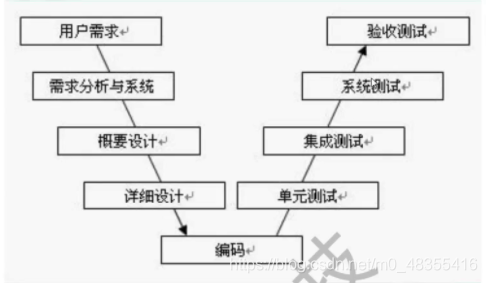
软件测试V模型



优点：左边开发的每一个阶段和右边测试的每一个阶段一一对应、是右边测试每一个阶段的主要依据

缺点：测试介入晚，前期的错误和风险到后期才发现，会失去及时纠正的机会

软件测试W模型



优点：测试阶段和开发阶段在两个独立的V模型里面，测试介入包比较早，在项目初期就介入，前期的风险可以及时被发现

缺点：W模型每一个阶段仍然是一个串行的过程，不能适应需求变化的项目，所以无法应用到敏捷开发

如何描述一个BUG

1.发现问题的版本

2.问题出现的环境

    不同的浏览器对同一个系统的同一个页面的解析不一定相同（eg：Windows、Firefox、Chorme、edge、360、搜狗、QQ、Mac、以及浏览器对应版本号，同时还有手机APP，IOS和Android不同的机型）

3.测试步骤和测试数据

    尤其是出现错误重现的步骤等

4.预期行为的描述

    尤其要以用户的角度来描述，更有助于后续更改跟进

5.错误行为具体描述

    描述错误的具体表现，错误行为描述、错误截图、错误日志等（crash等上传log，UI问题可以上传相关截图）

6.附件

    这里主要指一些特殊要求，比如故障的分类等

举例：



BUG级别的定义

1.Blocker（崩溃）

阻碍开发或测试工作的问题，造成系统崩坏、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。

线上出现崩溃级别的BUG怎么办？

回退到一个稳定的版本

2.Critical（严重）

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。（eg:直播画面失真、密码明文显示等）

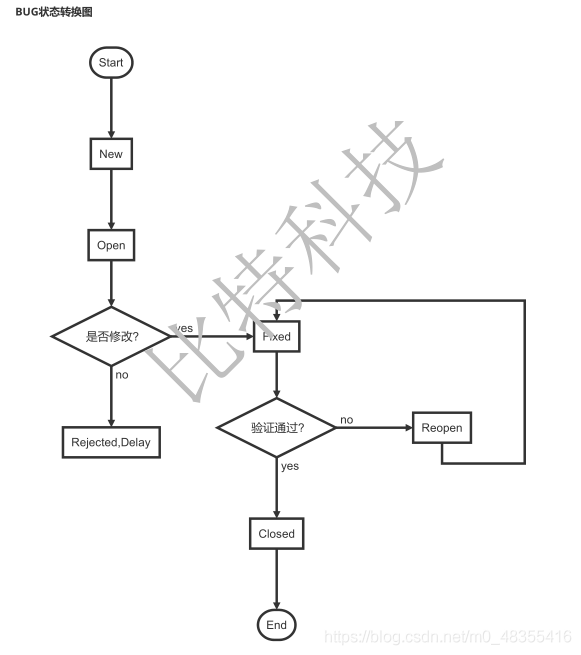
3.Major（一般）

功能没有完全实现但不影响使用，功能菜单存在缺陷但不会影响系统稳定性。

4.Minor（次要）

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。

BUG的生命周期



● 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斌验证通过，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

Bug的跟踪以及状态变更应该遵循一些基本原则：

测试人员对每一个缺陷的修改必须重新取一个包含更改后的代码的新版本进行回归测试，确保相同的问题不再出现，才能关闭缺陷。

对于拒绝修改和延迟修改的Bug，需要经过包含测试人员代表和开发人员代表、用户方面的代表（或代表用户角度的人）的评审。

如果遇到一个BUG，和开发人员发生冲突怎么办？

（1）先检查自己对发现BUG的描述是否清晰

（2）检查BUG的定级是否按照公司标准进行描述

（3）站在用户的角度去说服开发

（4）不断提升自己的业务水平及技术水平

（5）当发生争吵时，尽力避免争吵。与开发人员及产品经理开会商讨解决方案

————————————————

版权声明：本文为CSDN博主「小小小青台」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/m0\_48355416/article/details/116067888