

如何有效地报告 Bug

How to Report Bugs Effectively

Simon Tatham © 著

Dasn © 译

一九九九年

版权所有 Simon Tatham 1999

本文属于OPL（[OpenContent License](#)），请在复制和使用本文时自觉遵守 OPL。

对本文的任何意见和批评请发送至：

英文版：anakin@pobox.com

中文版：dasn@users.sf.net

目 录

1 引言	2
2 “程序不好用”	2
3 “演示给我看”	2
4 “告诉我该怎么做”	3
5 “哪儿出错了？在我看来一切正常哦！”	3
6 “出了问题之后，我做了……”	3
7 “我想粒子的跃迁与错误的极化有关”	4
8 “真是奇怪，刚才还不好用，怎么现在又好了？”	4
9 “我把磁盘装进了 Windows……”	5
10 小结	5

1 引言

为公众写过软件的人，大概都收到过很拙劣的 bug（计算机程序代码中的错误或程序运行时的瑕疵——译者注）报告，例如：

在报告中说“不好用”；

所报告内容毫无意义；

在报告中用户没有提供足够的信息；

在报告中提供了错误信息；

所报告的问题是由于用户的过失而产生的；

所报告的问题是由于其他程序的错误而产生的；

所报告的问题是由于网络错误而产生的；

这便是为什么“技术支持”被认为是一件可怕的工作，因为有拙劣的 bug 报告需要处理。然而并不是所有的 bug 报告都令人生厌：我在业余时间维护自由软件，有时我会收到非常清晰、有帮助并且“有内容”的 bug 报告。

在这里我会尽力阐明如何写一个好的 bug 报告。我非常希望每一个人在报告 bug 之前都读一下这篇短文，当然我也希望用户在给我报告 bug 之前已经读过这篇文章。

简单地说，报告 bug 的目的是为了让程序员看到程序的错误。您可以亲自示范，也可以给出能导致程序出错的、详尽的操作步骤。如果程序出错了，程序员会收集额外的信息直到找到错误的原因；如果程序没有出错，那么他们会请您继续关注这个问题，收集相关的信息。

在 bug 报告里，要设法搞清什么是事实（例如：“我在电脑旁”和“XX 出现了”）什么是推测（例如：“我想问题可能是出在……”）。如果愿意的话，您可以省去推测，但是千万别省略事实。

当您报告 bug 的时候（既然您已经这么做了），一定是希望 bug 得到及时修正。所以此时针对程序员的任何过激或亵渎的言语（甚至谩骂）都是与事无补的——因为这可能是程序员的错误，也有可能是您的错误，也许您有权对他们发火，但是如果您能多提供一些有用的信息（而不是激愤之词）或许 bug 会被更快的修正。除此以外，请记住：如果是免费软件，作者提供给我们已经是出于好心，所以要是太多的人对他们无礼，他们可能就要“收起”这份好心了。

2 “程序不好用”

程序员不是弱智：如果程序一点都不好用，他们不可能不知道。他们不知道一定是因为程序在他们看来工作得很正常。所以，或者是您作过一些与他们不同的操作，或者是您的环境与他们不同。他们需要信息，报告 bug 也是为了提供信息。信息总是越多越好。

许多程序，特别是自由软件，会公布一个“已知 bug 列表”。如果您找到的 bug 在列表里已经有了，那就不必再报告了，但是如果您认为自己掌握的信息比列表中的丰富，那无论如何也要与程序员联系。您提供的信息可能会使他们更简单地修复 bug。

本文中提到的都是一些指导方针，没有哪一条是必须恪守的准则。不同的程序员会喜欢不同形式的 bug 报告。如果程序附带了一套报告 bug 的准则，一定要读。如果它与本文中提到的规则相抵触，那么请以它为准。

如果您不是报告 bug，而是寻求帮助，您应该说明您曾经到哪里找过答案，（例如：我看了第四章和第五章的第二节，但我找不到解决的办法。）这会使程序员了解用户喜欢到哪里去找答案，从而使程序员把帮助文档做得更容易使用。

3 “演示给我看”

报告 bug 的最好的方法之一是“演示”给程序员看。让程序员站在电脑前，运行他们的程序，指出程序的错误。让他们看着您启动电脑、运行程序、如何进行操作以及程序对您的输入有何反应。

他们对自己写的软件了如指掌，他们知道哪些地方不会出问题，而哪些地方最可能出问题。他们本能地知道应该注意什么。在程序真的出错之前，他们可能已经注意到某些地方不对劲，这些都会给他们一些线索。他们会观察程序测试中的每一个细节，并且选出他们认为有用的信息。

这些可能还不够。也许他们觉得还需要更多的信息，会请您重复刚才的操作。他们可能在这期间需要与您交流一下，以便在他们需要的时候让 bug 重新出现。他们可能会改变一些操作，看看这个错误的

产生是个别问题还是相关的一类问题。如果您不走运，他们可能需要坐下来，拿出一堆开发工具，花上几个小时来好好地研究一下。但是最重要的是在程序出错的时候让程序员在电脑旁。一旦他们看到了问题，他们通常会找到原因并开始试着修改。

4 “告诉我该怎么做”

如今是网络时代，是信息交流的时代。我可以点一下鼠标把自己的程序送到俄罗斯的某个朋友那里，当然他也可以用同样简单的方法给我一些建议。但是如果我的程序出了什么问题，我不可能在他旁边。“演示”是很好的办法，但是常常做不到。

如果您必须报告 bug，而此时程序员又不在您身边，那么您就要想办法让 bug 重现在他们面前。当他们亲眼看到错误时，就能够进行处理了。

确切地告诉程序员您做了些什么。如果是一个图形界面程序，告诉他们您按了哪个按钮，依照什么顺序按的。如果是一个命令行程序，精确的告诉他们您键入了什么命令。您应该尽可能详细地提供您所键入的命令和程序的反应。

把您能想到的所有的输入方式都告诉程序员，如果程序要读取一个文件，您可能需要发一个文件的拷贝给他们。如果程序需要通过网络与另一台电脑通讯，您或许不能把那台电脑复制过去，但至少可以说一下电脑的类型和安装了哪些软件（如果可以的话）。

5 “哪儿出错了？在我看来一切正常哦！”

如果您给了程序员一长串输入和指令，他们执行以后没有出现错误，那是因为没有给他们足够的信息，可能错误不是在每台计算机上都出现，您的系统可能和他们的在某些地方不一样。有时候程序的行为可能和您预想的不一样，这也许是误会，但是您会认为程序出错了，程序员却认为这是对的。

同样也要描述发生了什么。精确的描述您看到了什么。告诉他们为什么您觉得自己所看到的是错误的，最好再告诉他们，您认为自己应该看到什么。如果您只是说：“程序出错了”，那您很可能漏掉了非常重要的信息。

如果您看到了错误消息，一定要仔细、准确的告诉程序员，这确实很重要。在这种情况下，程序员只要修正错误，而不用去找错误。他们需要知道是什么出问题了，系统所报的错误消息正好帮助他们。如果您没有更好的方法记住这些消息，就把它们写下来。只报告“程序出了一个错”是毫无意义的，除非您把错误消息一块报上来。

特殊情况下，如果有错误消息号，一定要把这些号码告诉程序员。不要以为您看不出任何意义，它就没有意义。错误消息号包含了能被程序员读懂的各种信息，并且很有可能包含重要的线索。给错误消息编号是因为用语言描述计算机错误常常令人费解。用这种方式告诉您错误的所在是一个最好的办法。

在这种情形下，程序员的排错工作会十分高效。他们不知道发生了什么，也不可能到现场去观察，所以他们一直在搜寻有价值的线索。错误消息、错误消息号以及一些莫名其妙的延迟，都是很重要的线索，就像办案时的指纹一样重要，保存好。

如果您使用 UNIX 系统，程序可能会产生一个内核输出（coredump）。内核输出是特别有用的线索来源，别扔了它们。另一方面，大多数程序员不喜欢收到含有大量内核输出文件的 EMAIL，所以在发邮件之前最好先问一下。还有一点要注意：内核输出文件记录了完整的程序状态，也就是说任何秘密（可能当时程序正在处理一些私人信息或秘密数据）都可能包含在内核输出文件里。

6 “出了问题之后，我做了……”

当一个错误或 bug 发生的时候，您可能会做许多事情。但是大多数人会使事情变的更糟。我的一个朋友在学校里误删了她所有的 Word 文件，在找人帮忙之前她重装了 Word，又运行了一遍碎片整理程序，这些操作对于恢复文件是毫无益处的，因为这些操作搞乱了磁盘的文件区块。恐怕在这个世界上没有一种反删除软件能恢复她的文件了。如果她不做任何操作，或许还有一线希望。

这种用户仿佛一只被逼到墙角的鼬（黄鼠狼、紫貂一类的动物——译者注）：背靠墙壁，面对死亡的降临奋起反扑，疯狂攻击。他们认为做点什么总比什么都不做强。然而这些在处理计算机软件问题时并不适用。

不要做鼯，做一只羚羊¹。当一只羚羊面对料想不到的情况或受到惊吓时，它会一动不动，是为了不吸引任何注意，与此同时也在思考解决问题的最好办法（如果羚羊有一条技术支持热线，此时占线。）。然后，一旦它找到了最安全的行动方案，它便去做。

当程序出毛病的时候，立刻停止正在做的任何操作。不要按任何键。仔细地看一下屏幕，注意那些不正常的地方，记住它或者写下来。然后慎重地点击“确定”或“取消”，选择一个最安全的。学着养成一种条件反射——一旦电脑出了问题，先不要动。要想摆脱这个问题，关掉受影响的程序或者重新启动计算机都不好，一个解决问题的好办法是让问题再次产生。程序员们喜欢可以被重现的问题，快乐的程序员可以更快而且更有效率的修复 bug。

7 “我想粒子的跃迁与错误的极化有关”

并不只是非专业的用户才会写出拙劣的 bug 报告，我见过一些非常差的 bug 报告出自程序员之手，有些还是非常优秀的程序员。

有一次我与另一个程序员一起工作，他一直在找代码中的 bug，他常常遇到一个 bug，但是不会解决，于是就叫我帮忙。“出什么毛病了？”我问。而他的回答却总是一些关于 bug 的意见。如果他的观点正确，那的确是一件好事。这意味着他已经完成了工作的一半，并且我们可以一起完成另一半工作。这是有效率并有用的。

但事实上他常常是错的。这就会使我们花上半个小时在原本正确的代码里来回寻找错误，而实际上问题出在别的地方。我敢肯定他不会像医生这么做。“大夫，我得了 Hydroxydyne（真是怪病——译者），给我开个方子”，人们知道不该对一位医生说这些。您描述一下症状，哪个地方不舒服，哪里疼、起皮疹、发烧……让医生诊断您得了什么病，应该怎样治疗。否则医生会把您当做疑心病或精神病患者打发走了，这似乎没什么不对。

做程序员也是一样。即便您自己的“诊断”有时真的有帮助，也要只说“症状”。“诊断”是可说可不说的，但是“症状”一定要说。同样，在 bug 报告里面附上一份针对 bug 而做出修改的源代码是有用的，但它并不能替代 bug 报告本身。

如果程序员向您询问额外的信息，千万别应付。曾经有一个人向我报告 bug，我让他试一个命令，我知道这个命令不好用，但我要看看程序会返回一个什么错误（这是很重要的线索）。但是这位老兄根本没试，他在回复中说“那肯定不好用”，于是我又花了好些时间才说服他试了一下那个命令。

用户多动动脑筋对程序员的工作是有帮助的。即使您的推断是错误的，程序员也应该感谢您，至少您想去帮助他们，使他们的工作变的更简单。不过千万别忘了报告“症状”，否则只会使事情变得更糟。

8 “真是奇怪，刚才还不好用，怎么现在又好了？”

“间歇性错误”着实让程序员发愁。相比之下，进行一系列简单的操作便能导致错误发生的问题是简单的。程序员可以在一个便于观察的条件下重复那些操作，观察每一个细节。太多的问题在这种情况下不能解决，例如：程序每星期出一次错，或者偶然出一次错，或者在程序员面前从不出错（程序员一离开就出错。——译者）。当然还有就是程序的截止日期到了，那肯定要出错。

大多数“间歇性错误”并不是真正的“间歇”。其中的大多数错误与某些地方是有联系的。有一些错误可能是内存泄漏产生的，有一些可能是别的程序在不恰当的时候修改某个重要文件造成的，还有一些可能发生在每一个小时的前半个小时中（我确实遇到过这种事情）。

同样，如果您能使 bug 重现，而程序员不能，那很有可能是他们的计算机和您的计算机在某些地方是不同的，这种不同引起了问题。我曾写过一个程序，它的窗口可以蜷缩成一个小球呆在屏幕的左上角，它在别的计算机上只能在 800x600 的解析度工作，但是在我的机器上却可以在 1024x768 下工作。

程序员想要了解任何与您发现的问题相关的事情。有可能的话您到另一台机器上试试，多试几次，两次，三次，看看问题是不是经常发生。如果问题出现在您进行了一系列操作之后，不是您想让它出现它就会出现，这就有可能是长时间的运行或处理大文件所导致的错误。程序崩溃的时候，您要尽可能的记住您都做了些什么，并且如果您看到任何图形，也别忘了提一下。您提供的任何事情都是有幫助的。即使只是概括性的描述（例如：当后台有 EMACS 运行时，程序常常出错），这虽然不能提供导致问题的直接线索，但是可能帮助程序员重现问题。

¹ 声明：我从没有真的看见过鼯和羚羊，我的比喻可能不恰当。

最重要的是：程序员想要确定他们正在处理的是一个真正的“间歇性错误”呢，还是一个在另一类特定的计算机上才出现的错误。他们想知道有关您计算机的许多细节，以便了解您的机器与他们的有什么不同。有许多细节都依仗特定的程序，但是有一件东西您一定要提供——版本号。程序的版本、操作系统的版本以及与问题有关的程序的版本。

9 “我把磁盘装进了 Windows……”

表意清楚在一份 bug 报告里是最基本的要求。如果程序员不知道您说的是什么意思，那您就跟没说一样。我收到的 bug 报告来自世界各地，有许多是来自非英语国家，他们通常为自己的英文不好而表示歉意。总的来说，这些用户发来的 bug 报告通常是清晰而且有用的。几乎所有不清晰的 bug 报告都是来自母语是英语的人，他们总是以为只要自己随便说说，程序员就能明白。

- **精确。**如果做相同的事情有两种方法，请说明您用的是哪一种。例如：“我选择了‘载入’”，可能意味着“我用鼠标点击‘载入’”或“我按下了‘ALT+L’”，说清楚您用了哪种方法，有时候这也关系。
- **详细。**信息宁多毋少！如果您说了很多，程序员可以略去一部分，可是如果您说的太少，他们就不得不回头再去问您一些问题。有一次我收到了一份 bug 报告只有一句话，每一次我问他更多事情时，他每次的回复都是一句话，于是我花了几个星期的时间才得到了有用的信息。
- **慎用代词。**诸如“它”，“窗体”这些词，当它们指代不清晰的时候不要用。来看看这句话：“我运行了 FooApp，它弹出一个警告窗口，我试着关掉它，它就崩溃了。”这种表述并不清晰，用户究竟关掉了哪个窗口？是警告窗口还是整个 FooApp 程序？您可以这样说，“我运行 FooApp 程序时弹出一个警告窗口，我试着关闭警告窗口，FooApp 崩溃了。”这样虽然罗嗦点，但是很清晰不容易产生误解。
- **检查。**重新读一遍您写的 bug 报告，您觉得它是否清晰？如果您列出了一系列能导致程序出错的操作，那么照着做一遍，看看您是不是漏写了一步。

10 小结

- bug 报告的首要目的是让程序员亲眼看到错误。如果您不能亲自做给他们看，给他们能使程序出错的详细的操作步骤。
- 如果首要目的不能达成，程序员不能看到程序出错。这就需要 bug 报告的第二个目的来描述程序的什么地方出毛病了。详细的描述每一件事情：您看到了什么，您想看到什么，把错误消息记下来，尤其是“错误消息号”。
- 当您的计算机做了什么您料想不到的事，不要动！在您平静下来之前什么都别做。不要做您认为不安全的事。
- 尽量试着自己“诊断”程序出错的原因（如果您认为自己可以的话）。即使做出了“诊断”，您仍然应该报告“症状”。
- 如果程序员需要，请准备好额外的信息。如果他们不需要，就不会问您要。他们不会故意为难自己。您手头上一定要有程序的版本号，它很可能是必需品。
- 表述清楚，确保您的意思不能被曲解。
- 总的来说，最重要的是要做到精确。程序员喜欢精确。