

第三章 节 BUG篇

本节课重点内容

- 软件测试生命周期
- 什么是bug
- 描述bug的要素
- bug级别
- bug的生命周期
- 与开发产生争执怎么办

1. 软件测试的生命周期

软件测试贯穿于软件的整个生命周期，针对这句话我们一起来看一下软件测试是如何贯穿软件的整个生命周期。

软件测试的生命周期是指测试流程，这个流程是按照一定顺序执行的一系列特定的步骤，去保证产品质量符合需求。在软件测试生命周期流程中，每个活动都按照计划的系统的执行。每个阶段有不同的目标和交付产物



各阶段具体内容：

| 需求分析 | 测试计划 | 测试设计与开发 | 测试执行 | 测试评估 | 上线 | 运行维护 |
|--|--------------------------|--|--------------------------|-----------------------------|--|---|
| 用户角度：软件需求是否合理 技术角度：技术上是否可行，是否还有优化空间 测试角度：是否存在业务逻辑错误、冗余、冲突等问题 | 制定测试计划：什么时候开发测试，什么时候结束测试 | 参考需求文档、技术文档等编写测试用例 写测试文档，明确标注使用到的测试方法，测试工具， | 充分利用测试用例和测试工具对项目尽可能做到全方面 | 测试是否通过，本次测试是否有遗留的BUG，最终测试人员 | 项目测试结束后，将项目发布到线上环境，测试人员需求跟踪上线并测试线上环境下软件的 | 测试人员需要参与项目的实施工作。测试人员对项目产品的业务和操作非常了解，加上测试人员的沟通表达能力一般都比较强，所以测试人员可以参与用户使用软件的培训，在试运行项目时收集 |

| | | | | | | |
|--|--------|--------|-------|-------------|--------|---------------|
| | 试，耗时多久 | 测试形式等等 | 的测试覆盖 | 需要产生出一个测试报告 | 运行是否正确 | 问题并及时反馈给相关负责人 |
|--|--------|--------|-------|-------------|--------|---------------|

2. BUG

2.1 bug的概念

定义：一个计算机bug指在计算机程序中存在的一个错误(error)、缺陷(flaw)、疏忽(mistake)或者故障(fault)，这些bug使程序无法正确的运行。Bug产生于程序的源代码或者程序设计阶段的疏忽或者错误。

准确的来说：

1. 当且仅当规格说明是存在的并且正确，程序与规格说明之间的不匹配才是错误。

2. 当需求规格说明书没有提到的功能，判断标准以最终用户为准：当程序没有实现其最终用户合理预期的功能要求时，就是软件错误。

2.2 描述bug的要素

为什么描述bug还有要素要求？

在心理学上说，人们在编写文档的时候，经常会出现自己想表达的和写出来的内容往往南辕北辙

bug描述：浏览器打开链接失败

该描述下，没有明确说明哪个浏览器，失败的具体表现是什么，对于开发人员来说无法捕捉到更多有效的信息，会造成沟通效率低下，工作质量低下等问题。

描述bug的基本要素：问题出现的版本、问题出现的环境、问题出现的步骤、预期结果、实际结果

案例：<https://www.101eduyun.com/>



问题出现的版本：谷歌浏览器版本 123.0.6312.123（正式版本）（64 位）

问题出现的环境：Windows家庭版

问题出现的步骤：

- 1、打开谷歌浏览器，输入网址<https://www.101eduyun.com/>
- 2、等待首页页面渲染完成

预期结果：二维码与登陆模块不会出现遮挡，二维码可以正常扫描

实际结果：二维码被登陆模块遮挡，二维码扫描失败

2.3 bug级别

通过定义bug的级别，能够明确看出问题的严重程度。工作中开发人员通常需要按照bug的级别来分配优先级来处理bug，除此之外，通过bug级别也能够体现出开发人员的开发质量。

案例：

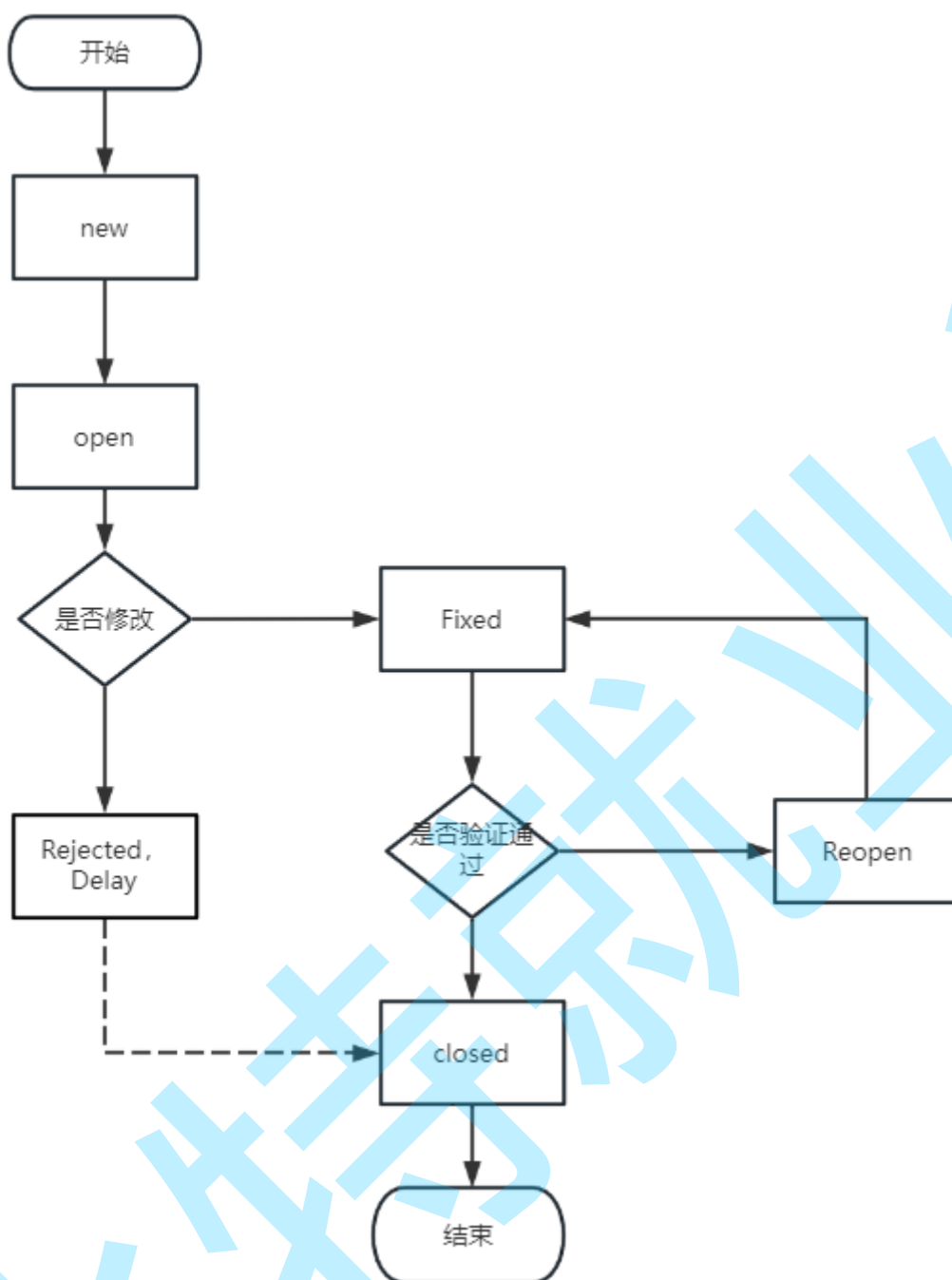
- 男朋友多看了几眼美女：次要
- 男朋友跟美女加微信聊天：一般
- 男朋友跟美女私下去吃饭：严重！！
- 男朋友跟美女去做头发：崩溃！坚决踹了！！！！

bug级别一般分为：崩溃、严重、一般、次要

| 崩溃 | 严重 | 一般 | 次要 |
|--|--|---|---|
| 阻碍开发或测试工作的问题；造成系统崩溃、死机、死循环，导致数据库数据丢失，与数据库连接错误，主要功能丧失，基本模块缺失等问题。如：代码错误、死循环、数据库发生死锁、重要的一级菜单功能不能使用等（该问题在测试中较少出现，一旦出现应立即中止当前版本测试）。 | 系统主要功能部分丧失、数据库保存调用错误、用户数据丢失，一级功能菜单不能使用但是不影响其他功能的测试。功能设计与需求严重不符，模块无法启动或调用，程序重启、自动退出，关联程序间调用冲突，安全问题、稳定性等。如：软件中数据保存后数据库中显示错误，用户所要求的功能缺失，程序接口错误，数值计算统计错误等（该等级问题出现在不影响其他功能测试的情况下可以继续该版本测试）。 | 功能没有完全实现但是不影响使用，功能菜单存在缺陷但不会影响系统稳定性。如：操作时间长、查询时间长、格式错误、边界条件错误，删除没有确认框、数据库表中字段过多等（该问题实际测试中存在最多） | 界面、性能缺陷，建议类问题，不影响操作功能的执行，可以优化性能的方案等。如：错别字、界面格式不规范，页面显示重叠、不该显示的要隐藏，描述不清楚，提示语丢失，文字排列不整齐，光标位置不正确，用户体验感受不好，可以优化性能的方案等（此类问题在测试初期较多，优先程度较低；在测试后期出现较少，应及时处理） |

2.4 bug的生命周期

测试人员在执行测试的过程中如有发现bug，需要在对应的bug管理平台来创建bug（bug生命起源），创建好的bug需要被开发人员修复，以及测试人员的持续跟踪和测试。



- New:新发现的Bug，未经评审决定是否指派给开发人员进行修改。
- Open：确认是Bug，并且认为需要进行修改，指派给相应的开发人员。
- Fixed:开发人员进行修改后标识成修改状态，有待测试人员的回归测试验证。
- Rejected：如果认为不是Bug，则拒绝修改。
- Delay:如果认为暂时不需要修改或暂时不能修改，则延后修改。
- Closed：修改状态的Bug经测试人员的回归测试验证通过，则关闭Bug。
- Reopen：如果经验证Bug仍然存在，则需要重新打开Bug，开发人员重新修改。

无效的bug：open->closed open-rejected-closed

2.5 与开发产生争执怎么办（高频考题）

在测试工作中，最常遇到的是和开发人员的PK，作为测试经理还会和项目经理、产品经理的PK进度、质量。作为一名测试人员，一般会遇到以下几种情况：



这不是bug
这个bug的级别太高了
bug影响不大，暂不修改

遇到争执不要怕，要理性的分析和反馈问题。

2.5.1 先检查自身，是否bug描述不清楚

如果能正确地、高质量地录入一个Bug，那么基本上已经成功地与开发人员沟通了一大半的关于Bug的信息。但是总有“书难达意”的时候，这时就需要测试人员主动与开发人员进行沟通了。如果测试人员发

现在写完一个缺陷后，好像还有很多关于Bug的信息没有表达出来，或者很难用书面语言表达出来时，

就应该在提交Bug后，马上找相关的程序员解释刚才录入的Bug，确保程序员明白Bug描述的意思，而

不要等待开发人员找自己了解更多的信息。

2.5.2 站在用户角度考虑并抛出问题

站在用户角度考虑问题 应该让开发人员了解到Bug对用户可能造成的困扰，这样才能促使开发人员更加积极地、高质量地修改Bug。在争执时，可以问一句：如果你是用户，你可以接受么？

2.5.3 BUG定级要有理有据

BUG定级时，不仅要参考BUG级别，还要考虑BUG是否会影响到流程，往往用户的BUG级别和我们的是有区别的，需站在用户的角度定考虑定位级别。

2.5.4 提高自身技术和业务水平，做到不仅能提出问题，最好也能给出解决方案

提高自身的业务和技术水平，不但要做到能提出问题，还能够提出解决问题的思路。这样才能更让人信服。

在工作中，你会发现同一个bug，资深测试工程师提出和初级测试工程师提出，两者的结果完全不同，

两者最大的差别是资深测试工程师往往会提出解决方案。而长此以往，权威性逐渐的建立起来，那么开

发人员看到bug的第一反应，就是这是一个bug，而不是这是一个bug吗？

注意：可以给出解决方案，但是不能喧宾夺主，命令式让开发人员按照自己的想法来修改。

2.5.5 bug评审

如果确实是bug，友好沟通不能解决问题，那么就召开bug评审。

bug评审主要解决两个问题：

- 1) 决定如何处理bug
- 2) 分析缺陷产生的原因，找出预防的对策

bug评审至少需要项目组各个方面的代表参加：

1) 测试代表

测试代表主要从Bug的具体表现、严重程度等方面提供信息，并提出自己对Bug的处理意见。需要注意的是，测试人员不应该一味地要求对Bug进行修改，因为修改可能带来回归的风险，同时带来的是回归测试的工作量，如果时间比较紧迫，修改后剩余的时间若不足以做一次有效的回归测试，可能不修改是个明智的选择。

2) 开发代表

开发代表主要从修改缺陷的难度和风险出发，考虑缺陷修改需要付出的代价，以及可能影响的范围、可能引发的风险等，如果决定要修改，还要讨论出修改的初步方案。

3) 产品代表

产品代表主要从产品的整体计划、用户的要求等方面对缺陷的修改必要性、缺陷修改的时间和版本提出自己的意见