## Git使用策略

**Fetch origin（同步远程仓库）**：所有组员在Git客户端工具上进行工作之前，必须先执行Fetch origin操作，保证本地仓库文件与远程仓库文件一致，以免出现覆盖他人工作或遗漏之前文件的错误。

**Branch（分支）**：在执行整个项目的不同阶段上，会通过branch来进行版本控制。由配置管理员进行每个阶段的branch创建，命名为当前阶段任务英文缩写。

**Branch-master（主分支）**：整个项目的主分支，最初建立的版本，后期会逐渐在此分支或新分支上进行项目编写。

**Branch-XXX（新版本分支）**：整个项目每进行到一个新的阶段，都会创建一个与之对应的新分支，用于进行版本控制。

**Commit（提交）**：每次对项目中的文档进行修改，完成后都需要进行提交，保证summary不为空，确定本地文档已修改。

**Push（上传远程仓库）**：对文档修改完成且提交之后，需要执行push操作，保证远程仓库取得本地仓库文档的修改。

**受控文档**：需要交付给顾客的产品项目，统一由文档整合员及项目经理修改。

**非受控文档**：不需要交付给顾客的产品项目。其下分别建有各个组员自己的专属目录，组员完成的自己所属的部分文档及自己专门的文档都存在此目录下。由小组成员分工完成的文档最后由整合员统一，交于项目经理审批，完成后存入受控文档中对应的项目目录下。

注意事项！！！：使用github客户端前请严格执行Fetch origin（同步远程仓库）操作，并确定当前所处分支为当前工作分支，切记！！！