


Shanghai Jiao Tong University




上海交通大学

# 软件工程

## Module10: 软件配置管理

上海交通大学计算机系



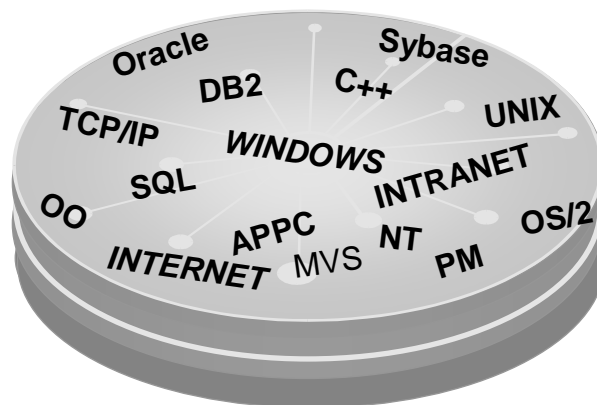
## 问题

- ◆ 开发人员在一种无法控制的状态下访问源代码
- ◆ 修改过的错误再次出现
- ◆ 集成过程拖延了产品投放市场的时间
- ◆ 多项目、多版本的管理与开发
- ◆ 同步和并行开发问题
- ◆ 是否能重构所有的Build?
- ◆ 人员流动引起知识资产的流失
- ◆ 由于管理不善致使未经测试的软件加入到产品中

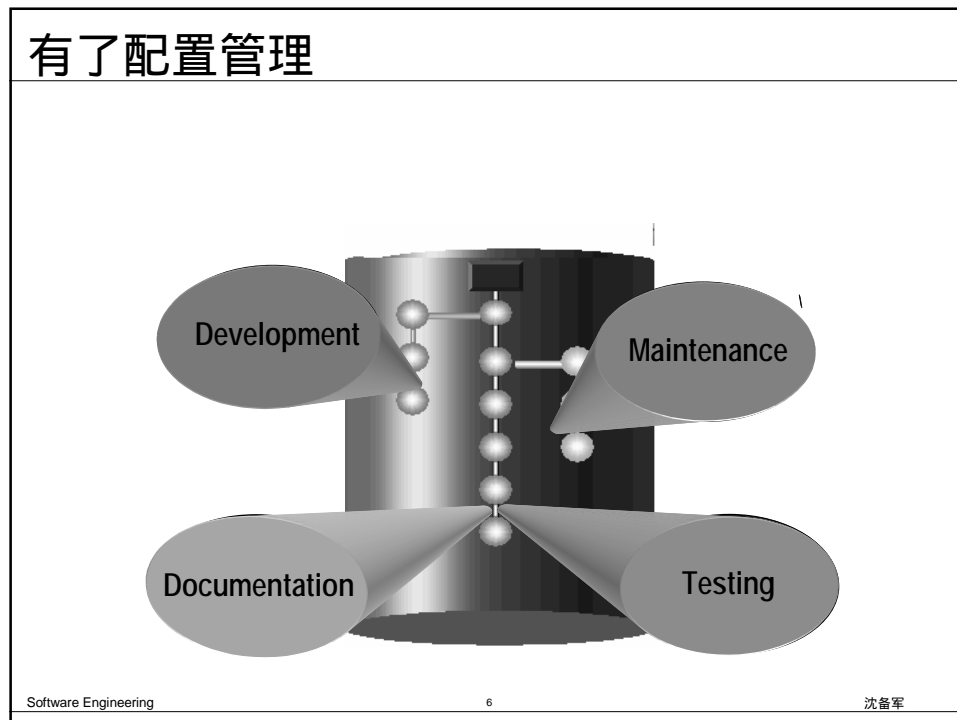
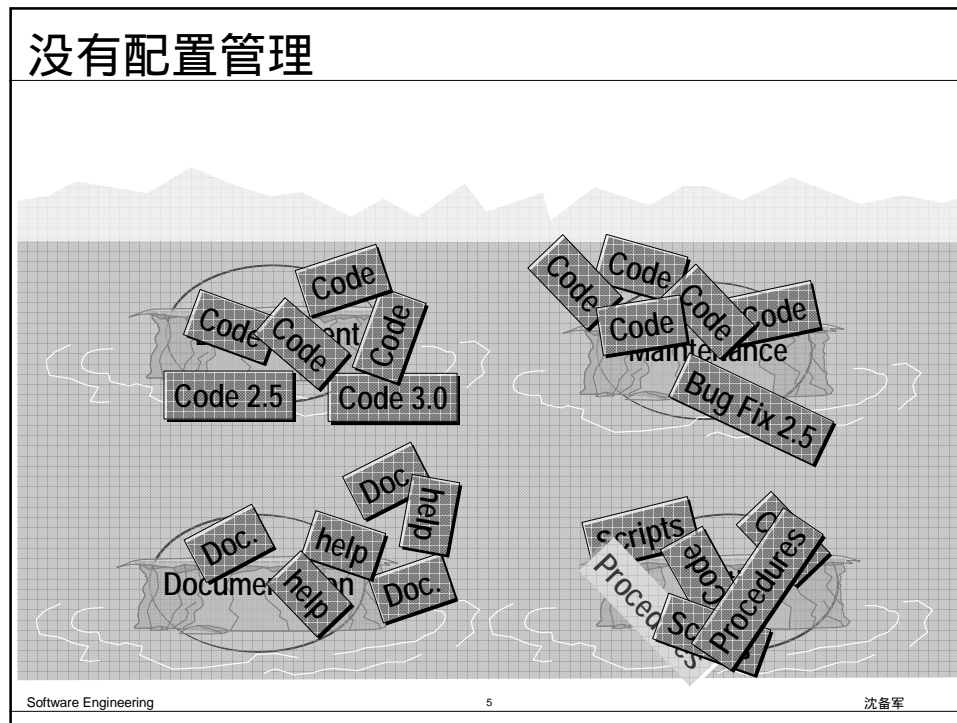
## 软件配置管理

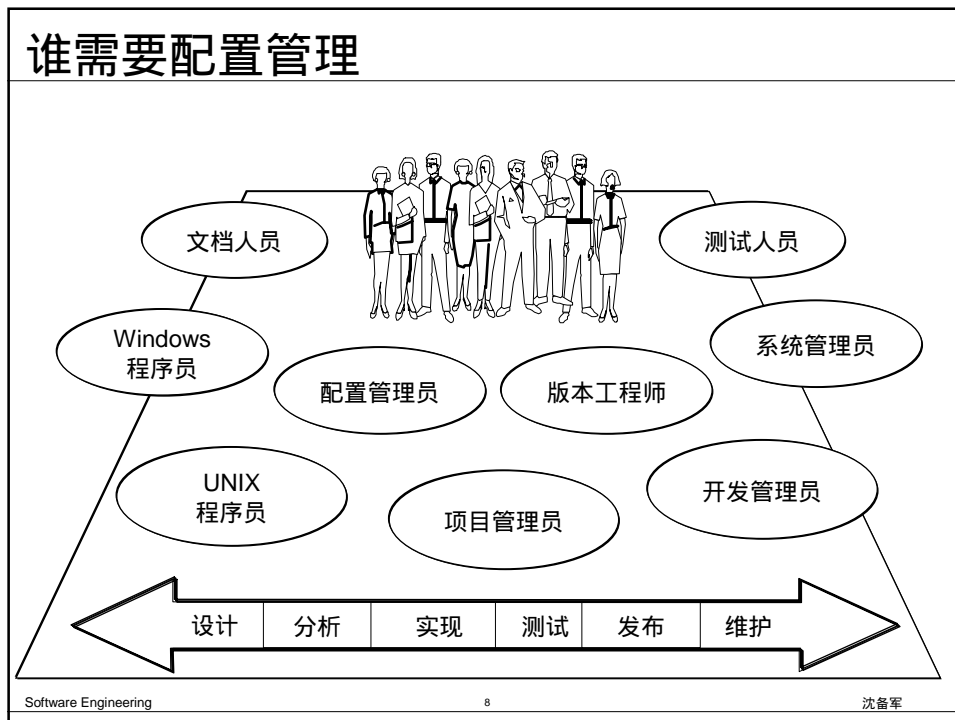
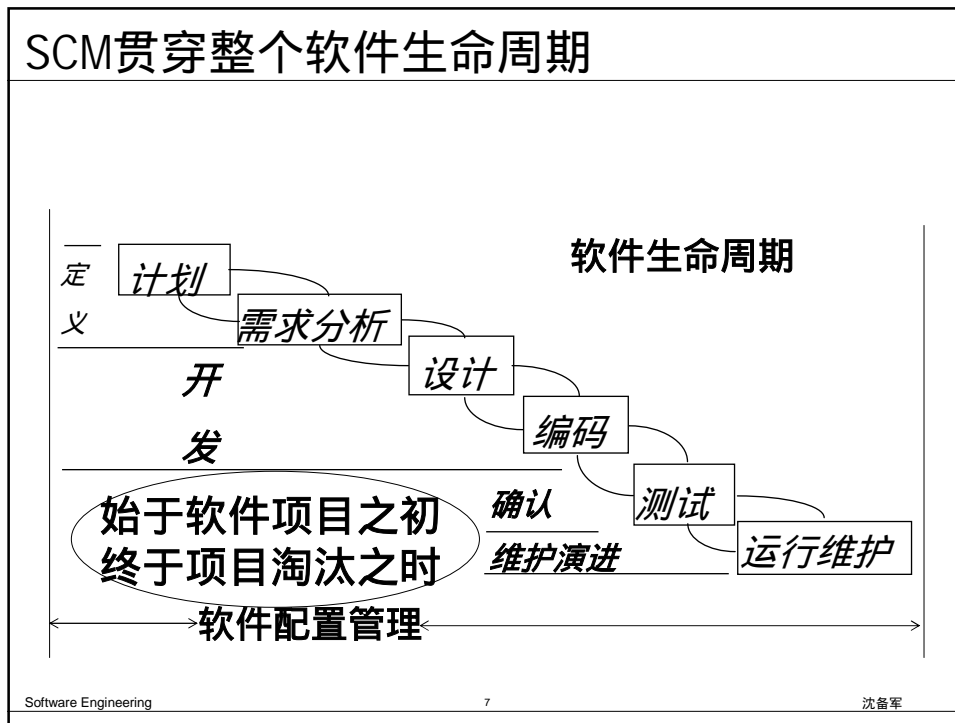
- ◆ 配置管理的必要性
- ◆ 配置管理的概念
- ◆ 配置管理的任务
- ◆ 微软每日编译

## 软件开发的复杂性



多操作系统  
多开发工具  
网络化  
团队方式  
异地开发



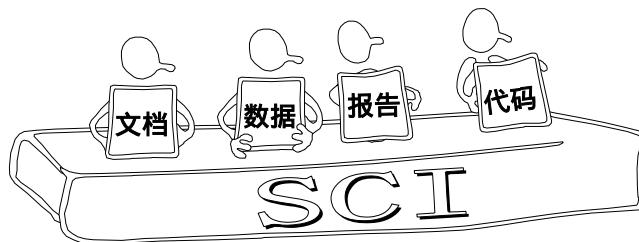


## 软件配置管理

- ◆ 配置管理的必要性
- ◆ 配置管理的概念
- ◆ 配置管理的任务
- ◆ 微软每日编译

## 软件配置的定义

软件配置是由在软件工程过程中产生的所有信息项构成的，它可以看作该软件的具体形态（软件配置项）在某一时刻的瞬间影像



## 软件配置管理的定义

“ 软件配置管理过程是在整个软件生存期中实施管理和技术规程的过程，它标识、定义系统中的软件项并指定基线；控制软件项的修改和发行；记录和报告软件项的状态和修改申请；保证软件项的完整性、协调性和正确性以及控制软件的存储、处理和交付。”

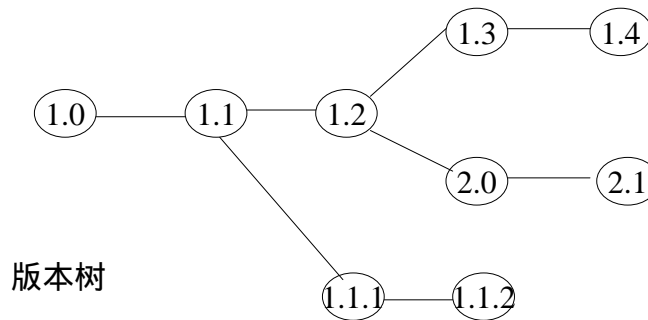
----- ISO/IEC 12207  
《信息技术 – 软件生存期过程》

## SCM是软件配置项的变更管理

- ◆ First Law of System Engineering
  - No Matter where you are in the system life cycle, the system sill change, and the desire to change it will persist throughout the life cycle.
- ◆ 变更主要来源：
  - 新的商业或市场条件，引起产品需求或业务规则的变化
  - 新的客户需要，要求修改数据、功能和服务
  - 企业改组或规模减小，导致项目优先级或软件工程队伍结构的变化
  - 预算或进度的限制，导致系统或产品的重定义
  - 纠正Bug，要求修改文档和代码

## 版本 version

- ◆ 表示一个配置项具有一组定义的功能的一种标识。随着功能的增加、修改或删除，配置项被赋予不同的版本号。一般在配置标识方案中给出版本标识方法。

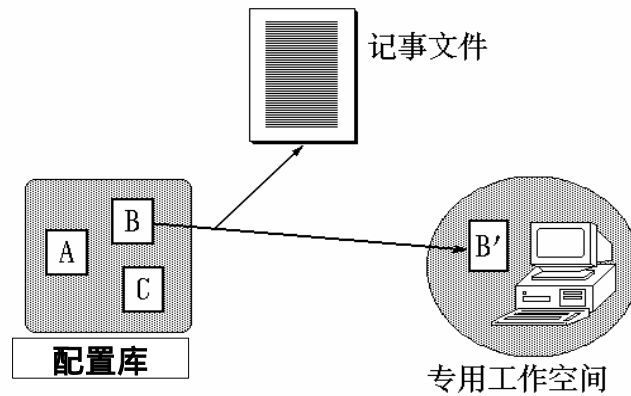


## 基线 baseline

- ◆ 通过正式的技术评审或测试的软件配置项才能成为基线。
- ◆ 成为基线后软件配置项必须严格控制变更。

## 配置库CM Repository

- ◆ 存储配置管理信息及软件配置项的版本信息。



## Check-in和Check-out

- ◆ 软件配置项通过检入(Check-in)，进入配置库，开始“冻结”
- ◆ 由于各种原因需要变更，从配置库中检出 (Check-out) 配置项
- ◆ check in和check out通过加锁协调多用户操作
- ◆ 每次check in时，在配置库上都会生成新的版本

## 软件配置管理

- ◆ 配置管理的必要性
- ◆ 配置管理的概念
- ◆ 配置管理的任务
- ◆ 微软每日编译

## 软件配置管理的任务



- 标识
- 版本控制
- 变更控制
- 配置审计
- 配置报告

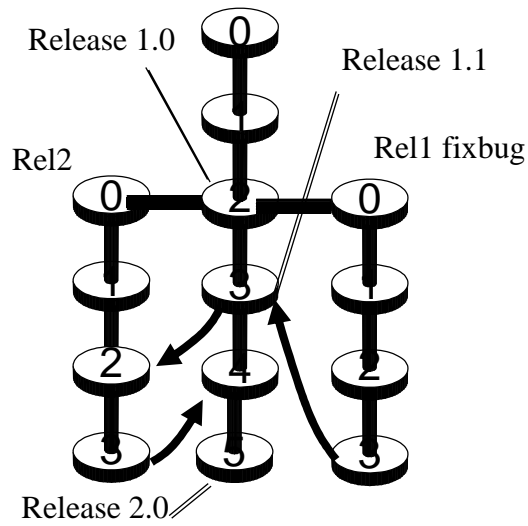
## 标识

- ◆ 每个配置项必须具有
  - 名字
  - 类型（如文档、程序、数据）
  - 项目标识符
  - 变更和版本信息
  - 相关配置及关系

## 版本控制

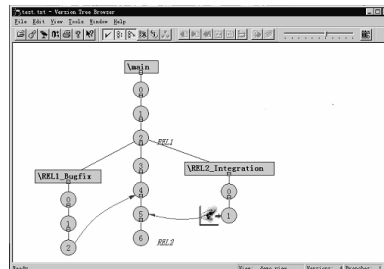
- ◆ 防止未授权的变更
- ◆ 保存文档的各个版本和版次
- ◆ 防止多个人同时修改文档 check-in/check-out
- ◆ 支持并行开发
- ◆ 支持基线

## 并行开发



## 版本控制工具

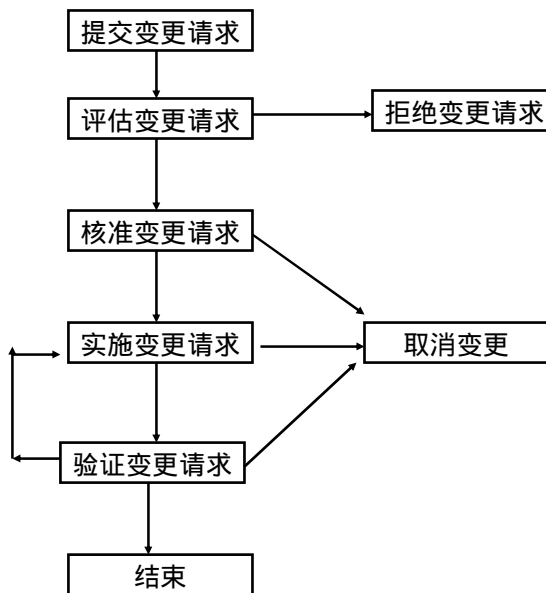
- ◆ Merant PVCS
- ◆ IBM Rational Clearcase
- ◆ Microsoft Visual Source Safe
- ◆ CVS
- ◆ ...

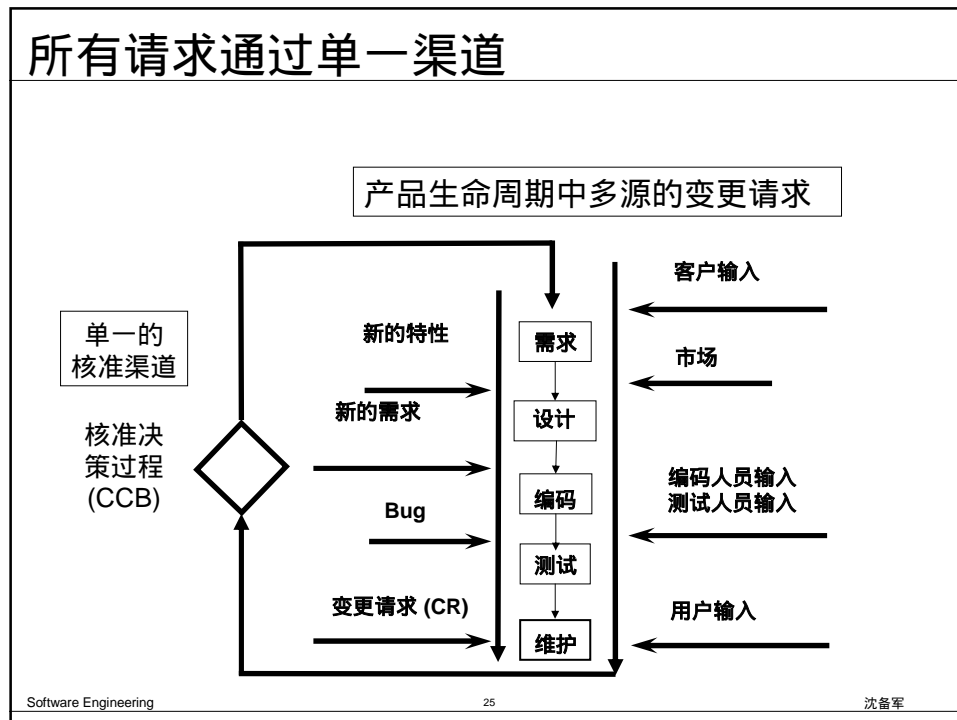


## 采用版本控制工具的最佳实践

- ◆ 合理组织项目及子项目结构
- ◆ 避免多人Check-Out
- ◆ 合理管理权限
- ◆ 避免长时间不Check-in以及不Get Last Version
- ◆ 避免强行修改未Check-Out的本地文件的Read-Only属性
- ◆ 建议代码Check-in之前需通过单元测试
- ◆ 每天或定期备份所有数据

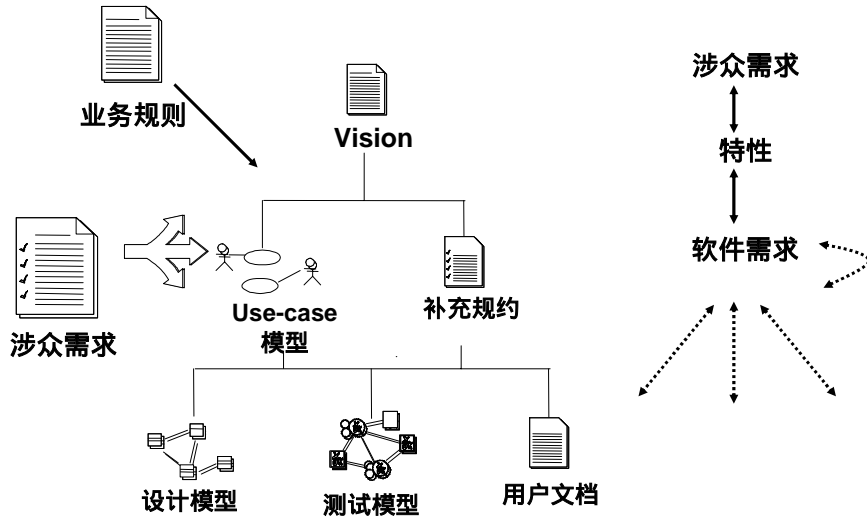
## 变更控制程序



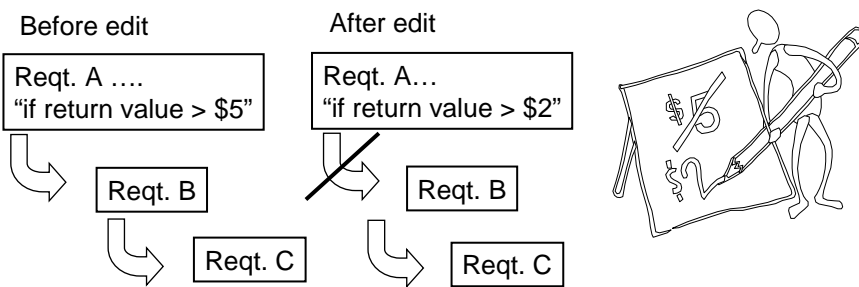


- ### 变更控制委员会 CCB
- ◆ 任务
    - 负责变更决策
    - 监督变更流程
  - ◆ 组成人员
    - 由所有利益方包括客户、开发人员、测试人员、配置管理人员和用户等的代表组成
    - 在小型项目中，项目经理或软件构架设计师一人即可担当此角色
- Software Engineering 26 沈备军

## 采用文档跟踪有助于管理变更

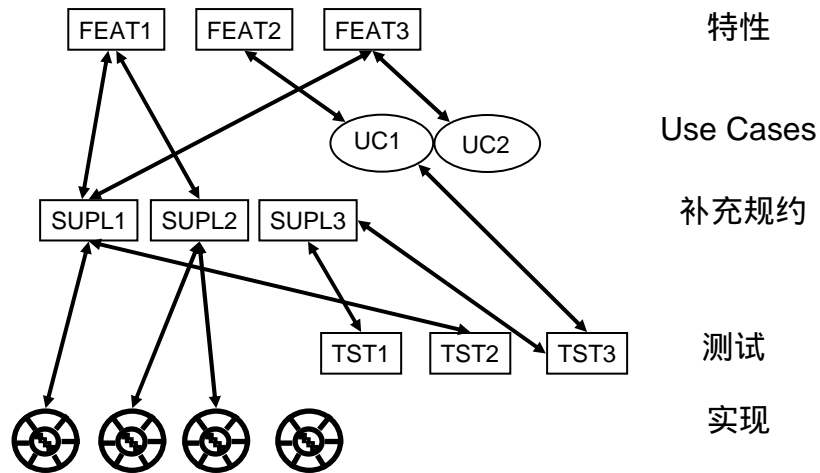


## 通过跟踪进行影响分析



- 变更后的文档其相关链接将标识为“suspect”
- Suspect 链接由用户处理

### 图中哪里错了?



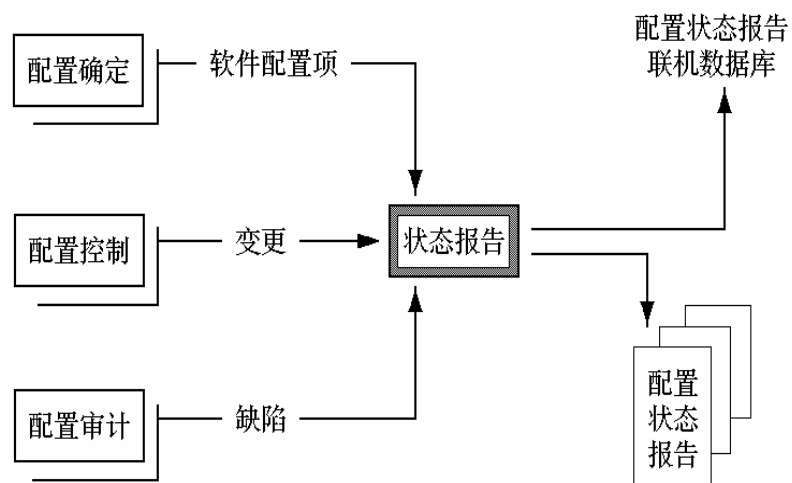
### 变更跟踪工具

- ◆ 加强开发团队与外界的沟通，用户、测试人员和市场销售人员可以直接通过Web来提交变更请求,并及时了解进展状况;
- ◆ 用数据库统一管理变更请求，避免遗漏和重复，并可以在此基础上，进行定量分析；
- ◆ 项目经理可以根据定量的数据来分配工作任务，并准确地掌握项目进度；
- ◆ 开发人员可以明确地了解他被分配的开发任务，并根据优先级依次完成；
- ◆ Rational ClearQuest等

## 配置审计

- ◆ 在ECO中说明的变更已经完成了吗？
- ◆ 加入了任意附加的修改吗？
- ◆ 是否已经进行了正式的技术评审，以评估技术的正确性？
- ◆ 是否适当地遵循了软件工程标准？
- ◆ 变更被highlighted吗？是否指出了变更的日期和变更的作者？配置对象的属性反映了变更吗？
- ◆ 是否遵循了标识变更、记录变更并报告变更的规程
- ◆ 所有相关的配置项被适当修改了吗？

## 配置状态报告



## 软件配置管理

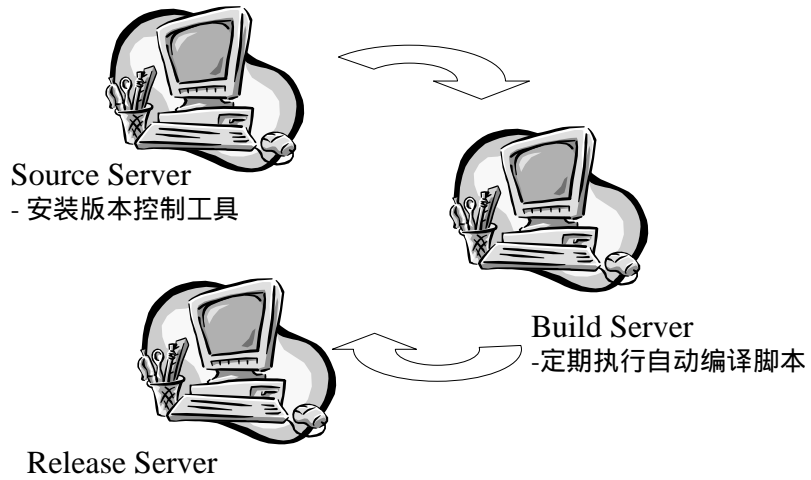
- ◆ 配置管理的必要性
- ◆ 配置管理的概念
- ◆ 配置管理的任务
- ◆ 微软每日编译



## 微软的每日编译

- ◆ 什么是每日编译
  - 每天都对所有的源代码进行一次完整的编译，生成一份可执行的产品程序
- ◆ 每日编译的目的
  - 统一的视图
  - 展示最新进展
  - 测试的基础

## 每日编译中的服务器



## 每日编译脚本的功能

- ◆ 获取最新代码
- ◆ 产生新版本号
- ◆ 编译
- ◆ 检查并发布编译结果
- ◆ 生成编译报告