一个团队多人开发如何管理git

在开发新的功能，如果还未完成提交会影响其他开发人员，所以每个人员可以创建一个属于自己的分支（每个开发人员拥有一个develop和一个从develop分支上创建出来的任务分支），功能开发完成后在合并到develop分支上。

分支合并冲突，多人开发容易发生冲突，解决冲突有自动解决(resolve)，手动解决（pull抓下来，再比较不同，手动合并，提交到本地仓库，再push到远程仓库）

分支管理策略：

主分支(master)

开发分支(develop)

功能分支(feature)

预发布分支(release)

bug分支

主分支master（自动建立）

提供给用户使用的正式版本，都在这个主分支上发布

开发分支develop

日常开发分支：这个分支用来生成代码的最新代码版本。如果想正式对外发布，就在master主分支上对develop分支进行合并。

功能分支feature

开发某种特定的功能（一般是一个任务或者问题，通常对应一个开发人员），从develop分支上面分出来，开发完成后，要合并入develop。

预发布分支release

发布正式版本之前（合并到master分支之前），我们可能需要一个预发布的版本进行测试。预发布分支是从develop分支上面分出来的，预发布结束后，需要合并到develop和master分支。然后删除该分支。

bug分支。

正式发布后，出现的bug,需要紧急修复，这时创建出一个bug分支，进行bug修复，修复完，合并到master和develop分支。然后删除该分支。

那些分支要被推送到远程？

master分支是主分支，要时刻与远程同步，一般发布最新版本就用master分支。

develop分支是开发分支，团队所有人在该分支上进行开发，所以也需要与远程同步。

bug分支一般用在本地使用修复bug,不需要推送到远程仓库中。

Feature分支，如果多人开发同一个功能，需要推送，一个人开发则不需要。

Release分支是系统管理，与开发人员无关。