

基于项目驱动的软件测试方案研究与实施

郝爱语

苏州工业职业技术学院, 中国·江苏 苏州 215000

摘要: 以资产管理系统的全过程测试为例, 详细阐述了功能测试、接口测试、单元测试、性能测试以及自动化测试等测试方案的设计与实施, 提出具体实训内容和实现步骤, 给出根据需求设计测试用例的部分范例, 根据测试执行结果最终得出测试报告的结论。

关键词: 项目驱动; 软件测试; 方案研究

Research and Implementation of Software Testing Scheme Based on Project-Driven Approach

Hao Aiyu

Suzhou Vocational Institute of Industrial Technology, China Jiangsu Suzhou 215000

Abstract: Taking the whole-process testing of an asset management system as an example, this paper elaborates on the design and implementation of testing schemes such as functional testing, interface testing, unit testing, performance testing, and automated testing. It proposes specific practical training content and implementation steps, provides some examples of designing test cases based on requirements, and finally draws conclusions from the test report based on the test execution results.

Keywords: Project-driven; Software testing; Scheme research

0 引言

ISO9000 定义测试是一种基于机器的, 对代码执行测试, 确认测试的活动。也有人认为, 软件测试是对软件质量的度量, 验证系统满足需求, 或确定实际结果与预期结果之间的区别; 确认程序正确实现了所要求的功能。以上定义说法不一, 包含了人们对软件测试不同方面和不同程度的理解, 各有侧重。有些定义强调测试过程中所做的事情, 有些定义侧重于测试的更一般的目标, 如评估质量、用户满意, 还有一些定义将重点放在预期结果上。从这些定义可以看出, 对软件测试的认识是一个由以单纯发现错误为目的到验证确认软件的功能特性、评估软件质量为目的的过程^[1]。

在产品测试中, 测试是使用特定的测试方法, 发现产品中的缺陷, 并检验产品是否满足规定的需求, 是保证产品质量的一种手段。测试已经广泛应用到社会的各行各业, 成为不可或缺的一种检测方法。不同的属性的产品使用不同的测试方法, 按被测试对象分, 有软件测试、硬件测试, 按功能分类可分为性能测试、兼容性测试、抗干扰测试等,

按测试工具角度分类可分为手工测试、自动化测试^[2]。

1 测试方案介绍

测试方案有别于测试计划, 测试计划是对测试全过程的组织、资源、原则等进行规定和约束, 并制订测试全过程各个阶段的任务以及时间进度安排, 并提出对各项任务的评估、风险分析和管理需求。用一句话概括就是: 测试计划是从管理角度对整个测试活动进行规划和控制。测试方案是描述被测对象需要测试的特性、测试的方法、测试环境的规划、测试工具的设计和选择、测试用例的设计方法、测试代码的设计方案。用一句话概括就是: 测试方案是从技术角度对整个测试活动进行规划和控制。测试方案需要在测试计划的指导下进行, 测试计划提出“做什么”, 而测试方案明确“如何做”。

2 测试项目介绍

资产管理系统符合一般企业实际管理需求, 能对企业的资产信息进行精确的维护, 有效服务, 从而减轻资产管理部门从事低层次信息处理和分析的负担, 该系统能解放管理员的双手和大脑, 可以提高工作质量和效率。系统采

用 B/S 结构，分布式数据库，包含个人信息、资产类别、资产借还等功能模块，有 Web 端和手机 APP 两个平台，Web 端包含超级管理员和资产管理两个角色，其中超级管理员主要维护一些通用的字典，资产管理维护部门、人员信息，并进行资产的日常管理；手机 APP 仅资产管理一个角色。功能见图 1 所示。

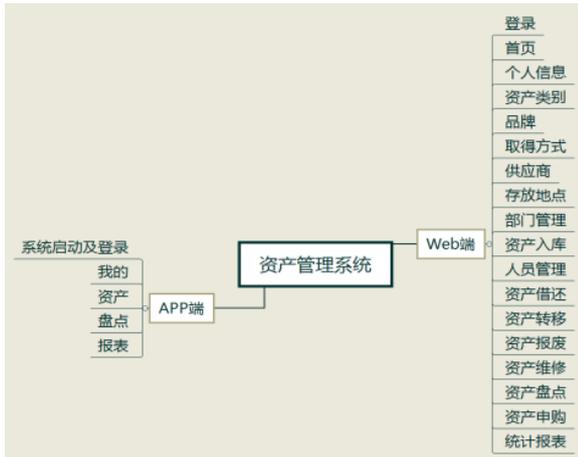


图1 资产管理系统的功能图

3 软件测试方案设计与实施

3.1 环境搭建与系统部署

测试环境是软件运行的平台，是进行软件测试所必需的工作平台和前提条件，一般来讲，

所需要搭建的环境主要是用于被测应用的系统测试，单元测试和集成测试由开发人员在开发环境中进行，而验收测试则在用户的最终应用环境中进行，因此都可以暂不考虑。资产管理系统测试环境实施步骤如下。

3.1.1 环境搭建

- (1) JDK 环境变量配置；
- (2) 查看 JDK 版本信息；
- (3) root 账号成功登录 MySQL。

3.1.2 系统部署

- (1) 初始化资产管理系统数据库命令；
- (2) 成功启动 Tomcat 服务；
- (3) 通过浏览器访问 Tomcat 主页；
- (4) webapps 目录；
- (5) 访问系统登录页。

3.2 功能测试

资产管理系统测试用例设计主要采用的是黑盒测试法，具体测试用例设计法有等价类划分法、边界值分析法、错误推测法、因果图法、场景图等。在测试策略上主要采用黑盒测试法进行手工测试，以需求分析文档中的功能模

块为单位，测试重点主要集中在数据输入、业务流程、数据输出和各功能模块之间的接口。主要进行了功能测试、界面测试、兼容性测试等。另外，测试采用了 UI 测试，即界面测试，测试用户界面的风格是否满足客户要求，文字是否正确，页面是否美观，文字，图片组合是否完美，操作是否友好等。部分功能测试用例见图 2 所示。

资产管理系统测试用例 (web端)								
测试用例编号	功能点	用例说明	前置条件	输入	执行步骤	预期结果	重要程度	执行用例测试结果
1、登录模块 (测试用例个数: 14个)								
Z00L-ST-SR3001-001	登录页面显示	登录页面正确性验证	正确显示登录页面	无	打开登录页面	1、页面显示文字和按钮文字显示正确； 2、图片正确显示； 3、正确显示用户名、密码输入框； 4、存在【登录】、【登一张】、【忘记密码】按钮；	低	通过
Z00L-ST-SR3001-002	登录测试	登录页面正确登录验证	1、正确显示登录页面； 2、任务ID已存在，用户名、密码匹配且已注册；	1、任务ID：已存在 2、用户名：ets001 3、密码：ets001 4、验证码：与图片一致	点击【登录】按钮	登录成功，进入首页。	高	不通过
Z00L-ST-SR3001-003	密码显示测试	密码输入后密码显示	正确显示登录页面	1、任务ID：已存在 2、用户名：ets001 3、密码：ets001 4、验证码：与图片一致	查看密码输入框	密码密文显示	高	通过
Z00L-ST-SR3001-004	登录测试	用户名错误 (为空)	1、正确显示登录页面； 2、任务ID已存在，用户名、密码匹配且已注册；	1、任务ID：已存在 2、用户名： 3、密码：ets001 4、验证码：与图片一致	点击【登录】按钮	提示：请输入您的用户名！	中	未执行
Z00L-ST-SR3001-005	登录测试	密码错误 (为空)	1、正确显示登录页面； 2、任务ID已存在，用户名、密码匹配且已注册；	1、任务ID：已存在 2、用户名：ets001 3、密码： 4、验证码：与图片一致	点击【登录】按钮	提示：请输入您的密码！	中	通过
Z00L-ST-SR3001-006	登录测试	用户名 (不存在) 错误	1、正确显示登录页面； 2、任务ID已存在，用户名、密码匹配且已注册；	1、任务ID：已存在 2、用户名：ets0 3、密码：ets001 4、验证码：与图片一致	点击【登录】按钮	提示：用户名不存在	中	通过

图2 资产管理系统测试用例范例

3.3 自动化测试

在企业内部通常存在许多不同种类的应用平台，应用开发技术也不尽相同，甚至在一个应用中就可能跨越了多种平台，或同一应用的不同版本之间存在技术差异，资产管理系统自动化测试部分脚本代码如下：

```

from selenium import webdriver
# 从 Selenium 中引入 WebDriver
import ddt
from time import sleep
import unittest
import csv
def read():
    data = csv.reader(open(r"D:\csv\csv4.csv","r"))
    i = 0
    list = []
    for row in data:
        if i != 0:
            list.append(row)
        i += 1
    return list
@ddt.ddt()
class TestA(unittest.TestCase):
def setUp(self):
self.driver = webdriver.Chrome()
# 使用 Selenium 模块的 WebDriver 打开谷歌浏览器
@ddt.data(*read())
def test_login(self, list):
self.driver.get("http://221.226.77.2/ams/front/login.do")
# 通过 get 方法发送网址打开资产管理系统登录页面
    
```

```

self.driver.find_element_by_xpath('//*[@id="fmedit"]/div[1]/label[2]/input').click()
# 点击超级管理员按钮
self.driver.find_element_by_name('taskId').send_keys('28')
# 输入任务 ID
self.driver.find_element_by_name('loginName').send_keys('stu001')
# 输入用户名
self.driver.find_element_by_name('password').send_keys('stu001')
# 输入密码
self.driver.find_element_by_tag_name('button').click()
# 点击“登录”按钮
self.driver.find_element_by_link_text('存放地点').click()
# 点击存放地点按钮,进入存放地点页面
self.driver.find_element_by_xpath('//*[@id="fmsearch"]/div/button').click()
# 点击“新增”按钮
sleep(3)
self.driver.find_element_by_xpath('//*[@id="title"]').send_keys(list[0])
# 输入存放地点名称
self.driver.find_element_by_xpath('//*[@id="assetTypeId"]').send_keys('其他')
# 输入存放地点类型
self.driver.find_element_by_xpath('//*[@id="remark"]').send_keys('阿斯蒂芬')
# 输入备注
self.driver.find_element_by_xpath('//*[@id="submitButton"]').click()
# 点击“保存”按钮
sleep(2)
self.driver.switch_to.alert.accept()
# 点击“确定”按钮
def tearDown(self):
    sleep(2)
    self.driver.quit()
# 退出系统
if __name__ == '__main__':
    data = read()
    for row in data:
        print(row)
unittest.main()
    
```

3.4 性能测试

在实时系统和嵌入系统中，提高符合功能需求但不符合性能需求的软件是不能被接受的。性能测试 (Performance Testing) 就是用来测试软件在集成系统中的运行性能的。性能测试可以发生在测试过程的所有步骤中，即使是在单元层，一个单独模块的性能也可以使用白盒测试来进行评估，然而，只有当整个系统的所有成分都集成到一起之后，才能检查一个系统的真正性能。目前，性能

测试技术创新驱动主要是引入人工智能与机器学习技术。利用机器学习算法分析历史测试数据，自动生成优化的测试用例，提高测试覆盖率；通过智能算法预测潜在性能问题，如预测系统在高并发下的响应时间变化，提前优化。某互联网企业应用 AI 技术后，性能测试效率提升 30%，缺陷发现率提高 20%^[1]。

性能测试是用于评估软件在运行时表现的一种测试方法，主要包括以下方面：

运行速度：测试软件在各种负载情况下的运行速度，如启动时间、响应时间、处理速度等。

系统资源占用量：测试软件在运行时所占用的系统资源，如内存、CPU、磁盘等，以确保软件在各种负载情况下都能正常运行。

系统响应时间：测试软件在各种负载情况下的响应时间，包括网络响应时间、用户界面响应时间等，以确保软件能够及时响应用户的操作^[4]。

以资产管理系统性能测试举例，采用 LoadRunner 工具的测试步骤如下：

(1) 虚拟用户脚本生成器 (virtual user Generator) VUG 在测试过程中，利用 VUG 进行：录制脚本，回放结果查看，编辑测试脚本；

(2) 压力调度控制台 (controller) 在测试过程中，利用 controller 进行：创建场景，运行场景，监控场景，收集测试数据；

(3) 压力结果分析器 (Analysis) 在测试过程中，利用 Analysis 把收集到的测试数据以图表的形式展示出来，生成测试报告 (见表 3、表 4)。

表3 “资产借还”性能测试用例设计

压力点名称	C_JH	脚本名称	C_JH
步骤	操作	是否设定事务	事务名称
1	输入URL地址并打开资产系统	否	无
2	输入用户名，密码，点击登录，登录成功	否	无
3	点击资产借还，进入资产借还管理页面	否	无
4	点击借用登记按钮，进入资产借用登记窗口	否	无
5	输入资产名称，使用人，借用日期，借用原因，点击提交，提交成功	是	T_JH
6	退出系统	否	无

表4 “资产报废”用例设计

压力点名称	C_BF	脚本名称	C_BF
步骤	操作	是否设定事务	事务名称
1	输入URL地址并打开资产系统	否	无
2	输入用户名, 密码, 点击登录, 登录成功	否	无
3	点击资产报废, 进入资产报废管理页面	否	无
4	点击报废登记按钮, 进入资产报废登记窗口	否	无
5	输入资产名称, 申请人, 报废日期, 报废方式, 报废原因, 点击提交, 提交成功	是	T_JH
6	退出系统	否	无

3.5 回归测试

软件在测试或其他活动中发现的缺陷经过修改后, 应该进行回归测试, 目的是验证缺陷得到了正确的修复, 同时对系统的变更没有影响以前的功能回归测试可以发生在任何一个阶段, 包括单元测试、集成测试和系统测试。

在回归测试中, 不仅对核心业务流程进行测试, 还对软件的边缘和异常路径进行探索性测试, 如输入特殊字符、进行异常操作等。同时, 验证新输入数据在所有显示位置的正确性, 确保数据的一致性和完整性^[5]。回归测试的策略包括:

(1) 完全重复测试: 重新执行所有在前期测试阶段建立的测试用例, 来确认问题修改的正确性和修改的扩散局部影响性。

(2) 选择性重复测试: 即有选择地重新执行部分在前期测试阶段建立的测试用例, 来测试被修改的程序。

回归测试是一个重用以前成果的测试, 很难预料到要经过多少次回归系统才能达到满意的水平, 因此, 理想的回归测试一般是采用自动化手段实现。回归测试的自动化法包括测试程序的自动运行、自动配置, 测试用例的管理和自动输入, 测试的自动执行, 测试信息与结果的自动采集, 测试结果的自动比较和结论的自动输出。对系统测试功能比较简单、测试界面相对稳定并且测试用例良好组织的测试来说, 采用“捕捉回放”工具是比较合适的, 这类工具有 QTP、Robot、SilkTest 等。为了实现测试用例的自动化并实现测试结果的自动判断, 脚本化的、包含控制结构、内部实现结果判断的测试用例是唯一的选择, 此类脚本语言有 TCL、Python、Perl 等。对于特定系统的、复杂的测试来讲, 如果没有通用的商用工具可供选择, 探索开发专用的自动化测试工具是灵活且易于扩充的方法。回归测试的自动化的问题是一个需要尽早考虑的问题, 在做测

试方案时就要考虑这种可能性, 必要时投入资源进行开发, 能形成可供继承与推广的工具则是最终目的。

3.6 测试报告

资产管理系统测试从功能性, 易用性, 兼容性等多个方面进行测试, 根据测试计划方案, 对系统的各个模块进行了相关测试, 主要包括功能测试、性能测试、白盒测试、自动化测试等环节。测试团队共为资产管理系统设计编写了 1253 个用例, 其中, Web 端测试严格按照测试用例执行, 一共有 1035 个用例, 共执行了 1035 个用例, 共通过了 973 个用例, 未执行 35 个用例, 未通过 27 个用例; 手机 APP 测试严格按照测试用例执行, 共有 218 个用例, 共执行 218 个用例, 共通过了 190 个用例, 未执行 10 个用例, 未通过 18 个用例。先后共发现了 221 个 Bug, 其中严重的级别有 4 个, 很高的级别有 4 个, 高的级别有 47 个, 中的级别有 107 个, 低的级别有 59 个。

4 结语

随着人工智能技术快速发展, 高效率软件测试工作需要更加合理、行之有效的测试方案, 方能解决诸如资产管理系统等软件因功能模块增加、开发周期延长以及多团队协作等问题, 从而保障软件的高质量高品质。本文基于项目驱动, 围绕实际案例的全流程测试设计与实施, 创新性提出基于功能测试、性能测试、自动化测试等多维度测试研究方案, 并细化为具体实施策略, 显著降低了软件缺陷发生概率, 同时也形成了一套标准化测试规范, 为软件质量的提升提供了有效途径。

参考文献:

- [1] 郭雷. 软件测试[M]. 北京: 高等教育出版社, 2022:2.
- [2] 吴刚. M 公司空调电控研发测试方法改善研究[D]. 广州, 华南理工大学, 2023.
- [3] 倪勤文, 宋文杰. 软件性能测试策略与方案的研究[J]. 质量与认证, 2025(6):65-68.
- [4] 袁飞晖, 高文杰. 远洋客船网络化服务平台软件系统测试方案设计[J]. 船舶物资与市场, 2023(5):99-102.
- [5] 马艳梅, 邓旭平, 牛璐琳, 李树政. 零碳智慧园区管理软件测试方案研究与效果评估[J]. 智能建筑与智慧城市, 2025(6):118-120.

基金项目: 苏州工业职业技术学院 2023 年度“软件质量保证与测试”精品课程(2023ZX07)。

作者简介: 郝爱语(1980-), 女, 江苏苏州人, 副教授, 硕士研究生, 研究方向: 软件测试、数据库开发、人工智能。