

测试执行和报告

上午10时30分

提纲

1. 测试执行流程
2. 测试日志
3. 软件缺陷描述
4. 报告软件缺陷的重要性
5. 报告软件缺陷的基本原则
6. 软件缺陷管理

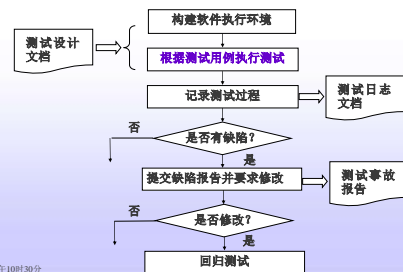
上午10时30分

提纲

1. 测试执行流程
2. 测试日志
3. 软件缺陷描述
4. 报告软件缺陷的重要性
5. 报告软件缺陷的基本原则
6. 软件缺陷管理

上午10时30分

1. 测试执行流程



上午10时30分

提纲

1. 测试执行流程
2. 测试日志
3. 软件缺陷描述
4. 报告软件缺陷的重要性
5. 报告软件缺陷的基本原则
6. 软件缺陷管理

上午10时30分

2. 测试日志

- ♦ **测试日志 (Test log)**：软件测试执行过程中的常规记录，构建软件测试日志的目的是为了对软件**测试过程的证实**；另外，当出现软件缺陷时，便于**追溯**软件缺陷的根源
- ♦ **测试日志**提供按**时间顺序**记录的测试执行过程中相关事件的详细记录

上午10时30分

2.1 测试日志的内容

1. 测试日志标识符
2. 描述
3. 活动和事件登记(Activity and event entries)
4. 事故报告标识符 (Incident report identifiers)

上午10时30分

2.1.1 测试日志的描述

- ◆ 描述日志中所有记录的信息，包括：
- 1. 识别包含版本/修订版本号的**测试项**；
- 2. 识别**测试管理环境属性**，包括设备，使用的硬件（内存、CPU型号、硬盘），使用的系统软件以及可用的资源等

上午10时30分

2.1.2 活动和事件登记

1. **执行描述**：被测试过程的标识符，在测试期间的所有现场人员及作用
2. **程序结果**：对于每一个执行过程，其可见的输出以及测试执行的成功或失败
3. **环境信息**：这个测试日志的环境信息
4. **异常事件**：一个异常事件发生的情况

上午10时30分

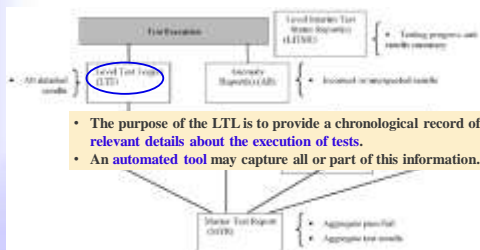
2.2 测试日志举例

BVW-200无线遥测软件测试日志：

- ◆ 2011年9月12日
- ◆ 8:30 李金利 启动软件开始测试；
- ◆ 8:40 李金利 设置系统工作条件：采样率：500Hz，采集信号类型：脑电，滤波：30Hz；系统启动的过程参见测试程序说明TP11-16
- ◆ 8:40 张祥同 将测试动物放到无线接收板上；
- ◆ 10:40 李金利 观察系统的运行情况，系统运行良好；
- ◆ 10:40 李金利 结束本次测试

上午10时30分

2.3 IEEE829-2008推荐的测试报告类型



上午10时30分

IEEE829-2008 Level Test Log Outline

1. Introduction
 - 1.1. Document identifier
 - 1.2. Scope
 - 1.3. References
2. Details
 - 2.1. Description
 - 2.2. Activity and event entries
 - Execution description: 过程id，所有参与的人员及其角色
 - Procedure results: 各用例的执行结果
 - Environmental information
 - Anomalous events
 - Anomaly Report identifiers
3. General
 - 3.1. Glossary

上午10时30分

提纲

1. 测试执行流程
2. 测试日志
3. 软件缺陷描述
4. 报告软件缺陷的重要性
5. 报告软件缺陷的基本原则
6. 软件缺陷管理

上午10时30分

3. 软件缺陷描述

- 在测试中通常会
产生与预期结果
不一致的测试结
果，这就产生了
软件缺陷
- 软件测试员要清
晰描述所发现软
件缺陷的各个方
面

序号	缺陷属性	描述
1	标识符	标识某个软件缺陷的唯一编号
2	描述	发生缺陷软件的版本、模块、环境以及触发的过程，产生的现象
3	缺陷类型	软件缺陷的分类，比如功能性缺陷，性能缺陷等
4	严重性	软件缺陷对于软件质量的破坏程度
5	优先级	缺陷被处理的紧急程度
6	状态	缺陷所处的生命周期
7	起源	缺陷引起的故障或事件第一次被检测到的阶段，包括：需求，构架，设计，编码，测试，发布等
8	再现性	缺陷是否可以再现

上午10时30分

提纲

1. 测试执行流程
2. 测试日志
3. 软件缺陷描述
4. 报告软件缺陷的重要性
5. 报告软件缺陷的基本原则
6. 软件缺陷管理

上午10时30分

4. 报告软件缺陷的重要性

- 软件测试的目的是尽早找到缺陷，并保证其得以修复
- 并非所有的缺陷都得到修复，有些可能完全忽略，有些可能推迟到下一个版本中
- 为什么找到的缺陷得不到有效修复呢？



解决问题？

上午10时30分

4.1 不修复软件缺陷的原因

- 没有足够的时间——由于资源的缺乏和项目进度要求，造成没有时间修复所有软件缺陷。
- 不算真正的软件缺陷——在某些特殊场合，错误理解、测试错误或者说明书变更会把软件缺陷当作附加功能来对待。
- 不值得修复：不常出现的软件缺陷和在不常用功能中出现的软件缺陷可以放过，这些归结为商业风险决策
- 软件缺陷报告不够有效：测试员没有建立足够强大的用例来说明特定缺陷是否应该修复

上午10时30分

4.1 不修复软件缺陷的原因

- 修复风险太大——修复一个软件缺陷可能导致其他软件缺陷出现，修改软件将冒很大的风险。通常而言，涉及到底层数据结构的修改都将存在较大的风险，不要盲目修改



上午10时30分

提纲

1. 测试执行流程
2. 测试日志
3. 软件缺陷描述
4. 报告软件缺陷的重要性
5. 报告软件缺陷的基本原则
6. 软件缺陷管理

上午10时30分

5 报告软件缺陷的基本原则

1. 尽快报告软件缺陷
2. 有效描述软件缺陷
3. 在报告软件缺陷时不作评价
4. 使用合适的工具，跟踪软件缺陷

上午10时30分

提纲

1. 测试执行流程
2. 测试日志
3. 软件缺陷描述
4. 报告软件缺陷的重要性
5. 报告软件缺陷的基本原则
6. 软件缺陷管理

上午10时30分

6 软件缺陷管理

- ♦ 为了保证能够修复所有记录的缺陷，测试组织内需要建立一套完整的缺陷管理过程和规则，对处于生命周期不同阶段的缺陷分别管理
- ♦ 通常使用专门的缺陷管理系统

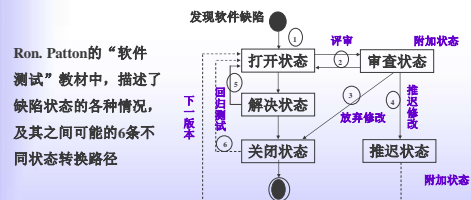
上午10时30分

6.1 软件缺陷的各种状态

1. 打开状态：缺陷首先被测试员发现，登记下来后转交给程序员修复的状态
2. 解决状态：程序员修复了代码，软件回到测试员手中的状态
3. 关闭状态：测试员重新对软件进行测试，确认缺陷已经修复的状态
4. 审查状态：项目管理员确定缺陷是否应该修复的状态，这是一个附加状态
5. 推迟状态：审查可能认定缺陷应该在将来进行修复的状态

上午10时30分

6.2 通用软件缺陷生命周期状态图



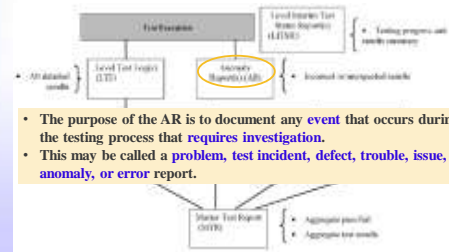
上午10时30分

6.3 软件缺陷管理系统

- ◆ BugZilla
 - ◆ BugFree
 - ◆ JIRA
 - Apache使用JIRA进行缺陷管理的例子
- <https://issues.apache.org/jira/issues/?filter=-4>

上午10时30分

IEEE829-2008推荐的测试报告类型



- The purpose of the AR is to document any **event** that occurs during the testing process that **requires investigation**.
- This may be called a **problem, test incident, defect, trouble, issue, anomaly, or error report**.

上午10时30分

IEEE829-2008 Anomaly Report Outline

1. Introduction
 - 1.1. Document identifier
 - 1.2. Scope
 - 1.3. References
2. Details
 - 2.1. Summary
 - 2.2. Date anomaly discovered
 - 2.3. Context
 - 2.4. Description of anomaly
 - 2.5. Impact
 - 2.6. Originator's assessment of urgency
 - 2.7. Description of the corrective action
 - 2.8. Status of the anomaly
 - 2.9. Conclusions and recommendations
3. General
 - 3.1 Document change procedures and history

- Inputs
- Expected results
- Actual results
- Unexpected outcomes
- Procedure step
- Environment
- Attempts to repeat
- Testers
- Observers
- open
- approved for resolution
- assigned for resolution
- fixed
- retested with the fix confirmed

上午10时30分

