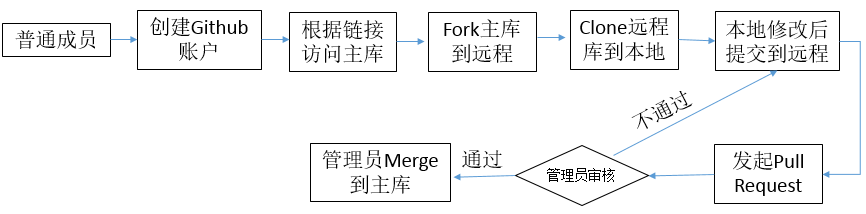
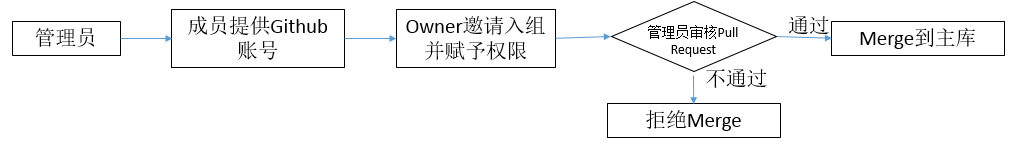
**Github组织协作流程：**



**普通成员：**

1. 创建Github账户：参见Github使用手册P2。
2. 根据链接访问主库：<https://github.com/PriTest/Primetonkb.git>。
3. Fork主库到远程仓库：参见Github使用手册P12。
4. Clone远程库到本地：参见Github使用手册P13/P19。
5. 本地修改后提交到远程：增加要提交的修改后Push到远程仓库，Push之前需要先同步本地库与远程仓库，以免因远程库存在变更而发生冲突。参见Github使用手册P14。
6. 发起Pull Request：远程仓库更新后，可向主库发起Pull Request，参见Github组织协作工具P7。
7. 管理员审核：具有管理权限的人员操作，经审核Pull Request不通过可拒绝Merge，打回普通成员处继续修改提交。
8. 管理员Merge到主库：审核Pull Request通过后管理员可把普通成员提交的变更Merge到主库。



**管理员：**

1. 成员提供Github账号：Owner通过Github账号对管理员发出邀请，管理员同意即可。
2. Owner邀请入组并赋予权限：权限分两类：Member或Owner，其中Member对组织其他成员可见、新建团队以及可被赋予更多权限；Owner享有组织最高权限，可增删仓库、成员、团队等。
3. 管理员审核Pull Request：通过后即可将变更与主库Merge，如不通过拒绝即可，参见Github组织协作工具P9。

**注意：**

每个开发者的仓库都有自己的分支路线，而这些分支路线会通过代码汇总映射到源仓库中去。

以**Primetonkb**为例，存在于整个项目的开始、开发、迭代、终止过程中的永久性分支只有两个master和develop。

**master**：主分支从项目一开始便存在，它用于存放经过评审，已经完全稳定的文件。所以，应该随时保持master仓库代码的清洁和稳定，确保入库之前是通过完全reivew的。master分支是所有分支中最不活跃的，大概每个月或每两个月更新一次，每一次master更新的时候都应该用git打上tag，说明有新版本发布了。

**dev**elop：开发分支，一开始从master分支中分离出来，用于成员存放基本稳定代码。之前说过，每个成员的仓库相当于源仓库的一个镜像，每个成员自己的仓库上也有master和develop。成员把新文件做好以后，是存放到自己的develop中，当整理完以后，可以向管理者/评审者发起一个pull request，请求把自己仓库的develop分支合并到源仓库的develop中。

所有成员的新提交文件会在源仓库的develop分支中进行汇总，当develop中的文件经过评审，已经逐渐趋于稳定了。这时候，我们就可以把develop分支合并到master分支中，发布一个最新版本。

任何人不应该向master直接进行无意义的合并、提交操作。正常情况下，master只应该接受develop的合并，也就是说，master所有更新应该源于合并develop的更新。