



软件配置管理

清华大学软件学院 龚云飞



目录



1

软件配置管理

2

团队开发的版本问题

3

GIT 简介

4

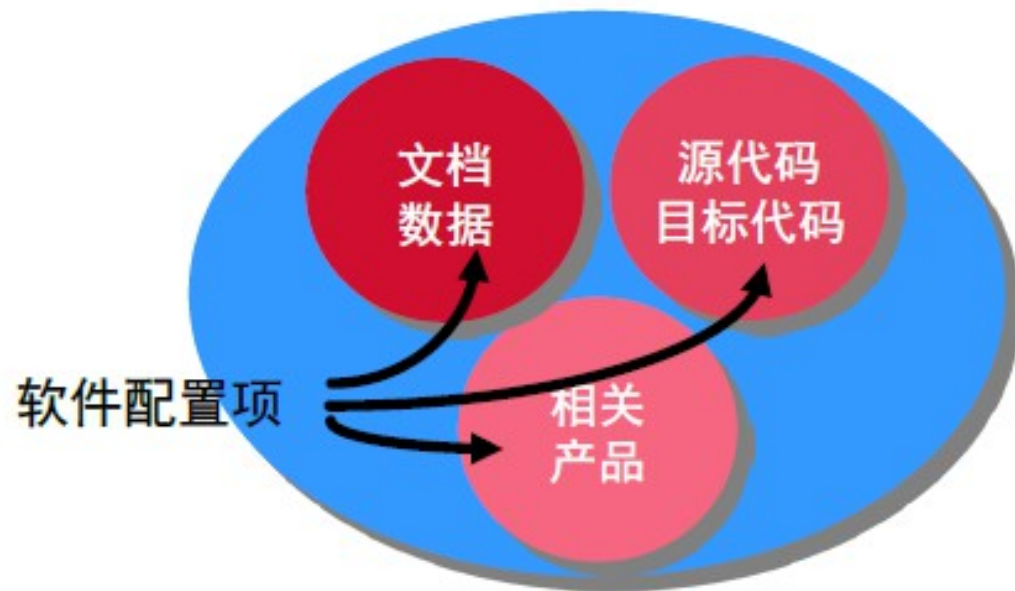
GIT 安装与 GITHUB 介绍

软件配置管理

- **软件配置管理**是一种标识、组织和控制修改的技术，它作用于整个软件生命周期，其目的是使错误达到最小并最有效地提高生产率。
- 软件配置管理的作用
 - 管理在软件生命周期中建立和修改的各种不同元素
 - 协调和整理所开发的产品
 - 管理软件的构建和测试环境
 - 管理发布和安装工具
 - 管理软件的改错和功能增加

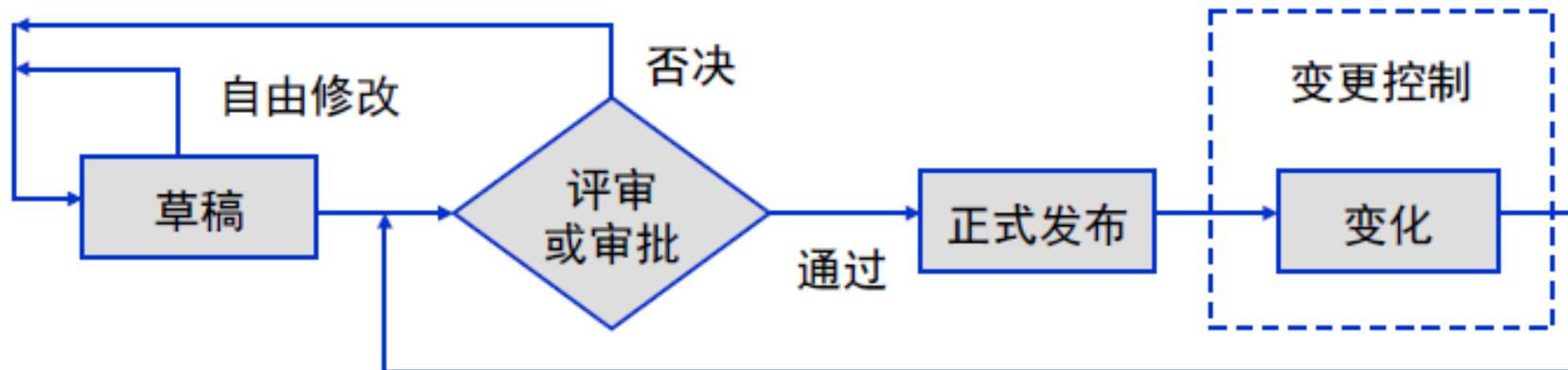
软件配置管理

- **软件配置项**（Software Configuration Item，简称 SCI）是为了配置管理而作为单独实体处理的一个工作产品或软件



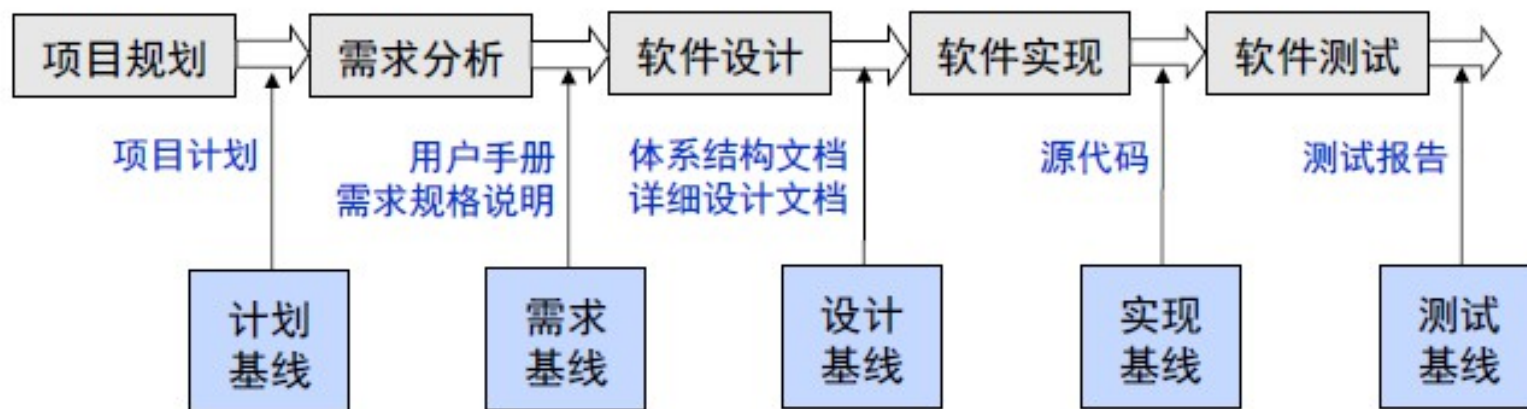
软件配置管理

- SCI 属性
 - 名称、标识符、状态、版本、作者、日期等
- SCI 状态
 - 草稿 (Draft) , 正式发布 (Released) , 变化 (Changing)



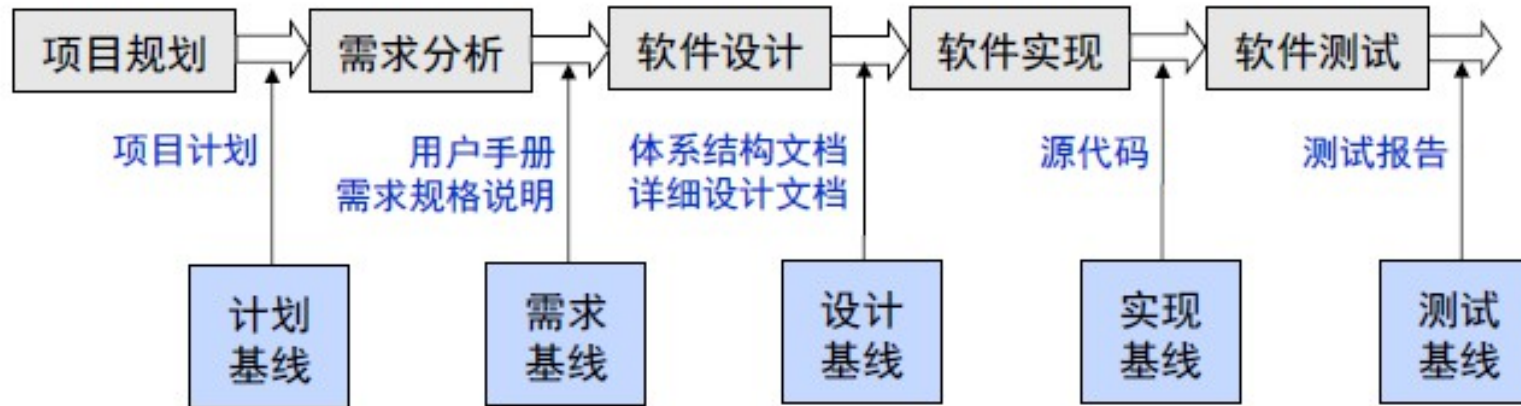
软件配置管理

- **基线（BaseLine）** 是已经通过了正式复审的规格说明或中间产品，它可以作为进一步开发的基础，并且只有通过正式的变化控制过程才能改变。
- 基线标志着软件开发过程的各个里程碑（Milestone）。



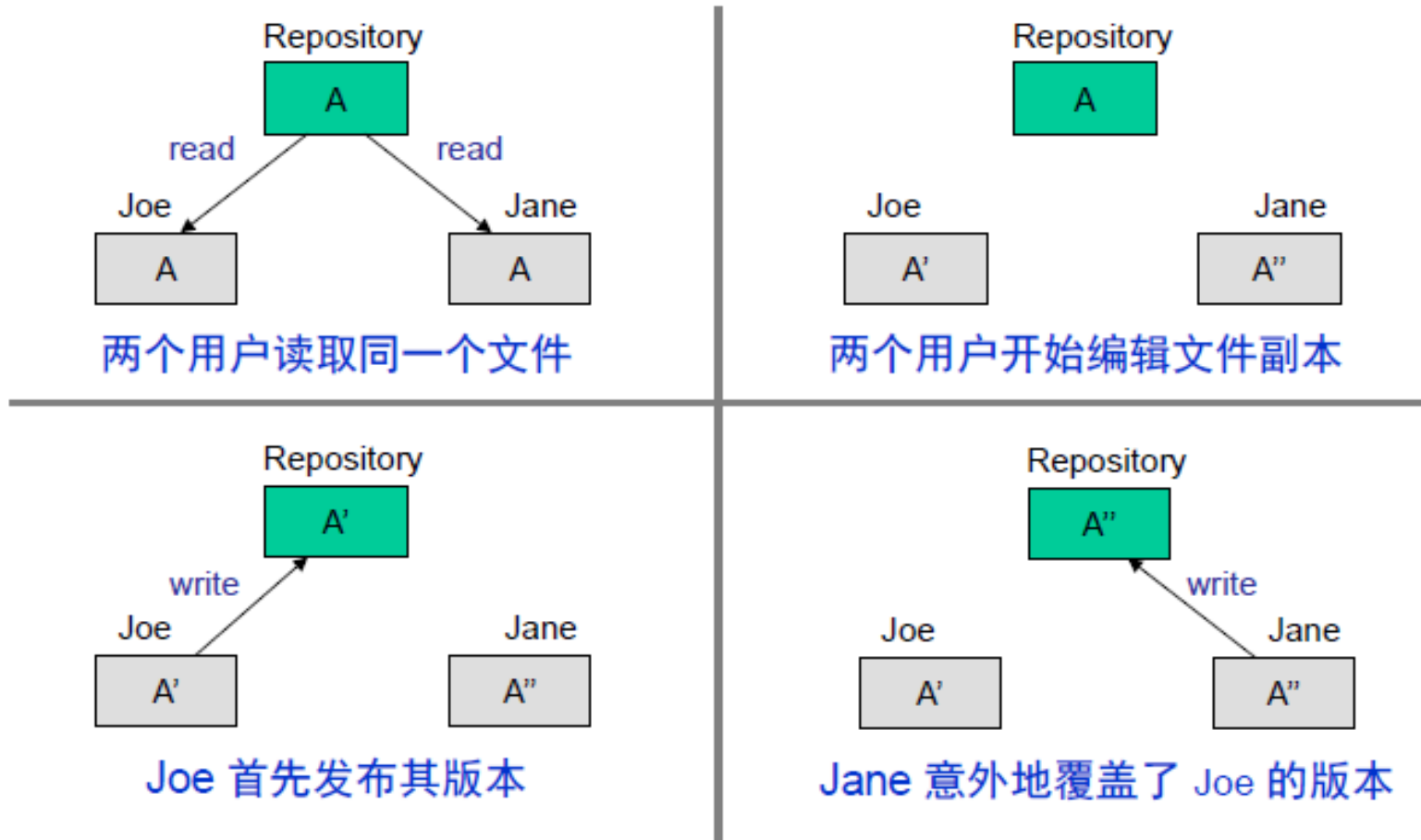
软件配置管理

- **基线（BaseLine）** 是已经通过了正式复审的规格说明或中间产品，它可以作为进一步开发的基础，并且只有通过正式的变化控制过程才能改变。
- 基线标志着软件开发过程的各个里程碑（Milestone）。



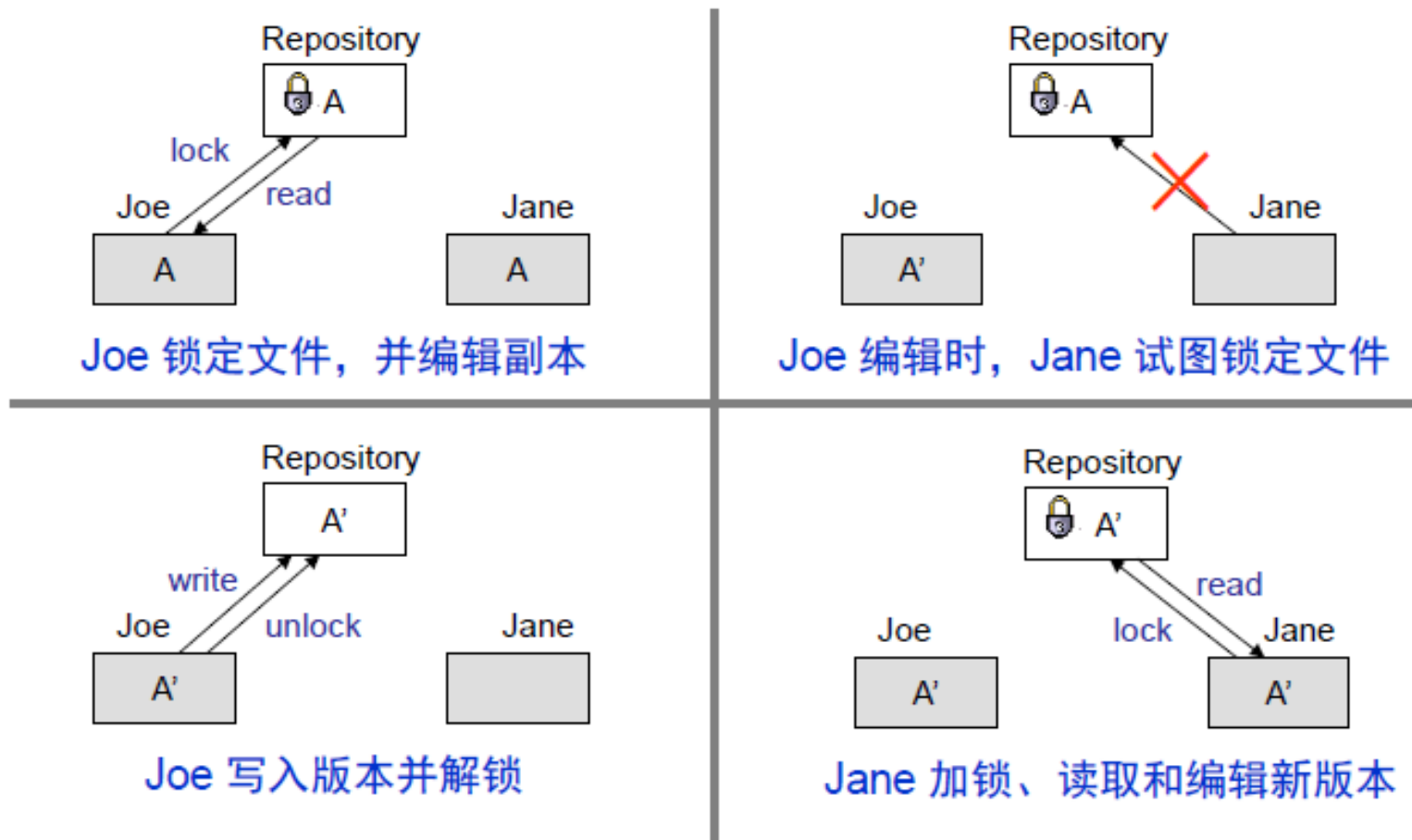
软件配置管理

■ 文件共享的问题



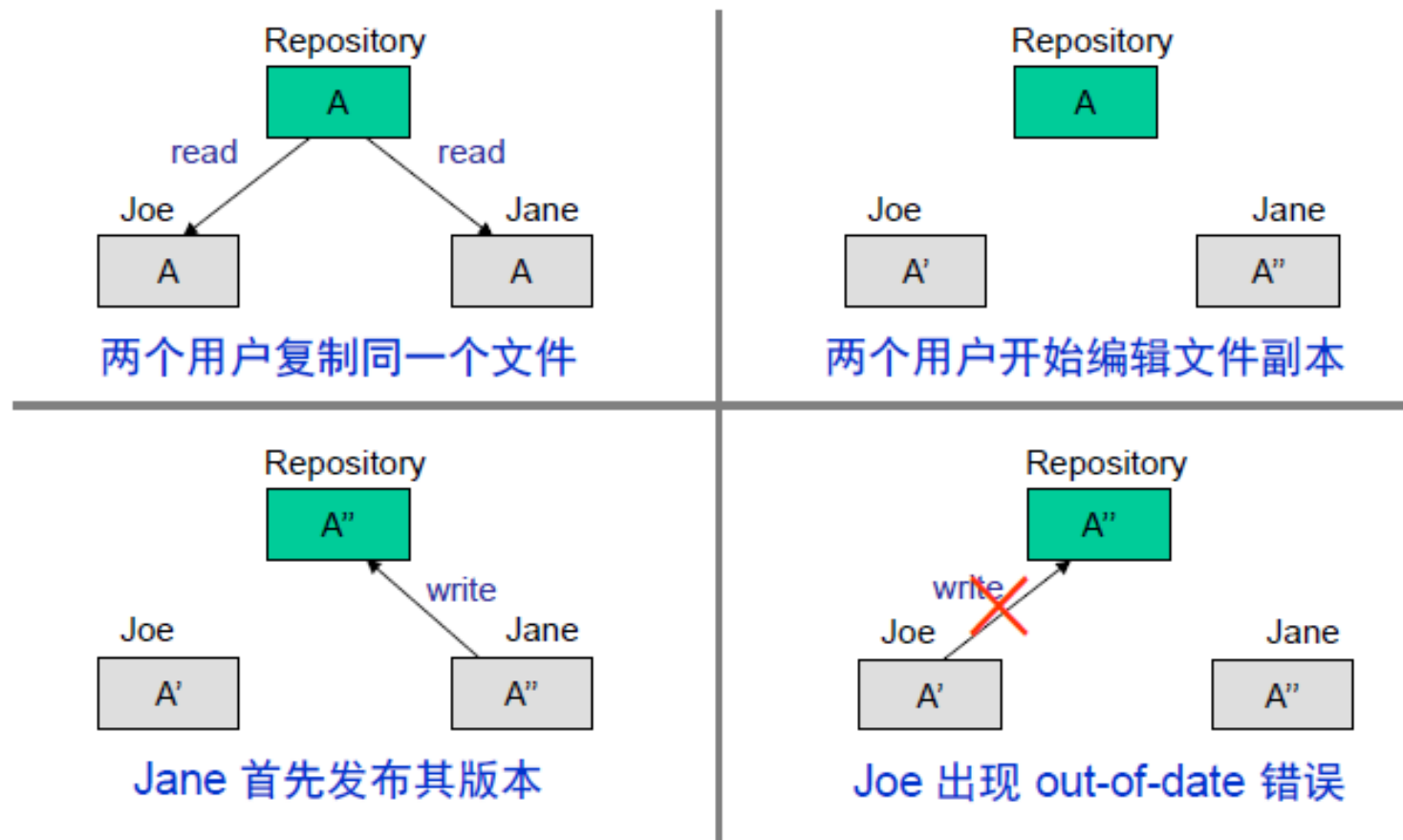
软件配置管理

■ 版本控制模型（独占工作模式）



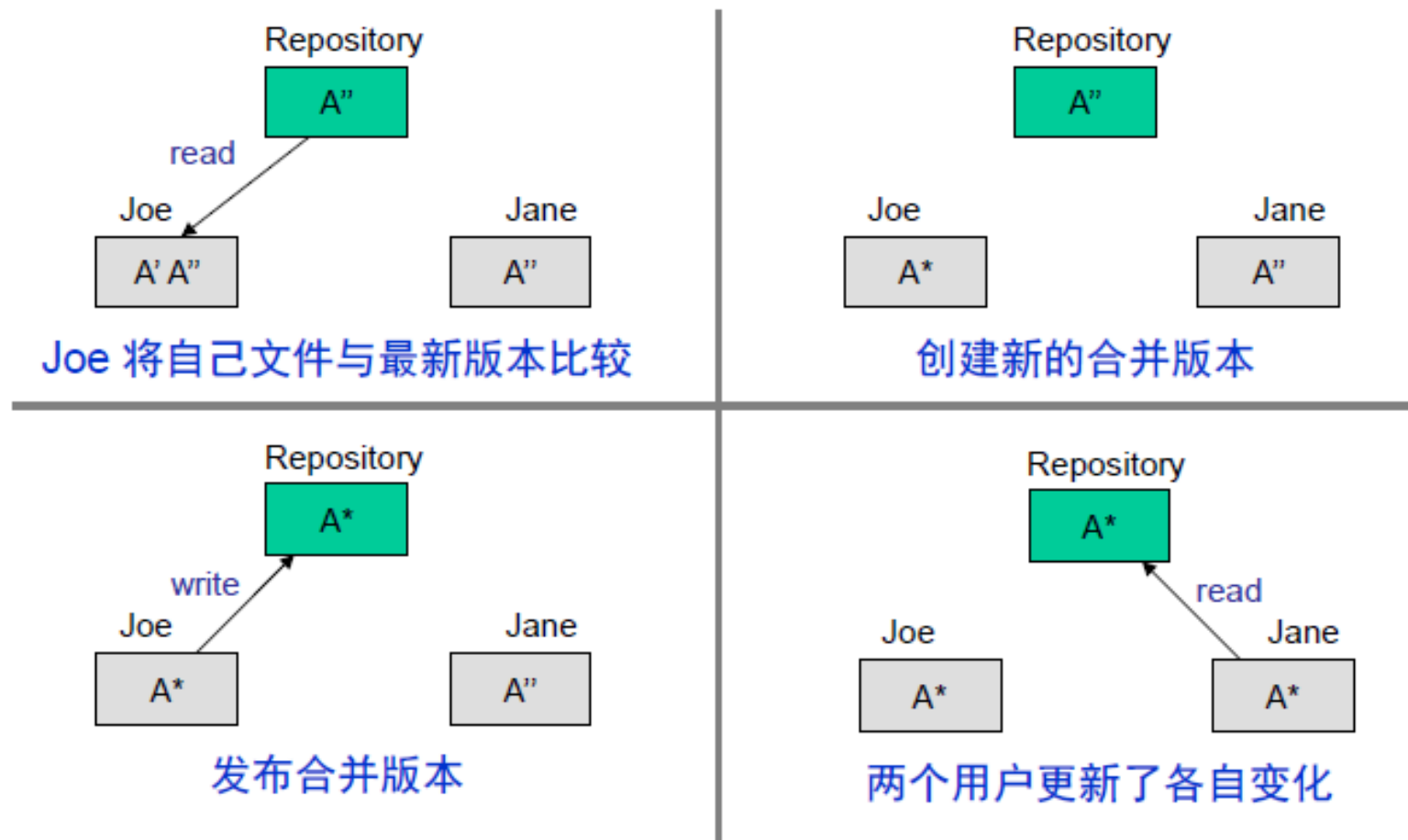
软件配置管理

■ 版本控制模型（并行工作模式）



软件配置管理

■ 版本控制模型（并行工作模式）



目录



1

软件配置管理

2

团队开发的版本问题

3

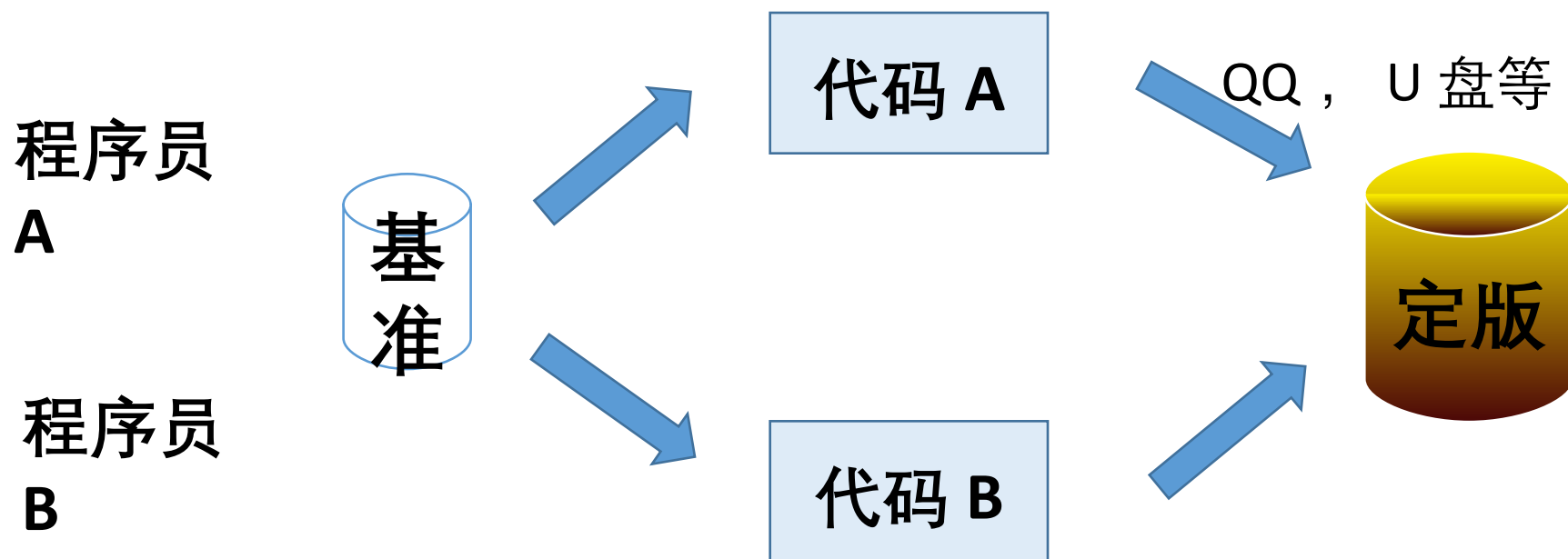
GIT 简介

4

GIT 安装与 GITHUB 介绍

团队开发的版本问题

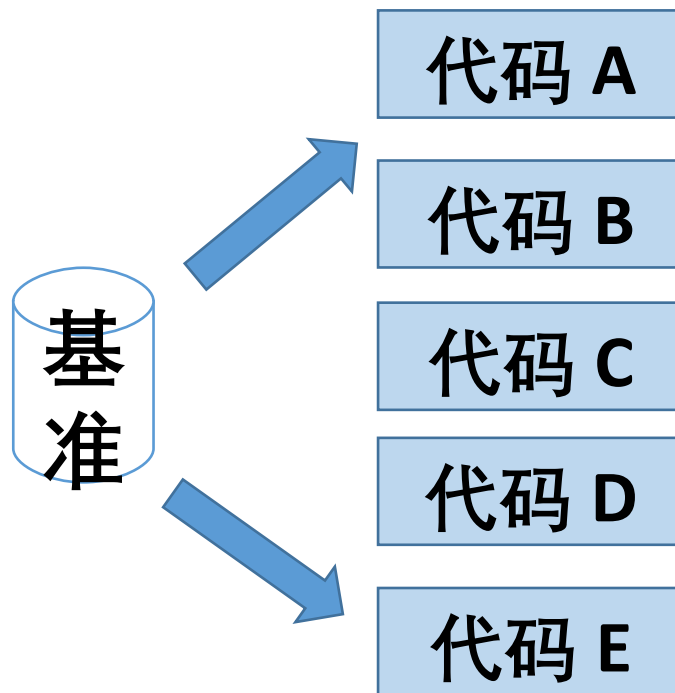
- 小程序开发
 - 涉及人数 1-2 人
 - 没有版本的概念



团队开发的版本问题

- 大项目开发
 - 涉及人数 4-5 人以上
 - 有明确的版本概念（迭代一版本，稳定版等等）

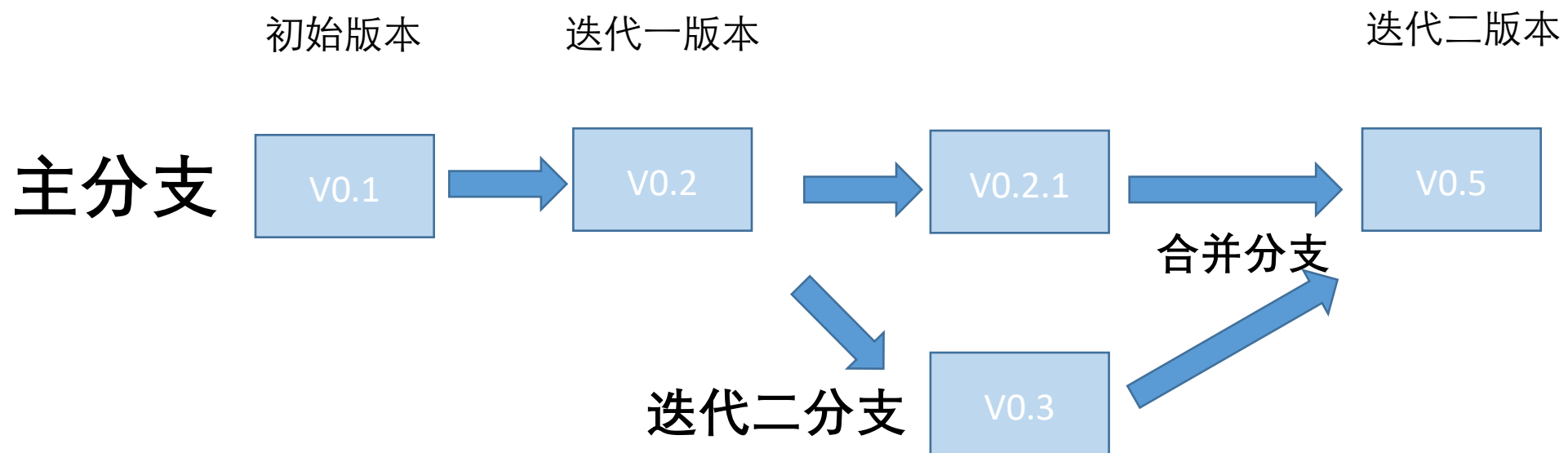
程序员
程序员
程序员
程序员
程序员
E



怎么合并？

团队开发的版本问题

- 大项目开发
 - 涉及人数 4-5 人以上
 - 有明确的版本概念（迭代一版本，稳定版等等）



目录



1

软件配置管理

2

团队开发的版本问题

3

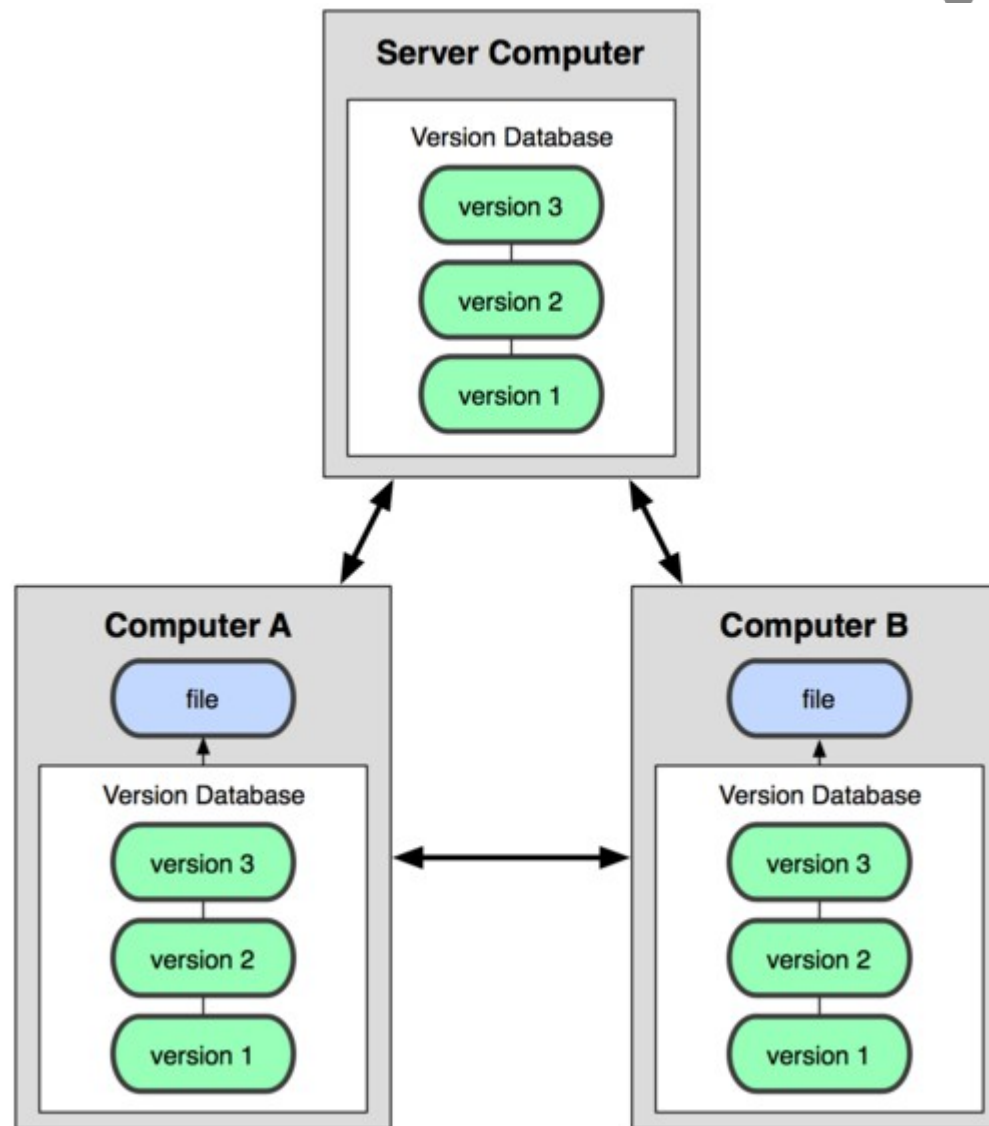
GIT 简介

4

GIT 安装与 GITHUB 介绍

GIT 简介

- GIT：开源的分布式版本控制软件
 - 常见版本控制工具的所有功能
 - 速度快
 - 分支与合并表现出色



GIT 简介



■ 场景一：打分支

■ 多个团队开发

■ 新功能团队和维护团队

■ 多个团队同时进行重构（软工课情况）

■ 操作步骤

1. Git Clone 到本地
2. Git Branch 打分支
3. 在分支上进行开发（Git Pull, Git Push）
4. Git Merge 合并分支

GIT 简介

- GIT 指令（远程仓库相关）
- Git clone
 - 从远程服务器 GIT 仓库克隆一份到本地
 - 针对已经存在的 GIT 项目，例如司徒组的项目
- Git init
 - 本地新创建了一个工作目录，用 Git init 告诉 GIT 来管理该目录
- Git remote
 - 查看 (-v)，添加 (add)，删除 (rm)，修改 (set-url) 远程服务器仓库

GIT 简介



- GIT 指令（分支相关）
- Git branch
 - 查看，创建，删除分支
- Git checkout
 - 切换本地仓库中的分支
- Git merge
 - 合并两个分支，合并时可能需要解决冲突。

GIT 简介



■ 场景二：日常修改提交功能点

- 团队开发的最常见的场景

■ 操作步骤

1. Git Pull 到本地仓库
2. 如果存在冲突，则进行冲突处理
3. 保证可以编译，并且功能点测试通过
4. Git Commit 到本地仓库
5. Git Push 到远程仓库

GIT 简介

- GIT 指令（常见指令）
- Git pull
 - 从远程或本地版本库中，将代码更新到本地
 - 类似于 svn 的 update
- Git add
 - 将当前更改或者新增的文件加入到 Git 的索引中（记入了版本历史）
 - 提交 (commit) 之前所需要执行的一步
- Git rm
 - 从当前的工作空间中和索引中删除文件

GIT 简介



- GIT 指令（常见指令）
- Git commit
 - 提交当前工作空间的修改内容到本地版本库
 - 类似于 svn 的 commit，但不发生网络通信。
- Git push
 - 将本地 commit 的代码更新到远程版本库中
 - 类似于 svn 的 commit，发生网络通信

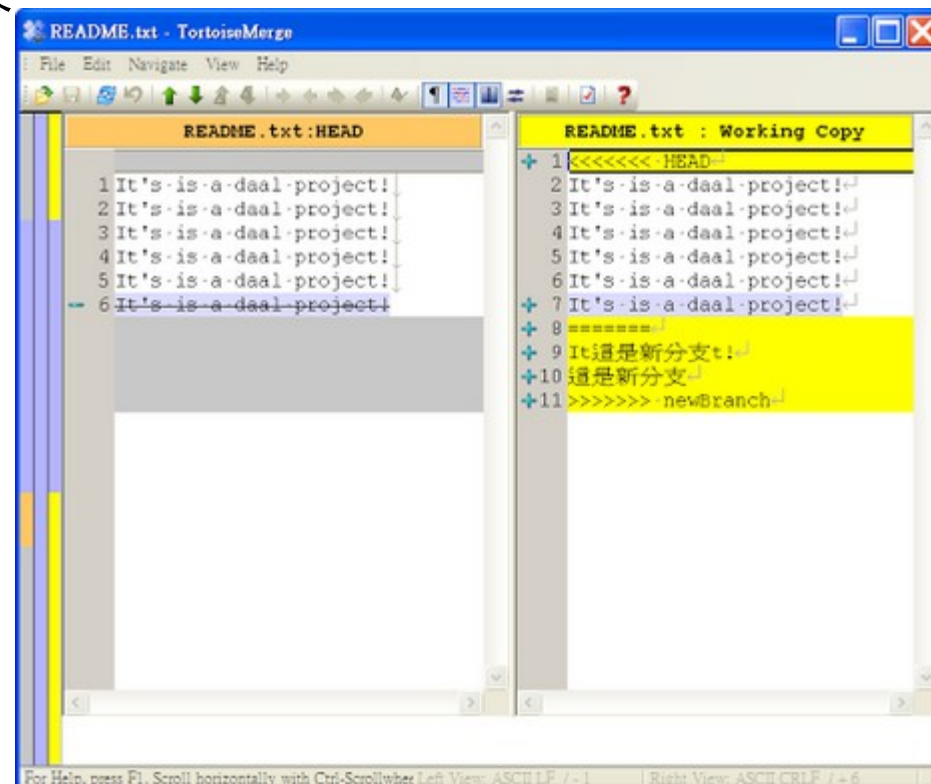
GIT 简介



- GIT 指令（常见指令）
- Git log
 - 查看历史日志
 - 可以查看所有历史版本的所有修改记录
- Git revert
 - 回滚到某个版本
 - 主要用于错误恢复

GIT 简介

- GIT 冲突处理
 - 多人提交代码时难免会发生冲突
- 解决冲突方法
 - 直接修改冲突文件
 - 借助于图形界面



目录



1

软件配置管理

2

团队开发的版本问题

3

GIT 简介

4

GIT 安装与 GITHUB 介绍

GIT 安装（windows 平台）

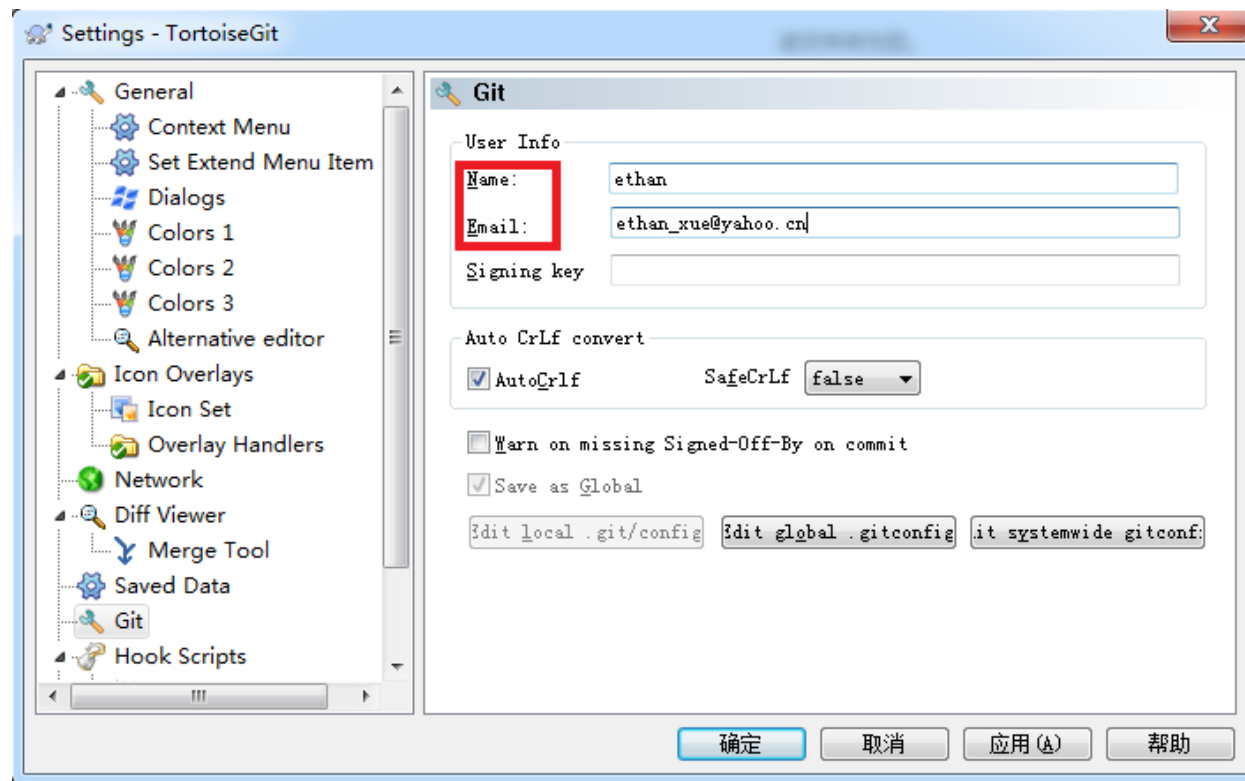
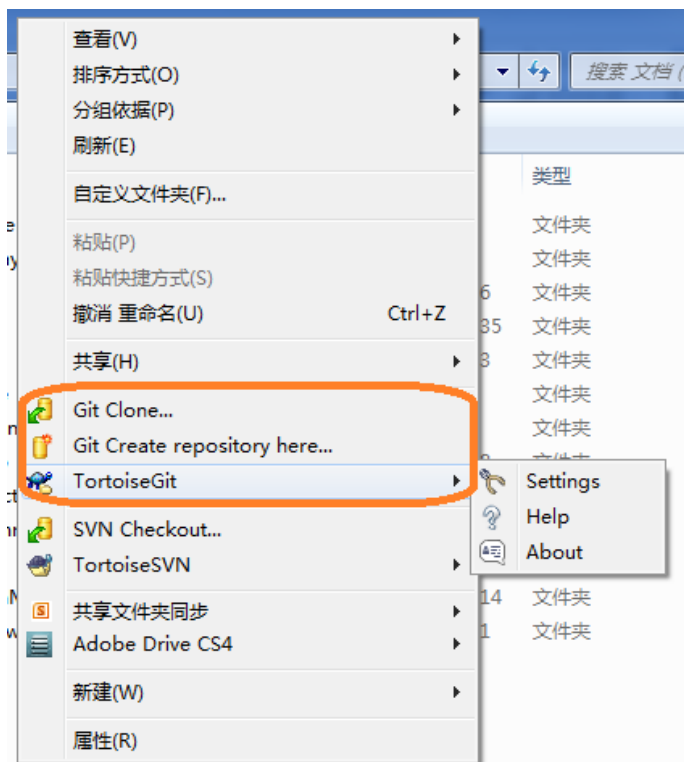
- Git for windows（又叫 msysGit，必须安装）
 - Git 的 windows 版本
 - 可以通过 shell 命令行的方式完成所有 git 操作
 - 提供特别简单的 GUI
- SourceTree
 - 源于 MAC 下的 GUI 工具，已有 windows 版本
- TortoiseGit（选择安装）
 - 与 TortoiseSVN 类似的 Git 的 GUI 版本
 - 与 windows 系统整合，功能相对较强大
- 各类 IDE 的 git 插件（选择安装）
 - 比如 Eclipse 的插件 EGit

GIT 安装（windows 平台）

- Git for windows + TortoiseGit 流程介绍
 - 下载 Git For Windows (<https://code.google.com/p/msysgit/downloads/list?q=full+installer+official+git>)
 - 安装 Git For Windows
 - 请不要安装 “ Windows explorer integration”
 - 换行符相关选项时，选择 “ Checkout as-is, commit as-is”
 - 下载 TortoiseGit (<https://code.google.com/p/tortoisegit/wiki/Download>)
 - 安装 TortoiseGit

GIT 安装（windows 平台）

- Git for windows + TortoiseGit 流程介绍
 - 右键菜单整合
 - 点击 settings，填写 用户信息



GITHUB 介绍



- 如何建立服务器版本库?
- 方法一：自己配置服务器
 - 安全性高
 - 配置复杂度高
- 方法二：利用公开的资源（Github，推荐）
 - 云平台托管
 - 使用方便

GITHUB 介绍

- Github
- 全球最大的开源社交编程及代码托管网站
- 托管各种 GIT 资源版本库 (repository)，分为 private 和 public
 - Public 资源版本库不限制个数，免费。
 - Public 资源版本库任何人可见，组内人员可提交。
 - Private 资源版本库收费，但 edu 的邮箱可以免费申请有个数限制的资源版本库。
 - Private 资源版本库组内人员可见，可提交

GITHUB 介绍

- Github 使用流程
 - 注册 Github 账号，请使用 mails.tsinghua.edu.cn 的邮箱
 - 新建资源仓库（ create a new repo ）
 - 选择资源为 private
 - 添加小组成员为可修改用户，添加老师、助教为可读用户
 - 建立成功后，记下远程版本库的地址，例如 <https://github.com/flyawayTHU/sepm.git>
 - 在本地使用 git 的 clone, pull, push 等操作



谢谢大家！

THANKS

