测试之功能测试

第一章 测试理论

1.1 软件测试的概念

Software Testing软件测试
在规定的条件下对程序进行操作（人工测试、自动化测试），以发现程序错误，衡量软件质量，并对其是否能满足需求进行评估。

软件测试目的
有特定的目的，有特定的方法，提交错误信息，跟踪错误信息，发现软件问题，检查系统是否满足用户需

软件测试

测试是检验软件是否符合用户需求，是否达到软件质量标准，由专门测试部门执行。

常见测试活动分为：

- 单元测试，对每一个代码函数测试
- 集成测试，对代码组成的模块进行测试
- 系统测试，针对每一个功能需求，完整的设计测试环境，指派测试人员执行

软件运维

此阶段软件已经交付上线提供用户使用，进入维护阶段，保证软件系统正常运转，如网站的7*24小时运行，因为软件系统可能出现如下问题：

- 由于用户增长造成的性能压力
- 服务器出现问题造成系统不可用
- 软件系统出现bug
- 系统升级出现未知bug

用户在使用过程中出现问题，反馈给技术支持人员，然后再递交给测试组，由研发组修复。

1.2 软件测试的流程

1. **业务流程相关人员**
    - **开发工程师**：修补bug
    - **产品经理**：熟悉产品业务流程，与其沟通
    - **客服人员/业务人员**：bug反馈给测试，进行bug验证
    - **实施人员/技术支持工程师**
    - **设计师**：阅读需求/理解需求，学习业务
2. **需求分析**
    - 整理需求点
    - 有疑点需要开会讨论，直至解决
3. **测试计划**
    - **测试文档内容**：目的性，期限性
    - 参与测试计划的人员：小王、小李、小张
    - **任务划分**：谁负责哪个功能模块，测试用例的编写
    - **时间规划**：测试用例的准备时间，产品上线测试时间
    - **文档生成**：测试用例，bug表单，软件测试报告
    - **资源申请**：服务器数量，测试类型的划分，所需的资源工具
4. **测试设计阶段**
    - 编写测试用例，参考需求文档、概要设计、详细设计文档，测试的覆盖率、产品稳定性、易用性等进行评估
5. **测试执行阶段**
    - 搭建测试环境，测试&验证，提交缺陷管理平台，并且对bug进行跟踪，直到测试过程中发现的bug解决，没有重现bug，测试结束
6. **测试评估阶段**
    - **置问题单**，通过继续，不通过打回

- **对整个测试过程和版本质量做出评价，确认是否可以上线**

7. **测试通过，上线**
    - 测试不通过，打回重新测试

1.3 软件测试的工作流程

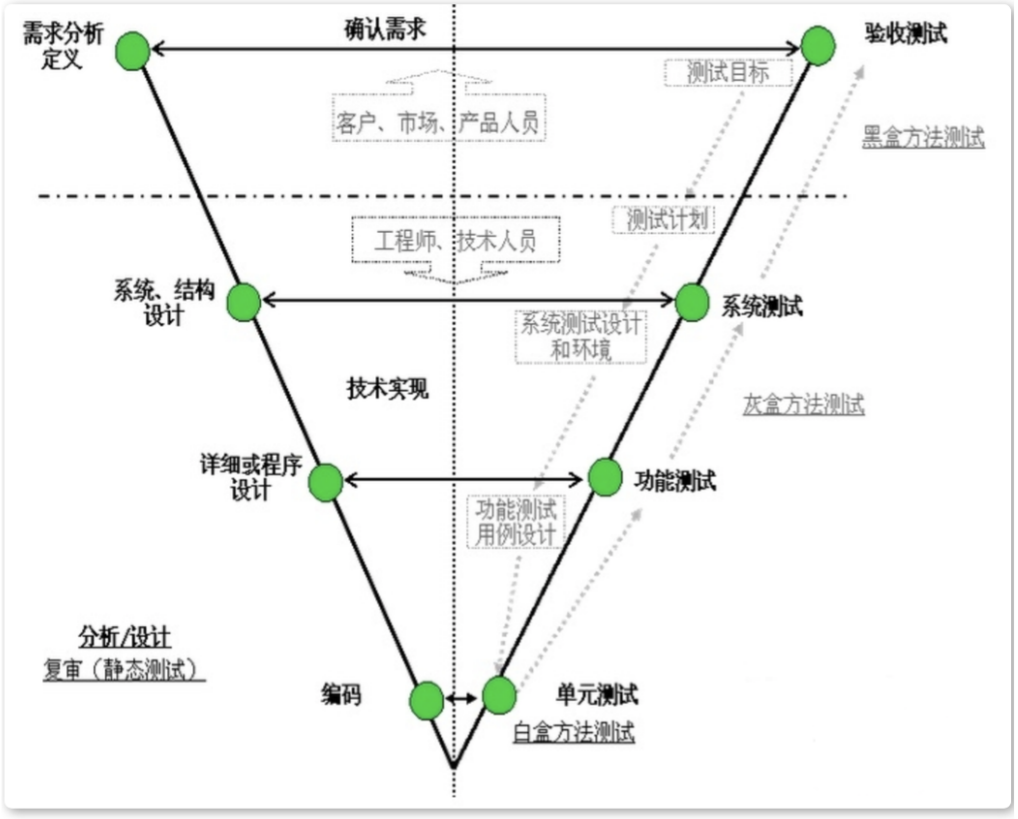
1. **业务流程相关人员**
    - **开发工程师**：修补bug
    - **产品经理**：熟悉产品业务流程，与其沟通
    - **客服人员/业务人员**：bug反馈给测试，进行bug验证
    - **实施人员/技术支持工程师**
    - **设计师**：阅读需求/理解需求，学习业务
2. **需求分析**
    - 整理需求点
    - 有疑点需要开会讨论，直至解决
3. **测试计划**
    - **测试文档内容**：目的性，期限性
    - 参与测试计划的人员：小王、小李、小张
    - **任务划分**：谁负责哪个功能模块，测试用例的编写
    - **时间规划**：测试用例的准备时间，产品上线测试时间
    - **文档生成**：测试用例，bug表单，软件测试报告
    - **资源申请**：服务器数量，测试类型的划分，所需的资源工具
4. **测试设计阶段**
    - 编写测试用例，参考需求文档、概要设计、详细设计文档，测试的覆盖率、产品稳定性、易用性等进行评估
5. **测试执行阶段**
    - 搭建测试环境，测试&验证，提交缺陷管理平台，并且对bug进行跟踪，直到测试过程中发现的bug解决，没有重现bug，测试结束

6. **测试评估阶段**
    - **置问题单**，通过继续，不通过打回
    - **对整个测试过程和版本质量做出评价，确认是否可以上线**
7. **测试通过，上线**
    - 测试不通过，打回重新测试

1.4 软件测试模型

随着软件测试的发展，测试人员在大量实践中总结出一些测试模型，对测试活动进行抽象，如V模型和瀑布开发模型有着一些共同的特性，V模型中的过程从左到右，描述了基本的开发过程和测试行为。
V模型的价值在于它非常明确地标明了测试过程中存在的不同级别，并且清楚地描述了这些测试阶段和开发过程期间各阶段的对应关系。
局限性：把测试作为编码之后的最后一个活动，需求分析等前期产生的错误直到后期的验收测试才能发现

V模型



模型部分解释

单元测试

也叫做模块测试，进行正确性检查的工作。

单元测试要从程序内部结构设计测试用例，每个模块可以独立进行测试。

单元是什么?

比如Python中的类

图形化软件的一个窗口，一个功能菜单

集成测试

也叫组装测试，也就是将所有单元测试，进行有序的组合测试

系统测试

将整个软件系统看做一个整体进行测试，对功能，性能，硬件，软件所有环节进行测试。

单元测试目的测试功能是否满足要求。

系统测试目的测试系统是否满足性能的要求，以及不同的机器系统平台中运行，如一

台机器我登陆多个qq，是否可
以运行，在windows，linux平台运行qq是否正常等方面。

验收测试
α测试
Alpha测试模拟实际操作环境下验收测试，如删档内侧，软件只是初步完成的产品，bug可能较多，不会进行上线提供用户访问。
β测试
Beta测试系统已经通过内部测试，大部分错误已经修复，即将正式发布，在多个真实环境下发布，如不删档公测。
对比α版本已经有了较大改进，但仍可能存在一些bug，需要大规模测试，例如DNF公司更新一个地图，提供公测
免费下载，由专业游戏玩家进行游戏结果反馈，开发者啊、再进行修复。
y版本
y版本指的是正式版本，与上线版本几乎无区别。

软件测试的W模型

W模型优点:

测试伴随整个软件开发生命周期,更早接入测试,可以更好的发现需求设计中的缺陷,修复成本也更低.

W模型的缺点

依赖于软件开发和测试的前后线性关系,还是无法解决需求变更调整的问题.对于项目中不产出文档的事例,测试工作无法执行

对于复杂业务,新人不足以测试需求,测试设计.

测试覆盖率

覆盖率是用来度量测试完整性的一个手段,同时也是测试技术有效性的度量

覆盖率=(至少执行一次的项目点)/项目总数

特点:

通过覆盖率数据,检验测试工作是否充足

分析测试的弱点,对比测试人员的测试结果

第二章 测试分类

功能测试 —测试人员用鼠标去手动测试(测试GUI),点点点测试

自动化测试-用程序测试程序(测试API),解放双手

性能测试-定位系统瓶颈,保证系统良好性能与稳定性

黑盒测试-测试应用程序的功能,验证结果是否正确

白盒测试-测试程序内部结构,以编程语言角度设计测试案例

2.1 软件测试功能分类

按照测试阶段划分

单元测试 Unit Testing

集成测试 Integration Testing

系统测试 System Testing

单元测试

什么是单元?内存条可以理解为是一个单元,一个功能小模块

单元测试是针对基本单元(软件设计最小单位)进行正确性检测工作.

单元测试目的是检测软件模块是否完成了需求设计书中的要求.

集成测试

集成测试是机遇单元测试的基础,将所有模块按照设计组装为一个子系统,验证

系统测试

结合测试工作人员安排,测试环境配置,计算机硬件环境配置,整合在一起,进行测试工作

单元、集成、系统测试的区别

测试方法不同:

单元测试属于白盒测试范畴

集成测试属于灰盒测试范畴

系统测试属于黑盒测试范畴

测试范围不同

单元测试检查单元内部数据结构、逻辑控制、异常处理等

集成测试检查模块之间的数据传输

系统测试检查整个软件是否满足了需求

2.2 回归测试

测试过程信息流

软件环境配置:需求文档、需求设计书、测试代码等

测试环境配置:测试需求说明书,测试计划书,测试用例等

回归测试定义

软件在测试或者其他活动中发现了缺陷且经过修复之后,应该进行回归测试活动

目的在于验证缺陷是否正确地被修复,并且是否影响到其他的功能

回归测试流程

1.制定测试策略,回归测试策略

2.确定回归测试版本

3.根据回归测试策略执行回归测试

4.回归测试通过,关闭缺陷跟踪单(禅道)

5.回归不通过,缺陷跟踪单返回给开发人员,重新修复问题,再次回归

回归测试策略设计

完全重复测试

重新执行所有先前定义的测试用例,来确认问题修改的正确性,以及软件修改后可能造成的影响扩散.

选择性重复测试

有选择性的重新选择执行部分测试用例

覆盖修改法,针对被修改的部分,重新构造测试用例验证已经没有缺陷

影响扩散法,再覆盖已修改的部分之上,分析其由于修改后是否间接造成了其他的额外缺陷

指标达成法,确定要达成的测试目标,例如测试用例覆盖率的百分比

回归自动化测试

由于回归测试的特性,是重新执行以前的成果,且无法预知需要回归几次后才能通过测试标准,因此回归测试可能由于枯燥乏味造成测试人员积极性低下,测试质量下降

因此需要自动化回归测试

良好的回归自动化测试活动包括如下:

程序自动化运行

程序参数自我配置

测试用例自动管理

测试活动自动执行

测试信息日志自动收集

测试结果自动分析与输出

实现测试用例自动化且进行结果分析,此类活动适合脚本化用例开发,进行逻辑控制,结果断言等,使用Python进行脚本开发灵活性较高

对于特定系统或者复杂测试环境下,商业测试工具难以解决问题,自主研发自动化测试工具是灵活的方法.

2.3 验收测试

在软件生命周期流程上来看,软件研发阶段,测试活动包括单元测试、集成测试,系统测试等企业内部进行的测试活动

在软件发布之前还有一个测试活动有真实用户介入,称为验收测试

软件项目开发分为两种形式:

企业自研产品

自己公司产品上线前由用户介入进行大范围内测,公测,检查软件正确性

α测试

β测试

公司承接业务开发,客户要什么就开发什么

产品研发好后,由客户方验收,检验产品需求正确性

验收测试概要

如何确定验收测试的执行时间、执行人员、执行环境

在通过内部系统测试活动以及软件配置审查之后,开始执行验收测绘

验收测试是以人为测试活动为主,由测试人员和用户群体为代表开展测试活动

验收测试原则上是在用户基地进行,在用户同意下,也可以在软件开发公司模拟测试环境

验收测试以 验收测试计划书或软件需求规格书进行标准化测试

验收测试结果分为两种:

软件功能、质量、性能、界面特性与用户需求一致,用户接纳软件结果

反之,软件不被用户接纳

2.4 α测试β测试

α测试活动好比游戏的删档内测

由企业内部员工在模拟真实环境下进行的测试活动,进行结果反馈

员工在开发人员的接入下,通过指导进行软件使用,且随时记录下使用过程中的错误

由真实用户在软件测试内测版本,开发版本环境下进行的产品体验/测试活动,进行结果反馈

α测试的目的是评价软件产品的FLURPS(即功能、局域化、可用性、可靠性、性能和支持)

α测试即为非正式验收测试,尤其注重产品的界面和特色

β测试好比游戏的不删档公测

由多个用户在实际操作环境下进行产品功能测试,如不同的手机型号,平台,网络环境等

β测试不受开发者控制,自主开展测试活动,然后进行结果反馈

2.5 测试划分

测试分为四大阶段

测试计划,测试方案,测试用例,测试报告

生命周期各测试方法对比

	单元测试	冒烟测试	集成测试	系统测试	验收测试
测试阶段	编码后	提测后	单元测试后	冒烟测试通过后	软件发布前
测试对象	最小模块覆盖	整个系统	模块间联调	整个系统	整个系统
测试人员	白盒测试或开发	黑盒测试	白盒测试或开发	黑盒测试	用户或需求方
测试依据	代码、注释、详细设计文档	冒烟测试用例	单元测试模块、概要设计文档	需求说明文档、测试用例	用户需求、验收标准文档
测试方法	白盒测试	黑盒测试，自动化测试	黑盒与白盒结合	黑盒测试	黑盒测试

软件测试分类说明

名称	说明
性能测试	性能测试是为验证系统性能指标进行测试。
负载测试	负载测试是通过改变系统负载方式来发现系统中所存在的性能问题，如内存泄露，性能瓶颈。
压力测试	在高负荷且长时间的稳定性压力测试和极限负荷情况下进行测试。验证系统稳定性。
恢复测试	检查系统的容错能力。强制性迫使系统失败，然后验证系统恢复能力重启系统。
易用性测试	测试软件是否易用性，一般根据用户反馈信息验证。
回归测试	指错误被修正后，软件功能发生变化后重新测试，确认不会影响其他功能。
Alpha测试	模拟实际操作环境下验收测试，如删档内侧
Beta测试	系统已经通过内部测试，大部分错误已经修复，即将正式发布，在多个真实环境下发布，如不删档公测。

2.6 黑盒测试

- 黑盒称为功能测试，一是按顺序检测程序每个功能，二是按功能模块优先级测试。
- 对软件的界面和功能进行测试，只考虑其整体特性，不考虑其内部实现
- 需要根据说明书、用户手册进行功能测试
- 如电视遥控器，就是一个标准的黑盒测试，我们不管是液晶的还是其他方式，只需要查看
  遥控器是否能控制空调
- 要求多组数据，多次测试才能得到准确的报告

黑盒测试类型

- 功能性测试，网站顺序使用的功能，手机使用功能，显示器是否正常显示...
- 容量测试，海量数据的处理，如1亿人用微信，1亿人用支付宝，这个数值容量
- 负载测试，系统在短时间内处理海量数据时，系统的健康状况指标
- 恢复性测试,小米手机秒杀活动太过热火,网站挂了,不该影响用户数据

2.7 白盒测试

白盒测试就打开了盒子，可以看到软件的内部代码
对比黑盒测试，白盒测试更严谨，对软件的源码和架构进行测试。
需要软件源代码，流程图等设计文档。
依据被测软件分析程序内部构造，并且根据内部结构设计测试用例，针对内部控制流程进行测
试
白盒测试是基于程序结构的逻辑驱动测试

黑盒,白盒测试,相辅相成,白盒测试通过了,还得运行软件,查看软件的性能好坏,界面美丑,是否易用等等方面

2.8 软件测试方法选择

已知产品需求规格,但不知道其系统内部实现,进行黑盒测试证明需求是否实现

已知产品内部实现,通过测试每种内部操作是否符合需求,属于白盒测试

为什么要白盒测试

只验证产品基本需求得到实现,为什么要费力去测试产品内部实现呢?仅仅黑盒测试,只能够满足一定的覆盖率,不能够消除更多隐藏的缺陷.

而白盒测试能够从内部逻辑检测,检测覆盖率更高,给予软件质量更多的保证

白盒测试特点

理解软件需求

了解软件代码实现

检测代码逻辑

检测代码中文件路径

白盒测试成本较高

白盒测试的方法

静态分析:控制流分析,数据流分析,信息流分析

动态分析:逻辑覆盖测试(分支测试、路径测试)、程序插装

逻辑覆盖测试

语句覆盖

判定覆盖

条件覆盖

路径覆盖

逻辑覆盖率的统计通过程序插装可以体现

程序插装

调试代码时候,往往会在程序中print()语句,能够很方便的打印出我们想要的信息,进一步了解程序执行的一些动态信息,比如程序执行顺序,计算后的变量的具体指等.

程序插装技术能够按照用户的要求,获取程序部分信息,是测试工作的有效手段.

黑盒测试的优点

1. 比较简单，不需要了解程序内部的代码及实现;
2. 与软件的内部实现无关;
3. 从用户角度出发，能很容易的知道用户会用到哪些功能，会遇到哪些问题;
4. 基于软件开发文档，所以也能知道软件实现了文档中的哪些功能;
5. 在做软件自动化测试时较为方便。

黑盒测试的缺点

1. 没有清晰的测试规格说明，测试用例难以设计
2. 自动化测试的复用性较低。
   3. 无法控制程序内部路径，可能缺少较多测试点

白盒测试的优点

1. 帮助软件测试人员增大代码的覆盖率，提高代码的质量，发现代码中隐藏的问题。

白盒测试的缺点

1. 程序运行会有很多不同的路径，不可能测试所有的运行路径;测试基于代码，只能测试开
   发人员做的对不
   对，而不能知道设计的正确与否，可能会漏掉一些功能需求;系统庞大时，测试开销会非常
   大。

灰盒测试

介于白盒和黑盒之间,针对被测试对象信息的不同,采用不同的测试方法

检验被测试对象整体特性信息,采用黑盒测试

检验被检测对象内部实现,采用白盒测试

既要观察被测对象整体信息,又要观察内部具体信息,信息占比不同,这就是所谓灰盒测试

软件测试文档

软件研发如同工厂生产,整个过程会有产品输出,输出的产品有两类

编辑好的代码程序,如QQ.exe

用户使用手册,QQ使用手册

源代码,QQ源代码

1.需求分析:软件测试依据

2.概要设计:根据需求分析的内容,进行方案设计,明确是否覆盖到软件的需求

3.详细设计:包含方案、策略、架构、体系、接口名、接口文档、数据结构算法,保证开发过程细致

4.测试设计:测试需要进行那些内容,包含哪些功能点,是否会影响到其他功能,例如第三方支付页面跳转

5.测试用例:用例是进行测试的依据,测试的规范条文

6.测试报告:测试成功数量、失败数量、测试是否通过,允许产品上线等

动态测试与静态测试

静态测试:不运行被测试的软件测试,采用例如代码阅读、文档评审、代码分析等方式进行软件测试

动态测试:按照预先规划好的测试数据与流程执行被测软件系统

静态分析技术

定义:静态分析是不执行程序而分析程序的技术

功能:检查软件功能与需求是否一致,没有明显缺陷与歧义

测试点:

程序程序代码语法上是否一致性和完整性

文档描述是否规范、精准、便于查阅

程序与文档之间描述的一致性

动态分析技术

对软件系统进行数据驱动运行,与期望结构进行对比检查的过程

脚本录制工具,进行反复测试,回归测试,如Robot/QTP/Selenium工具

性能测试工具,LoadRunner、Robot等

第三章 测试需求

3.1 软件测试需求

软件需求分析(Software Requirement Analysis)是研究用户需求的产物,完全理解用户对软件需求的功能,确认用户软件功能需求,建立可确认,可验收的基本依据.

软件测试需求是设计测试用例的依据

保证测试质量

衡量测试覆盖率的指标

总而言之就是,明确怎么测,何时测试,多少人力,多少物力

软件测试需求分析

依据研发的软件产品类型,需求来源,产品用户群体,进行不同对象的需求分析

针对特定产品的研发

需求来源:市场分析、市场调研

用户群体:市场调研人员

需求特点:根据市场动态,研发主流产品

针对特定市场的研发

需求来源:客户需求

用户:指定的甲方

需求特点:根据市场动态,研发主流产品

针对特定项目的研发

需求来源:客户需求

用户:指定的甲方

需求特点:客户想要啥,我们做啥

测试需求分析的设计

依据需求文档提取测试点,根据测试点来编写测试用例

测试要点分析:

正确性、必要性、优先级、明确性、完整性、可修改性、一致性

1.通过分析需求描述中的输入、输出、处理、限制、约束等,给出对应的验证内容,也被称为功能测试

2.通过分析模块之间的业务顺序,各个功能模块之间传递的信息和数据,对存在交互功能的功能项,给出对应验证.

结果,称为功能 交互测试

3.考虑到需求的完整性,充分覆盖软件需求的各种特定,包含隐性需求的验证,如界面的验证、注册账号的唯一性

验证,测试角度是界面、易用性、兼容性、安全性、性能压力

第四章 测试分类

4.1 测试计划是什么

测试计划是一个叙述了预定的测试活动的范围、途径、资源以及进度安排的文档

此文档确认了测试项、被测试特征、测试任务、人员安排,以及任何偶发事件的风险.

通过收集项目与产品相关的信息，对测试范围，测试风险进行分析，对测试用例，工作量，资源和时间进行估算，对测试采用的策略、方法、环境、资源、进度做出合理安排。
##项目成功的要素
如下四点：时间、成本、范围、质量时间由整个项目计划覆盖
成本由合同覆盖，甲方定制
范围由需求文档覆盖
质量：由QA计划或测试计划覆盖
##为什么要指定测试计划不以规矩不成方圆
定制测试计划使得软件测试是有计划，有组织的软件质量保证活动。如果没有计划，工作就会很松散，随意。

4.2 测试策略

整体测试策略

- 使用哪些测试方法？执行黑盒测试，是否需要白盒，自动化
- 规定各个阶段的测试工作重心

测试开始/中断/完成的标准

- 符合什么标准，可以进行测试
- 符合什么标准，测试必须中断或暂停
- 符合什么标准，测试可以结束测试类型定义
- 功能测试
- 安装卸载测试
- 兼容性测试
- 易用性测试

测试技术

- 黑盒测试工具
- 白盒测试工具
- 自动化脚本编写

第五章 缺陷管理

5.1 什么是缺陷

软件中任何不满足用户需求的问题,都可以识别为软件缺陷

从产品内部看,缺陷是产品开发或者维护中存在的错误,异常等问题

从产品外部看,缺陷是违背了某种功能的实现

5.2 缺陷识别方法

用户体验差,进行反馈

界面上有明显的错误信息

功能缺失,与需求说明书不符合

程序运行崩溃,停止运行

程序逻辑不正确,与用户使用手册不符

程序性能差,不能够承载压力访问

5.3 缺陷出现的原因

需求出现的原因

需求不断变化

工期短,软件复杂

编码中出错

设计文档不完善

沟通交流少

软硬件不支持

5.4 常见软件缺陷术语

- bug 泛指软件程序的漏洞和缺陷。
- Debug 调试（英语：Debug）是发现和减少计算机程序或电子仪器设备中程序错误的一个过程。
- 错误（mistake）人为造成软件的故障
- 缺陷（bug&defect）软件产品中存在的错误
- 故障（Fault） 某个组件功能不能完成所有任务的一个意外情况
- 失效(Failure)软件功能完全损坏,无法使用

5.5 缺陷的类型

A.功能遗漏

规定的时间或者功能未再=在产品系统中体验,如缺少忘记密码功能

B.程序错误

没有按照用户需求正确实现,可能需求理解出错,可能编码出错

C.额外的功能

5.6 缺陷的重现

可以重现的缺陷，存在一系列明确的重现操作步骤，条件和数据使得缺陷可以稳定的反复出现。
不可复现的缺陷，无法找到明确的步骤，缺陷出现是随机的，只做记录，不做报告。
不可重现的缺陷，后续再深入的挖掘，尝试转为可再现的缺陷，再进行缺陷报告。

5.7 缺陷报告

缺陷报告指的是测试执行过程中，发现软件缺陷，进行书写记录的文档，提供给开发人员或者测试负责人作为定位缺陷的依据，也用作缺陷数量统计的重要依据。
提供准确、完整、简洁的缺陷报告是软件测试专业性、高质量的评价标

5.8 撰写缺陷报告5C原则

5c原则

- Correct（准确）：每个组成部分的描述准确，不会引起误解；
- Clear（清晰）：每个组成部分的描述清晰，易于理解；
- Concise（简洁）：只包含必不可少的信息，不包括任何多余的内容；
- Complete（完整）：包含复现该缺陷的完整步骤和其他本质信息；
- Consistent（一致）：按照一致的格式书写全部缺陷报告。

缺陷报告结构

- 报告的缺陷信息具体、准确
- 缺陷特征描述与复现缺陷步骤
- 缺陷类型以及严重级别
- 缺陷标题
- 缺陷基本信息
- 测试环境描述
- 缺陷严重程度

撰写缺陷报告注意点

- 按照缺陷发生的因、果进行书写
- 避免模糊不清的词语，比如 功能不对、登录不起作用，应该具体描述 某一个功能如何不对、登录具体流程如何不正确
- 中立评价，公正表达自己对软件缺陷的理解，对软件错误以及造成的事实进行描述

5.9 常见软件缺陷

编号	问题描述
中文英统一	不要使用中英文混合提示，不要去挑战用户的英文能力..
容错性	例如用户注册，需要限制手机号长度，年龄范围等，输入错误需要有醒目提示
用户体验	比如某高校学生信息登记网，填写一堆信息后，由于一个信息填写错误，内容全部被清空，还得重新输入，用户体验极差
兼容性	需要考虑操作系统、浏览器类型、版本，网络类型
错别字	例如网站的"登录"写成"登陆"
安全性	注意SQL注入，XSS攻击
UI友好度	比如删除、保存按钮离得太近，用户手指头难以正确点击...
.....	未完待续

模块名称	注册页面		
版本号	V1.2	测试人	
缺陷类型	功能错误	严重级别	B
可重复性	是	缺陷状态	New
测试平台	Win2000 Professional	浏览器	IE 10
简述	系统规定注册用户名的长度为6-20字符。用少于6个字符的用户名可以成功注册		
操作步骤	1. 进入GoGo求职招聘网站首页 2. 点击免费注册会员，进入用户注册页面 3. 单击“同意”按钮，进入用户注册页面 4. 在用户名文本框中输入“abc” 5. 正确输入其它信息 6. 单击“提交”按钮，提示注册成功		
实际结果	注册成功		
预期结果	提示用户名错误，不能注册成功		
注释	建议修改		

5.10 bug是什么

bug泛指软件程序的漏洞和缺陷。

值测试工程师或用户发现与提出的，软件可以改进的细节部分、或者与需求文档存在功能偏差的实现。

测试工程师职责就是发现bug，提交bug信息给研发人员，研发人员修复bug。

5.11 bug案例

例如登录时，要求输入账号密码。

输入正确的账号密码：

结果提示：用户名不存在/密码

再三确认账号密码是否错误，可以重新再注册一个账号进行登录如新账号也是账号不存在，此登录已经是bug了！

5.12 bug的类型

想要确定bug的类型，需要对产品有较深的理解。

对bug定义划分如下：

- 代码错误（指的是研发代码有误，功能未实现）
- 设计缺陷
- 界面优化
- 性能问题
- 配置相关
- 安装部署
- 安全相关
- 标准规范
- 测试脚本
- 其他（功能类、界面类、性能类、易用性类、兼容性类、其他）

5.13 bug严重程度

顾名思义就是软件缺陷对软件质量造成的破坏程度，将会给软件使用带来怎样的影响。

Bug等级越高，可能被修复的等级也越高，公司也会根据测试提交的bug数量以及bug等级作为绩效考核标准。

判断bug的等级，如下分类：

1.致命错误

- 常规操作缺引起系统崩溃、死机、死循环
- 造成数据库泄露的安全问题，如恶意攻击造成的账户私密信息泄露
- 涉及金钱隐患，金钱计算bug（如金额不足，还可以购买产品，对公司金额造成重大损失

2.严重错误

- 重要功能未实现，如点击注册无反应，无法登录
- 非常规操作（误操作）导致程序崩溃、死机、死循环
- UI界面的严重问题
- 密码明文显示，数据库泄露

3.普通错误

- 不影响产品运行，不会影响故障，但对产品外观界面影响较大，如删除了好友，好友却未消失
- 功能无法正常显示功能
- 操作界面错乱
- 查询数据错误
- 页面未做输入格式限制
- 删除错误未给与提示

4.错误提示

- 网页UI错误，
- 错别字
- 标点符号

5.14 bug处理优先级

优先级（Priority）指的是缺陷被修复的紧急程度。

- 立即解决：软件缺陷已导致软件系统崩溃，需立即修复
- 高优先级：缺陷比较严重，已经影响用户正常使用
- 普通优先级：缺陷排队，等待修复
- 低优先级：等待开发人员闲时进行修复，缺陷影响很小

5.15 常见bug管理状态

在不同的缺陷管理系统中，对bug的标记状态有如下种类：

- New 缺陷初始状态
- Open 缺陷修复中
- Fixed 缺陷修复完毕
- Closed 缺陷已通过回归测试，关闭缺陷
- Reopen 缺陷未通过回归测试，重新修复
- Postpone：推迟修改缺陷
- Rejected： 开发拒绝修复缺陷

- Duplicate：重复提交的缺陷
- Abandon：开发与测试经过讨论，确定不是缺陷，可以标记缺陷关闭

5.16 测试准入标准

- 开发编码结束，并且在开发环境完成单元测试
- 开发需求规定的功能均实现，如没有完全实现，请提供测试范围
- 已经完成集成测试，被测系统的基本流程已经走通，界面上功能均实现，经过代码评审并符合软件编码规范
- 开发提交最新版本代码，告知测试组进行测试
- 测试环境兼容性确认

5.17 测试准出标准

序号	准出标准	是	否	时间
1	被测项目是否满足需求原型?			
2	所有测试用例是否通过评审?			
3	所有测试用例都已执行?			
4	测试覆盖率是否100%?			
5	所有发现的软件缺陷是否记录在缺陷管理系统（禅道）?			
6	一二级缺陷修复率是否100%?			
7	三四级缺陷修复率是否95%?			
8	所有遗留问题是否有解决方案?			
9	性能指标是否达标?			
10	兼容测试是否达标?			
11	安全性测试是否达标?			
12	是否填写测试总结报告?			

第 六章 软件易用性

6.1 易用性定义

易用性是用户体验的一个重要方面，网站建设者一般会沉溺于自己的思维习惯，而造成用户使用的不畅。

易用性不仅是专业UI/UE人员需要研究，对于网站建设其他岗位的人也应该了解一定的方法去检验和提升网站的易用性。

通常对易用性有如下定义：

- 易理解（Easy to discover）：用户必须在几秒钟知道网站主营内容
- 易操作（Easy to use）：初次接触网站功能或界面时，是否可不用帮助文档就能使用网站。
- 易学习（Easy to learn）：通过简单的帮助文件，用户就能解决遇到的问题
- 效率性（efficiency）：用户熟悉网站功能或界面后，能够快速达到使用目的
- 出错率（mistake）：在使用网站过程中，用户出现了多少错误，错误严重级别
- 满意度（satisfaction）：用对对网站的使用感受，用户满意度表单填写等易用性 Usability（又被译为 可用性），指的是用户实用软件产品时是否感觉方便，但是易用性 和 可用性 还有一定区别，可用性 指的是软件是否可使用，易用性 指的是是否方便使用。

比如淘宝、京东我们日常的都会去使用，其实他们不断的在更新网站内容，但是我们一般难以察觉，只会感觉 越来越好用。

比如用户想要在某网站购买python视频学习，只需要三次点击即可购买

- 点击学位课菜单
- 点击python中级/高级课程
- 点击购买

6.2 导航性测试

考虑如下问题，决定一个软件应用是否易于导航

- 导航是否直观
- 系统的主要部分是否可以通过主页获取
- 系统是否需要站点地图、搜索引擎或是其他导航的帮助例如淘宝网的导航栏，能够快速定位用户需要的资源信息

6.3 UI测试

人体工程学的主要目标是达到易用性。

用于与软件交互的方式称为用户界面或者UI。

检查用户界面的风格是否满足用户需求，是否通用合理、文字是否正常显示，页面是否美观、图片尺寸是否合理等。

在软件应用中，适当的图片设计和动画能够起到广告宣传的作用，又能够美化页面，一个软件应用的图形包括图片、动画、边框、颜色、字体、背景、按钮等等。

- 确保图片有明确含义，网站构图设计
- 图片尺寸尽量较小，并且能够精准表达某件事情，一般超链接到具体网站页面
- 网站字体风格是否一致
- 背景颜色与字体颜色是否和谐
- 图片质量与大小，传输速率的控制，对用户停留率有重要意义以上更多是在用户产品体验中考虑，不仅仅是测试

6.4 内容测试

测试人员检验软件应用是否提供正确的信息、准确的信息。

6.5 整体界面测试

指的是对整个软件应用的页面结构设计，给与用户的整体感受

- 用户能否迅速定位想要的信息
- 整体应用设计风格是否一致，和谐

第七章 软件兼容性

7.1 软件兼容性定义

兼容测试（Compatibility Test Suite）简称CTS，指对所设计程序与硬件、软件之间的兼容性的测试。
通俗的理解就是被测软件在不同的硬件平台（PC|Mobile），不同软件（浏览器），不同的操作系统（windows、linux、macos），不同的网络环境（2g、3g、4g、wifi）是否能友好的运行。

7.2 软件兼容性作用

兼容性测试是软件测试过程不可少的一个过程，没有兼容性测试是不完整的测试。

- 进一步提升产品质量，提升用户体验
- 兼容性测试可以使软件可以和其他软件尽可能多的友好相处
- 兼容性测试能够保证软件存在价值，提升软件市场竞争力

7.3 测试方法

- 购买真机，指定系统版本测试
- 利用云服务测试，在特殊情况下使用模拟器
- 实在难以购买的设备，选取同类机型作为替代