

第三部分 软件测试应用

3.5 测试文档书写方法





- 理解测试计划和测试报告书写的意义
- 掌握测试计划和测试报告的书写方式



1 测试计划



2 测试报告

测试计划



- 什么是测试计划
 - 测试计划Testing plan, 描述了要进行的测试活动的范围、方法、资源和进度的文档。它确定测试项、被测特性、测试任务、谁执行任务、各种可能的风险。
- 为什么制定测试计划
 - 方便团队成员开展工作
 - 作为管理的依据



- 怎样写测试计划

点子大家谈

测试计划



Step1: 了解基础知识

1. 了解测试覆盖率
2. 了解测试方法
3. 了解任务大致分配





• Step2:查阅并熟悉标准

文档(IEEE 829 Format)

IEEE Test Plan Template

1. [Test Plan Identifier](#)
2. [References](#)
3. [Introduction](#)
4. [Test Items](#)
5. [Software Risk Issues](#)
6. [Features to be Tested](#)
7. [Features not to be Tested](#)
8. [Approach](#)
9. [Item Pass/Fail Criteria](#)
10. [Suspension Criteria and Resumption Requirements](#)
11. [Test Deliverables](#)
12. [Remaining Test Tasks](#)
13. [Environmental Needs](#)
14. [Staffing and Training Needs](#)
15. [Responsibilities](#)
16. [Schedule](#)
17. [Planning Risks and Contingencies](#)
18. [Approvals](#)
19. [Glossary](#)





- **Step3: 书写说明**（包含简单的摘要、目标、范围、时间表等）



测试计划



- **Step4: 定义目标**

- 定义哪些测试哪些不测试
- 常见的包括：模块测试、集成测试、系统测试等等



测试计划



- **Step5: 写出需要的资源**
 - 包含人力资源和软硬件资源





- **Step6:** 写出测试过程中可能的风险和依赖



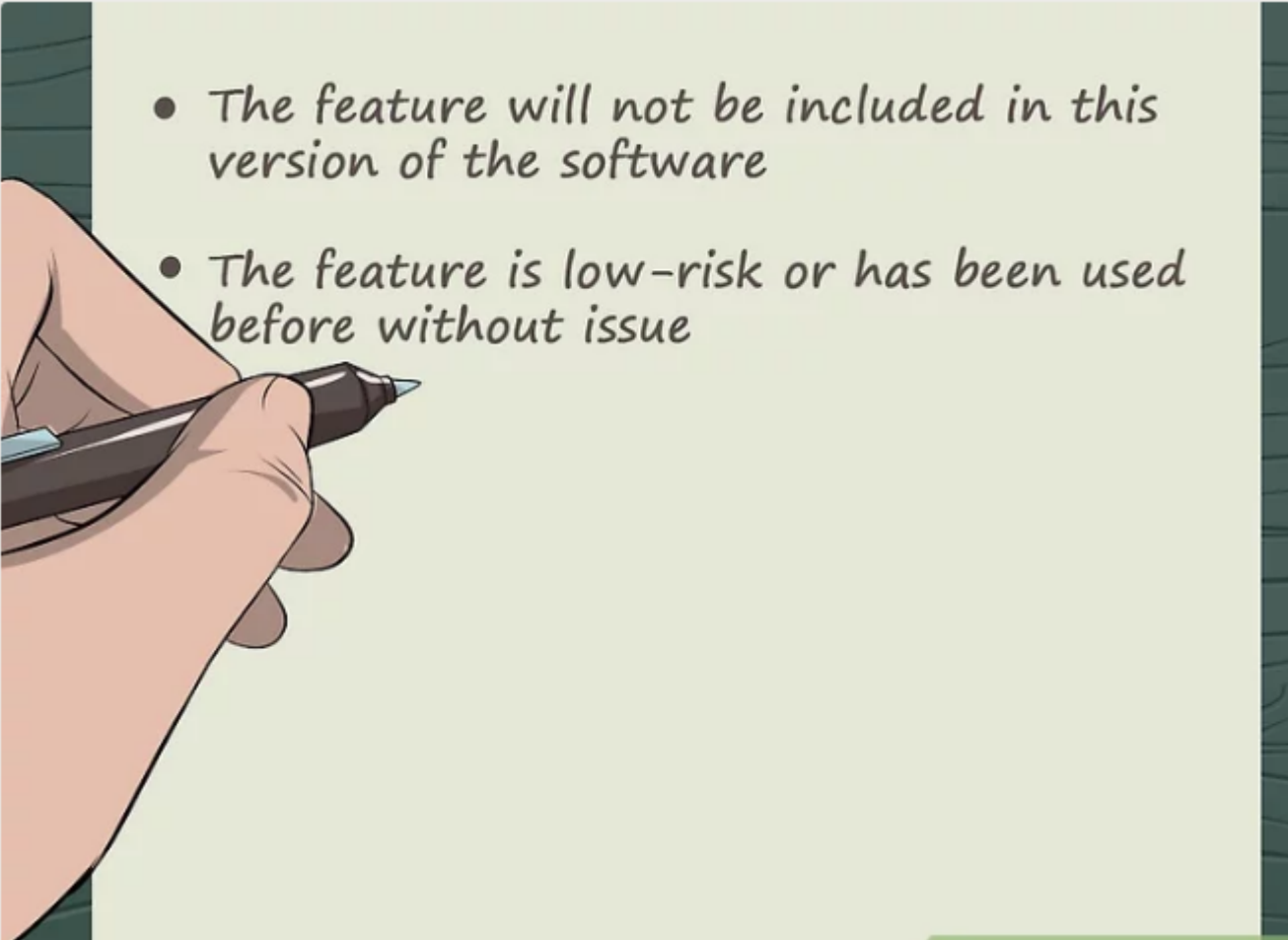


- **Step7:** 写出你将如何测试以及测试完成后会有哪些可交付成果





- **Step8:** 列出哪些功能不测试，以及不测试的原因

- 
- The feature will not be included in this version of the software
 - The feature is low-risk or has been used before without issue

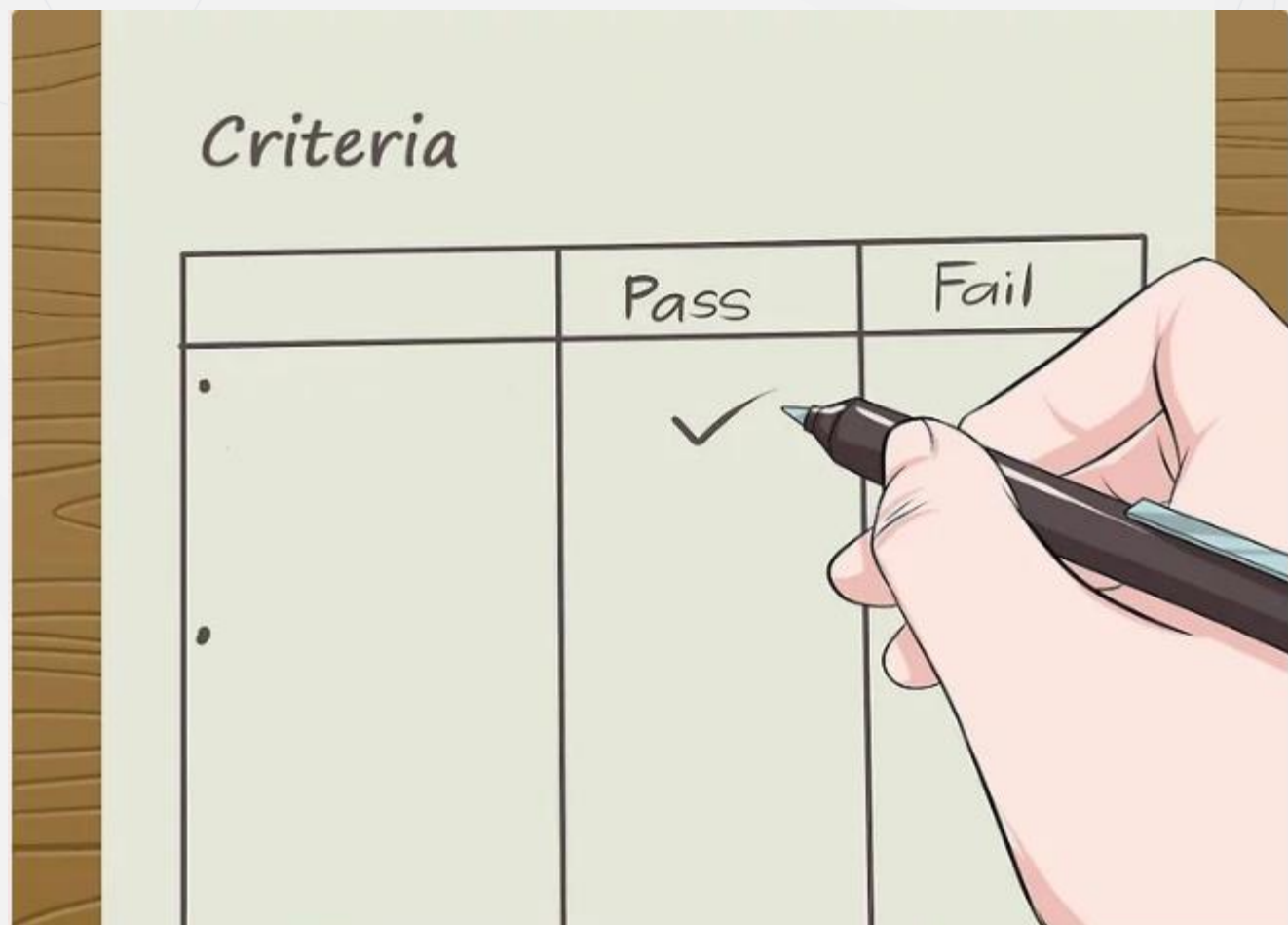


- **Step9:** 写出测试策略和将要使用的工具以及收集的信息



测试计划

- **Step10: 制定通过或失败的标准**



测试计划



- **Step11:** 列出在测试期间将产生的文件清单





1 测试计划



2 测试报告



- 什么是测试报告
 - 测试报告 (test report) 就是把测试的**过程**和**结果**写成文档，对**发现**的问题和**缺陷**进行分析，为纠正软件的存在的质量问题提供**依据**，同时为软件**验收**和**交付**打下基础
- 什么情况写测试报告
 - 测试完毕或一个阶段完毕，需要写出测试报告

测试报告

• 怎样写测试报告（实例）

- ▣ 1. 引言
 - 1.1 编写目的
 - 1.2 项目背景
 - 1.3 相关定义
 - 1.4 参考资料
- ▣ 2. 测试计划的执行
 - 2.1 执行进度
 - 2.2 人资耗费
 - 2.3 成果统计
- ▣ 3. 测试效果的评估
 - 3.1 需求覆盖
 - 3.2 测试结果
 - 3.3 用例的执行
- ▣ 4. 系统 Bug 分析
 - 4.1 Bug 统计信息
 - 4.2 Bug 状态分布
 - 4.3 Bug 级别分布
- ▣ 5. 软件质量的评价
 - 5.1 目前能力
 - 5.2 项目风险
 - 5.3 实施建议
 - 5.4 遗留问题

3、结论

- 1 测试结论
- 2 详细测试结果
- 2.1 各测试项结果
- 2.2 每个测试项使用数据
- 2.3 遗留问题清单
- 3 测试环境与测试数据
- 3.1 测试环境
- 4 BUG 统计及解决情况
- 5 测试局限
- 6 项目其他信息
- 6.1 项目概述
- 6.2 参考与引用文档
- 7 确认意见

测试报告

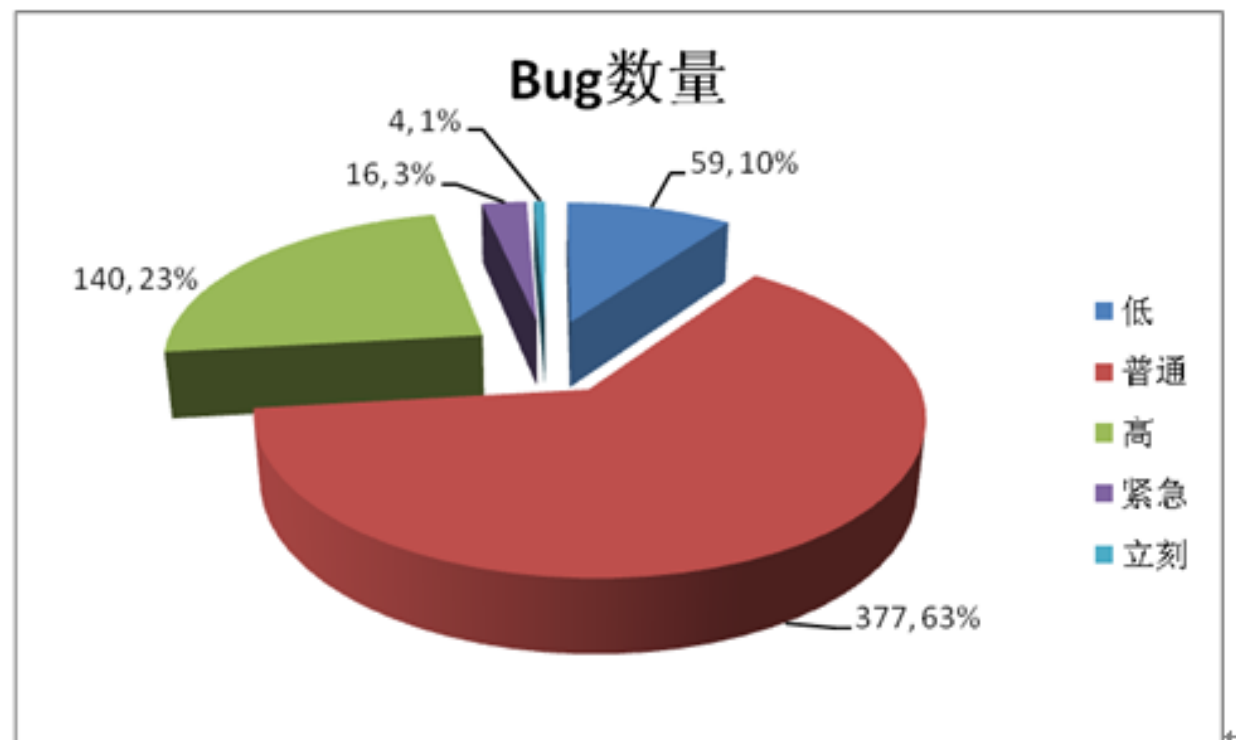


• Bug 分析

测试周期:		航司名称: 官网校验及价格更新到报价库					开发人员:		所属小组: 无				
		新功能bug					原有功能bug	发现问题用例未覆盖到的个数	已解决	未解决	Reopen>=2次的bug数统计		测试评语 (模块A)
	测试时间	总数	紧急	严重	一般	优化							
第一轮测试	2016-01-05	2		2									1、官网价格校验时 A当做了处理前的和 2、若qunar搜索出 确的报价, 但过段
第二轮测试	2016-01-05	0							1	1			
第二轮测试	2016-01-06	0											
第四轮测试	2016-01-08	0											
bug总数		2							1	1			2、若qunar搜索 后会更改成正确 误的报价 — —



根据 4.1 中的 Bug 统计信息表我们还可以得出各级别 Bug 的出现情况：



根据该图可以了解到，系统中出现 Bug 最多的是【普通】和【高】级别的 Bug。

【立刻】和【紧急】级 Bug 级别高，该类 Bug 易造成系统崩溃，此类 Bug 发现的越多，就越能够保证系统的稳定性。

【高】级 Bug 级别较高，该类 Bug 常因影响其他操作，因而修改的优先级较高。

【普通】级 Bug 为常规性 Bug，此类 Bug 发现的越多，就越能够保证系统的完善性。

【低】级 Bug 级别较低，我们在这部分发现的 Bug 较少，说明对于系统还有更多的细节部分的 Bug 还没有发现。

• Bug 分析

内容总结



- 测试计划：定义、书写方法
- 测试报告：定义、书写方法



Question