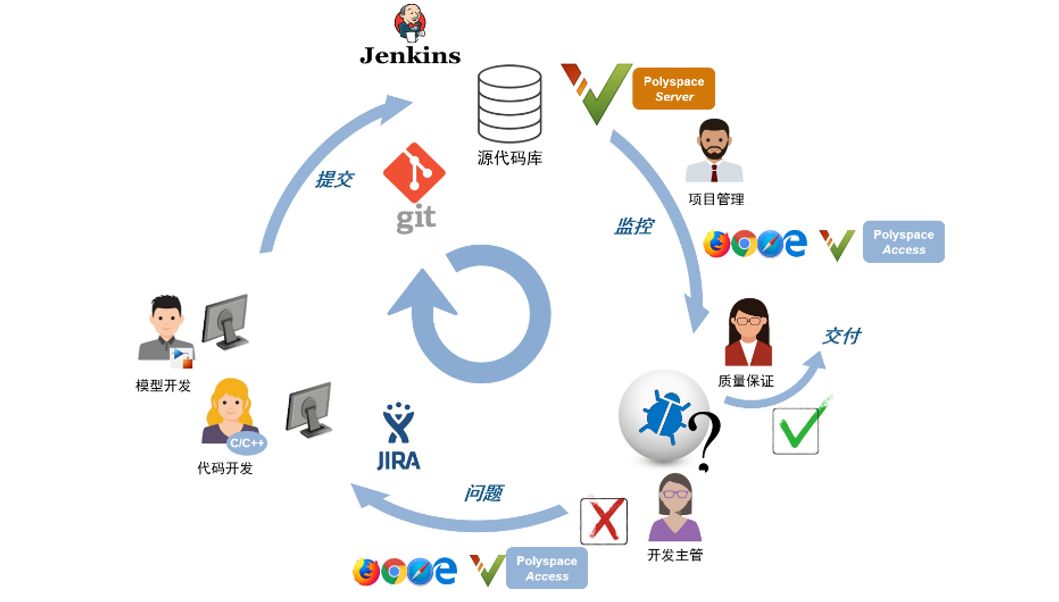
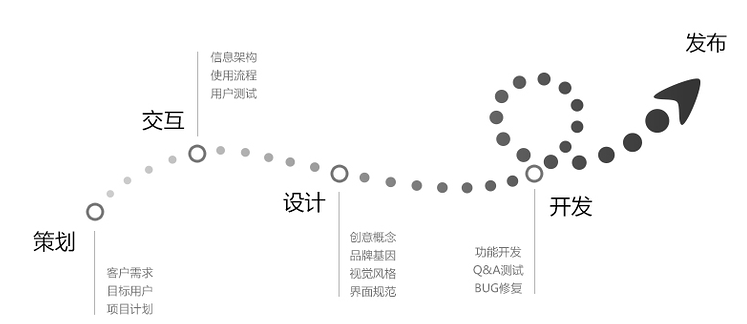
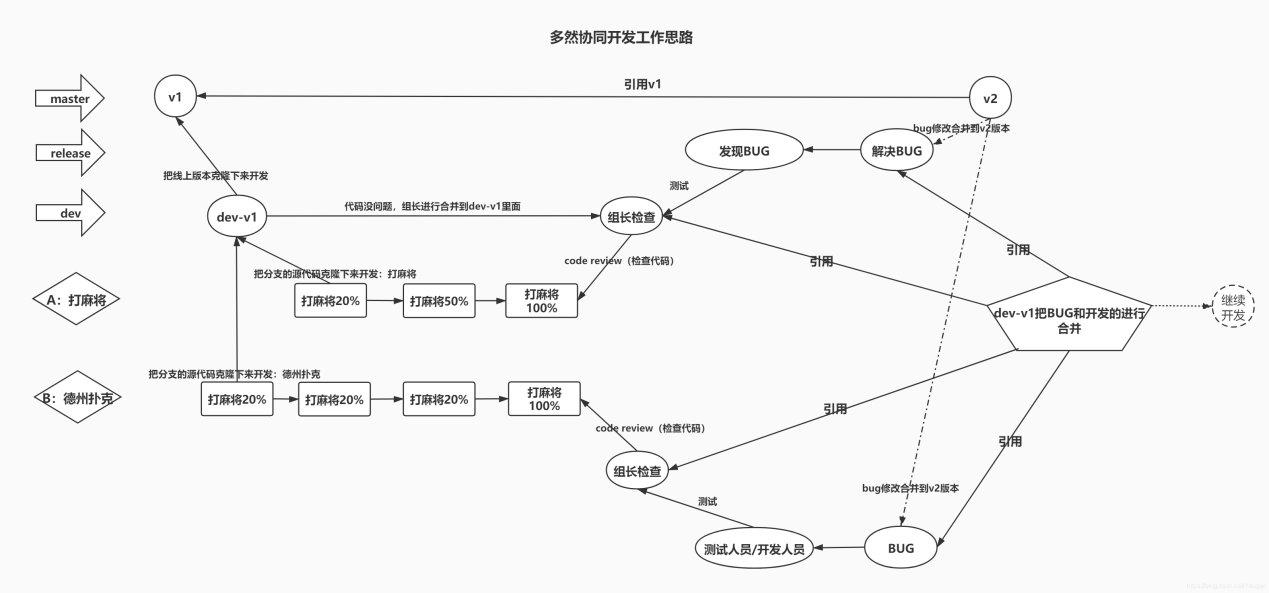
[](http://www.elecfans.com/d/1020007.html)





**本地目录**

mkdir xxxx 项目目录

mkdir xxxx 目录开发目录

* 1
* 2

**基本版本**

git init

touch app.py # 开发了一个功能

git add .

git commit -m "基本版本"

* 1
* 2
* 3
* 4

**创建组织**

组织的名字

邮箱

创建第一个组织下的仓库

* 1
* 2
* 3

**推送到组织下面的仓库**

git remote add origin xxxxx

git push origin master

* 1
* 2

**给版本打标签**

版本号 xxxxxxxxxxxxxxxxxx => v1

git tag -a v1 -m "xxxx"

git push origin --tags # 将标签推送上去

* 1
* 2
* 3

### 多人协同开发之邀请成员

**创建dev分支,并且推送到GitHub**

git branch dev

git checkout dev

git push origin dev

* 1
* 2
* 3

**到项目中邀请成员，给写权限**

**成员第一次来做开发**

mkdir xxx

git clone xxxxxxx

git checkout -b ddz # 创建ddz 并且切换到ddz分支

touch ddz.py

git add .

git commit -m "xxx"

git push origin ddz

* 1
* 2
* 3
* 4
* 5
* 6
* 7

**多人协同开发之代码review**

git checkout -b release #创建release分支

* 1

也可以测试人员 拉取到本地进行测试

### Git补充

* **配置文件**
* 项目配置文件
* .git/config
* 全局配置文件
* ~/.gitconfig
* 系统配置文件
* /etc/gitconfig
  + 1
  + 2
  + 3
  + 4
  + 5
  + 6
* **免密登录**
* URL
* https://用户名:密码@github.com/WuPeiqi/xxx.git
* SSH
  + 1
  + 2
  + 3
  + 4

### Git忽略文件

**创建 .gitignore文件**

\*.h # 点.h结尾的文件都不管理

!a.h # 取反

files/ # 目录下的文件 不管理

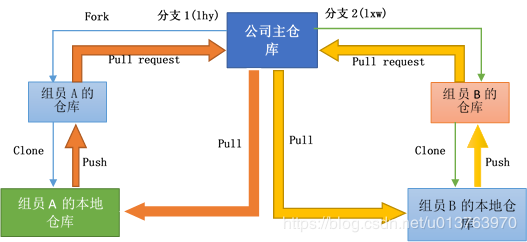
* 1
* 2
* 3

### 项目的文件

issue， 文档以及任务管理

wiki, 项目文件

**这一篇文章主要是学习Gitee协作开发的使用。这里我们用了一个企业账号来和两个社区账号进行演示。**



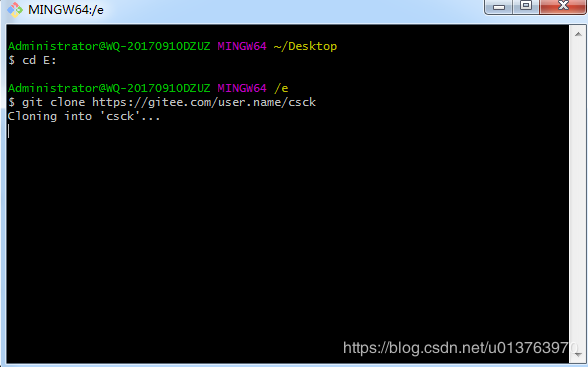
**1.首先，企业账号下新建一个名称为协作开发测试仓库的主仓库，同时在仓库下新建两个分支lhy和lxw，同时仓库成员添加小组成员A和B。lhy用来存放组员A开发的代码部分，lxw用来存放组员B开发的代码部分。**

  
组员A、B此时在各自的账户下可以看到这个仓库。  
下面我们以组员A进行操作，组员B操作同理。

**2.组员A首先进入这个公司的仓库，然后右上角Fork该仓库中到自己的仓库中。这里需要主仓库允许被Fork功能勾上。**


**3.组员A从自己的仓库中clone项目到本地仓库。**

先复制仓库地址。  
我们将本地仓库的地址默认为E盘：  
运行Git Bash，cd到相应位置，开始clone远程仓库：  
输入git clone 复制的地址，完成后在E盘下出现了csck文件夹。  
  
以上就建立完了本地仓库与主仓库和自己的仓库的联系，接下来主要是协作开发的流程

**开发流程**

首先进入项目目录，查看本地分支及所有分支，通过git命令我们可以看到仓库下所有分支：

Administrator@WQ-20170910DZUZ MINGW64 /e/csck (master)

$ cd e:csck //进入项目目录

Administrator@WQ-20170910DZUZ MINGW64 /e/csck (master)

$ git branch -a //查看本地和远程所有分支

\* master

remotes/origin/HEAD -> origin/master

remotes/origin/lhy

remotes/origin/lxw

remotes/origin/master

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

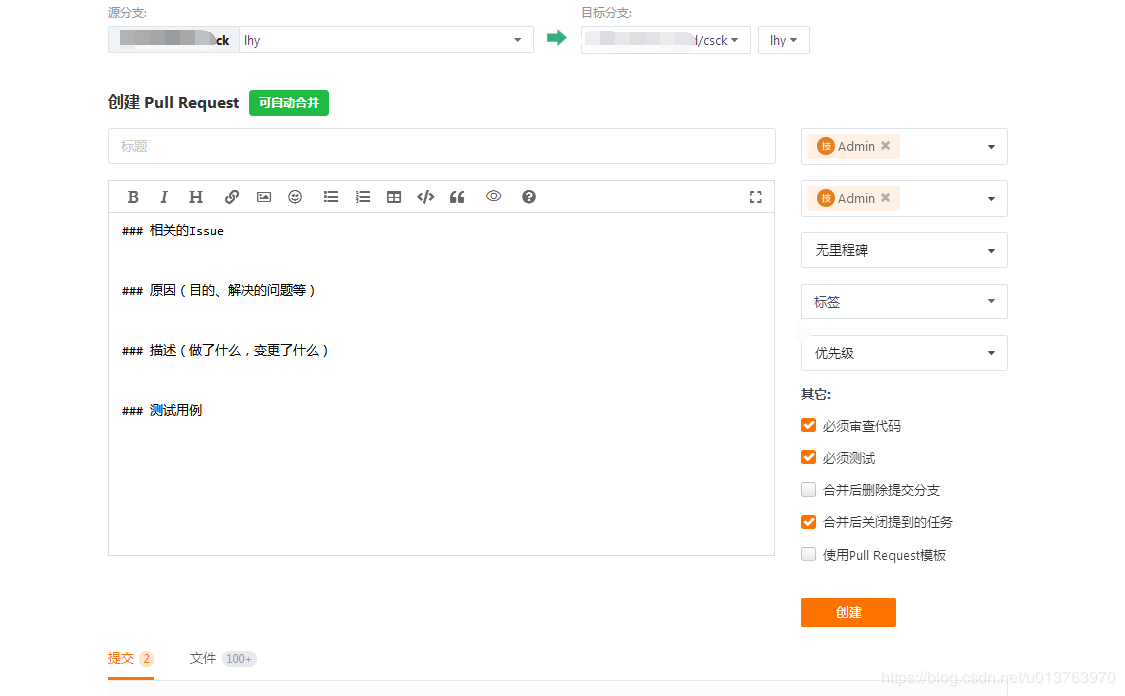
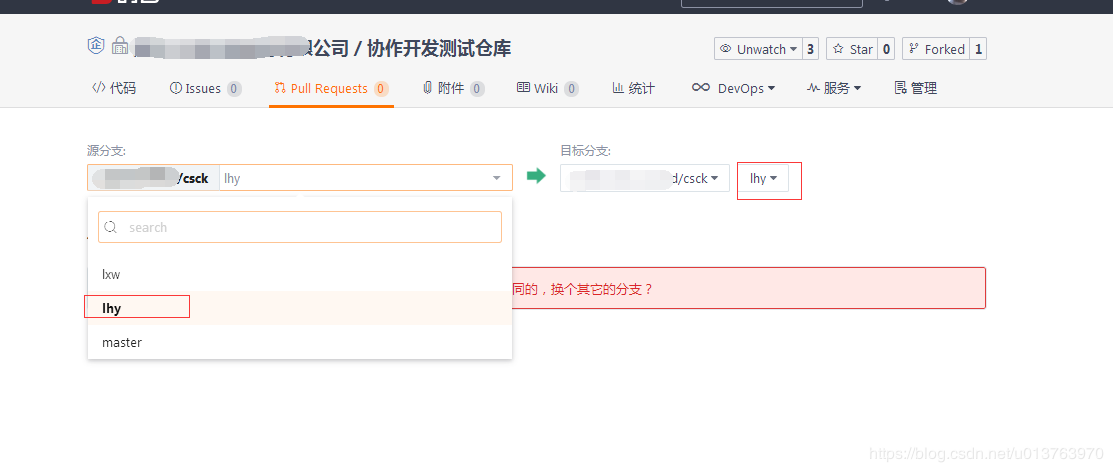
这里我们可以看到当前分支为master，下面组员A切换分支到lhy下进行开发工作：

$ git checkout lhy

Switched to a new branch 'lhy'

Branch 'lhy' set up to track remote branch 'lhy' from 'origin'.

* 1
* 2
* 3

A开发完成后push代码到组员A的仓库，如何push可以参照上一篇文章，然后新建一个Pull Request，请求将自己仓库下的lhy分支合并到公司主仓库下的lhy分支：  
管理员及测试人员审核测试就可以合并分支即可。  
