**90分钟整体掌握GIT的使用技巧**

**课程概要点：**

* GIT体系概述
* GIT常用命令讲解与演示
* 版本迭代分支管理办法

**讲师介绍：**



## **一、GIT体系概述**

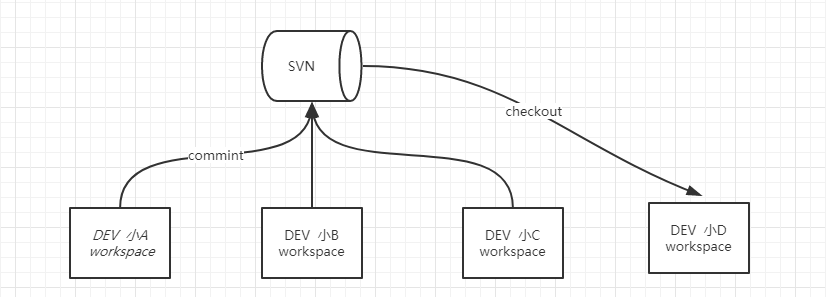
### **提问：**

大家公司是用什么工具来管理代码版本？

有没有用过Git 来管理？

GIT和SVN有什么区别呢？

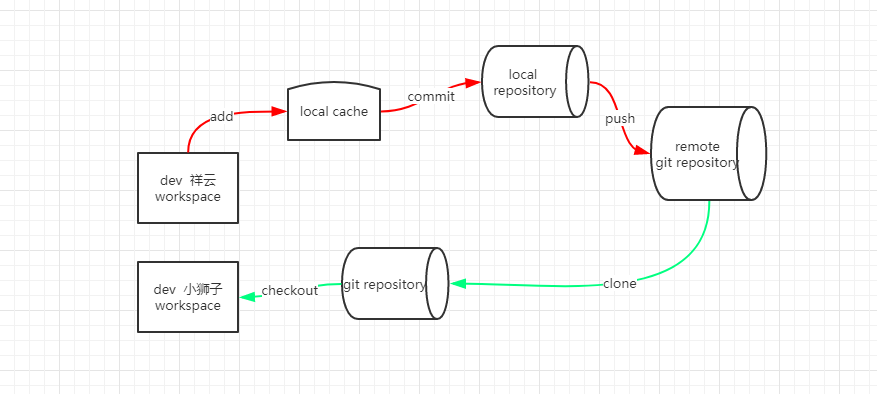
### **SVN 使用流程**



**主要步骤：**

提交 、checkout

### **GIT 使用流程**



主要步骤：

添加(add) 、提交(commit)、远程推送(push)、远程克隆(clone)、远程更新(pull)

其它命令：branch、tag、remote、checkout、merge、log、status 、fetch、rebase等

**GIT 与 svn 主要区别：**

1. 基于本地进行完整的版本管理，不强依懒远程仓库。
2. GIT把内容按元数据方式存储，而SVN是按文件。
3. GIT分支和SVN的分支不同。

## **二、GIT常用命令讲解与演示**

**安装git 客户端安装**

官方客户端： [httpsd://git-scm.com/downloads](https://git-scm.com/downloads)

其它客户端：https://tortoisegit.org/download/

branch(分支)

tag(标签)

remote(远程仓库)

**完整模拟从项目添加到push 过程**

创建项目

初始化git仓库

提交文件

远程关联

push 至远程仓库

**本地初始化GIT 仓库:**

#基于远程仓库克隆至本地

git clone <remote\_url>

#当前目录初始化为git 本地仓库

git init <directory>

基于mvn 模板创建项目

mvn archetype:generate

本地添加

#添加指定文件至暂存区

git add <fileName>

#添加指定目录至暂存区

git add <directory>

#添加所有

git add -A

#将指定目录及子目录移除出暂存区

git rm --cached target -r

#添加勿略配置文件 .gitignore

本地提交

#提交至本地仓库

git commit -m '提交评论'

#快捷提交至本地仓库

git commit -Am '快添加与提交'

分支管理

#查看当前分支

git branch [-avv]

#基于当前分支新建分支

git branch <branch name>

#基于提交新建分支

git branch <branch name> <commit id>

#切换分支

git checkout <branch name>

#合并分支

git merge <merge target>

#解决冲突，如果因冲突导致自动合并失败，此时 status 为mergeing 状态.

#需要手动修改后重新提交（commit）

远程配置

#查看远程配置

git remote [-v]

#添加远程地址

git remote add origin http:xxx.xxx

#删除远程地址

git remote remove origin

#上传新分支至远程

git branch origin -u

#将本地分支与远程建立关联

git branch --track --set-upstream-to=origin/test test

tag 管理

#查看当前

git tag

#创建分支

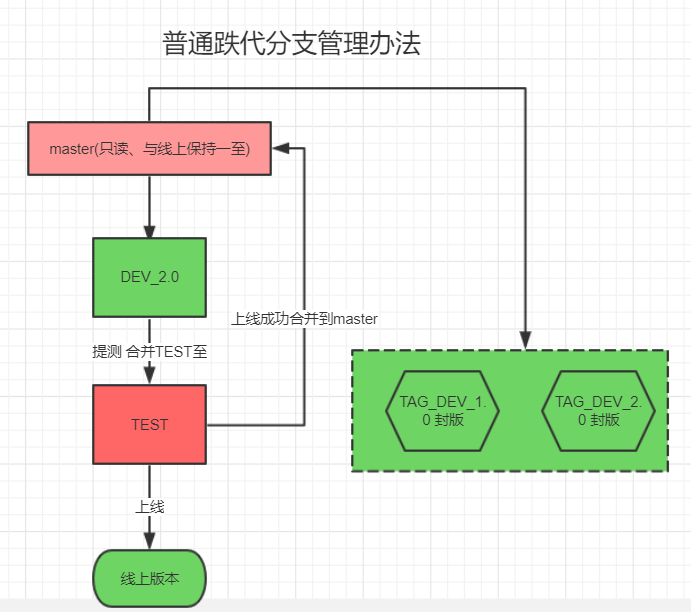
git tag <tag name> <branch name>

#删除分支

git tag -d <tag name>

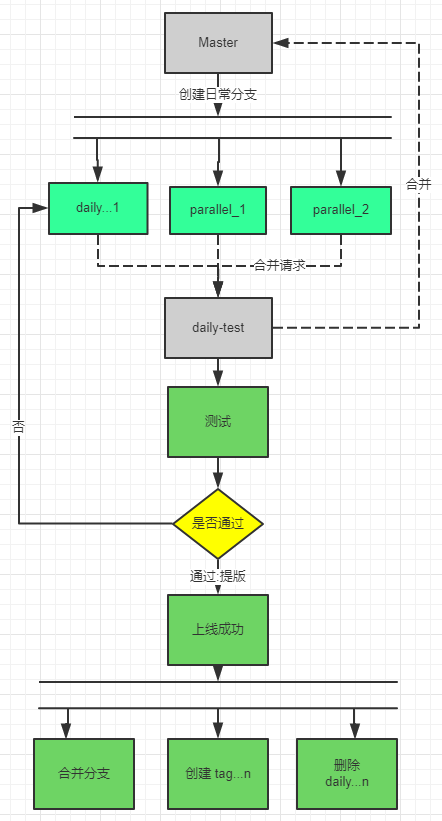
三、版本迭代分支管理办法

**普通迭代分支管理**

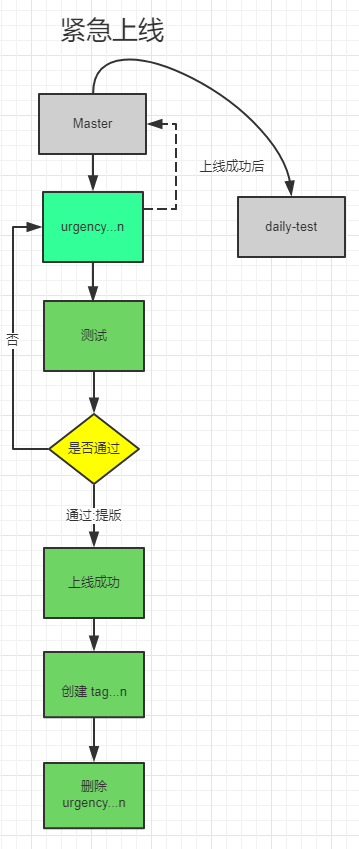


**高频迭代版本分支管理办法**

日常上线：



紧急上线：



**原则：**

**1、master 与 daily-test 为只读分支**

**2、所有的合并原则上不可逆向进行**

**3、日常或紧急分支上线之后就不可在使用（建议删除）**

**4、master 与线上完全同步**

**5、daily-test 版本必须大于等于master**