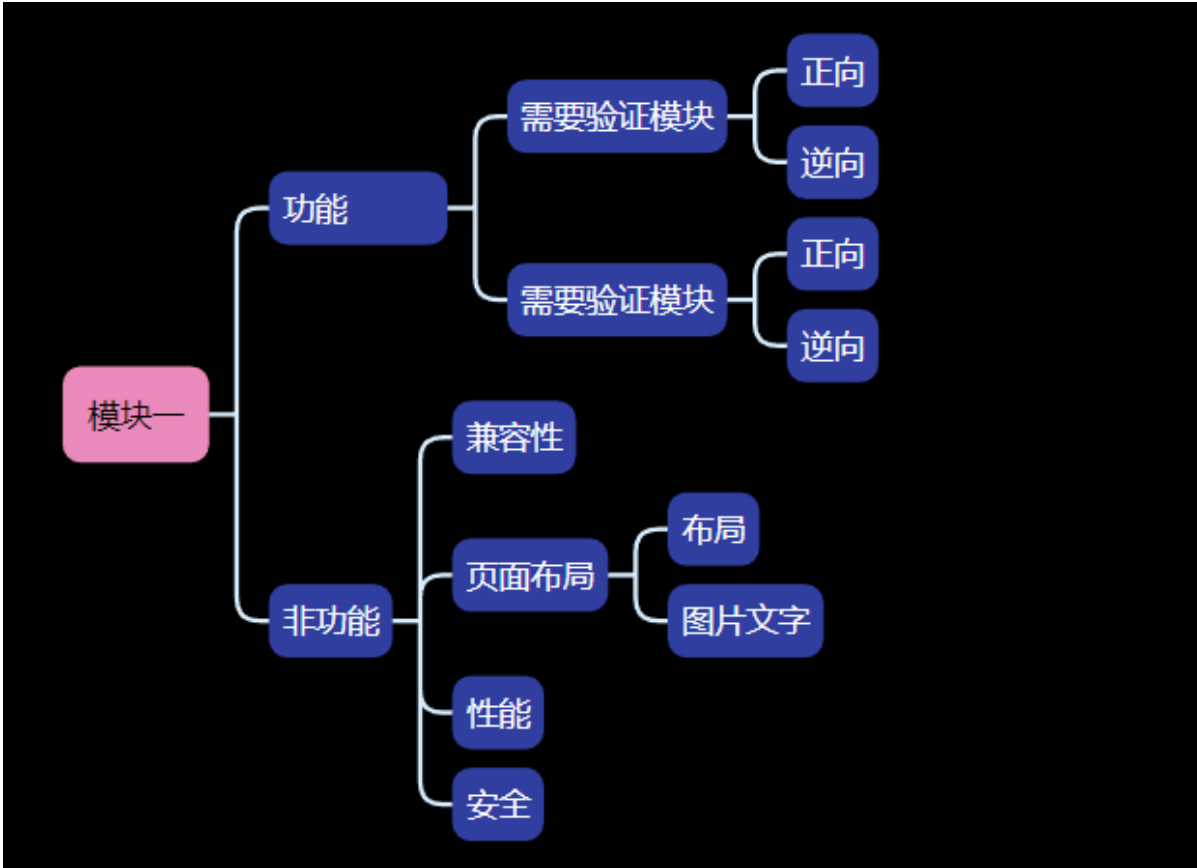


功能测试

软件测试：使用技术手段验证软件是否满足需求



阶段划分

- 单元测试：针对程序源代码进行测试
- 集成（接口）测试：单元与单元之间的接口进行测试
- 系统测试：针对系统整体功能+兼容+文档（说明、安装文档）
- 验收测试：内（公）测

代码可见度划分

- 黑盒测试：不见源代码，针对功能（系统测试）
- 灰盒测试：看见部分源代码，针对接口测试（集成测试）
- 白盒测试：针对程序源代码进行测试（单元测试）

质量模型

测试应该考虑的方向（软件、硬件）

五大重点：功能、性能、兼容、易用性、安全性

其他：可靠性、可维护性、可移植性（数据的转移）

测试流程

1. 需求分析（评审）：各部门协调分析需求
2. 计划编写：测什么、谁来测、怎么测
3. 用例设计：编写测试文档
4. 用例执行：开始测试
5. 缺陷管理：提交管理缺陷
6. 测试报告：

测试用例

用户使用的案例

作用：①防止漏测 ②软件测试的标准

用例编写规范

1. 用例编号：项目_模块_编号
2. 用例标题：预期结果（测试点）
3. 模块/项目：所属项目/模块
4. 前置条件：执行用例的前提条件
5. 优先级：用例的重要程度，p0—p4（p0最高）
6. 测试步骤：操作步骤
7. 测试数据：所需数据
8. 预期结果：期望结果

用例设计方法

数据无穷无尽

常见单个输入框：边界+等价类

正向逆向分析

等价类划分法

解决类型划分问题

1. 明确需求
2. 分类：有效 / 无效数据集合
3. 提取数据，编写用例

边界值法

解决边界限制问题

优化：开内闭外（开区间选包含的点，闭区选不包含的点）

上点：边界上的点（不考虑）

离点：距离边界最近的点（开内闭外）

内点：范围内的点（选取中间范围）

判定表

以表格形式解决多条件有依赖关系测试

适合4个以内条件依赖关系

超过4个则用正交法

- 条件桩：问题条件（条件项）
- 动作桩：可能的操作（动作项）

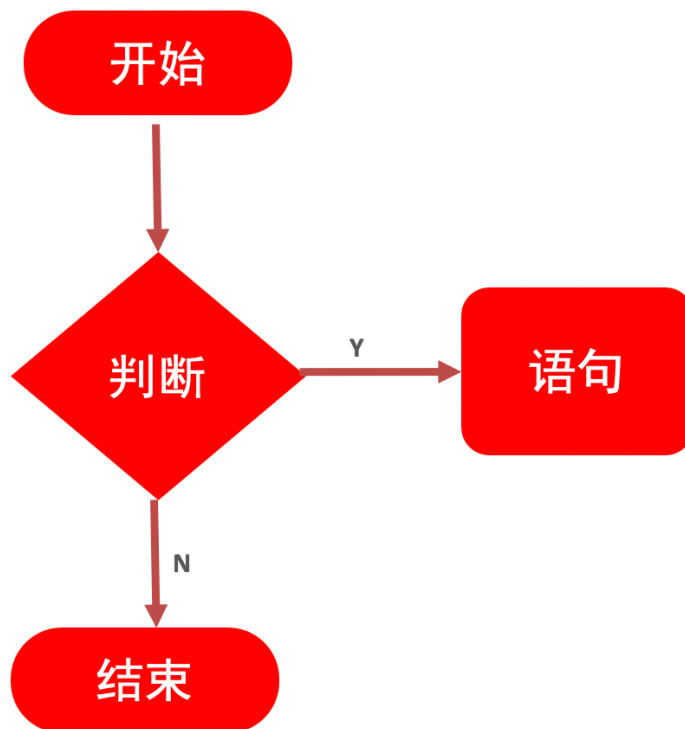
业务测试覆盖

先测试业务，再测试单功能、单模块、单页面

使用流程图梳理业务流程

标准流程图

- 使用标准图形和箭头来表达程序或业务的走向



错误推荐法

根据经验来测试

测试已完成，时间充裕，对出错多的模块再测试

时间紧，对重要易出错的模块测试

用例执行

说明：执行结果与用例的期望结果不一致（含义），为缺陷。

- 执行失败的用例

提示：用例执行不通过为缺陷，需要进行缺陷管理

缺陷

软件中存在的各种问题，即bug

缺陷标准

1. 少功能
2. 功能错误
3. 多功能
4. 缺少隐性功能
5. 易用性（软件测试人员专业角度）

缺陷产生原因

- 1、需求文档
- 2、架构设计
- 3、编码实现
- 4、环境（硬件、软件）
- 5、解决A，新缺陷B

回归测试

常规项目回归：项目新增模块，测试新增模块及关联的旧模块。

非常规项目（银行、部队、航天）：新增功能，必须全部复测。

缺陷核心要素

1. 缺陷标题：描述缺陷的核心问题
2. 预置条件：产生的前提
3. 复现步骤：
4. 预期结果：
5. 实际结果：
6. 必要附件：

缺陷提交要素

1. 缺陷编号
 2. 缺陷标题
 3. 缺陷状态
 4. 严重程度
 5. 优先级
 6. 所属模块
 7. 缺陷描述
 8. 附件

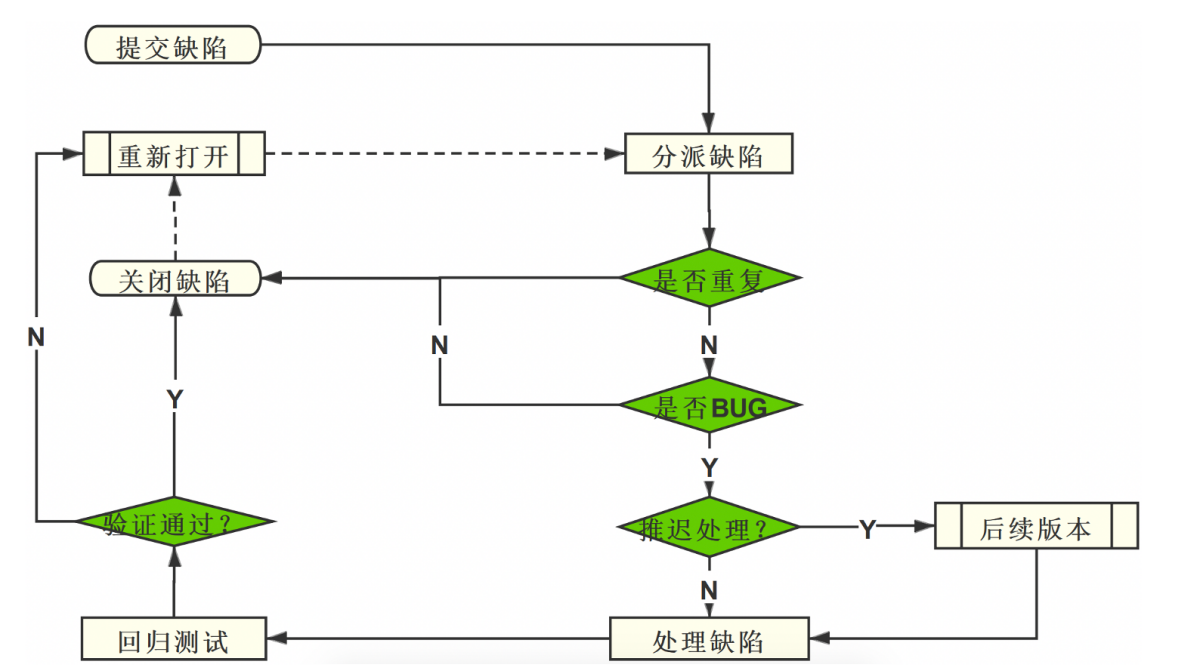
缺陷类型

- 1、功能错误
 - 2、UI页面错误
 - 3、兼容性
 - 4、数据（数据库）
 - 5、易用性
 - 6、建议
 - 7、架构缺陷

缺陷管理

excel或者管理工具

缺陷跟踪流程



提交注意事项

1. 可复现:
2. 唯一性: 一次提交一个
3. 规范性: 符合公司项目要求

自动化测试

程序或工具代替人工

回归测试、压力测试、兼容性、提高测试效率

优点：

1. 短时间运行最多的Bug
2. 减少人为错误
3. 脚本重复运行
4. 解决手工测试的局限性（图片）

web自动化测试

对web项目的自动化测试

手工测试之后

web自动化要求

1. 需求变动不频繁
2. 需要回归测试项目
3. 项目周期长

自动化测试工具

性能测试
