**bug的定义**

软件的Bug，狭义概念是指软件程序的漏洞或缺陷，广义概念除此之外还包括测试工程师或用户所发现和提出的软件可改进的细节、或与需求文档存在差异的功能实现等。

我们的职责就是，发现这些Bug，并提交给开发，让开发去修改

**bug的类型**

要确定一-个bug的类型， 需要对项目(或产品)有比较深的理解。这个划分对于开发定位问题影响很

小，但对于问题类型的统计就比较重要了。

常见的bug类型划分(禅道系统为例，可自定义) :

代码错误

界面优化

设计缺陷

**bug的等级**

bug等级的划分有分三级或四级，也有分五级的。如果是等级越高，那么可能被修复的等级也

会高一些，然后有些公司还会根据你提的bug数量和bug等级来考察你的绩效。很多情况下，我们提交

bug大致的等级差不多即可，没有严格区分。

如何来判断bug的等级(严重程度)，一般可以参照下面的判断条件。

(1)致命错误:

1、常规操作引起的系统崩溃、死机、死循环、闪退

必现! -就绝对1级、偶现不一-定是1级

2、造成数据泄漏的安全性问题，比如恶意攻击造成的账户私密信息泄露(暂时发现不到)

3、金额计算出错

根据影响范围/损失大小判定，不应涉及金额计算就提交1级

4、阻断性bug.所有1大部分测试工作进行不下去

多参考公司之前提的bug/或者自己公司有bug等级划分;1级类的bug慎重提

(2)严重错误:

1、重要功能不能实现;

2、错误的波及面广，影响到其它重要功能正常实现;

3、非常规操作导致的程序崩溃、死机、死循环、闪退

4、外观(界面)难以接受的缺陷

5、密码明文显示，(界面+数据库)

6、偶现的致命性bug .

(3)一般错误:

不影响产品的运行、不会成为故障起因，但对产品外观和下道工序影响较大的缺陷

1、次要功能不能正常实现;

2、操作界面错误(包括数据窗口内列名定义、含义不一-致) ;

3、简单的输入限制未放在前端进行控制;-接口测试深入

4、删除操作未给出提示

5、偶现的严重性bug

(4)细微错误:

程序在一些显示上不美观，不符合用户习惯，或者是一些文字的错误

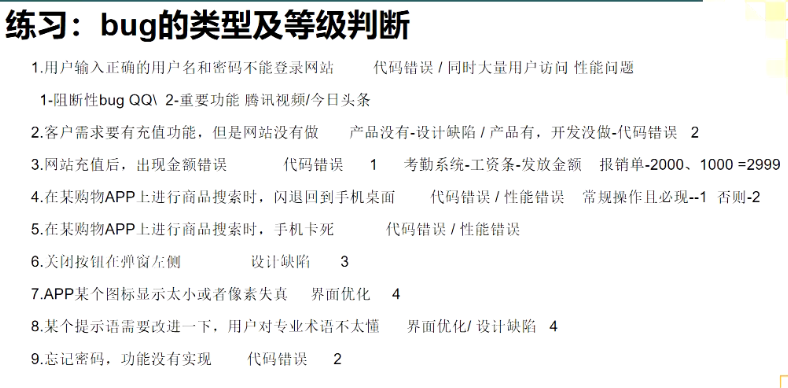
1、界面不规范;

2、辅助说明描述不清楚;

3、提示窗口文字未采用行业术语;

4、界面存在文字错误;

改进建议:可以提高产品质量的建议，包括新需求和对需求的改进。



**bug的生命周期(管理流程)**

面试/笔试过程中经常会被问道的问题。

bug的生命周期，就是一个bug被发现到这个bug被关闭的过程(状态)。

步骤：

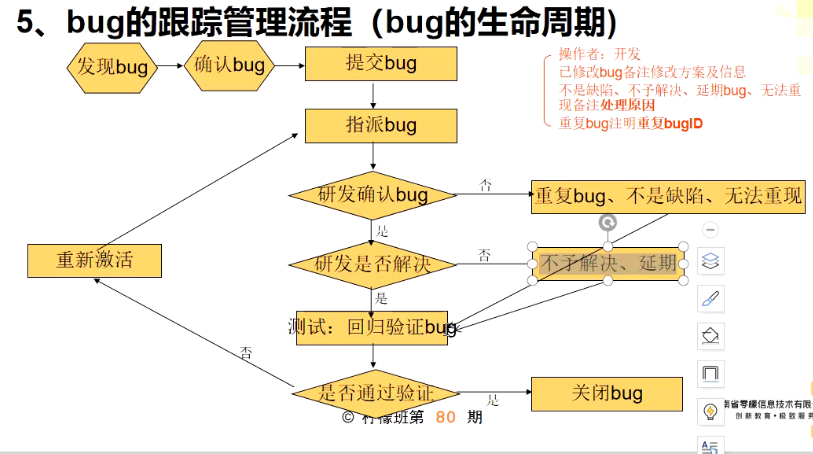
生命周期中一般缺陷状态:激活(提bug,指派) ->已解决(bug已修复、不子解决、延期、不是缺陷

、重复bug)->(验证通过)--> 关闭。

如果待验的bug在验证时没有解决好，我们需要重新打开(激活) ->指派->已解决>待验，循环这个

过程。

中间其他状态:拒绝、延期等



**bug的跟踪管理-状态处理**

1.**已经指派的bug**---已经指派给开发的，需要自己关注bug的走向，随时关注并进行跟踪，如果一

直未修复，提醒开发修改，以免开发忘记；如果已经修复等待测试环境更新后进行验证。催着改bug

2.**已解决的bug**----等待测试环境更新后进行验证，验证通过则关闭;验证不通过则重新打开指派给开

发。

3.**重复bug**-----先查看是否跟开发指定的bug重复。如果确定是重复则关闭，如果不重复，说明

原因，重新激活打开指派给开发。

4.**不是缺陷**-----再次依据需求确认，是否是bug，如果依然觉得是缺陷跟开发沟通，列举出来觉得是

bug的点，沟通未达一致找产品确认，是bug注明情况并再次指派给开发，产品确认不是bug,就直接关闭bug。

5.**无法重现**-----确认开发环境是否跟测试环境一致? 包括操作步骤、浏览器、环境、特定账号、输入数据等，尽量找到重现步骤,如果多个测试版本验证之后，如开发所说重现不了，依据bug的严重程度跟产品、开发一起确认是否关闭; bug高,开发测试尝试解决,但是会拿小本本把这个bug记录下来,等

到测试任务结束后，再来研究研究。如果找到重现原因,注明清楚并再次指派给开发

6.**不予解决**----先沟通开发，能否说服他，说服不了找产品经理进行确认。确认不予解决进行关闭;确

认需要解决请备注原因并打开指派给开发

7.**设计如此**----先确认是否设计如此，若不是沟通开发,若不行，找产品经理进行确认。确认设计如此

进行关闭;确认是问题，备注原因重新指派给开发

8.**延期修改**---请看下bug严重程度,是否影响当前版本发布?与产品经理进行确认。不予延期请根据情

况进行激活与情况说明;确认延期则做好记录，后续版本进行关注一不关闭

7、bug的跟踪管理缺陷管理工具

**常见的缺陷管理平台**

禅道(zentao)

bugzilla、jira:都还不错， 也比较强大。但是搭建起来很困难

bugfree

Readmine

easybug:免费开源，在线网站类型的

Mantis:这个还可以用

QC(QualityCenter)、TD

不管是开源还是商业的缺陷管理工具，它们本质都是一-样的, 用来管理bug的生命周期。掌握其中一款

工具，自然就会用其他的，稍微有一点点区别的，别人加以指点，就可以明白了。

**bug的跟踪管理-如何提交bug**

提交BUG的包含内容

**bug标题**一一标题要清晰简洁， 写明bug描述;如果没有选择功能模块，最好在标题中标注功能模块。

让查看bug的人员清楚知道你所表达的意思。模板：bug的功能模块 + bug的操作+ bug的结果

**重现步骤**一一详细写下发现bug的测试过程。能指导开发重现这个bug。附上测试数据

步骤一写明步骤，附上登录账号及测试数据信息

结果一出现bug的结果，粘贴**bug截图**、 日志截图

期望一写清楚预期结果

bug类型和严重程度一一便于后续测试结果分析， bug的统计

bug测试环境一一例如:什么系统，哪个版本等。兼容性问题、难以重现问题

附件一一日志文件， 文件类型测试数据。出现bug视频。图片、崩溃日志文件等

所有以上，参考公司前辈写的bug,依葫芦画瓢，拓展测试思维

