

# Ch.2 软件开发中的协作

2018 / 3

## § 2.0 即将面对的问题

2个人一起写代码怎么把代码合到一起？  
10个人呢？



## 其他问题

- 不小心删除了一个已经写了1000行的代码的文件，灾难吗？
- 开发机器坏了，怎么也打不开了，怎么办？
- 还是觉得3天前的写法比现在的更好，怎么倒回去？
- 代码中有问题，怎么定位由谁来解决？

## 更进一步

- 当前我要做哪些任务、写些什么代码，谁告诉我？
- 怎么告诉一个人他要做什么？
- 一个任务怎么确认已经做完了？
- 10个不同职能的人怎么知道工作进度？1000个人呢？



# 目录

- §2.1 认识团队
- §2.2 沟通
- §2.3 协作

# 一个团队中都有哪些人员？





# 项目参与者角度

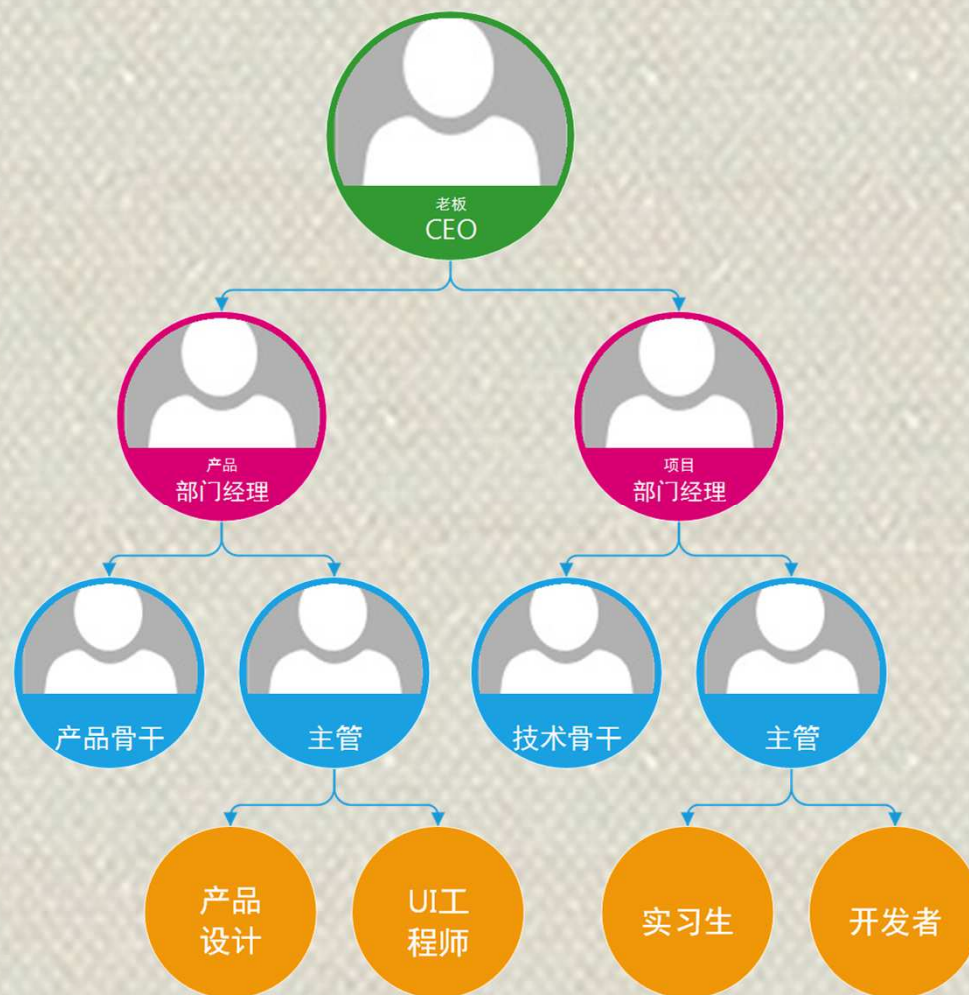
- 开发者
  - 写代码的
- 音乐音效制作人
  - 写歌曲的
- 翻译
  - 写外文的
- 文案
  - 写文字描述的
- 美工
  - 画画、做动画的

# 项目相关人员

- 项目经理
- 架构师
- 需求分析师
- 系统分析师
- 数据库工程师（DBA）
- 开发人员
- 测试人员
- 运维人员



# 组织结构角度



# 目录

- §2.1 认识团队
- §2.2 沟通
- §2.3 协作



# 沟通

- 为什么要沟通（内部、外部）
  - 团队管理需要（认同感、工作方式、团队文化）
  - 项目共识需要（反馈和上情下达）

# 项目中都有哪些沟通方式

- 会议
- 邮件
- 流程单
- 协作工具
- 工作交接
- .....



# 会议

- 立项会
- 评审会
- 项目节点会（里程碑会议）

# 最有效的沟通方式

- 文档化 立字据
- 流程化 流水线



# 目录

- §2.1 认识团队
- §2.2 沟通
- §2.3 协作

## 回到问题

2个人一起写代码怎么把代码合到一起？  
10个人呢？



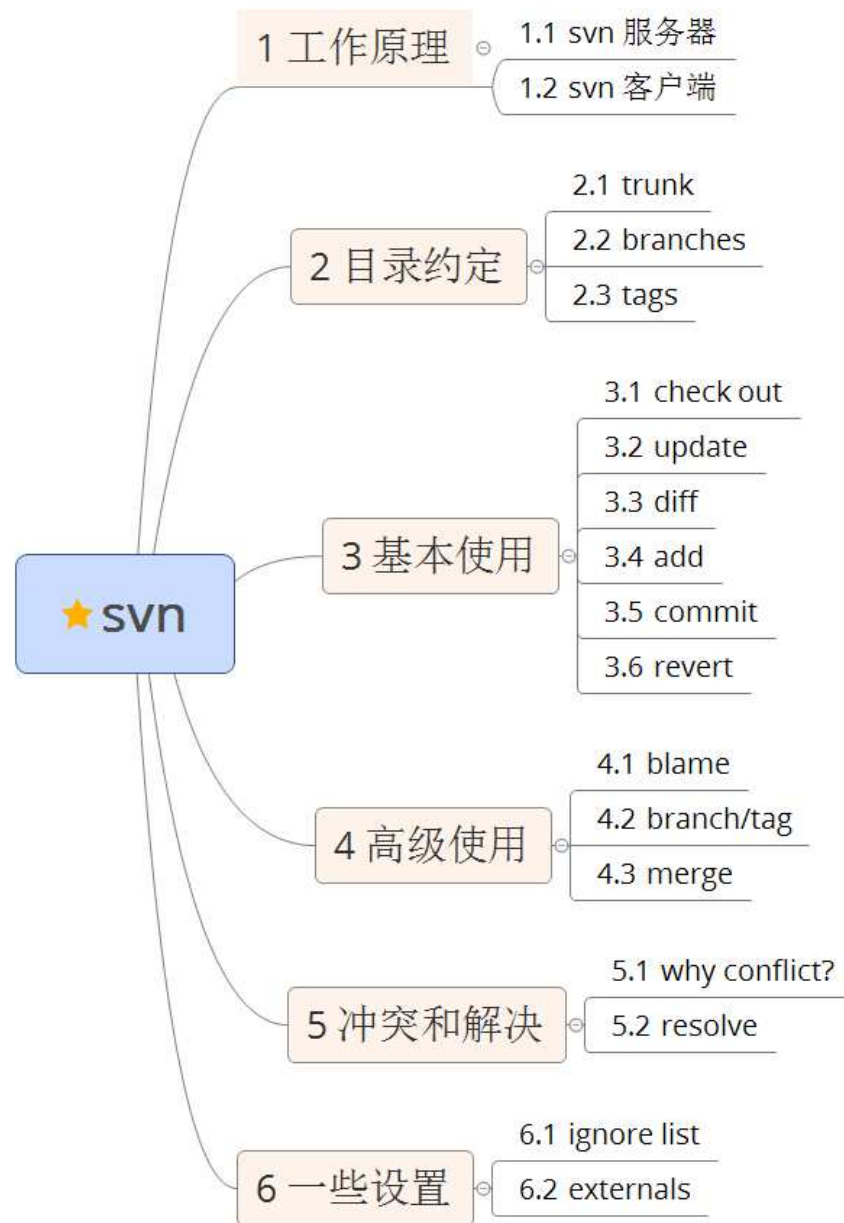
# 好的解决办法

- 使用版本控制系统
- SVN、Git、Mercurial、VSS 、CVS

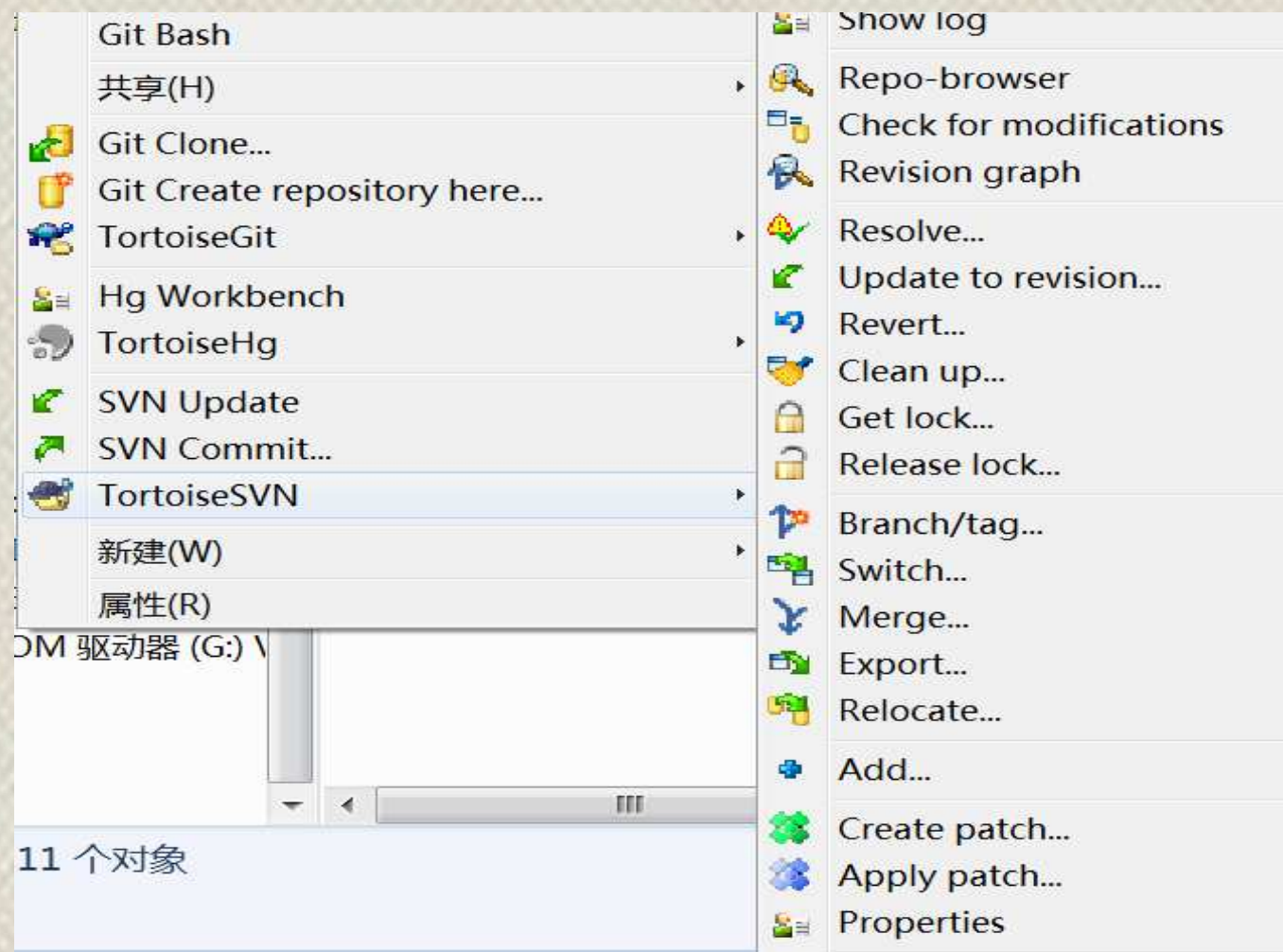
# **GIT/Subversion**

- **SVN : 本地工作目录和版本库**
- **Git :本地版本库和中央版本库**
- **推荐安装： tortoise SVN/GIT, SourceTree/**

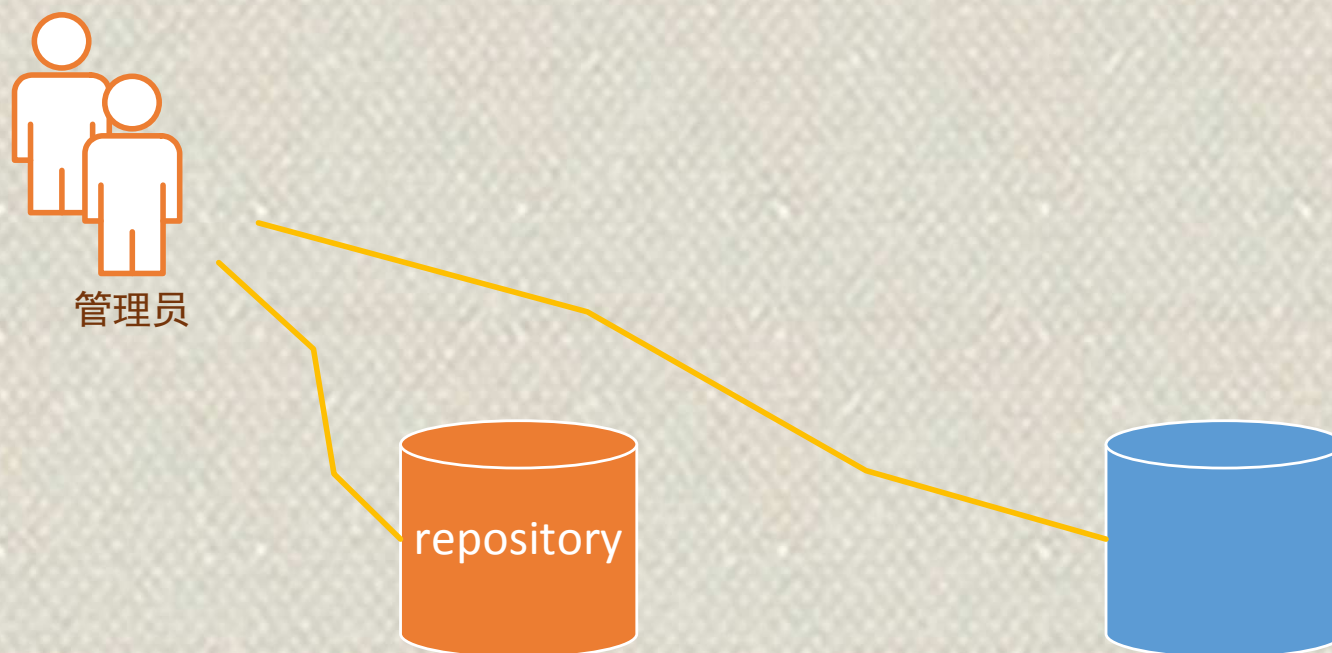


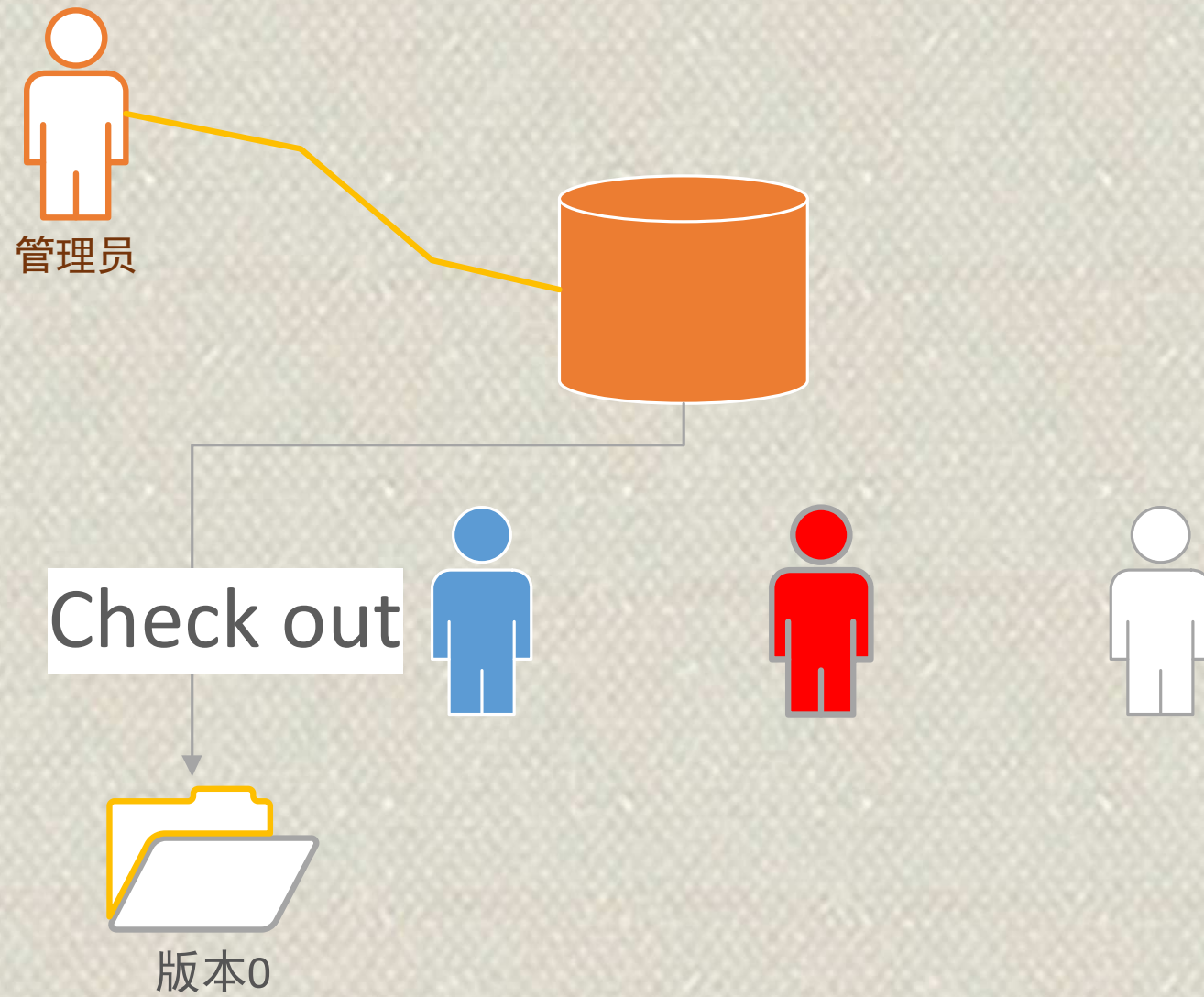


## 一些基本操作的说明

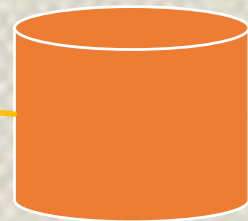










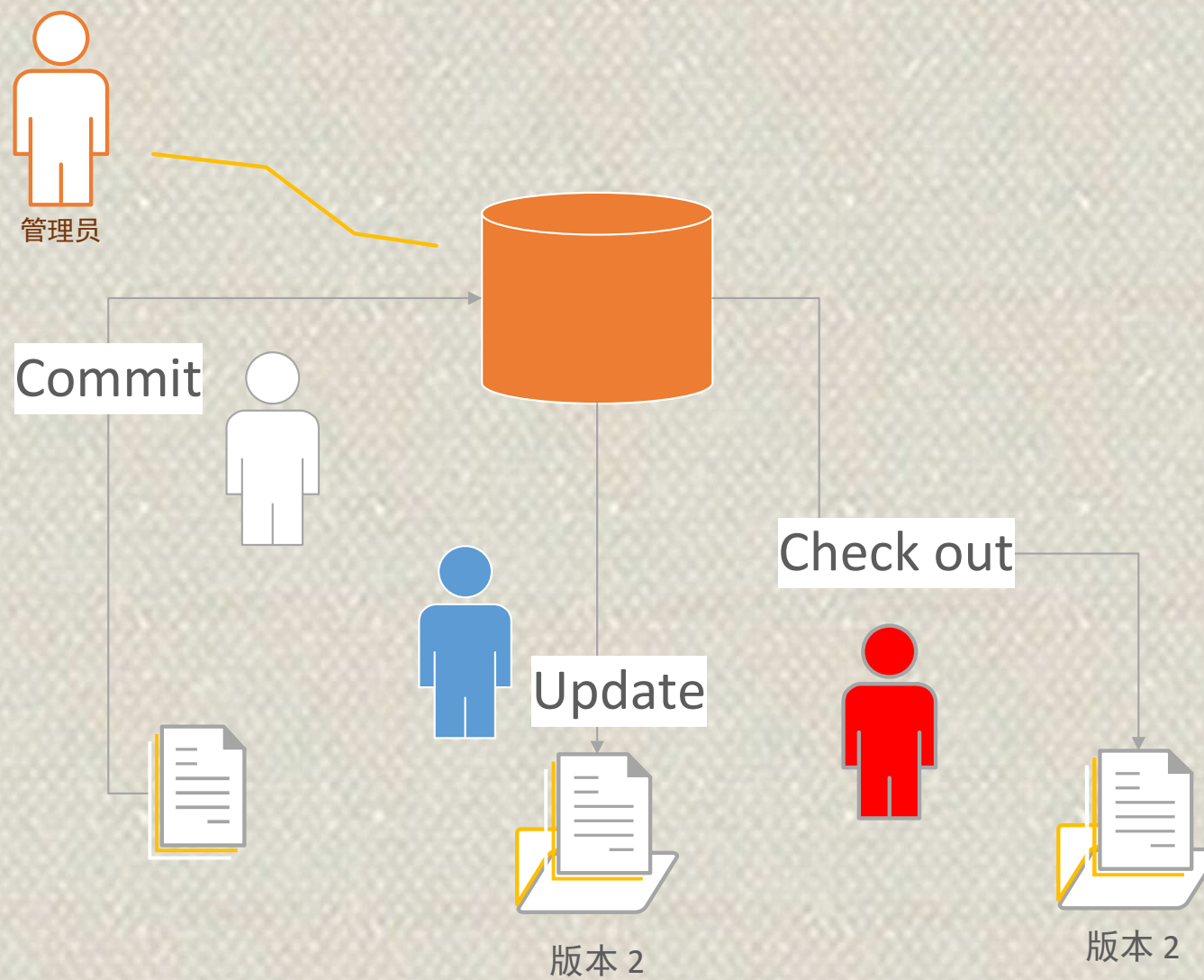


Commit

Check out

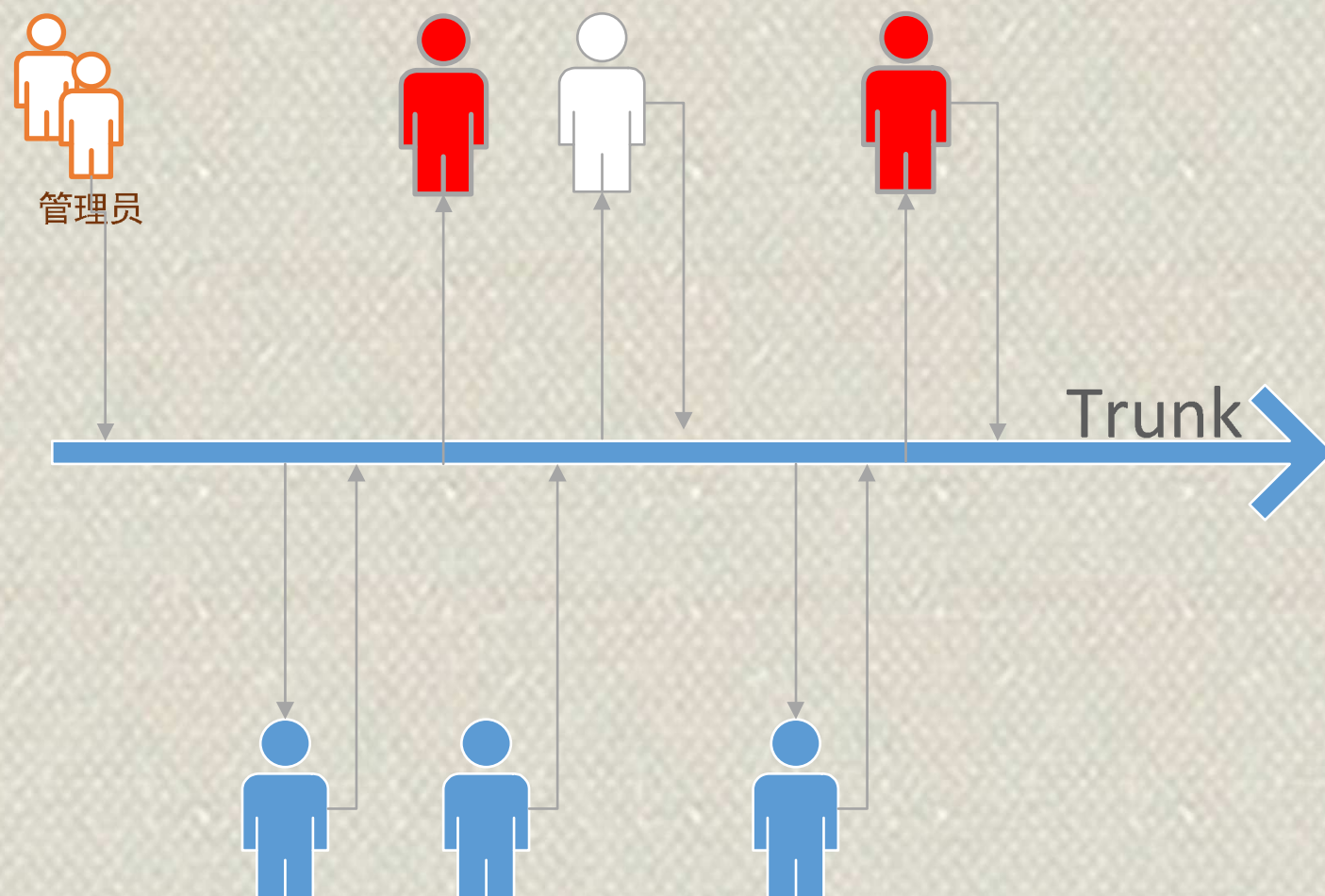


版本1





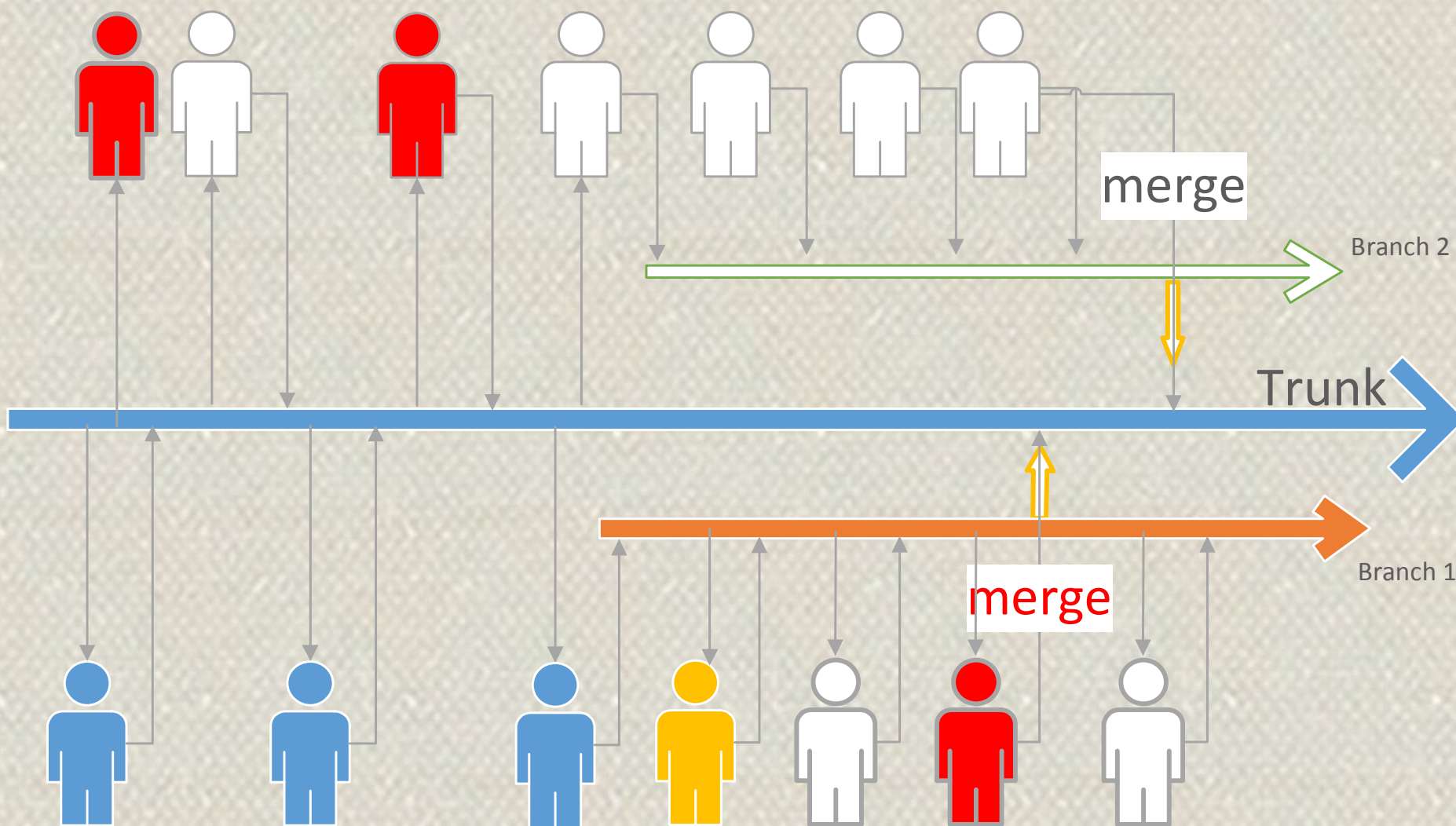
# 流程示例



# Branch和Tag的应用

- Branch 是做什么用的？
- Tag的作用
- Branch 的merge操作





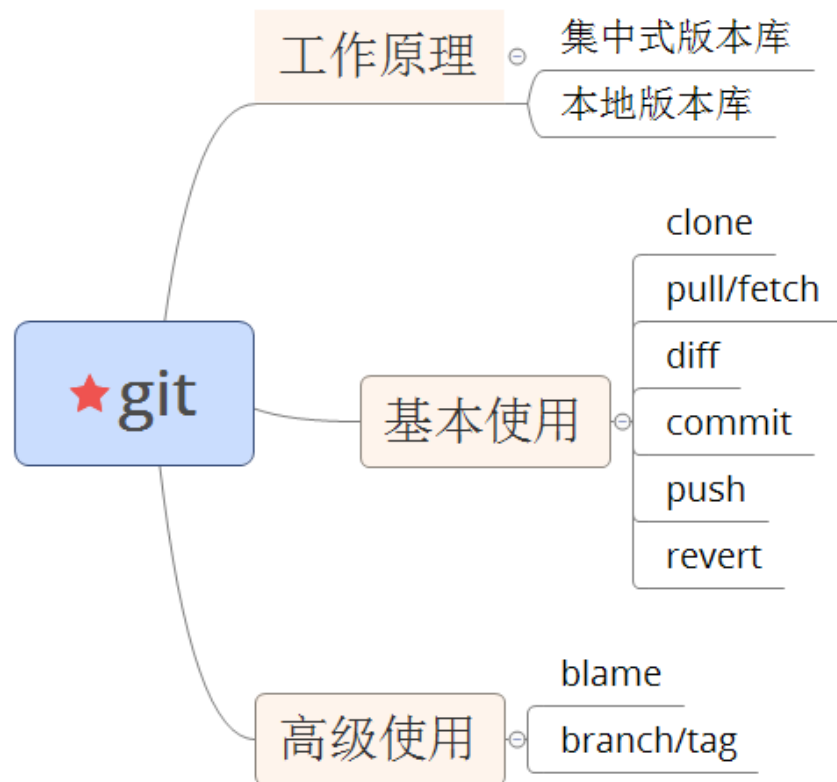
# 日志及update to Revision

- Diff
- Update
- Revert



# 新的问题

- 断网了，怎么办？

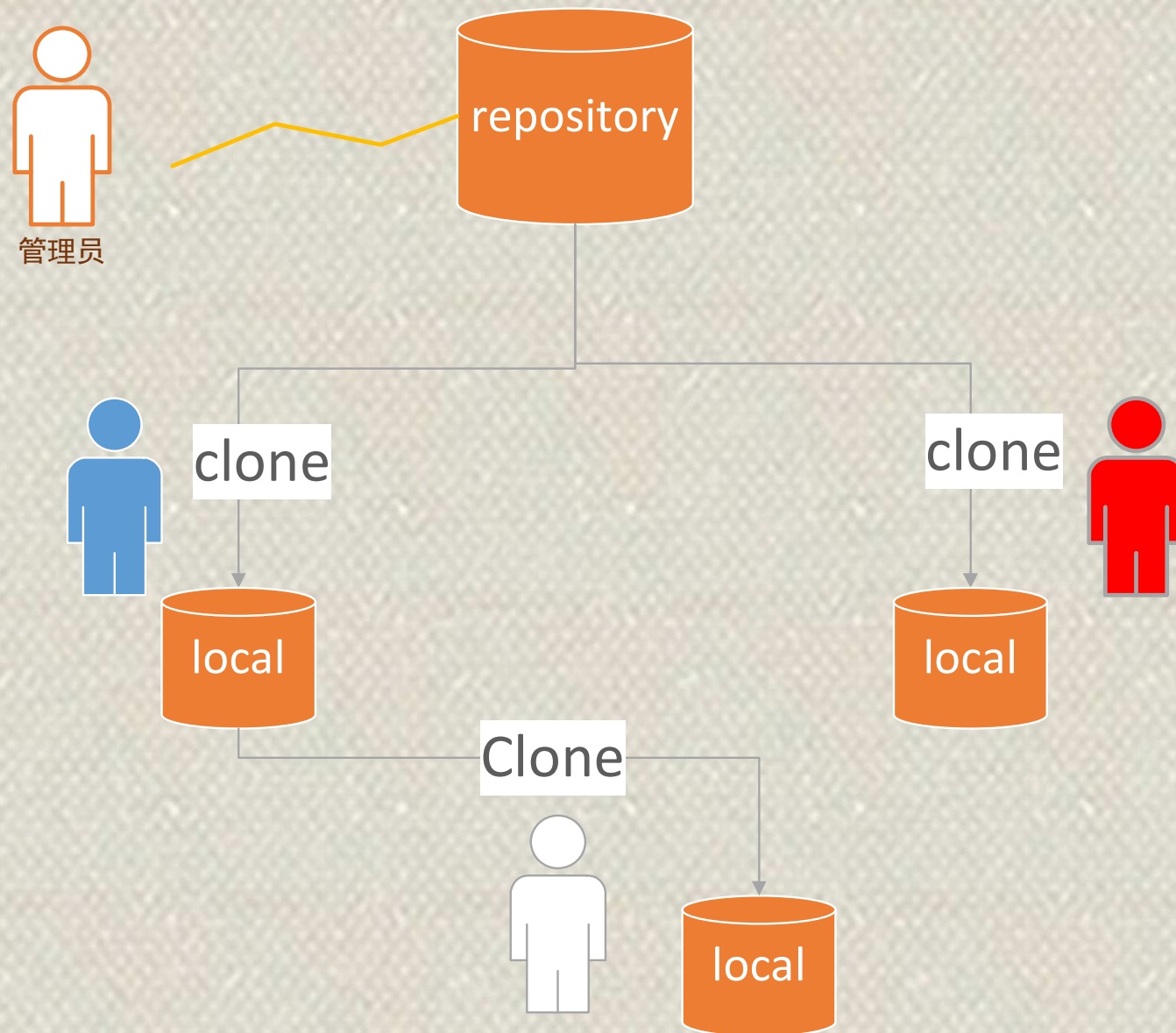




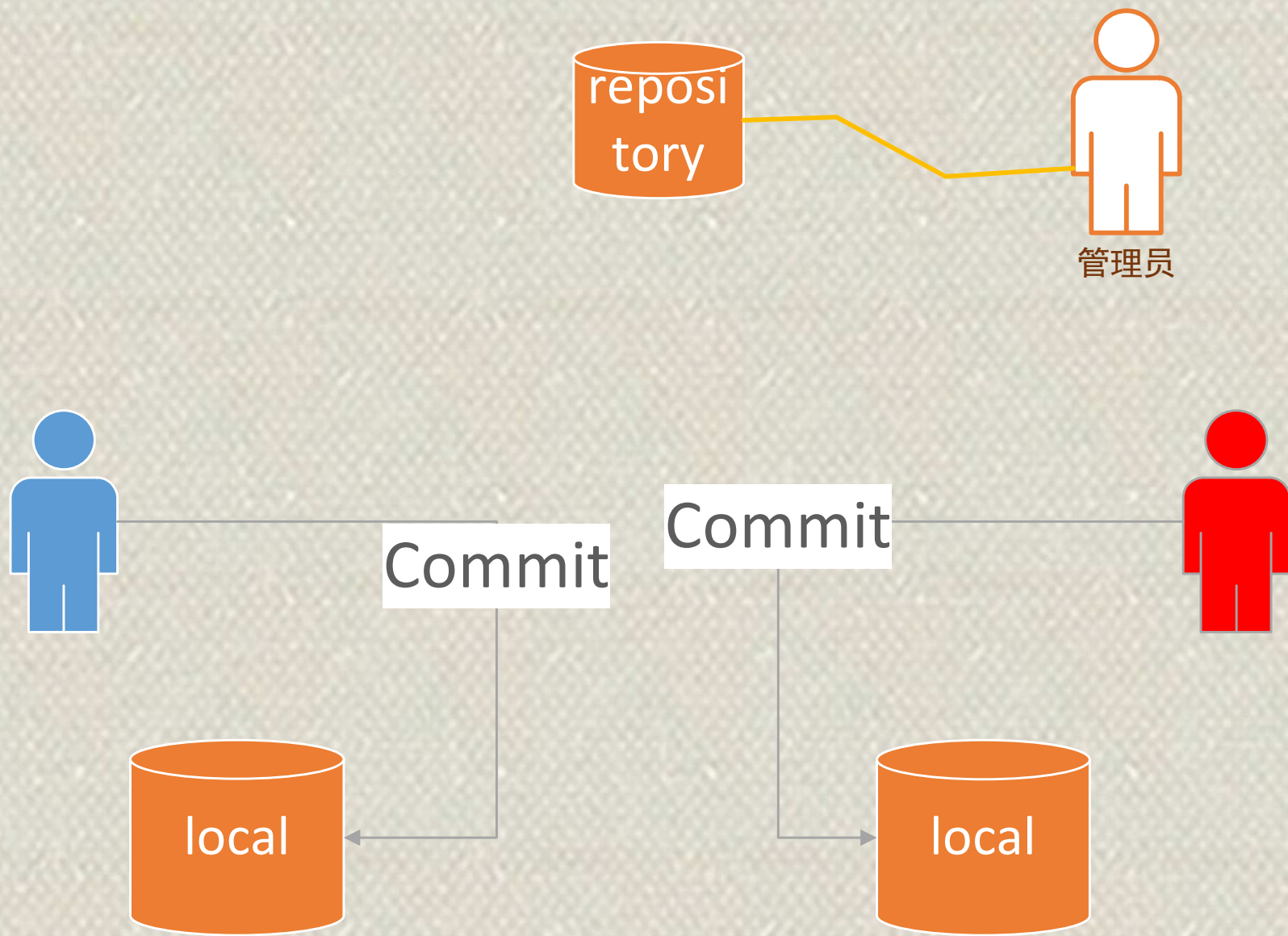
# 和Subversion的主要区别

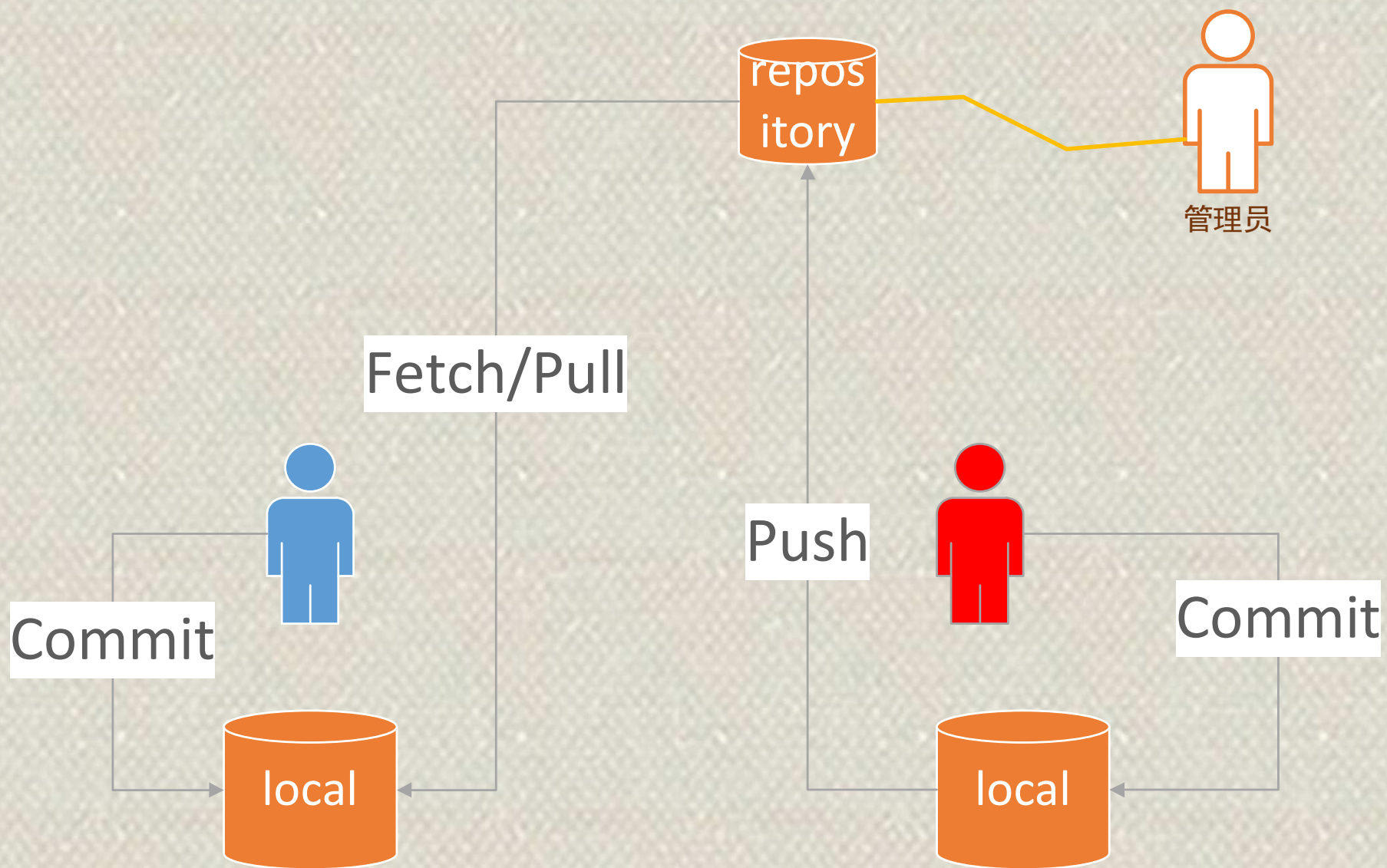
- 分布式，可以不断地进行clone
- git本没有公共服务器的概念

# GIT





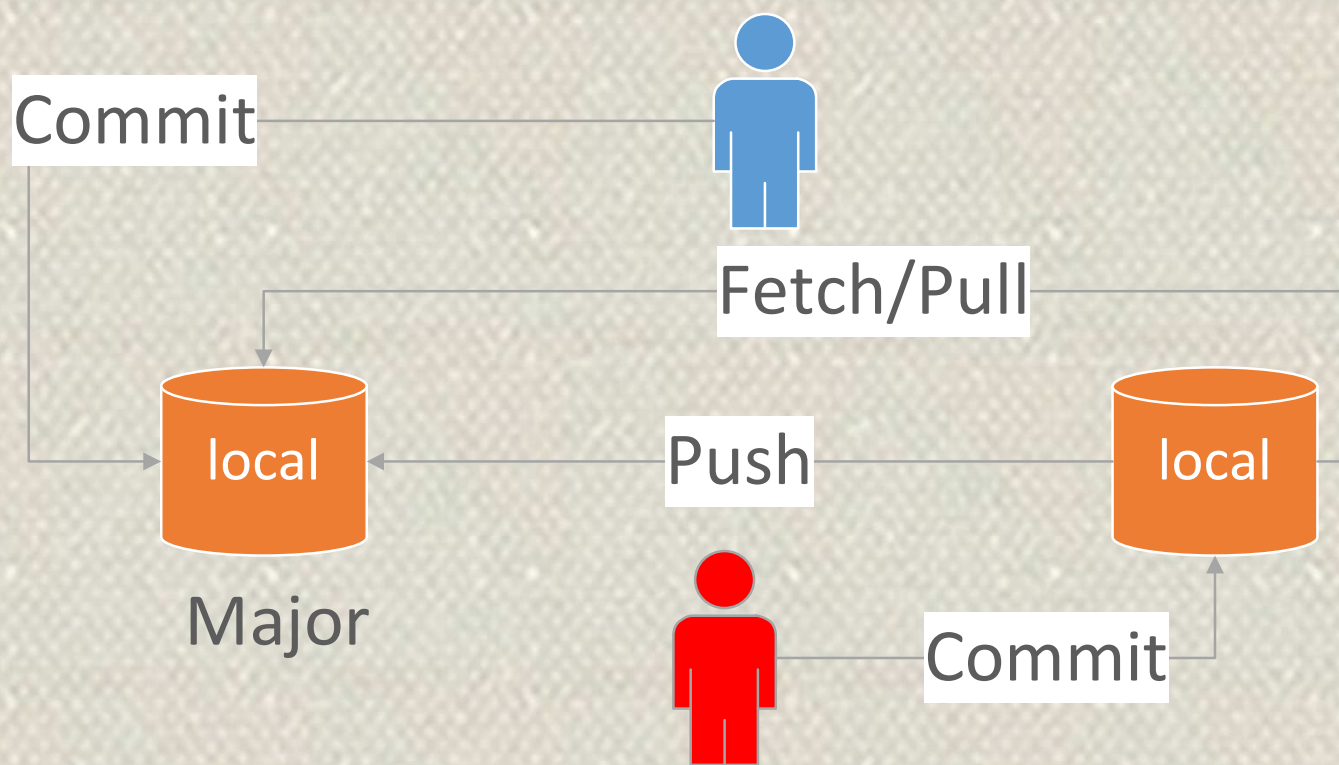






# 思考

- 是否真的不使用集中的服务器呢？





# 一些资源的推荐

- svnspot <http://code.svnspot.com/>
- Github <https://github.com/>
- Gitlab <https://gitlab.com/>
- 码云 <https://gitee.com/>

# 使用GitHub

- 就是一个应用Git的一个集中的管理各版本库的平台
- 你可以建立多个库
- 你可以邀请其他人参与你的项目
- 你可以向你clone的库申请提交push的权限