GIT分支操作及协作参考规范

为了在项目管理中更好的进行版本控制及源码管理，推荐采用git进行维护管理，考虑到团队协作及提交代码各种操作需要统一流程，以避免不必要的冲突或分支错乱导致影响开发效率，作出如下规范说明，请所有开发人员参考遵循。

1. 分支命名（以下为大型项目且团队协作人数较多时参考规范，小项目不建议）  
   master（主分支，永远是可用的稳定版本，不直接在该分支上开发）  
   develop（开发主分支，所有新功能以这个分支来创建自己的开发分支，该分支只做只合并操作，不接在该分支上开发）  
   feature-xxx（功能开发分支，在develop上创建分支，以自己开发功能模块命名，功能测试正常后合并到develop分支）  
   feature-xxx-fix(功能bug修复分支，feature分支合并之后发现bug，在develop上创建分支修复，之后合并回develop分支。PS:feature分支在申请合并之后，未合并之前还是可以提交代码的，所以feature在合并之前还可以在原分支上继续修复bug)  
   hotfix-xxx（紧急bug修改分支，在master分支上创建，修复完成后合并到 master）
2. 提交规范  
   git commit 的描述能为后期搜索提供方便，提交时备注不要出现无意义的描述。  
   描述中应尽量提现此次提交做了什么及为何如此做，最好包含 修改、新增、删除、注释、停用、修复、测试 等关键词便于检索，或为后期回忆变更原因提供辅助信息。  
   示例1：修改XX接口 或 修改XX函数功能等  
   示例2：新增A接口、B接口、C接口。。。  
    （删除、注释、停用）XX（文件、接口、功能、函数等）  
    修复 XXX BUG  
   示例3：临时测试XXX（接口、功能、函数等）
3. 协作注意事项  
   推荐每人使用独立的账号进行提交及推送操作，如果必须要多人共用一个账号操作时，在提交描述里必须添加提交人，以便后期查询。