5EülJiit%-Eqa , , 
É5EtjJE , Edevelop}JqatöÆ0 
develop-EffN5-E 
develop 
5 
release*OE : release/classstatic_20170220 

上图是沪江的gitflow的流程

 //基本所有的公司的git’代码管理，都遵循GitFlow的流程：

**git-flow工具:让代码提交流程化，防止程序员不理解流程，导致代码提交流程出问题！！！！**

master： 主分支, 保留发布版本，线上代码和master是同步的

develop： 开发分支, 保留所有历史版本 ：所有新项目的feature分支，都是从develop拷贝产生【qa测试】

Feature分支【feature/name】：feature 功能特性分支, 实现功能开发：各个项目都有各自的feature分支，功能一旦开发完成本地自测没问题，就需要合并到develop分支【develop分支内的代码都是最近的版本发布用到的，不是最近的发布的代码不能合并到develop，不然会被错带上线】：单人开发功能环节

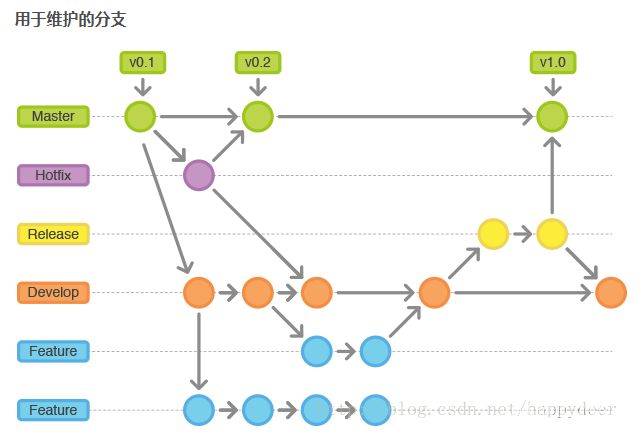
Release【release/name】： 负责代码测试,bug修复,分包发布等：release的创建，标志着不再添加新功能， 存放测试版本+上线；【yz测试】

Release分支的建立，标志着这个项目的开启测试到发布的过程【不再添加新功能，只修复bug，所以一旦Release分支创建，正常来说是不能在去rebase最新的develop分支了，因为要开发的功能之前在develop中以及全部写好了，进入的是准备发布阶段】。

Hotfix分支【hotfix/name】： 紧急bug修复分支, 负责处理线上bug，从master分支拷贝【因为是线上出问题，所以拷贝和线上同步的master，因为可能有其他项目上线，所以从master拷贝，最后上线之前，还得rebase一下master来确保修复bug期间是否有其他项目上线，如果有的话还得让测试做回归测试】

注意：hofix和release是处理发布上线的分支，发布完以后需要合并到master并且给master打赏对应的版本标签【如何是hotfix，不但要有版本标签，还得有hotfix的bug修复标签】

下面是统一的gitflow的图解流程



//gitflow基本大同小异，关键还是各个公司有自己的差异化操作

1. Master分支只能是通过合并，不能自己进行提交操作
2. release发布和hotfix发布后，都需要合并到master并且打tag；同时代码也要合并到develop上去，让master和develop都包含线上的功能
3. Hotfix或Release代码上线之前，还需要再rebase一下master，因为中间可能有人又提交上线的一部分功能【不管是版本上线还是bug修复】，所以只要中间时间有点长，就需要和团队内部人员确认是否有新功能上线，或自己rebase一下master

只要中间没有其他项目上线，那么就可以直接上线，如果rebase master分支的时候，出现了新代码，那么就还得和测试说，需要做一个回归测试。

1. 在自己的feature分支开发的时候，需要频繁的去rebase当前的develop分支，因为可能又新功能添加进去了，而这个新功能可能会对我的功能产生交叉影响，所以需要过一段时间就rebase一下develop，正常分配开发任务的时候，类似有交叉的功能，都会被分到同一个feature分支，但是也可能分错，所以需要rebase develop

如果rebase当前的develop分支的时候，频繁出现修改相同文件，那么就要和另外一个feature分支的开发同时商量使用同一个feature分支，因为都是交叉的功能。

1. Develop分支上，集成的都是下一个版本要发布的功能，所以比较远的功能，是不能合并到develop分支上的，因为测试和发布的release分支是从develop分支那边派生出来的
2. 沪江的测试分为：分支测试【个人功能：feature】；qa测试【develop分支，多功能集成测试（多个开发人员的功能）】；yz测试【develop上的qa测试通过了，就创建release分支进行yz测试】
3. 如果某些develop的功能不需要上线，那么一旦提交到develop或者release，就需要reset到之前的版本，然后再把其他功能集成进来，这个比较麻烦