如何写有效的缺陷报告

这是翻译一个名字叫做 Kelly Whitmill 的人写的文章，他在写这篇文章之前有 18 年的软件测试经验，18 年中主要做为 team leader 来负责通过寻找并且实施有效的测试方法和测试工具来达到系统的要求，擅长利用有限的资源来尽可能的模拟环境，在自动测试上有浓厚的兴趣。从大公司到小公司都有过丰富的经历，现在在 IBM 公司工作。

介绍

缺陷报告是测试过程中可以提交的最重要的东西。它的重要性丝毫不亚于测试计划，并且比其他的在测试过程中的产出文档对产品的质量的影响更大。所以很有必要学习如何写出有效的缺陷报告。有效的缺陷报告将能够：

- 减少开发部门的二次缺陷率
- 提高开发修改缺陷的速度
- 提高测试部门的信用度
- 增强测试和开发部门的协作

为什么测试人员从开发那里得到的反馈比从其他部门得到的更多？一定程度上这个答案就是缺陷报告，依照一些简单冶）O他捐）- @这单熜的单　砑ＺＳ　　针~褪　<蕲事>　缺陷报告，而

缺陷注释

下面是确保你下一篇缺陷报告是有效的几个关键点：

- Condense－精简，清晰而简短
- Accurate－准确，这到底是不是一个 bug？还是用户操作错误，或者是理解错了，等等？
- Neutralize－用中性的语言描述事实，不带偏见，不用幽默或者情绪化的语言。
- Precise－精确，这到底是什么问题？
- Isolate－定位，这到底是个什么样的问题？尽量缩小这个问题的范围。
- Generalize－还有没有其他的某些地方存在这样的问题？
- Re-Create－如何引发和重现这个 bug？（环境，步骤，前提条件）
- Impact－影响，　ＮｍＥ弧　稽猵没疽　，　/　　4　；

- Evidence－证据，如何证明确实存在这个 bug？

写有效的缺陷报告并不需要很好的文字功底，只要确认你正确回答了上面的问题，关键就是确认你覆盖了所有的你的缺陷报告的查看者关注的要点就好了。

有效缺陷注释的要点

精简

清晰而简短。首先，去掉不必要的词；其次，不要添加无关的信息。包含相应的信息是最重要的，但是确保这些信息都是有用的。不管什么原因，对于那些没有描述清楚如何重现或者难以理解的问题，你都应该提供更多的信息。写过多的不必要的信息也是问题的一种。

精简的例子	缺陷注释
不要这样写： TMI（Too Much Information）	当我正在专心测试的时候，报内存错误，这时我发现一个我不熟悉的 GUI，我试了好多边界值以及错误的条件，但是运行正常。最后我请空了数据，并且点击了前进按钮，这时系统异常中止了，多次的反复尝试证明，在任何情况下，只要“ 产品描述 ”这个字段没有数据，点击前进或者退出甚至取消，系统都会中止。
要这样写：	在产品信息页面，如果产品的描述字段为空，前进，退出和取消的功能会使系统意外中止。

中得到了教训和经验。

中性的语言

客观的说这个问题，测试人员在报 Bug 时，不要使用幽默的或者其他带有感情色彩的语句。在你看来好笑的问题，，对于那些迫于 schedule 的压力，日夜加班的做出这些东西开发人员来说就不见得可笑了。用带有感情色彩的语句，把 bug 报的声情并茂除了造成 team 内部的沟通屏障和协作困难外，对修改 bug 没有任何好处。甚至如果开发人员曾经怀疑你并且打回一个你报的 bug，而现在你找到证据证明你是对的的话，也不要这么做，你要做的就是报出这个问题，并且加上对开发人员有帮助的信息，长此以往，有助于增加你的信誉度。在提交 bug 之前，仔细阅读你的 bug 的描述，删除或者修改哪些可能让别人产生歧义的句子。

中性语言例子： 这个例子是测试人员在开发打回一个 bug，要求提供更多的信息还有出错的数据时的回复	Bug 的注释
不好的：	输入任何的非法数据都会中止。
好的：	功能 ABC 在任何非法的数据都会中止，例如：-1,-36,-32767

精确

看你的 bug 报告的人可能不需要知道这是什么样的问题，所以测试人员有必要正确的描述出自己所希望的情况，其中一些描述是操作步骤和结果，例如：“ 我按了回车键，然后现象 A 出现，接着按了后退键，现象 B 出现，接着输入命令 ‘ XYZ ’，现象 C 出现 ”，看到这样的说明，很难让别人明白你到底想说明什么问题，三个现象中哪一个是错误的。无论如何，当 Bug 的描述很长时，一定要在开头对这个 bug 做概要的描述。不要认为你在 bug 报告中写的那些抽象的话别人能够理解，也不要认为看了你的描述，别人会跟你有一样的结论。你的目的不是写一份很难让别人理解的高深的文章，而是一份不被别人误解的描述。想做到这个只有清晰准确并且客观的描述你发现的问题，而不是简单说明发生了什么。

精确例子：	Bug 描述
不好的： 这个藐视很难让人知道到底是什么问题。是打印端口延时还是打印机没有准备好或者是打印机的显示面板的信息不正确。	问题发生在取消打印时打印端口延时，打印机的就绪灯始终不亮，此时打印机显示面板上显示“ PRINTING IPDS FROM TRAY1 ”
好的描述： 在描述之前，用简短的语言概述时如何发生的这个问题。	当打印机正在打印时，取消打印会让打印机挂起。 问题发生在取消打印时打印端口延时，打印机的就绪灯始终不亮，此时打印机显示面板上显示“ PRINTING IPDS FROM TRAY1 ”

定位

对于测试人员应该把问题定位到什么程度，每个公司或者每个部门都有不同的规定和期望值。不管这些要求有什么不同，对于一个测试人员来说必须做一些有效的事情来定位发现的问题。在试图隔离一个问题的时候，需要考虑下面的几点：

- 尝试着找到最短，最简单的步骤来重现这个问题，这通常需要花很长的时间。
- 问问自己会不会是外部的什么特殊的原因引起的这个问题？例如，系统挂起或延时，会不会是因为网络的问题？如果你在做点对点的测试，你是否能够说出是哪一个组件出现了错误？有没有什么办法来缩小判断出错组件的范围？
- 如果你在测试一个存在多种输入条件的项目，可以尝试不断的改变输入值，然后查看结果，直到你找到是哪个值导致的错误。

在问题描述中，在尽可能的范围内，精确描述你所使用的测试输入值。例如，如果你在测试中发现打印一份脚本的时候会出错，那你应该首先想到是不是打印所有的这种类型的脚本都会出错。

测试人员定位 bug 的能力，一定程度上是自己做为一个 tester 的附加价值的体现。有效的 bug 定位能够节省同一小组的所有的人的时间，同时也节省了你回测 bug 的时间。

归纳

一般情况下，开发人员只会“ 精确的 ”按照你的报告来修改 bug，而不会因为测试人员没有进行归纳而把相应的其他的相关 bug 都改好。例如，我报了这样一个 bug，就是我的文字处理程序的保存功能有问题，当保存一个名为“ 我的文件 ”的文件时文字处理程序会当掉。但是这样的情况同时发生在保存一个文件长度为 0 的文件，或者试图保存在一个远程的磁盘，一个只读磁盘时都会产生这个情况。如果能这样写的话会节省开发很多的时间，并且能很大的提高系统的质量。

当你发现一个问题时，采用合理的步骤来确定这个问题是通常会发生还是偶然一次出现或者是在特殊条件才出现。

归纳示例：	Bug 报告内容
不要这样写：	当用非法字符做为文件名时，系统会出现“ 文件找不到 ”的错误提示信息。
要这样写：	当用非法字符做为文件名时，系统会出现“ 文件找不到 ”的错误提示信息。每一次当我插入字符时都存在这个问题，但是不插入字符就没有问题。

重现

有些 bug 很容易就重现，有些就很难，如果你能重现一个 bug，你应该准确的解释必需的条件。你应该列出所有的步骤，包括精确的组合，文件名以及你碰到或者重现这个问题的操作顺序。如果你能构确认这个问题在任何文件，任何的操作顺序等条件下都会发生，那也最好能够给出一个明确的示例用来帮助开发来重现。

如果你经过测试却发现不能再重现一个问题，那就尽可能多的提供有效的信息给你的开发人员。在开发没有重现或者开发没有给你反馈之前，不要清除你的测试数据，或者至少你要备份这些数据。在你没有验证这个问题如何重现之前不要确信一个问题是可以重现，如果你无法重现或者没有做验证是否可以重现，一定要标注在 bug 的报告中。

影响

如果在客户的环境中出现一个 bug 的影响是什么？很多 bug 是很明显的。例如：进入系统时，敲击回车键，系统会 down 掉。还有的问题不是这么明显的，你可能在某一个窗口发现一个排版错误或者拼写错误，对于测试人员来说可能会觉得是很小的问题，甚至是微不足道的问题，但是对于用户来说，这是他接触你的产品的第一件事，而且如果这个单词事比较敏感的话，就很不妙了，在这种情况下，即使我们认为这是很小的问题，也必须在给客户实施前修改好。如果你觉得这个问题不会给客户带来重大的影响，那就可以发布出去了。

调试

开发人员会如何调试这个问题？有没有跟踪、截图、日志等对捕获这个 bug 有帮助的信息包含在你的 bug 报告中？文件的位置和访问权限设定的如何？

证据

有什么可以证明这个 bug 确实存在？你是发已经提供了期望结果和实际结果？有没有文档来支持你的期望结果？既然你提交了 bug，也就意味着你认为这是一个 bug 了，那就提供你可以提供并且可以说服其他人认可这确实是一个 bug 的信息吧！这些信息可能包括操作指导，文档，必备条件等等，还有可能是客户以前反馈过来的零碎的信息，或者是竞争对手的软件中的一些标准，又或者来源于以往版本中的结果。并且不要认为每个人看到一件事情后的反应都跟你一样，也不要期望别人能跟你得出一样的结论，并且不要认为在过去三个礼拜以后你还能认为这是一个 bug。考虑一个所有支持你认为这是 bug 的原因并且归入你的 bug 报告中。如果你意识到别人可能认为它不是 bug 的话，你也必须这样做。

强化记忆

你不需要每次报 bug 的时候都来查看一下上面的关注点，你需要强化记忆，然后在你每一次报 bug 的时候可以信手拈来。这是提高软件质量消费成本最小并且最有效的办法，即使有可能这只是自己认为的。通常一份写的不好的 bug 报告不是因为我们不能写好，而是因为我们没有考虑和回答正确的问题，这就是强化记忆的作用。

不知道你有没有发现一个有趣的问题，就是在前面提到的那些单词的首字母合起来就是“ CAN PIG RIDE

这个问题有没有被有效的隔离和定位？

有没有对这个问题进行归纳总结？

附加 Bug 的调试信息。（如何查看系统的日志，截图等）

有效的 bug 报告，在很多情形下，并不是全都必须的，而是通过回答正确的问题来得到有效报告的充分条件。下面的十点：

- Condense
- Accurate
- Neutralize
- Precise
- Isolate
- Generalize
- Re-Create
- Impact
- Debug
- Evidence

提供了一个快速的列表来保证你确实回答了正确的问题，并且如果这样做的话会对你在做的项目或者你所在的部门甚至整个公司都有好处。

Bug 的标题

Bug 的标题在很多情况下是一个有力的和项目组成员之间的沟通工具，在很多情形下，能够做决定的人都不会去仔细的看 bug 的描述，而只是查看 bug 的标题，然后就下结论。通常的产品经理，team leader 还有其他的经理们也都是只会关注到 bug 的标题。

Bug 的标题必须简短而且要求描述和传达出准确的信息。因为有长度的限制，这个概要的描述一般都比较短，使用一些不会引起歧义的简写比好的语法和句子更好。因为许多使用者习惯于用关键词来搜索，所以在 bug 的标题中使用一些精练的关键词事很有必要的，象挂起，异常中止，拼写错误等都是比较有效的和有力的搜索关键字。如果长度允许，在 bug 的标题中有必要加上诸如环境，影响等 5 个 W（why，when，who，where，what）的问题。

有一些 bug 跟踪的工具会自动把 bug 的第一行的 bug 描述做为 bug 的标题，永远不要使用这样的默认标题，要尽可能的特殊和精确。例如：下面的标题就没有提供足够多的信息。

标题：在保存和恢复数据成员时出错。

一个比较好的标题可能是这样：在 WINNT 环境下，XYZ 的保存和恢复数据失败，数据丢失。

通常在 bug 的标题中你不会得到任何你想得到的信息，下面是一个写 bug 标题应该注意的几点：

简单，明确的说明问题（不能只是说出现问题）

建议（如果长度允许的话）：

- 使用有意义的单词
- 描述环境和影响

- 回答5W1H的问题
- 使用简写
- 相对于描述清楚而言，语法不是很重要
- 不要使用测试工具提供的默认的标题

总结

测试人员为了寻找和发现软件中的问题会花费大量的时间，一旦发现了，就要想方设法如何用最小的代价来促使这个问题得到解决。确保 bug 报告中包括了必要的适当的信息比高超的写作技巧更重要。

不是所有的人都会完整的查看 bug 的报告，很多拍板的人通常只依靠 bug 的标题来帮助他们做决定，所以用精确的语言来描述你的 bug 标题很有必要。

你的 bug 报告写的越好，在实际中为了修改这个 bug 花费的时间相对来说就会越少。你的信誉度和产品中的贡献度也会得到很好的提升，因为你的 bug 报告够好并且可以信赖，其他测试人员的工作也会得到相应的提升。

参考：
Rex Black, *The Fine Art of Writing a Good Bug Report*,
http://www.rexblack.consulting.com