

Git flow 工作流

Author: 张全地
2022.05.31

start →

Git branch分支

长期分支

1. master（主分支）

- 版本库初始化以后自动建立的
- 代码库中有且仅有一个主分支，不可删除
- 不要直接修改master分支代码
- master分支对外发布，每一次推送应该打标签（tag）做记录，方便追溯

2. develop（开发分支）

- 基于master分支创建
- 代码库中有且仅有一个开发分支，不可删除
- 不要直接修改master分支代码

短期分支（协助分支）

3. feature（功能分支）

- 新功能或新特性开发分支
- 基于develop分支创建
- 新特性或新功能开发完成，合到develop分支
- 临时分支，功能完成后可删除

4. release（预发布分支）

- 基于develop分支创建（feature分支合并到develop分支之后）
- 临时分支，产品上线后可选删除
- 发布到内部测试区，供测试人员测试

5. hotfix（补丁分支）

- 基于master创建
- 对线上的版本打补丁或bug修复
- 修复完成后，合并到master分支和develop分支
- 临时分支，修复上线后可删除

Gitflow 工作流

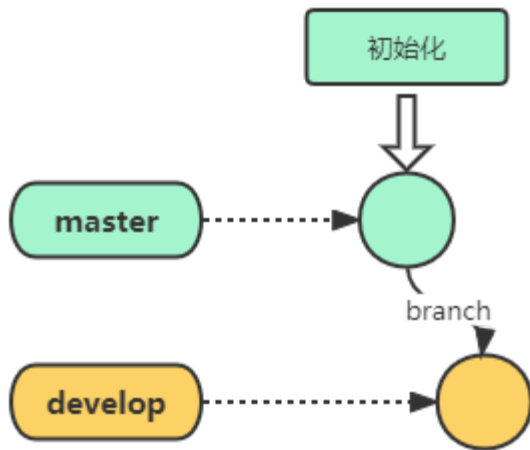
GitPrime CEO:

Vincent Driessen (文森特·德里森) 2010年

博文: 成功的 Git 分支模型

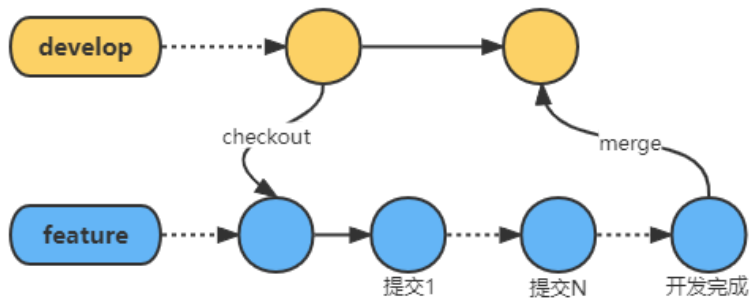
创建项目初始化仓库

- 初始化仓库，自动创建master
- 基于mater创建develop分支



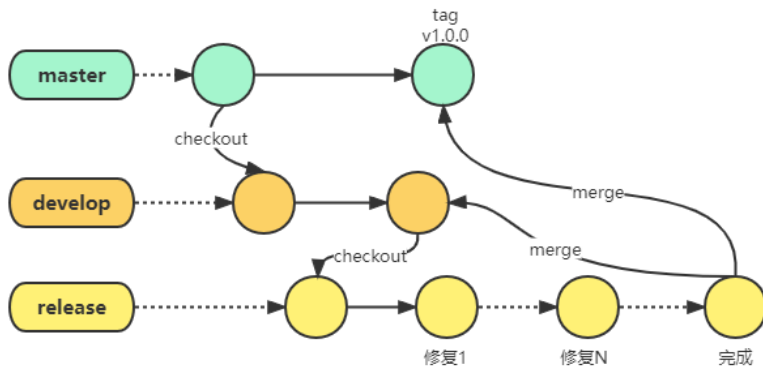
开发创建功能分支

- 基于develop创建feature分支
- 功能分支完成，merge到develop分支



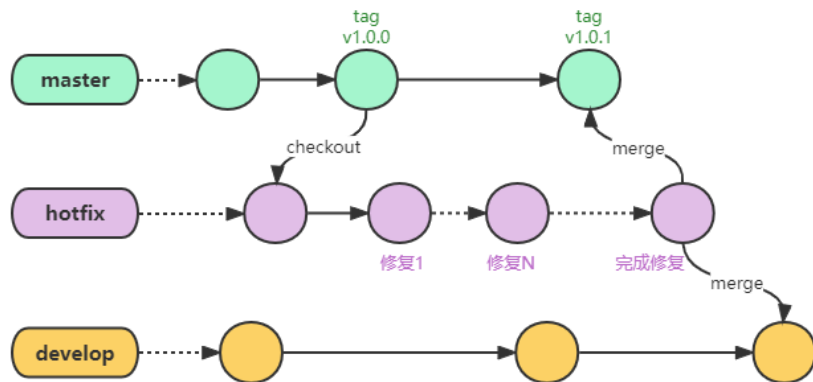
预发布

- 基于develop创建release分支
- 完成release后merge到master分支，同时merge到develop分支

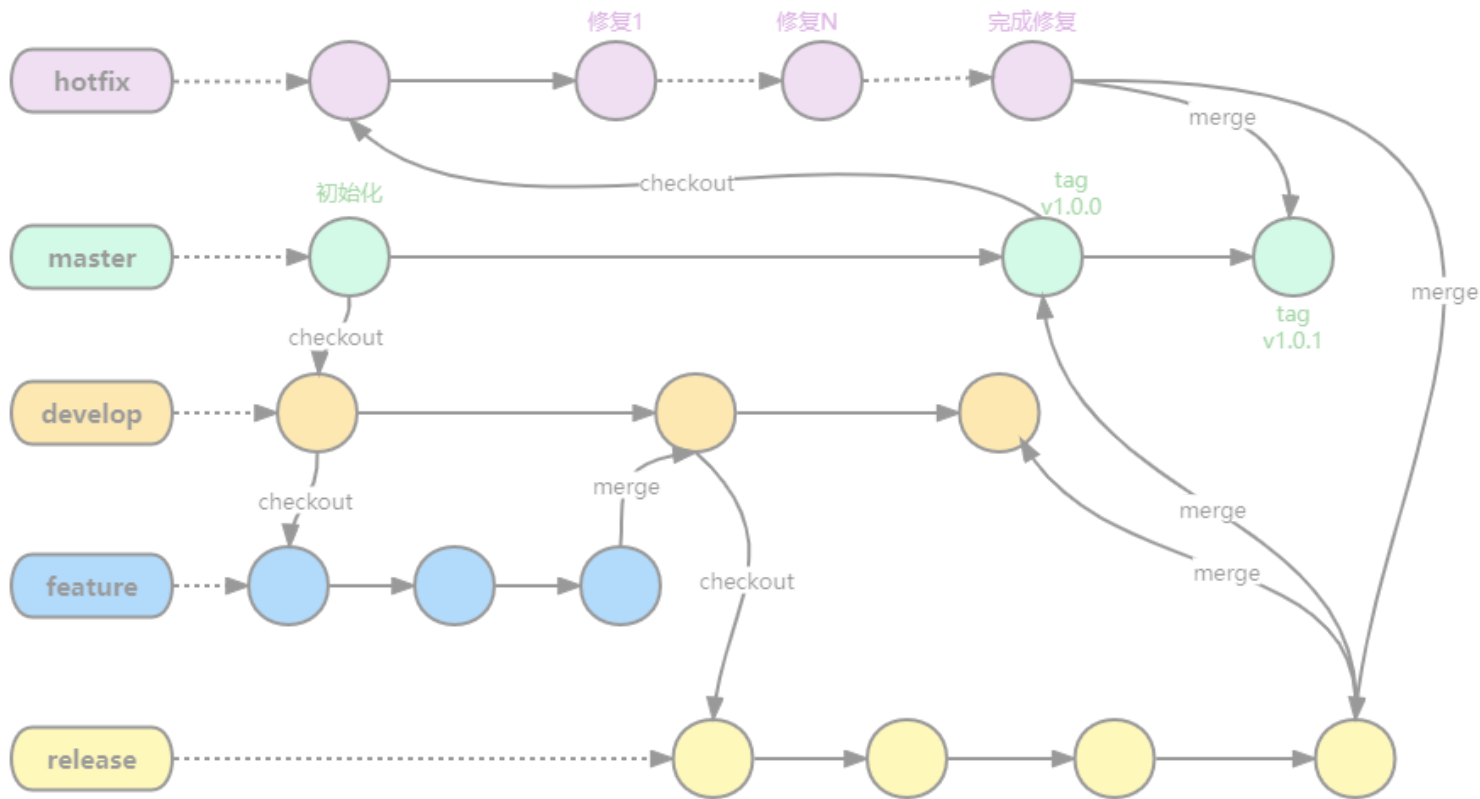


hotfix修复（线上版本出现问题）

- 基于master创建hotfix分支
- 修复完成后，merge到master分支，同时合并到develop分支



完整的gitflow工作流程图



Git flow落地

初始化项目

- 创建并切换到Develop分支

```
git checkout -b develop master
```

- 创建feature功能分支

```
git checkout -b feature-x develop
```

功能开发

- 开发完成后，将功能分支合并到develop分支：

```
#切换到develop
git checkout develop
#合并功能分支到develop
git merge --no-ff feature-x
#切换到功能分支并删除
git branch -d feature-x
```

预发布 => 发布

- 创建release预发布分支

```
git checkout -b release-1.2 develop
```

- 完成后，合并到master分支

```
#切换到master分支
git checkout master
#合并release到master
git merge --no-ff release-1.2
# 对合并生成的新节点，做一个标签
git tag -a 1.2
```

- 再合并到develop分支

```
#切换到develop分支
git checkout develop
#合并release到develop
git merge --no-ff release-1.2
```

- 合并完成之后删除release分支

```
git branch -d release-1.2
```

hotfix修复

- 创建hotfix分支

```
git checkout -b fixbug-0.1 master
```

- 完成后，先合并到master分支

```
#切换到master分支  
git checkout master  
#合并hotfix到master  
git merge --no-ff fixbug-0.1  
#打标签  
git tag -a 0.1.1
```

- 再合并到develop分支

```
#切换到develop分支  
git checkout develop  
#合并hotfix到develop  
git merge --no-ff fixbug-0.1
```

- 删除hotfix分支

```
git branch -d fixbug-0.1
```

落地方式二

安装git-flow工具

落地方式三

使用Sourcetree可视化工具管理

安装Sourcetree工具

最新版本Sourcetree安装教程

初始化项目

clone示例项目: <https://gitee.com/quandizhang/git-flow.git>

使用git工作流初始化

- 点击确定后，本地会自动创建并切换到develop分支。

- 我们将本地develop推送到云端

接下来，我们创建feature功能分支

- 基于develop分支创建 填写分支名称

- 创建完成 自动切换到新创建的分支

- 同样，我们需要把新创建的分支推送到云端仓库

- 完成feature功能分支

- 合并到develop, 并删除功能分支:

创建release预发布分支

- 基于develop创建 填写预发布版本名称

- 创建完成

- 同样需要推送到远程仓库：

- 完成预发布版本

- 标签（版本号） 删除release分支 合并到master分支，合并到develop分支

查看标签：

创建hotfix

- 基于master创建 填写补丁名称

- 同样，需要推送到远程仓库：

- 完成补丁修复

- 标签（版本名称） 合并到master, 合并到develop 删除hotfix分支

- 这边会生成一个新标签: