

CHAPTER 9

Software Project Management

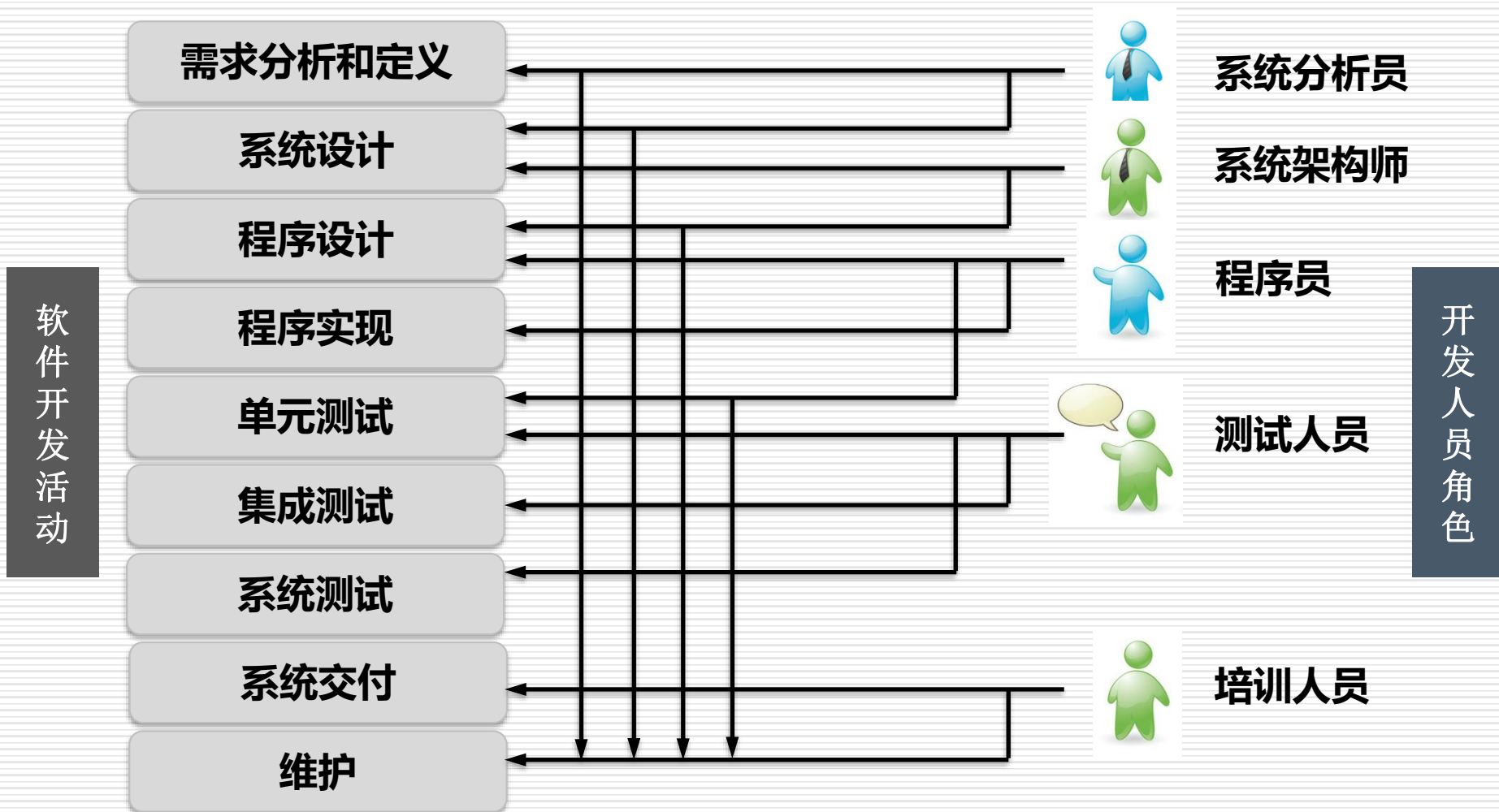
Outline

- ☐ **Estimating Cost**
- ☐ **Planning**
- ☐ **Team organization**
- ☐ **Configure management**
- ☐ **Management standards**

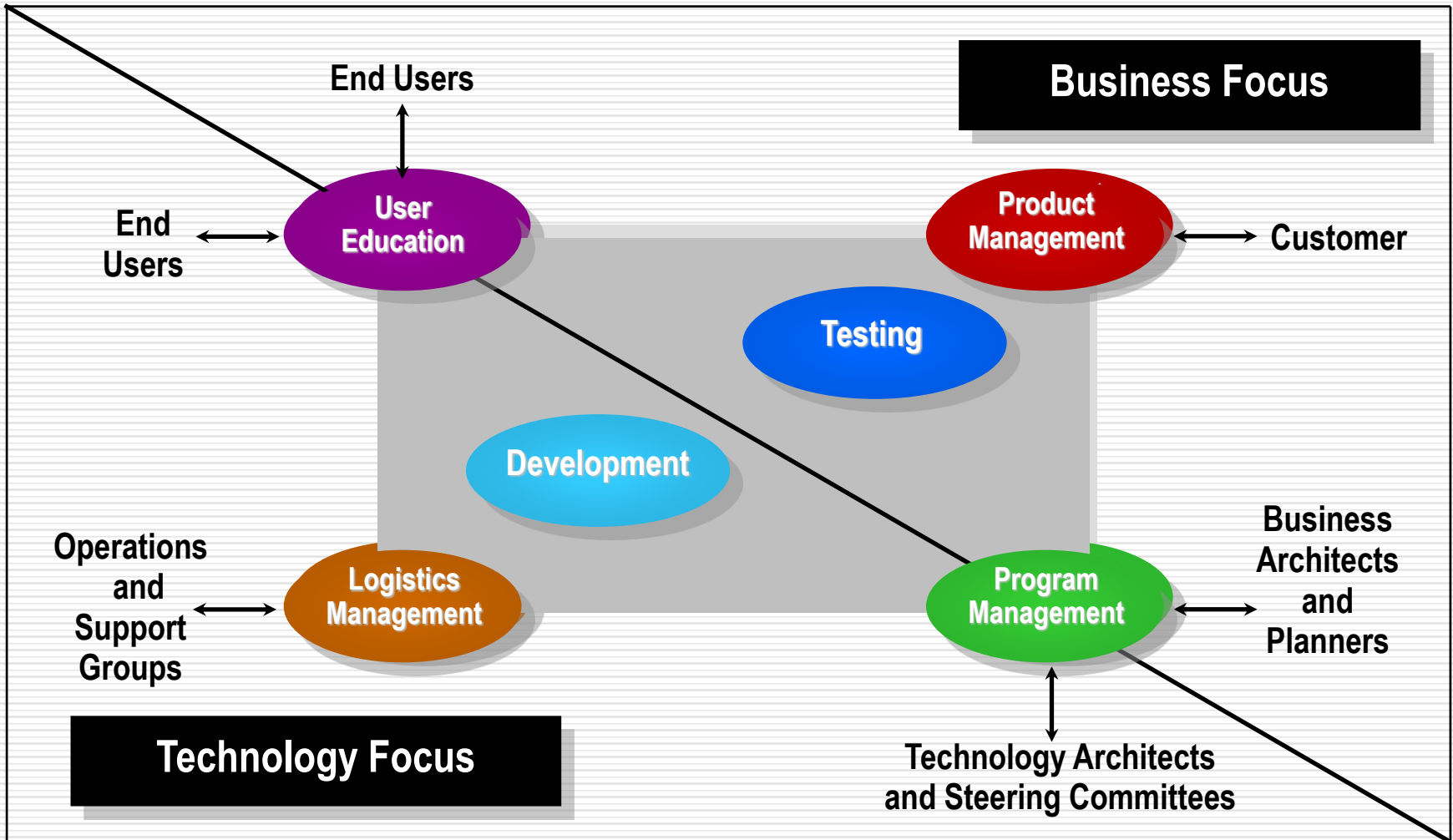
Today's Outline

- **Team and organization (管人)**
- **Communication (管交流)**
- **Configuration management (管配置)**
- **Management standards**
 - ✓ **ISO 9000**
 - ✓ **CMM**

团队的角色分工

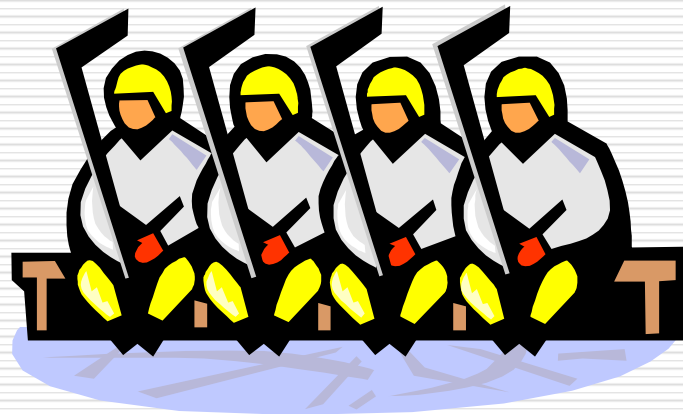
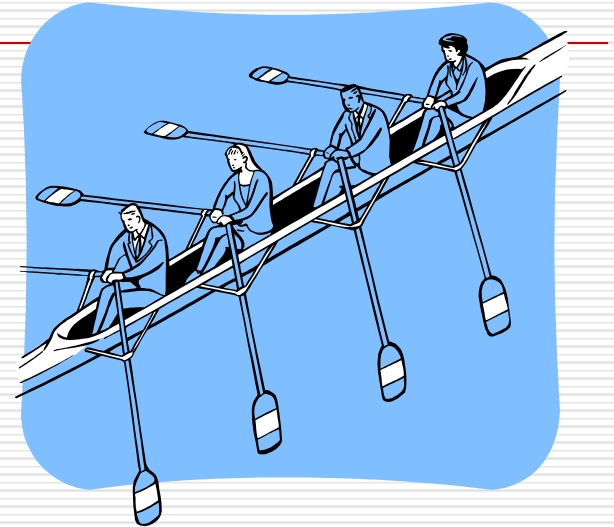


微软开发团队的角色分工



Selecting staff 选人

- People are the most important assets in SE.
- An important task of project management is team selection.



- Information comes from:
 - ✓ Information provided by the candidates.
 - ✓ Information gained by interviewing and talking with candidates.
 - ✓ Recommendations and comments from other people who know or who have worked with the candidates.

选人

案例描述

假设你是一个报警系统开发公司的项目经理，公司希望进入**养老院**市场，帮助独立生活的老年人和残疾人提供技术服务。你需要组建一个6人的开发团队，他们可以基于公司现有的报警处理技术开发新产品。

显然，你的首要任务是从公司内部或者外部选择合适的团队成员。



你如何选择团队成员？这样选择的理由是什么？

考察的因素

✓ **Application domain experience**

For a project to develop a successful system, the developers must understand the application domain. It is essential that some members of a development team have some domain experience.

✓ **Platform experience**

This may be significant if low-level programming is involved. Otherwise, not usually a critical attribute.

✓ **Programming language experience**

This is normally only significant for short duration projects where there is not enough time to learn a new language. While learning a language itself is not difficult, it takes several months to become proficient in using the associated libraries and components.

✓ **Problem solving ability**

This is very important for software engineers who constantly have to solve technical problems. However, it is almost impossible to judge without knowing the work of the potential team member.

Staff factors

✓ Educational background

This may provide an indicator of the basic fundamentals that the candidate should know and of their ability to learn. This factor becomes increasingly irrelevant as engineers gain experience across a range of projects.

✓ Communication ability

This is important because of the need for project staff to communicate orally and in writing with other engineers, managers and customers.

✓ Adaptability

Adaptability may be judged by looking at the different types of experience that candidates have had. This is an important attribute as it indicates an ability to learn.

✓ Attitude

Project staff should have a positive attitude to their work and should be willing to learn new skills. This is an important attribute but often very difficult to assess.

✓ Personality

This is an important attribute but difficult to assess. Candidates must be reasonably compatible with other team members. No particular type of personality is more or less suited to software engineering.

选人条件

- 牢固掌握计算机软件的基本知识和技能。
- 善于分析和综合问题，具有严密的逻辑思维能力。
- 工作踏实、细致，不靠碰运气，遵循标准和规范，具有严格的科学作风。
- 工作中表现出有耐心、有毅力、有责任心。
- 善于听取别人的意见，善于与周围人员团结协作，建立良好的人际关系。
- 具有良好的书面和口头表达能力。

人员的选择



应用领域经验

平台经验

编程语言经验

教育背景

解决问题能力

沟通能力

适应性

工作态度

个性

人员的选择



人员的选择



如果你负责的项目工作出现进度落后问题，
不同性格的人将会怎样与你交流？

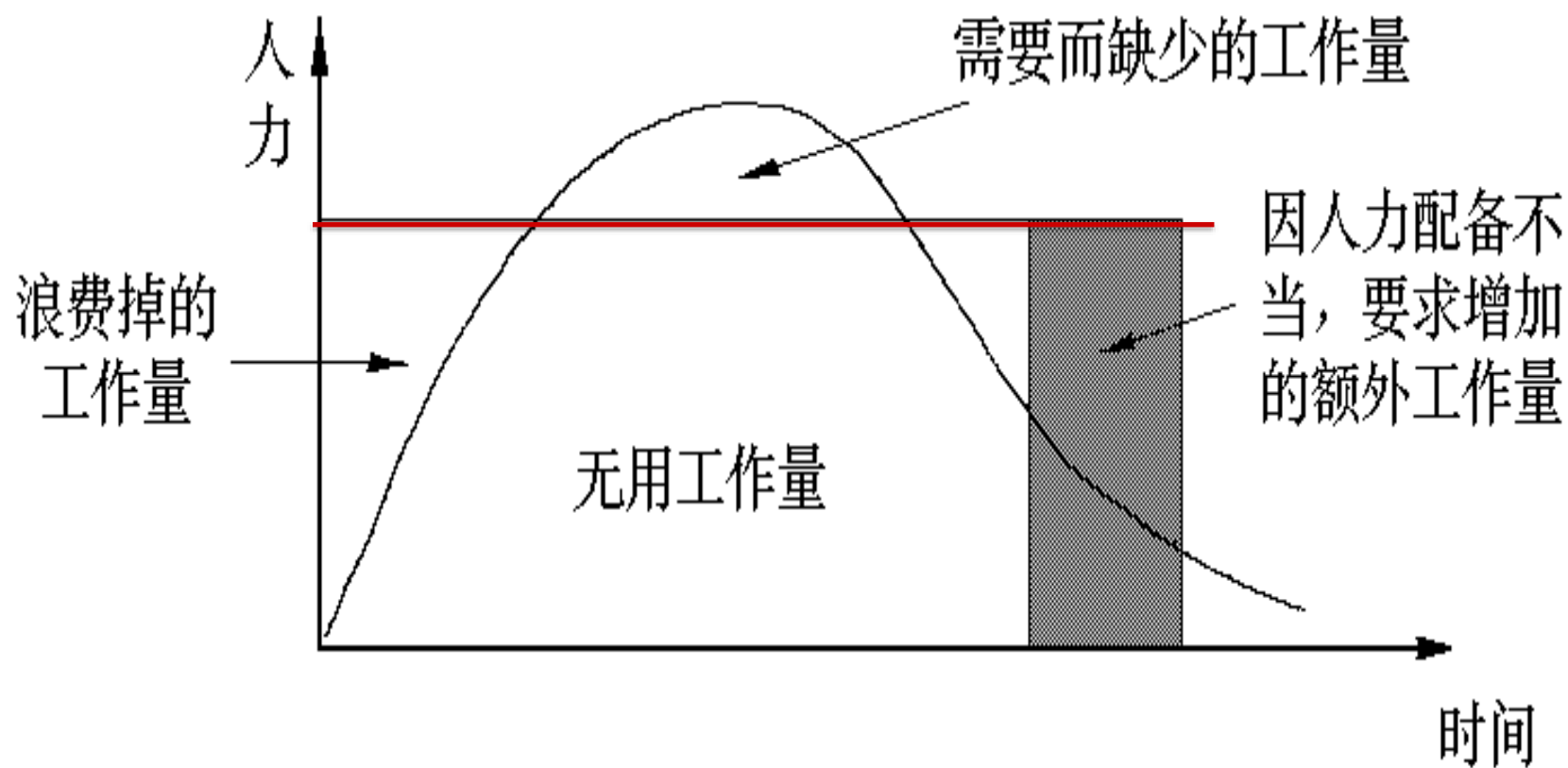
- ✓ **理性外向型：**告诉你什么时候必须完成工作，将会为你安排一个新的进度。
- ✓ **感性外向型：**告诉你什么时候必须完成工作，可能会提供一些建议让工作步入正轨。
- ✓ **理性内向型：**询问你什么时候可以完成，在分析自己意见时希望知道落后的原因。
- ✓ **感性内向型：**询问你什么时候可以完成，并问你能够帮你做些什么。

人员的选择



- ✓ 应该考虑团队中的技术、经验和个性是否**整体均衡**。
- ✓ 选择**性格互补**的成员组成的团队可能比仅仅根据技术能力选择成员的团队更有效率。
- ✓ 团队的领导力来自于成员的**尊重**，而不是名义上的头衔。

人员配备数量



人员配备

- ✓ 按此曲线，需要的人力随开发进展逐渐增加，在编码与单元测试阶段达到高峰，以后又逐渐减少。
- ✓ 如果恒定地配备人力，在开发初期将会有部分人力资源用不上而浪费掉。在开发中期，需要人力不够，造成进度的延误。在开发后期就需要增加人力以赶进度。
- ✓ 恒定地配备人力将浪费人力资源。
- **重质量：**软件项目是技术性很强的工作，要任用少量有实践经验、有能力的人员去完成关键性的任务。
- **重培训：**培养所需技术人员和管理人员是有效解决人员问题的好方法。
- **双阶梯提升：**人员提升应分别按技术职务和管理职务进行，不能混在一起。

人员配备

□ 对项目经理人员的要求

- ✓ 把用户提出的非技术性要求加以整理提炼，以技术说明书的形式转告给分析员和测试员。
- ✓ 能说服用户放弃一些不切实际的要求，以保证合理的要求得以满足。
- ✓ 能够把表面上似乎无关的要求集中在一起，归结为“需要什么”，“要解决什么问题”。这是一种综合问题的能力。
- ✓ 要懂得心理学，能说服上级领导和用户，让他们理解什么是不合理的要求。但又要使他们毫不勉强，乐于接受，并受到启发。

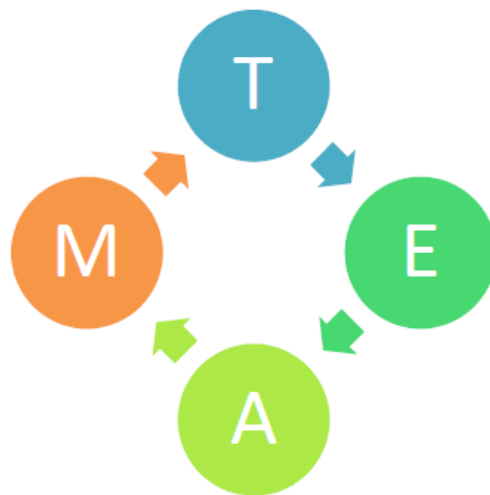
团队管理

- ☐ **Most software engineering is a group activity**
 - ✓ **The development schedule for most non-trivial software projects is such that they cannot be completed by one person working alone.**
- ☐ **Group interaction is a key determinant of group performance**
- ☐ **Flexibility in group composition is limited**
 - ✓ **Managers must do the best they can with available people.**

团队的概念

团队：由少量的人组成，具有互补的技能，对一个共同目的、绩效目标及方法做出承诺并彼此负责。

T	together	结合	众
E	everyone	大家	志
A	achieve	完成	成
M	Mission	任务	城



群体和团队

- 信息共享

- 中性（有时消极）

- 个体化

- 随机的或不同的

目标

协同配合

责任

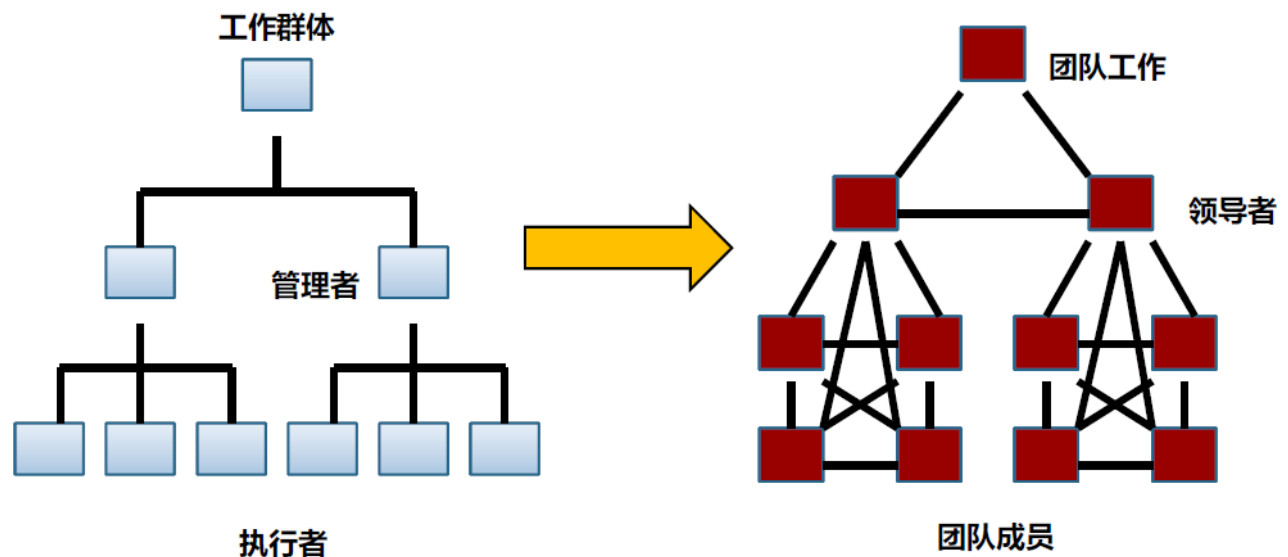
技能

- 集体绩效

- 积极

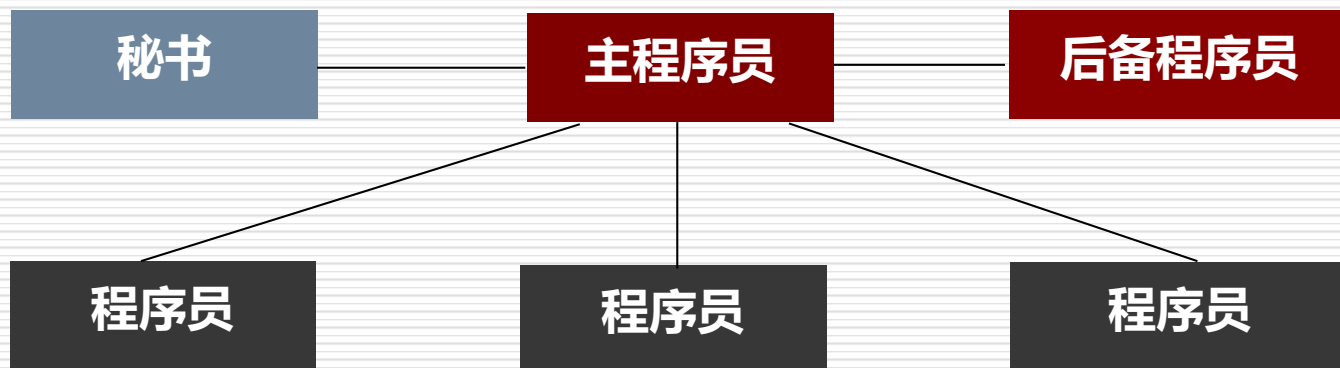
- 个体的或共同的

- 相互补充的



软件开发团队组织

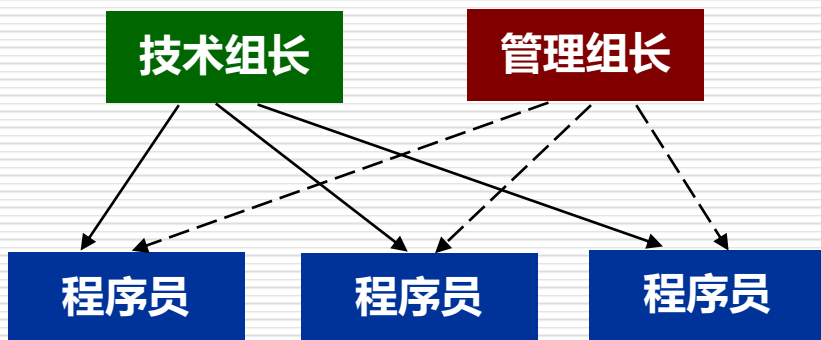
主程序员式组织结构：以主程序员为核心，主程序员既是项目管理者也是技术负责人，团队其他人员的职能进行专业化分工。



- **优点：**成员之间采取简单的交流沟通模式
- **缺点：**很难找到技术和管理才能兼备的主程序员

软件开发团队组织

矩阵式组织结构：将技术与管理工作进行分离，技术负责人负责技术决策，管理负责人负责非技术性事务的管理决策和绩效评价。



在这种组织中，明确划分技术负责人和管理负责人的权限是十分重要的。



微软开发团队组织

□ 微软开发团队的特点：

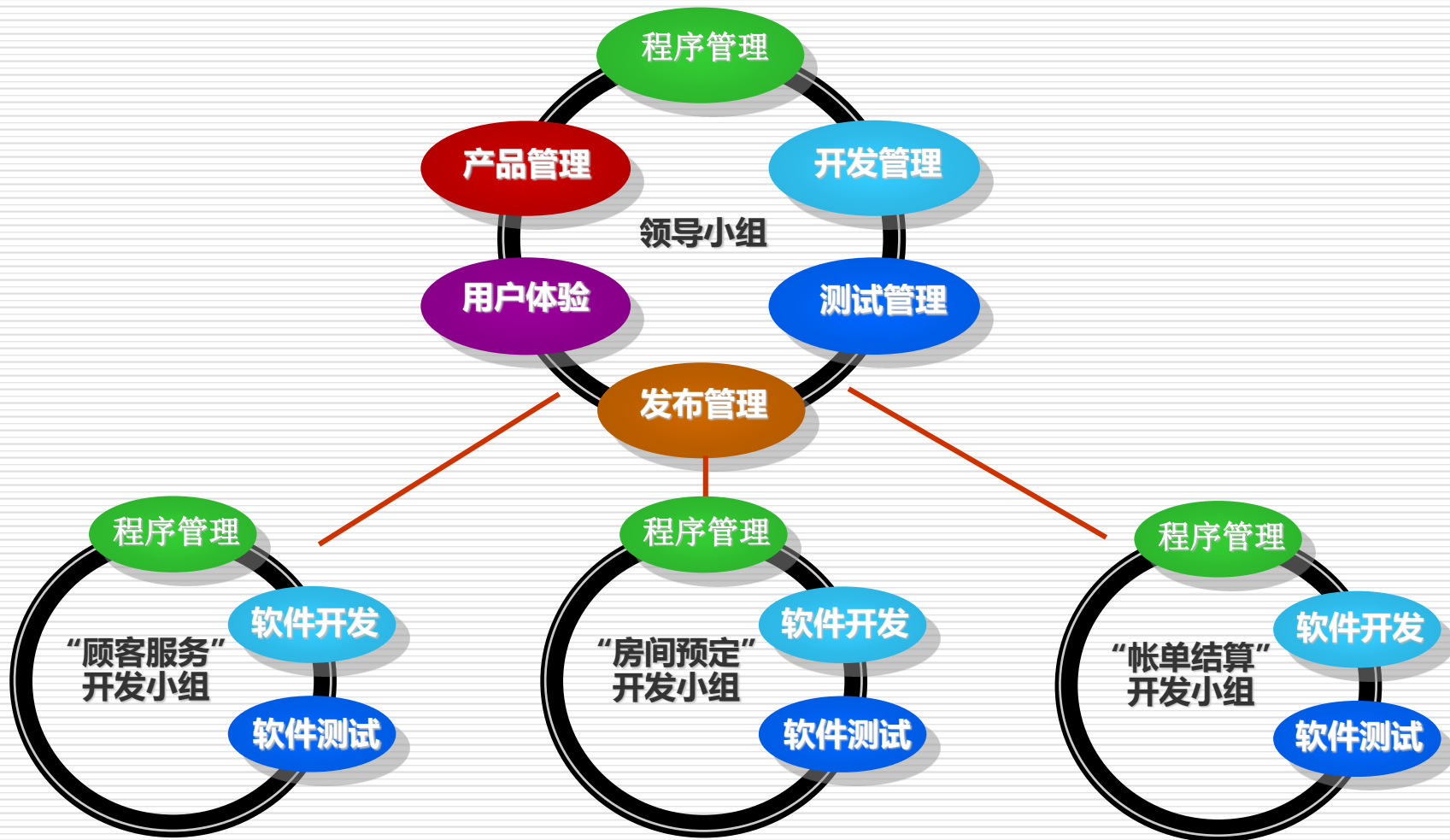
- ✓ 小型的、多元化的项目组织、对等的
- ✓ 相互依赖的角色与共同分享的职责
- ✓ 具备专深的技术水平和业务技能
- ✓ 具有强烈的产品意识，关注最终发布的软件产品
- ✓ 清晰的目标和远景
- ✓ 人人参与设计
- ✓ 项目组成员在同一地点办公
- ✓ 对于规模较大的项目，采取类似小型项目组的运作模式



Microsoft®

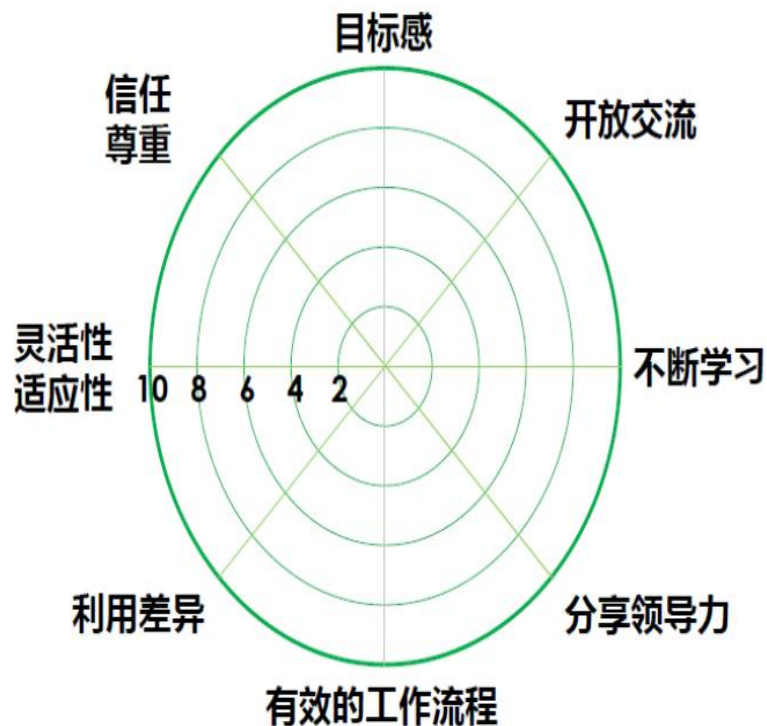


微软开发团队组织



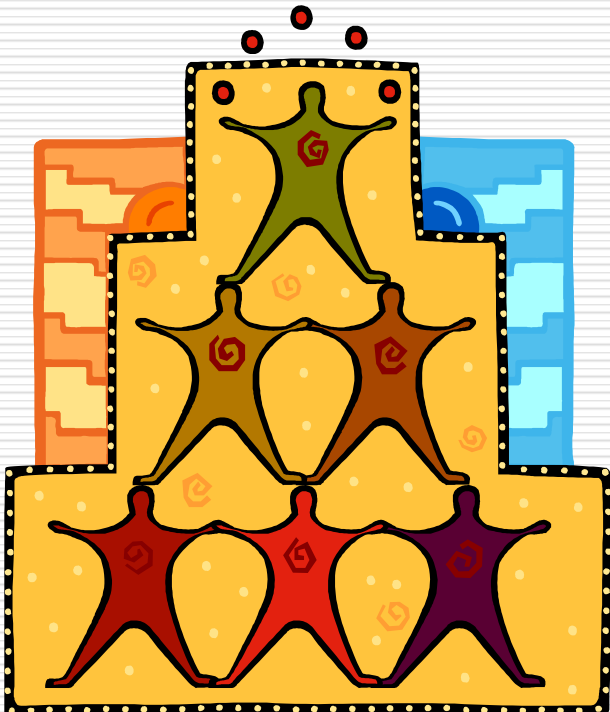
高效项目团队的特征

- ✓ 营造一种支持性的人力资源环境（沟通，交流，团队学习）
- ✓ 团队成员的自豪感
- ✓ 让每一位成员的才能与角色相匹配
- ✓ 设定具有挑战性的团队目标
- ✓ 正确的绩效评估



Group organization

- Small software engineering groups are usually organised informally without a rigid structure.



- For large projects, there may be a hierarchical structure where different groups are responsible for different sub-projects.

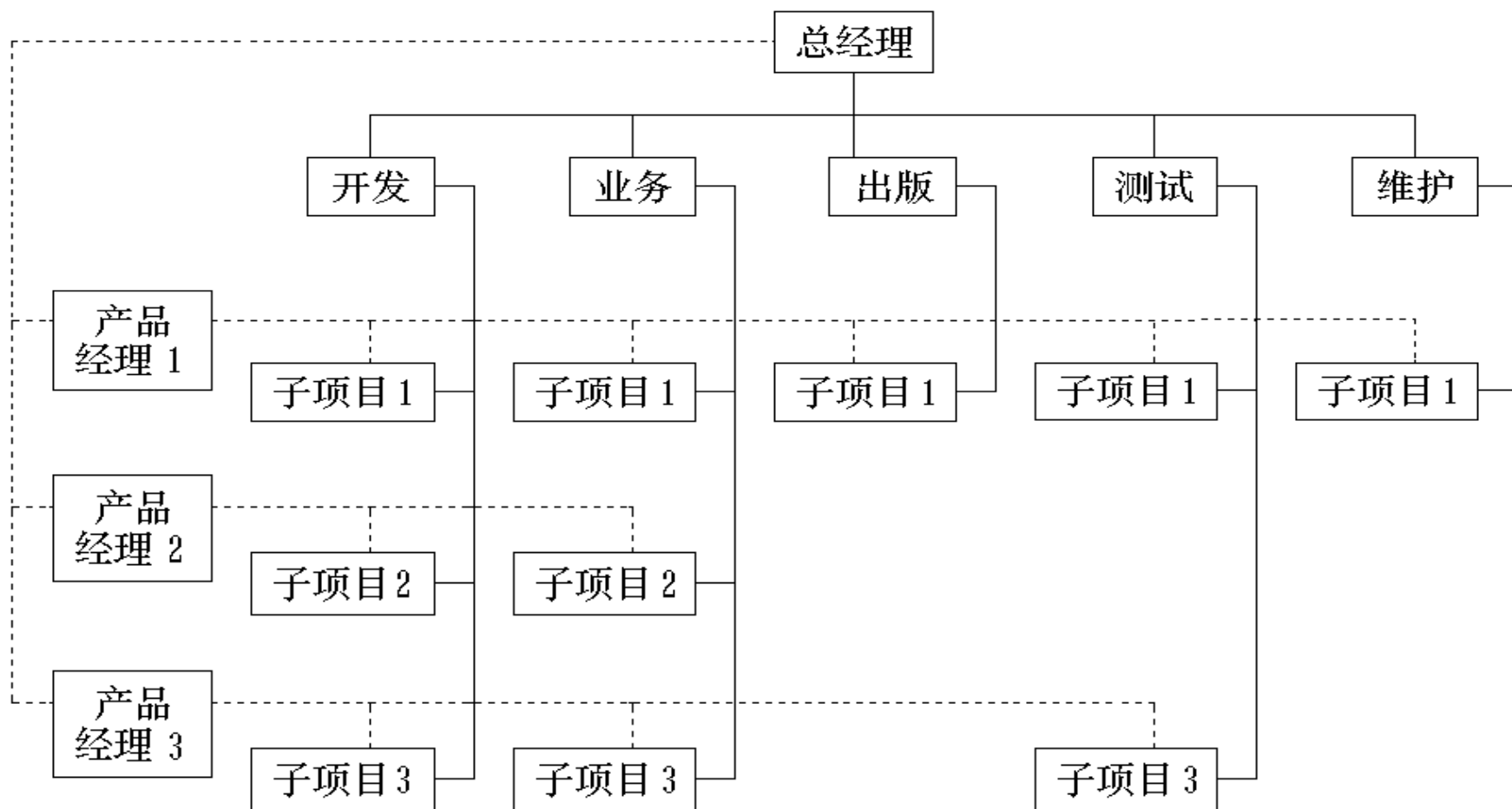
What is an organization?

An organisation is

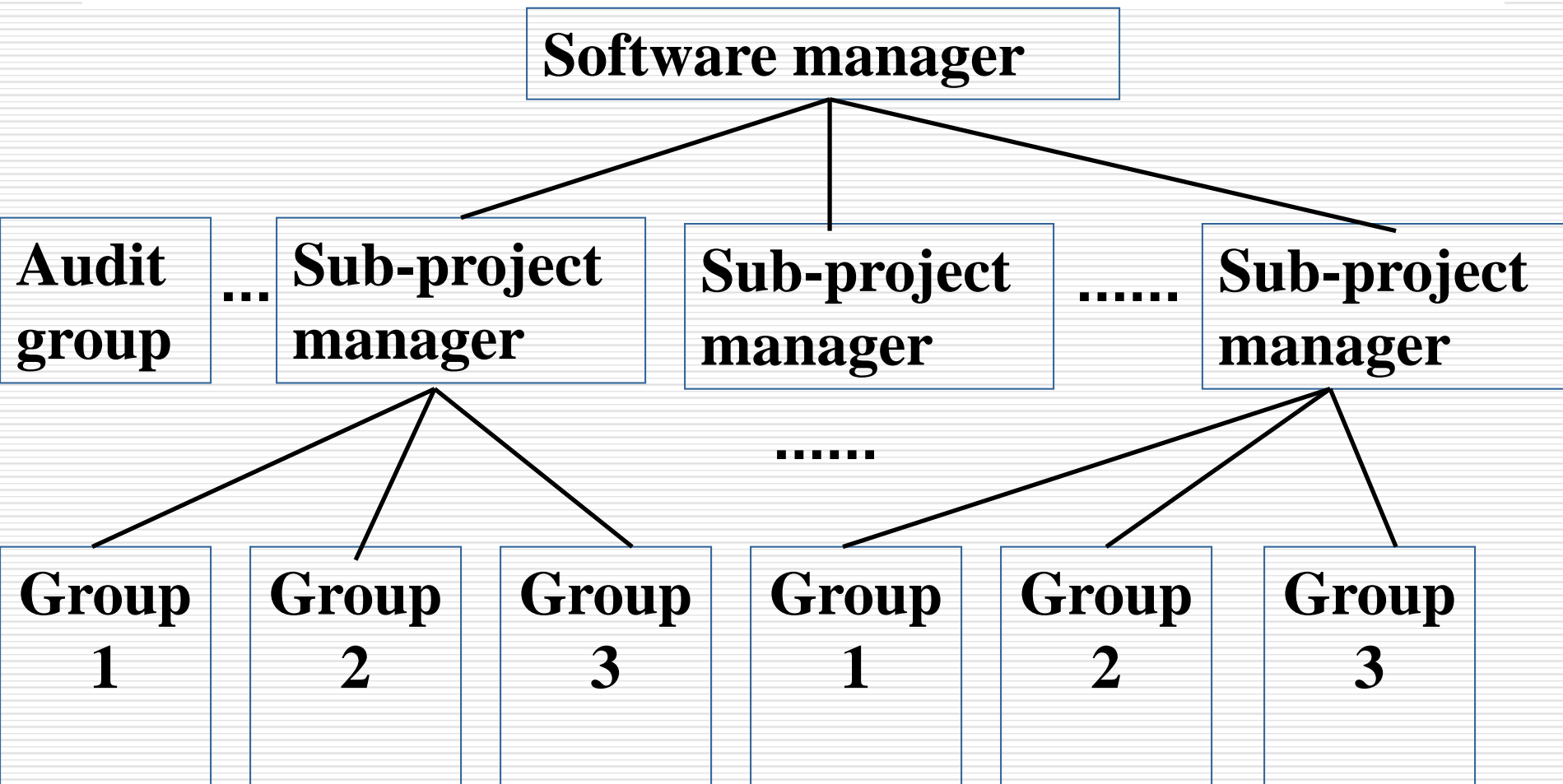
a group of resources established for a particular purpose.

A person (group of people) recognizes a need for a particular product/service and an organization is established to meet that need.

Organizational hierarchy



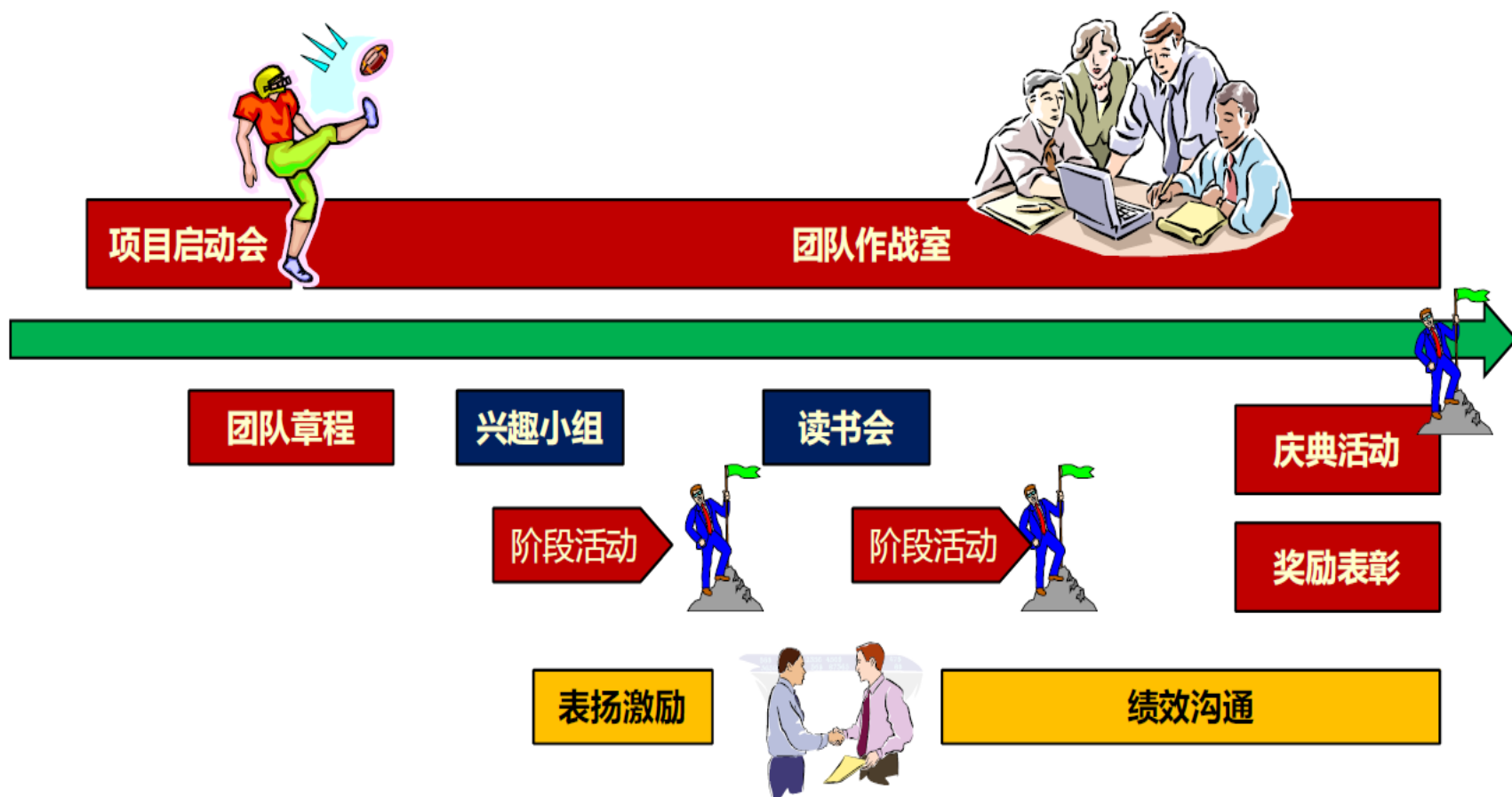
Software organization



A manager often directly manages **6~9** persons.

A team often consists of **30~50** persons.

团队建设活动



沟通和交流的概念

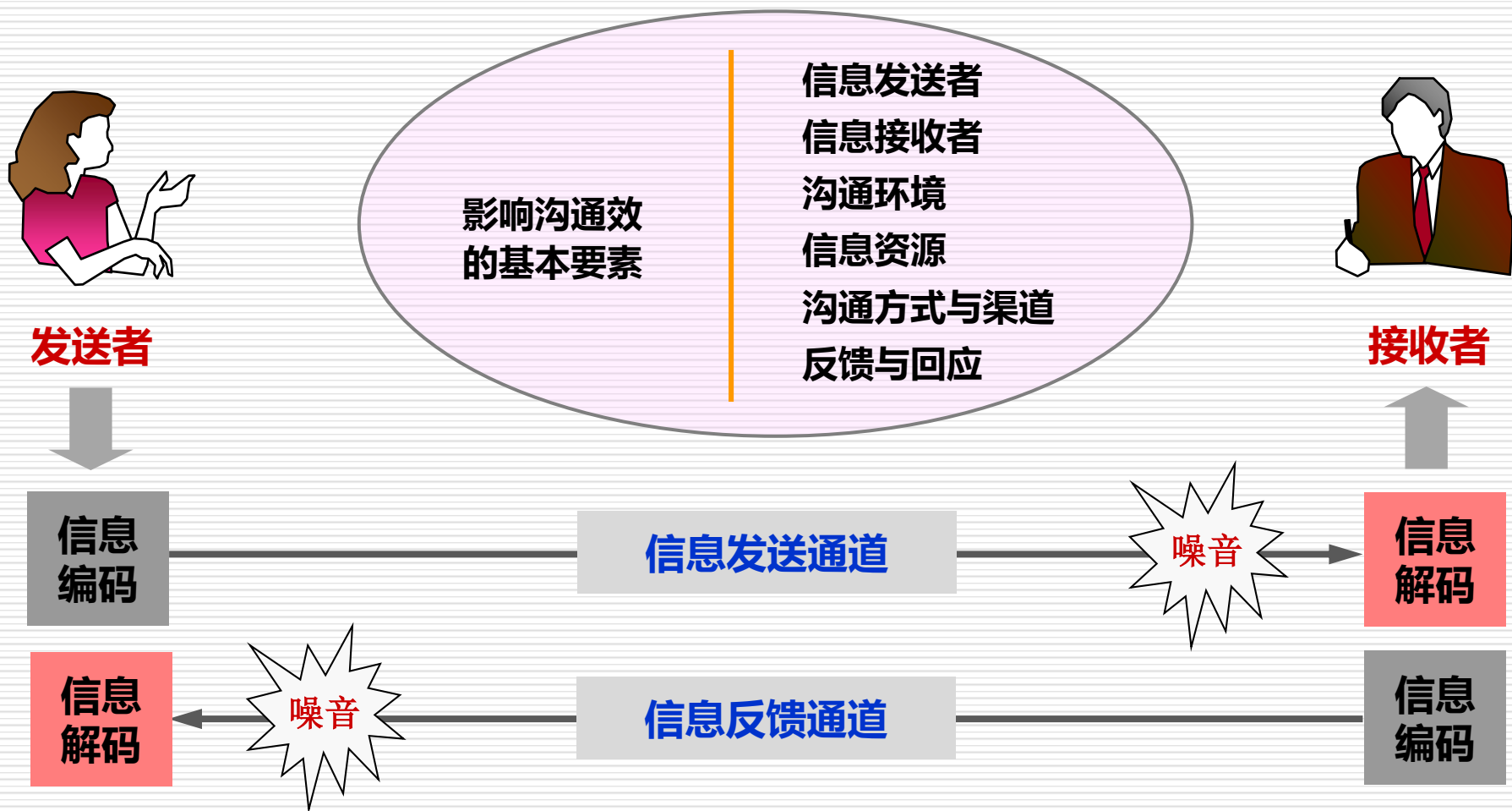
沟通是为了达到一定的目的，将信息、思想、情感在个人或群体之间进行传递或交流的过程。



- ✓ 沟通就是相互理解
- ✓ 沟通就是提出和回应问题与要求
- ✓ 沟通交换的是信息和思想
- ✓ 沟通是一种有意识的行为

“沟通是你被理解了什么，而不是说了什么”

沟通的模型



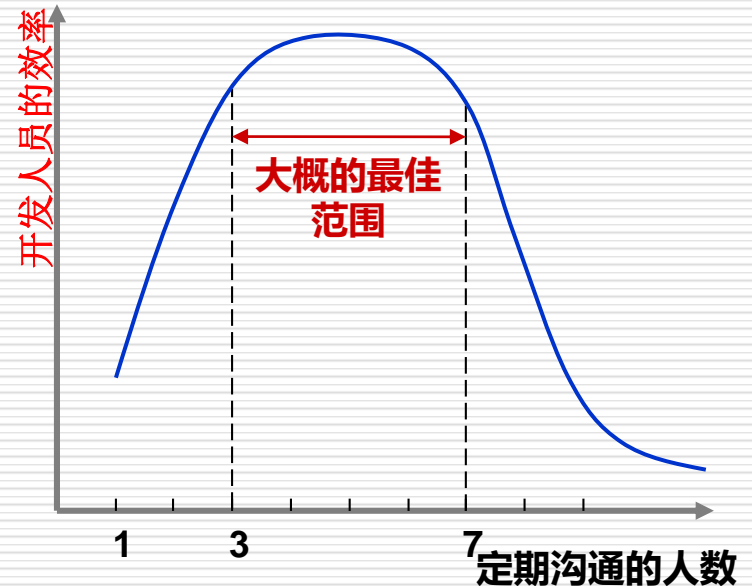
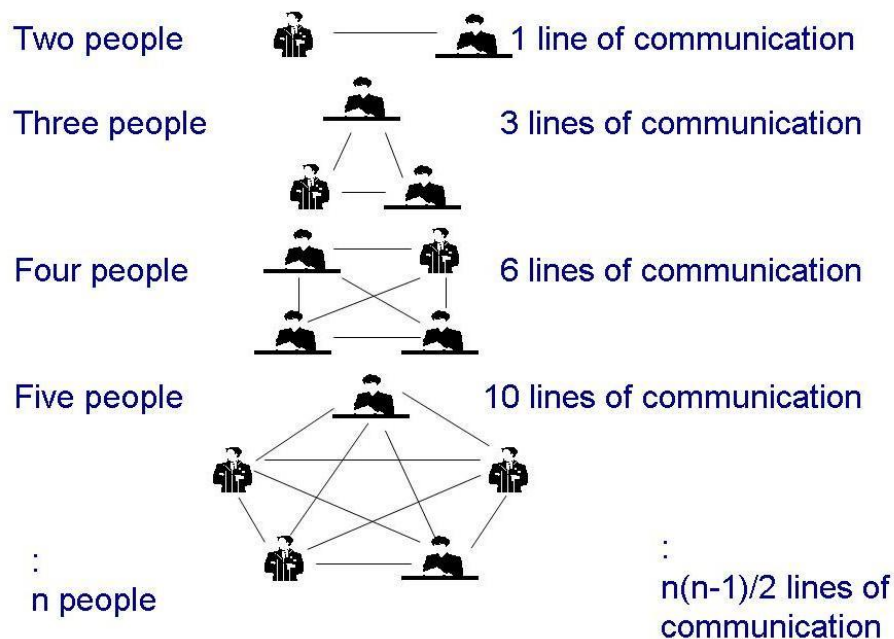
沟通的复杂性

软件开发是一项错综复杂关系的工作，沟通与交流的工作量极大。

- **举例一：** 1个农夫可在10天内采摘完一块草莓地，那么同样的草莓地是否可以用10个农夫可在1天内采摘完 ？
- **举例二：** 1头大象需要孕育22个月才能生下1头小象，那么增加大象的数量是否可以加快这个过程呢？
- **举例三：** 假设开发某个模块需要2人月的工作量，那么是否可以认为2个程序员在1个月内就可以完成这个模块的开发 ？

沟通的复杂性

Brooks法则： 向一个进度延迟的软件项目中增加人员可能会使其进度更加推迟。



常用沟通方式

口头	<ul style="list-style-type: none">✓ 正式的：演讲、报告、汇报、谈判、会议✓ 非正式：谈话、电话、打招呼
书面	<ul style="list-style-type: none">✓ 正式的：合同、报告、会议纪要、报表、备忘录✓ 非正式：笔记、便条
非语言	<ul style="list-style-type: none">✓ 正式的：手语、信号灯、音乐✓ 非正式：表情、声调、拥抱、握手
工具	<ul style="list-style-type: none">✓ 电话、传真、手机、微信、Email、面对面、协作工具

思考讨论

你认为以下情形应该选择什么合适的方式？

- ✓ 公司主管副总经理通知你，说他想详细了解项目进展情况，并且想在从今天开始的一周后与你见面。
- ✓ 客户在检查项目阶段成果时，指出上个月曾经提出的某个产品特性没有包含在其中，并抱怨早就以口头方式反映给了项目组成员，但作为项目经理的你却一无所知，而那位成员解释说把这件事忘记了。
- ✓ 你负责的一个项目准备在月底召开项目工作汇报会议，需要通知所有与会代表。
- ✓ 你作程硕士研究生，即将开始论文工作，你需要经常与导师进行联系和交流。
- ✓ 你作本科毕业生，即将开始毕业设计，你需要经常与导师进行联系和交流。

沟通技巧

自然赋予我们人类一张嘴、两只耳朵，也就是让我们多听少说。

—— 苏格拉底



- ✓ 努力做一个百分之百的听众
- ✓ 善于抓住关键点
- ✓ 努力排除干扰
- ✓ 培养分清主次的能力
- ✓ 通过提问和行为表示赞同与鼓励
- ✓ 对所听的内容进行及时反馈

项目沟通活动

项目组内的沟通：

- ✓ 任务分配清晰
- ✓ 项目组成员的四个主要沟通需求：职责、协调、状态、授权
- ✓ 会议：项目启动会、成员进度汇报、项目进展会
- ✓ 设置沟通期望
- ✓ 及时、公开、恰到好处



管理层和客户之间的沟通：

- ✓ 谁、为什么需要信息？
- ✓ 需要什么类型的信息？何种详细程度？频率如何？
- ✓ 当你与管理层或客户沟通时，你的目标是什么？采用什么样的方法来沟通？

项目会议

第一次项目团队会议（至关重要）

➤ 目标

- ✧ 项目概况：范围与目标、总体进度、方法和程序
- ✧ 确定项目人员的角色和任务
- ✧ 确立团队的工作模式

➤ 形式

- ✧ 重大项目：精心准备、集中1-2 天；前期介绍与建立基本规则
- ✧ 一般项目：简单有效；回顾项目范围与成员互相自我介绍

➤ 建立基本规则

- ✧ 计划决策、追踪决策、管理变动决策、关系决策



项目会议

项目阶段进展会议（每月一次）

- ✓ 向项目干系人和高层管理者汇报项目进展
- ✓ 解决需要高层管理者支持的问题



项目组周例会（每周一次）

- ✓ 明确短期目标，制定具体工作计划
- ✓ 协调资源需求
- ✓ 解决项目组工作中发生的任何问题



项目组内部会议（每天一次）

- ✓ 通报项目组成员的工作进展
- ✓ 确定项目组需要向上汇报并解决的问题

召开有效的会议

三星公司开会的三个原则：

1. 周三不开会；
2. 会议时长1小时，最多不超过1.5小时；
3. 将会议的内容整理成一张纸。

三星公司开会的七项规定：

1. 严格遵守时间；
2. 在会议材料中写明投入会议的经费；
3. 参加者限制为必要的合适人选和负责人，会议规模尽可能小；
4. 明确会议目的；
5. 事先分发会议资料；
6. 要让所有参加者发言；
7. 尽可能地减少会议记录，仅记录决定了的事项并进行保存。



演讲与表达的艺术

演讲是一种面对面向观众表达自己的观点、展示自己的产品或研究成果的有效手段。



演讲 = “讲” + “演”

语言技巧：7%

演讲使用的句式、语法、用词等

语音技巧：38%

演讲者的嗓音、音域、音调、语音的吸引力、通过声音传达出来的可信度等

形象技巧：55%

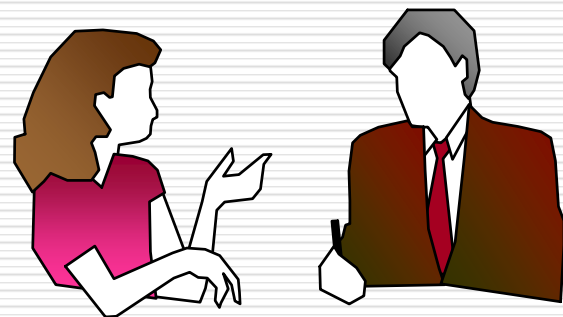
演讲者的外观、衣着、手势、眼神等肢体语言

语言的表达

能够用最简单的语言把一件事情说清楚，那才是最高的境界。

语言要简短、明确

- ✓ 避免使用冗长的句子
- ✓ 演讲与写作不同
- ✓ 力图在最简洁的语句中清晰地表达自己的思路



演讲的准备

演讲之前要有充分的准备

- ① 演讲的背景：主题、听众身份、人数等
- ② 组织演讲材料：开门见山，充分阐释，篇末点题
 - ✓ 演讲要有中心主题
 - ✓ 要有一个简短、有力、吸引人的开头
 - ✓ 要在主题部分充分表达自己的思想，所有内容为中心主题服务
 - ✓ 结尾要有力：归纳概括、鼓励行动或唤起情感
- ③ 制作幻灯片
- ④ 演讲彩排：在演讲之前最好先练习，最后一分钟准备

制作幻灯片

幻灯片只是演讲的辅助工具：

- ✓ 力求简单，每一页不超过 7 行，字号要大且清晰
- ✓ 一定不要照着幻灯片内容念，应该用脑“讲”
- ✓ 把复杂的信息内容组织得有条不紊，充分利用图形



使用演示程序或多媒体信息：

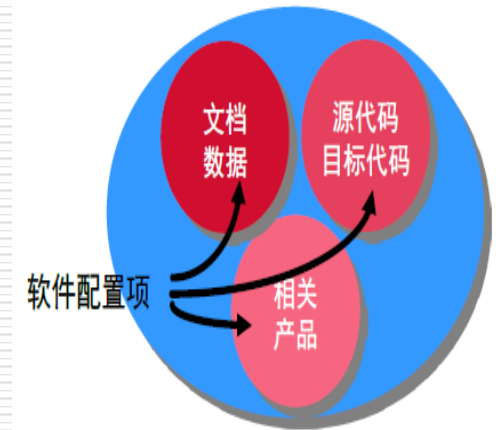
- ✓ 先给人看一个比较好的演示或图片，引起人们的兴趣
- ✓ 然后给比较差的，之后越来越好
- ✓ 最后一定给出最好的演示或图片



Software Configuration Management

Software Configuration

- programs
- documents
- data



Identification

**Version
Control**

Version Change

Auditing Reporting

“Make” Facility

软件配置管理

- 对软件产品进行标识、组织、存储、更改和发放，记录报告其状态，验证软件产品的正确性和一致性，并对上述工作的审计的过程。
 - ✓ 如何标识软件产品
 - ✓ 如何描述软件产品？
 - ✓ 如何对其软件产品的版本进行控制？
 - ✓ 如何控制软件产品的变更？
 - ✓ 如何制定软件配置计划？
 - ✓ 如何利用工具支持软件配置活动……

软件配置项

❑ **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。

- 计算机程序（源代码和可执行程序）；
- 描述计算机程序的文档（供技术人员、用户使用）；
- 数据（程序内包含的、或在程序外的）。

为了开发出高质量的软件产品，软件开发人员不仅要努力保证每个软件配置项正确，而且必须保证一个软件的所有配置项是完全一致的。

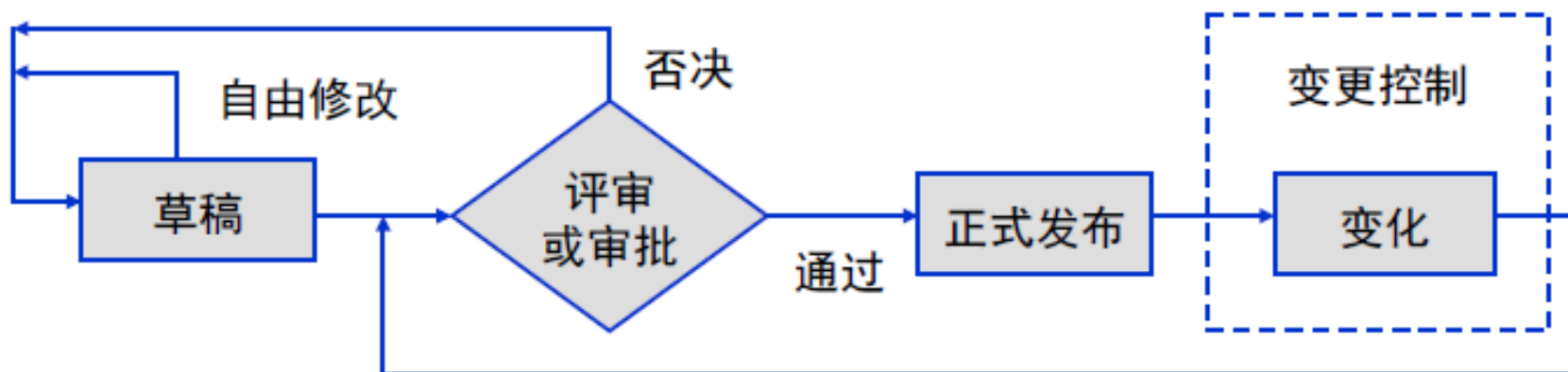
软件配置管理

□ 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

- **软件配置管理**是在软件的整个生命期内，管理变化的一组活动。在软件项目启动时就开始，并且一直持续到软件退役才终止的一组跟踪和控制活动。
- 在开发软件的过程中，**变化（变动）既是必要的，又是不可避免的**。变化也很容易失去控制，如果不能适当地控制和管理变化，势必造成混乱，并产生许多严重的错误。
- **变化的种类**
 - ①标识变化； ②控制变化； ③确保适当地实现了变化； ④向需要知道这类信息的人报告变化。
- 软件配置管理的目标是，使变化更正确且更容易被适应，在必须变化时，减少所需花费的工作量。

配置管理过程

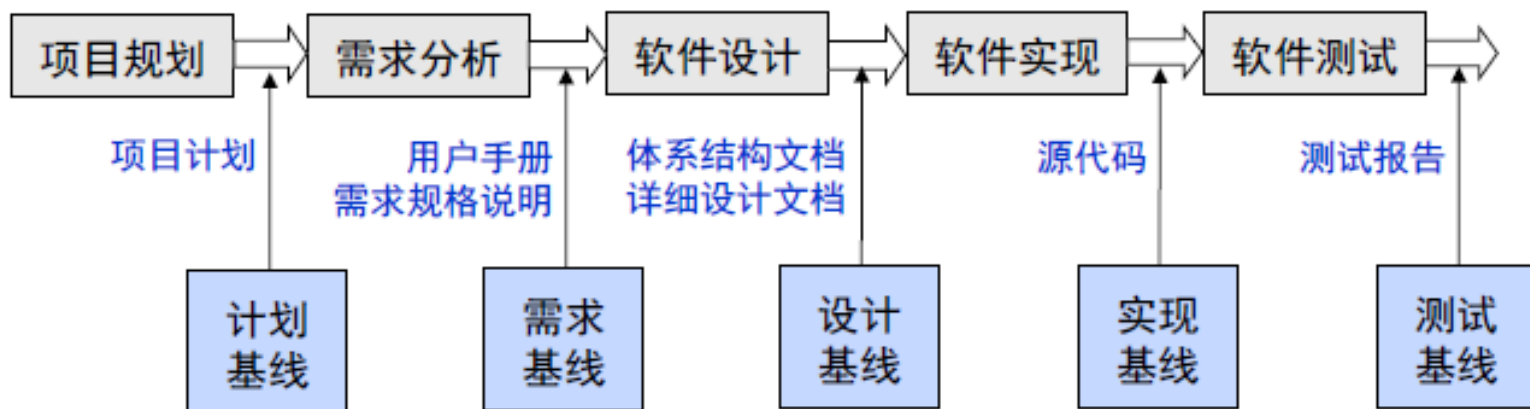
- 1、 标识配置项
- 2、 进行配置控制（**关键**）
- 3、 执行配置审计
- 4、 记录配置状态

配置控制是核心：

- ✓ 存取控制（开发库、**基线库**、产品库）
- ✓ 版本控制
- ✓ 变更控制
- ✓ 产品发布控制

基线 Baseline

- ❑ **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)**。



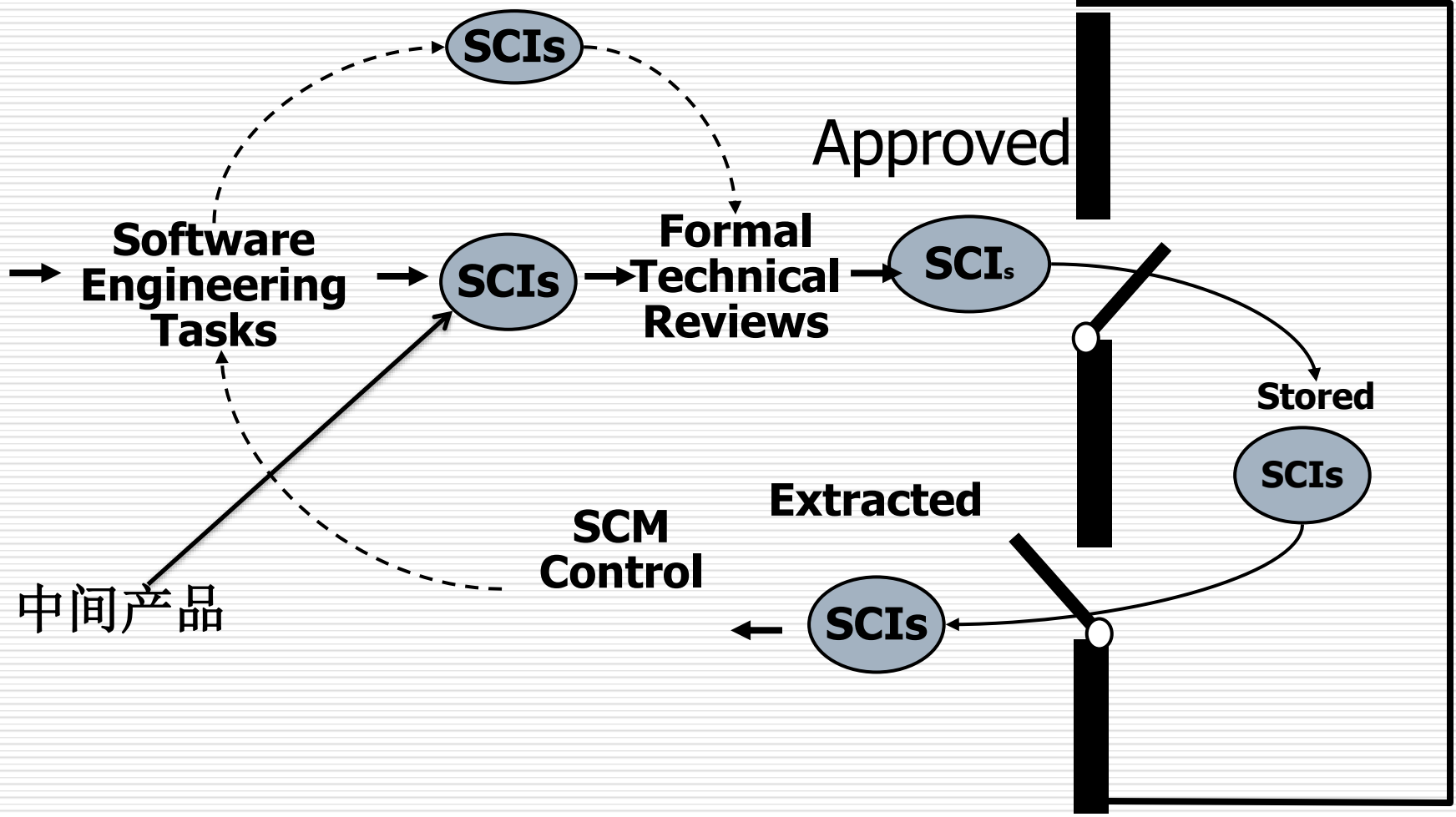
基线 Baseline

IEEE把基线定义为：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它。

- 基线由一组配置项组成，这些配置项构成了一个相对稳定的逻辑实体。基线中的配置项被“冻结”了，则不能再被任何人随意修改。
- 基线通常对应于开发过程中的里程碑（Milestone），一个产品可以有多个基线，也可以只有一个基线。基线的主要属性有：名称、标识符、版本、日期等。
- 通常将交付给客户的基线称为一个“Release”，为内部开发用的基线则称为一个“Build”。

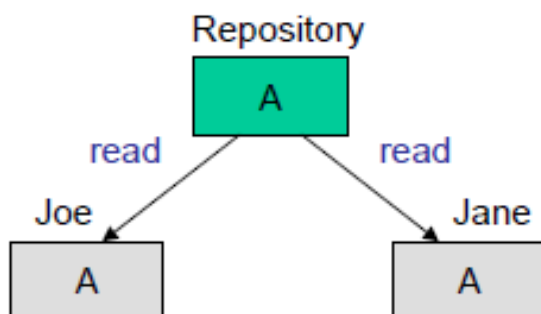
Baselines

Modified

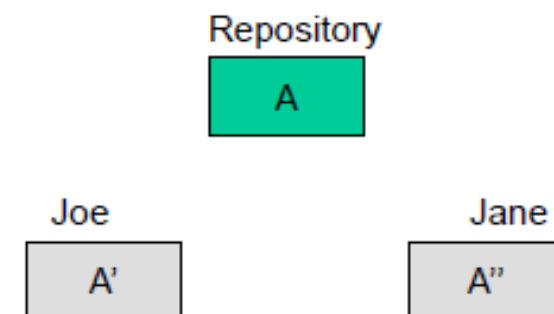


软件配置管理

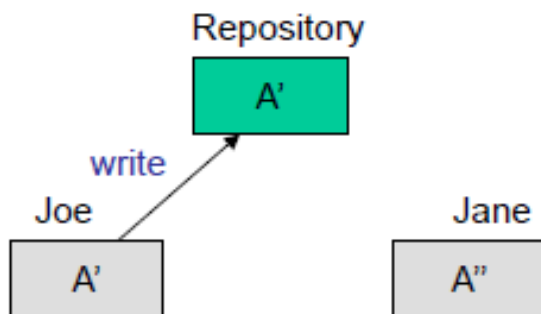
□ 文件共享的问题



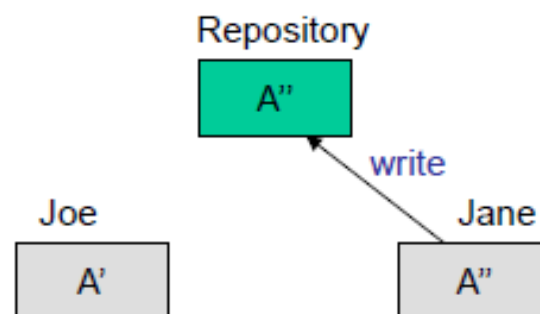
两个用户读取同一个文件



两个用户开始编辑文件副本



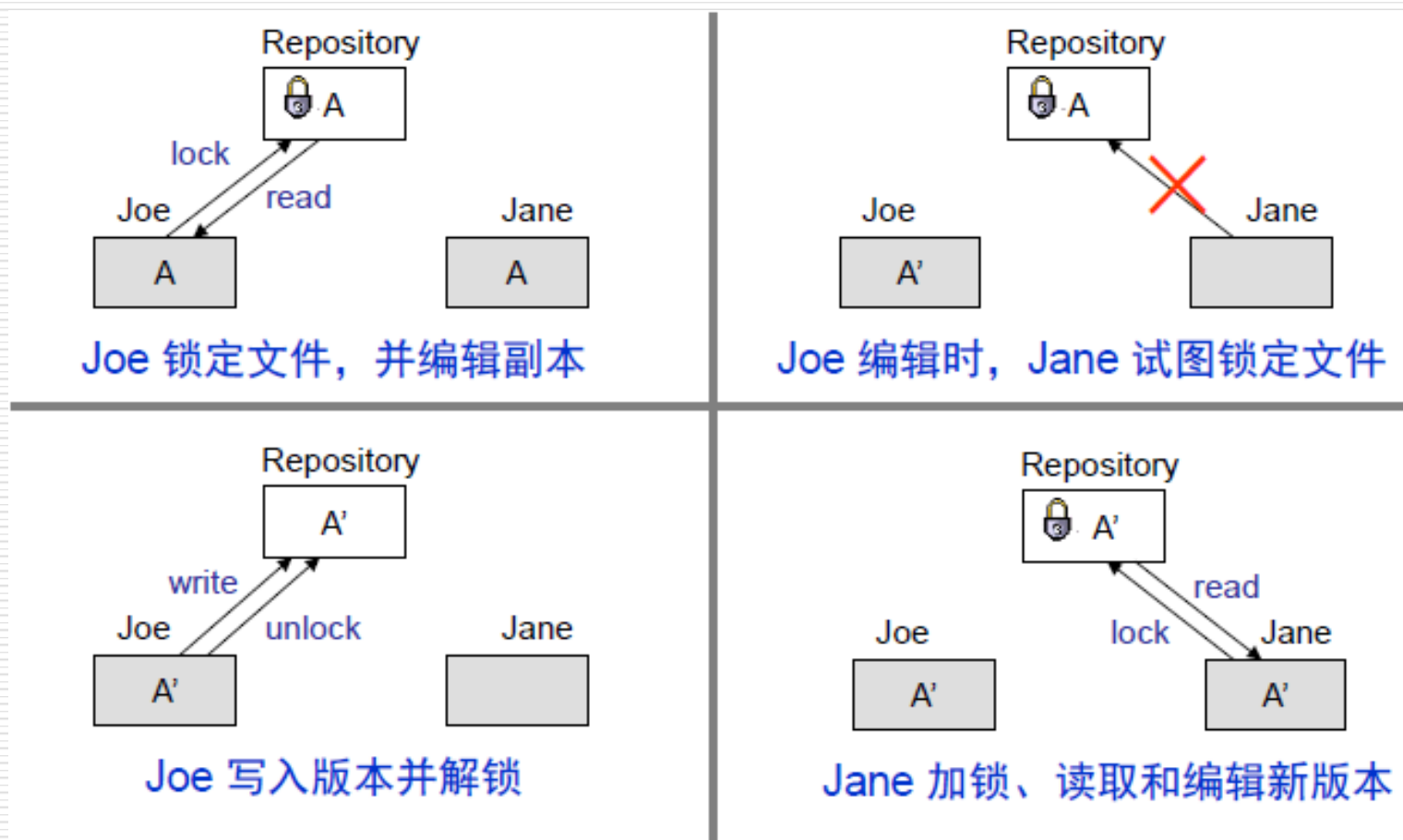
Joe 首先发布其版本



Jane 意外地覆盖了 Joe 的版本

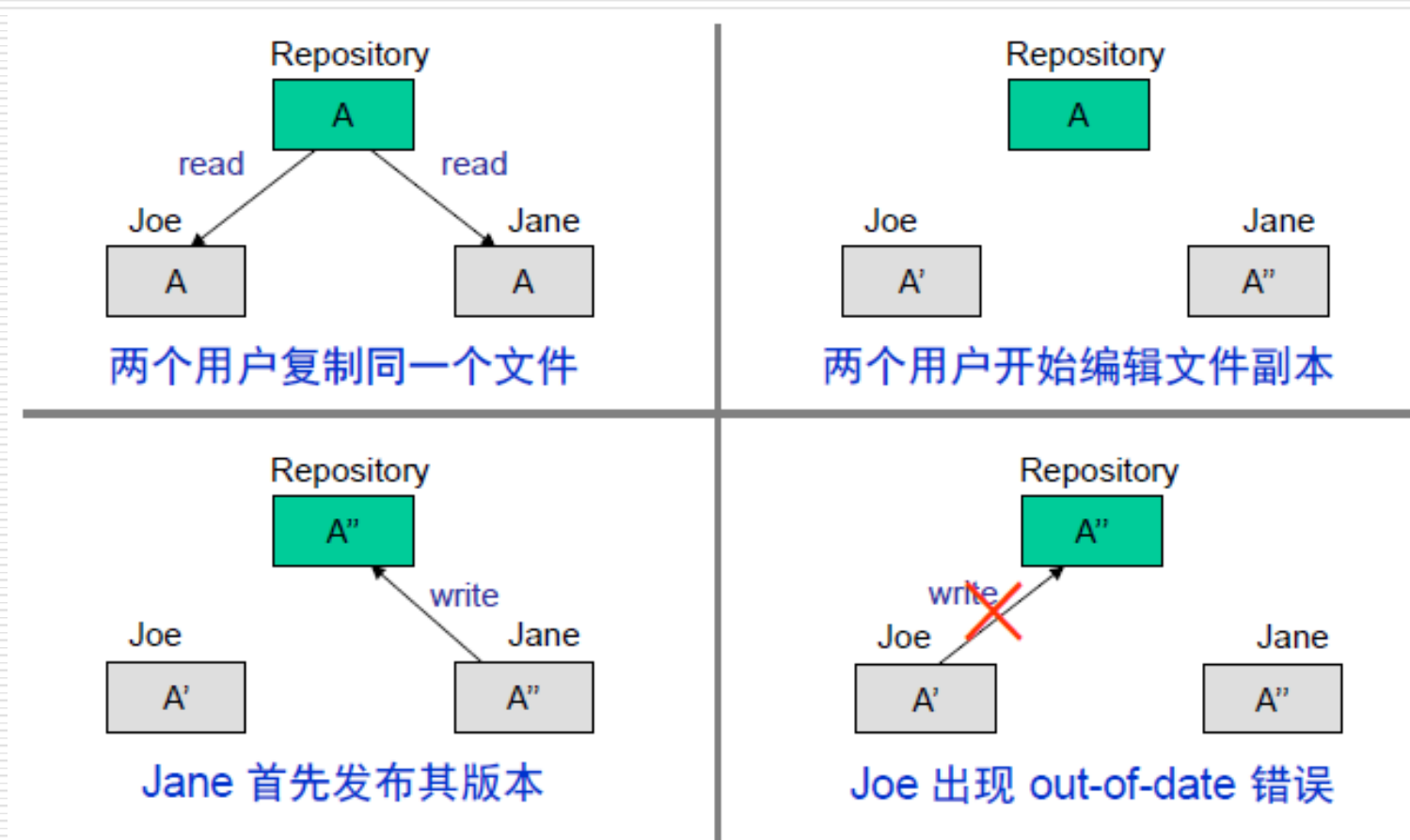
软件配置管理

□ 版本控制模型（独占工作模式）



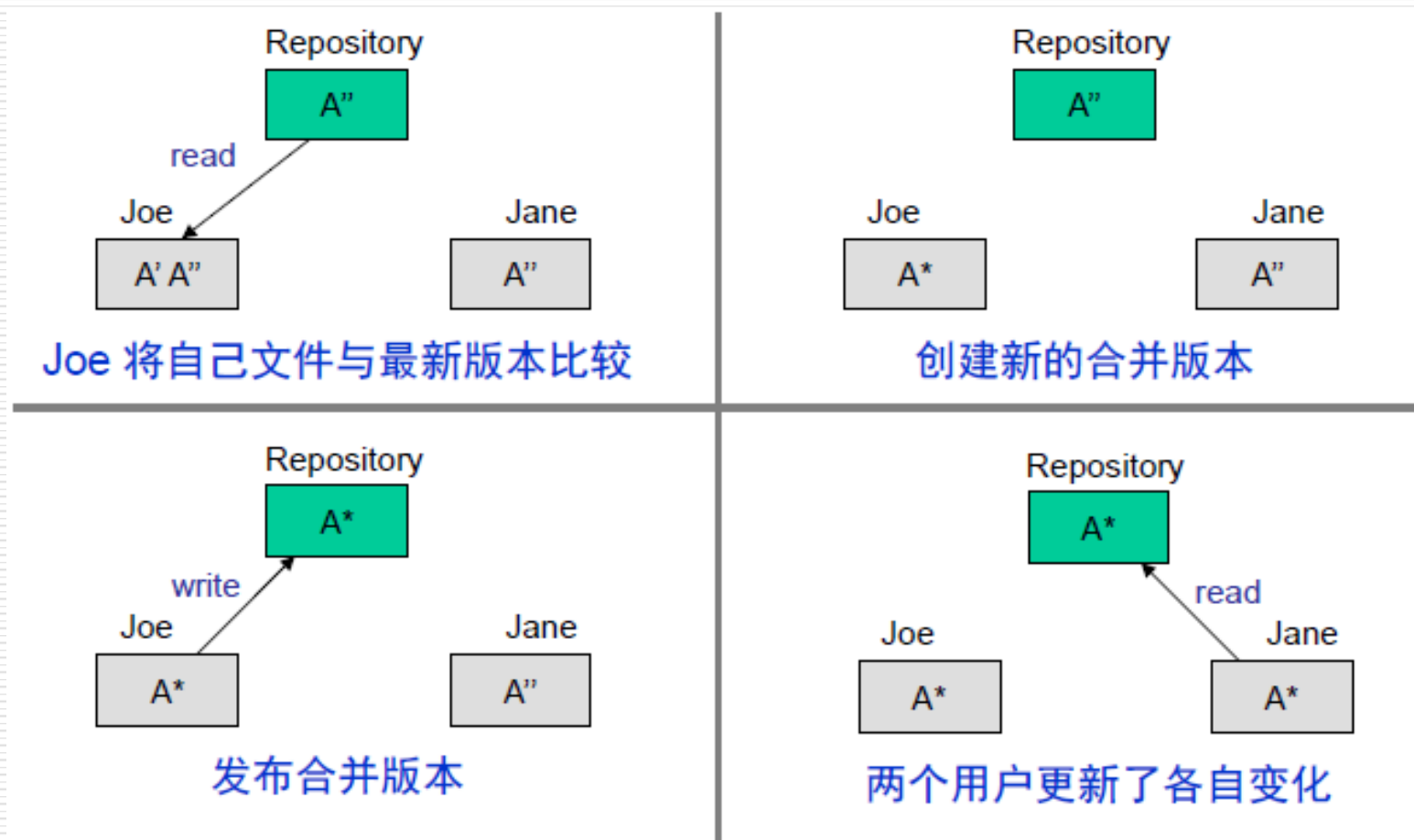
软件配置管理

□ 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）



配置管理工具

1、 SourceSafe

2、 CVS

3、 ClearCase

1、SourceSafe

- SourceSafe是Microsoft公司推出的配置管理工具，是Visual Studio的套件之一。SourceSafe是国内最流行的配置管理工具，用户量绝对是第一位。
- SourceSafe长得很象早先土气的文件管理器，的确难看。但是难看不碍事，SourceSafe的优点可以用8个字来概括“简单易用，一学就会”，这个优点是它老妈Microsoft遗传下来的，是天生的。
- 虽然SourceSafe并不是免费的，但是在国内人们以接近于零的成本得到它，网上到处可以下载啊。当然Microsoft也不在乎这个小不点的软件，它属于“买大件送小件”的角色。如果你合法地得到Visual Studio，你就得到了免费的SourceSafe。
- SourceSafe的主要局限性：
 - ✓ 只能在Windows下运行，不能在Unix, Linux下运行。SourceSafe不支持异构环境下的配置管理，对用户而言是个麻烦事。这不是技术问题，是微软公司产品战略决定的。
 - ✓ 适合于局域网内的用户群，不适合于通过Internet连接的用户群，因为SourceSafe是通过“共享目录”方式存储文件的。

2、CVS

- CVS 是 Concurrent Version System（并行版本系统）的缩写，它是著名的开放源代码的配置管理工具。
- CVS的官方网站是<http://www.cvshome.org/>。官方提供的是CVS服务器和命令行程序，但是官方并不提供交互式的客户端软件。许多软件机构根据CVS官方提供的编程接口开发了各色各样的CVS客户端软件，最有名的当推Windows环境的CVS客户端软件——WinCVS。WinCVS是免费的，但是并不开放源代码。
- 与SourceSafe相比，CVS的主要优点是：
 - ✓ SourceSafe有的功能CVS全都有，CVS支持并发的版本管理，SourceSafe没有并发功能。CVS服务器的功能和性能都比SourceSafe高出一筹。
 - ✓ CVS服务器是用Java编写的，可以在任何操作系统和网络环境下运行。CVS深受Unix和Linux的用户喜爱。Borland公司的JBuilder提供了CVS的插件，Java程序员可以在JBuilder集成环境中使用CVS进行版本控制。
 - ✓ CVS服务器有自己专用的数据库，文件存储并不采用SourceSafe的“共享目录”方式，所以不局限于局域网，信息安全性很好。
- CVS的主要缺点在于客户端软件，真可谓五花八门、良莠不齐。Unix和Linux的软件高手可以直接使用CVS命令行程序，而Windows用户通常使用WinCVS。安装和使用WinCVS显然比SourceSafe麻烦不少，这是令人比较遗憾的。

3、ClearCase

- IBM（Rational公司）的ClearCase是软件行业公认的功能最强大、价格最昂贵的配置管理软件。
- ClearCase主要应用于复杂产品的并行开发、发布和维护，其功能划分为四个范畴：版本控制、工作空间管理（Workspace Management）、构造管理（Build Management）、过程控制（Process Control）。ClearCase通过TCP/IP来连接客户端和服务端。另外，ClearCase拥有的浮动License可以跨越UNIX和Windows NT平台被共享。
- ClearCase的功能比CVS、SourceSafe强大得多，但是其用户量却远不如CVS、SourceSafe的多。主要原因是：
 - ✓ ClearCase价格昂贵，如果没有批量折扣的话，每个License大约5000美元。对于中国用户而言，这无疑是天价。
 - ✓ 用户只有经过几天的培训后（费用同样很昂贵），才能正常使用ClearCase。如果不参加培训的话，用户基本上不可能无师自通。

