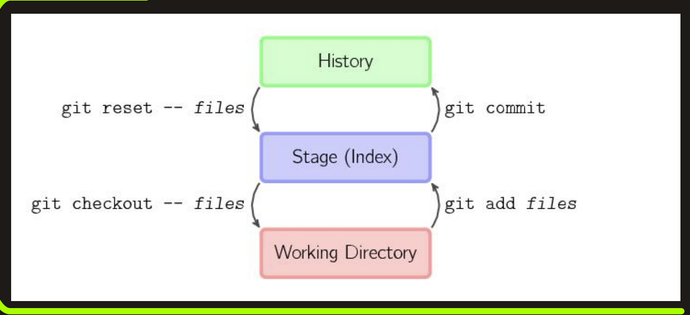
Git使用

组成部分：

- 工作区（Working Directory） 编辑文件内容时位置
- 暂存区（Stage） 准备提交的快照
- 历史记录区（History） commit 后的记录区

三者关系：

相互转变：



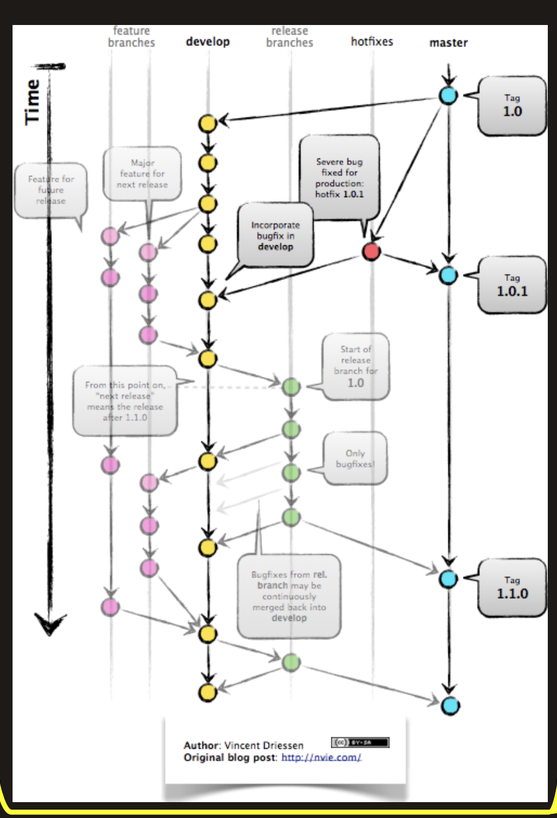
GitFlow：

GitFlow：指git操作流程标准 由 Vincent Driessen 提出

分支结构：

1. master 主分支
2. develop 主开发分支，包含确定即将发布的代码
3. feature 新功能分支，一般一个新功能对应一个分支，对于功能的拆分需要比较合理，以避免一些后面不必要的代码冲突
4. release 发布分支，发布时候用的分支，一般测试时候发现的 bug 在这个分支进行修复
5. hotfix 分支，紧急修 bug 的时候用

工作流程:



- develop master创建 开发主干
  - 不需要合并到其他分支
  - 接受feature合并请求
  - 接受release合并请求
  - 接受hotfix合并请求
- feature develop创建 新功能开发分支,可同时多条分支 开发完成合并到develop
- release develop创建 发布分支
  - 发布到测试环境进行测试
  - 不停的重复发布–>测试–>修复–>重新发布–>重新测试
  - 测试修复结束：
    - 合并到 develop
    - 合并到 master
- hotfix master创建 紧急修复一些 Bug
  - 测试–>修复–>测试–>修复..
  - 测试修复结束：
    - 合并到 develop
    - 合并到 master
- master 初始化创建
  - 发布主干
    - 不需要合并到其他分支
    - 接受hotfix合并请求
    - 接受release合并请求
  - 上线版本分支, 版本tag

Code Review

- PR (pull request) 开发者–>开发–>检查Review(和脚本)–>编译ok–>提交
- MR (merge request) 管理者–>检查Review–>合并
- fork fork代码 –> clone,修改 –> 提交(PR) –> 原创作者Review –> 合并(MR)

githooks

- git 在执行某些特定操作时会触发的一系列程序或脚本等可执行文件
  - 自动化管理时,自动检测语法,编译,上传到第三方等
- pre-commit： 由 git commit 触发， 比如触发语法,代码规范检测
- commit-msg： 由 git commit 触发, 规范提交信息hook, 不符拒绝提交
- update, merge, checkout等等都可以hook

.gitignore

忽略上传文件,比如pods文件

```
#CocoaPods
Pods/

# Xcode
#
# gitignore contributors:
remember to update Global/
Xcode.gitignore, Objective–
C.gitignore & Swift.gitignore

# Mac OS X Finder and
whatnot
.DS_Store

## Build generated
build/
DerivedData/

## Various settings
*.pbxuser
!default.pbxuser
*.mode1v3
!default.mode1v3
*.mode2v3
!default.mode2v3
*.perspectivev3
!default.perspectivev3
xcuserdata/

## Other
*.moved–aside
*.xcuserstate
*.xccheckout

## Obj–C/Swift specific
*.hmap
*.ipa
*.dSYM.zip
*.dSYM


Podfile
Podfile.lock
```

Podfile.lock

.git

维护信息:

- COMMIT_EDITMSG：保存最近一次 commit message 提供给用户使用；
- FETCH_HEAD：保存本地的 FETCH_HEAD 记录，用于 git fetch 时和远程仓库的版本号做对比；
- HEAD：这个文件包含了一个当前 branch 的引用，通过这个文件 git 可以得到下一次 commit 的parent；
- config:当前仓库的 git 配置文件；
- description:仓库的描述信息，主要给 gitweb 等 git 托管系统使用；
- hooks:这个目录存放一些 shell 脚本，可以设置特定的git命令后触发相应的脚本；
- index:这个文件就是暂存区（stage），是一个二进制文件；
- info:当前仓库的一些信息，里面有一个 exclude 文件，记录了被 .gitignore 忽略的文件；
- logs:保存所有的更新引用记录，里面的 refs 文件夹，分文件记录了各个分支的更新引用记录；
- objects:该目录存放所有的 git 对象，对象的 SHA1 哈希值的前 2 位是文件夹名称，后 38 位是对象文件名。
- refs:具体的引用，Reference Specification，这个目录一般包括三个子文件夹，heads、remotes和 tags，比如，heads 中的 master 文件标识了当前仓库 master 分支指向的当前 commit，以此类推；
- sourcetreeconfig：Source Tree 的配置信息。