

# 版本发布流程制定

## 规定1

所有人员建立自己的分支。可以是缩写。



### 目的

处理近期需要上传到test的修改。

## 规定2

**建立test和master版本发布分支**

### 目的

test分支只用来提供给测试部门测试使用。

master是正式环境分支。只能用test合并到master。

所有不准备上线的代码不允许合并到test。

出现误操作请联系运维协助处理。

## 规定3

**建立temp分支**

### 目的

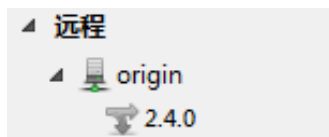
应对市场部门需要长期测试，却又不准备上线的项目。

temp分支拥有独立的域名和数据库。

当temp分支有部分代码要上线，有部分代码不上线的情况发生时。由负责写代码的程序员挑选代码上线。

## 规定4

**建立长期分支**



## 目的

应对大版本的升级。例如官网2.0后台。第四期需求。

大版本升级较不平凡。当各部门认为大版本已经可以正式发布时。各部门需约定一个时间来做大版本的升级。

升级时将大版本整个合并到test分支做测试。

并由大家一起解决合并时遇到的冲突问题。

升级后请把自己修改了一半没上传到test的代码自行保存。升级后统一拉取新版本，在新版本的基础上再做修改。

## 规定5

**对于长期分支的修改，只允许将分支检出后修改,并做版本控制。长期分支的测试只在自己本机做测试**

### 目的

为了避免出现一些人将长期分支和自己分支混起来，然后直接合并到test做测试。这会让一些后面才上线的内容直接上线

## 规定6

**所有人按照规范编写代码。写好备注。**

### 目的

由于大版本合并的时会出现很多冲突。

如果程序员自己都忘记掉自己写的是什么，解决起来会很麻烦。

## 规定7

**对于bug的修改。可以直接上传test,再合并到master**

### 目的

当测试人员发现bug 且该bug已经上线时。为了快速处理bug 。可以直接修改test分支。而不用在自己的分支上修改然后合并到test。因为这时候你的代码可能只写了一半，不能提交。

但是请一定在改bug时写好这是改BUG的注释（git 和代码里都要写）。避免其他人合并代码时出现冲突。又把你改的bug还原。也避免大版本升级合并时。出现冲突不知道该用test分支还是用大版本分支。

## 规定8

**面对市场紧急上线或者其它上线需求，请不要自行做决定破坏上面的规则。应与运维部门联系协助处理**

### 目的

为了保证以上规定有序执行。不出现大的影响正式环境的代码上传到线上。不要觉得自己可以解决就自行解决想着解决后还原，因为总有忘记的情况。