

# gitlab项目规范

## 分支定义

2个永驻分支： master 和 develop，其他分支均为临时分支。

分支名称	是否永驻	基础分支	合并到分支	分支含义
master	yes	-	-	主分支
hotfix	no	master	develop,master	热修分支
develop	yes	master	-	开发分支
release	no	develop	develop,master	发布分支
feature	no	develop	develop	功能分支

## 分支的命名和作用

- 主分支: **master**
  - 用途：与线上版本保持一致。
  - 创建分支：初始化时创建master分支。
  - 命名规则：该分支只能叫master，设置为gitlab保护分支。
  - Push来源：禁止直接在master上push代码。
  - Merge来源：release分支和hotfix分支，由组长操作。
  - Tag：merge代码到master分支之后，需打tag，tag命名规则v{xxx}，其中{xxx}为上线或热修日期，如v20190109。
  - 删除分支：不可删除。
- 热修分支: **hotfix**
  - 用途：修复线上问题(经内部评审后)。
  - 创建分支：组长基于master分支拉取hotfix分支。
  - 命名规则：hotfix/v{xxx}，其中{xxx}为master的tag号，如**hotfix/v20190109**。
  - Push来源：组员修复bug后直接push代码到此分支。
  - 打包命名规则：PUB{xxx}，{xxx}为上线热修的日期，如 **PUB20190110**。

- Merge到：线上热修完成后merge到develop和master，由组长操作。
- 删除分支：线上热修完成且合并到develop和master后，由组长删除。

- 开发分支：develop

- 用途：产品开发，整合功能分支。
- 创建分支：组长基于master分支拉取develop分支。
- 命名规则：该分支只能叫develop，设置为gitlab保护分支。
- Push来源：禁止直接在develop上push代码。
- Merge来源：hotfix分支、release分支、feature分支，由组长操作。
- 删除分支：不可删除。

- 发布分支：release

- 用途：打包发布（包括测试环境与线上环境）。
- 创建分支：根据发布计划，组长基于develop分支拉取release分支。
- 命名规则：release/v{xxx}，{xxx}为转测日期，如release/v20190109。
- 打包命名规则：PUB{xxx}，{xxx}为部署环境时的日期，例如 PUB20190109。
- Push来源：组员修复测试环境问题后，直接push代码到此分支。
- Merge到：修复Bug完成后merge到develop分支。由组长操作。
- 删除分支：合并到master分支之后即可删除，由组长操作。

- 功能分支：feature

- 用途：功能开发和功能内测。
- 创建分支：组员基于develop分支拉取feature分支。
- 命名规则：feature/{xxx} 其中{xxx}为特征描述，如feature/ak。
- Merge到：合并到develop分支，由组长操作。
- 删除分支：合并到develop后，或取消开发计划时，由组员操作。

## 对应场景

---

- 功能开发

- 有新需求加入到开发计划后，组员对需求进行设计，划分功能分支，从develop分支拉取功能对应的feature分支。
- 组员根据开发计划，自行开发，每过一段时间就可以将代码push到对应的feature分支上，要求push后的代码可以正常编译。
- push代码之前，先pull一下远端代码，将其他同事已提交的代码拉取到本地。
- 对于提交代码时出现冲突的情况，组员可自行解决，需保证不覆盖他人代码，且合并后能正常编译。
- 功能开发完成后，可进入内测阶段。
- 内测阶段，组员将全部代码push到feature分支上。如测试发现问题，直接在此分支上修改，直至内测通过。
- 内测通过后，即可将feature分支merge到develop分支，此过程由组长操作。merge完成后即可删除此feature分支，删除feature分支的操作由组员完成。
- 对于开发计划取消的情况，组员可直接删除feature分支。

## ● 转测

- 根据发布计划，组长基于develop分支创建release分支。
- 组员根据测试计划，打部署包，提交给测试同事在测试环境部署。
- 测试中出现的问题，在禅道中记录并分配给相关组员修改。组员直接在release分支上修改提交。
- 在release分支上的修改，需merge到develop分支，防止bug在其他分支重现。
- 转测后禁止合并develop中下一次发布计划的相关代码到此分支。
- 转测通过后，打部署包，提交给运维同事在线上环境发布。
- 上线完成后，将release分支merge到master分支，并删除release分支，此过程由组长完成。

## ● 热修

- 线上环境发现问题后，组内讨论修改方案，如需修改代码，组长基于master分支拉取hotfix分支。
- 组员直接在hotfix分支上修改代码。
- 修改完成并内测通过后，打部署包，提交给测试同事在测试环境部署。
- 如测试未通过，组员继续在hotfix分支修改代码，直至测试通过。
- 测试通过后，将部署包提交给运维同事在线上环境发布。
- 热修完成，将hotfix分支代码合并到master分支和develop分支，防止bug在其他分支重现。
- 合并完成后，删除hotfix分支。此过程由组长操作。

