绍功能自动化工具 QPT,以及性能测试工具 LoadRunner 的基本使用方法。特别之处是理论与实践联系紧密,以中国移动手机软件实例测试操作为案例,讲述怎样对测试软件进行测试需求分析、测试用例设计、测试实施、测试操作记录、测试报告撰写、测试结果分析等。

全书共分为 12 章,第 1 章与第 2 章着重介绍软件测试基本概念和测试原理,软件测试与开发的关系,测试方法分类,测试的过程模型。第 3~6 章介绍软件测试的核心方法和技术,分别介绍白盒测试、黑盒测试、单元测试、集成测试、系统测试和面向对象的测试的知识点和相关技能,并辅以实例说明,从不同的角度选择不同的方法和技术进行测试用例的设计、测试用例的实施,以达到全面理解和掌握软件测试的知识、方法和技术。第 7~9 章,介绍自动化测试的基本概念,自动化测试工具的使用,特别介绍了功能测试工具 QTP 和性能测试工具 LoadRunner 的基本内容和使用方法,并辅以上机操作实例。第 10 章介绍软件测试管理的基本知识,软件测试管理的过程,测试的进度安排,风险控制,测试范围分析及测试工作量的估算,测试报告的撰写及评估,管理工具的使用。第 11 章介绍软件质量保证的基本标准和指标,介绍了软件能力成熟度模型(CMM)、ISO 9001、IEEE 相关概念及内容。第 12 章是手机软件系统测试的一个案例,从手机的基本构成、功能等入手,主要以中国移动智能终端系统(CRM)为案例,从软件需求分析、测试用例设计与实施、撰写测试报告、测试结果分析各过程等进行详细的介绍,并对手机测试相关工作人员的素质和技能提出了相关要求。

本书的特点:

- ① 软件测试知识点全面。本书包括基本的软件测试理论知识,也包括当今业界常用的

测试方法和使用。

② 科学的、系统的工程观点和方法。全书以软件工程开发系统的科学思想方法,将软件测试贯穿整个软件生命周期,介绍软件测试在软件生命周期中各个阶段采用的方法和应用。

③ 理论联系实际。本书各个章节提供大量的应用实例以说明各个测试知识点的运用,并在第12章中以中国移动智能终端系统的测试案例,全面说明测试过程的运用。并且每章后附有习题和练习。

本书适合作为高等院校计算机、软件工程专业高年级本科生、研究生的教材,同时可供软件测试人员、开发人员、广大科技工作者和研究人员参考。

本书由曾文、肖政宏主编。参加编写人员及相关章节安排为:肖政宏编写第1章、第2章;曾文编写第3~7章、第12章;盘茂杰编写第8章、第9章;韩腊萍编写第10章、第11章。本书在编写过程中得到广东技术师范学院计算机学院的老师和学生的支持,在此表示感谢。

编 者

2016年4月

目 录

第 1 章 软件测试概述	1
1.1 软件测试的背景	1
1.1.1 软件的缺陷及其影响	1
1.1.2 软件测试的产生与发展	2
1.2 软件测试的基本概念	4
1.2.1 软件测试的定义	4
1.2.2 软件测试用例	4
1.2.3 软件测试环境	7
1.2.4 软件测试人员的要求	8
小结	8
习题	8
第 2 章 软件开发过程与软件测试	9
2.1 软件开发过程概述	9
2.1.1 软件开发的阶段、活动及角色	9
2.1.2 软件开发的过程模型	10
2.1.3 软件测试与软件开发的关系	13
2.2 软件测试的基本原则	13
2.3 软件测试方法的分类	15
2.3.1 静态测试与动态测试	15
2.3.2 黑盒测试与白盒测试	16
2.3.3 人工测试与自动化测试	17
2.3.4 其他测试分类	17
2.4 软件测试方法在软件开发过程的运用	18
2.5 软件测试的过程模型	18
2.5.1 V-model	18
2.5.2 W-model	19
2.5.3 H-model	20
2.5.4 X-model	21
2.5.5 Pretest-model	22

2.5.6 测试模型的使用	23
小结	24
习题	24
第3章 白盒测试	25
3.1 白盒测试基本概念	25
3.2 静态白盒测试方法	25
3.2.1 检查设计和代码	26
3.2.2 正式审查	26
3.2.3 编码标准和规范	27
3.2.4 通用代码审查清单	28
3.3 程序复杂度及度量方法	30
3.3.1 流图的概念	30
3.3.2 环形复杂度	32
3.3.3 图矩阵	33
3.4 动态白盒测试方法	33
3.4.1 逻辑覆盖法	34
3.4.2 基本路径法	41
3.4.3 循环测试	43
3.4.4 数据流测试	45
3.5 白盒测试的流程与要求	49
3.5.1 白盒测试流程	49
3.5.2 白盒测试要求	50
3.6 白盒测试运用实例	63
小结	67
习题	67
第4章 黑盒测试	68
4.1 黑盒测试的基本概念	68
4.2 黑盒测试方法	68
4.2.1 等价类划分法	69
4.2.2 边界值分析法	75
4.2.3 决策表法	78
4.2.4 因果图法	80
4.2.5 其他黑盒测试方法	83
4.3 黑盒测试的依据和流程	85
4.3.1 黑盒测试的依据	85
4.3.2 黑盒测试的流程	86
4.4 黑盒测试运用实例	86

4.5	黑盒测试与白盒测试的比较	89
4.5.1	白盒测试的优缺点	89
4.5.2	黑盒测试的优缺点	89
4.5.3	黑盒测试与白盒测试的比较	89
	小结	90
	习题	90
第5章	单元测试、集成测试和系统测试	91
5.1	单元测试基本概念	91
5.1.1	单元测试的任务	91
5.1.2	单元测试的环境	93
5.1.3	单元测试的过程	94
5.2	单元测试的策略与方法	95
5.2.1	静态测试与动态测试相结合	95
5.2.2	白盒测试与黑盒测试相结合	95
5.2.3	人工测试与自动化测试相结合	95
5.3	集成测试的概述	96
5.3.1	集成测试的定义	96
5.3.2	集成测试的目标	96
5.4	集成测试的方法	96
5.4.1	大爆炸集成测试	96
5.4.2	自顶向下集成测试	97
5.4.3	自底向上集成测试	98
5.4.4	三明治集成测试	99
5.4.5	其他集成测试策略	99
5.5	集成测试阶段的测试过程	101
5.5.1	集成测试计划阶段	101
5.5.2	集成测试设计阶段	102
5.5.3	集成测试实施阶段	102
5.5.4	集成测试执行阶段	103
5.5.5	集成测试评估阶段	104
5.6	集成测试与单元测试的比较	104
5.6.1	测试的单元不同	104
5.6.2	测试的依据不同	104
5.6.3	测试的空间不同	104
5.6.4	测试使用的方法不同	104
5.7	系统测试的概述	104
5.7.1	系统测试定义和技术要求	104
5.7.2	系统测试的内容	105

5.8	系统测试的方法与过程	108
5.8.1	系统测试方法	108
5.8.2	系统测试过程	109
	小结	112
	习题	112
第6章	面向对象的测试	113
6.1	面向对象测试的基本概念	113
6.1.1	面向对象技术的特点及其对软件测试的影响	113
6.1.2	面向对象的测试模型	117
6.2	面向对象的测试方法概述	117
6.2.1	面向对象的测试方法	117
6.2.2	面向对象测试的相关概念	118
6.3	面向对象的单元测试(类测试)	119
6.3.1	基于服务的测试	120
6.3.2	基于状态的测试	123
6.3.3	测试驱动的实现与代码的组织	130
6.4	面向对象的集成测试和系统测试	137
6.4.1	面向对象的集成测试	137
6.4.2	面向对象的系统测试	139
	小结	141
	习题	141
第7章	软件测试自动化	142
7.1	软件测试自动化的基本概念	142
7.1.1	测试自动化的定义	142
7.1.2	自动化测试使用的术语和技能	143
7.1.3	自动化测试的设计和体系结构	144
7.1.4	自动化测试的过程模型	145
7.1.5	自动化测试的脚本编写与测试运行	147
7.2	自动化测试的方案与选择	147
7.2.1	自动化测试的前提条件	147
7.2.2	自动化测试适合的场合	147
7.2.3	自动化测试选择原则	148
7.3	自动化测试的工具与选择	149
7.3.1	自动化测试工具分类	149
7.3.2	自动化测试工具的选择	150
	小结	151
	习题	152

第 8 章 QTP 测试工具	153
8.1 QTP 简介	153
8.1.1 QTP 的启动	153
8.1.2 QTP 的操作	153
8.2 QTP 的基本功能	155
8.2.1 录制与编辑测试脚本	156
8.2.2 调试与运行测试脚本	161
8.2.3 分析测试结果	162
8.3 QTP 的测试使用	164
8.3.1 录制测试脚本与执行	164
8.3.2 基本测试	168
8.3.3 数据驱动测试	175
小结	179
习题	179
第 9 章 LoadRunner 测试工具	180
9.1 LoadRunner 简介	180
9.1.1 性能测试的基本概念	180
9.1.2 LoadRunner 概述	181
9.2 LoadRunner 的基本功能	182
9.2.1 创建虚拟用户	182
9.2.2 创建负载	182
9.2.3 实时监测	183
9.2.4 分析测试结果	183
9.2.5 重复测试保证系统发布的高性能	183
9.2.6 其他特性	183
9.3 使用 LoadRunner 负载/压力测试	184
9.3.1 制定负载测试计划	184
9.3.2 开发测试脚本	186
9.3.3 创建运行场景	189
9.3.4 运行测试场景	195
9.3.5 监视与分析结果	196
9.4 LoadRunner 测试实例	200
9.4.1 录制与回放	200
9.4.2 单机运行测试脚本	205
9.4.3 创建场景并进行配置	205
9.4.4 执行测试场景	210
9.4.5 结果分析	212

9.4.6 实例总结.....	215
小结.....	216
习题.....	216
第 10 章 软件测试管理	217
10.1 测试计划	217
10.1.1 测试计划的目标	217
10.1.2 测试计划的作用	219
10.1.3 测试策略的制订	219
10.1.4 测试计划的制订	220
10.1.5 测试计划模板	221
10.2 测试范围分析与工作量估算	223
10.2.1 测试范围分析	223
10.2.2 测试工作量估算	224
10.3 资源安排和进度管理	225
10.3.1 确定测试资源	225
10.3.2 测试进度管理	226
10.4 测试风险的控制	227
10.4.1 风险管理的要素与方法	228
10.4.2 常见的风险与特性	229
10.5 测试报告与测试评估	230
10.5.1 测试报告	230
10.5.2 测试评估	232
10.6 测试管理工具	235
10.6.1 测试管理系统的基本构成	235
10.6.2 测试管理工具简介	236
小结	238
习题	239
第 11 章 软件质量保证	240
11.1 软件质量标准	240
11.1.1 软件质量标准	240
11.1.2 衡量软件质量常用的指标	241
11.2 工作现场测试和软件质量保证	242
11.2.1 现场测试	242
11.2.2 软件质量保证	242
11.3 能力成熟度模型	243
11.3.1 CMM 的引入和定义	243
11.3.2 CMM 的基本内容	244

11.4	ISO 9001	246
11.4.1	ISO 9000 系列标准的引入	246
11.4.2	ISO 9001 简介	247
11.5	IEEE 简介	249
11.5.1	IEEE 概述	249
11.5.2	IEEE 829 测试文档国际标准	250
小结	251
习题	251
第 12 章	手机软件测试案例	252
12.1	手机基本知识	252
12.1.1	手机的主要功能	252
12.1.2	手机的基本结构	254
12.1.3	手机软件测试时间	254
12.2	手机软件测试流程和方法	255
12.2.1	手机测试的流程	255
12.2.2	手机测试的方法	255
12.2.3	手机测试常用的技术	256
12.2.4	测试相关文档说明	257
12.3	中国移动智能终端系统软件测试	259
12.3.1	中国移动智能终端系统简介	259
12.3.2	系统架构	261
12.3.3	测试需求分析	264
12.3.4	测试用例的设计与实现	266
12.3.5	撰写测试报告	278
12.3.6	测试结果分析	288
12.4	手机软件测试工程师的素质要求	289
12.4.1	项目领导的任务和能力	289
12.4.2	管理员的工作内容及技能	289
12.4.3	测试工程师的职责和素质	289
小结	290
习题	290
参考文献	291

本章学习目标

- 了解软件测试的产生及发展。
- 理解软件测试的基本概念。
- 掌握软件测试原则及测试用例相关知识。

本章介绍软件测试的产生与发展,再介绍软件测试的基本知识、软件测试的定义、软件测试用例、软件测试的环境及对软件测试人员的要求。

1.1 软件测试的背景

从 1946 年第一台计算机产生开始,就有了“程序”的概念,也就是早期的软件。随着计算机软件和硬件的飞速发展,计算机应用范围越来越广泛,各种高级语言随之产生,软件的地位也日益重要和突出。由于软件的规模和复杂性与日俱增,及软件本身的固有特点——是一种逻辑实体,具有抽象性,不能像硬件一样批量生产,因此软件产品的质量难以控制。

1.1.1 软件的缺陷及其影响

1. 什么是软件缺陷

软件缺陷是指计算机系统或程序中存在的各种各样的破坏正常运行能力的问题、错误或者功能缺陷、瑕疵,主要表现形式是结果出错、功能失效、与用户需求不一致等。

在 IEEE 1983 of IEEE Standard 729 中给出了软件缺陷的定义:软件缺陷就是软件产品中所存在的问题,最终表现为用户所需要的功能没有完全实现,不能满足或不能全部满足用户的需求。

从产品内部看,软件缺陷是软件产品开发或维护过程中所存在的错误、误差等各种问题。从外部看,软件缺陷是系统所需要实现的某种功能的失效或违背。

从现在软件开发的过程来看,软件主要的生命周期经历了软件需求分析、软件设计和软件编码等环节,因此软件缺陷表现的形式多种多样,不仅体现在功能失效方面,还体现在其他方面。主要类型有以下 5 种。

- (1) 软件未实现产品说明书要求的功能。
- (2) 软件出现了产品说明书不应该出现的错误。
- (3) 软件实现了产品说明书未提到的功能。
- (4) 软件未实现产品说明书虽未明确提及但应该实现的功能。

(5) 软件难以理解、不易使用、运行缓慢——从测试员的角度看——最终用户会认为不好。

2. 存在软件缺陷的案例及影响

由于有软件缺陷的存在,会出现一些估计不到的后果,从而存在一些危害。软件缺陷的危害有大有小,小的缺陷可能使软件看起来不美观,使用起来不方便;大的缺陷则可能给用户带来重大的经济损失,甚至危害到生命。下面几个例子足以说明。

1) 千年虫问题(产生约 1974 年)

20 世纪 70 年代,早期的程序员为了节省计算机程序的存储空间,将存储时期中的年份的 4 个数字中只保留了后两位数字。如“1978”只存“78”,并简单地认为只有到了 2000 年,程序开始计算 00 或 01 这样的年份才会发生问题,但那时程序肯定会升级或替换了。到了 1995 年,这些程序仍在使用的,而编程的程序员已退休,没人想到深入程序中去检查 2000 年兼容问题,到了 20 世纪末的最后阶段,全世界的计算机中的各类软件硬件系统,都需要对此类系统进行升级或更换以解决潜在的 2000 年问题,为此付出了数千亿美元的代价。

2) 爱国者导弹防御系统(1991 年)

美国爱国者导弹防御系统即星球大战计划的缩略版本,首次应用在海湾战争中对抗伊拉克飞毛腿导弹的防御战中,在对抗几枚导弹中失利,并在多哈的一次战斗中使 28 名美国士兵死亡。分析发现该系统存在一个软件缺陷,系统时钟的一个很小的计时错误积累起来了误差 14 小时,跟踪系统不再准确。而在多哈的这次袭击中,系统已经运行了 100 小时。

3) 英特尔奔腾浮点除法缺陷(1994 年)

在计算机的“计算器”程序中输入以下算式:

$$(4\ 195\ 835/3\ 145\ 727) \times 3\ 145\ 727 - 4\ 195\ 835$$

若结果是 0,表明计算机没有问题。若得出其他结果,就说明计算机使用的带有浮点除法软件存在缺陷。老式英特尔奔腾处理器,就将这个软件缺陷烧录在一个计算机芯片中。

这个问题是在 1994 年 10 月 30 日由弗吉利亚州 Lynchburg 学院的 Thomas R. Nicely 博士在他的一个实验中发现的。这种情况较少见,只有在处理高精度的数学及工程计算中才会导致出现错误。大多数税务和商务应用的用户不会遇到这种情况。

4) “冲击波”病毒(2003 年)

2003 年 8 月 11 日,“冲击波”计算机病毒在美国发作,美国政府机关、企业及个人的成千上万台计算机中招。“冲击波”病毒在网上迅速传播,结果造成十几万台电子邮件服务器瘫痪,给全世界的 Internet 通信带来惨重的损失。分析其原因,此病毒只是利用微软 Messenger Service 中的一个小缺陷,就攻破了计算机的安全屏障,并使基于 Windows 操作系统的计算机崩溃。

5) 诺基亚手机平台缺陷(2008 年)

2008 年 8 月,诺基亚公司的 Series40 手机平台存在严重的缺陷,使黑客能够在他人的手机上安装和激活应用软件。分析其原因是, Series40 手机所使用的是旧版的 J2ME,这使得黑客能够远程访问本应受限的手机功能。

1.1.2 软件测试的产生与发展

1. 软件测试的产生

由 1.1.1 节可知,软件的缺陷各种各样,产生的原因也不相同,只要有了计算机软件,就

可能存在问题,有了问题,就需要进行测试,发现问题,解决问题。因此在 20 世纪 50 年代后期开展了软件测试工作。

经过大量的测试研究以及测试的积累,软件缺陷产生的主要原因可以归纳如下。

- (1) 需求解释有错误;
- (2) 用户定义错误;
- (3) 需求记录错误;
- (4) 设计说明错误;
- (5) 编码说明有误;
- (6) 程序代码有误;
- (7) 其他有误,如数据输入等。

2. 软件测试的发展

早期软件测试理论与技术并不完善,也不太受重视。最近二十来年软件测试技术发展较快,并且市场需求巨大。

对于软件测试的发展,并没有明确的阶段划分。若从测试的基本思想或导向来分,可以分为以下 3 个阶段。

1) 初级阶段(1957—1971 年)

在早期,软件开发过程没有计划性,也没有“规范化”,软件规模小,复杂程序低,软件测试只是所谓的“调试”,发现运行中的问题,并加以改正。投入的成本很少,人员也少,只是由程序员兼职测试。

到了 1957 年,软件测试开始受到重视,并以功能验证为导向,测试工作常常安排在程序编码之后,此时缺乏有效的测试技术和方法。

2) 发展阶段(1972—1982 年)

1968 年,北大西洋公约组织的计算机科学家们在联邦德国召开的国际会议上提出了“软件工程”的思想,软件的开发从混乱无序发展到结构化开发过程。受“软件工程”的影响,软件测试也得到发展,因此,1972 年在美国北卡罗来纳大学(The University of North Carolina)召开了历史上第一次关于软件测试的正式会议。从此出现了专业的软件测试人员,软件测试开始受到重视,并开始进行相关的研究、探索和发展。

3) 成熟阶段(1983 年至今)

20 世纪 80 年代开始,计算机得到非常广泛的应用,软件的开发进入了高速发展阶段,各行各业都需要开发和应用相应的软件,软件产业逐渐走向成熟,软件规模越来越大,软件复杂程度也越来越高,同时软件的质量要求也越来越高,各种标准应运而生,如 IEEE、CMM 等。此时软件的测试技术也得到相应发展,测试的理论知识越来越多,对测试人员素质的要求越来越高,人员也越来越多,并且开发出了相关的自动化测试技术。

3. 修复软件缺陷的成本

因为软件开发过程要有计划,有条理的开发,也就是现在所说的使用软件工程的方法开发,因此,软件开发要经历需求分析、设计、编程、测试,到公开发布的过程。在整个过程中,都有可能出现各种各样的软件缺陷。随着开发时间的推移,软件缺陷修复成本成倍增长。假如早在进行分析时发现相关功能缺失,立即补上就可以了,付出的代价小得几乎可以忽略不计。如果在发布时发现缺失某个功能,那么此时加上一个功能,相当于重新开发一样,这

时的修补费用会高许多。因此要尽早进行测试。

1.2 软件测试的基本概念

在工业制造的生产过程中,对每一道工序的生产都有质检人员进行相关检查,来确保产品的质量。同样,软件测试是对软件产品进行检验,以确保软件产品的质量。

1.2.1 软件测试的定义

软件测试专家 G. J. Myers 早在 1979 年给软件测试下过定义:软件测试是为了发现错误而针对某个程序或系统的执行过程。也就是说,软件测试以寻找系统中存在的错误为目的,一个成功的测试必须是发现错误的测试。因此,G. J. Myers 给出以下与测试相关的三个要点。

- (1) 测试是为了证明程序有错,而不是证明程序无错误;
- (2) 一个好的测试用例在于它能发现至今未发现的错误;
- (3) 一个成功的测试是发现了至今未发现的错误的测试。

1983 年,IEEE 给软件测试的定义是:使用人工或自动的手段来运行或测定某个软件系统的过程,其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别。

1990 年,IEEE 再次给出软件测试的定义:

- (1) 在特定的条件下运行系统或构件,观察或记录结果,对系统的某个方面做出评价;
- (2) 分析某个软件项以发现现存的和要求的条件的差别并评价此软件项的特性。

以上的定义都有一定的片面性,我们知道,现在的软件开发是以软件工程理论为基础的,开发过程分为需求分析、设计、编码等,不能只对系统程序进行测试,找出系统程序中的错误,而对分析、设计等过程发生的错误视而不见。众多的资料表明,60%以上的软件错误并不是程序的错误,而是分析和设计的错误。

V&V,代表 Verification 和 Validation,即“验证”和“有效性确认”。测试从用户需求出发,以需求为依据,来对产品进行检验。

- (1) “验证”是检验软件是否已正确实现了产品规格说明书所定义的系统功能和特性。
- (2) “有效性确认”是确认所开发的软件是否满足用户真正需求的活动。这说明软件测试不仅要通过运行程序或系统来进行检验,而且要对软件系统相关文档进行检查和评审,以确认这些文档的内容是否满足客户要求。

软件产品由文档、数据和程序组成,那么软件测试就是对软件开发过程中形成的文档、数据以及程序进行相关的测试。

1.2.2 软件测试用例

1. 软件测试用例定义

在进行软件测试时,需要对系统的某项功能或特性进行验证,这就需要从不同的方面获得不同的数据输入来进行验证。

IEEE 标准 610(1990)给出的定义:测试用例是一组测试输入、执行条件和预期结果的

集合,目的是要满足一个特定的目标,比如执行一条特定的程序路径或检验是否符合一个特定的需求。

测试用例就是为了某个测试点而设计的测试操作过程序列、条件、期望结果及相关数据的一个特定的集合。这就关系到输入与输出的数据,以及相关的环境。

2. 测试用例的元素

软件测试的目的是为了保障软件的质量,软件的质量包括正确性、可靠性、可维护性、可读性、结构化、可测试性、可移植性、可扩展性、安全性、用户友好界面、易用性等。那么什么时候开始测试?测试的内容是什么?在哪里进行测试?怎样执行测试?什么时候结束测试?

根据软件开发过程,软件测试不仅是对程序的测试,而是贯穿软件开发整个生命周期的过程。因此,软件开发的各个阶段都是需要测试的,包括需求分析说明书、概要设计和详细设计说明书以及源代码,都是软件测试的对象。

软件测试设计的关键问题可以概括为以下的5W1H。

Why: 为什么测试?对功能、性能、可用性、容错性、安全性等测试,检验是否符合相关要求。

What: 测什么?测试的对象可以是文档、代码、图表等。

Where: 在哪里测?测试用例的环境,包括系统的硬件、软件和网络环境等。

When: 什么时候测?测试用例所需的前提条件,尽早开始。

Which: 什么数据?测试用例设计的各种数据。

How: 如何执行?结果怎样?要根据测试用例设计的步骤来执行,最后进行结果比较,确定是否一致。若一致才能通过测试。

3. 测试用例设计的基本原则

测试用例是测试的基础,测试用例的质量直接关系到测试结果的质量,从而关系到整个测试产品的质量。那么什么样的测试用例是符合高质量要求的测试用例呢?可从以下两个层次考虑。

(1) 低层次——从单个测试用例看,衡量其描述的规范性、可理解性及可维护性等。

(2) 高层次——以满足某一个测试目标或测试任务来衡量一组测试用例的结构、设计思路和覆盖率等。

因此,对单个测试用例要求满足:单一性(一个测试用例只面向一个测试点);各项信息描述清楚、准确;测试目标性强;具有可操作性;操作步骤简单准确;操作结果可验证;测试环境正确,前提条件充分等。对于整体测试用例的要求比单个测试用例考虑的因素要多些,一组测试用例是针对一个功能模块或一个产品的测试,整体质量的最重要的指标是测试用例的覆盖率,当然覆盖率越高,质量越好。其次,作为一个测试用例的集合,要考虑一个合理的结构层次,尽量减少重复的或多余的测试用例。因此要求更高,除了上面的要求外,还需要:覆盖率高;易用性强;易维护性;粒度适中;冗余少或没有;并有负面测试。

测试用例的基本原则有以下三个。

(1) 代表性。测试用例能代表并覆盖各种合法的或不合法的、边界内的或越界的以及极限的输入数据、操作和环境的设置。