

怎样提交BUG



提交一个好的bug，开发人员看到bug之后，很快能从描述中知道这个bug在什么环境下出现，怎样重现bug。通过bug中附带的截图或log，有助于定位bug的原因，减少修复bug的时间。

很多软件公司，在入职培训时一般都包含《怎样提交bug》这样的课程；而不少初创公司因为种种原因，提交bug很不规范，有的就是一个截图，有的就是一句话，开发人员不得不花大量的精力与bug提交者沟通细节，在弄清楚原委之后，才开始定位问题，花费了不少沟通时间。最近接触到的几个团队就属于后者。

为什么写这篇文章？

犹豫着要不要写比较基础的文章，想起了《Great CEO Within》这本书里的一段话：**如果一句话要说两次，把它写下来**

。好，我还是写下来吧。

先有一个好的缺陷管理工具，Jira、Bugzilla、Git等等，挺多的，选一个适合自己的就好。

现在，就可以提交bug了。开发人员、测试人员、用户都可以提交bug。

一个bug有如下几个元素。

□ 标题

用于简单描述这个bug，不要太长，简洁一点。标题的目的是方便检索，一目了然。

□ 描述

标题只是概括的描述一个bug，至于细节，就应该在描述部分写清楚了。

□ 重现步骤

怎样重现一个bug，最好能详细地描述重现的步骤，是稳定的重现，还是偶尔重现。有助于开发人员定位问题。

□ 期望结果、实际结果

根据之前的重现步骤，期望中的结果是怎样的，实际结果是怎样的。有时候，实际结果就是当初的设计，而bug提交者的“期望行为”，与设计的期望行为并不相符。对比期望结果和实际结果，开发人员很容易辨识这种情况。

□ 截图、Log

截图有助于开发人员了解bug的情况，而Log有助于开发人员定位bug的原因。

□ 优先级

根据重要性设置bug，阻碍用户使用的bug，优先级最高，需要立即解决；而不影响用户使用的bug，如文字对齐等，可以优先级调低点，在高优先级的bug修复完之后再修复。

□ 环境

写明操作系统，是Windows、Linux，还是MacOS。如果是Windows，还应写清楚Windows的版本号，是Win 7还是Win 10。同理，如果是Linux，写清楚是Ubuntu 18.10，还是SUSE Linux Enterprise Server 15。

如果是Web应用，还要加上浏览器的名称和版本。

如果不写清楚，很有可能开发人员无论如何都没法重现bug，联系bug提交者之后才知道，需要在ubuntu上才能重现，而他之前都是在window上尝试重现的。花费10多秒钟，避免可能浪费的10分钟，很有价值的。

□

版本、目标版本

版本方便开发者知道当前出错的版本是什么，而目标版本方便bug提交者管理自己的期望，并知道何时验证它。能够重现这个bug的当前系统版本是什么，即将在哪个版本中修复它。否则，当bug提交者看见bug处于修复状态，但他去验证的时候，发现仍然没修复。与开发人员联系，才知道目标版本是3.1，而他验证的版本是2.2。

□

开发人员 (Assignee)

即修复这个bug的开发人员，方便联系和跟踪。开发人员如果知道谁来修复，可以直接设置开发人员，如果不知道，可以设置一个默认者，之后默认者再分配给相应的开发人员。

□

问题根因与解决方案

这是开发人员要填的。在修复bug时，注明问题的原因是什么，以及怎样解决的。

最后，给一个Eclipse的样例吧：https://bugs.eclipse.org/bugs/show_bug.cgi?id=204101。

做好上面几点，可以为开发人员节省大量的时间。

□

相关文章：

□

使用Nginx实现服务器中多容器共存

□

怎样让小程序服务器免备案

□

利用CSS3的animation实现闪亮的效果