## Bug编写规范

**背景**

报告软件测试错误的目的是为了保证修复错误的人员可以重复报告的错误，从而有利于分析错误产生的原因，定位错误，然后修正之。因此，报告软件测试错误的基本要求是准确、简洁、完整、规范。需要掌握的报告技术归纳如下：

**Bug规范**

1. 测试人员发现的所有Bug都要记录到JIRA中，不允许口头传达；\*\*\*\*\*(重要)
2. 项目：要在正确的项目中提交bug，如FTP客户端，可直接在FTP客户端中点击创建问题，也可在其他项目中创建问题后点击项目后的下拉菜单选择对应的项目。\*\*\*\*\*(必需填写)
3. 模块名称：选择bug出现的对应模块名称，如果多模块，请多选。
4. 问题类型：
5. 如果属于功能Bug，类型选择“功能Bug”。
6. 如果属于页面Bug，类型选择：“页面Bug”。
7. 如果属于建议，类型选择：“建议”。
8. 如果属于问题，类型选择：“问题”。
9. 主题：要简洁、准确，完整，揭示错误实质；不要太长。要确信标题是足够短，以便能够在显示错误的屏幕上和由缺陷跟踪系统生成的报表中显示完全。标题不必是完美的语法，而应简短并一针见血。 如果只有个别浏览器中出现该问题，可以在标题中标明该浏览器。 \*\*\*\*\*(必需填写)
10. 严重程度：可参考最下放的《Bug严重程度及优先级的划分》。
11. 优先级：可参考最下放的《Bug严重程度及优先级的划分》。
12. 发现版本：选择对应测试的版本。（测试提交bug的时候填写）
13. 修复版本：选择对应验收的版本。（开发修复bug的时候填写）
14. 环境：环境主要填写测试的浏览器环境或软件的运行系统环境。
15. 描述：步骤描述要完整，准确，简短 保证快速准确的重复错误，“完整”即没有缺漏，“准确”即步骤正确，“简短”即没有多余的步骤。\*\*\*\*\*(必需填写)
16. 期望结果：写出期望得到的正确结果或者测试用例中的预期结果。
17. 实际结果：每个Bug都要有一个实际结果，不好重现的一定要抓图。不好描述也的可以抓图，将图上传。\*\*\*\*\*(必需填写)
18. 报告人：系统会自动生成无需填写。
19. 修复人：需要测试人员进行指派，指派给对应的开发进行修复。
20. 备注：填写测试用例编号和抓图的编号，或有好的想法或者额外信息，要写进去。提供解决Bug额外的信息，这也是为解Bug提供信息。总之信息越多越好。 有无规律的Bug一定要在备注说明，或只有自已环境出现的Bug也要注明。

**其他需要注意的地方：**

1、检查拼写和语法错误　 在提交每条缺陷或错误之前，检查拼写和语法，确保内容正确，正确的描述错误。自己多读两遍。

2、 尽量使用业界惯用的表达术语和表达方法

3、每条错误报告只包括一个错误，可以使错误修正者迅速定位一个错误，集中精力每次只修正一个错误。校验者每次只校验一个错误是否已经正确修正。

4、 根据错误的现象，总结判断错误的严重等级。 如果不能正确定位就默认为“一般Bug”。

1. 要尽量避免提交重复Bug。如果和别人共同测试一个模块，要看看别人有没有提这个bug。 如果自己单独测试，要看看类似功能有没有相同的问题，如果有相同的问题，只需要提一个bug，在备注中说明有类似问题的功能和测试用例编号， 。必要时可以向程序员请教和沟通。

### Bug严重程度及优先级的划分

Priority(优先级)和Severity(严重程度)是提交bug的两个重要属性。通常， 人员在提交Bug时，只定义Bug的Severity, 即该Bug的严重程度，而将Priority交给Project Leader 或Team Leader来定义，由他们来决定该Bug被修复的优先等级。某种意义上来说，Priority的定义要依赖于Severity，在大多数情况下，Severity越严重，那这个Bug的Priority就越高。你知道如何合理定义bug的Sevrity么？ 通常Bug管理系统里Severity分为四个等级Blocker, Critical, Major, Minor/Trivial（也可自定义，但通常是这四个）, 而priority分为五个等级：Immediate, Urgent, High, Normal, Low。

**Severity（严重程度）**

1. Blocker（有妨碍的）:

即系统无法执行、崩溃或严重资源不足、应用模块无法启动或异常 退出、无法测试、 造成系统不稳定。

严重花屏

内存泄漏

用户数据丢失或破坏

系统崩溃/死机/冻结

模块无法启动或异常退出

严重的数值计算错误

功能设计与需求严重不符

其它导致无法测试的错误, 如服务器500错误

2.Critical（紧要的）：

即影响系统功能或操作，主要功能存在严重缺陷，但不会影响到系统稳定性。

功能未实现

功能错误

系统刷新错误

数据通讯错误

轻微的数值计算错误

影响功能及界面的错误字或拼写错误

安全性问题 

3. Major（严重的）：

即界面、性能缺陷、兼容性。

操作界面错误（包括数据窗口内列名定义、含义是否一致）

边界条件下错误

提示信息错误（包括未给出信息、信息提示错误等） 

长时间操作无进度提示

系统未优化（性能问题）

光标跳转设置不好，鼠标（光标）定位错误

兼容性问题

4.Minor/Trivial（次要的/不严重的）:

即易用性及建议性问题。 

界面格式等不规范

辅助说明描述不清楚 

操作时未给用户提示

可输入区域和只读区域没有明显的区分标志

个别不影响产品理解的错别字

文字排列不整齐等一些小问题

**Priority（优先级）**

1.Immediate（立刻）

即“马上解决”，表示问题必须马上解决，否则系统根本无法达到预定的需求。

2. Urgent（紧要、优先）

即“急需解决”，表示问题的修复很紧要，很急迫，关系到系统的主要功能模块能否正常 。

3. High（高度重视）

即“高度重视”，表示有时间就要马上解决，否则系统偏离需求较大或预定功能不能正常实现。

4. Normal（正常）

即“正常处理”，进入个人计划解决，表示问题不影响需求的实现，但是影响其他使用方面，比如页面调用出错，调用了错误的等。

5. Low（稍缓）

即”低优先级”，即问题在系统发布以前必须确认解决或确认可以不予解决。