**前言**：

生产无小事，生产环境的任何问题无论大小都要当成故障对待，要想保证版本质量，必须提升全体员工的这种意识，让每个人都把生产问题当成很重要的事去对待。保证生产环境版本质量会涉及方方面面的因素，比如代码质量、测试保障、上线前准备工作等等，每个环节都有很多学问需要不断的总结提炼才能逐步提升版本质量。

做项目从代码开发、测试、联调、上线，各个环节都需要严格的管理流程来管控，只靠员工的经验、自觉性是远远不够的或者说不可靠，不管控好每个环节，版本质量出问题是迟早的事，轻者客户投诉，重则会客户扣款影响合作，所以必须重视和加强。

先说一下做项目如何避免一些常见的低级错误，所谓低级错误就是不用动脑就能做好的事却除了问题，这种错误一旦出现，无法跟客户解释，客户基本也都是零容忍。

**常见问题1：版本管理流程不健全。**

项目管理过程中版本管理至关重要，一般项目都是迭代开发上线，就是上线之后定期或者不定期的有新需求或优化要上线，如果没有版本管理或者管理不完善版本就会很乱，出问题是必然的。

**措施：**

正常情况下的代码起码要拆分出3个分支，开发分支、测试分支、生产分支，开发人员只在开发分支开发代码，开发完成之后同步到测试分支，发布测试环境给测试人员测试使用，这样可以避免开发人员继续开发新需求和修改BUG影响测试人员版本稳定性，测试通过之后代码提交到生产分支（提交后在版本上载前要全量测试），生产分支的代码是用来上线用的，任何人不能随意改动和提交，这里必须做好管控（通过svn或者get权限控制，通过人工管理不靠谱），代码提交之前组长要做代码检查，避免出现错误。

**常见问题2：版本发布缺少打包机制**

版本每次发布都需要打包，打包之前要修改一些文件，多数都是配置文件，比如测试环境的IP地址改成生产环境IP等，有时候可能改的还不止一个文件，只要有一点点的疏漏，版本必然出现低级问题。

**措施：**

能用脚本做的尽量不要人工去改，人工改可能会改错、改漏，而这种改动在上版本之前又无法检查，问题不容易被发现，当上了生产出了问题就是低级问题。修改文件、替换文件等都写成脚本来做，经过严格测试保证脚本没问题，以后打包都用程序做，这样打包出问题的几率几乎为零。

**常见问题3：版本发布前后台程序不一致**

项目基本都有前后台，如果前后台程序没有管理好，前台后台版本不一致是必然要出错的，后台增加了对新增表字段的处理，但是发布过去的前台是旧的，不出问题是不可能的，前端调用后端程序必然报错。

**措施：**

这个问题要从两方面来保障

1，做好版本管理，调研完需求生成任务单分派给前后台的开发人员开发，开发完之后提交测试，测试人员测试的时候开发人员会继续开发新的需求，此时新需求不准提交到测试版本，等上一个版本的需求测试没问题同步到生产环境之后才允许开发人员提交下一个版本的需求代码，这样就不会乱，否则谁想提交代码就提交，完全没有管控，整个代码里面哪个需求提交了多少代码，前后台是否同步提交了也没人说得清，不出问题才怪。

2，运维人员发布版本做好目录规划，每个版本前后台都清晰的规划好目录，避免搞乱，如果版本管控好了，运维人员发布的时候发错了就是典型的低级问题。

**常见问题4：上一个版本修复的BUG这个版本又出现了**

这个问题是版本拆了分支没有管控好导致的，因为每次发布的都是生产分支的代码上线，当版本上线后用户使用过程中出现了问题，客户会提给开发处理，开发人员为了紧急处理问题，找到问题后直接在生产代码修改了，测试人员测试没问题就上生产了。问题已经修复，但是没有在开发分支同步代码，当下个版本上线时直接从开发分支拿代码放到测试分支再同步生产分支，导致把之前生产分支修复的BUG给覆盖了，这个问题大家一听问题很明显啊，但是这种问题会经常出现，这个也是非常经典的低级问题华为做移动这么多年，这种问题依然还经常出现。

**措施：**

版本上载后客户使用过程中提的问题，一定要严格按流程来走，先在开发分支改，同步到测试分支测试，没问题再同步到生产分支打包发布，切忌直接在生产分支修改BUG，这必须明令禁止。

**常见问题5：版本又出现了上次出现的类似的问题**

版本上载经常遇到这次又出现跟上次版本类似的问题，这种也是客户不能容忍的，会觉得一种问题反复出现，比如程序用到的数据库参数忘记提交了等等这种也是典型的低级问题。

**措施：**

随着版本上载的增加，每次遇到问题都记录到一个checklist表格中，每次版本上载之前都从头到尾核对一次，避免以前忘记做的操作，这次又忘记了，这份checklist不断完善，记录越来越多，问题就会越来越少。

公司gitlab

<http://116.62.118.242>

http://192.168.0.208