svn 集中式，git分布式

**注：**

**在实际使用分布式版本控制系统的时候，其实很少在两人之间的电脑上推送版本库的修改，因为可能你们俩不在一个局域网内，两台电脑互相访问不了，也可能今天你的同事病了，他的电脑压根没有开机。因此，分布式版本控制系统通常也有一台充当“中央服务器”的电脑，但这个服务器的作用仅仅是用来方便“交换”大家的修改，没有它大家也一样干活，只是交换修改不方便而已。**

**微软自己也有一个集中式版本控制系统叫VSS，集成在Visual Studio中。由于其反人类的设计，连微软自己都不好意思用了。**

**安装git:**

如果你碰巧用Debian或Ubuntu Linux，通过一条**sudo apt-get install git就可以直接完成Git的安装，非常简单。**

如果是其他Linux版本，可以直接通过源码安装。先从Git官网下载源码，然后解压，依次输入：**./config，make，sudo make install这几个命令安装就好了。**

msysgit是Windows版的Git，从**[https://git-for-windows.github.io](https://git-for-windows.github.io/)下载（网速慢的同学请移步[国内镜像](http://pan.baidu.com/s/1skFLrMt" \l "path=%252Fpub%252Fgit)），然后按默认选项安装即可。**

**安装完成后，还需要最后一步设置，在命令行输入：**

$ git config --global user.name "Your Name"

$ git config --global user.email "email@example.com"

注意**git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然也可以对某个仓库指定不同的用户名和Email地址。**

1.分支合并分为两种方式：merge（融合）

rebase（变基）

2.发生冲突的场景：

远端有人修改的部分和自己本地修改的部分相同时，在本地向远端同步的时候就会产生冲突。

比如：有人在远端的master分支修改了a文件的b部分，你在本地也修改了a文件的b部分，

两人修改的结果不一样，你本地向远端做同步就会产生冲突。

解决方式：就是人为的去统一意见，然后重新合并。

3.理解GitHub Flow（用于团队协作）

GitHub Flow是一个非常轻便的，基于分支的工作流。非常适合代码部署非常频繁的团队和项目。

步骤：

1）创建一个分支

2）添加新版本

3）开启一个Pull Request（拉取请求） --核心

4）讨论和代码审核

5）合并分支，然后部署

Collaborators（协作者）

4.fork开源项目

5.GitHub Issues(事务卡片)

6.GitHub Pages搭建网站