## 研发团队Git使用流程

### 分支流程说明

在项目中长期存在以下两个分支：

* master: 主分支，记录上线版本的迭代，该分支代码与线上代码保持完全一致。
* develop: 开发分支，该版本记录开发过程的迭代，与开发迭代保持一致。

在项目中可能会存在以下短期分支：

* feature-\*: 功能分支，用于开发新的功能，不同的功能创建不同的功能分支，功能分支开发完成并且自测通过后，合并到develop分支，并在git库和本地库删除该分支。
* bugfix-\*:bug修复分支，用于修复当前开发中不紧急的bug,自测bug及普通测试bug均需要创建bugfix分支进行开发，开发完成并且自测通过后合并到develop分支，并在git库和本地库删除该分支。
* release-\*:发布分支，用于代码的上线准备，该分支从develop分支创建，创建后部署到产品测试服务器进行测试，测试中的bug需要开发人员在该release分支上进行修复。待所有bug修复完成后，需要合并该release分支到master分支和develop分支。
* hotfix-\*:紧急bug修复分支，该分支只有在线上出现致命bug需要修复时使用，从master分支创建，待修复完成后需要合并release分支用于测试人员的快速测试，测试通过后合并该分支到master分支进行上线，同时合并到develop分支。

### 分支命名规范：

##### 功能分支

功能分支的分支名称应该为能够准确描述该功能的英文简要描述：

feature-functionDescription

例如开发的功能为*添加用户*，则可以创建分支名称为*feature-userAdd*。

##### bug修复分支，紧急bug修复分支

bug修复分支的分支名称为禅道中bug的编号或该bug的英文简要简称：

bugfix-#02 或bugfix-bugDescription

hotfix-#02 或bugfix-bugDescription

比如，需修复的bug（用户不能登录）在禅道中的编号为13，则可以创建一个分支名为bugfix-#13或bugfix-userCannotLogin的分支。

##### release分支

release分支作为作为预发布的分支交付给测试人员进行测试，命名分支名称为当前版本：

release-v1.1.0

### 常用指令：

（请自行熟悉以下指令）

git checkout

git pull

git add

git commit

git push

git branch [-d]

git merge

git cherry-pick

git stash

### 注意事项：

* 所有的新功能开发，bug修复（非紧急）都要从develop分支拉取新的分支进行开发，开发完成自测没有问题再合并到develop分支
* release分支发布到测试环境，由开发人员创建release分支（需要测试人员提出需求）并发布到测试环境，如果测试过程中发现bug，需要开发人员track到该release分支修复bug，上线前需要测试人员提交merge request到master分支，准备上线，同时需要合并回develop分支。
* 只有紧急情况下才允许从master上拉取hotfix分支，hotfix分支需要最终同时合并到develop和master分支（共两次merge操作）
* 除了master和develop分支，其它分支在开发完成后都要删除