

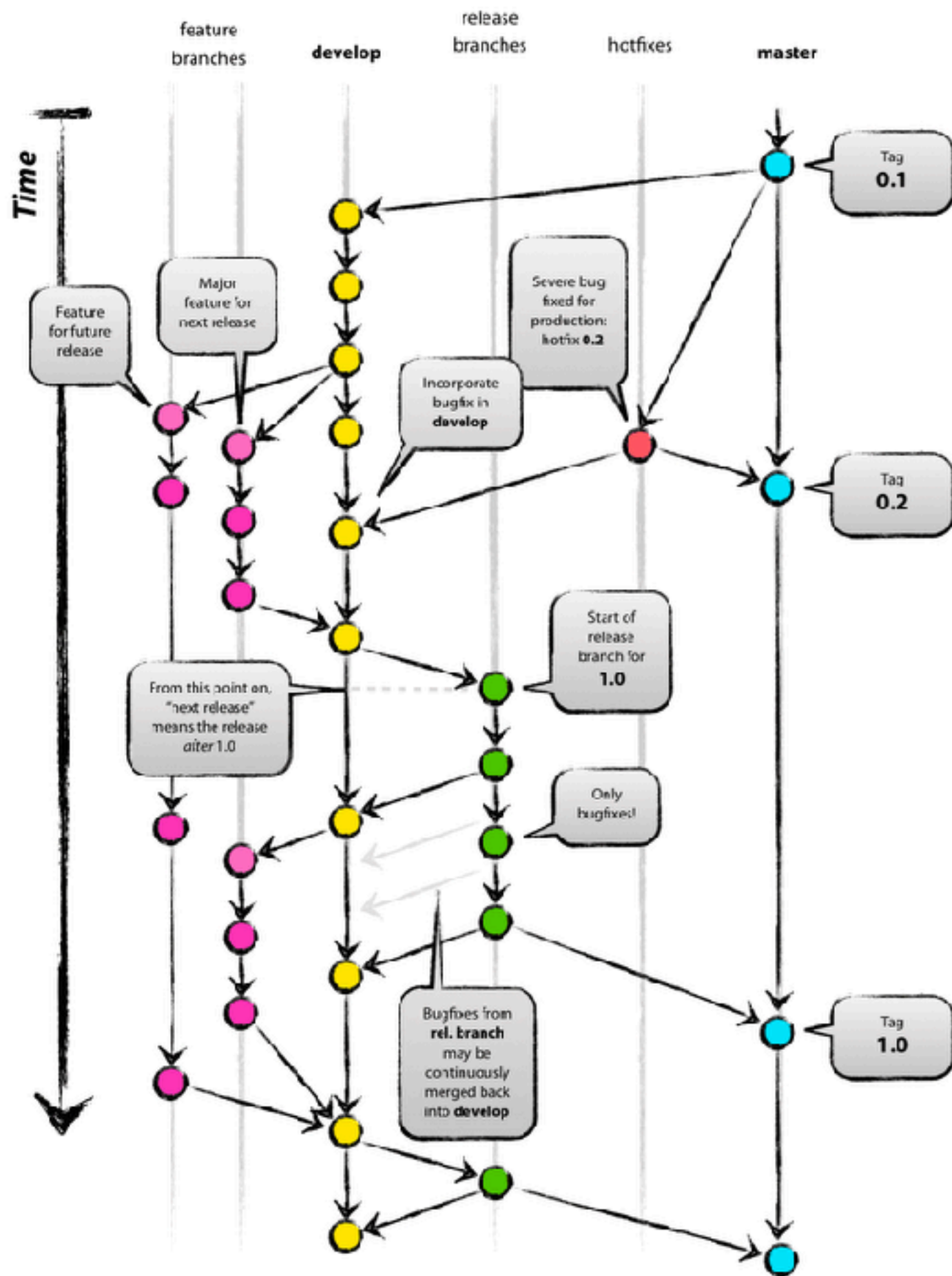


王建华

# git的简介

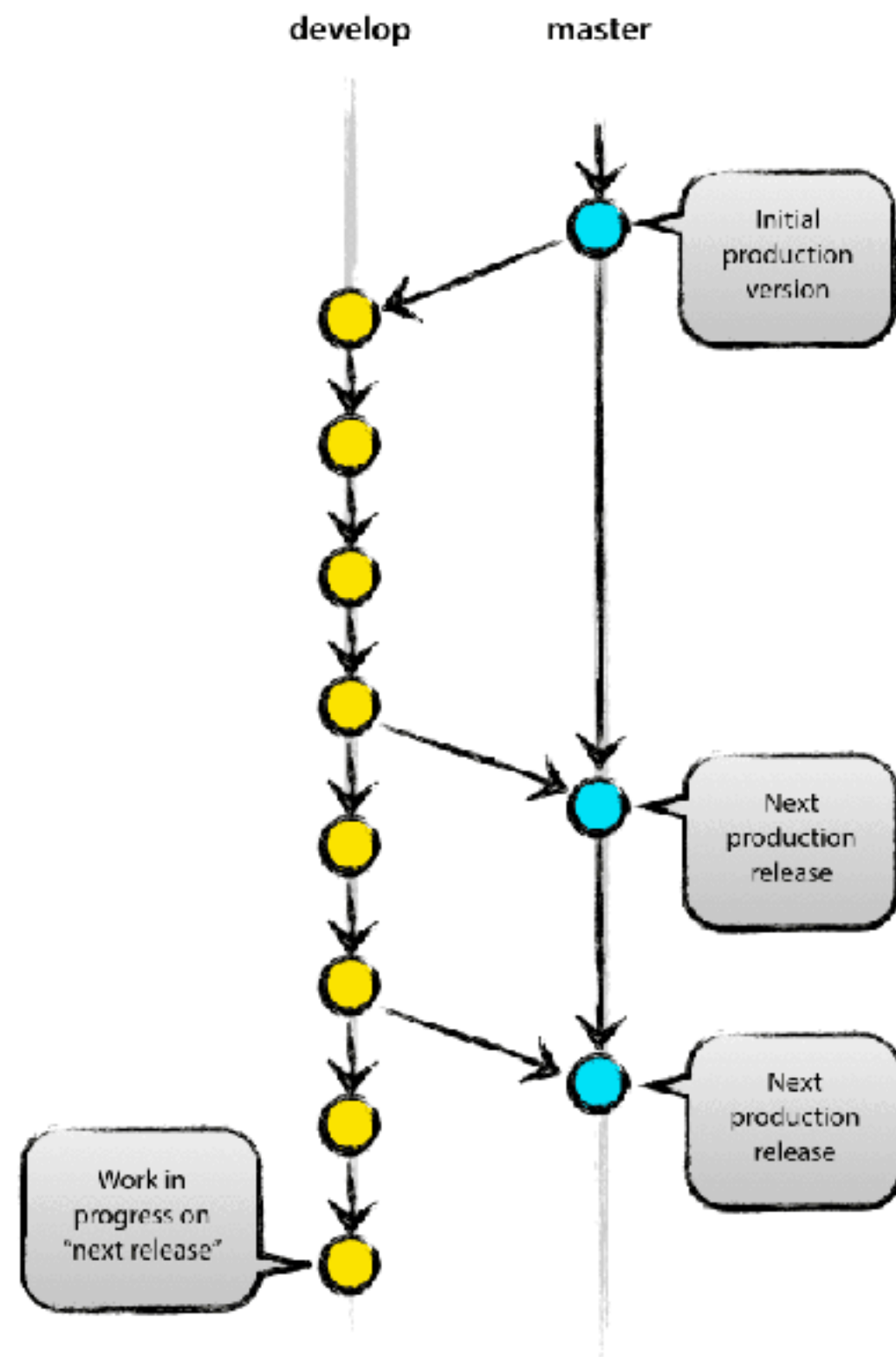
- 分布式版本控制软件
- 作者林纳斯·托瓦兹，2005年面世最初用来管理linux内核
- 基于git发展出很多款代码远程托管服务  
(gitlab,github,git@OSC)

# git workflow



# 主分支

master(准备发布生产环境的分支)  
develop (开发新功能的分支)

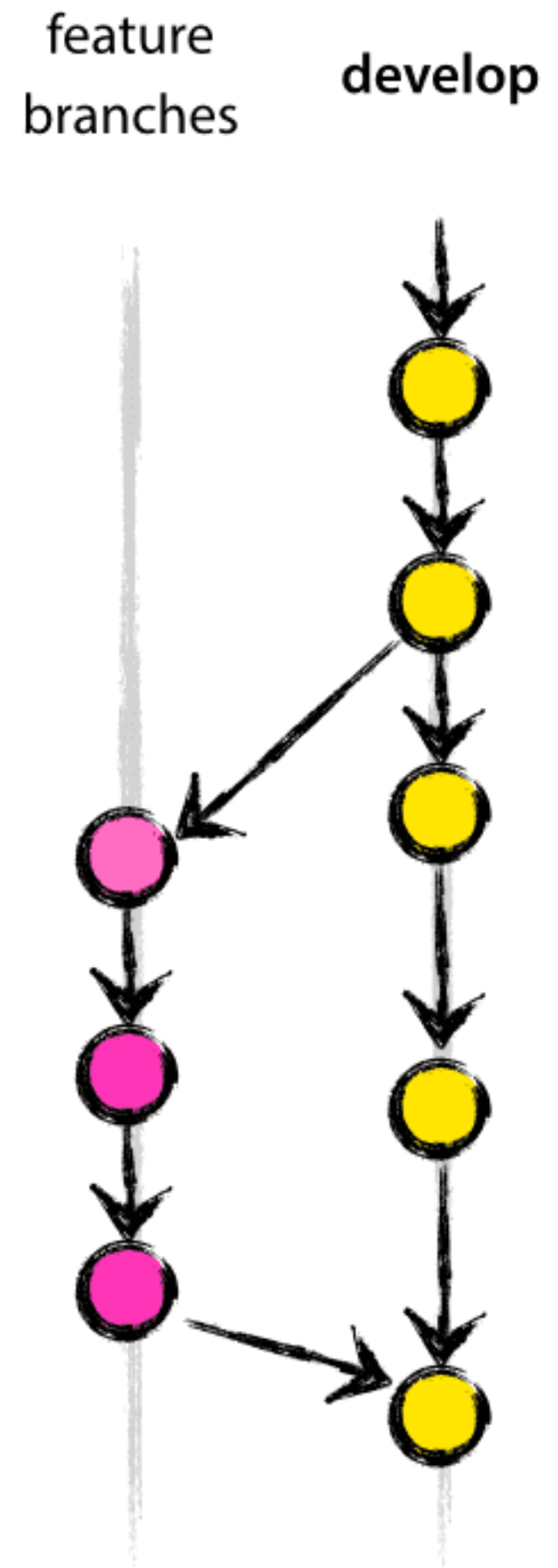


# 支持分支

- Feature (开发新功能分支)
- Release (提供生产环境发布的版本)
- Hotfix (bug修复分支)

# feature分支

- 从develop拉出，最终必须合并回develop
- 命名规则：理论上除去master, develop, release-\*, 以及hotfix-\*, 但是同时要明确你这个分支的主要新功能，拒绝用你们的名字拼音来写. ex: feature/add-new-source

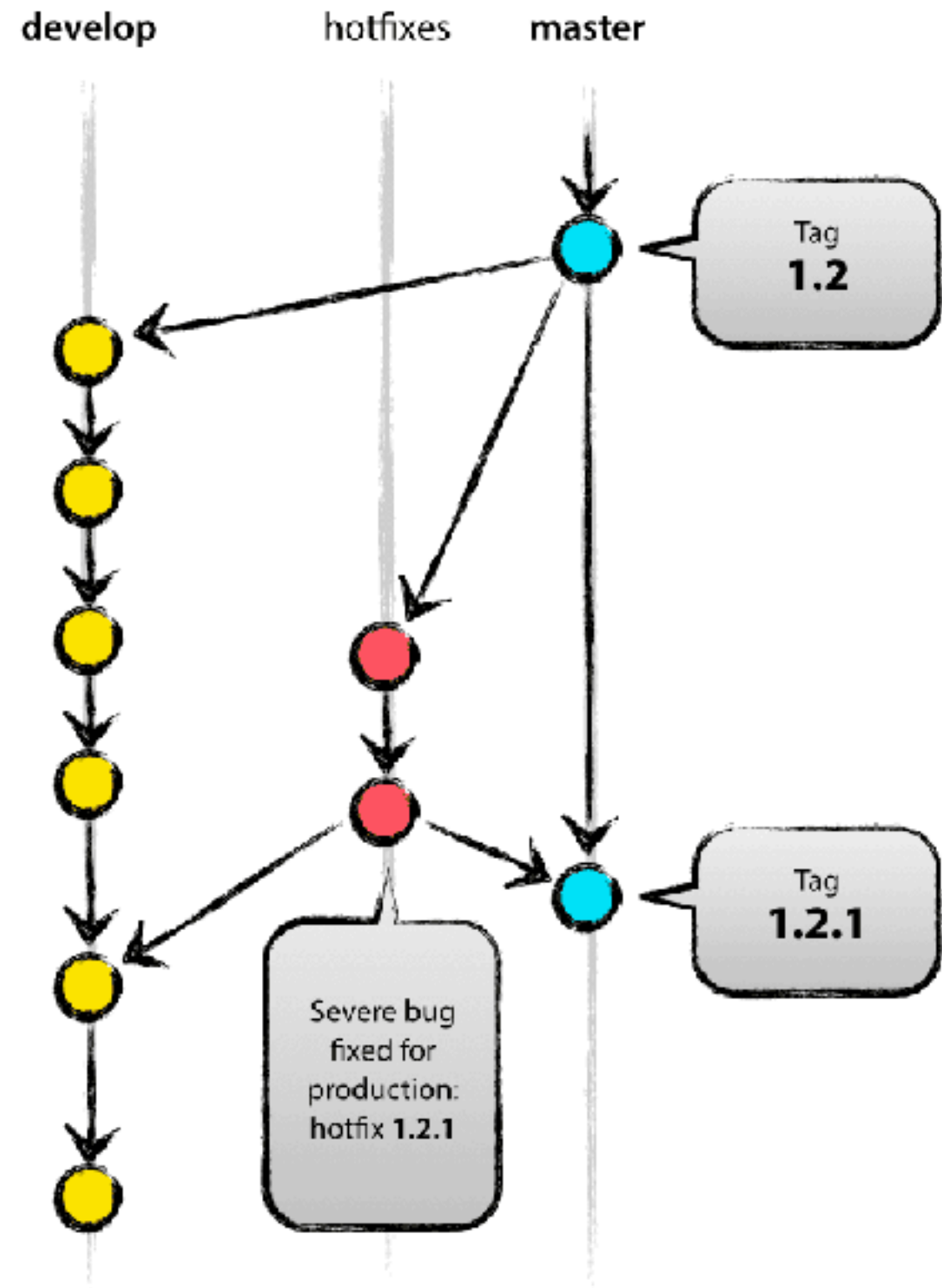


# release分支

- 从develop拉出，同时最终合并到develop以及master
- 命名规则： release-\*

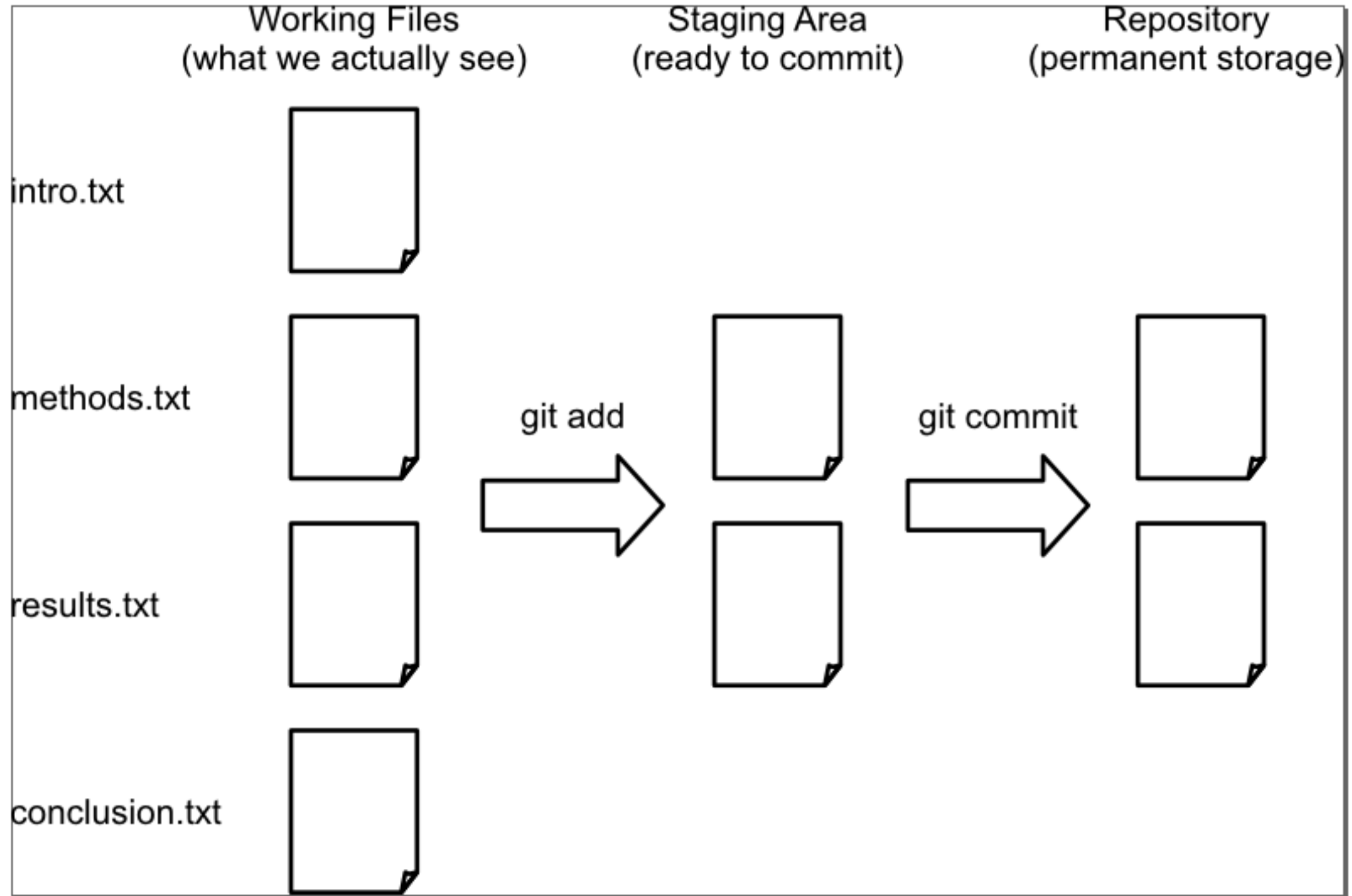
# hotfix分支

- 从master分支拉出，同时最终回到develop以及master
- 命名规范：hotfix-\*





# 本地提交过程



# 常用命令

查看，提交，添加，删除，找回，重置，修改

- `git show` # 显示某次提交的内容 `git show $id`
- `git checkout <file>` # 抛弃某个文件工作区域修改
- `git checkout .` # 抛弃当前工作区域所有修改
- `git add .` # 添加所有修改过的文件到缓存区
- `git rm <file>` # 删除某个文件，同时删除本地文件
- `git rm <file> --cached` # 从版本库删除文件，但不删除本地文件
- `git reset .` # 从暂存区恢复到工作文件
- `git reset <file>` # 指定文件从暂存区恢复到工作文件
- `git reset --hard` # 恢复到上一次提交，放弃本次修改
- `git commit --amend` # 修改上次的提交信息
- `git revert <$id>` # 对某一次提交做反操作，放弃某一次提交

查看差异

- `git diff` # 当前文件与暂存区的差异
- `git diff --cached` # 暂存区与版本库的差异
- `git diff <br1> <br2>` # 比较两个版本之间的差异

查看提交记录

- `git log <file>` # 查看某个文件的提交记录
- `git log -p <file>` # 查看某个文件每次详细的提交记录信息
- `git log -p -2` # 最近两次的详细提交记录

本地分支管理

- `git branch -r` # 查看远程分支
- `git branch -b <new_branch>` # 创建新分支
- `git branch -v` # 查看各个分支最后的提交时间
- `git branch --merged` # 查看已经被合到当前分支的分支
- `git checkout <branch>` # 切换到某个分支
- `git branch -d` # 删除某个分支
- `git branch -D` # 强制删除某个分支

远程仓库

- `git remote -v` # 查看远程服务器地址和仓库名称
- `git remote origin` # 添加远程仓库地址

# 常见问题

- 当我在开发一个新功能，然后突然线上出bug了，但是我本地代码还没提交此时怎么办？
- 代码冲突了咋办？
- 远程分支地址修改了咋办？
- 待补充