

软件测试培训

——缺陷管理

Terence Zhou

缺陷管理

- 软件测试的根本目的是什么？

缺陷管理

- 软件测试中经常使用各种术语来描述软件出现的问题, 如下一些通用的术语:
 - ◆ 软件错误 (Software Error)
 - ◆ 软件缺陷 (Software Defect)
 - ◆ 软件故障 (Software fault)
 - ◆ 软件失效 (Software failure)
- 区分这些术语很重要, 它关系到测试工程师对软件失效现象与机理的深刻理解. 由于软件内部逻辑复杂, 运行环境动态变化, 且不同的软件差异可能很大, 因而软件失效的机理可能也有不同的表现形式, 但总的来说, 软件失效的机理可描述为:
- 软件错误→软件缺陷→软件故障→软件失效

缺陷管理

软件错误:在可以遇见的时期内,软件将有人来开发.在整个生存期的各个阶段,都贯穿着人的直接或间接的干预.然而人难免犯错误,这必然给软件留下不良的痕迹.软件错误是指在软件生存期内的不希望或不可接受的人为错误,其结果是导致软件缺陷的产生.可见,软件错误是一种人为过程,相对于软件本身,是一种外部行为.

软件缺陷:软件缺陷是存在于软件(文档,数据,程序)之中的那些不希望或不可接受的偏差.其结果是软件运行于某一特定条件时出现软件故障,这时称软件被激活.

软件故障:软件故障是指软件运行过程中出现的一种不希望或不可接受的内部状态.比如:软件处于执行一个多余循环过程时,我们可以软件出现故障.若此时没有适当的措施(容错)加以处理,便产生软件失效.软件故障是一种动态行为.

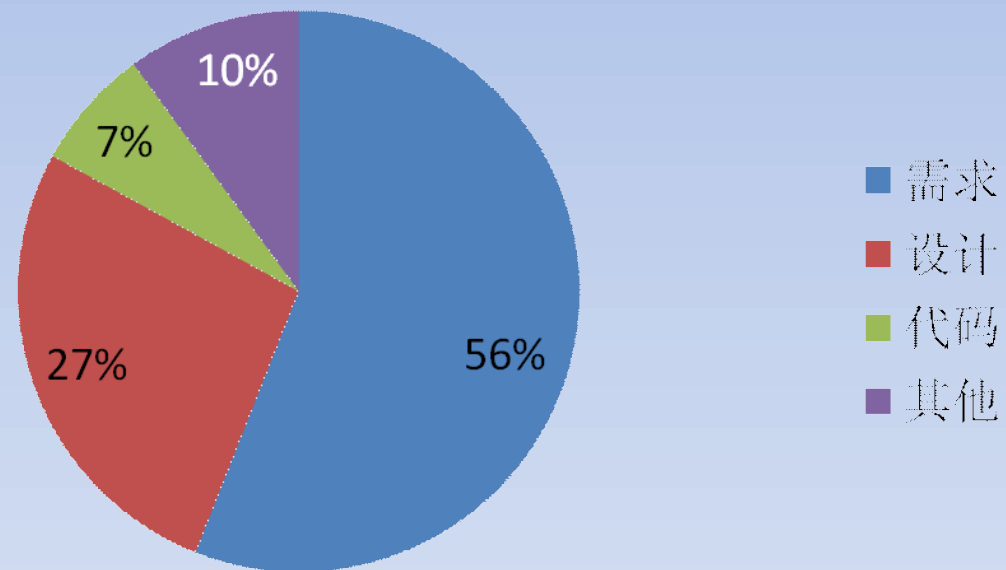
软件失效:软件失效是指软件运行时产生的一种不希望或不可接受的外部行为结果.

缺陷管理

综上所述, 软件错误是一种人为错误. 一个软件错误必定产生一个或多个软件缺陷. 当一个软件缺陷被激活时, 便产生一个软件故障; 同一个软件缺陷在不同条件下被激活, 可能产生不同的软件故障. 软件故障如果没有及时容错措施加以处理, 便不可避免地导致软件失效.

缺陷管理

缺陷原因分布图



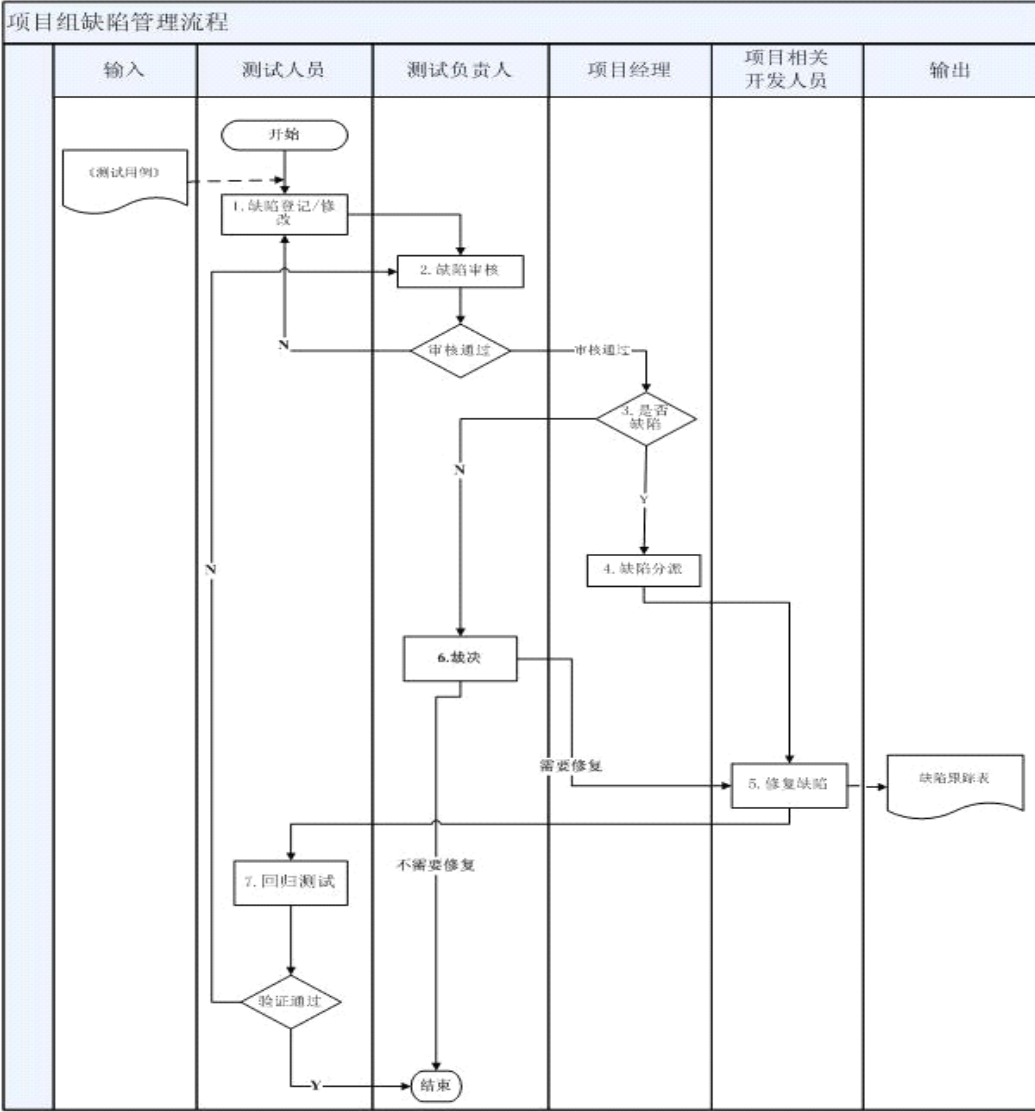
缺陷管理-目的

- 缺陷管理目的：
- 缺陷管理目的是对各阶段测试发现的缺陷进行跟踪管理，以保证各级缺陷的修复率达到标准。主要实现以下目标：
 - 及时了解并跟踪每个被发现的缺陷；
 - 确保每个被发现的缺陷都能被处理；
 - 收集缺陷数据并根据缺陷趋势曲线识别测试过程阶段；
 - 收集缺陷数据并在其上进行分析，作为组织过程的财富。

缺陷管理-人员职责

- 参与缺陷管理过程人员角色职责：
 - 项目经理（PM）
 - 负责指派缺陷给相关责任人。
 - 项目测试负责人（TM）：
 - 决定缺陷管理方式和工具，拟定决策评审计划；
 - 管理所有缺陷关闭情况；
 - 审核测试人员提交的缺陷；
 - 对测试人员的工作质量进行跟踪与评价。
 - 测试人员（TE）
 - 负责报告系统缺陷记录，且协助项目人员进行缺陷定位；
 - 负责验证缺陷修复情况，且填写缺陷记录中相应信息；
 - 负责执行系统回归测试；
 - 提交缺陷报告；
 - 负责被测软件进行质量数据和分析。
 - 项目相关开发人员（DE）
 - 修改测试发现的缺陷，并提交成果物做再测试；
 - 负责接收各自的缺陷记录，并且修改；
 - 负责提供缺陷记录跟踪中其它相应信息。
 - 质量保证人员（SQA）
 - 监控项目组缺陷管理规程执行情况。

缺陷管理-流程图



缺陷管理-过程介绍

- 缺陷登记:
- 缺陷审批:
- 是否缺陷:
- 缺陷分派:
- 修复缺陷:
- 缺陷回归测试:

缺陷管理-缺陷来源介绍

- | 缺陷来源 | 缩写 | 描述 | |
|-------------------------------|----|-------------------------|-------|
| • Cause-Requirement | | 由于需求的问题引起的缺陷 | C-R |
| • Cause - Design | | 由于设计的问题引起的缺陷 | C-D |
| • Cause - Code | | 由于编码的问题引起的缺陷 | C-C |
| • Cause - Test | | 由于测试的问题引起的缺陷（测试用例设计问题等） | C-T |
| • Cause - Integration & Other | | 由于集成或其它问题引起的缺陷 | C-I&O |

缺陷管理-缺陷相关属性

缺陷属性	描述
缺陷描叙 (Summary)	简单描述缺陷，主要是什么缺陷
缺陷发现提交者 (Detected By)	描叙缺陷是由谁发现提出的。
缺陷发现时间 (Detected on Date)	描叙缺陷发现提出时间。
缺陷严重性 (Severity)	描述缺陷的严重性。
缺陷分给谁 (Assigned to)	指缺陷分派给谁。
缺陷在哪个版本发现 (Detected in Version)	描叙缺陷发现的版本
缺陷被修改的时间 (Modified)	描叙缺陷被修改的时间。
计划修复时间 (Plan fixed Data)	描叙缺陷计划完成修复的时间。
缺陷优先级 (priority)	描述缺陷的优先级。
缺陷所属项目 (Project)	描述缺陷所属的工程。
是否是重现缺陷 (Reproducible)	描述缺陷是否是重现缺陷。
缺陷的状态 (Status)	描述缺陷的状态
缺陷所属于的模块 (subject)	描述缺陷所属的模块。
缺陷详细描述 (Description)	缺陷详细描述，包括缺陷产生的步骤，缺陷的实际结果，缺陷的理想结果，建议等。
缺陷实际关闭的版本 (Closed in Version)	描述缺陷实际关闭的版本。
缺陷实际修复所花的时间 (Actual Fixed Time)	描述缺陷实际修复所花的时间
缺陷修复完成时间 (Closing Date)	描述缺陷实际关闭的时间。
注释 (Comments)	描叙对缺陷的注释。
附件 (Attachments)	添加缺陷附件。

缺陷管理-缺陷等级定义

等级	说明	现象描述（部分例子）	优先级
A类	致命错误	由于程序所引起的死机,非法退出; 死循环; 数据库发生死锁; 因错误操作导致的程序中断; 与数据库连接错误; 数据通讯错误; 导致测试无法继续执行。 可能影响其他模块功能。	立即处理或解决
B类	很严重的错误	程序错误; 程序接口错误; 数据库的表、业务规则、缺省值未加完整性等约束条件; 关键功能完全不能实现; 程序运行不稳定,如出现不可继续进行操作的错误; 程序运行出现难以捕捉和不可再现的错误; 响应其他业务流程的错误。	在发现的两天内完成。
C类	一般严重错误	操作界面错误（包括数据窗口内列名定义、含义是否一致） 打印内容、格式错误 简单的输入限制未放在前台进行控制 删除/退出操作未给出提示 数据库表中有过多的空字段 功能不完整,如菜单、按钮不响应 对错误没有处理信息	系统上线前必须修复完成
D类	一般性错误	界面不规范; 辅助说明描述不清楚; 输入输出不规范; 提示窗口文字未采用行业术语; 可输入区域和只读区域没有明显的区分标志。	正常排队等待修复或方便时修复
E类	较小错误	Tab键跳转不正常; 窗口控件的Z-Order不正确; ; 窗口中的按钮或者控件缺少快捷字母,或快捷字母冲突; 文字表述中有错别字或歧义; 测试人员所提出的建设性意见。	方便时再修复

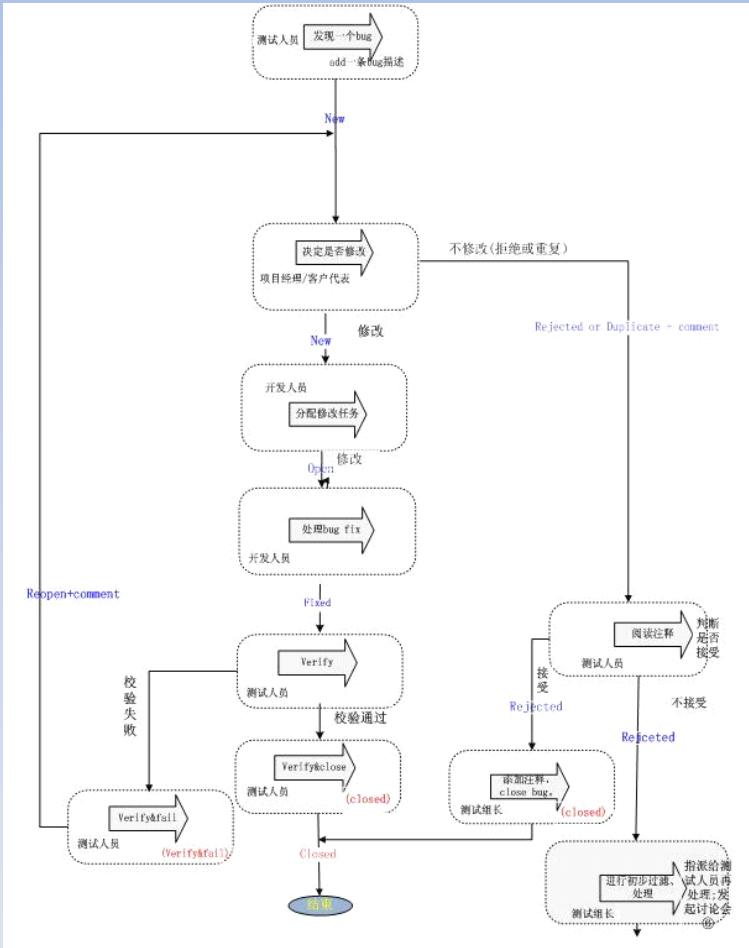
缺陷管理-缺陷修复优先级

优先级	描述
紧急 (5-Urgent)	缺陷很紧急且很严重，得立即修复。
很高优先级 (4-very High)	例如，软件的主要功能错误或者造成软件崩溃，数据丢失的缺陷。
较高优先级 (3-High)	例如，影响软件功能和性能的一般缺陷。
一般优先级 (2-Medium)	例如，本地化软件的某些字符没有翻译或者翻译不准确的缺陷。
低优先级 (1-Low)	例如，对软件的质量影响非常轻微或出现几率很低的缺陷。

缺陷管理-缺陷状态

缺陷状态	描述
新提交 (New)	新提交的缺陷状态
激活 (Open)	缺陷已提交, 正在处理
已拒绝 (Rejected)	拒绝“已提交的缺陷”, 不需要修改或不是缺陷
已解决 (Fixed)	缺陷已修改
重激活 (Reopen)	缺陷修改未通过再测试, 或因其他原因造成缺陷再次打开
重复缺陷 (Duplicate)	缺陷重复出现, 已经被提交过。
已关闭 (Closed)	确认缺陷已被修复, 将其关闭

缺陷管理-缺陷状态转换图



缺陷管理-怎样专业的描述缺陷

- 软件缺陷的有效描述规则，主要是：

1. 单一准确

每个报告只针对一个软件缺陷。在一个报告中报告多个软件缺陷的弊端是常常会导致缺陷部分被注意和修复，不能得到彻底的修正。

2. 可以再现

提供缺陷的精确操作步骤，使开发人员容易看懂，可以自己再现这个缺陷，通常情况下，开发人员只有再现了缺陷，才能正确地修复缺陷。

3. 完整统一

提供完整、前后统一的软件缺陷的步骤和信息，例如：图片信息，Log文件等。

4. 短小简练

通过使用关键词，可以使软件缺陷的标题的描述短小简练，又能准确解释产生缺陷的现象。如“主页的导航栏在低分辨率下显示不整齐”中“主页”、“导航栏”、“分辨率”等是关键词。

5. 特定条件

许多软件功能在通常情况下没有问题，而是在某种特定条件下会存在缺陷，所以软件缺陷描述不要忽视这些看似细节的但又必要的特定条件（如特定的操作系统、浏览器或某种设置等），能够提供帮助开发人员找到原因的线索。如“搜索功能在没有找到结果返回时跳转页面不对”。

6. 补充完善

从发现bug那一刻起，测试人员的责任就是保证它被正确的报告，并且得到应有的重视，继续监视其修复的全过程。

7. 不做评价

在软件缺陷描述不要带有个人观点，对开发人员进行评价。软件缺陷报告是针对产品、针对问题本身，将事实或现象客观地描述出来就可以，不需要任何评价或议论。

缺陷管理-工具介绍

- Test Director Hp公司

- Clear Quest IBM Rational公司

THANKS!