

11.4号周三

## 笔记: bug的管理流程&禅道使用

### 1、bug的定义

**狭义:** 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。

### 2、bug的类型

代码(功能) 错误====功能没有实现

界面优化====页面显示有问题

设计缺陷====不符合需求

### 3、bug的等级

数字越小, 严重级别越高。1,2,3,4,

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
- 2、造成数据泄露的安全性问题, 比如恶意攻击造成的账户私密信息泄露
- 3、涉及金钱计算
- 4、阻断性测试, 所有测试工作进行不下去(冒烟测试)

#### (2) 严重错误

- 1、重要功能不能实现: =====分析需求
- 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、查询错误, 数据错误显示
- 4、简单的输入限制未放在**前端**进行控制
- 5、删除操作未给出提示
- 6、偶现的严重性bug

前端: 岗位名, 前端开发工程师, 负责貌美如花

后端: 岗位名, 软件开发/研发工程师, 负责赚钱养家(实现功能的)

#### (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

改进错误：可以提高产品质量的建议，包括新需求和对需求的改进。

#### 4、bug的生命周期（管理流程）

新建（提bug）-指派-已解决-待验-关闭。

#### 5、bug的跟踪管理流程

发现bug--确认bug--提交bug--指派bug--研发bug（否）重复bug、不是缺陷、无法重现--研发是否解决（否）不予解决、延期----回归验证bug---是否通过验证（是）关闭bug（否）重新激活--指派bug。

#### 6、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，请大家注意自己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----先去看下是否跟开发指定的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，沟通未达成一致找产品确认，确认是bug注明情况并再次指派给开发，产品确认不是bug、就不纠结，直接关闭bug，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。

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，如果多个版本验证之后，如开发所说重现不了，依据bug的严重程度跟产品、开发一起确认关闭；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---找产品经理进行确认。确认设计如此进行关闭；确认需要解决请备注原因并打开指派给开发

7、设计如此--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-----**不关闭**