

09 论高质量的BUG

更新时间: 2020-04-21 09:28:22



“立志是事业的大门，工作是登堂入室的旅程。——巴斯德”

从一个“不完美”的 bug 聊起

这次的主题源于我曾经的一次**发火**，在项目总结会议上，我们在做 BUG Review 的时候，发现了类似这样一个 BUG：XXX充值接口调用报错，XXX购买接口调用报错。完了，就这么简单，我仔细观察了一下 BUG 的详情，神马都木有。并且这样的一个 BUG 还是由一个有着几年工作经验的同学提出的，顿时很想说一句：你的 BUG 提得这么棒棒的，你家里人知道吗？

“捉虫”、找 BUG 是测试人员必须具备的能力，无论在任何公司，测试人员在执行测试任务的过程中，发现 BUG、提出BUG 以及跟踪 BUG 都是十分必要且重要的工作。可以说，BUG 是我们与开发人员沟通的必备桥梁，但是呢，总是会有这样一种现象发生：我们提出的 BUG 质量不高，让开发人员无法一下子弄明白。这样就需要更多的线下沟通，经常会出现这样的对话：

三 优秀测试工程师的必备思维39讲 / 09 论高质量的BUG

- *DEV: *你那存了日志么? 还是在哪台服务器上我能看一下?
- *TESTER: *等会啊, 我回去发给你。
- *DEV: *算了算了, 你在我本地操作下我看看吧。

如此一来折腾也增加了沟通成本, 浪费了时间, 光是讲明白一个问题大概 5 到 10 min 就过去了, 更重要的是, 在非常忙碌的情况下, 大家也会感到很烦躁, 影响心情。所以提 BUG 不仅仅是我们测试人员的技能, 更应该考虑我们 BUG 的面向对象: 开发人员。开发人员需要什么信息, 我们就自然应该在提出时候包含。

编写高质量的BUG

我们先换位思考一下, 假如我们是开发, 会让我们觉得无法理解弄清楚的几种情况:

1. BUG 过于简单, 只有一句话, 描述不清

开发人员最怕的就是这种类型的 BUG, 举个例子:



这时候心里的 OS 就有了“灵魂三问”: 有什么问题? 怎么操作出来的问题? 给个问题的截图好不好?

2. 发现 BUG 的场景过程没有完全保留, 重现困难
3. 多个问题合并成一个, 开发不容易定位, 也有可能漏过问题

我们知道了自身的症状, 那么就从这里开始, 一起聊一聊一个优秀的 BUG, 应该包含哪些方面的内容呢?

• 标题

其实每一个 BUG 也都是一个小的文档, 既然是文档, 我们首先就要做好一个“标题党”, 当然, 此“标题党”非彼标题党。作为一只优秀的标题, 要清晰明确简洁地说明两个“W”:

三 优秀测试工程师的必备思维39讲 / 09 论高质量的BUG

也记住另外一个规范：每条缺陷报告只包括一个缺陷。就像刚刚说的，这样可以让开发认真的定位和对待单一的BUG，对于测试人员自己来说，也只需要每次校验一个 BUG 是否修正。像最开始我提到的让我发火的 BUG 就实不足取。

• 环境配置

如果涉及环境因素的话，比如，有些兼容性问题只在某个浏览器、某个手机系统甚至是某个型号的机器上等等，就需要指明问题复现所在的环境配置信息。

• 前置条件

有时候，有些问题是需要某些特殊条件、特殊操作或者特殊数据下才会引发的，这时候最好添加上这部分的描述，便于问题的复现。

• 重现步骤

从用户角度出发来描述重现步骤，步与步之间不应该有太大的业务跳跃。每一个步骤尽量只记录一个操作，保证快速准确的重复缺陷，确认步骤完整，准确，简短：“完整”就是说没有缺漏，“准确”是步骤正确，“简短”则是要没有多余的步骤，直指问题的核心。

• 结果

这里的结果其实包含两方面，一是“期望结果”，二是“实际结果”，之所以会产生 BUG，一定是实际结果不符合期望而导致的，所以对于结果一定要描述清楚，否则的话就会变成：

W：你错了没？

M：我哪儿错了？

W：你自己哪儿错了你都不知道？！

相信各位童鞋都深有体会（OS：榴莲键盘主板便宜出售，量大从优）。

• 优先级

凡事都有轻重缓急，所以对于 BUG 来说也是一样，表示自己对于该 BUG 的紧急程度的评估。一般情况下，我们分为下边几级：

三 优秀测试工程师的必备思维39讲 / 09 论高质量的BUG

2. **High**: 高优先级 BUG, 希望尽快解决, 一般属于比较大的核心业务功能缺陷, 但暂时不影响后续测试。
3. **Medium**: 一般优先级, 大多数情况下是非核心业务缺陷, 建议在发布前要解决的BUG。
4. **Low**: 低优先级 BUG, 如果能够在发布前解决更好, 实在工时紧张可以酌情向后推迟修改, 大多数属于优化、用户体验等建议。

• 附件

附件是非常非常重要的! 为了更加方便开发人员定位修改问题, 也是方便测试人员自己去回归测试 BUG, 适当的附件内容是很有益处的, 附件的内容可以多种多样, 比较常见的包括:

1. **截图**: 发生问题时的截图对于重现问题有着很重要的意义, 也是问题真实存在的证据。
2. **视频**: 有很多时候, 单单依靠截图也很难重现 BUG, 这时候如果在缺陷发生或者测试重现的时候能够录制一个短视频, 那么对于开发人员来说, 简直就是一场“及时雨”。
3. **错误日志**: 如果你能够找到 BUG 发生时候的具体日志, 那对于你自身的提升和开发人员定位问题的好处都是显而易见的。开发人员可以通过此找出报错的打码, 测试人员也可以学到更多关于服务器、中间件包括项目业务和部署深入的内容。

• 发生原因分析

这里就需要大家有一定的代码基础了。如果我们可以自己通过日志等内容定位发生问题的代码和大体原因, 我们对于当前系统的理解和测试就可以算是更上了一层楼。当然, 这不是必须的, 需要大家量力而为。

如果你能够完美地做到上边几点, 那么这就诞生了一个个出自于你的高质量的 BUG 了, 不仅仅有助于提高项目组中定位和修复问题的效率, 同样也锻炼体现了你自己的能力, 也更容易得到开发和领导的认可。那么问题来了, 刚刚提到了那么多方面, 如果每个都这么做, 这要耗费多少精力啊? 所以, 回到我们最初的想法, 提出缺陷的意义是更好、更有效地服务于开发和测试团队之间的沟通和未来的回顾, 大可不必拘泥于某种特定的格式, 适合你的适合当前缺陷的才是最好的, 并不是每一个项目都需要覆盖完美, 只是一旦需要, 我们希望能够看到它们。

难以重现的BUG

三 优秀测试工程师的必备思维39讲 / 09 论高质量的BUG

按照之前的步骤重新操作了一遍，咦，发现很诡异的事情发生了：刚刚的问题无法重现了！再次尝试多次，仍无法重现。于是，我们心安理得地骗自己：可能是刚才网络异常了吧。可是不知道什么时候，又偶遇了同样的问题，可是又难以重现，这时候该怎么办呢？

首先，尽量在第一时间截图留痕，如果在问题第一次偶发时候没有在意，那么一旦重现失败后续又再次出现的时候，一定要截图存档。

其次，尝试多次（我们暂定20次左右）仍然不能重现，不要继续浪费时间，尽快地去看当时发生问题时的日志，通过日志来追溯当时可能发生的情况，例如多人同步操作，数据库死锁，服务器断线等情况。

接下来，通过当时的数据、环境、配置、特殊操作等再尽量多尝试几次，同时，可以与组内其他测试人员交叉测试，不管是否可以发现，都要把问题提出，并且通知开发，标识该问题为难以重现的 BUG，请开发人员与自己一同通过白盒或代码走查的方式尝试定位。

最后，如果仍难以发现并修改，那么就需要及时评估该问题的影响范围，影响较小的留存后续跟踪，影响较大的则上报项目经理协调解决。

提出 BUG 的四重境界

就如同我们打游戏一样，同样一个英雄，在不同的玩家手里可以用出不一样的效果，测试亦然。同样是发现 BUG，不同的 tester 同样会有不同的境界。

第一重：筑基。

顾名思义，这一重境界还是铸造基础阶段，我们能够找出问题，但是可能描述不太清楚。如：

在进行添加购物车、结算操作时，报错。上述的 BUG 描述仅仅能够做到发现问题，至于是什么操作，什么报错，并不能够指引开发人员去具体复现和定位。所以如果要打分的话，这样的 BUG 我可以给打 30 分。

第二重：灵寂。

在武侠中，这是能力大提升后的平稳阶段，也是步入更高殿堂的前阶段。在这重境界中，我们开始不但能够找到问题，也能够描述清楚问题。

三 优秀测试工程师的必备思维39讲 / 09 论高质量的BUG

这样的描述加上附件已经基本能够达到让开发人员看懂并着手解决了，在没有代码能力的前提下，可以算是将功能测试做到了不错的境界。所以，这样的描述可以打到 60 分。

第三重：元婴。

这是真正步入修真殿堂的一步，对于测试来说，也是更进一步的体现。在这重境界中，我们开始去阅读、去审查代码，去尝试找出代码存在的问题。

添加商品进入购物车，点击结算未发生跳转，错误信息页面提示：“error:null”。截图及日志见附件。根据日志排查，在XX类中XX行位置，报空指针异常，可能由于前边对XX参数未赋值。

这样我们做到的不仅仅是发现问题，描述问题，同时定位错误发生的原因。这样我们可以达到 80 分。

第四重：大乘。

巩固修为的果实，慢慢累积力量，直到圆满，也就意味着我们测试能力的大成，对我们的要求也就更高了，基本可以做到：**教开发人员写代码**。这也是我觉得测试的最高境界了。

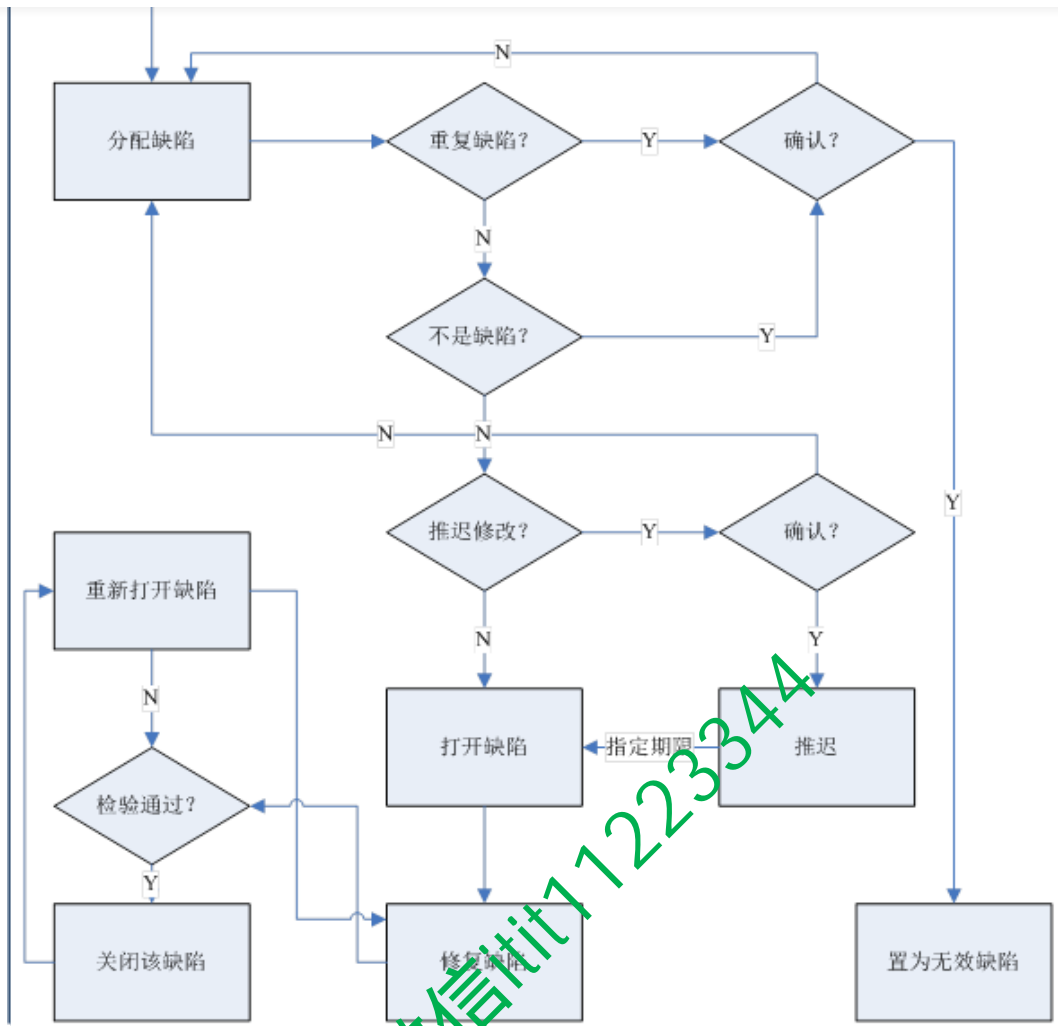
添加商品进入购物车，点击结算未发生跳转，错误信息页面提示：“error:null”。截图及日志见附件。根据日志排查，在XX类中XX行位置，报空指针异常，分析由于前边对a参数未赋值。建议修改：从前边调用用户信息查询接口的返回对象中，将其中的 b 参数赋值到 a 参数中，再调用结算接口。

在这重境界里，对我们的要求是：找到问题、描述清楚、定位问题且提供解决思路。能做到这样的话，我们可以打个 99 分了。教开发写代码，是我们做功能测试的巅峰，也是大家不断提升自己能力的体现。

BUG的后续跟踪

BUG 提交完不是结束，只是一个流程的开始。你仍然需要实时跟踪 BUG 的进展，甚至对于一些优先等级高的 BUG，还需要去关注修改的进度。通常情况下一个 BUG 的生命周期是：

优秀测试工程师的必备思维39讲 / 09 论高质量的BUG



所以，按照上边来说，我们的状态变化就会是：New -> Open -> Fixed -> Close。

当然，除此之外，任何一个大型项目结束后，还需要对 BUG 进行分析和汇总，分析问题出现的原因及未来避免方案。总结开来，BUG 是我们测试生涯中不可缺少的一部分，同时也不是非常简单的工作，我们需要不断磨砺自己的技术能力，提出高质量的 BUG，也让我们未来的测试道路越走越好，越走越宽。

← 08 厉兵秣马：如何设计优秀的测试用例

10 从青铜到黄金：功能测试进阶路上还需要哪些非测试技能 →

精选留言 6

欢迎在这里发表留言，作者筛选后可公开显示

三 优秀测试工程师的必备思维39讲 / 09 论高质量的BUG

上条留言说错了：清款→情况

👍 0 回复

2020-04-16

chenchen2008 回复 **慕瓜1314147**

谢谢同学反馈，已经对文章做更正。

回复

2020-04-21 09:29:39

慕瓜1314147

老师，文章中有个错字需修改：bug流程图上方：清款→请看

👍 0 回复

2020-04-16

农民端的二狗

1其实真的多去阅读日志，发现问题后通过日志去代码中定位，自己对系统的理解确实变得更深，而且开发会很认可你。2很多时候，需求了解清楚后，走读代码，比对修改，参考需求提出问题，第一更清楚了修改用意，第二讨论过程中双方能一起发现很多问题。

👍 1 回复

2020-02-23

仲夏rww

我也还在第二重啊，已经尽自己最大努力把Bug描述清楚了。想问下老师，怎么才能看懂Bug的日志？

👍 3 回复

2019-09-12

风落几番 回复 **仲夏rww**

你的问题跟前边有一丢丢相像，其实不是看懂日志的问题，而是你没有习惯尝试去看。如果是一些后端异常的话，日志一定会打印出来错误的信息和对应错误的类、报错的行数。这样结合着代码就很容易能够分析出来。如果开发人员日志打的比较规范的话，单从日志里的错误信息就可以发现端倪。如果日志太多了，那么要是可以重现，就重现一下观察增量日志；如果不能重现，那么尽量去找一些日志中会打印的关键信息来筛选~

回复

2019-09-16 10:58:53

慕侠5476819

三 优秀测试工程师的必备思维39讲 / 09 论高质量的BUG

风落几番 回复 慕侠5476819

这个回复没有办法发图，蛋蛋的忧伤。。这么说吧，你这个问题呢，不是日志例子的问题，而是你要开始试着看日志，学会看日志。大多数时候我们都是用log4j或者相关组件来打的。那么一旦出现错误，日志中会打印出错误的信息msg，错误的代码行数。这样结合着你去查看代码、对于代码的理解能够更直接方便的定位问题，不要总觉得这是开发应该做的。

回复

2019-09-16 10:55:48

慕丝8558034

提了那么久的bug，原来我还在第二重啊，冲鸭！！！

👍 4 回复

2019-09-11

干学不如一看，干看不如一练

一手微信itit1223344