## Git

Git是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。

分布式相比于集中式的最大区别在于开发者可以提交到本地，每个开发者通过克隆（git clone），在本地机器上拷贝一个完整的Git仓库。

从一般开发者的角度来看，Git有以下功能：

1. 从服务器上克隆完整的Git仓库（包括代码和版本信息）到单机上。
2. 在自己的机器上根据不同的开发目的，创建分支，修改代码。
3. 在单机上自己创建的分支上提交代码。
4. 在单机上合并分支。
5. 把服务器上最新版的代码fetch下来，然后跟自己的主分支合并。
6. 生成补丁（patch），把补丁发送给主开发者。
7. 看主开发者的反馈，如果主开发者发现两个一般开发者之间有冲突（他们之间可以合作解决的冲突），就会要求他们先解决冲突，然后再由其中一个人提交。如果主开发者可以自己解决，或者没有冲突，就通过。
8. 一般开发者之间解决冲突的方法，开发者之间可以使用pull 命令解决冲突，解决完冲突之后再向主开发者提交补丁。

从主开发者的角度（假设主开发者不用开发代码）看，git有以下功能：

1. 查看邮件或者通过其它方式查看一般开发者的提交状态。
2. 打上补丁，解决冲突（可以自己解决，也可以要求开发者之间解决以后再重新提交，如果是开源项目，还要决定哪些补丁有用，哪些不用）。
3. 向公共服务器提交结果，然后通知所有开发人员。

优点：

* + 1. 适合分布式开发，强调个体。
    2. 公共服务器压力和数据量都不会太大。
    3. 速度快、灵活。
    4. 任意两个开发者之间可以很容易的解决冲突。
    5. 离线工作。

缺点：

* + 1. 学习周期相对而言比较长。
    2. 不符合常规思维。
    3. 代码保密性差，一旦开发者把整个库克隆下来就可以完全公开所有代码和版本信息。

## Visual SourceSafe

Microsoft Visual SourceSafe是美国微软公司出品的版本控制系统，简称VSS。

Visual SourceSafe 是一种版本控制系统产品，它提供了还原点和并行协作功能，从而使应用程序开发组织能够同时处理软件的多个版本。该版本控制系统引入了签入和签出模型，按照该模型，单个开发人员可以签出文件，进行修改，然后重新签入该文件。当文件被签出后，其他开发人员通常无法对该文件进行更改。通过源代码管理系统，开发人员还能够回滚或撤消任何随后产生问题的更改。

作为一种版本控制系统，Visual SourceSafe 能够：

* 防止用户无意中丢失文件。
* 允许回溯到以前版本的文件。
* 允许分支、共享、合并和管理文件版本。
* 跟踪整个项目的版本。
* 跟踪模块化代码（一个由多个项目重用或共享的文件）

VSS的三大特点:

1. 支持大型项目和分布式团队. 用 Visual Studio 从全球任何可以进行 HTTP 或 HTTPS Web 访问的地方连接到 Visual SourceSafe 数据库。
2. 增强的自定义开发团队可以利用新的自动化和可扩展性功能，自动执行其过程以及扩展 Visual SourceSafe 的功能.
3. Web 开发和工作效率的提高. Visual SourceSafe 的 Visual Studio 源代码管理集成进行了更新，以便提高开发人员的工作效率，尤其是在开发 Web 服务和 Web 站点时