分支管理

主分支（master）：稳定版本代码分支，是对外可以随时编译发布的分支。

- 唯一分支
- 不允许直接Push代码，只允许往这个分支发起merge request。
- 只接受hotfix、release分支的merge。
- 合并后打上版本标签。

热修复分支（hotfix）：针对现场紧急问题、bug修复的代码分支。

- 按需建立，存在多个，完成后关闭。
- 从master分支拉取。
- 修复完后merge到master分支、dev分支。

发版分支（release）：版本发布分支，用于版本发布前测试。

- 唯一分支，针对最近一次上线日的测试内容。
- 从dev分支拉取。
- 测试中bug在该分支修复，发布完成后merge至master分支及dev分支。
- 正式上线前需要需要在该分支上将master分支作为基线进行rebase操作。

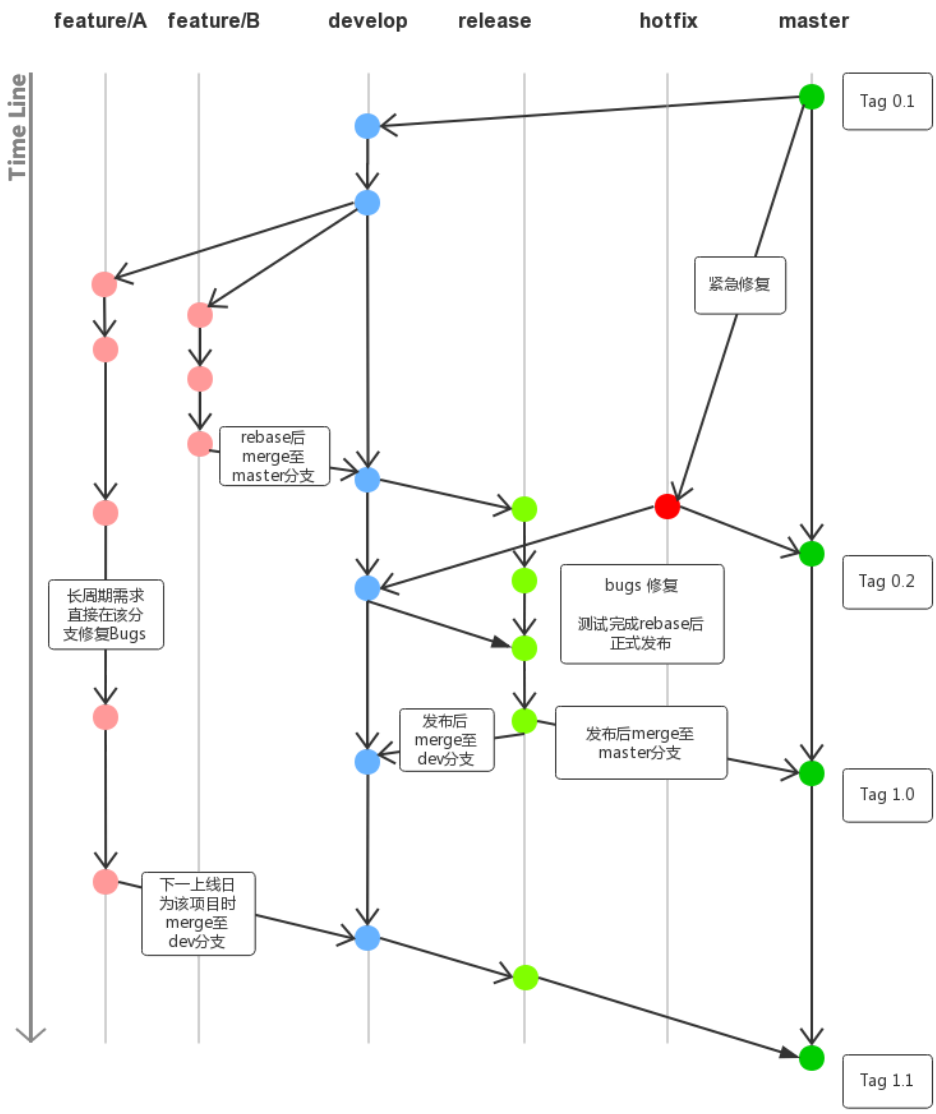
开发分支(dev)：开发版本分支，用于各类开发任务的开发管理。

- 唯一分支，尽量保持稳定。
- 只允许将最近一个上线时间点的代码merge至该分支。
- 不允许直接Push代码，只允许往这个分支发起merge request。

功能分支(feature)：功能分支，用于某一需求的开发工作。

- 和需求任务一一对应，存在多个，完成后关闭。
- 从dev分支拉取，用于对应需求的功能开发。
- 定期以dev分支为基线进行rebase操作。
- 长周期项目，基于该分支直接进行测试及Bug修复，在其上线日为最近一个上线日时merge入dev分支。

> `GitLab`在创建项目仓库后一定要把`master`分支和`develop`分支给保护起来，为它们设置权限。只有系统负责人可以进行推送和删除等操作。被保护的分支在列表中会有特殊的标记进行区分。



分支命名规范

- feature —— 使用需求编号命名；
- hotfix —— bug对应的编号命名；

提交规则

提交：

- 提交时的粒度是一个小功能点或者一个 bug fix，这样进行恢复等的操作时能够将「误伤」减到最低；
- 用一句简练的话写在第一行，然后空一行稍微详细阐述该提交所增加或修改的地方；
- 不要每提交一次就推送一次，多积攒几个提交后一次性推送，这样可以避免在进行一次提交后发现代码中还有小错误。
- commit message 规范（ ***待补充*** ）

推送：

- 当自己一个人进行开发时，在功能完成之前不要急着创建远程分支。

拉取：

- 使用rebase的方式进行拉取。

合并：

- 在将其他分支的代码合并到当前分支时，如果那个分支是当前分支的父分支，为了保持图表的可读性和可追踪性，可以考虑用 git rebase 来代替 git merge；反过来或者不是父子关系的两个分支以及互相已经 git merge 过的分支，就不要采用 git rebase 了，避免出现重复的冲突和提交节点。
- 合并前把本地的多个commits sqush成一个再提交。
- master分支及dev分支不允许进行rebase操作。

分支流程

开发功能

1. 在确定需求后，创建一个对应的 feature 分支。如果是多人配合的话，创建分支并做一些初始化工作之后就推送创建远程分支；否则，直到功能开发完毕要合并进 develop 前，不要创建远程分支。
2. 功能开发完并自测之后，先切换到 develop 分支将最新的代码拉取下来，再切换回自己负责的 feature 分支把 develop 分支的代码合并进来。合并方式参照上文中的「合并」，如果有冲突则自己和配合的人一起解决。
3. 到 GitLab 上的项目首页创建合并请求（merge request）。
4. 「来源分支」选择要被合并的 feature 分支且「目标分支」选择 develop 分支后点击「比较分支」按钮，在出现的表单中将处理人指派为系统负责人。
5. 系统负责人在收到合并请求时，应该先做下代码审核看看有没有明显的严重的错误；有问题就找负责开发的人去修改，没有就接受请求并删除对应的 feature 分支。
6. 在将某次发布的所需功能全部开发完成时，就可以交付测试了。

测试功能

1. 进入测试前，创建一个 release 分支部署到测试环境进行测试；若发现了 bug，相应的开发人员就在 release 分支上或者基于 release 分支创建一个分支进行修复。

发布上线

1. 当确保某次发布的功能可以发布时，负责发布的人将 release 分支合并进 master 和 develop 并打上 tag，然后打包发布到线上环境。
2. 建议打 tag 时在信息中详细描述这次发布的内容，如：添加了哪些功能，修复了什么问题。

修复问题

1. 当发现线上环境的代码小问题时，相关开发人员基于master分支在本地创建 hotfix 分支进行修改。如果严重问题，回滚到上一个 tag 的版本。