质量控制部Bug管理手册

|  |  |
| --- | --- |
| Version | V1.8 |
| Date of final modification | 2017/1/5 |

Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Revised version | Description | Author | Reviewer |
| 2015-09-22 | V1.0 | Create file | 邓志坚 | 陈诚/赖庆珊/廖常鉴 |
| 2015-09-30 | V1.1 | Update file | 邓志坚 | 陈诚/赖庆珊/廖常鉴 |
| 2015-10-13 | V1.2 | Update file | 邓志坚 | 陈诚/赖庆珊/廖常鉴 |
| 2016-03-25 | V1.3 | Update file | Martin |  |
| 2016-07-08 | V1.4 | Update file | 张亮 | QA组 |
| 2016-07-26 | V1.5 | Update file | 邓志坚 | QA组 |
| 2016-09-21 | V1.6 | 1.更新Bug分类  2.新增Bug状态转换规则 | 陈诚 | VR项目组 |
| 2016-10-12 | V1.7 | 1.更新状态切换规则  2.更新指派规则 | 陈诚 | VR项目组 |
| 2017-1-5 | V1.8 | 1.添加Bug提交注意事项  2.添加需求类和争议类问题处理规范 | 邓志坚 | VR项目组 |

目录

[一. Bug分类原则 4](#_Toc471380742)

[1.1 功能问题(Title标签为[FN]), 跟踪标签为”缺陷” 4](#_Toc471380743)

[1.2 界面问题(Title标签为[UI]), 跟踪标签为”缺陷” 4](#_Toc471380744)

[1.3 体验类问题(Title标签为[UX], 跟踪标签为”缺陷”) 4](#_Toc471380745)

[1.4新功能类问题(跟踪为”需求”) 4](#_Toc471380746)

[1.5需求类问题(跟踪为”需求”) 4](#_Toc471380747)

[二. Bug严重等级判断原则 5](#_Toc471380748)

[2.1 Bug严重等级计算公式 5](#_Toc471380749)

[2.2Bug严重等级对应风险系数表 5](#_Toc471380750)

[2.3问题后果严重程度对应分数表（Severity） 5](#_Toc471380751)

[2.4问题发现可能性对应分数表（Detection） 7](#_Toc471380752)

[2.5问题出现频率对应分数表（Frequency） 8](#_Toc471380753)

[三.Bug提交规范 8](#_Toc471380754)

[3.1 跟踪标签 8](#_Toc471380755)

[3.2主题 8](#_Toc471380756)

[3.3描述格式 8](#_Toc471380757)

[3.4状态 9](#_Toc471380758)

[3.5 优先级 10](#_Toc471380759)

[3.6 类别 10](#_Toc471380760)

[3.7开始日期 10](#_Toc471380761)

[3.8计划完成日期 10](#_Toc471380762)

[3.9 是否用例范围 10](#_Toc471380763)

[3.10 测试手段 10](#_Toc471380764)

[3.11 引入阶段 10](#_Toc471380765)

[3.12附件提交规范 10](#_Toc471380766)

[3.13注意事项： 11](#_Toc471380767)

[四.Bug处理规则 12](#_Toc471380768)

[4.1 测试目的 12](#_Toc471380769)

[4.2执行流程 13](#_Toc471380770)

[4.3QA验证Bug规则 14](#_Toc471380771)

[4.4需求类问题处理 15](#_Toc471380772)

[4.5争议类问题处理 15](#_Toc471380773)

[4.6 开发人员(RD)权责 15](#_Toc471380774)

[4.7 研发组长权责 16](#_Toc471380775)

[4.8 产品经理权责 16](#_Toc471380776)

[4.9 QA权责 16](#_Toc471380777)

[4.10各角色对Bug状态转换的权限 17](#_Toc471380778)

# 一. Bug分类原则

## 1.1 功能问题(Title标签为[FN]), 跟踪标签为”缺陷”

FN类Bug判定规则：

* 功能实现与产品需求不符
* 功能重复或者多余
* 在用户层面，功能的使用性、方便性、易用性不够

## 1.2 界面问题(Title标签为[UI]), 跟踪标签为”缺陷”

UI类Bug判定规则：

* 界面显示异常、不美观
* 模块或者文字排版、格式不统一
* 语言文字翻译不正确

## 1.3 体验类问题(Title标签为[UX], 跟踪标签为”缺陷”)

体验类Bug判定规则：

* 产品功能易用性, 友好性, 简洁性不佳.
* 语言文字翻译不符合用户习惯
* 产品功能或者界面不符合用户使用习惯或与产品的理念不符.

## 1.4新功能类问题(跟踪为”需求”)

新功能类别的问题主要有以下几种：

* 建议新增的功能需求
* 市场或客户要求新增的功能

## 1.5需求类问题(跟踪为”需求”)

需求类别的问题主要有以下几种：

* 需求定义不完整
* 需求细节未定义，如非功能性需求、提示信息缺失、异常情况处理缺失
* 需求定义存在前后冲突或者二义性

# 二. Bug严重等级判断原则

## 2.1 Bug严重等级计算公式

Bug严重等级判断原则是，基于Bug导致的后果严重程度、问题被用户发现的可能性、问题重现的概率综合计算得出风险系数，并对照风险系数表，得出Bug的严重等级。计算公式如下：

Risk= Severity \* Detection \* Frequency

## 2.2Bug严重等级对应风险系数表

各级风险系数对应Bug严重等级如下表所示：

|  |  |
| --- | --- |
| 严重等级 | 风险系数值 |
| Blocking | (800,1000] |
| Critical | (640,800] |
| Fatal | (400,640] |
| Medium | (120, 400] |
| Low | (0,120] |

**注意：一切有可能引起批量退货的问题，都为Blocking级别问题（用户体验、交互设计、内容质量）**

## 2.3问题后果严重程度对应分数表（Severity）

|  |  |  |
| --- | --- | --- |
| 分值 | 严重程度定义 | 示例 |
| 10 | 缺陷发生后，产品主要功能会失效、业务会陷入瘫痪状态、关键数据损坏或丢失，且无法自行恢复（如无法自动重启恢复） | 1. 产品主要功能失效/和用户期望不符,用户无法正常使用； 2. 由程序引起的死机、反复重启等，并且故障无法自动恢复 3. 死循环、死锁、内存泄露、内存重释放等 4. 系统存在严重的安全漏洞 5. 用户的关键数据损坏或丢失并不可恢复 |
| 8 | 缺陷发生后，主要功能无法使用、失效、存在可靠性、安全、性能方面的重要问题，但在出现问题后一般可以自行恢复（如自动重启恢复） | 1. 系统单一功能错误不影响业务流程或者有替代方法 2. 产品重要功能不稳定 3. 由程序引起的非法退出、重启等，但是可以自行恢复 4. 文档与产品严重不符、缺少，或存在关键性错误 5. 产品难于理解和操作 6. 产品无法进行正常的维护 7. 产品升级后功能出现丢失、性能下降 8. 性能不达到系统规格 9. 产品不符合标准规范，存在严重的兼容性问题 |
| 5 | 缺陷发生后系统在功能、性能、可靠性、可维护性、可安装性等方面的一般问题 | 1. 出现多余功能(软件做了其不该做的) 2. 产品一般性的功能失效 3. 特殊情况下产生错误，但不影响正常业务（如异常数据或者异常操作）   4．异常数据未做控制，操作成功 |
| 3 | 缺陷发生后，对用户只会造成轻微影响，这些影响一般在用户可以忍受范围内 | 1. 用户文档错误（包括错别字和描述错误以及截图与实际不一致） 2. 界面文字错误，翻译错误 3. 界面设计不规范,没有考虑易用性问题 4. 产品的提示信息不够清晰准确，难于理解 5. 长时间操作未给用户提供进度提示 |
| 1 | 建议性问题 | 1. 新功能建议 2. 产品的输出正确，但是不够规范 |

## 2.4问题发现可能性对应分数表（Detection）

|  |  |  |
| --- | --- | --- |
| 分值 | 可能性定义 | 示例 |
| 10 | 基本功能操作，用户必须使用，无备选路径 | 1. 开关机、用户登录、识别图像 2. 系统升级、软件安装卸载 3. 播放视频、玩游戏 4. WIFI/BT基本功能 5. 用户登录、注销 6. 幻视识别图片 |
| 8 | 重要功能操作，用户经常使用，使用频率高 | 1. 下载应用、录制、分享、我的订单、AR天气、点赞 2. 进入应用界面主要入口，功能切换 |
| 5 | 辅助功能操作，用户很少使用，使用频率中 | 1. 查看系统信息、查看版本信息、查看网站备案信息 2. WIFI/BT高级设置 3. 安装使用未使用我司SDK开发的第三方APP 4. 幻视发现页预览主题、进入品牌主页 |
| 2 | 冷僻操作、异常操作 | 1. 不间断、连续插拔SD卡1小时 2. 使用工具每秒点击页面数百次切换页面操作 3. 不间断、连续开启关闭摄像头1小时   4．使用播放器播放不支持的音频/视频格式 |

## 2.5问题出现频率对应分数表（Frequency）

|  |  |  |
| --- | --- | --- |
| 分值 | 可能性定义 | 示例 |
| 10 | 有条件必然重现 | 1. 在同样条件下，测试3次出现一次 |
| 8 | 有条件概率重现 | 1. 在同样条件下，测试5次出现一次 |
| 5 | 无规律重现 | 1. 测试20次出现1次，无特定规律 |
| 2 | 无法重现 | 1. 不间断、连续插拔SD卡1小时 |



# 三.Bug提交规范

## 3.1 跟踪标签

跟踪标签用于标识该问题归属**缺陷**、**需求**还是**任务**类型

## 3.2主题

**主题格式：[FN/UI][硬件版本][系统版本][应用版本]问题描述...(发生概率)**

例如：[FN] [P3][0.5.4\_0223][Cinema\_v20160323]选片大厅最后一部影片后面的刷新功能点击以后没有反应.(10/10)

**标签含义说明：**

* **[FN]**功能问题，**[UI]**界面问题
* **[硬件版本]**当前测试设备的硬件版本
* **[系统版本]**当前测试系统的系统版本
* **[应用版本]**当前发生Bug的应用名称和版本号，比如[Settings\_v20160323],如果测试的时候应用模块是内置到版本中的,则无需写此项
* **问题描述:** 应该言简意赅，让RD能第一时间看懂问题的重点.
* **发生概率**：x/10. x定义：Tester操作10次的情况下Bug重现的次数.

## 3.3描述格式

描述格式如下（粗体部分是固定格式）：

——————————————————————————————

**[前提条件]：**手机内存/SD卡有保存多部电影

**[复现步骤]**：

1.进入选片大厅

2.一直滑动影片直到最后一部

3.点击刷新图标

**[Bug现象]：**点击以后没有效果.

**[期望结果]**：点击以后可以刷新影片信息

**[发生时间]**：0：16

**[Log地址]**：\\10.0.1.55\vr\_glass\03.LOG\201509\1738

**[备注]：只在一台机器上出现**

——————————————————————————————

*【规则提示】：*

（1）发生时间是指当前设备的系统时间；

（2）有附Log的Bug必须填写发生时间，没有Log的Bug无须填写；

（3）Log尽量附到Bug上面，如果不能附上需填写Log路径；

（4）偶现Bug需尽可能多的备注测试环境和前提条件，供RD参考分析；

（5）复现步骤尽量控制在6步以内；

## 3.4状态

**New：**新提交Bug时的状态，表明RD还未进行处理。

**Accept：**RD分析并接受Bug，表明正在修复中。

**Fix：**RD已完成Bug修复，等待QA验证。

**Need Clarification：**RD需要QA提供更多关于Bug的资讯。

**Not Reproduce：**RD无法分析Bug且无法再复现该Bug，请求QA再次复现。

**Reject：**因某种原因，RD拒绝修复。

**Duplicate：**和其他Bug重复。

**Reopen：**QA验证Bug依然存在，请求RD继续修复。

**Pending：**Bug暂时无法验证或者暂时不修复。

**Monitor：**需要长期跟踪或者无法重现的重大Bug可将状态改为Monitor

*【规则提示】：*

（1）研发组长需提供给测试组RD Feature Owner文档，测试组依据此指派RD，如未提供，则默认指派给研发组长；

（2）RD需每日处理完New状态的Bug，如果对应RD请假，则由组长代理；

（3）测试组需每日追踪New状态的Bug并提醒对应RD处理；

（4）New，Fix，Need Clarification，Not Reproduce，Reject，Duplicate，Reopen状态的Bug超过3天未更新状态，当前Bug Owner要负拖延责任；

（5）RD在Accept后需填写计划完成日期，如有客观原因未能按时完成，请及时更新计划完成日期并备注说明原因；

（6）QA根据计划完成日期追踪Bug修复进度，如逾期未修复，将会被Highlight.

（7）Fix Bug需填写Bug发生原因分析，Bug版本修复方案说明及Bug验证的版本；

（8）Bug当前Owner在转给其他人时需要按照规则详细备注原因，如果未备注原因对应Owner有权直接打回，由此带来的时间成本由Bug Owner承担；

## 3.5 优先级

Bug优先级分为：立刻，优先，正常，稍缓

*【规则提示】：*

（1）研发应根据优先级来排序优先处理的Bug；

（2）优先级为立刻的Bug，QA在提交之后应立即告知RD去处理；

## 3.6 类别

Bug发生对应的根源模块,例如：App Store、App Manager、Cinema、Driver、Settings等

## 3.7开始日期

QA提交Bug时默认填写的开始日期是提交Bug的日期，RD接受Bug后可手动更改开始日期或者保持默认；

## 3.8计划完成日期

RD在接受bug之后需要根据实际的工作量合理填写计划完成的日期，

## 3.9 是否用例范围

如果Bug属于用例覆盖范围内,则选择"是",反之则填"否",如果是拓展测试/自由测试发现的Bug,则填写"AD-HOC".

## 3.10 测试手段

发现Bug所用的测试手段,主要有:功能测试,功能交互,界面测试,中断测试,异常测试等

## 3.11 引入阶段

Bug的引入阶段主要分为：需求引入、设计引入、编码引入、修改后引入，如果不确定引入阶段则可与产品、开发确认,确认之后再选择对应的阶段.

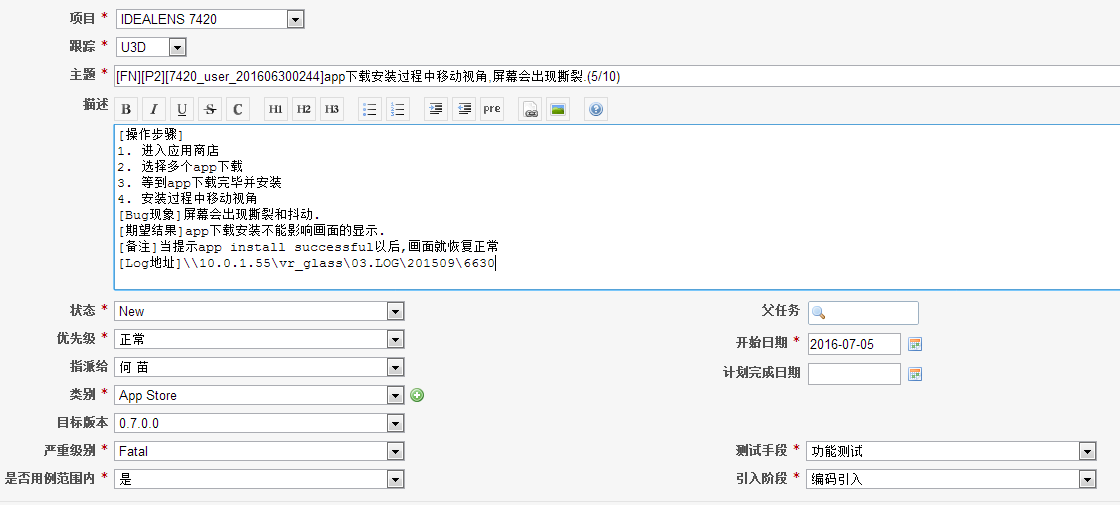
## 3.12附件提交规范

如果Bug附件size小于20M，则可点击“**选择文件**”可上传Bug附件.

注意事项：

* 如无特殊原因，Bug尽可能的附上图片，视频和日志；
* 如果附件Size大于20M，可上传至服务器：\\10.0.1.55\vr\_glass\03.LOG\ 并将Log地址写在问题描述里.
* 如果此Bug需要特殊的测试数据(电影、apk 等)，则测试数据也要作为附件上传

Bug整个描述规范图示如下：



## 3.13注意事项：

1.Bug提交颗粒度要求：一个Bug只描述一个问题；

2.涉及到“菜单"/"画面"/"按钮"尺寸大小、高低如果觉得不合适的，请先内部讨论以后再开Bug给开发.因为每个人的标准不一样，只有汇总了一定的采样数据以后尽量取到平均值或者一定的范围再反馈给开发

3.Bug现象及复现步骤描述要求：描述内容必须要与Bug Title一致，切勿用多余、宽泛、不具体、有歧义的语言对开发造成误导，如果现象描述中涉及到数据就需要给出具体数据，涉及到对比，则描述清楚对比参照物和产品本身的差异，重点是描述到具体的点上；

4.期望结果描述要求：如果是属于需求文档定义内容的Bug，复制需求文档的描述，或者在此基础上做不更改本意、客观的扩展描述即可，如果是需求文档定义范围外的Bug，期望结果一律描述为——“参照产品经理更新后的设计实现”；

5.Bug严重级别的定义要求：务必按照Bug定级计算公式得出；

6.Bug规范检查要求：测试人员在写完Bug后，第一时间自己进行检查，确认无误后提交Bug，PTM目前按照2天一次频率抽查Bug规范性；

7.对比类问题的提交【不一定是问题，但是与Android原生或者竞品对比有差异】

7.1.当项目测试中出现与原生Android不一致的情况时，提交Bug时，需要准确的描述原生的Android是怎么样的，我们的软件现象是怎么样的，两者的差异是什么，这类问题极有可能上升为争议类问题，由PM组织评审是否解决；

7.2.当项目测试中出现与竞品表现不一致的情况时，提交Bug时，需要准确的提供竞品对比的测试数据，以证明我们产品在与竞品的对比中处于劣势或者不能满足某一种用户场景，一定是有对比数据的支撑；

8. 提交Bug时如果遇到以下三类Bug：

            1.测试只看到一次，从来没有复现，开发也没看到过现象，纯once的Bug。

            2.测试发现问题，拿给开发看到了，但是从来没有复现过，纯once的Bug。

            3.测试发现问题，开发没看到/看到了，测试很难复现，没找到真正的复现。

 开发如果把bug再走回给我们，必须看到他们规范的回复：已分析hh:mm-hh:mm时段log，通过分析XXX字段，发现无异常。如果没有看到回复信息,请将Bug转给我.

对于以上三类Bug,提交之后,请把对应的Leader添加到跟踪者里.(U3D->谢明辉 ， Android->李强,   驱动->孙启东,  屏幕驱动显示->张毅 )



# 四.Bug处理规则

## 4.1 测试目的

* 及时验证Bug是否被修复且未引入新问题
* 根据RD Fix的Bug情况，确定重点测试方向
* 协助开发处理Need Clarification, Not Reproduce的问题
* 及时处理 Reject的Bug,分析原因,保证项目质量

## 4.2执行流程



Bug管理流程图

说明：

1.只有验证未通过的Bug才能Reopen,其余的情况都是转为New

2. Bug状态只要有变更,就一定要写上备注

3.如果在处理Reject Bug的过程中遇到死循环,需求或者设计方面的Bug请联系产品经理，如果是功能性有争议的Bug请联系项目经理

4.如果是Fix但是需要长期跟踪的Bug,可以将Bug改为Monitor,待产品出货以后再关闭

5. 如果是老Bug已修复但是引起新Bug,请关闭老Bug，另开新Bug

## 4.3QA验证Bug规则

|  |  |  |
| --- | --- | --- |
| Bug状态 | 执行规则 | 回复模板 |
| Fix | 发生概率为100%的Bug，验证5次之后关闭，并附上验证Version | 此Bug在Vx.x.x上验证Pass.验证次数：0/5 |
| 发生概率非100%但可以通过多次操作重现的Bug，验证30次之后关闭，并附上验证Version | 此Bug在Vx.x.x上验证Pass.验证次数：0/30 |
| Once/Twice的Bug，如果Bug属于应用层的问题或者Bug严重级别在Fatal以下,则在连续三周之内,每周抽取一个相对稳定的版本验证30次，若三个版本都不能重现,则将Bug关闭并附上验证Version.如果Bug属于驱动层且Bug级别在Fatal及其以上,则每周都需要抽取一个版本进行验证,直到产品出货以后关闭. | 此Bug在Vx.x.x & Vy.y.y&Vz.z.z上验证未重现.验证次数：0/30 |
| 若Bug未修复，需将状态改为reopen，附上验证Version以及附件，且将Owner切回修改此Bug的RD | 此Bug在Vx.x.x上验证Fail，具体现象：……… 验证次数：x/y. |
| 若Bug已经修复，但是引发其他新问题，则关闭此Bug再另外开一个Bug，并将新Bug title加入此Bug备注. | 此Bug在Vx.x.x上已经修复，但是引发新问题.关闭此问题，继续追踪新Bug#xxxxx |
| Not Reproduce | 若能重现则将状态改为New，补充详细信息和附件或直接向RD当面演示;若无法重现,如果Bug属于应用层的问题或者Bug严重级别在Fatal以下,则在连续三周之内,每周抽取一个相对稳定的版本验证30次，若三个版本都不能重现,则将Bug关闭并附上验证Version.如果Bug属于驱动层且Bug级别在Fatal及其以上,则每周都需要抽取一个版本进行验证,直到产品出货以后关闭. | 此Bug在Vx.x.x & Vy.y.y & Vz.z.z上验证未重现.验证次数：0/30 |
| Reject | 如能接受RD的理由则Close，如果不能接受，则将状态New，并写上备注，将Owner转回对应的RD. 如果是需求或者设计方面的Bug则转给产品经理，如果是功能性有争议的Bug则转给项目经理 | 根据Bug实际情况进行描述 |

## 4.4需求类问题处理

1.     【需求测试阶段】发现的需求文档\交付设计\UI图问题，一律由测试提交给产品经理，由产品经理逐级解决，最后产品经理需要在文档上进行更新后关闭；

2.     【版本测试阶段】发现的需求缺失、需求定义有误等需求相关问题，如果是以功能Bug的现象被发现，测试提交Bug到对应的开发人员并同时添加产品经理为跟踪人员，由开发人员找产品经理确认后，产品经理更新需求稳定，开发人员按产品经理的设计修复问题；

3.     【版本测试阶段】发现的需求缺失、需求定义有误等需求相关问题，如果是非功能Bug的现象被发现，而是测试基于用户体验、个人感觉的原因提出，建议类问题一律由测试人员提交建议给产品经理，由产品经理来判断是否修改设计；

## 4.5争议类问题处理

1.     当项目中出现有争议的Bug时【已确认是问题，争议的焦点是Bug是否解决】，测试和开发人员应先做沟通，如果无法协商到一致，则由Bug owner转交给PM处理，由PM与PTM在项目过程中分阶段商定Bug的解决范围；

2.     当项目中出现有争议的Bug现象时【已确认是问题，已确认需要解决，争议的焦点是Bug如何解决】，测试和开发人员应先做沟通，拒绝私下协商，由Bug owner转交给产品经理，由产品经理确认问题后做设计更改，开发人员按产品经理的设计修复问题（此过程中需要测试人员给产品经理提供自己的观点或是具体数据，而不是独立的提出修改建议）；

3.     当出现验证Bug时，发现Bug的部分被修改，部分因为各种原因没有修过到预期结果时，拒绝关闭，必须要达到期望结果要求时才能关闭；

## 4.6 开发人员(RD)权责

* 此处的开发人员泛指Android，U3D，设计和驱动组的所有同事；
* 每日及时Review指派给自己且状态为New或者Reopen的Bug：
* 如果接受此Bug，并且着手开始修复，则将状态改为Accept并填写计划修复完成的日期，此日期作为QA部门追踪修复进度的标准；
* 如果认为此Bug属于产品设计，不认同是Bug，则将状态改为Reject，并写上原因，同时将Owner指派给Bug作者；
* 如果此Bug缺失Log，或者有不理解的地方需要QA澄清，则需要将状态改为Need Clarification，并填写需要澄清的具体内容，同时将Owner指派给Bug作者，如比较紧急还应立即告知Bug作者立即处理；
* 如果此Bug缺失Log且无法再复现，则需要将Bug状态改为Not Reproduce，并填写无法复现的版本号和一些帮助QA重新复现的分析说明，同时将Owner指派给Bug作者；
* 如果此Bug和其他Bug重复或者同属于一个Root Cause，则需要将Bug状态改为Duplicate，并且备注和哪一个Bug重复，同时将Owner指派给Bug作者；
* 如果此Bug在经过分析之后应归属其他RD来负责解决，则需要将Bug Owner指派给对应的RD，并且详细备注原因；
* 如果此Bug需要其他RD来协助解决，请Bug Owner及时与对应RD联系并确认其解决日期，同时将此日期备注到Comment中；Bug修复的计划完成日期在对应RD解决日期之后累加；切勿此时将Bug Owner切换到QA；
* Bug修复完成之后，RD需要先使用版本自测，自测通过之后将状态需要改为Fix，并详细注明自测的版本（或者指明QA验证的版本），Bug原因分析和修复说明，同时将Bug Owner切换为Bug作者；
* 对于其他RD转过来的Bug如未详细注明原因，有权直接打回并提请对方RD Leader注意，应尽量避免此类问题发生
* RD不能将未修复的Bug直接切Fix或指派给QA；如果确实不能修复，需要将Bug Owner指派给项目经理；项目经理需和RD Leader及QA协商，如确认不修复，由项目经理转QA关闭此Bug，如确认需要修复，由项目经理将Bug转到对应的RD；

## 4.7 研发组长权责

* 每日及时Review本组状态为New或者Reopen的Bug，及时督促组员处理；当日17：00前New或者Reopen Bug≥5且排名居首的RD组长将被Highlight；
* 每日及时Review本组状态为Accept的Bug的计划完成日期，临近或者已过计划完成日期的Bug，及时督促和指导组员，及时澄清原由；超出计划完成日期Bug数≥5且排名居首的RD组长将被Highlight；
* 每日及时Review本组的所有Open Bug，协调组内人力，避免某个RD负荷过重导致Bug修复延期；
* 如果组内RD请假，RD对应的Bug应由组长或者组长指派其他同事来及时处理；

## 4.8 产品经理权责

* 每日及时Review指派给自己且状态为New或者Reopen的新功能需求：
* 如果接受此需求，则将状态改为Accept并填写计划完成的日期，此日期作为QA部门追踪进度的标准；产品经理完成需求说明书之后再将Bug指派给RD，同时附上详细需求定义文档；
* 如果不接受，请切换状态为Reject，并详细备注原因；
* 如果Bug在当前版本不做修改,则将Bug改为Pending

## 4.9 QA权责

* 每日及时Review自己提交的Bug：
* 状态为New和Reopen且超过1天RD未处理，QA应及时提醒对应RD去处理，如果RD请假，则告知对应RD Leader去处理；
* 状态为Accept但是计划完成日期已到期，QA应及时提醒对应RD和RD Leader并澄清原由；
* 状态为Fix，Not Reproduce，Reject，Need Clarification，Duplicate，Not Reproduce的Bug，QA应每日处理完毕；需要Monitor的Bug不切Pending状态；
* 状态为Pending的Bug，QA应每日查看当前版本是否达到验证条件并及时处理；

## 4.10各角色对Bug状态转换的权限



* **状态转换规则说明：**
* 缺陷参与的角色有: QA, RD(包括硬件,驱动,Android,U3D,设计,PDM), PM(项目经理).
* 跟踪标签更换为”缺陷/需求(TBD)”, 取消之前以设置部门为跟踪标签的情况.
* 缺陷的最终状态有Reject, Duplicate, Closed
* 仅PM有权Pending缺陷, 其余角色无权限
* **指派规则说明：**
* QA提交Bug时根据模块Owner划分表初始指派给RD(必须指派RD,不可置空), Redmine不能做权限设定, QA提交时需特别注意.
* RD不可指派Bug给QA(Redmine已做权限设定), RD和QA对缺陷有争议时可指派给PM裁决.
* 最终状态的Bug需指派为空:

QA确认Reject, Duplicate无误之后即为最终状态,必须指派为空.

Closed直接是最终状态, 也必须指派为空.

* **注意事项：**
* Bug未到达最终状态时, 会一直挂在RD头上, 直到达到最终状态经QA确认之后指派为空.
* QA需每日及时处理Reject, Duplicate, Need Clarification, Monitor, Fix的Bug,尽可能降低因QA的原因而影响RD解决Bug的进度, 否则会被RD complain
* QA提交Bug时尽量注意不要提交By Design或者重复的Bug, 可能会被RD complain, QA组内部也会有对应处理.