outline

- 编码
- 配置管理
- 质量管理
- 软件度量

No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle.

—*First Law of System Engineering*

概述

- 开发软件系统的过程中，变更是不可避免的
  - 用户要变更需求 ← 知道了更多他们想要的功能
  - 开发人员要变更技术方法 ← 知道了哪个技术方法更好
  - 项目管理人员要变更项目策略 ← 知道了哪种管理方法更好
  - **Why?**

概述

- 软件配置管理（**S**oftware **C**onfiguration **M**anagement）
  - 变更管理
  - 一组管理变更的活动（umbrella activity）
  - 贯穿于整个软件过程中的保护性活动
  - “如果你不控制变更，那么变更将控制你”
  - 一个未受控制的变更流可以很容易的将一个运行良好的软件项目带入混乱，结果会影响软件质量并且会推迟软件交付。
  - 目标
    - 每个工作产品都可以标识、跟踪和控制
    - 每个变更都可以跟踪和分析
    - 每个需要知道变更的人员都得到通知

一个SCM场景

- 一个典型的SCM场景包括
  - 项目经理：管理软件项目组
  - 配置管理员：负责配置管理过程
  - 软件工程师：负责开发、维护软件产品
  - 用户：使用产品
- 由4人组成的团队开发的15000行代码的小型软件
- 项目经理的角色和任务
  - 确保在预定的期限内开发完成
  - 控制开发的进度，识别问题并对其做出反应
  - 生成并分析软件系统状态报告

一个SCM场景

- 配置管理员的角色和任务
  - 确保代码的开发、变更和测试过程能够被跟踪
  - 负责建立正式的变更机制，包括变更请求、评估变更和授权变更
- 软件工程师的角色和任务
  - 高效地工作
  - 有效地沟通和协调
  - 通过配置管理系统上传、下载代码，解决代码冲突
- 用户的角色和任务
  - 遵守正式的变更请求过程

一个SCM场景

- SCM系统支持所有的角色和任务
  - 项目经理将SCM系统看作审核机制
  - 配置管理员将SCM系统看作控制、跟踪机制
  - 软件工程师将SCM系统看作变更、构建和访问控制机制
  - 用户将SCM系统看作质量保证机制

SCM概念

- 软件配置（Software Configuration）
  - 计算机程序（源代码和可执行程序）
  - 描述计算机程序的文档（针对技术开发者和用户）
  - 数据（包含在程序内部和程序外部）
  - 在技术文档中明确说明最终组成软件产品的功能或物理属性
  - 包含了所有在软件过程中产生的信息
- 其中每一项就称为一个软件配置项（Software Configuration Item）
  - 一个文档、一个全套的测试用例或一个已命名的程序构件
  - 特定版本的编辑器、编译器和其他CASE工具

基线

- 基线（baseline）

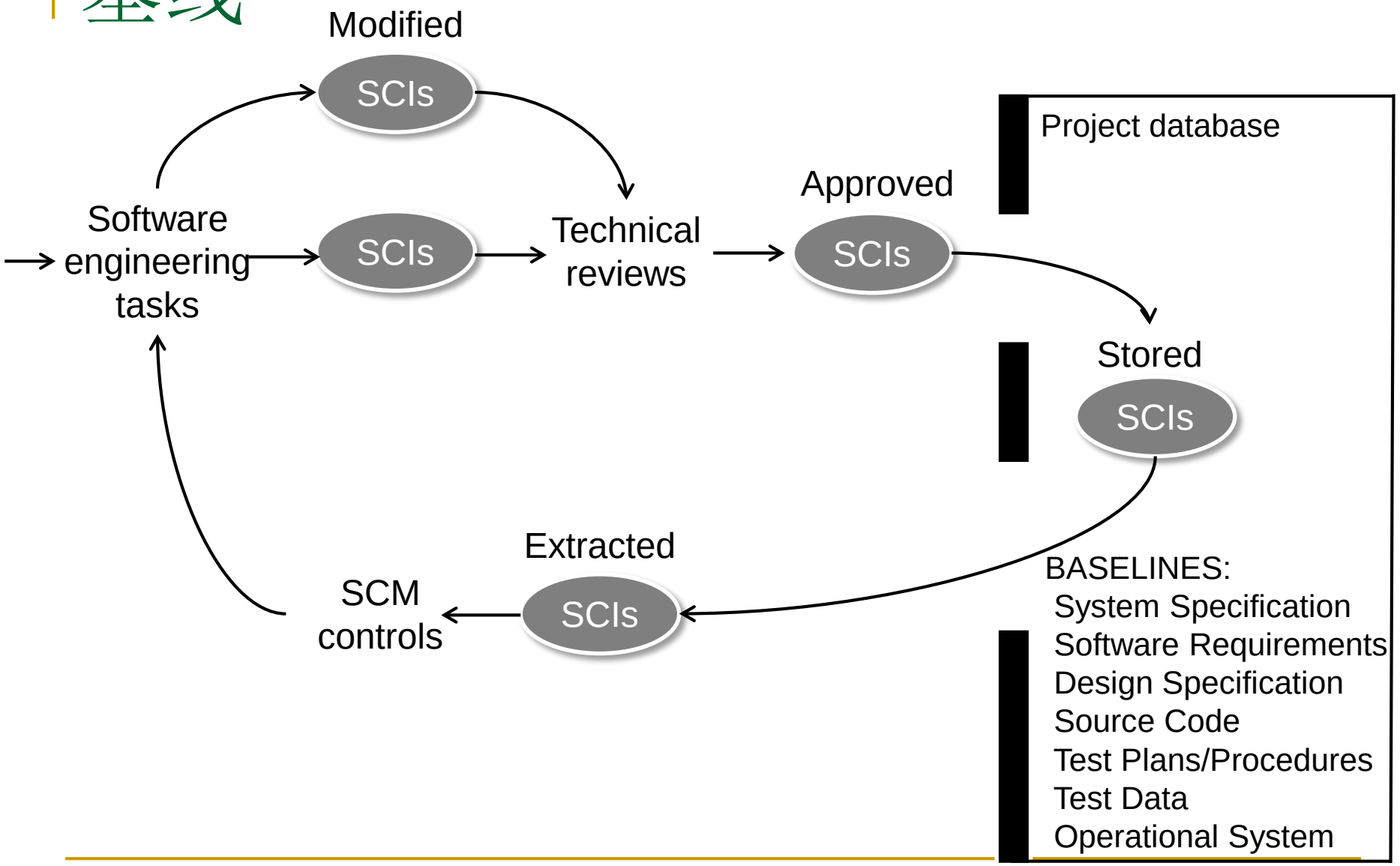
A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

—IEEE Std. No.610.12-1990

基线

- 基线是评审过的一个或多个SCI，每一个基线都是下一步开发的出发点和基础，是软件开发中的里程碑。
  - 设计模型已经文档化、评审过，错误已被发现并修改，已被认可了，就成为一条基线
- 在SCI变成基线之前，变化可以迅速而非正式地进行。一旦基线已经建立，变化可以进行，但是，必须应用特定的、正式的规程来评估和验证每个变化。

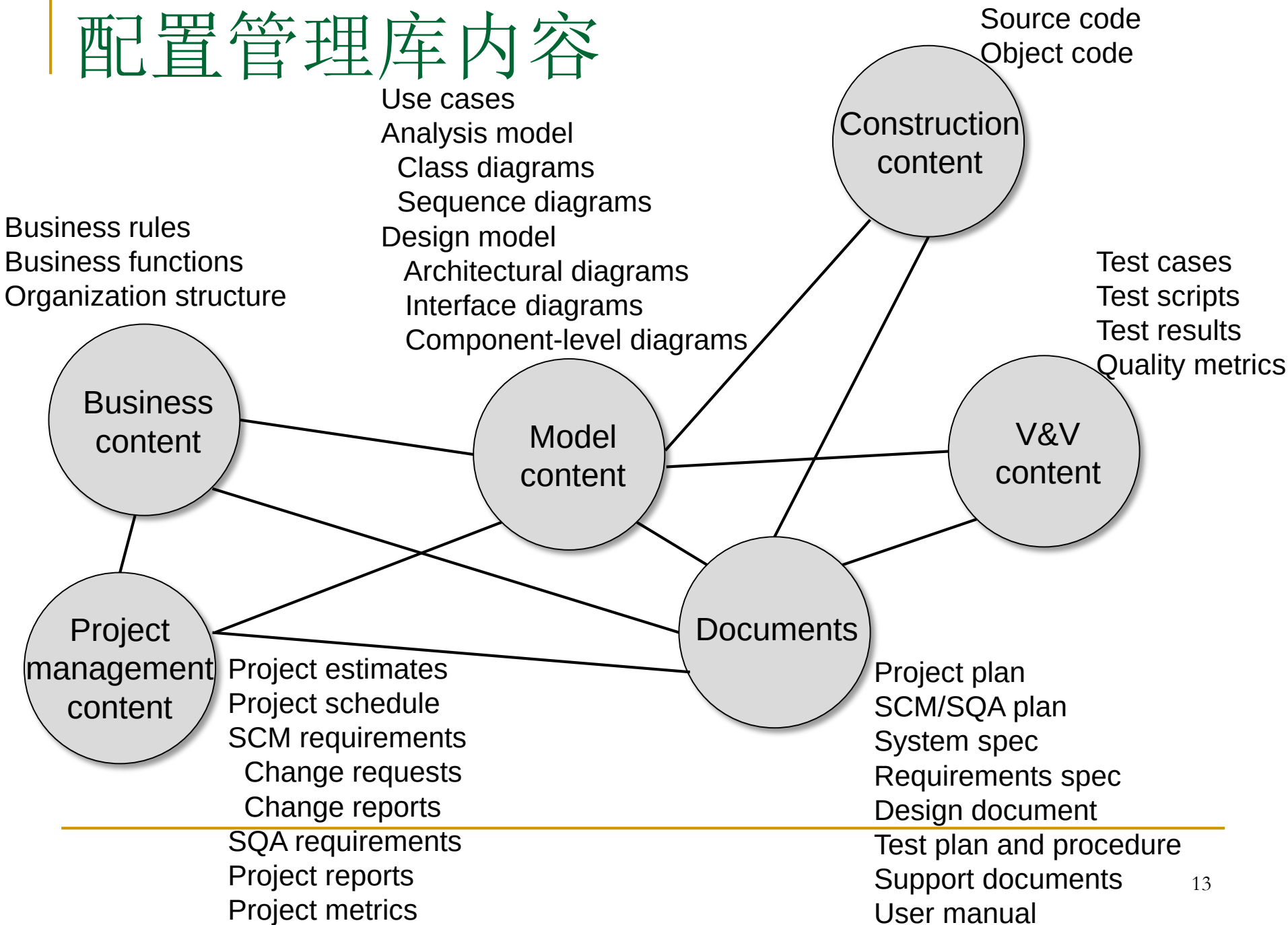
基线


Modified
SCIs
Project database
Approved
Software engineering tasks
SCIs
Technical reviews
SCIs
Stored
SCIs
Extracted
SCM controls
SCIs
BASELINES:
System Specification
Software Requirements
Design Specification
Source Code
Test Plans/Procedures
Test Data
Operational System

配置管理库（SCM Repository）

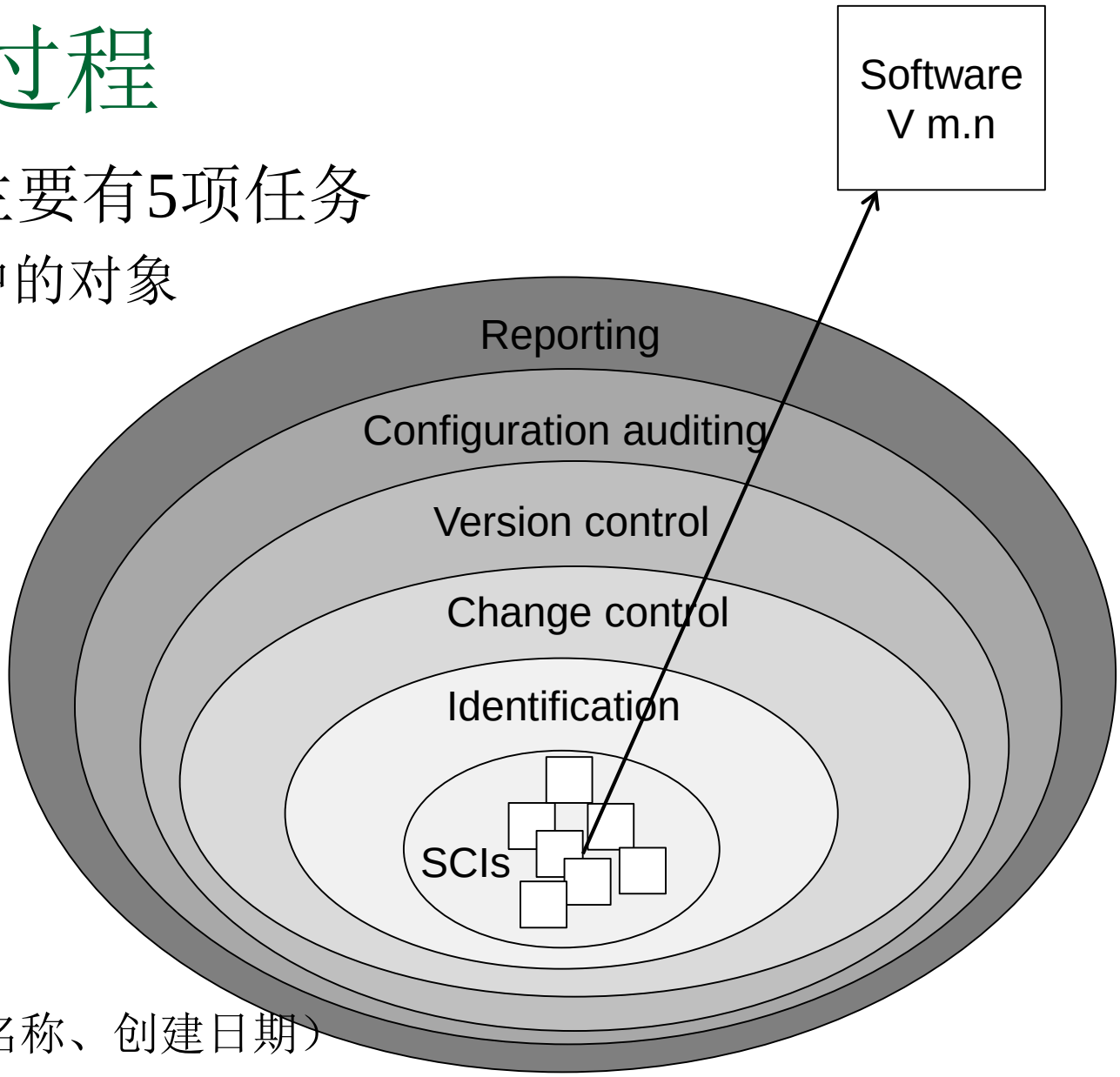
- 配置管理库也称受控库，用于存储软件配置项以及相关配置管理信息。
- 配置管理库特征
  - 版本控制：保留所有历史版本，能够回溯
  - 关联跟踪
    - 一个UML类图变更了，配置管理库能够检测出相关的类和接口的描述，以及代码也需要修改，并把受影响的SCI提交给开发人员
  - 需求跟踪
    - 正向跟踪：检查SRS中的每个需求是否都能在后继工作产品中找到对应点
    - 逆向跟踪：检查设计文档、代码、测试用例等工作产品是否都能在SRS中找到出处
  - 审计跟踪：变更源（when, why, who）

配置管理库内容


Source code
Object code
Construction content
Use cases
Analysis model
Class diagrams
Sequence diagrams
Design model
Architectural diagrams
Interface diagrams
Component-level diagrams
Business rules
Business functions
Organization structure
Test cases
Test scripts
Test results
Quality metrics
Business content
Model content
V&V content
Project management content
Documents
Project estimates
Project schedule
SCM requirements
Change requests
Change reports
SQA requirements
Project reports
Project metrics
Project plan
SCM/SQA plan
System spec
Requirements spec
Design document
Test plan and procedure
Support documents
User manual

配置管理过程

- 软件配置管理主要有5项任务
  - 标识软件配置中的对象
  - 变更控制
  - 版本控制
  - 配置审计
  - 状态报告
- 例：当一个新的SCI被创建时
  - 标识该SCI
  - 无变更控制
  - 生成一个版本号
  - 创建SCI的记录（名称、创建日期）
  - 报告给需要知道的人

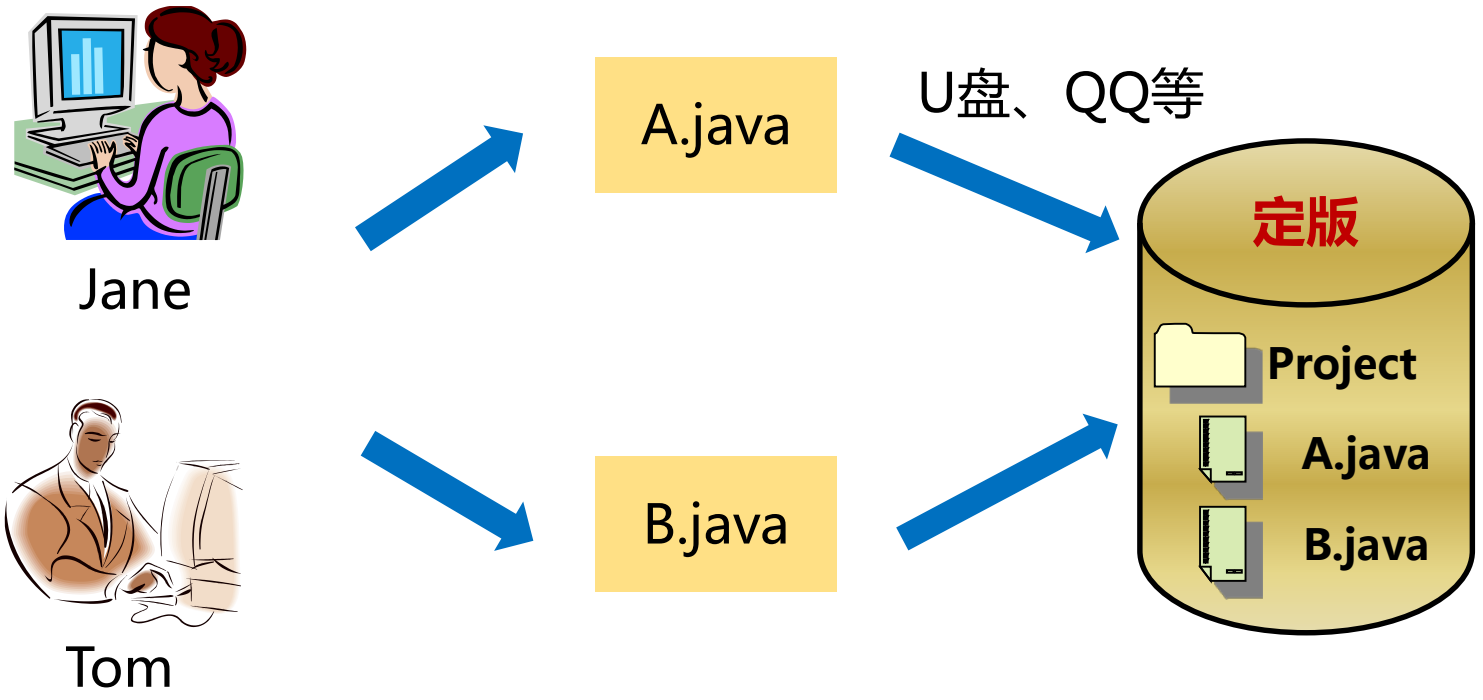


标识

- 标识软件配置中的对象
  - 单独命名每个配置项
  - 标识出两类对象
    - 基本对象：软件工程过程中产生的信息单元（源代码、测试用例集）
    - 聚集对象：基本对象和其他聚集对象的集合
  - 每个对象都有一组能唯一标识它的特征
    - 名称：无二义性地标识该对象的一个字符串
    - 描述：标识该对象所表示的SCI类型、变更、版本等的一组数据项
    - 资源表：该对象需要的实体（数据类型、变量名）
    - 实现：基本对象（“文本单元”），聚集对象（Null）

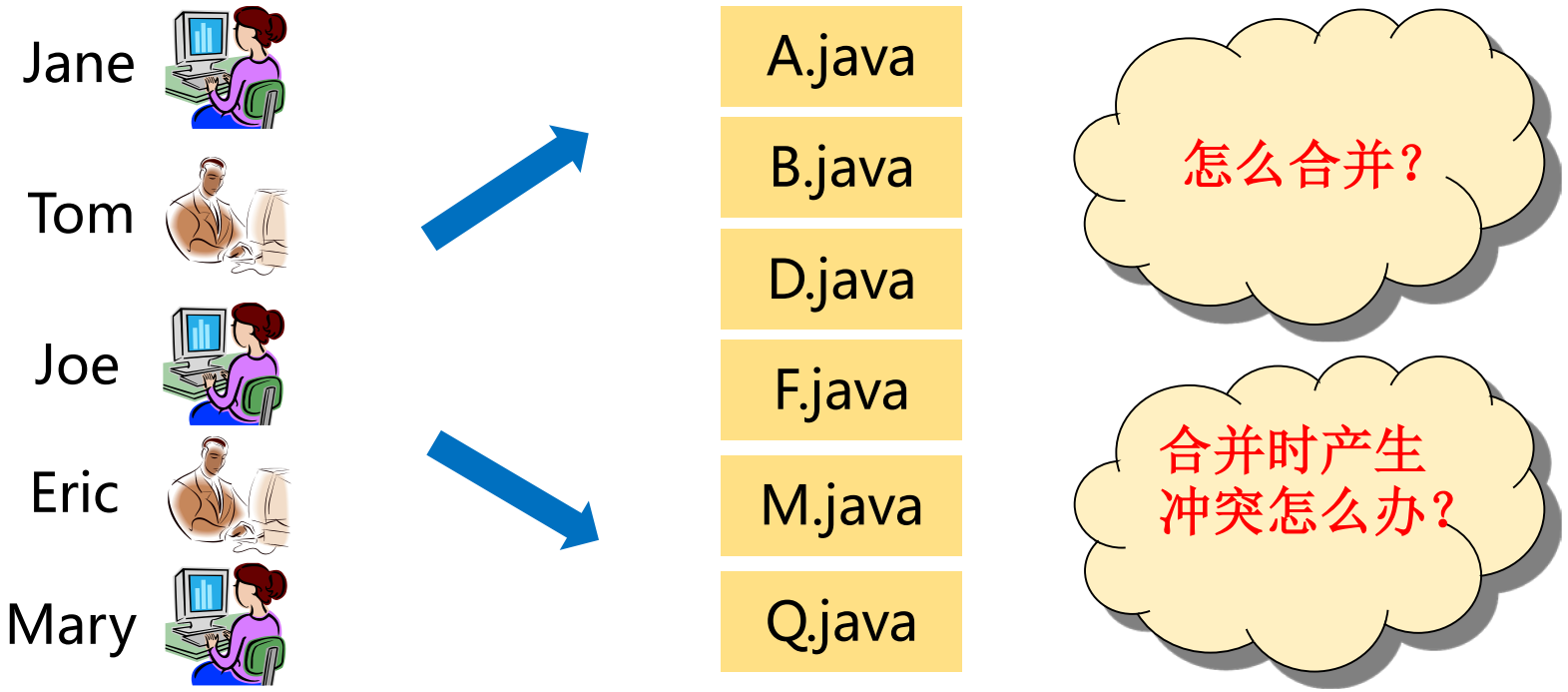
团队开发的版本问题

- 场景1：小型程序开发
  - 单独开发，两个程序员不修改同一个代码文件
  - 手工合并代码



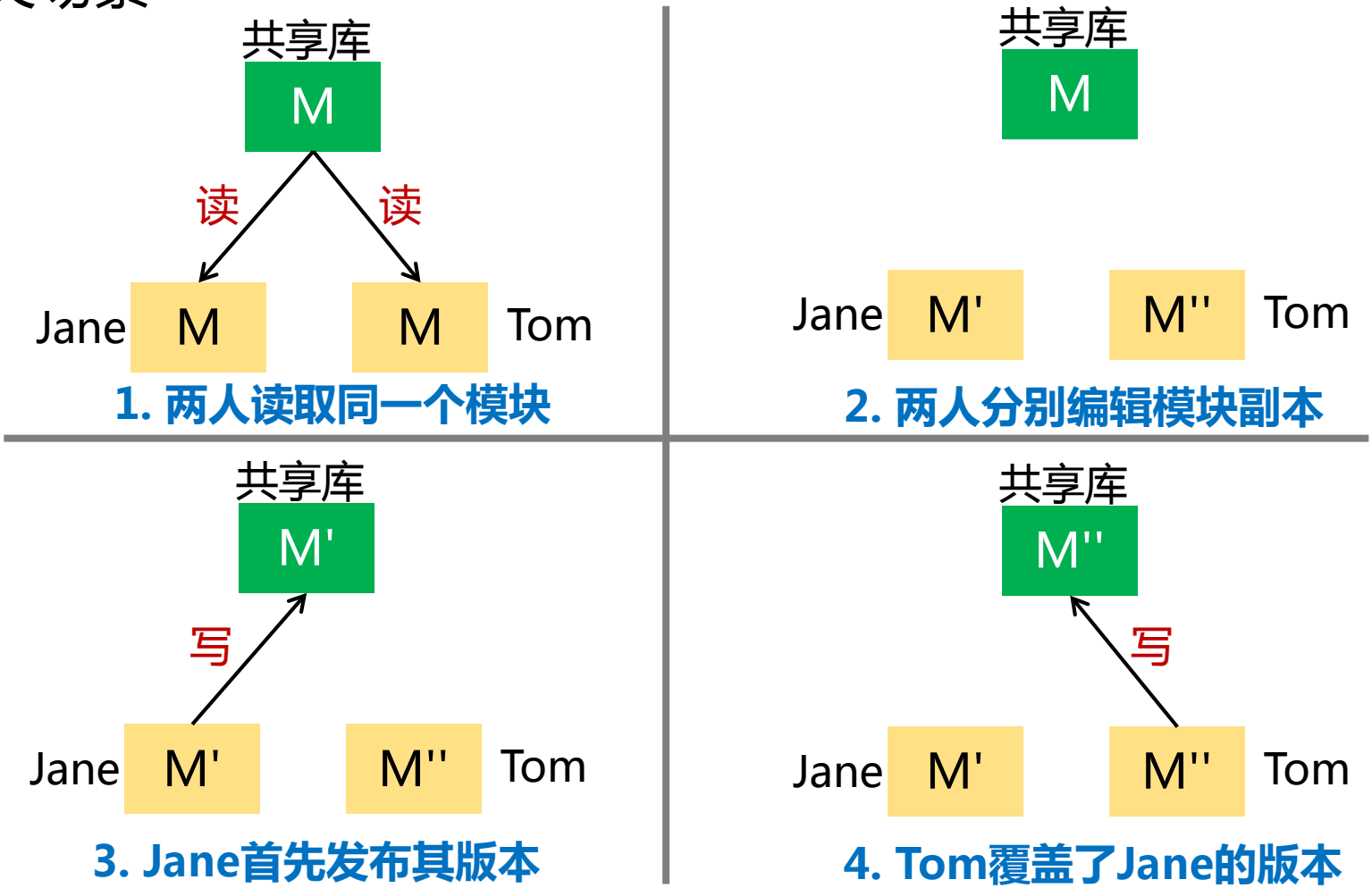
团队开发的版本问题

- 场景2：大型项目开发
  - 多人协作开发
  - 版本管理



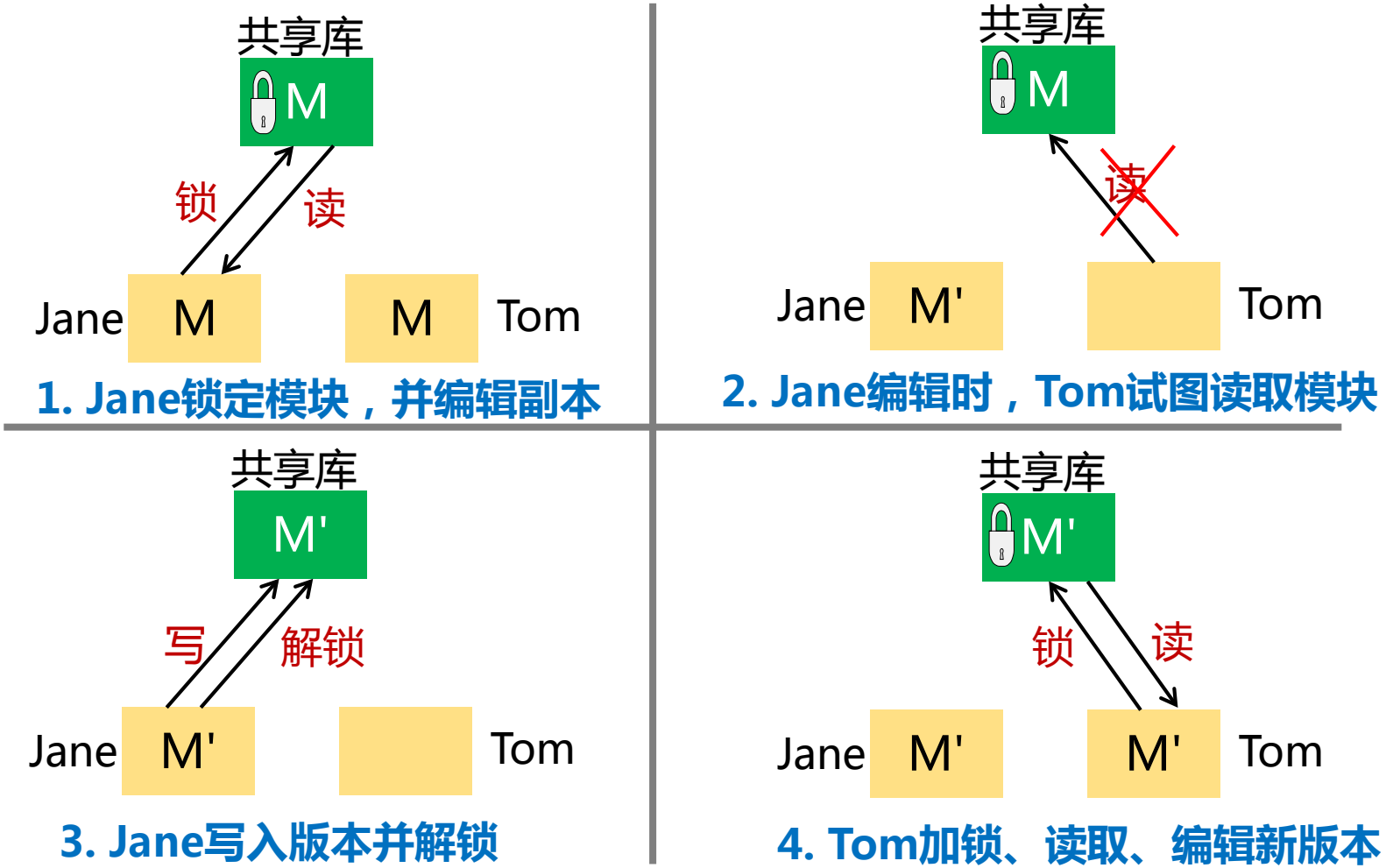
团队开发的版本问题

- 冲突场景



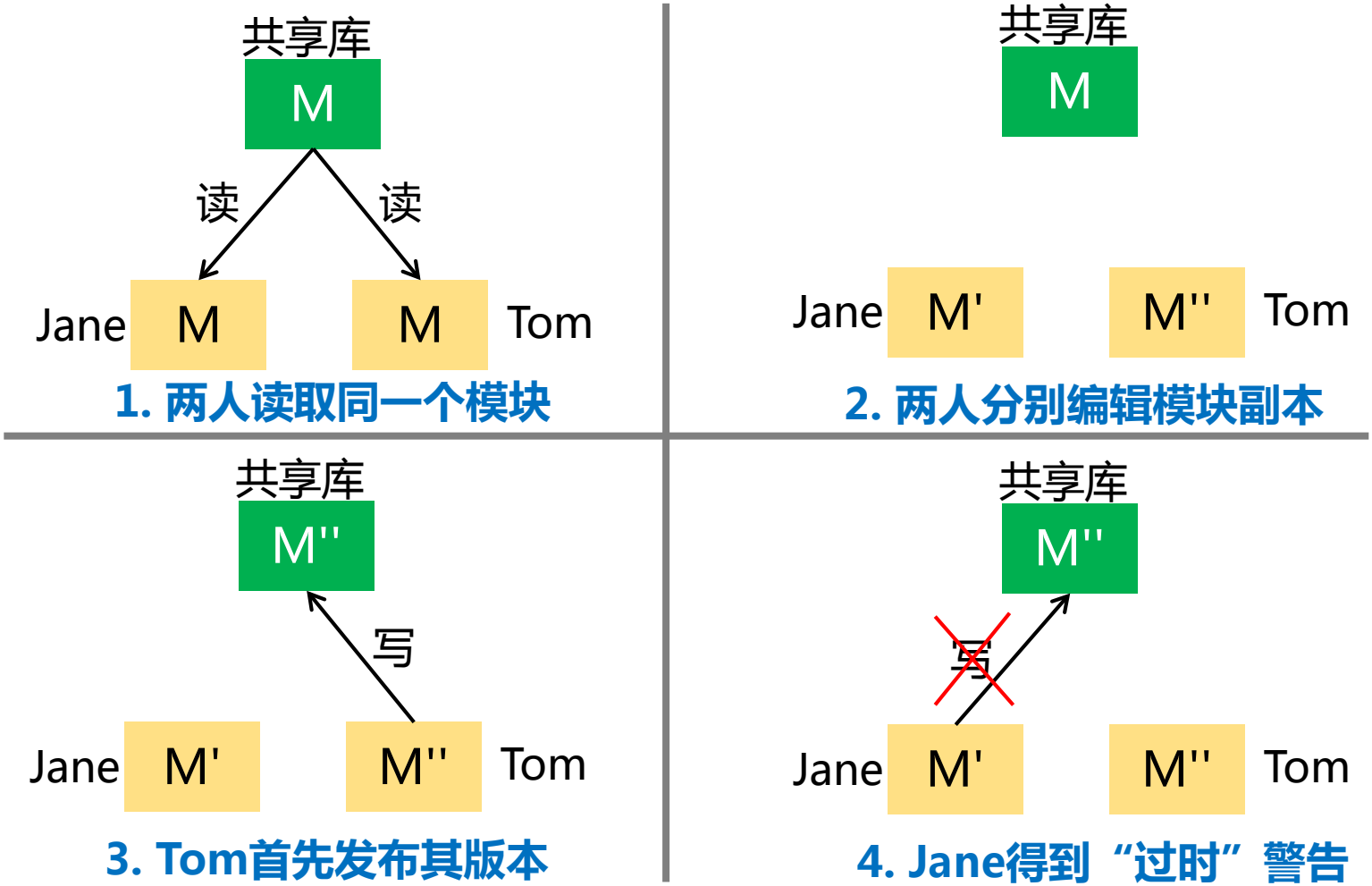
团队开发的版本问题

- 冲突解决策略1：独占工作模式



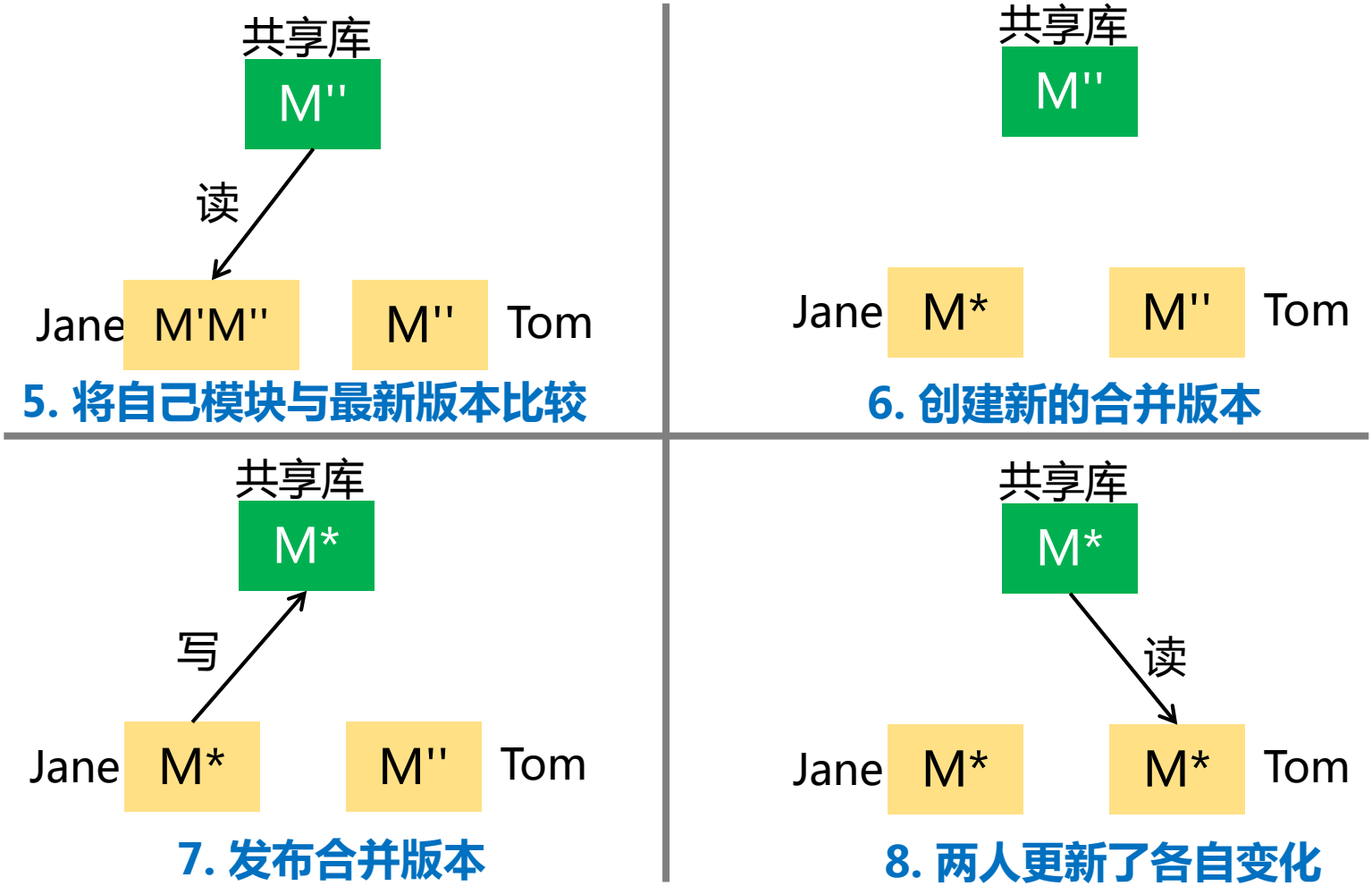
团队开发的版本问题

- 冲突解决策略2：并行工作模式



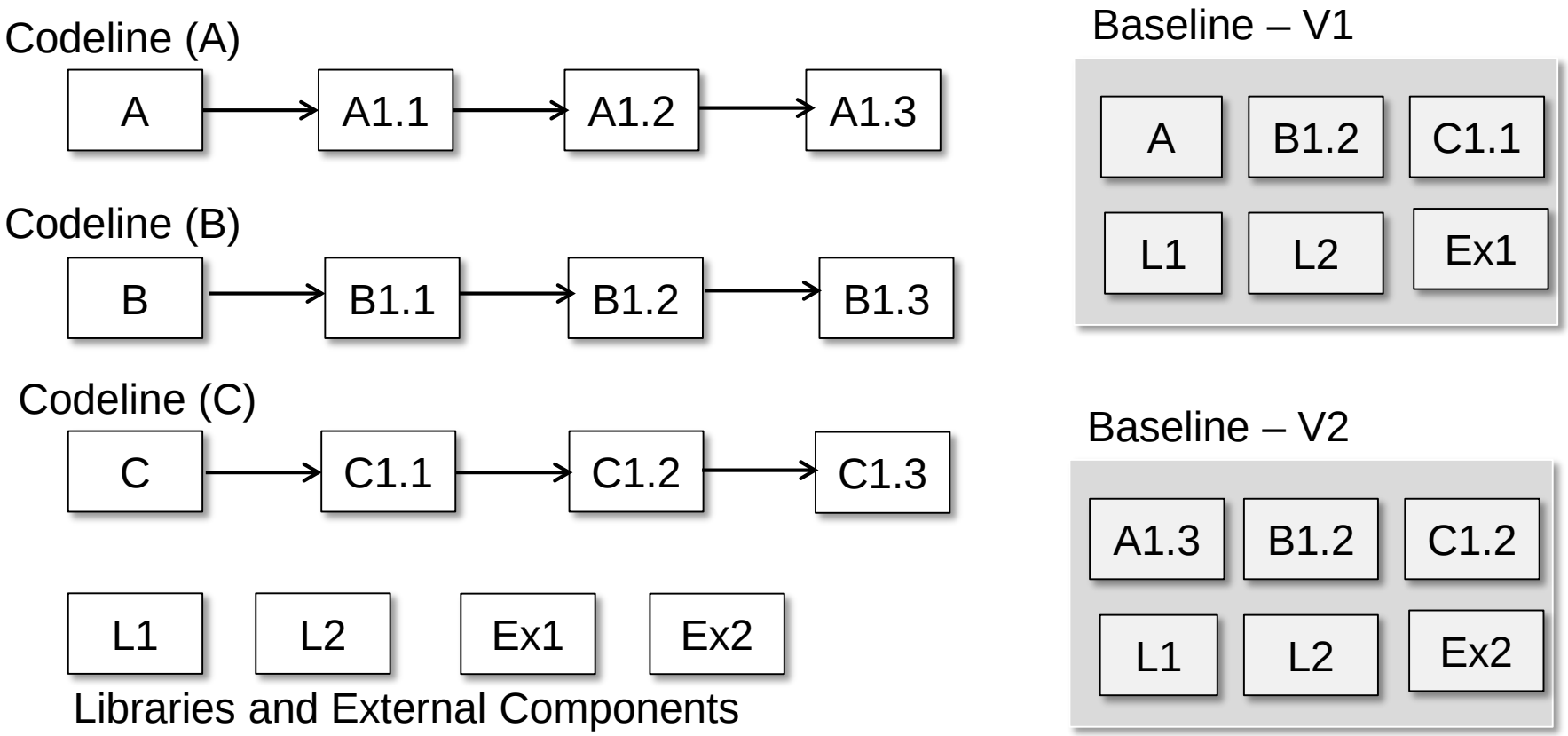
团队开发的版本问题

- 冲突解决策略2：并行工作模式



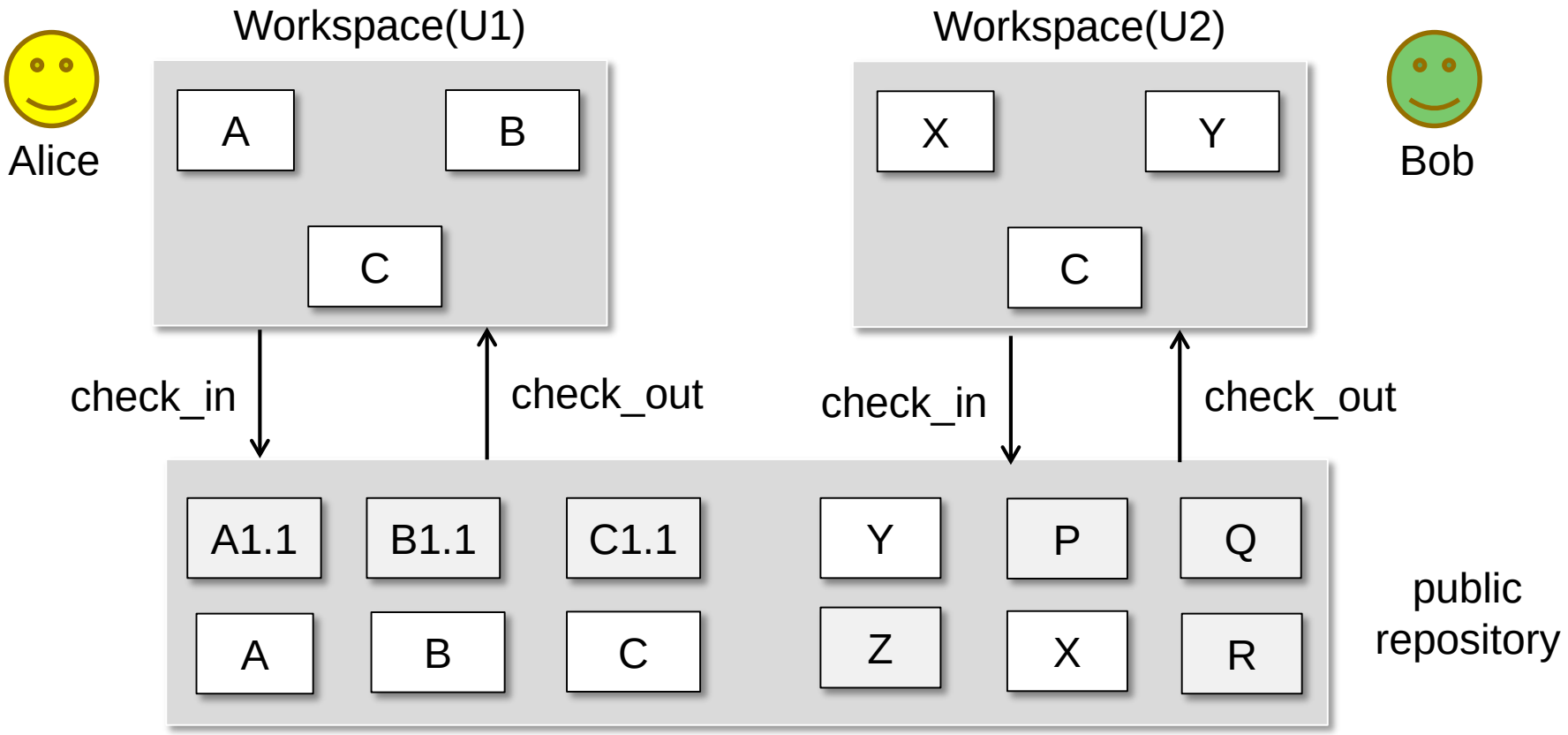
版本控制

- 版本控制
  - 管理在软件工程过程中所创建的构件的不同版本



版本控制

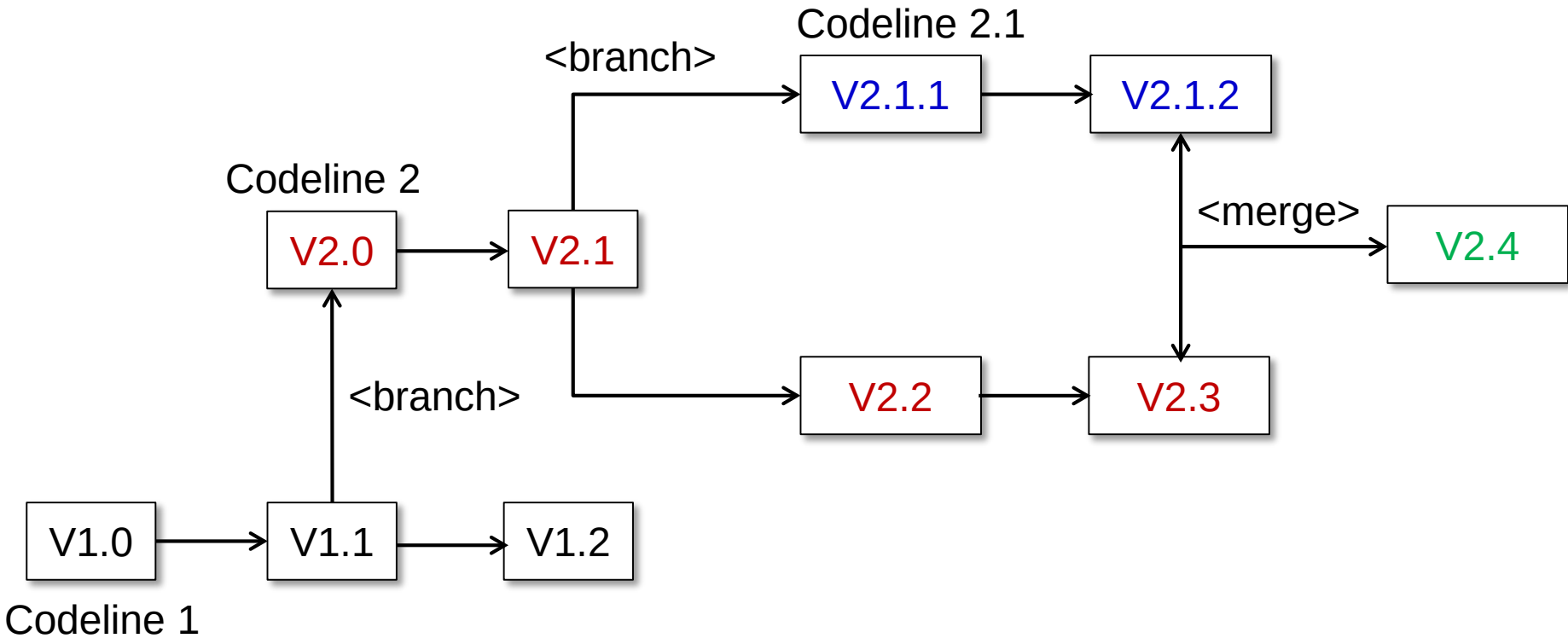
- 场景：软件开发是团队活动，不同的团队成员需要同时修改同一个构件



版本控制系统：CVS & Subversion

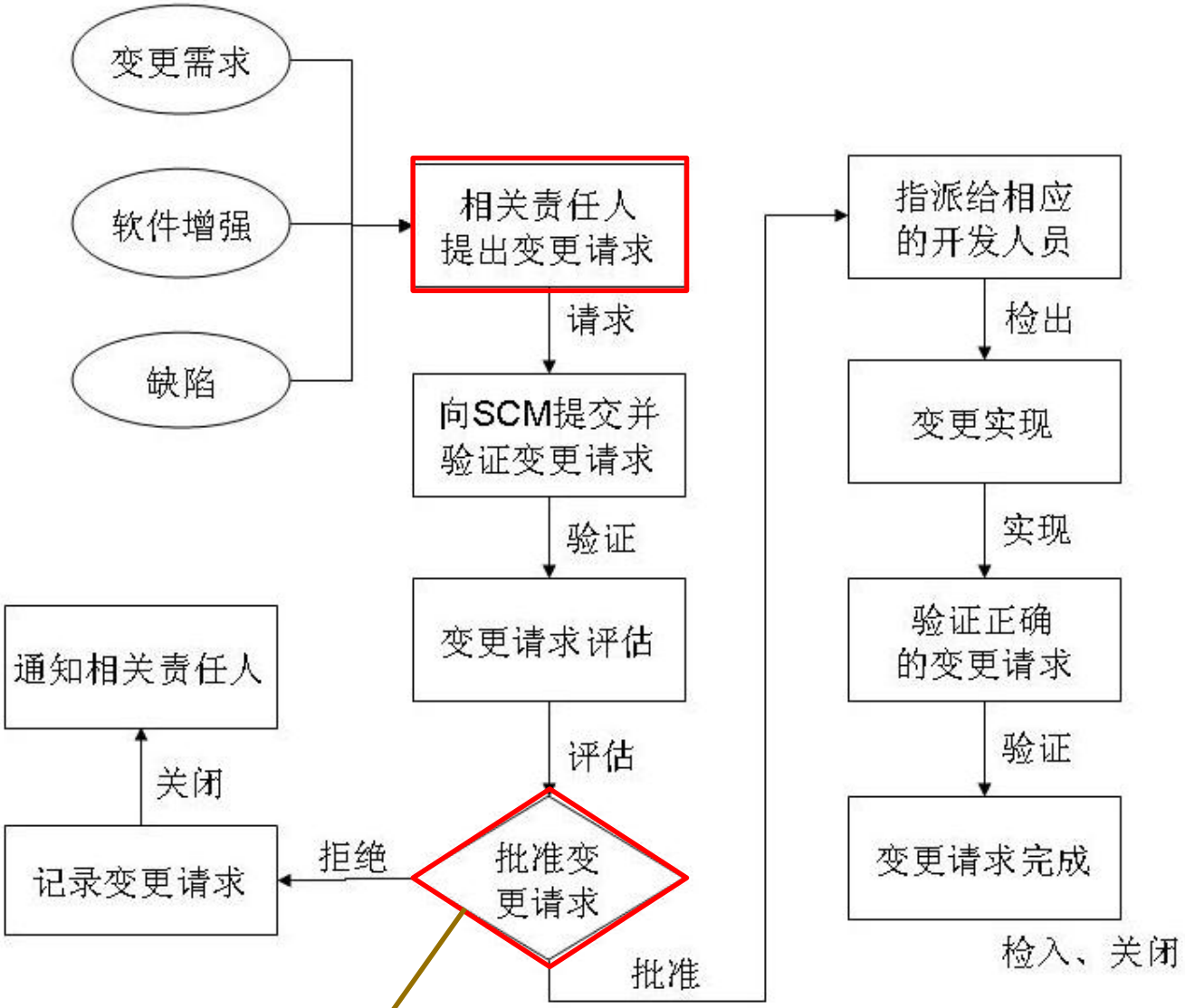
分支与合并

- 独立开发并修改同一个构件的结果是产生版本分支



- 如果所做的修改在代码的不同部位，则自动合并
- 如果在代码的相同部位做修改，则发生冲突，需要开发人员检查并解决冲突

变更控制



变更控制委员会（change control board, CCB）

修改记录

- 在序言性注释处增加构件的修改记录
- 编写脚本，扫描所有构件的修改记录，形成构件变更报告

```
// VIDEO project (XEP 6087)
//
// APP-SYSTEM/AUTH/RBAC/USER_ROLE
//
// Object: currentRole
// Author: Zeng Tao
// Creation date: 13/11/2015
//
// © HOHai University 2015
//
// Modification history
// Version Modifier   Date          Change        Reason
// 1.0     ZengTao    11/11/2015    Add header    Submitted to CM
// 1.1     QiRz       13/11/2015    New field     Change req. R23/02
```