GitFlow总结归纳

一些相关概念

develop分支：develop是我们进行开发时所用的分支，一般来说develop分支主要用于开发。

2．Feature分支：主要进行新功能的开发，是从develop分支上拉出来的一个分支，主要用于产品功能的创建和修改，当新功能完成后会合并develop上

3.release分支：主要用于产品的测试和bug一些文档的处理，是从develop分支上拉出来的一个分支，当测试或者修改bug完成后合并到master分支和develop分支上

4.hotfix分支：故名思议热更新，当发现新bug时会进入release分支。

5.要理解 git常用一些命令 比如 什么是暂存区等一些相关概念及一些常见的命令操作

如 git add 和git commit git diff git head git log git relog等

演示操作流程

下面为演示流程，以远程库github为例

1.创建一个develop分支

1).首先先创建一个本地仓库 git init

2).在maste分支上创建一个名为develop 的分支 git checkout –b develop(创建一个名为develop 的分支，并定位到该分支

3).建立与远程仓库github 的联系 git remate add origin url(意味远程仓库的所在地址)

4).本地仓库与远程仓库进行通信 git push -u origin develop

2.当需要进行feture新功能开发时

1).创建一个名为feature的分支，为了方便记忆区分，一般通常命名为feature+其他一些名字 git checkout -b feature/ zgl

2).与远程仓库建立连接，push到远程仓库 git push –u origin feature/zgl

当功能开发完成后

3). git add zglTest.txt git commit –m”zgl”

4).与远程库做一下比较是否有冲突 git pull origin feature/zgl

5).当确定没有冲突以后，切换到develop分支 git checkout develop

6).合并feature分支 git merge --no-ff feature

7).push到远程仓库 git push –u orgin develop

8).删除feature分支和远程的feature分支 git branch –D feature/zgl和 git branch –delete feature/zgl

3.当需要创建一个release分支时

1).创建一个release 分支(通常release分支需要用版本号命名) git checkout –b release-1.0.1 develop

2).创建一个远程的release git push –u release-1.0.1

比如当测试完后后

3).首先切换到master分支和master进行合并 git checkout master和git merge –no-ff release-1.0.1

4).push到远程的master git push

5).切换到develop分支进行和develop进行合并 git checkout develop 和git merge –no-ff release-1.0.1

6).删除合并完的release分支和远程仓库的release分支

7).打印一下版本号 git tag –a v1.0.1 master和git –tags

3.hotfix分支的使用与release分支的使用非常相似，只不过hotfix是从master分支上拉出来的，完成hotfix后会进入release分支，步骤与release分支步骤基本无差