**测试缺陷管理规范**

**版本记录**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **版本号** | **修改日期** | **修改内容** | **修改者** | **审核者** | **状态** | **发布范围** |
| V0.0.0 | 2017-XX-XX | 新建文档 | XXX | XXX | 初稿 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目录

[1 引言 4](#_Toc29352)

[1.1 目的 4](#_Toc1068)

[1.2 适用范围 4](#_Toc30949)

[1.3 术语和定义 4](#_Toc24031)

[2 缺陷的属性 4](#_Toc26796)

[2.1 缺陷的严重程度 4](#_Toc20870)

[2.2 缺陷的优先级 5](#_Toc6122)

[2.3 缺陷的状态 6](#_Toc11223)

[3 缺陷的处理流程 7](#_Toc15600)

[3.1 Bug处理流程图 7](#_Toc23672)

[3.2 Bug处理流程描述 7](#_Toc6427)

[4 Bug处理的基本规范 8](#_Toc5739)

[4.1 准确记录 8](#_Toc13820)

[4.2 快速重现 8](#_Toc15672)

[4.3 特别关注需求性bug 9](#_Toc12391)

[5 Bug处理注意事项 9](#_Toc19488)

[5.1 bug处理注意事项 9](#_Toc10579)

[6 Bug统计与分析规则 9](#_Toc4116)

[6.1 禅道统计类型 9](#_Toc29303)

[6.2 测试日/周报缺陷统计汇总 10](#_Toc4685)

[6.3 Bug缺陷分析总结 10](#_Toc32000)

# 1 引言

## 1.1 目的

缺陷是产品与规定要求不相符的部分，会存在于软件产品的整个生命周期中，本文规范软件测试过程中的出现的缺陷，通过测试活动及早发现软件系统中的缺陷，并确保缺陷被有效标识、跟踪、和修改，保证软件系统能够达到要求的质量。

## 1.2 适用范围

适用于软件的整个生命周期。

不限于测试过程发现的缺陷。评审，用户使用等过程中发现的缺陷都是应当按照本流程进行登记跟踪管理。

## 1.3 术语和定义

软件缺陷：软件或程序中存在的某种破坏正常运行能力的问题、错误，或者隐藏的功能缺陷。

严重程度：缺陷严重程度是指因缺陷引起的故障对[软件](http://baike.baidu.com/view/37.htm" \t "_blank)产品的影响程度。

优先级：缺陷必须被修复的紧急程度。

# 2 缺陷的属性

## 2.1 缺陷的严重程度

|  |  |
| --- | --- |
| **严重程度** | **含义** |
| 1-致命 | ①　需求书中的重要功能未实现  ②　造成系统崩溃，死机  ③　常规操作造成程序非法退出，死循环，通讯中断或异常，数据破坏丢失或数据库异常，且不能通过其他方法实现功能的；  ④　关系到金钱的；  ⑤　关系到用户的私密信息的。 |
| 2-严重 | ①　重要功能基本能实现，但系统不稳定（偶尔闪退），一些边界条件下操作会导致run-time error，文件操作异常，通讯异常，数据丢失或破坏等错误；  ②　重要功能不能按正常操作实现，但可通过其他方法可实现；  ③　错误的波及面广，影响到其他重要功能正常实现； |
| 3-一般 | ①　次要功能不能正常实现（筛选、过滤有点错误）；  ②　操作界面错误（包括数据窗口内列名定义，含义不一致）；  ③　打印内容，格式错误；  ④　查询错误，数据错误显示；  ⑤　简单的输入限制未放在前台进行控制；  ⑥　删除操作未给出提示；  ⑦　数据库表中有过多的空字段（接口测试用到）；  ⑧　因错误操作迫使程序中断； |
| 4-细微 | ①　程序在一些显示上不美观，不符合用户习惯，或者是一些文字的错误  ②　界面不规范；  ③　辅助说明描述不清楚；  ④　输入输出不规范（错位）；  ⑤　提示窗口文字未采用行业术语；  ⑥　可输入区域和只读区域没有明显的区分标志（可输入是白色输入框；不可写是灰色的输入框）；  ⑦　界面存在文字错误；  ⑧　在功能实现方式上如果需求中没有明确定义，而没有按常规实现，并且不比常规方式实现优越的。（如用户名第一位用数字或特殊字符） |
| 5-建议(非缺陷) | ①　可提高产品质量的建议，包括新需求和对需求的改进；  ②　需求文档没有规定的特性，如果实现会对系统功能或易用性有所提高。 |

## 2.2 缺陷的优先级

|  |  |
| --- | --- |
| **优先级** | **含义** |
| 高 | 导致测试暂停，无法进行；上午发现的，下午18：00前解决；下午发现的晚上21：00前解决；晚上发现的次日12：00前解决。严重影响到上线的、难以定性的可适当延期1-2天解决） |
| 中 | 导致部分功能无法测试；需要优先解决，解决周期1天。 |
| 低 | 不影响测试的进行；可在方便时解决，解决周期2天。 |

## 2.3 缺陷的状态

|  |  |
| --- | --- |
| 缺陷状态 | 描述 |
| 初始状态 | 测试或客服人员提交一个新的缺陷，等待开发人员分配修改负责人 |
| 驳回 | 要求缺陷的提交者再次对缺陷进行说明 |
| 已分配 | 已分配给开发人员，待修改状态 |
| 已解决 | 缺陷已被开发人员修复，等待测试人员验证 |
| 关闭 | 测试人员验证已修复的缺陷 |
| 激活 | 测试人员验证，缺陷没有修改正确 |
| 遗留 | 经项目负责人/产品经理验证此缺陷在本版本中不用修改 |

# 3 缺陷的处理流程

## 3.1 Bug处理流程图



## Bug处理流程描述

1.测试人员发现bug提交给开发。

2.开发人员判断是否是bug。

3.如果是bug，进行修改，修改完成后更改bug状态为已解决。

4.如果不是bug，退回给测试人员并描述退回原因，或为设计如此，或为外部原因，或者不能重现。

5.开发人员修改完成的bug，由测试人员进行验证，确认修改正确，关闭bug。

6.验证未通过的bug重新激活，开发人员继续修改，直至验证通过，关闭bug。

7.测试人员需要对开发人员退回的bug进行确认。

8.确认不是bug关闭。

9.如与开发人员意见不一致，认为是bug，需提交项目负责人/产品经理仲裁。

10.项目负责人/产品经理确认是bug由开发人员修改，不是bug由测试人员关闭。

# 4 Bug处理的基本规范

## 4.1 准确记录

1．描述一定是问题的描述，要求简练明了（一看描述就知道是什么问题）。

2．步骤不要写成一堆（步骤是重现的步骤，要1. 2. 3. 步骤的写清楚）。

3．一定要抓图说明问题，并上传附件到缺陷管理系统。

## 4.2 快速重现

很多bug有时无法重现，具体原因分析：

1. Bug的描述不准确。
2. Bug重现的环境不一致。
3. 开发与测试的版本不一致。
4. Bug本身存在偶发性。
5. 开发没有在集成环境下验证。
6. 在修复另外一个bug把其中一个bug修复了。

**注意原则：**

1. Bug描述步骤必须准确。
2. 在测试偶发的bug必须定位（可以找开发人员一起定位）。
3. 开发提交一个测试版本，在开发那边会同时建立一个集成环境，与测试环境一致，如果出现在本地无法重新的bug，请到开发集成环境下验证。如果仍无重现，请找测试员确认

## 4.3 特别关注需求性bug

需求性bug必须与需求人员讨论，并确认其对客户的影响后，才可以报告到bug库中。

# Bug处理注意事项

## 5.1 bug处理注意事项

1. 开发人员原则上没有关闭Bug的权限，所有的Bug必须要由测试人员验证之后才能关闭，

2. 开发人员把Bug设置为“不改”和“挂起”和“延期处理”之前，必须经过项目负责人/产品经理确认，必要时需要和客户进行沟通。

3. 原则上允许Invalid（无效的）的Bug存在，但是为了提高Bug的质量，在新建一个Bug之前，要对Bug进行充分的验证，确保其有效性。

4. 原则上允许Duplicate（重复的）的Bug存在，但是为了提高Bug的质量，在新建一个Bug之前，要对已有Bug库进行检索，确保该Bug不会被重复提交。

5. 所提交的Bug必须包含必要且尽可能详细的信息，以便于开发人员更快的定位Bug及修复Bug。包括Bug复现需要的步骤，具体的错误信息及截图，Bug所产生的环境，期望结果及预期结果。

# 6 Bug统计与分析规则

## 6.1 禅道统计类型

1．登录禅道可以对bug进行统计和查询，以便获得需要的bug信息。

选择具体查询类型包括：ID,BUG标题，所属模块，状态，指派给，抄送给，严重程度，优先级，Bug类型，修改者，修改日期，创建者，创建日期，解决者，解决日期，重复bug，关闭者，关闭日期，截止时间，浏览器，关键词，影响版本，激活次数，解决方案，所属系统，功能点，所属轮次，bug状态，修改内容，解决版本，引入原因。

2．选择查询的比较条件：等于，不等于，大于等于，小于等于，包含，不包含等。

3．选择对于条件值并提交查询。

## 6.2 测试日/周报缺陷统计汇总

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **第X轮测试缺陷情况：** | | | | | | | | | | | |
| **系统名称** | **已关闭** | | | | | **未关闭** | | | | | **总计** |
| **A级** | **B级** | **C级** | **D级** | **合计** | **A级** | **B级** | **C级** | **D级** | **合计** |
| XXXX |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **总计** |  |  |  |  |  |  |  |  |  |  |  |

## 6.3 Bug缺陷分析总结

本部分对上述缺陷和其他收集数据进行综合分析

**统计汇总数据：此次需求测试共发现缺陷 个，已修复验证通过 个，遗留 个。**

缺陷综合分析：

**缺陷发现效率 ＝ 缺陷总数/执行测试用时**

可到具体人员得出平均指标

**用例质量 ＝ 缺陷总数/测试用例总数 ×100％**

**缺陷密度 ＝ 缺陷总数/功能点总数**

缺陷密度可以得出系统各功能或各需求的缺陷分布情况，开发人员可以在此分析基础上得出那部分功能/需求缺陷最多，从而在今后开发注意避免并注意在实施时予与关注，测试经验表明，测试缺陷越多的部分，其隐藏的缺陷也越多。