使用Git辅助开发肯定要有一套流程标准，否则每一个人的commit提交习惯不同，可能最终会出现比较混乱的局面。2010年，有组织制定一套Git开发流程标准，希望大家遵守，当然标准不是硬性的，不同公司有所不同。

下面简单介绍一下，此套标准的总体原则，初学者可以做一下借鉴，当然可能公司有更为科学的方式。

**一.分支种类：**

当在实际应用开发中，特别是比较复杂的项目，Git不可能只存在一个分支。

比如主分支存放当前线上稳定版代码，还有正在进行测试的非稳定版本分支，或者还有正在开发新功能的分支。

开发建议Git有如下分支（不是影响固定，仅供参考）：

（1）.master。

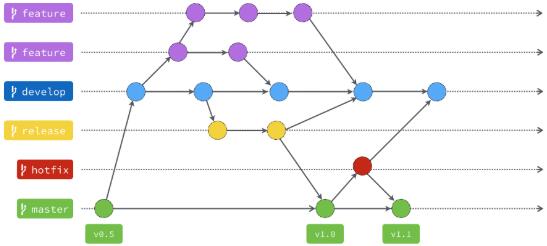
（2）.develop。

（3）.hotfix。

（4）.release。

（5）.feature。

图示如下：



**二.分支种类作用介绍：**

**1.master分支：**

此分支通常用来存放项目的稳定版本，主要特点如下：

（1）.内容来源主要是分支合并过来，不推荐开发者直接commit提交。

（2）.由于master分支可以是稳定版本，可以随时上线，所以通常版本标签都是打在master分支各个提交之上。

关于标签的用法可以参阅[Git tag标签用法详解](http://www.softwhy.com/article-8541-1.html)一章节。

**2.develop分支：**

develop开发分支通常和feature特征分支配合使用，它是所有feature特征分支的基础。

当在feature分支中测试完新开发的功能后，可以将其合并到develop分支。

**3.hotfix分支：**

当出现紧急问题，比如master线上分支出现代码问题，可以在此分支中进行修补，修补之后然后再合并到master分支。

特别说明：在合并到master主分支的同事，还要合并一份到develop分支，否则之后将develop合并到master时候，可能会导致修复的问题复现。

**4.release分支：**

当develop分支开发到自认为足够稳定的状态，将此分支合并到release分支（在此分支进行上线前的最后测试）。

最后测试完成后，再合并到master分支和develop分支。

合并到master分支是非常好理解的，因为要上线运行。合并到develop是因为release分支后续可能还会发现问题，所以要将与develop分支同步，以防止以后develop再合并到release分支出现问题。

**5.feature分支：**

此分支作用其在介绍develop分支的时候已经涉及，在feature进行新功能的开发，开发完成后再合并到develop。