软件项目的参与人员

- **高级管理者：**负责项目宏观管理的管理人员，战略规划、项目高层协调等
- **项目(技术)管理者：**计划、激励、组织和控制软件开发的人员

产品经理
项目经理

- **开发人员：**拥有开发软件所需技能的人员
  - 系统分析员、系统架构师、设计师、程序员、测试人员、质量保证人员、数据库管理员（DBA）、支持工程师、配置管理人员…
- **客户：**进行投资、详细描述待开发软件需求、关心项目成败的组织/人员
- **业务专家：**对某行业业务非常了解，既可以是客户方的成员，也可以是开发方的专/兼职人员
- **最终用户：**一旦软件发布成为产品，最终用户就是直接使用软件的人

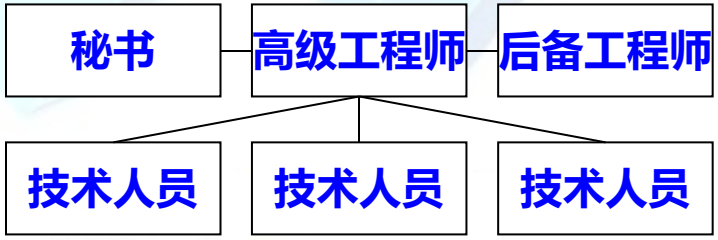
软件开发团队

- “最好的”团队取决于项目经理的管理风格、团队里的人员数目与技能水平、项目的总体难易程度
- 组建团队时应考虑以下要素：
  - 从项目需求来看：
    - 待解决问题的难度
    - 待开发软件系统的规模
    - 待开发软件系统的技能要求
    - 交付日期的严格程度
    - 共同工作的时间
    - 彼此之间的人际关系与友好交际程度
    - ……
  - 从个人能力来看：
    - 应用领域经验
    - 开发平台经验
    - 编程经验
    - 教育背景
    - 沟通能力
    - 适应能力
    - 工作态度
    - 团队协作能力
    - ……

软件开发团队的组织方式(1)

- 一窝蜂模式（chaos team)：没有明确分工，存活的时间一般都不长
- 主治医师模式: (Chief-Programmer Team, surgical team)
  - 手术台上，有一个主刀医师，其他人 (麻醉、护士、器械) 各司其职，为主刀医师服务
  - 首席程序员（Chief-programmer) 处理主要模块的设计和编码，其他成员从各种角度支持他的工作（backup programmer, admin, tool-smith, specialist)
  - 主治医师模式的退化: 学校里，软件工程的团队模式往往退化为“一个学生干活，其余学生跟着打酱油”

➔明星模式 (Super-star model)



软件开发团队的组织方式(2)

- 社区模式 (Community Model):
  - 由很多志愿者参与，每个人参与自己感兴趣的项目，贡献力量
  - 好处是“众人拾柴火焰高”， 但是如果大家都只来烤火，不去拾柴，或者捡到的柴火质量太差，最后火也熄灭了
  - “社区” 并不意味着“随意”，一些成功的社区项目(例如开发和维护Linux 操作系统的社区)都有很严格的代码复审和签入的质量控制

➔开源项目(Open Source Project)

软件开发团队的组织方式(3)

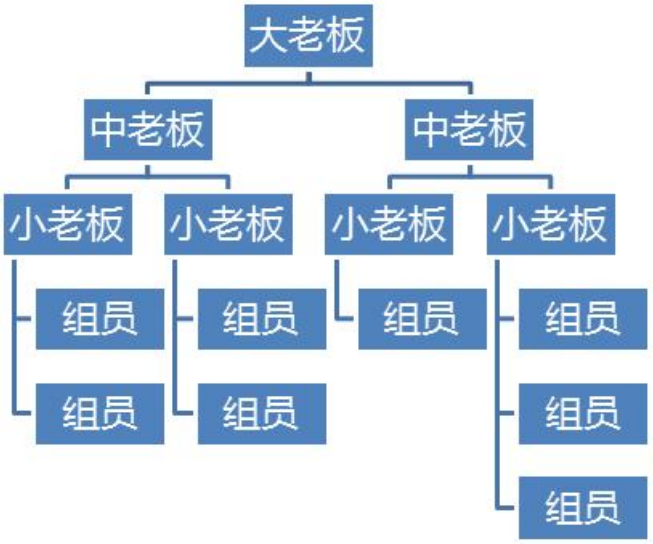
- 交响乐团模式 (Orchestra)
  - 人多，门类齐全，各司其职，各自有专门场地，演奏期间严格遵循纪律
  - 演奏靠指挥协调，各自遵循曲谱(工作流程)
  - 演奏的都是练习过多次的曲目，重在执行

  ➔类似于“工厂”，严格遵循预定的生产流程，“规格严格”
- 爵士乐模式 (Jazz Band)
  - 演奏时没有谱子，没有现场指挥，平时有arranger起到协调和指导作用
  - 模式：主乐手先吹出主题，其余人员根据这个主题各自即兴发挥；主乐手最后再加入，回应主题，像是对曲子的总结
  - “强调个性化的表达，强有力的互动，对变化的内容有创意的回应”

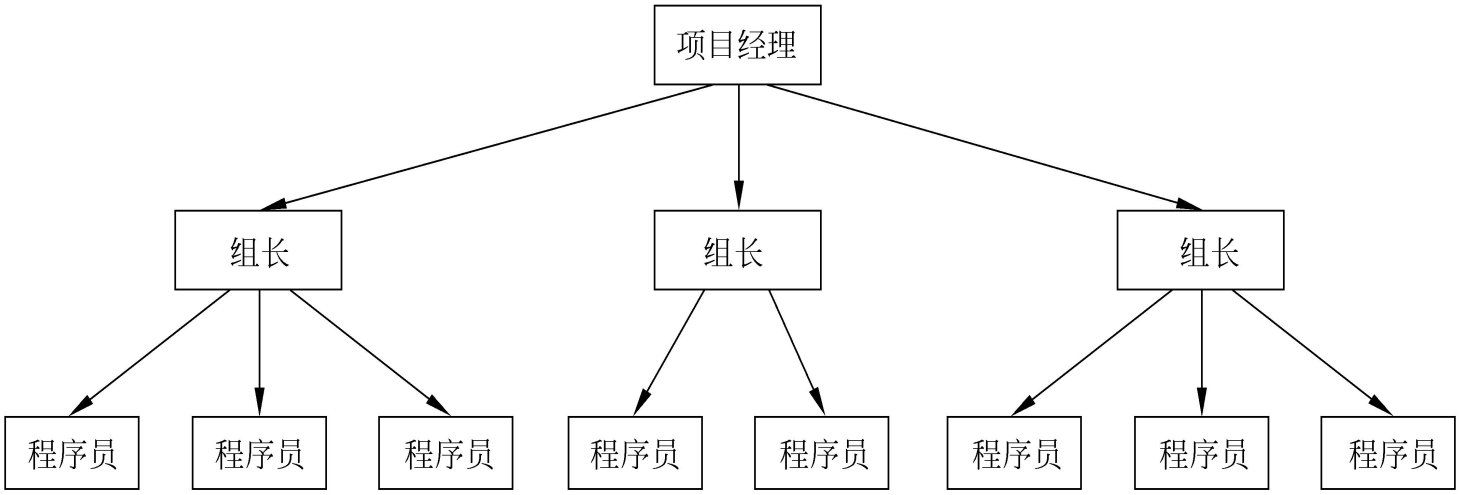
  ➔类似于一群天才构成的敏捷团队，“功夫到家”，率性而为

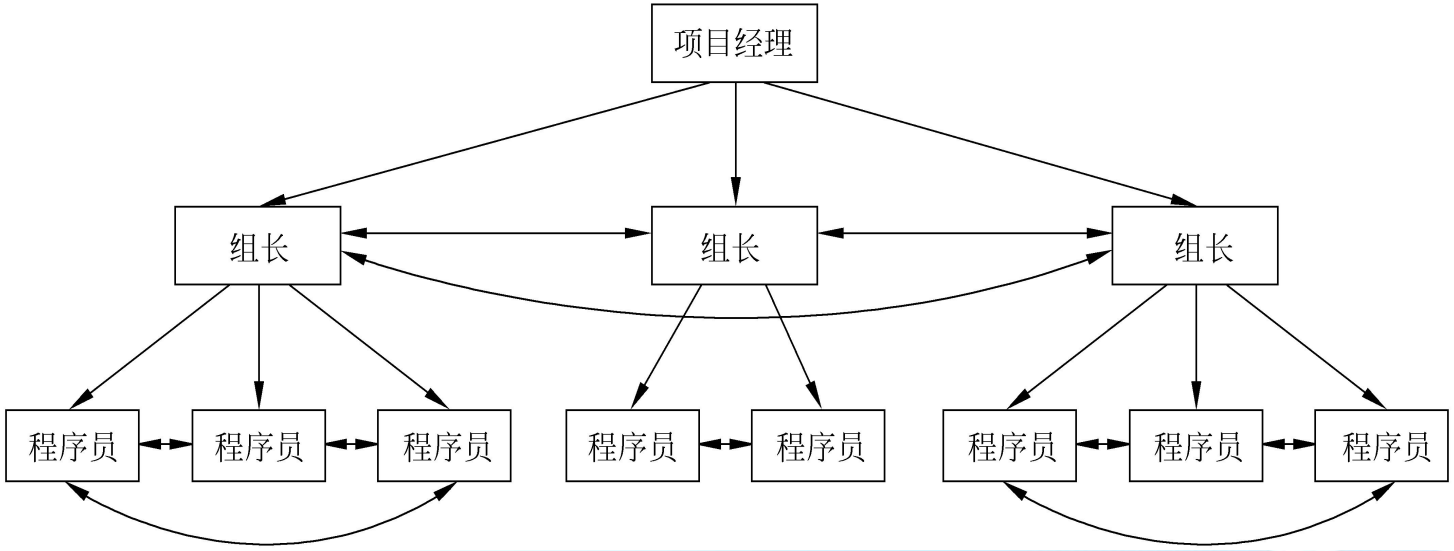
软件开发团队的组织方式(4)

- 功能团队模式 (feature team)
  - 具备不同能力的同事平等协作，共同完成一个项目开发
  - 在这个项目完成之后，这些人又重新组织，和别的角色一起去完成下一个功能， 他们之间没有管理和被管理的关系
- 官僚模式 (bureaucratic model)
  - 成员之间不仅有技术方面的合作和领导，同时还混进了组织上的领导和被领导关系，跨组织的合作变得比较困难



大型项目的技术管理组织结构

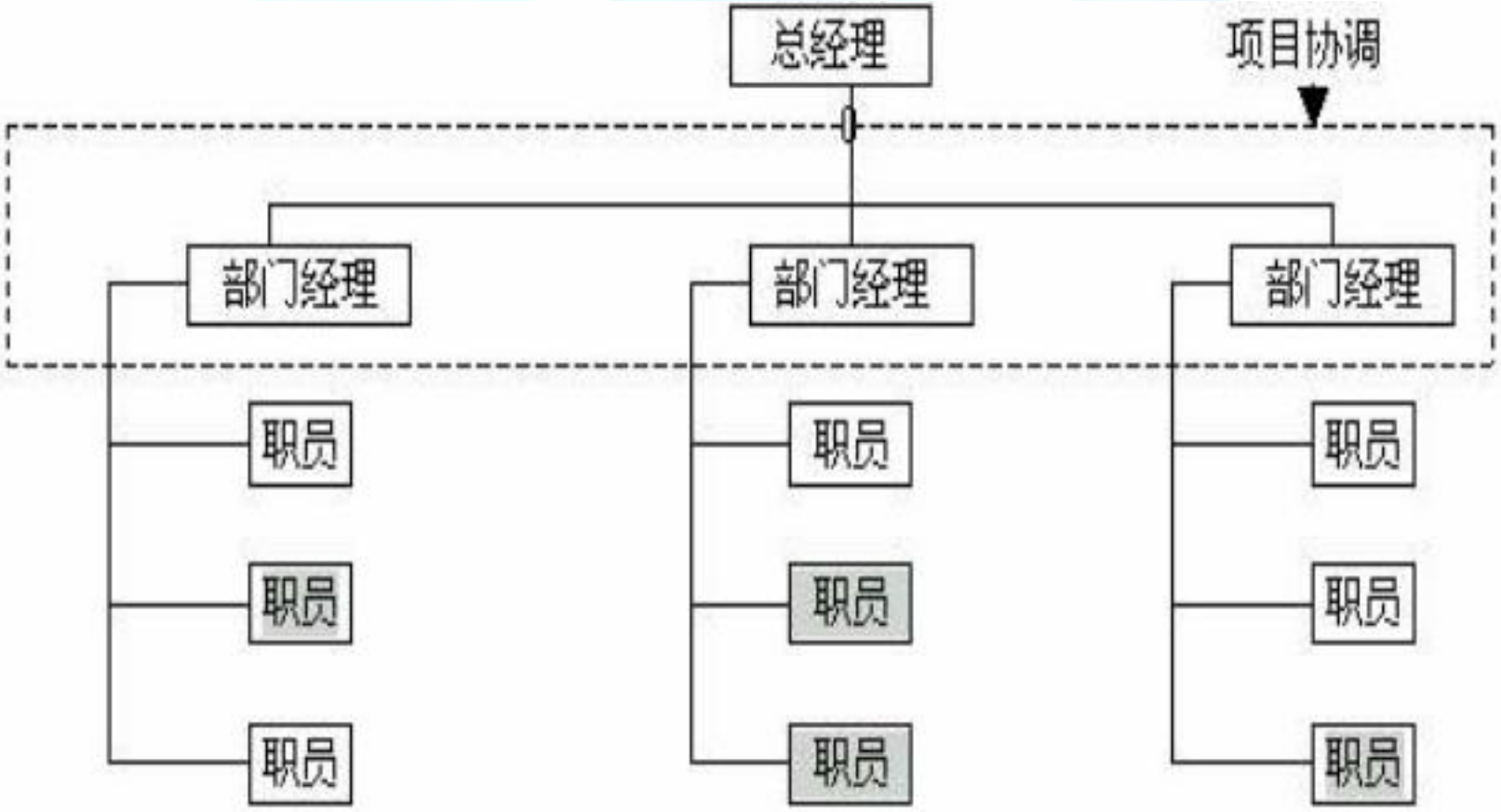




组织结构的主要类型

- 职能型
- 项目型
- 矩阵型

组织结构的主要类型 – 职能型



组织结构的主要类型 – 职能型

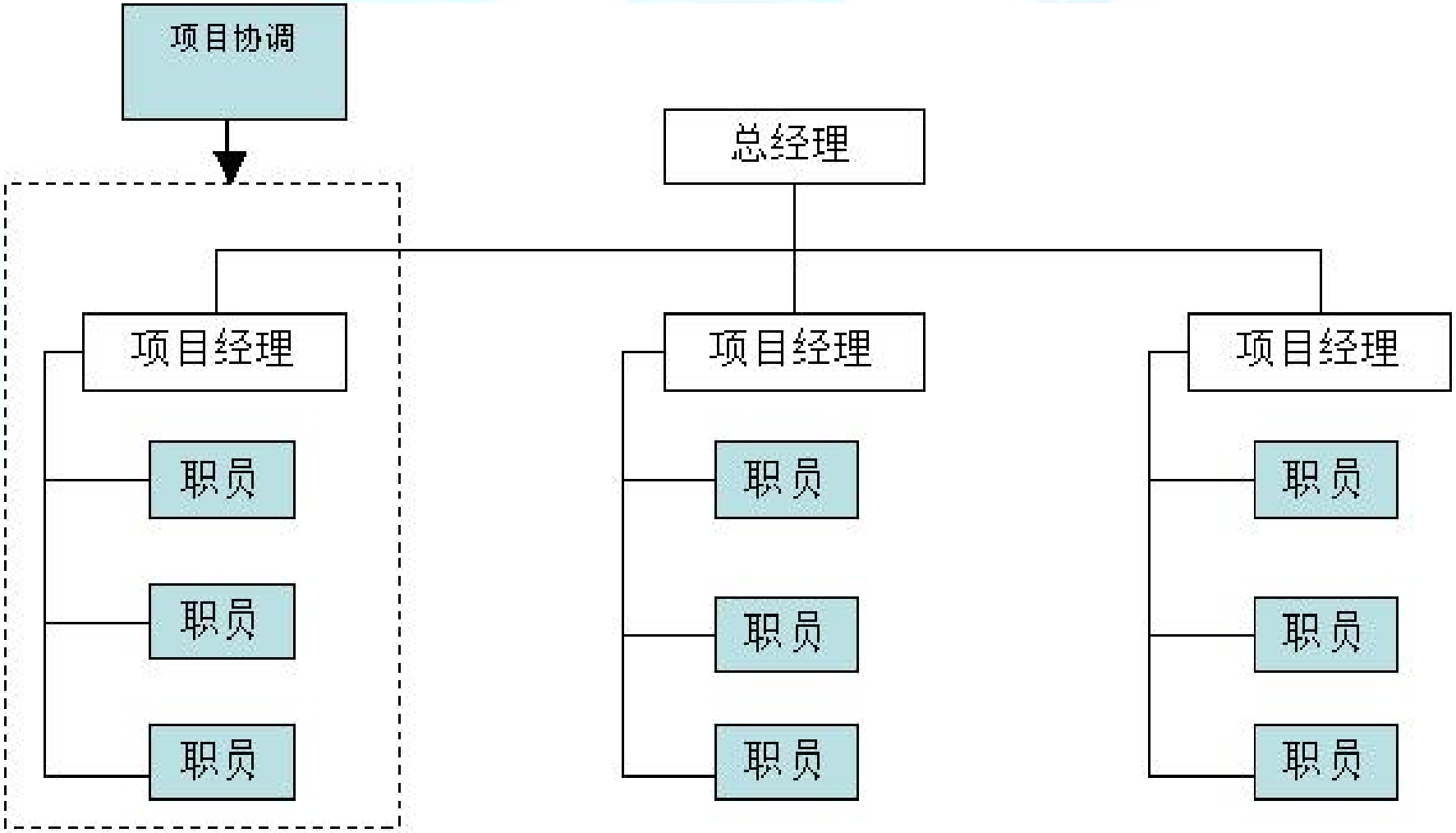
优点：

- 以职能部门为主体承担项目，可充分发挥职能部门的人力优势
- 职能部门内部技术专家可以被多个项目共享，节约人力资源
- 同一职能部门内部专业人员便于交流、支援
- 项目成员有调离时，容易在部门内部增员，保持项目技术连续性
- 项目成员将项目工作与本职能部门工作融合，减少因项目临时性带来的不确定性

缺点：

- 客户利益与职能部门利益发生冲突，项目及客户利益往往不优先考虑
- 当项目需要多个职能部门共同完成，或者部门内部承担多个项目时，资源的平衡就会出现问题
- 当项目需要多个职能部门共同完成，由于权力分割不利于各部门沟通，项目经理没有足够权力控制项目进展
- 项目成员在行政上隶属于各职能部门的领导，项目经理对项目成员没有足够的控制权力，沟通成本高

组织结构的主要类型 – 项目型



组织结构的主要类型 – 项目型

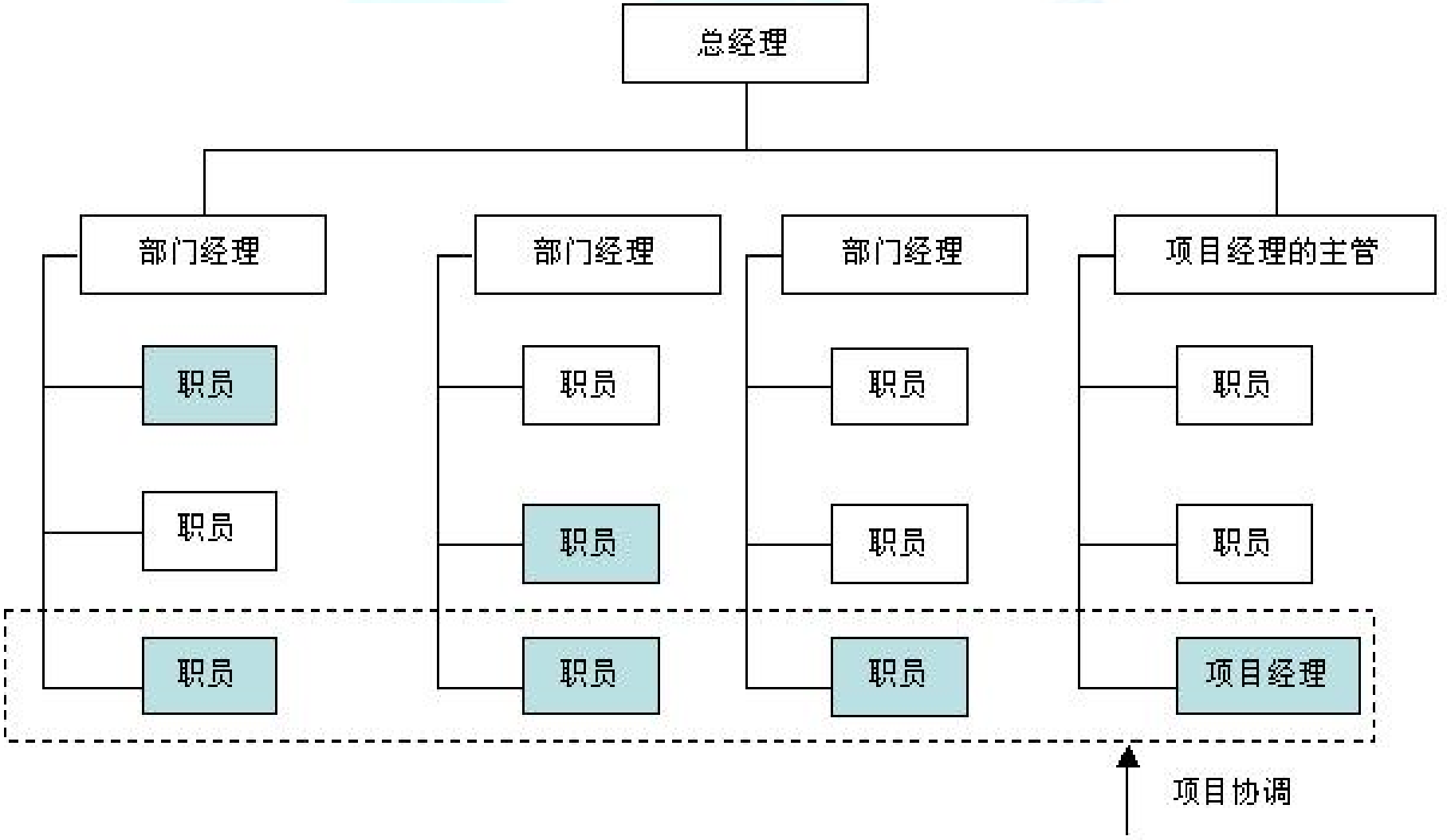
优点：

- 项目经理全权对项目负责，有权根据项目需要调配组织内部资源
- 项目型组织的目标单一，完全以项目为工作中心，有利于项目完成
- 项目经理对项目成员有全部权力，项目成员只需对项目负责，避免多重领导、无所是从的局面
- 组织结构简单，易于操作，沟通简洁、快速，提高工作效率

缺点：

- 不同项目团队的资源不能共享
- 各个项目组之间无沟通机制，影响公司长远发展
- 项目开发完成后，项目团队即解散，对于成员来说，缺乏事业上的连续性和安全感
- 项目团队之间缺乏信息交流，跨组共享经验和技术较难

组织结构的主要类型 – 矩阵型



组织结构的主要类型 – 矩阵型

优点：

- 专职的项目经理负责整个项目，以项目为中心，能迅速解决问题，在最短的时间内调配人员组成团队，将不同职能的人集中在一起
- 多个项目可以共享各个职能部门的资源
- 既有利于项目目标的实现，也有利于公司长远目标方针的贯彻
- 项目结束后可以回到原来部门，项目成员顾虑减少了

缺点：

- 容易引起职能部门经理与项目经理权力的冲突
- 资源共享可能会引起项目组之间的冲突
- 项目成员有项目经理和原职能部门领导等多重领导，会有一定的焦虑和压力

项目人员职责计划

项目开发团队确定后，要制定人员职责计划：

- 责任分配矩阵(RAM，Responsibility Assignment Matrix)
- 组织分解结构(OBS，Organization Breakdown Structure)
- 文本描述(表格等)

项目人员职责计划 – 责任分配矩阵(RAM)

- 用来对项目团队成员进行分工的工具，明确其角色和职责
- 责任分配矩阵能反映出每个团队成员和开发活动的关系，从而避免责权不清。

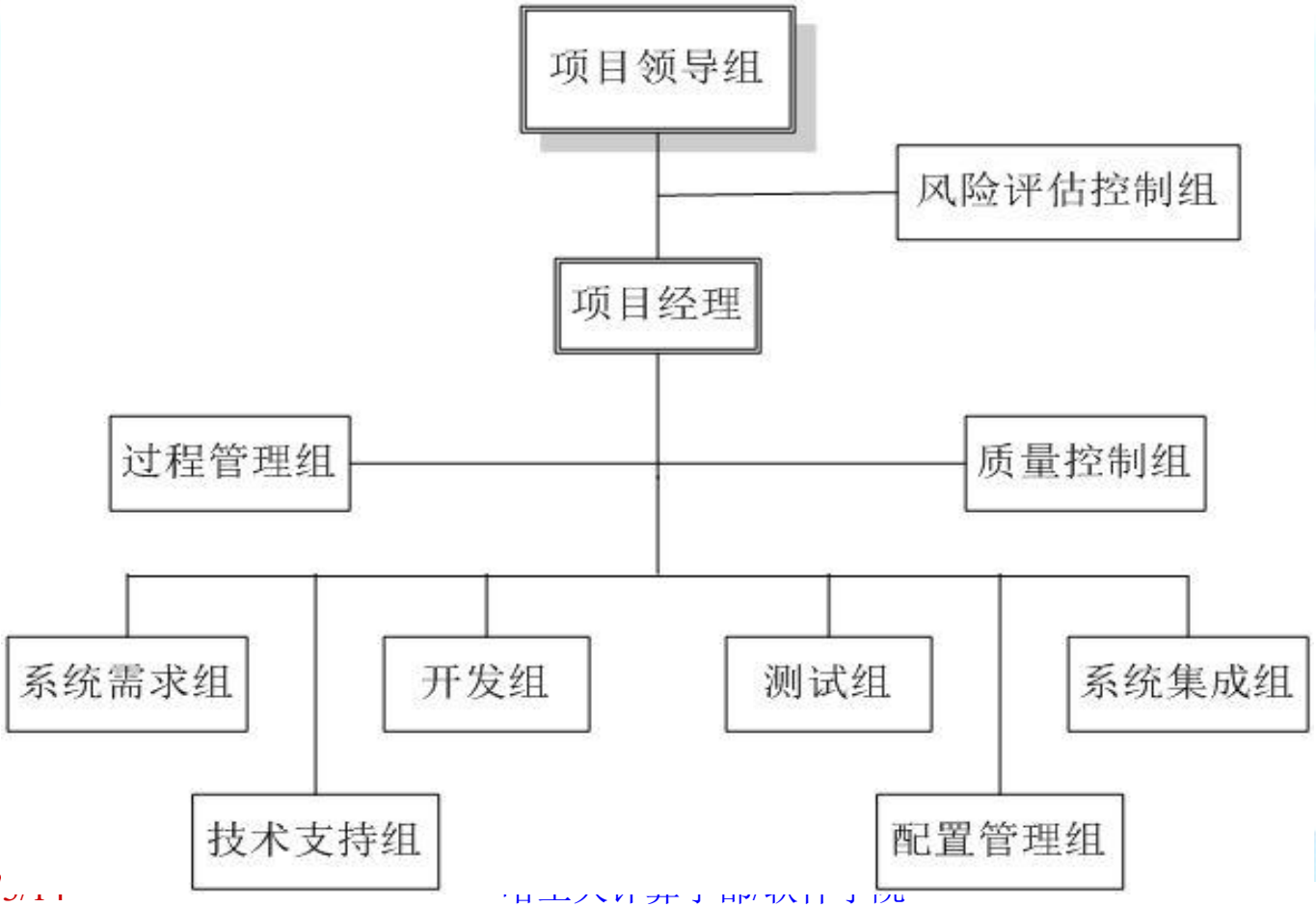
例如：

项目人员 / 活动	项目经理	应用开发人员	网络工程师	专家
建立应用软件	A	C	P	
测试应用软件	A	P	P	
应用软件打包	R		R	P
测试发布应用软件	R	R		C
在工作站上安装应用	A		P	C

注：A = Approver（批准），R = Reviews（评审），P = Participant（参加），C = Creator（建立）。

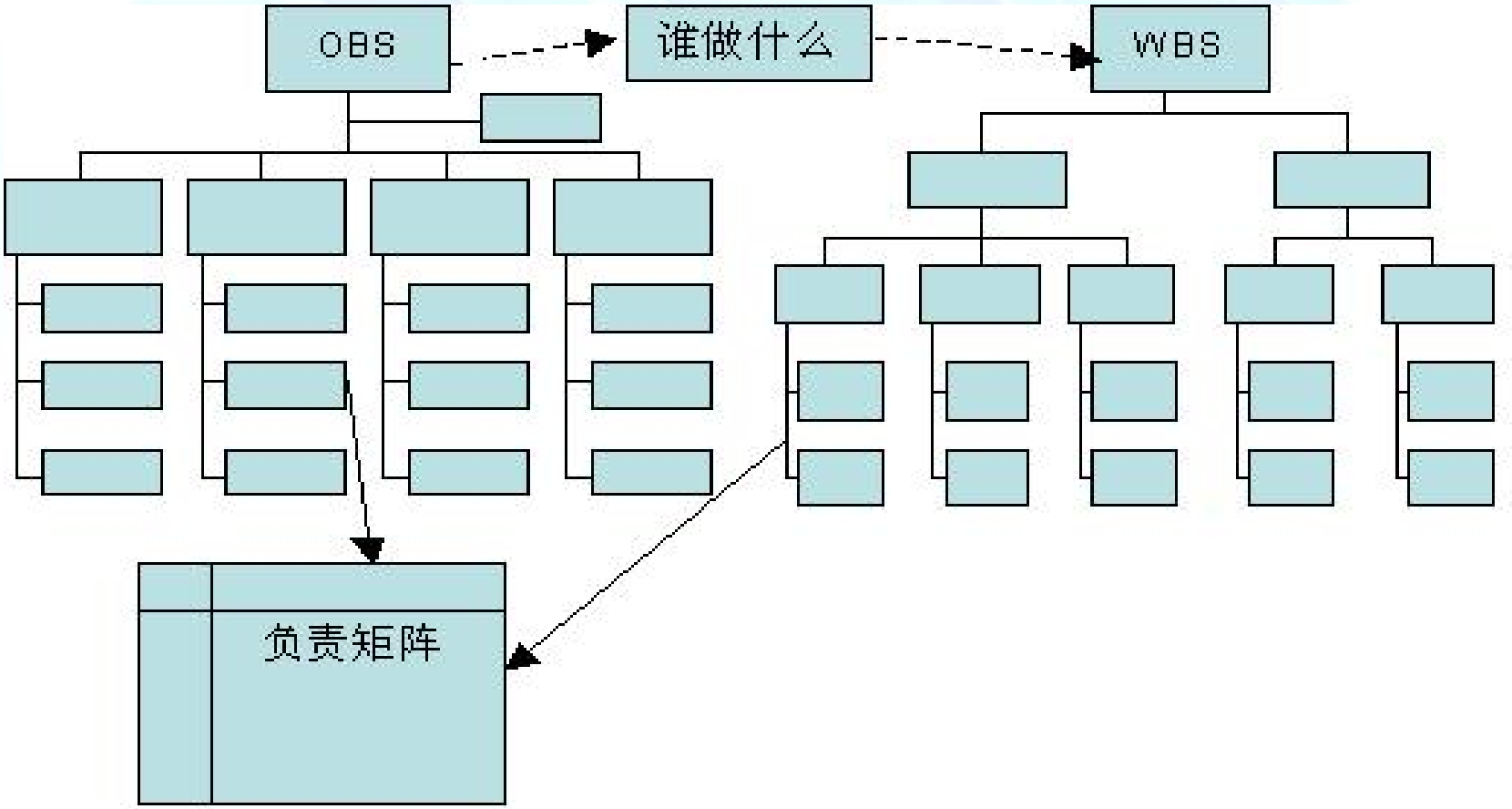
项目人员职责计划 – 组织结构图(OBS)

- 用组织结构图来展示项目团队成员及其隶属组织关系、请示报告关系
- 组织结构图和前面讲到的WBS之间可以建立关联关系，构成责任矩阵



项目人员职责计划 – 组织结构图(OBS)

- 用组织结构图来展示项目团队成员及其隶属组织关系、请示报告关系
- 组织结构图和前面讲到的WBS之间可以建立关联关系，构成责任矩阵



项目人员职责计划 – 组织结构图

- 用组织结构图来展示项目团队成员及其隶属组织关系、请示报告关系
- 组织结构图和前面讲到的WBS之间可以建立关联关系，构成责任矩阵
- 例如：某项目的OBS与WBS之间的责任矩阵关系

单位	WBS 任务							
	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.1.6	1.1.7	1.1.8
系统部门	R	RP					R	
软件部门			RP					
硬件部门				RP				
测试部门	P							
质量保证部门					RP			
配置管理部门						RP		
后勤部门							P	
培训部门								RP

注：R表示负责者（部门），P表示执行者（部门）。

项目人员职责计划 – 文本描述

- 使用表格等文本性的文档来详细定义项目小组及项目成员的职责
- 例如：某项目的职责说明表

序号	名称	人员数量	组长	职能
1	总体组	7	曲剑	• 负责项目计划和进度控制 • 协调和安排项目任务 • 协调和管理各项目小组工作
2	平台组	6	刘建强	设计和完善三务合一信息化平台
3	设计组	4	刘建强	• 系统分析 • 概要设计 • 详细设计
4	需求组	6	李娥	• 需求调研、需求分析 • 编写需求规格说明书 • 需求确认签字 • 客户培训
5	开发组	16	赵伟宏	• 程序开发 • 单元测试
6	测试组	5	周林红	• 编写测试计划、测试用例 • 进行功能测试、集成测试、压力测试和回归测试 • 提交测试报告
7	实验组	3	陈明	• 项目的具体实施工作 • 系统的安装、调试、部署工作

软件项目管理-团队计划

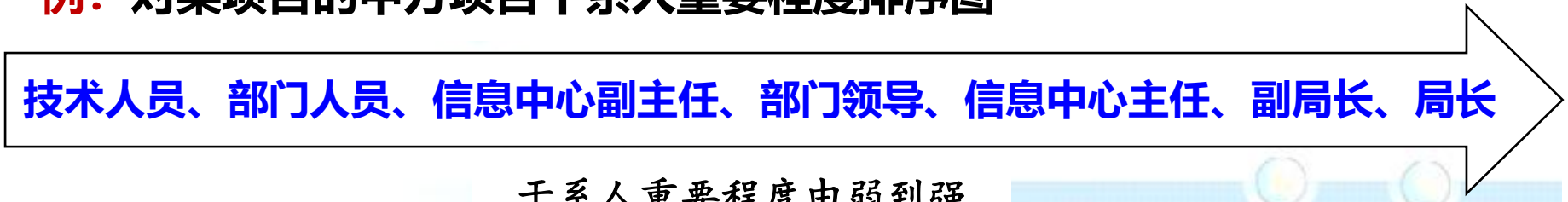
识别项目干系人

识别出干系人，分析和记录他们的相关信息，如联络信息、他们的利益、参与度、影响力及对项目成功的潜在影响。下面所列人员都可以是主要项目干系人：

- **项目经理**：负责对项目进行管理的人员
- **客户**：使用项目产品的组织或者个人，指项目产品的购买者
- **用户**：产品的直接使用者
- **项目执行组织**：其员工主要投入项目工作的组织
- **项目团队成员**：具体从事项目工作，并直接或者间接向项目经理负责的人员
- **项目出资人**：为项目提供资助的个人或者团体
- **项目承包人**：依据合同而投入项目实施工作的一方，不具有对项目产品的所有权
- **供货商**：一个项目常常离不开供货商，它提供项目组织外的某些产品，包括服务

按照重要性对干系人进行分析

例：对某项目的甲方项目干系人重要程度排序图



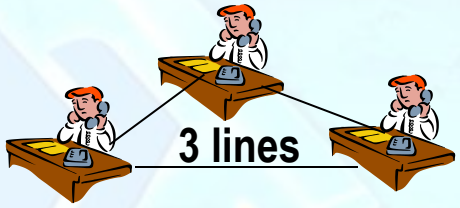
干系人重要程度由弱到强

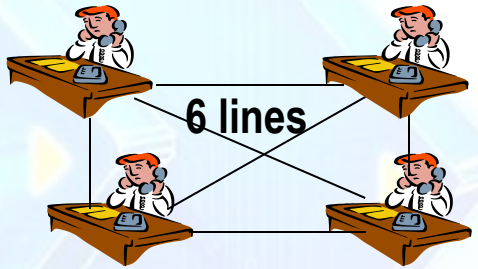
项目沟通计划 – 沟通内容及方式

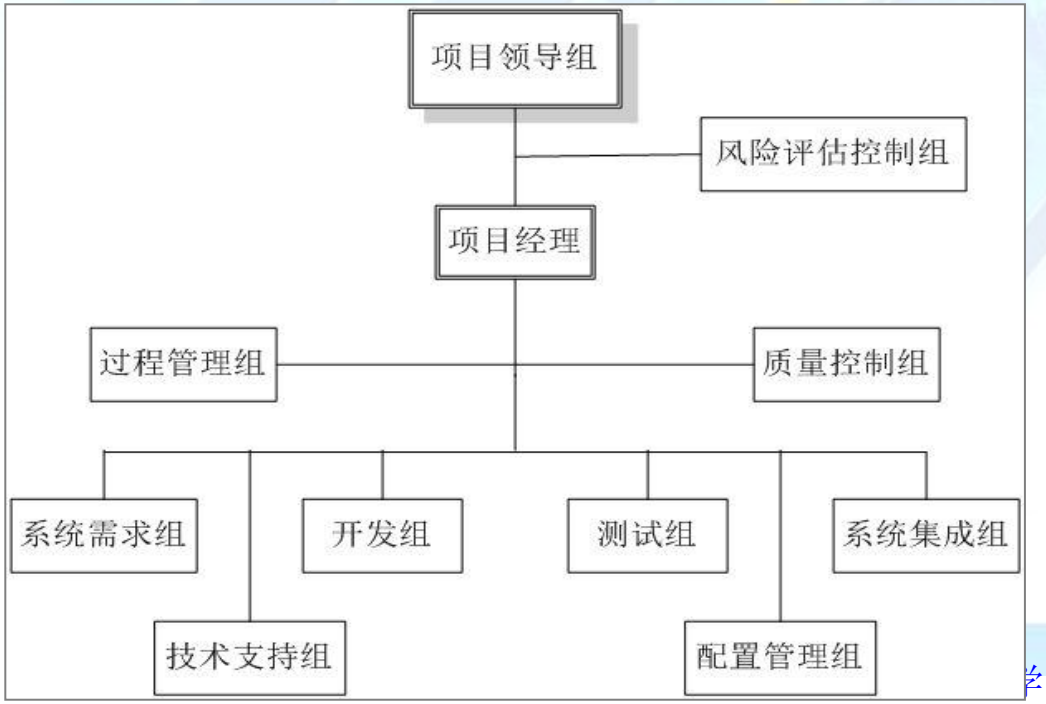
- 沟通计划是确定谁需要信息，需要什么信息，何时需要信息，以及如何将信息分发给他们
- 项目沟通方式：
  - 内部沟通与外部沟通
  - 正式沟通与非正式沟通
  - 官方沟通与非官方沟通
- 沟通内容：
  - 向上沟通：针对高层次相关方
  - 向下沟通：针对承担项目工作的团队和其他人员
  - 横向沟通：针对项目经理或团队的同级别人员

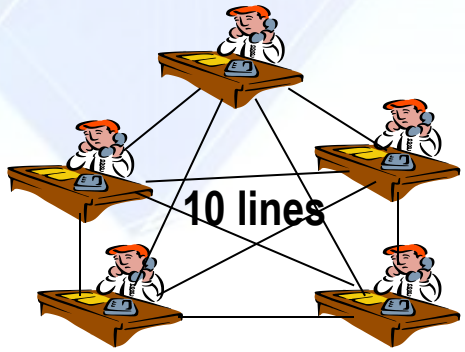
项目沟通计划 – 沟通渠道

- 沟通渠道像连接每个人的电话线一样，人越多，沟通渠道越多
- 实践中，可以从项目组织图方式展示项目团队成员及报告、沟通关系









N persons need n(n-1)/2 lines

项目沟通计划 – 编制沟通计划表

沟通管理计划

一、基本情况

项目名称			沟通计划制定日期	年 月 日	沟通计划审批人（可多部门）		沟通计划审批日期	年 月 日

二、沟通管理计划

角色/人员名称	沟通级别	沟通需求			发布信息归档		信息发布				特殊邮件（非新华邮箱
		所需信息（描述需要哪些信息）	时间要求（频度）	沟通方式	归档格式（具体到文件名）	归档人	发布方式	发布时间	发布人	发布人类型	
CTO	指导委员会级别	项目任务的进展情况	每月	项目月汇报会议	项目月汇报	甲	电子邮件	项目月汇报会议后2个工作日内	乙		
张三、李四	项目组级别	周例会情况	每周	电子邮件	项目周例会会议纪要	甲	电子邮件	项目周例会后2个工作日内	乙		

角色/人员名称	可以是角色，例如：指导委员会成员，可以是具体人员名称，还可以是某个组
沟通级别	暂定内容，可补充，包括：指导委员会级别、PMO级别、项目组级别
沟通方式	包括但不限于：电子邮件、XXX会议、电话、短信……，其中会议需要明确具体名称
归档格式	一份或多份具体的文档，需具体到文件名，便于归档人操作
归档人	归档人一般为一个人，若存在多人的情况，需要描述各自负责的文档章节内容
发布方式	默认适用于一种信息发布方式

敏捷团队

敏捷的角色	Scrum 角色
□ 产品负责人(Product owner) □ 团队促进者(Team facilitator) □ 跨职能团队成员(Cross-functional team member)	□ Product Owner 产品负责人 □ Scrum Master Scrum主管 □ 开发团队

敏捷团队

- 最有效的敏捷团队往往由三到九个成员组成(黄金人数5-9人)
- 理想情况下，敏捷团队应该集中在一个工作场所工作
- 团队成员100%为专职成员，协同工作，自组织团队
- 敏捷鼓励自我管理团队

敏捷团队

仆人式领导

- 仆人式领导是通过对团队服务来领导团队的
- 注重理解和关注团队成员的需要和发展
- 仆人式领导为团队赋权
- 旨在使团队尽可能达到最高绩效

敏捷团队

敏捷方法提倡高度透明

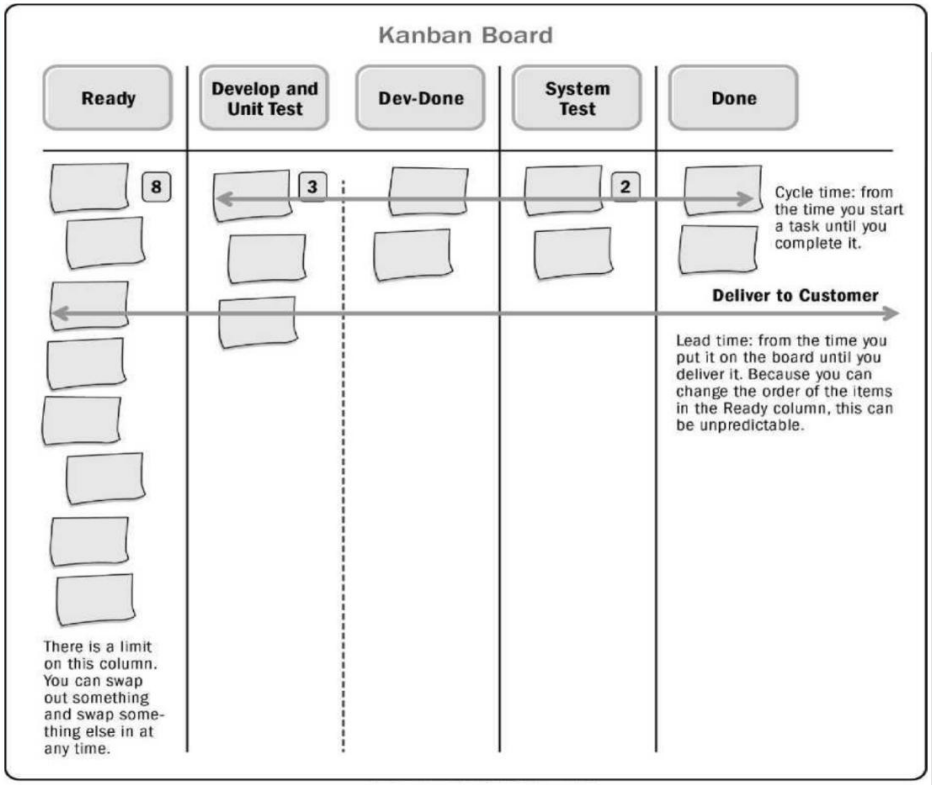
- 邀请所有相关方参与项目会议和审查

软件项目管理-团队计划

敏捷团队

敏捷方法提倡高度透明

- 将项目信息发布到公共空间



敏捷团队

大规模敏捷框架：Scrum of Scrums

Jeff Sutherland 和 Ken Schwaber 首次提出并实施