

1 分支说明

功能分支 workflow 包含分支: **feature, master, hotfix**

1. feature 分支

当有需求需要开发时, 可以从 **master** 分支创建一个 **feature** 分支, 在 **feature** 分支做开发测试, 测试通过后可以结束 feature 开发分支, feature 分支会被合并回 master 分支, 并创建 tag;

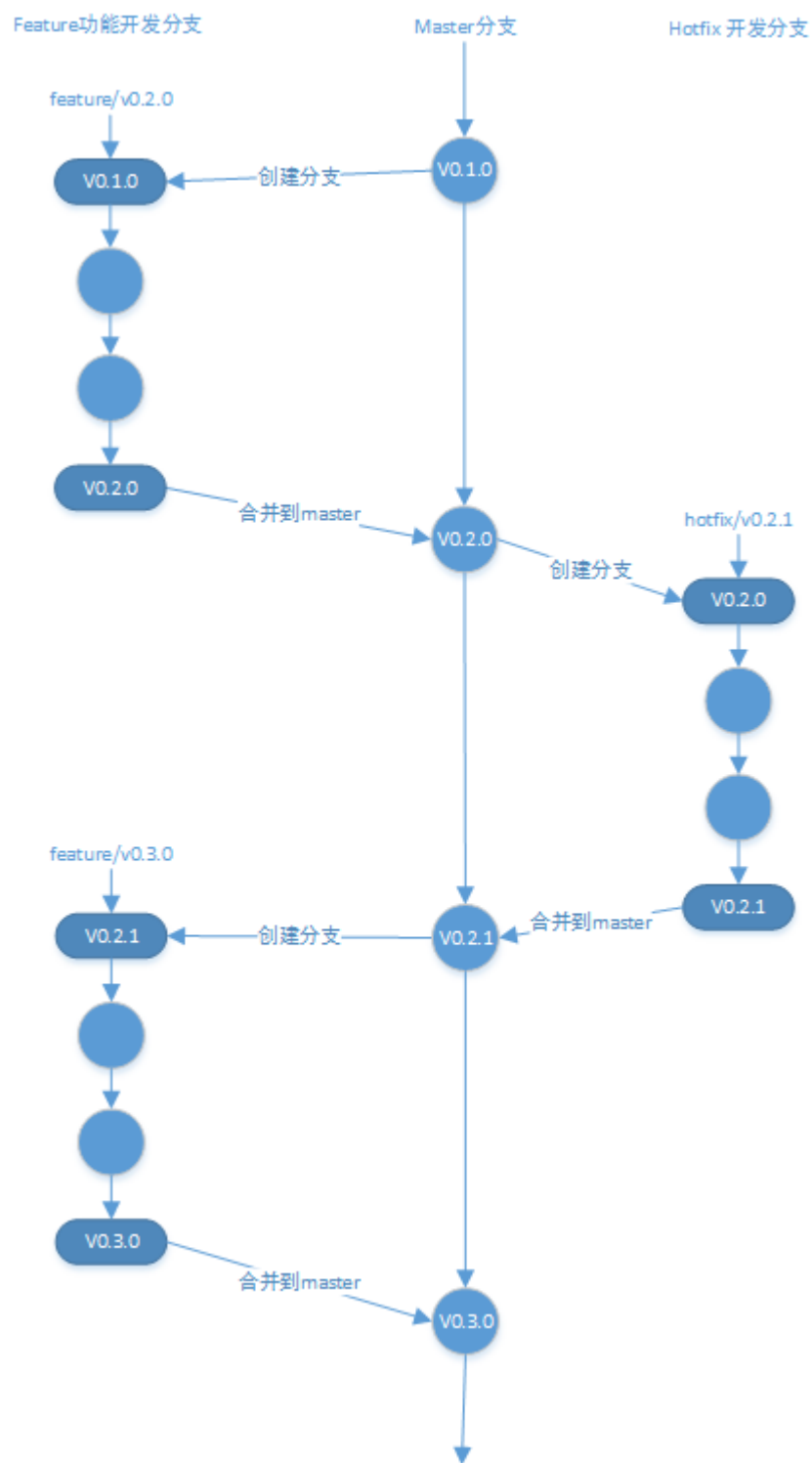
2. master 分支

master 保存所有版本的信息, 预发布和生产环境使用 master 分支 tag 发布。
用来做 hotfix 分支和 feature 分支的创建。

3. hotfix 分支

如果线上版本出现问题, 可以从 master 的 tag 创建 hotfix 分支做修复测试, 然后部署, 发布完成后, 结束 hotfix 分支, 合并回 master 分支

4 流程图



说明：
1 箭头表示时间轴

2 分支命名规范

命名规范：分支主名称-主版本号.子版本号.修正版本号.自定义

示例：hotfix-v1.0.1

或者：feature-v1.0.0

或者：feature.功能模块名称.v0.1.0.xxx(运维用) hotfix.功能模块名称.v0.1.0.xxx(运维用)
管理策略：

- 1 项目初版本时，版本号可以为 0.1.0，也可以 1.0.0；
- 2 当项目在进行了局部修改或 bug 修正时，主版本号和子版本号都不变，修正版本号加 1；
- 3 当项目在原有的基础上增加了部分功能时，主版本号不变，子版本号加 1，修正版本号复位为 0，因而可以被忽略掉；
- 4 当项目在进行了重大修改或局部修正累积较多，而导致项目整体发生全局变化时，主版本号加 1；

3 版本管理和 tag 命名规范

版本管理基于 tag 号版本命名：由 tag 号保证唯一性。具体规范如下：

命名规范：v 主版本号.子版本号.修正版本号.自定义

示例：v1.0.0.20180101 v1.0.1.release

管理策略：

- 1 项目初版本时，版本号可以为 0.1.0，也可以 1.0.0；需要上线的 tag 版本号命名与提测的分支版本命名的主版本号.子版本号.修正版本号需要保持一致。
例如提测是 feature-v1.0.1 分支，合并到 master 后 tag 命名：v1.0.1
- 2 当项目在进行了局部修改或 bug 修正时，主版本号和子版本号都不变，修正版本号加 1；
- 3 当项目在原有的基础上增加了部分功能时，主版本号不变，子版本号加 1，修正版本号复位为 0，因而可以被忽略掉；
- 4 当项目在进行了重大修改或局部修正累积较多，而导致项目整体发生全局变化时，主版本号加 1；

5 自定义部分参考：

- **alpha 版**：内部测试版。 α 是希腊字母的第一个，表示最早的版本，一般用户不要下载这个版本，这个版本包含很多 **BUG**，功能也不全，主要是给开发人员和 测试人员测试和找 **BUG** 用的。
- **beta 版**：公开测试版。 β 是希腊字母的第二个，顾名思义，这个版本比 **alpha** 版发布得晚一些，主要是给“部落”用户和忠实用户测试用的，该版本任然存 在很多 **BUG**，但是相对 **alpha** 版要稳定一些。这个阶段版本的软件还会不断增加新功能。如果你是发烧友，可以下载这个版本。
- **rc 版**：**Release Candidate**（候选版本），该版本又较 **beta** 版更进一步了，该版本功能不再增加，和最终发布版功能一样。这个版本有点像最终发行版之前的一个类似 预览版，这个的发布就标明离最终发行版不远了。作为普通用户，如果你很急着用这个软件的话，也可以下载这个版本。
- **stable/release 版**：稳定版。在开源软件中，都有 **stable** 版，这个就是开源软件的最终发行版，用户可以放心大胆的用了。