

# 软件测试方法和 技术实践教程

王丹丹 编著  
朱少民 主审



清华大学出版社

21世纪高等学校规划教材 | 软件工程



# 软件测试方法和 技术实践教程

王丹丹 编著

清华大学出版社  
北京

## 内 容 简 介

本书是计算机科学与技术专业、计算机软件专业以及其他相关专业学生学习软件测试理论时的配套实验教材。在介绍软件测试的主要方法的同时,以实验指导书的形式给出运用各种方法的软件测试案例,便于读者学习实践。

全书分为两篇:第1篇介绍软件测试的原理与方法,着重介绍黑盒功能测试的等价类划分法、边界值法、判定表法和 Pair-wise 方法以及白盒逻辑覆盖测试的实验原理及具体的实验案例;第2篇介绍软件测试的技术,着重介绍单元测试、集成测试和系统测试的实验原理及具体的实验案例。

本实验教材具有一定的实用性和指导性,可以作为高等院校计算机、软件工程等专业高年级本科生和研究生学习软件测试的实验指导书,同时可供需要了解和学习软件测试的开发人员和广大科技工作者参考。

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

软件测试方法和技术实践教程/王丹丹编著. —北京:清华大学出版社,2017

(21世纪高等学校规划教材·软件工程)

ISBN 978-7-302-46114-2

I. ①软… II. ①王… III. ①软件—测试 IV. ①TP311.55

中国版本图书馆 CIP 数据核字(2017)第 006072 号

责任编辑:黄芝 王冰飞

封面设计:傅瑞学

责任校对:徐俊伟

责任印制:王静怡

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

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

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:11

字 数:275千字

版 次:2017年6月第1版

印 次:2017年6月第1次印刷

印 数:1~2000

定 价:29.00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

软件测试是一门对于工程实践能力要求很高的课程,在整个计算机科学与技术学科的人才培养规划里,是一门偏重于实践的核心课程,旨在培养学生的计算机实际开发能力。目前,许多高校在计算机专业或软件工程专业开设了这门课程,可见软件测试课程的重要性。

事实上,由于软件测试这门课程重在培养学生的工程实践能力的特点,从软件测试教学大纲和教学计划可以看出,所安排的实验节数非常多,应该是全部学时的一半以上,因为学生只有通过一个个软件测试案例的实验,才能够真正掌握软件测试的各种方法并且做到融会贯通。

目前,软件测试课程的教材数量多而且质量都很好,但是基于培养软件测试实际动手能力的实验配套教材还不多,学生反映即使买到了合适的软件测试教材,但是很难买到真正实用的软件测试实验指导用书。原因是这类书籍由于偏重实践环节,所涉及的课题和实验的准备需要很长的时间,不像一般的理论教材那么好写,所以出书十分困难。但是学生要想掌握软件测试方法,在实际测试环节中,又迫切地需要好的、详细介绍实施软件测试方法具体步骤的实验教程。

本人多年从事软件工程、软件测试及软件质量保证的研究并讲授相关课程,长期指导学生进行实际的软件测试活动,在软件测试的课堂上多年采用清华大学出版社出版、朱少民老师编写的《软件测试方法和技术》一书,按照朱老师教材的章节编写思路,结合自己多年辅导学生进行软件测试学习所积累的案例,编写了这本软件测试的实验教程。本书充实了软件测试方法和技术的实验环节案例,详细描述了各种软件测试常用的方法和技术在实践中应用的步骤和过程,由浅入深、循序渐进,有些章节的实验甚至分步骤详尽地做了阐述。所以,本书应该是一本容易入门的、浅显易懂的实验教材,特别适合作为一本学习软件测试的入门实验指导书。书中所涉及的所有案例都是笔者这些年在课堂上带着学生反复实践过的,相信会对学习软件测试的读者有所启发。

本书设想成为清华大学出版社出版、朱少民老师编写的《软件测试方法和技术》一书的配套实验类指导书,因此在内容与篇章结构上尽量与朱老师的原书一致,可以说本书是《软件测试方法和技术》一书的实践教程和重要补充。同时,本书注重实践环节的积累,用具体的案例来讲述软件测试理论应用的具体方法,实用性和指导性很强。

本书对软件测试的主要方法和重要技术均做了介绍,本着理论与实践相结合的原则,在介绍软件测试某个具体的方法之后,以实验指导书的形式相应地给出一个利用此方法的实验案例,便于读者学习实践。

在各章的实验指导书中,除说明本次实验的目的、所要求的实验环境、本次实验的内容之外,对于实验相关的软件测试理论,都以【实验原理】的形式再次简明扼要地加以阐明;对于软件测试初学者感觉困难的地方,都以【实验步骤】的形式予以具体讲解,读者只要按照实验步骤进行操作,就可以很容易地完成整个实验。提醒读者不要急于阅读【实验指导】部分,

因为【实验指导】是类似习题答案的内容,因此为了培养和锻炼自己的软件测试能力,可以先不阅读【实验指导】,自己完成整个实验之后再阅读这一部分,看看自己的测试设计方法与【实验指导】有什么不同,从而达到学习提高的目的。对于初学者容易犯的错误都以【实验中需要注意的问题】的形式进行了概括总结。有些实验还有【实验拓展】部分,提出实验拓展问题,或者是需要进一步进行实验的内容。

对于本实验教程的所有实验案例,读者都可以参阅书中的【实验指导】,自己进行相同的实验操作,所涉及的可测程序和相应的软件系统平台、测试工具软件,都可以从本书指定的配套网站下载使用。

本书所涉及的多个测试实验均可被读者实际操作,相信许多读者凭借本书,可以真正摆脱只能学习软件测试理论,无法动手进行实际测试的困境。各章的实验指导阐述细致,浅显易懂。

本书参考了软件测试的相关书籍以及互联网上的一些软件测试理论文章,特别是朱少民老师编写的《软件测试方法和技术》一书,对于各位作者表示深深的谢意。

本书还参考了我的学生郭莹和范逸飞的毕业设计论文。作为毕业设计导师,向他们表示感谢。

由于本人水平有限,尽管参照了很多的文献和听取多方的意见,但由于时间问题和本人能力问题,书中难免存在漏洞与误区,还望读者朋友指正。

作 者

2017年1月

# 目 录

## 第 1 篇 软件测试的原理与方法

<b>第 1 章 测试用例设计</b> .....	3
1.1 软件测试的基本理念 .....	3
1.1.1 软件质量 .....	3
1.1.2 软件缺陷 .....	4
1.1.3 软件测试 .....	4
1.2 测试用例设计 .....	5
1.2.1 测试用例的类型 .....	5
1.2.2 测试用例的一般结构 .....	6
1.2.3 设计测试用例需考虑的因素 .....	8
<b>第 2 章 黑盒测试原理与方法</b> .....	10
2.1 黑盒测试概述 .....	10
2.2 等价类划分法 .....	11
2.2.1 等价类划分法的原理 .....	11
2.2.2 等价类划分法的实验 .....	13
2.3 边界值分析法 .....	20
2.3.1 边界值分析法的原理 .....	20
2.3.2 边界值分析法的实验 .....	23
2.4 判定表法 .....	31
2.4.1 判定表法的原理 .....	31
2.4.2 判定表法的实验 .....	33
2.5 Pair-wise 方法 .....	36
2.5.1 Pair-wise 方法的原理 .....	36
2.5.2 Pair-wise 方法的实验 .....	38
<b>第 3 章 白盒测试原理与方法</b> .....	46
3.1 白盒测试概述 .....	46
3.1.1 语句覆盖 .....	47
3.1.2 判定覆盖 .....	47
3.1.3 条件覆盖 .....	48



3.1.4	判定/条件覆盖 .....	49
3.1.5	条件组合覆盖 .....	49
3.1.6	基本路径覆盖 .....	50
3.1.7	白盒测试与黑盒测试的比较 .....	51
3.2	白盒测试实验 .....	52

## 第2篇 软件测试的技术

<b>第4章</b>	<b>单元测试 .....</b>	<b>59</b>
4.1	单元测试概述 .....	59
4.1.1	什么是单元测试 .....	59
4.1.2	单元测试中的测试用例设计 .....	60
4.1.3	单元测试的过程 .....	60
4.1.4	单元测试的主要测试手段 .....	61
4.2	单元功能测试 .....	61
4.2.1	单元功能测试概述 .....	61
4.2.2	单元功能测试实验 .....	62
4.3	单元逻辑覆盖测试 .....	76
4.3.1	单元逻辑覆盖测试概述 .....	76
4.3.2	静态测试 .....	77
4.3.3	驱动程序和桩程序 .....	77
4.3.4	利用驱动程序和桩程序进行单元测试 .....	77
4.3.5	利用JUnit进行单元测试 .....	83
4.3.6	利用JUnit进行单元测试的实验 .....	87
<b>第5章</b>	<b>集成和系统测试 .....</b>	<b>96</b>
5.1	集成测试概述 .....	96
5.2	系统测试概述 .....	97
5.3	利用业务流进行集成和系统功能测试 .....	98
5.4	利用业务流进行集成和系统功能测试的实验 .....	99
5.5	系统性能测试 .....	117
5.5.1	性能测试定义与要点 .....	117
5.5.2	性能测试术语 .....	117
5.5.3	性能测试流程 .....	118
5.5.4	系统性能测试软件 LoadRunner .....	119
5.5.5	利用 LoadRunner 软件进行系统性能测试的实验 .....	127
<b>附录A</b>	<b>管理信息系统单元测试共通点检查表 .....</b>	<b>153</b>
<b>附录B</b>	<b>“实验设备管理系统”程序安装说明 .....</b>	<b>157</b>
<b>参考文献</b>	<b>.....</b>	<b>166</b>

第 1 篇

# 软件测试的原理与方法

第1章 测试用例设计

第2章 黑盒测试原理与方法

第3章 白盒测试原理与方法



# 第1章

## 测试用例设计

### 1.1 软件测试的基本理念

软件测试是软件工程的一个重要环节,可以说它贯穿于软件开发的整个生命周期之中。软件测试的目的是发现软件中隐藏的缺陷和错误并加以完善,以满足用户需求定义,提高软件用户的满意度。软件质量的好坏在相当大的程度上取决于有没有进行完善的软件测试。

#### 1.1.1 软件质量

众所周知,产品的质量是能否满足用户需求的关键,比如一部高质量的手机,它一定具备以下的特点。

- 通话信号稳定,没有时断时续信号不好的现象;
- 音质好,没有杂音;
- 功能全,可以上网、发邮件、拍照、听音乐、聊天、购物消费等;
- 美观、轻便、操作起来手感舒适;
- 价格合理;
- 维修服务好。

如果我们把软件作为产品,那么软件产品的质量又是指什么呢?概括地说,软件质量就是“软件与明确的和隐含的的定义的需求相一致的程度”。具体地说,软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准,以及所有专业开发的软件都应具有的隐含特征的程度。软件质量具体包括以下几个方面。

(1) 性能(Performance):指系统的响应能力,即要经过多长时间才能对某个事件作出响应,或者在某段时间内系统所能处理的事件个数,包括操作响应速度、计算机资源使用消耗情况等。

(2) 可用性(Availability):指系统能够正常运行的时间比例。

(3) 可靠性(Reliability):指系统应用或者错误面前,在意外或者错误使用的情况下维持软件系统功能特性的能力;可靠性是用户使用的根本,可靠性低意味着用户在使用时系统频频出现故障,无法满足用户的使用需要。

(4) 健壮性(Robustness):指在处理或者环境中系统能够承受的压力或者变更能力。

(5) 安全性(Security):指系统向合法用户提供服务的同时能够阻止非授权用户使用的企图或者拒绝服务的能力。

(6) 可修改性(Modification):指能够快速地对系统性能价格比进行变更的能力。

(7) 可变性(Changeability):指体系结构扩充或者变更成为新体系结构的能力。

(8) 易用性(Usability):衡量用户使用软件产品完成指定任务的难易程度;通常包括简单安装、轻松使用以及具有友好的界面等。

(9) 可测试性(Testability):指软件发现故障并隔离定位其故障的能力特性,以及在一定的时间或者成本前提下进行测试设计、测试执行的能力。

(10) 功能性(Functionality):指系统所能完成所期望工作的能力。

(11) 互操作性(Inter-Operation):指系统与外界或系统与系统之间的相互作用能力。

### 1.1.2 软件缺陷

如果软件中存在缺陷和错误,那么软件产品的质量一定是有问题的,它一定是一个无法满足用户需要的甚至给用户带来麻烦的系统。

我们通常把程序中隐藏的功能缺陷或错误叫作 Bug。软件缺陷的主要类型/现象包括以下几点。

- 功能、特性没有实现或部分实现;
- 设计不合理,存在缺陷;
- 实际结果和预期结果不一致;
- 运行出错,包括运行中断、系统崩溃、界面混乱;
- 数据结果不正确、精度不够;
- 用户不能接受的其他问题,如存取时间过长、界面不美观等。

总之,一个好的、质量高的软件应该是相对的无产品缺陷(Bug Free)或只有极少量的缺陷,它能够准时递交给用户,所用的费用都是在预算内的并且满足客户需求,是可维护的。与其他产品一样,有关质量的好坏的最终评价依赖于用户的反馈。

### 1.1.3 软件测试

软件测试就是要尽快地、最大程度地发现软件中的问题,以保证和提高软件产品的质量。其目的不仅仅是为了证明软件实现了用户需求定义的内容,对软件需求分析、设计规格说明和编码进行最终复审,还要检验软件在用户定义以外的不合理输入情况下的反应,即系统的可靠性和容错能力如何,可以说软件测试是软件质量保证的关键步骤和重要手段。

有一种错误的观点认为软件测试只是在设计和编码阶段结束之后的对软件系统进行的各种综合测试,是开发后期的一个阶段,是系统交付使用之前的最后一道工序。实际上,软件测试应该贯穿于软件开发的整个周期,在软件需求定义、设计、编码的各个过程结束之前都应该进行复审,这样做的目的是可以大大地减少后续过程的返工,提高软件开发效率,有效地控制软件工程的成本。

综上所述,软件测试是由“验证(Verification)”和“有效性确认(Validation)”这两个方面的活动构成的整体。

### 1. 验证(Verification)

检验软件是否已正确地实现了产品规格书所定义的系统功能和特性,是否正确地构造了软件?即是否正确地做事;验证开发过程是否遵守已定义好的内容;验证产品满足规格设计说明书的一致性。

### 2. 有效性确认(Validation)

确认所开发的软件是否满足用户真正需求的活动,是否构造了正是用户所需要的软件?即是否正在做正确的事。验证产品所实现的功能是否满足用户的需求。

各个阶段软件测试的依据是前一阶段所形成的文档,软件测试中发现的错误可能是系统前期工程各个阶段的问题的反映。为了改正软件测试中发现的错误,必须找到错误的代码进行重新编写,或是错误的设计进行重新修改,然后再将修改后的代码进行重新测试,直到问题被全部解决为止。需要注意的是,回归测试的关联性一定要引起充分的重视,修改一个错误而引起更多错误出现的现象并不少见,往往程序的重新设计和修改会给原来没有发现错误的代码带来影响,所以,在代码更改之后,有时还要对相关关联的功能重新进行测试,因为所修改的程序代码可能会对其产生影响。

## 1.2 测试用例设计

既然软件测试的目的是为了发现程序中的错误而执行程序的过程,那么软件测试就是要根据软件开发各阶段的规格说明和程序的内部结构,精心设计一批测试用例,即输入一些数据而得到其预期的结果,并利用这些测试用例去运行程序,以发现程序错误的过程。好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案;成功的测试是发现了迄今为止尚未发现的错误的测试。

测试用例设计是测试最基本、最关键的工作,可以说,懂不懂测试、会不会测试就是看一个人会不会写测试用例,即是否能够找到正确的、高效的测试数据。所以在软件测试实验中必须掌握测试用例的设计方法。

测试用例是输入、执行条件和一个特殊目标所开发的预期结果集合,是可以独立进行测试执行的最小单元,换句话说,测试用例包括输入数据、操作步骤和期望结果。

### 1.2.1 测试用例的类型

测试用例按测试目的不同可分为以下几种类型。

#### 1) 需求测试用例

测试是否符合需求规范,通常是按照需求执行的功能逐条地编写输入数据和期望输出。一个好的需求用例是可以用少量的测试用例就能够覆盖所有的程序功能。

### 2) 设计测试用例

测试是否符合系统逻辑结构,检测的是代码和设计是否完全相符,是对底层设计和基本结构上的测试。设计测试用例可以涉及到需求测试用例没有覆盖到的代码空间,例如,界面的设计等。

### 3) 代码测试用例

测试代码的逻辑结构和使用的数据,是基于运行软件和数据结构上的,它要保证可以覆盖所有的程序分支、语句和输出。

以上3种用例所用的数据又可分为正常数据、边缘数据和错误数据。

#### 1) 正常数据

在测试中所用的正常数据的量是最大的,而且也是最关键的。少量的测试数据不能完全覆盖需求,但人们要从中提取出一些具有高度代表性的数据作为测试数据,以减少测试时间,测试所花费的时间直接影响到测试以及整个工程的进度,并影响整个工程项目的成本。

#### 2) 边缘数据

边缘测试是介于正常数据和错误数据之间的一种数据。例如,若使用 SQL Server 数据库,则可把 SQL Server 关键字(如“'”“AS”“Join”等)设为边缘数据。其他边缘数据还有 HTML 的“HTML”“<>”等关键字以及空格、@、负数、超长字符等。边缘数据要根据不同的系统特点,并且依靠测试人员的丰富经验来制定。

#### 3) 错误数据

显而易见,错误数据就是编写与程序输入规范不符的数据,从而检测输入筛选、错误处理等程序的分支。可以说错误数据对于检验软件可靠性是必不可少的。

由于执行测试用例的数据量巨大,还要进行回归测试,所以可以考虑使用自动测试工具,但提取测试数据仍要依靠编写测试用例人员所掌握的软件测试方法和软件测试工作经验。这里还要注意,有时程序中的某些错误自动测试也许不能找到,手动测试所找到的错误会比自动测试所找到的要多。

## 1.2.2 测试用例的一般结构

测试用例的一般结构如下所述。

(1) 标志符(Identification): 唯一,必须。相当于测试用例的编号,唯一标识该测试用例的值。

(2) 所属模块: 可选。

(3) 测试项(Test Items): 必须。测试的对象。

(4) 测试用例名称: 必须。

(5) 测试环境要求: 可选。测试环境的软硬件配置情况,特别是系统性能测试时,在不同的测试环境下系统的性能差别很大,测试环境要求的描述更加重要。

(6) 输入标准(Input Criteria): 必须。包括输入数据、前提条件、操作步骤等。前提条件是指事先设定、条件限制,如已经登录、某个选项已经选上等。

(7) 输出标准(Output Criteria): 必须。

(8) 测试用例之间的关联：可选。某些测试用例之间是有关联的，在测试用例设计表上要明确地说明它们之间的关联性。

(9) 优先级：可选。优先级别高的问题是需要尽快解决的问题，往往是某些重要的核心功能方面的缺陷(Bug)，如果不加以解决的话，其他页面或程序功能都无法进行测试；或者是系统底层的问题，往往是牵一发而动全身，影响的模块比较多，需要尽快地解决，这样的软件测试用例可以设计为优先级高的用例。对于比较复杂的软件系统，程序的错误按照程度可以分为：建议修改、警告、错误、崩溃等。

(10) 关联的缺陷标识符：可选。根据系统复杂情况，可以将所测试出来的缺陷和错误单独列到一个软件 Bug 文档中，那么此处就要写明关联的缺陷标识符。

图 1-1 和图 1-2 为测试用例的两个示例。

**【示例 1：书写不够规范的测试用例】**

测试目标：验证输入错误的密码是否有正确的响应。

测试环境：Windows XP 操作系统和浏览器 Firefox。

输入数据：用户邮件地址和口令。

步骤：

1. 打开浏览器。
2. 单击页面右上角的“登录”链接，出现登录页面。
3. 在电子邮件的输入框中输入：test@gmail.com。
4. 在口令后面输入：xxxabc。
5. 单击“登录”按钮。

期望结果：

登录失败，页面重新回到登录页面，并提示“用户密码错误”。

图 1-1 书写不够规范的测试用例

**【示例 2：书写规范的测试用例】**

ID: LG0101002

用例名称：验证输入错误的密码后是否提示正确。

测试项：用户邮件地址和口令。

环境要求：Windows XP SP2 和浏览器 Firefox 3.0.3

参考文档：软件规格说明书 SpecLG01.doc

优先级：高

层次：2(即 LG0101 的子用例)

依赖的测试用例：LG0101001

步骤：

1. 打开浏览器。
2. 单击页面右上角的“登录”链接，出现登录页面。
3. 在电子邮件的输入框中输入：test@gmail.com。
4. 在口令后面输入：xxxabc。
5. 单击“登录”按钮。

期望结果：

登录失败，页面重新回到登录页面，并提示“用户密码错误”。

图 1-2 书写规范的测试用例



### 1.2.3 设计测试用例需考虑的因素

在测试用例设计中应该充分考虑以下因素。

(1) 测试用例应该具有代表性、典型性。

由于测试不可能穷尽进行的缘故,测试用例是测试输入的所有可能情况的代表,因此测试用例必须具有代表性、典型性。

(2) 寻求系统设计、功能设计的弱点。

测试用例需要考虑到正确的输入,也需要考虑错误的或者异常的输入,需要分析怎样使得这样的错误或者异常能够发生,因此在设计测试用例时,往往要寻找那些系统设计中易出错之处,有针对性的设计一些测试用例,以发现系统的问题所在。

(3) 考虑用户实际的诸多使用场景。

软件系统最终是提供给用户使用的,软件测试的依据是软件工程前期所做的系统需求分析时得到用户确认的软件系统规格说明。因此,在设计软件测试用例时,必须设计足够的使用场景,模拟用户各种各样的使用情形,这也是构筑整个系统的初衷。

在设计测试用例时应该尽量地注意以下几个方面的问题。

(1) 尽量地避免含糊的测试用例。

例如,在测试登录页面时,不要用“输入正确的密码后,程序工作正常”“输入错误的密码后,程序工作不正常”这样的含糊的测试用例,而是具体设计一些满足系统设计规格的密码进行测试。如果系统要求密码必须是由数字组成并且必须是6位数字,则可以用“123456”作为正确的密码测试用例,用“123”作为违反系统规格的测试用例去进行测试。总之,测试用例要具体化,不能用语言笼统描述,需要落实到具体的数据和操作步骤上,并且根据系统的规格说明,想好每一个测试用例相对应的预期测试结果。

(2) 尽量地将具有相类似功能的测试用例抽象并归类。

前面我们提到过,软件测试的时间和人员成本是项目管理层面所必须考虑的重要因素,软件测试的测试用例设计方法和具体设计结果直接影响到软件测试进度和成本。因此,根据不同的软件工程项目特点和项目工程的不同测试阶段,选择合适的软件测试方法是非常重要的,同时测试用例的设计应该尽量地进行分类和归纳,能够用一个测试用例完成的测试不要采用冗余的多个测试用例。

(3) 尽量地避免冗长和复杂的测试用例。

如果一个测试用例非常冗长和复杂,就要考虑将其分割为独立的、小的测试用例进行测试,因为过于复杂的测试用例一定包含了多个复杂的测试因素,不能准确地反映程序出现错误的位置。

设计测试用例是软件测试的主要工作之一,测试用例非常重要,它的作用包括以下几点。

(1) 有效性。软件测试是无法穷举进行的,采用具体的、有限的测试用例可以节约测试时间,提高测试效率。

(2) 可复用性。不同的测试人员执行相同的测试用例得到一致的结果,测试用例设计