

软件测试 Software Testing

24. 报告所发现的缺陷

程适

cheng@snnu.edu.cn

计算机科学学院

2016 年 12 月 23 日



陕西师范大学
SHAANXI NORMAL UNIVERSITY

Outline

- 软件缺陷的描述
- 软件缺陷相关的信息
- 软件缺陷跟踪和分析
- 软件缺陷跟踪系统

报告所发现的缺陷

- 发现了缺陷，还需要描述缺陷产生的过程或现象，报告给开发人员，这就是软件缺陷的报告。
- 如何报告所发现的软件缺陷？就是要准确、清楚地描述内容，这其中要借助一些工具来创建记录软件缺陷的日志文件。

软件缺陷的描述

- 1 清晰准确的软件缺陷描述可以减少软件缺陷从开发人员返回的数量；
- 2 提高软件缺陷修复的速度，使每一个小组能够有效地工作；
- 3 提高测试人员的信任度，可以得到开发人员对清晰的软件缺陷描述有效的响应；
- 4 加强开发人员、测试人员和管理人员的协同工作，让他们可以更好地工作。

软件缺陷的描述

- 软件缺陷的生命周期
- 严重性和优先级
- 缺陷的其他属性
- 完整的缺陷信息
- 缺陷描述的基本要求

软件缺陷的生命周期

- 生命周期的概念是一个物种从诞生到消亡经历了不同的生命阶段，那么软件缺陷生命周期指的是一个软件缺陷被发现、报告到这个缺陷被修复、验证直至最后关闭的完整过程。
- 在整个软件缺陷生命周期中，通常是以改变软件缺陷的状态来体现不同的生命阶段。

软件缺陷的生命周期

缺陷状态	描述
激活或打开 (Active or Open)	问题还没有解决，存在源代码中，确认“提交的缺陷”，等待处理，如新报的缺陷
已修正或修复 (Fixed or Resolved)	已被开发人员检查、修复过的缺陷，通过单元测试，认为已解决但还没有被测试人员验证
关闭或非激活 (Close or Inactive)	测试人员验证后，确认缺陷不存在之后的状态
重新打开	测试人员验证后，还依然存在的缺陷，等待开发人员进一步修复
推迟	这个软件缺陷可以在下一个版本中解决
保留	由于技术原因或第三者软件的缺陷，开发人员不能修复的缺陷
不能重现	开发人员不能复现这个软件缺陷，需要测试人员检查缺陷复现的步骤
需要更多信息	开发能复现这个软件缺陷，但开发人员需要一些信息，例如缺陷的日志文件、图片等

严重性和优先级

- 致命的 (fatal): 致命的错误, 造成系统或应用程序崩溃 (crash)、死机、或造成数据丢失、主要功能完全丧失等。
- 严重的 (critical): 严重错误, 指功能或特性 (feature) 没有实现, 主要功能部分丧失, 次要功能完全丧失, 或致命的错误声明
- 一般的 (major): 不太严重的错误, 这样的软件缺陷虽然不影响系统的基本使用, 但没有很好地实现功能, 没有达到预期效果。
- 微小的 (minor), 一些小问题, 对功能几乎没有影响, 产品及属性仍可使用, 如有个别错别字, 文字排列不整齐等。

缺陷的其他属性

- 标识 (ID): 是标记某个缺陷的唯一的表示, 可以使用数字符号表示
- 类型 (type): 是根据缺陷的自然属性划分的缺陷种类
- 可能性 (frequency): 指缺陷在产品中发生的可能性, 通常可以用频率来表示
- 来源 (source): 指缺陷所在的地方, 如文档、代码等
- 根源 (root cause): 指造成上述错误的根本因素, 以寻求软件开发流程的改进、管理水平的提高。

完整的缺陷信息

- 步骤，提供了如何重复当前缺陷的准确描述。应简明而完备、清楚而准确
- “期望结果”，与测试用例标准、或设计规格说明书、或用户需求等一致，达到软件预期的功能。
- “实际结果”，测试人员收集的结果和信息，以确认缺陷确实是一个问题，并标识那些影响到缺陷表现的要素。

缺陷描述的基本要求

- 单一准确：每个报告只针对一个软件缺陷
- 可以再现：提供这个缺陷的精确通用步骤
- 完整统一：提供完整、前后统一的再现软件缺陷的步骤和信息
- 短小简练：通过使用关键词，使缺陷标题的描述短小简练

缺陷描述的基本要求

- 特定条件：不要忽视一些看似细节但又必要的特定条件
- 补充完善：保证 Bug 被正确地报告
- 不做评价：不要带有个人观点，不要对开发人员进行评价

软件缺陷相关的信息

- 软件缺陷的图片信息
- 使用 WinDbg 记录软件缺陷信息
- 使用 Soft-ICE 记录软件缺陷信息
- 分离和再现软件缺陷

软件缺陷的图片信息

- 采用图片的格式
- 什么情况下需要附上图片

使用 WinDbg 记录软件缺陷信息

- 配置
- 使用
- 调试方式
- 常用命令

使用 Soft-ICE 记录软件缺陷信息

- 跟踪软件运行时的变量、内存等状态；
- 捕捉系统崩溃时所需的信息；
- 记录产品发生缺陷的地方同时生成日志文件。

分离和再现软件缺陷

- 分离和再现软件缺陷的步骤
- 分离和调试软件缺陷之间的区别

软件缺陷跟踪和分析

- 软件缺陷跟踪管理时测试工作的一个重要部分，测试的目的时尽早发现软件系统中的缺陷，而对软件缺陷进行跟踪管理的目的是确保每个被发现的缺陷能够及时得到处理。

软件缺陷跟踪和分析

- 软件测试过程简单说就是围绕缺陷进行的，缺陷跟踪管理的目标如下：
 - 确保每个被发现的缺陷都能够被解决，“解决”的意思不一定是被修正，也可能是其他处理方式，总之，对每个被发现的缺陷的处理方式必须能够在开发组织中达到一致。
 - 收集缺陷数据并根据缺陷趋势曲线识别测试过程的阶段；决定测试过程是否结束有很多种方式，通过缺陷趋势曲线来确定测试过程是否结束是常用并且较为有效的一种方式；
 - 收集缺陷数据并在其上进行分析，作为组织的过程财富。

软件缺陷跟踪和分析

- 软件缺陷处理技巧
- 缺陷趋势分析
- 缺陷分布分析
- 缺陷跟踪方法

软件缺陷处理技巧

- 审阅：当一个新测试人员发现一个缺陷，通过页面提交到缺陷跟踪数据库中。在缺陷被分配给开发人员之前，为了保证缺陷描述的质量和减少开发人员的抱怨，最好由其主管或其他资深测试工程师进行审阅。
- 拒绝：如果审阅者决定对一份缺陷报告进行较大的修改，应该和报告人一起讨论，由报告人修改缺陷报告，然后再次提交。
- 完善：如果测试员已经完整地描述了问题的特征并将其分离，那么审查者就会肯定这个报告。

软件缺陷处理技巧

- 分配：当开发组接受完整描述特征并被分离的问题时，测试员将它分配给适当的开发人员。
- 测试：一旦开发人员修复一个缺陷，还需要得到测试人员的验证，没经过测试人员的验证，缺陷是不能被关闭的。
- 重新打开：如果缺陷没有通过验证，那么测试人员将重新打开这个缺陷。重新打开一个缺陷，需要加注释说明。

软件缺陷处理技巧

- 关闭：如果通过验证测试，那么测试人员将关闭这个缺陷。只有测试人员才能关闭缺陷，开发人员没有这个权限。
- 暂缓：如果每个人都同意将确实存在的缺陷移到以后处理，应该指定下一个版本号或修改的日期。一旦新的版本开始时，这些暂缓的缺陷应该重新被打开。

缺陷趋势分析

- 软件质量标准一般要求：在测试结束前高优先级的缺陷必须被全部处理完，所以有必要监控这类缺陷随时间的变化，例如生成相应的趋势图，以判定整个产品开发是否按预期进度进行，测试是否可以按时结束。

缺陷分布分析

- 缺陷分布报告
- 缺陷趋势报告
- 缺陷年龄报告
- 测试结果进度报告

缺陷跟踪方法

- 当前缺陷状态
- 项目发展趋势

软件缺陷跟踪系统

- 软件缺陷跟踪系统拥有软件缺陷跟踪数据库，它不仅有利于软件缺陷的清楚描述，还提供统一的、标准化的报告，使所有人的理解一致；
- 软件缺陷跟踪数据库允许自动连续地给软件缺陷编号，还提供了大量供分析和统计的选项，这是手工方法无法实现的；

软件缺陷跟踪系统

- 基于缺陷跟踪数据库，可快速生成满足各种查询条件的、必要的缺陷报表、曲线图等，开发小组乃至公司的每一个人都可以随时掌握软件产品质量的整体状况、测试/开发的进度；
- 基于跟踪数据库提供了软件缺陷属性并允许开发小组根据对项目的相对和绝对重要性来修复缺陷；

软件缺陷跟踪系统

- 可以在软件缺陷的生命周期中管理缺陷，从最初的报告到最后的解决。确保了每一个缺陷不会被忽略，同时，它还可以使注意力保持在那些必须尽快修复的重要缺陷上；
- 当缺陷在它的生命周期中变化时，开发人员、测试人员以及管理人员将熟悉新的软件缺陷信息。一个设计良好的软件缺陷跟踪系统可以获取历史记录，并在检查缺陷的状态时参考历史记录；

软件缺陷跟踪系统

- 在软件缺陷跟踪数据库中关闭每一份缺陷报告，它都可以被记录下来。当产品送出去时，每一份未关闭的缺陷报告都提供了预先警告的有效技术支持，并且证明测试人员找到特殊领域突然出现的事件中的软件缺陷。

小结

- 软件缺陷的描述
- 软件缺陷相关的信息
- 软件缺陷跟踪和分析
- 软件缺陷跟踪系统

小结

- 缺陷报告和处理的规范过程，包括缺陷描述的基本信息和缺陷生命周期。
- 如何正确、有效地描述软件缺陷，并借助工具提供各种必要的信息，以及开发人员和测试人员如何协同工作，分离、跟踪和处理软件缺陷，以保证优先级高的缺陷及时能得到解决。

小结

- 测试人员需要建立软件缺陷跟踪数据库或系统，收集各种软件缺陷的数据，进行趋势分析和分布性分析
- 了解测试的进度和产品质量状况，并找到软件开发过程中薄弱的环节，改进软件开发过程。

致谢

谢谢，欢迎提问！