



项目管理

项目进度管理

本单元要点

- 一、进度管理的基本概念及过程
- 二、进度估算的基本方法
- 三、任务资源估计
- 四、编制进度计划
- 五、案例分析



项目计划内容与前期准备

- **项目计划**

- 进度管理
 - 任务与资源分配
- 质量管理
- 费用管理
-

- **制定项目计划的内容**

- 创建工作分解结构
- 制定进度计划
- 编制预算
-

- **项目执行控制**

- 每周报告、检查、例会、
-



工作基础：工作分解结构的主要目的和用途

- 1) 明确和准确说明项目的范围；
- (2) 为各独立单元分派人员，规定这些人员的相应职责；
- (3) 针对各独立单元，进行时间、费用和资源需要量的估算，
- (4) 确定项目进度测量和控制的基准；
- (5) 将项目工作与项目的财务帐目联系起来；
- (6) 自上而下将项目目标落实到具体的工作上
- (7) 确定工作内容和工作顺序；
- (8) 估计项目整体和全过程的费用。



一、进度管理过程

- **进度是对执行的活动和里程碑制定的工作计划日期表。**
- **进度管理是为了确保项目按期完成所需要的过程。**
- **进度管理重要性：**
 - 按时完成项目是项目经理最大的挑战之一
 - 时间是项目规划中灵活性最小的因素
 - 进度问题是项目冲突的主要原因，尤其在项目的后期。



1.1 项目进度（时间）管理过程

- ❑ 活动定义（Activity definition）
- ❑ 活动排序（Activity sequencing）
- ❑ 活动历时估计（Activity duration estimating）
- ❑ 任务资源估计
- ❑ 制定进度计划（Schedule development）
- ❑ 进度控制（Schedule control）－项目跟踪



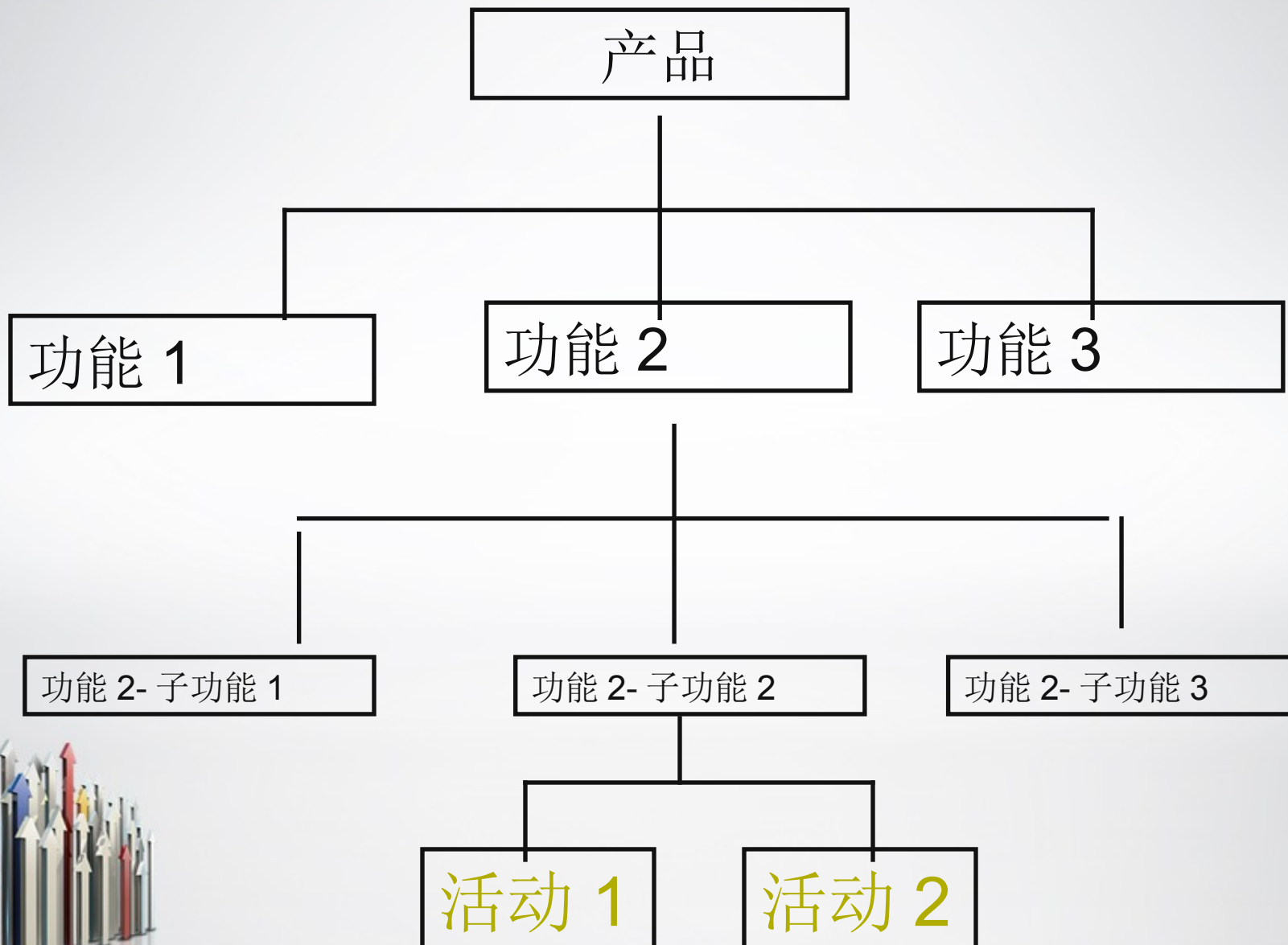
活动定义（Defining Activities）

- 确定为完成项目的各个交付成果所必须进行的诸项具体活动



活动定义

项目管理



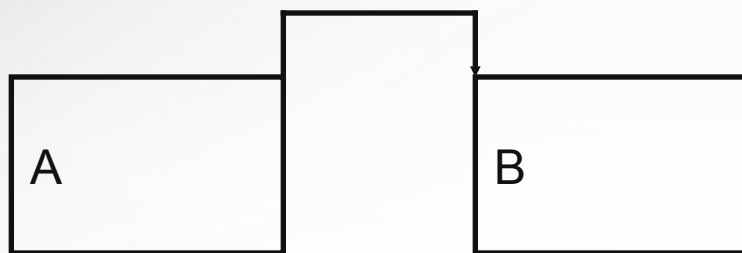
项目活动排序

- 项目各项活动之间存在相互联系与相互依赖关系，
- 根据这些关系进行适当的顺序安排

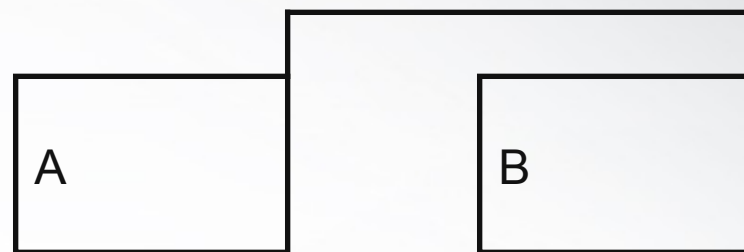
前置活动（任务） —— 〉 后置活动（任务）



任务（活动）之间的关系



结束 - 开始



结束 - 结束



开始 - 开始



开始 - 结束



任务（活动）之间排序的依据

- 强制性依赖关系
- 软逻辑关系
- 外部依赖关系
- 里程碑

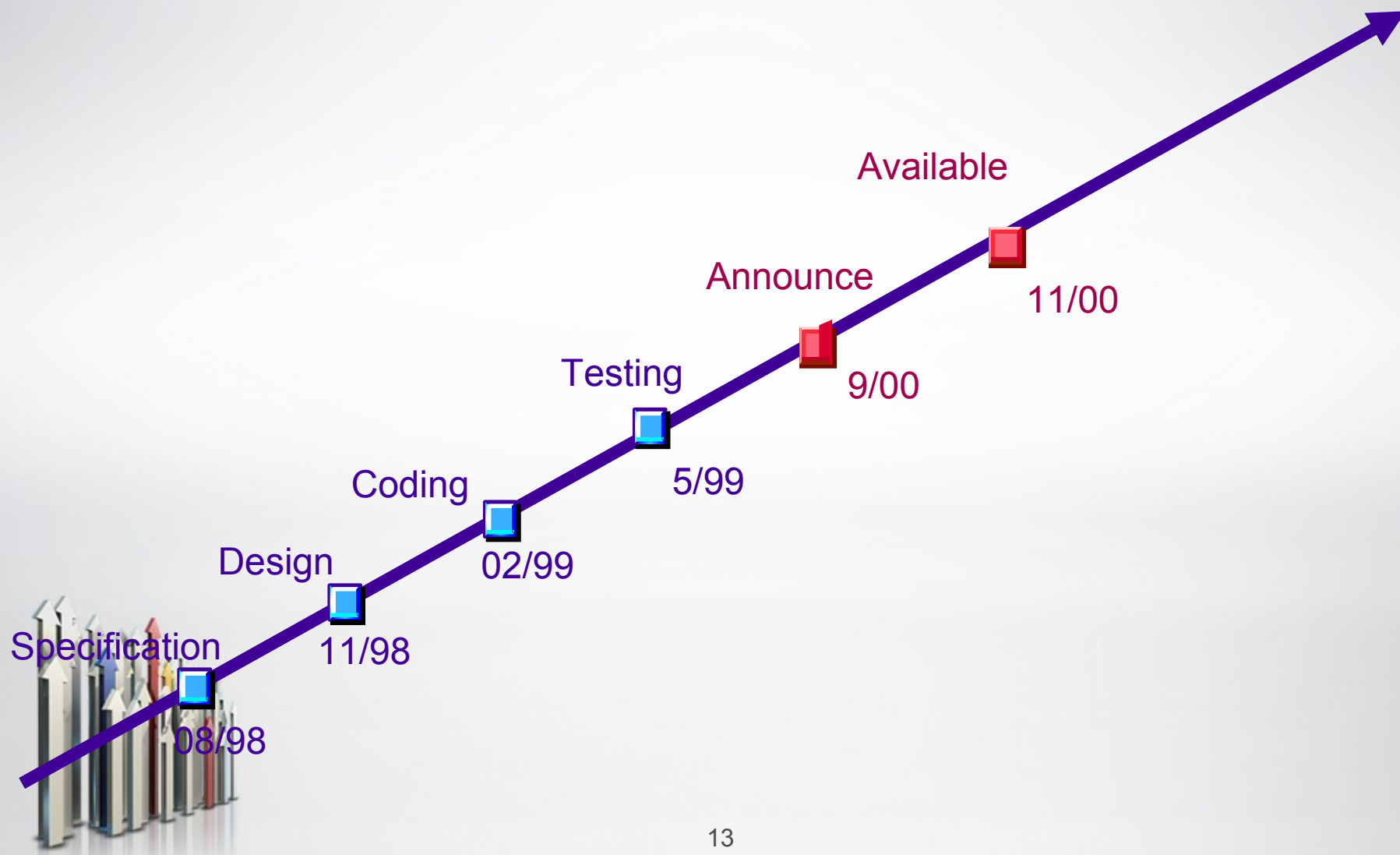


1.2 进度管理图示

- 网络图
- 甘特图
- 里程碑图
- 资源图



里程碑图示



里程碑图示

里程碑图

图6-7

事件	一月	二月	三月	四月	五月	六月	七月	八月
A			△ ▼					
B				▽				
C					△			
D						△		
E							△	
F								△

里程碑图示

- ❑ 里程碑显示项目进展中的重大工作完成
- ❑ 里程碑不同于活动
 - ❑ 活动是需要消耗资源的
 - ❑ 里程碑仅仅表示事件的标记



进度计划表格——里程碑计划

里程碑计划是以项目某些重要事件的完成或开始时间点作为基准所形成的计划，是个战略计划或框架，以中间产品或可实现的结果为依据。里程碑计划通常可用里程碑图或表的形式表达

里程碑事件	1月	2月	3月	4月	5月	6月	7月	8月	9月
A		▲							
B			▲						
C					▲				
D							▲		
E									▲

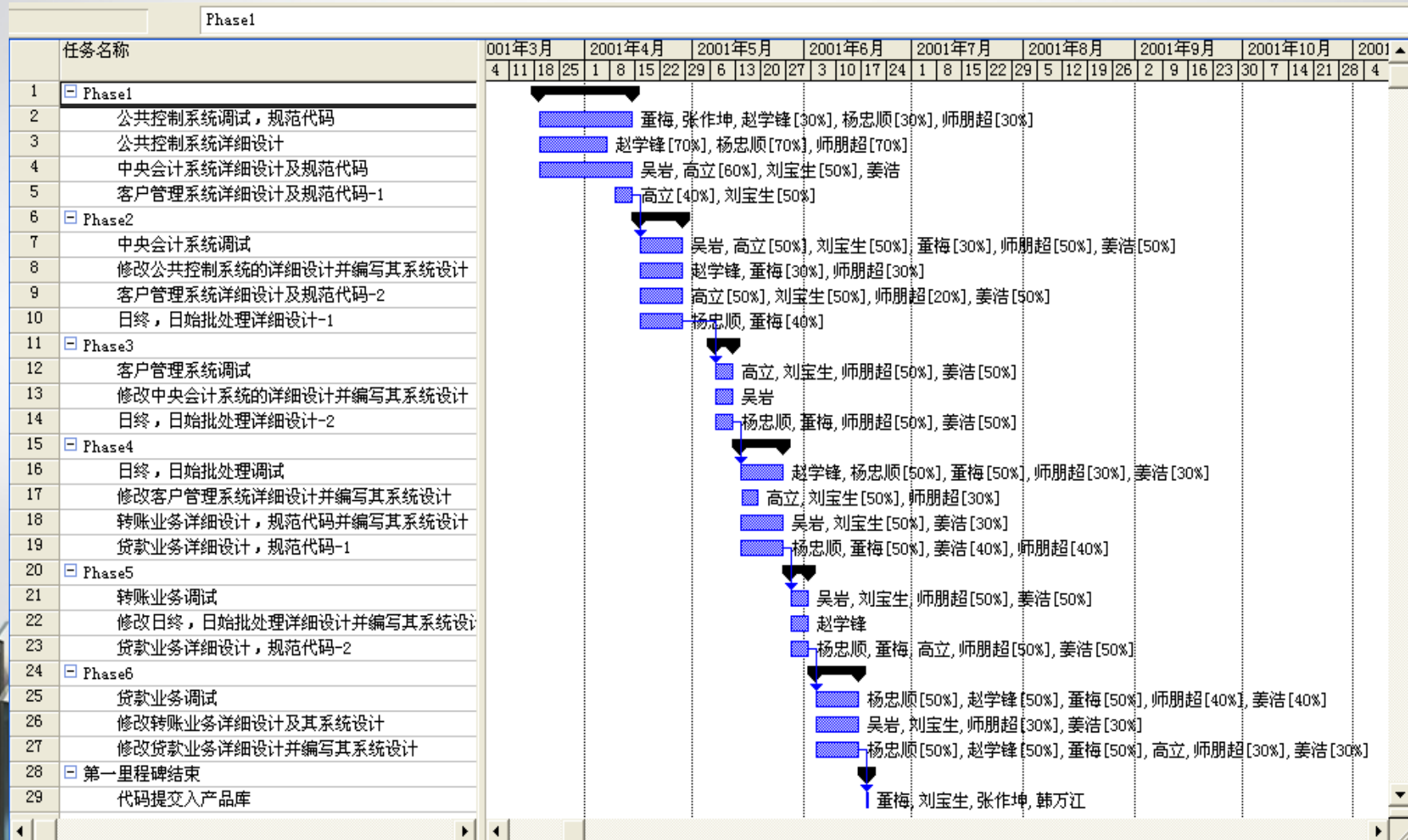


甘特图

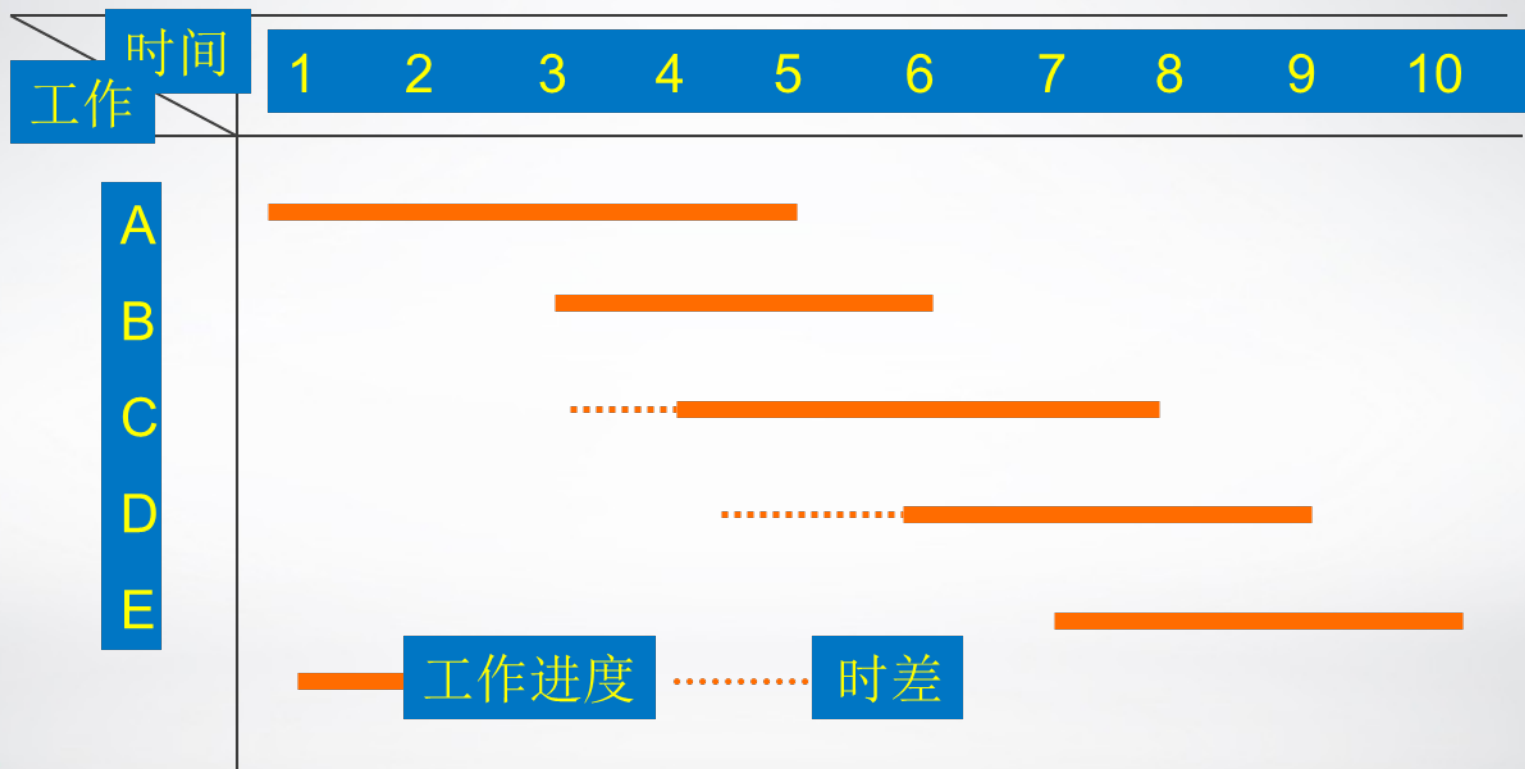
- 显示基本的任务信息
- 可以查看任务的工期、开始时间和结束时间以及资源的信息。
- 只有时标，没有活动的逻辑关系



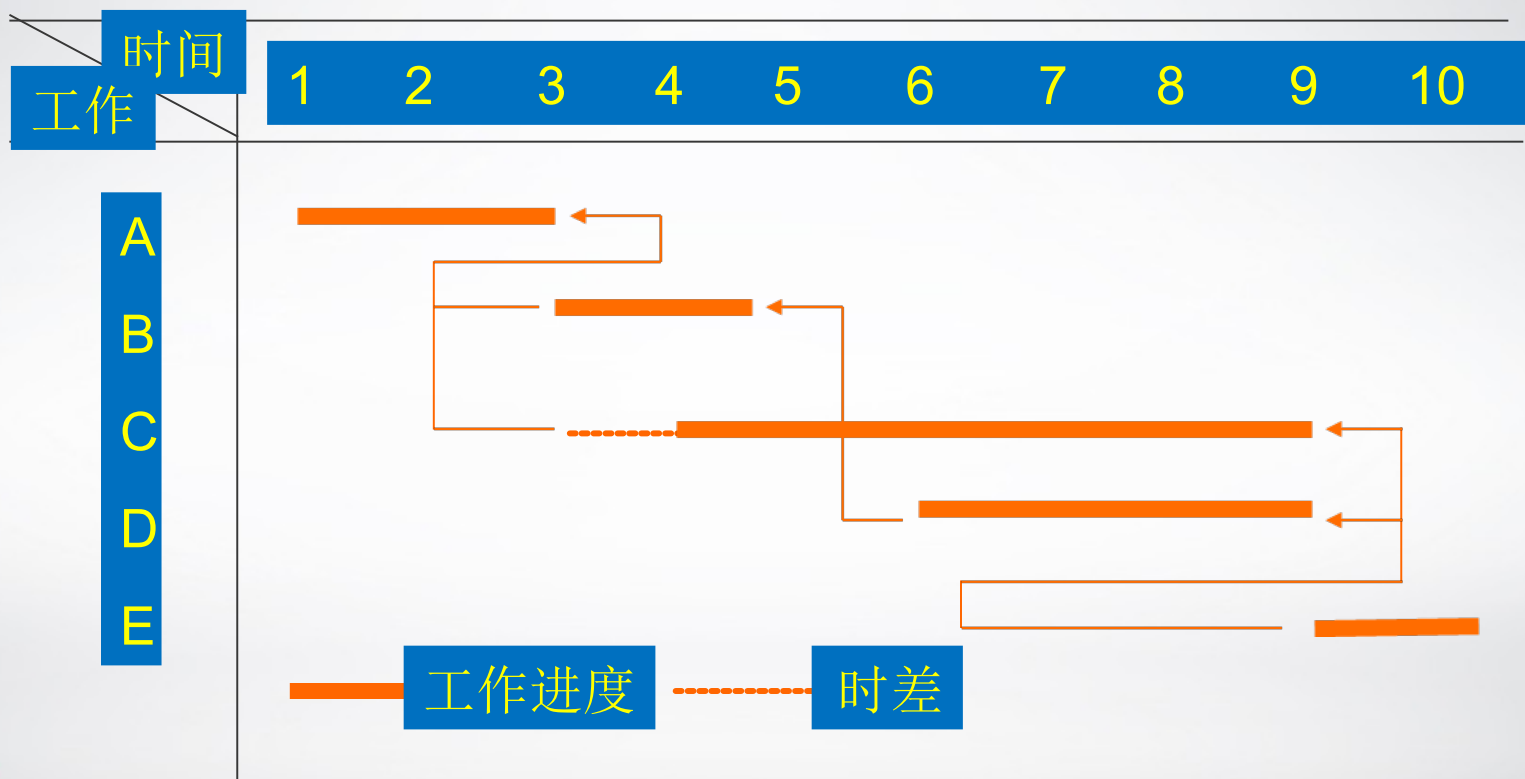
甘特图 - 实例



(1) 带有时差的甘特图



(2) 具有逻辑关系的甘特图

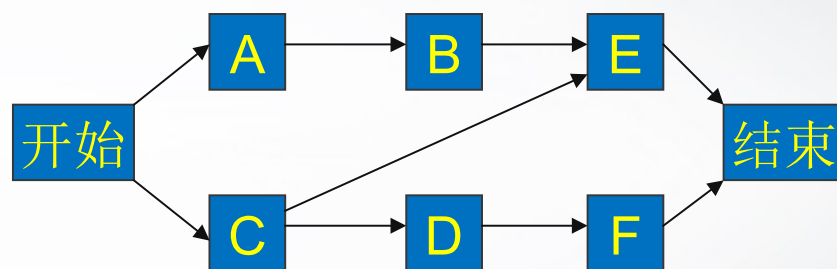


网络计划技术

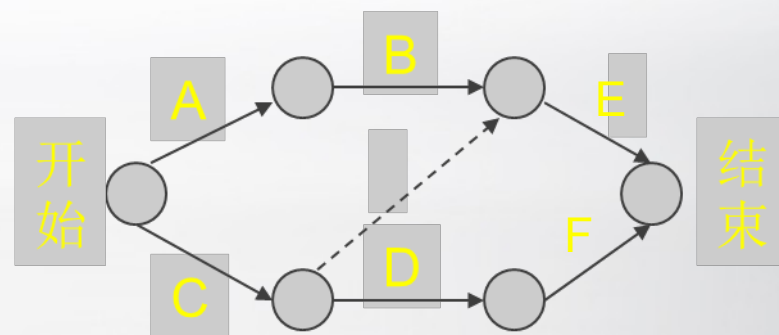
项目管理

网络计划技术是用网络计划对各种进度进行安排和控制，以保证预定目标的科学的计划管理技术。所谓网络计划就是用节点和箭线表示各种关系的网络图。它有两种表示形式：

（1）前导图法：这是一种用节点表示工作、箭线表示工作关系的项目网络图，如：



（2）箭线图法：这是一种用箭线表示工作、节点表示工作关系的项目网络图，如：



网络图

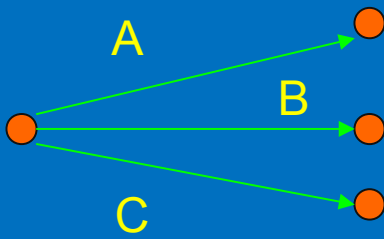
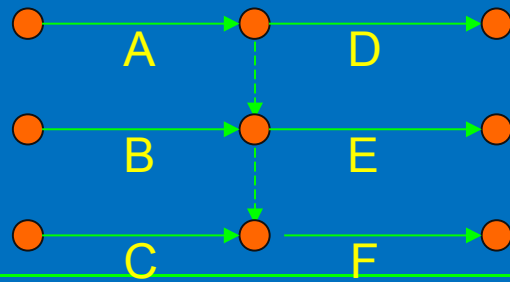
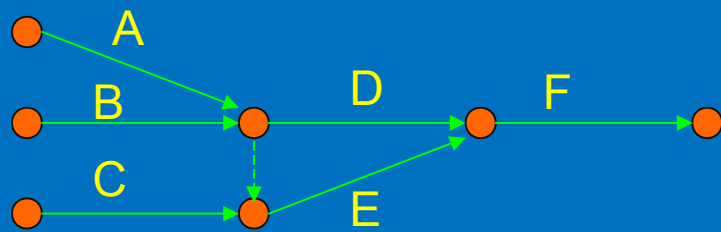
- 网络图是活动排序的一个输出
- 展示项目中的各个活动以及活动之间的逻辑关系
- 网络图可以表达活动的历时

- 前导图法（PDM）或 单代号网络计划（AON）
- 箭线图法（ADM）或双代号网络计划（AOA）
- 条件图法
- 图形评审技术 GERT
- 网络模板：标准的网络图



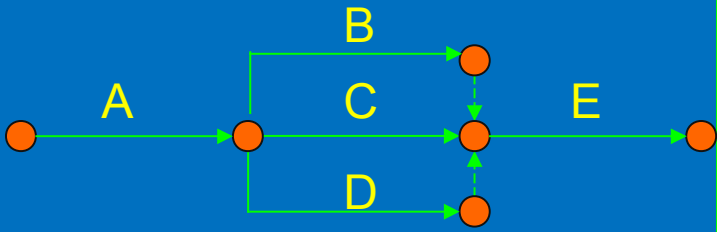
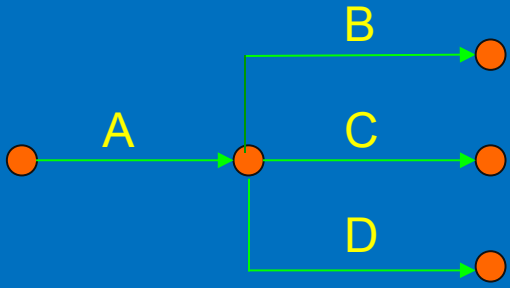
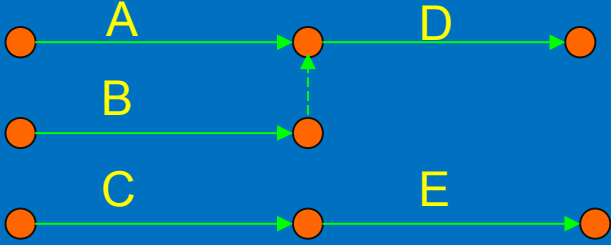
网络图中常见的逻辑关系及其表示方法

项目管理

序号	工作之间的逻辑关系	表示方法
1	A、B、C 平行进行	 The diagram shows a single starting node on the left. Three arrows branch out from this node to three separate ending nodes on the right. The top arrow is labeled 'A', the middle arrow is labeled 'B', and the bottom arrow is labeled 'C'.
2	A 完成后, D 才开始, AB 均完成后, E 才开始, ABC 均完成后, F 才开始。	 The diagram shows three horizontal paths. The top path has a node, an arrow labeled 'A' to a second node, and an arrow labeled 'D' to a third node. The middle path has a node, an arrow labeled 'B' to a second node, and an arrow labeled 'E' to a third node. The bottom path has a node, an arrow labeled 'C' to a second node, and an arrow labeled 'F' to a third node. A vertical dashed line connects the second nodes of the top, middle, and bottom paths, indicating that D starts after A, E starts after both B and C, and F starts after A, B, and C.
3	AB 均完成后, D 才开始, ABC 均完成后, E 才开始, DE 完成后, F 才开始。	 The diagram shows three horizontal paths. The top path has a node, an arrow labeled 'A' to a second node, and an arrow labeled 'D' to a third node. The middle path has a node, an arrow labeled 'B' to a second node, and an arrow labeled 'E' to a third node. The bottom path has a node, an arrow labeled 'C' to a second node, and an arrow labeled 'F' to a third node. A vertical dashed line connects the second nodes of the top, middle, and bottom paths, indicating that D starts after both A and B, E starts after both B and C, and F starts after D and E.

网络图中常见的逻辑关系及其表示方法

项目管理

序号	工作之间的逻辑关系	表示方法
4	A 完成后, B C D 才开始, B C D 完成后, E 才开始。	
5	A 完成后, B C D 才开始, 但对 B C D 的结束不作要求。	
6	A、B 完成后, D 才开始, B C 完成后, E 才开始。	

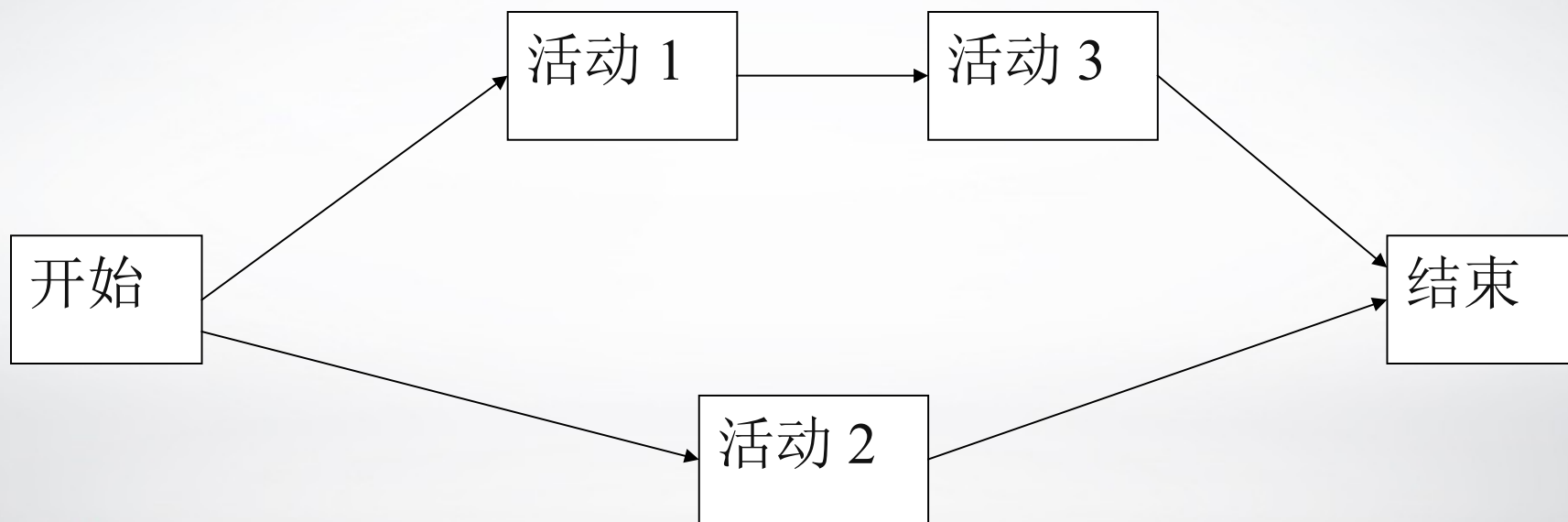


PDM (Precedence Diagramming Method)

- ❑ 构成 PDM 网络图的基本特点是节点 (Box)
- ❑ 节点 (Box) 表示活动 (工序, 工作)
- ❑ 用箭线表示各活动 (工序, 工作) 之间的逻辑关系.
- ❑ 可以方便的表示活动之间的各种逻辑关系。
- ❑ 在软件项目中 PDM 比 ADM 更通用

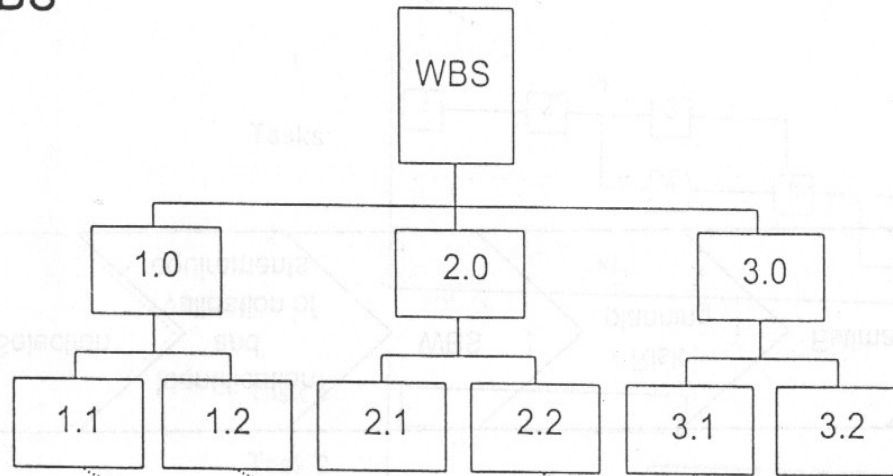


PDM 图例

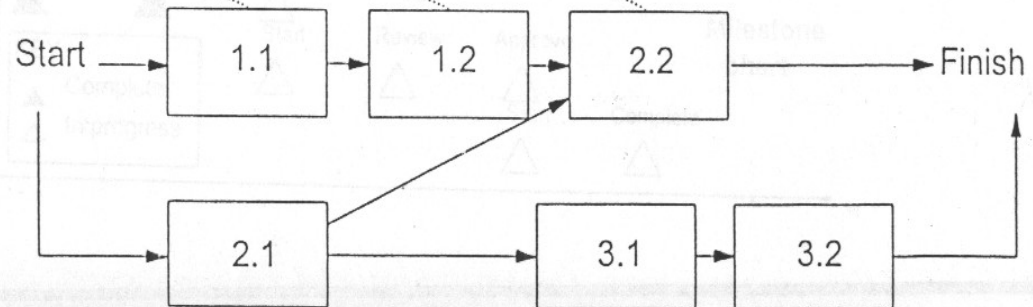


Network Relationship to WBS

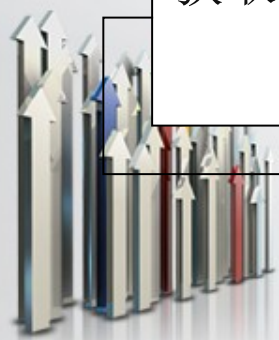
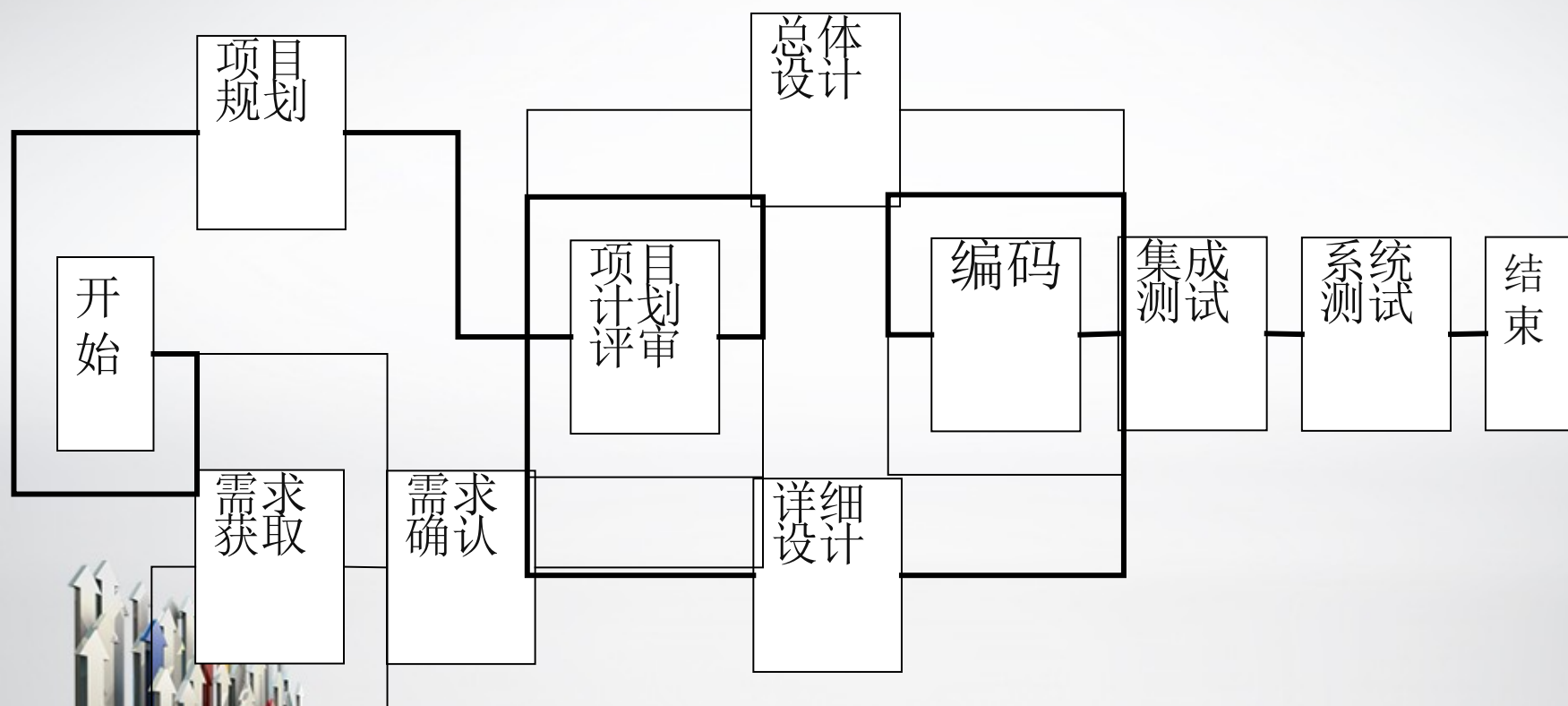
WBS



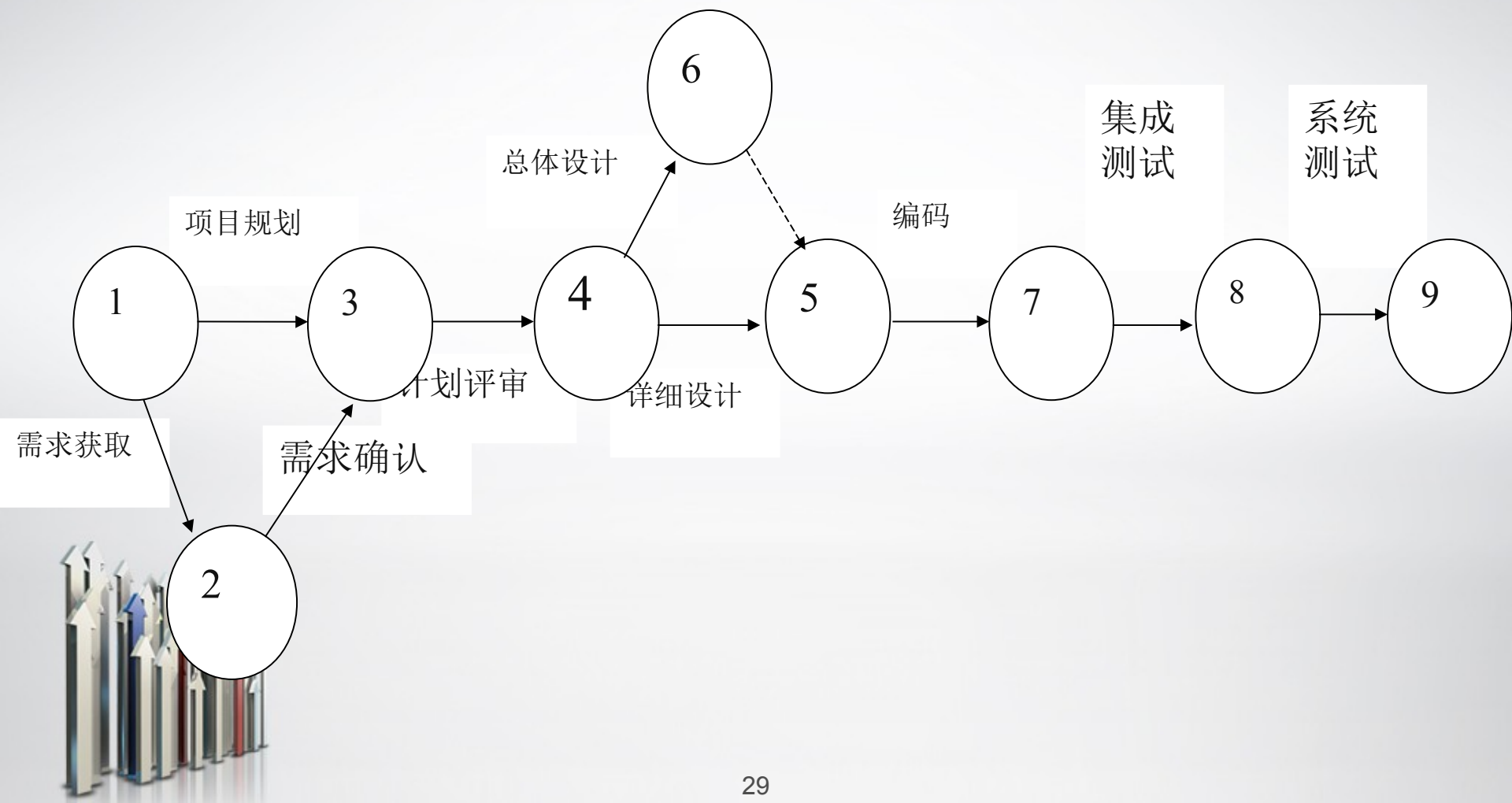
Network



节点图 PDM (Precedence Diagramming Method) – 优先图法图例 Project 软件使用



箭线图 ADM 图例



ADM (Arrow Diagramming Method)

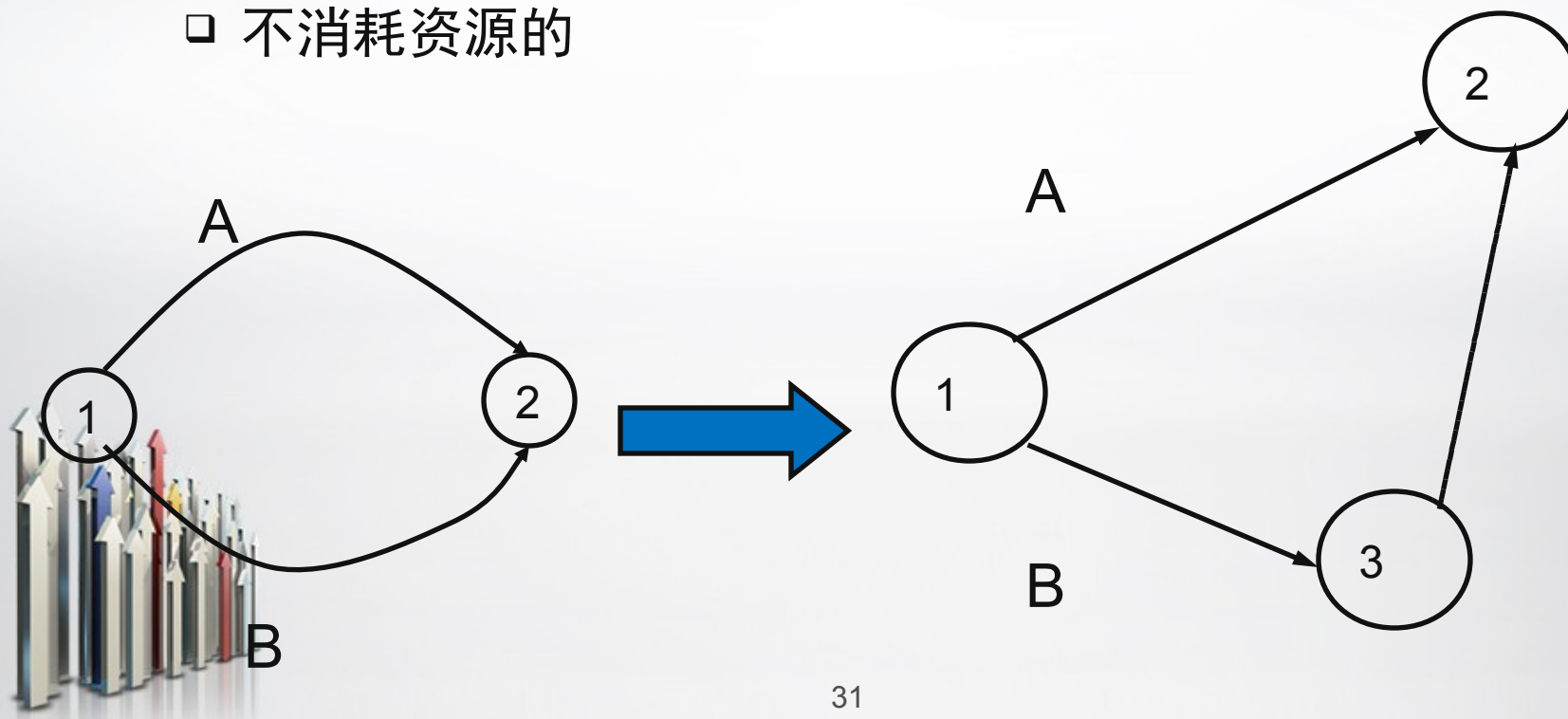
- ❑ ADM 也称为 AOA (activity-on-arrow) 或者双代号项目网络图,
- ❑ 在 ADM 网络图中, 箭线表示活动 (工序 \ 工作),
- ❑ 节点 Node (圆圈 : circle) 表示前一道工序的结束, 同时也表示后一道工序的开始 .
- ❑ 只适合表示结束 - 开始的逻辑关系



ADM 图例 - 虚活动

□ 虚活动

- 为了定义活动
- 为了表示逻辑关系
- 不消耗资源的



项目进度估算的基本方法

- ❑ 基于规模的进度估算
 - ❑ 定额估算法
 - ❑ 经验导出模型
- ❑ CPM
- ❑ PERT
- ❑ 基于进度表的进度估算
- ❑ 基于承诺的进度估计
- ❑ 其它策略



项目进度估算 – 历时估计

- **项目进度估算是估计任务的持续时间——历时估计**
 - 每个任务的历时估计
 - 项目总历时估计



关键路径法估计

(CPM : Critical Path Method)

- 根据指定的网络顺序逻辑关系，进行单一的历时估算
- 当估算项目中某项单独的活动，时间比较确定的时候采用



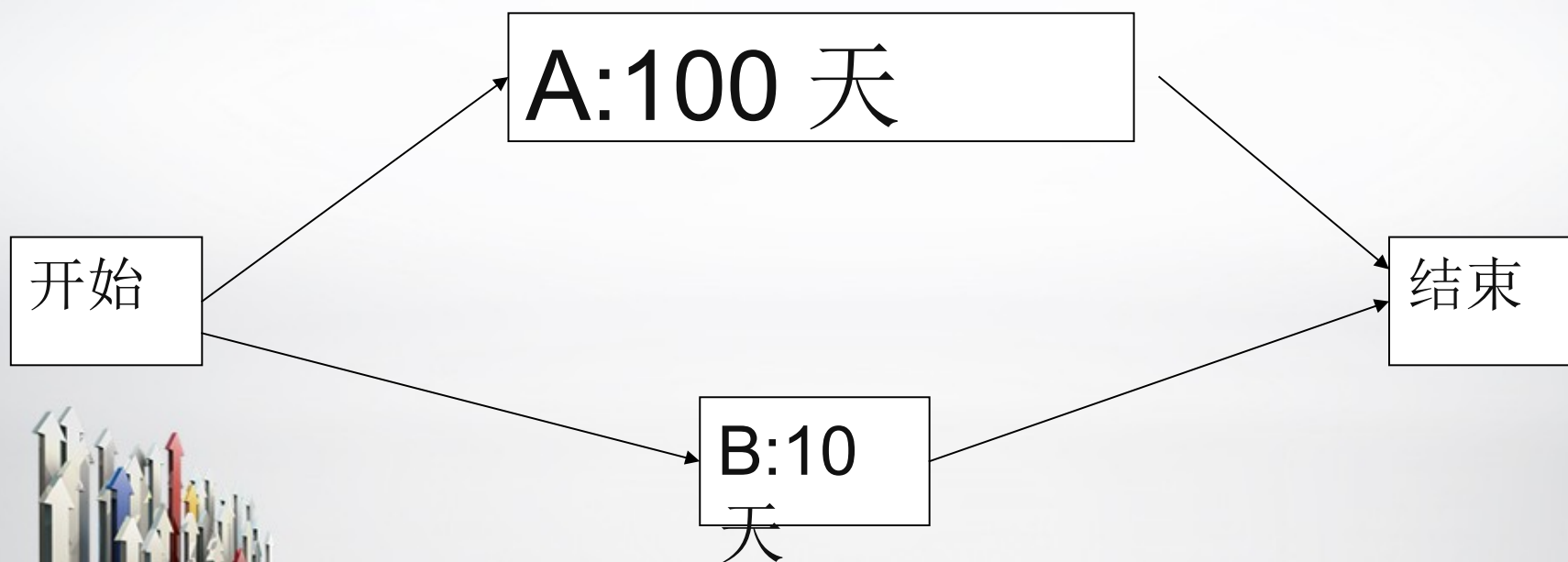
关键路径 (critical path)

关键路径：从项目开始到项目完成有许多条路径，在整个网络图中最长（节点最多或者花费时间最多）的路径就叫关键路径。

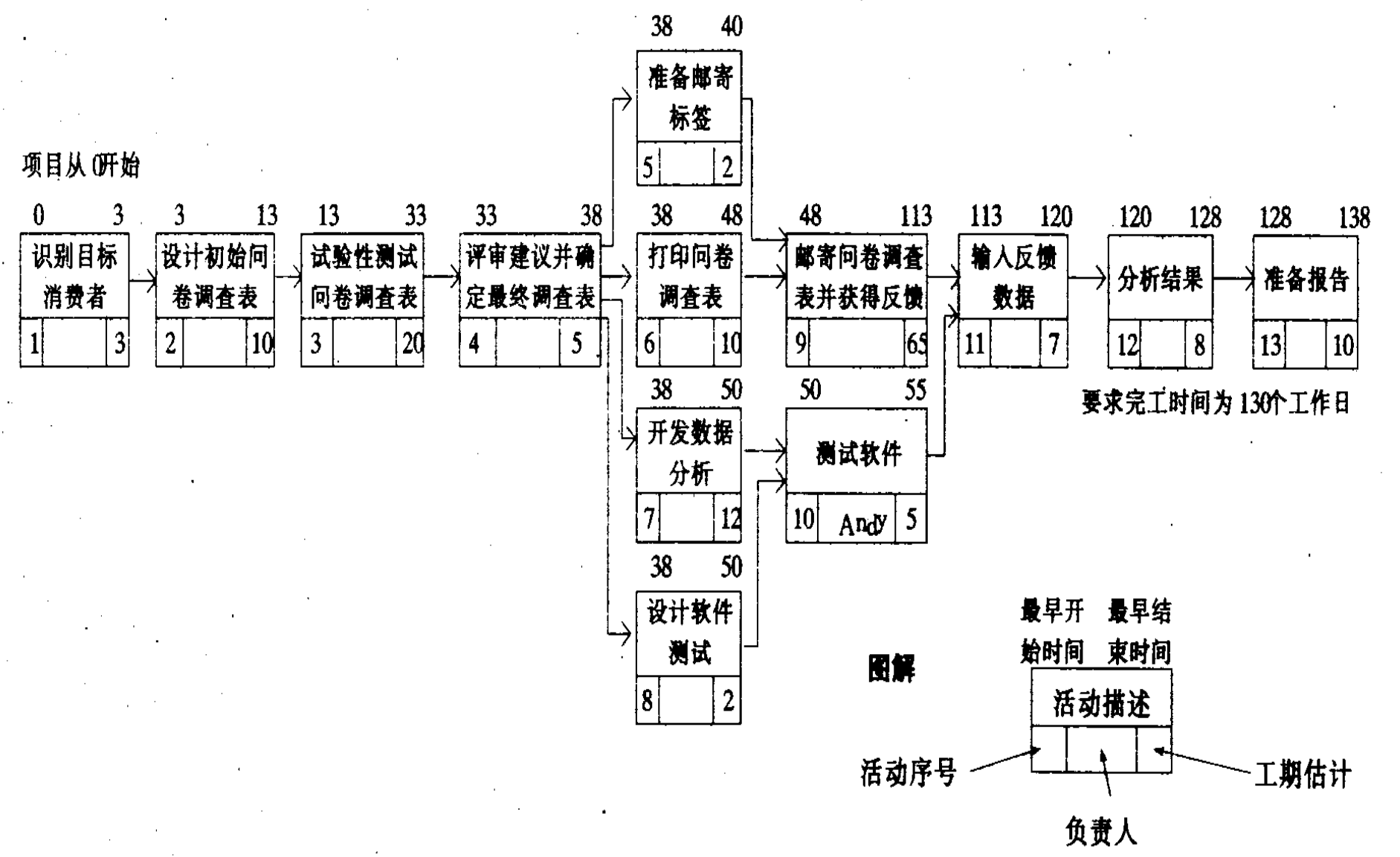
非关键路径 (noncritical path)：
在整个网络图中非最长的路径都叫非关键路径。



CPM 估计

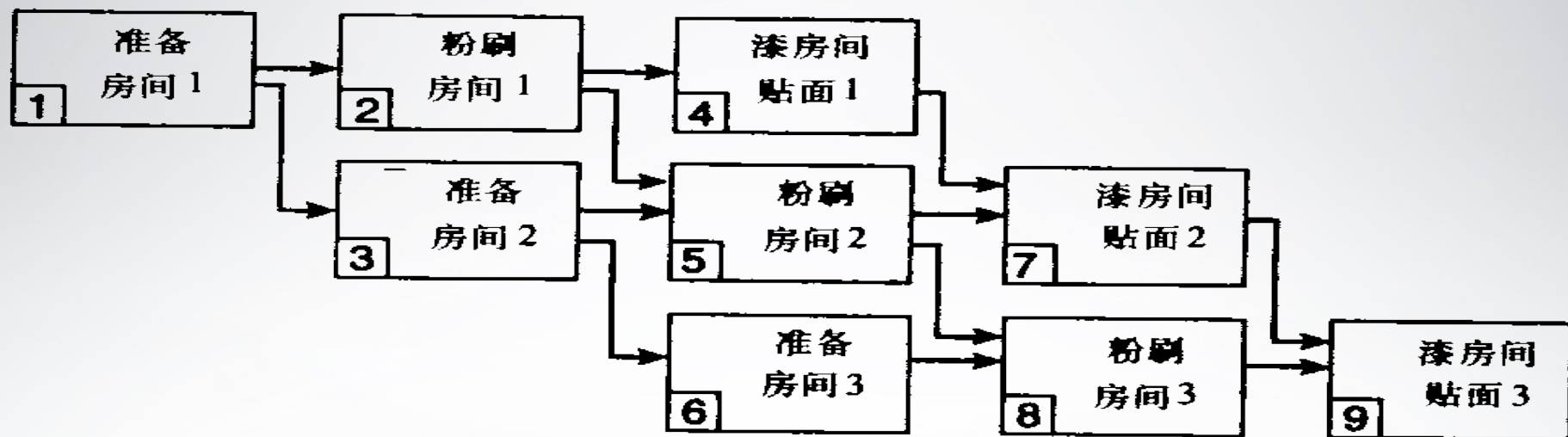


实例 某产品市场研究项目单代号网络图

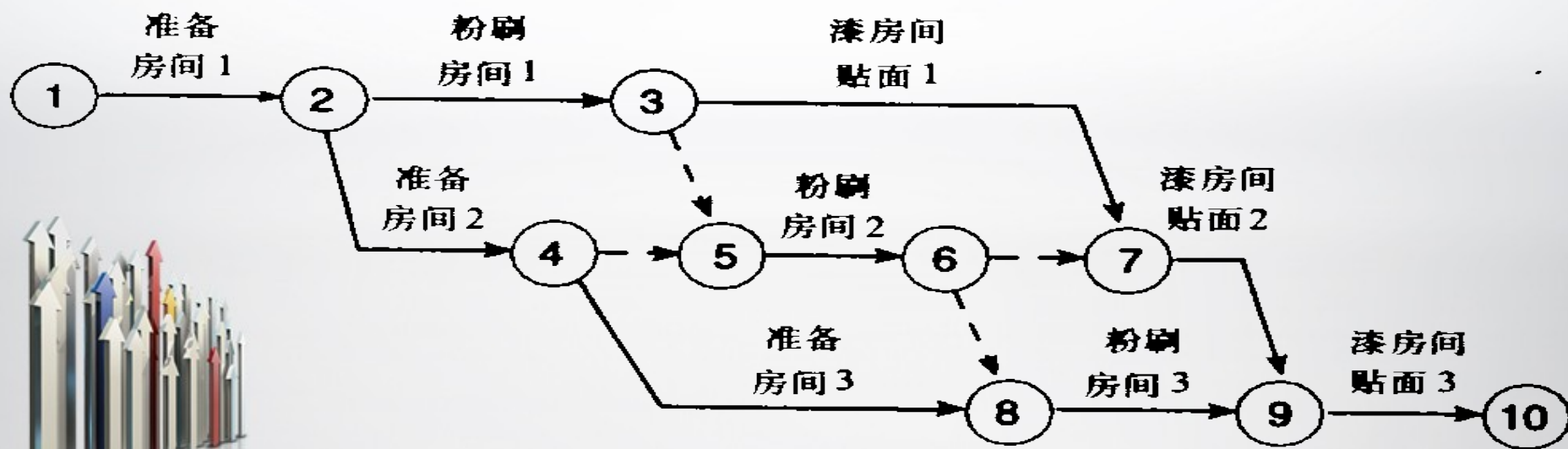


粉刷房间的项目安排（并行交叉安排）

项目管理



a) 用节点表示活动形式



b) 用箭头表示活动形式

本单元要点

- 一、进度管理的基本概念及过程
- 二、进度估算的基本方法
- 三、任务资源估计
- 四、编制进度计划
- 五、案例分析



工序、工期三值估计



- To(乐观估计 Optimistic Time)
- Tm(最可能估计 Most likely Time)
- Tp(悲观估计 Pessimistic Time)



基于进度表估算

1. 可能的最短进度表
2. 有效进度表
3. 普通进度表



可能的最短进度表 – 人员

- ❑ 人才库中前 10% 的最拔尖的人,
- ❑ 有几年应用领域的工作经验,
- ❑ 开发人员掌握了应用领域的详细知识,
- ❑ 目标明确, 努力工作,
- ❑ 分享成果, 团队和谐
- ❑ 不存在人员调整



可能的最短进度表 - 管理

- 理想的项目管理
- 开发人员可以专著于本职工作
- 采用矩形员工模式



可能的最短进度表 - 工具支持

- ❑ 有先进的工具
- ❑ 开发人员可以无限制的使用资源
- ❑ 工作环境理想，在集中的工作区域开发
- ❑ 交流工具畅通



可能的最短进度表 - 方法

- 使用最时效的开发方法和开发工具
- 设计阶段开始的时候已经完全了解需求
- 需求不变更



可能的最短进度表 - 压缩

- 尽可能的压缩进度，直到不能压缩



基于进度表估算

1. 可能的最短进度表
2. 有效进度表
3. 普通进度表



有效进度表 – 人员

- ❑ 人才库中前 25% 的最拔尖的人,
- ❑ 有 1 年应用领域的工作经验,
- ❑ 目标有共同的看法, 相互之间没有严重冲突,
- ❑ 采用有效的人员模式
- ❑ 人员调整少于 6%



有效进度表 - 其它

- 有效的工具
- 主动的风险管理
- 优良的物理环境
- 沟通工具方便



基于进度表估算

1. 可能的最短进度表
2. 有效进度表
3. 普通进度表



普通进度 - 人员

- ❑ 人才库中等以上的人
- ❑ 对应用领域一般熟悉
- ❑ 开发人员对应用领域有一定的经验，但不丰富
- ❑ 团队不是很有凝聚力，但解决冲突时，有一定的经验
- ❑ 每年经历人员调整 10-12%



普通进度 - 其它

- ❑ 工具在一定程度上使用
- ❑ 风险管理不像理想那样得力
- ❑ 交流工具容易使用,
- ❑ 工作环境有些一般, 不是很理想
- ❑ 进度压缩一般



三种进度比较

- ❑ 可能的最短进度简直无法实现
- ❑ 有效进度代表了“最佳进度”
- ❑ 普通进度是为一般项目实用的



项目进度估算的基本方法

- 基于规模的进度估算，
 - 定额计算法
 - 经验导出方程
- PERT
- CPM
- 基于进度表的进度估算
- 基于承诺的进度估计
- 其它策略



基于承诺的进度估计

- 从需求出发去安排进度
- 不进行中间的工作量（规模）估计
- 要求人员做出进度承诺，非进度估算



基于承诺的进度估计 - 优点

- 有利于团队对进度的关注
- 有利于团队在接受承诺之后的士气高昂



基于承诺的进度估计 - 缺点

- ❑ 工作人员估计的乐观
- ❑ 易于产生大的估算误差



工程评价技术（PERT）

- ❑ (Program Evaluation and Review Technique) 利用网络顺序图逻辑关系和加权历时估算来计算项目历时的技术。
- ❑ 当估算项目中某项单独的活动，存在很大的不确定性时采用。



工程评估评审技术（PERT）

- 它是基于对某项任务的乐观，悲观以及最可能的概率时间估计
- 工序、工期三值估计
 - T_o (乐观估计 Optimistic Time)
 - T_m (最可能估计 Most likely Time)
 - T_p (悲观估计 Pessimistic Time)
- 采用加权平均得到期望值 $E = (O + 4m + P) / 6$,
 - O 是最小估算值：乐观 (Optimistic) ,
 - P 是最大估算值：悲观 (Pessimistic) ,
 - M 是最大可能估算 (Most Likely) 。



PERT 计算举例

PERT 权重值 =

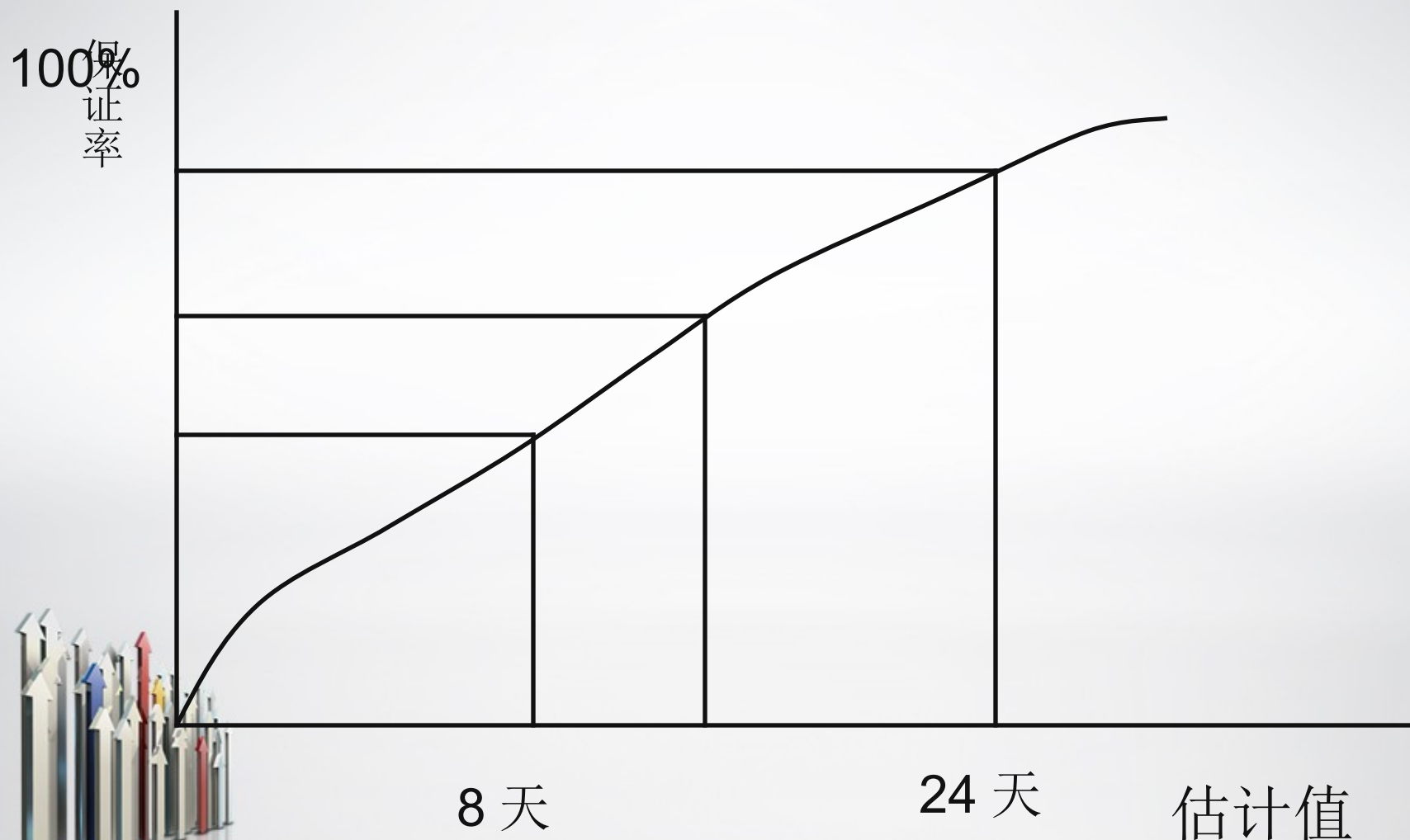
$$\frac{8 \text{ 日} + 4 \times 10 \text{ 日} + 24 \text{ 日}}{6} = 12 \text{ 日}$$

当 8 日 = 乐观估算，10 日 = 最可能估算，24 日 = 悲观估算

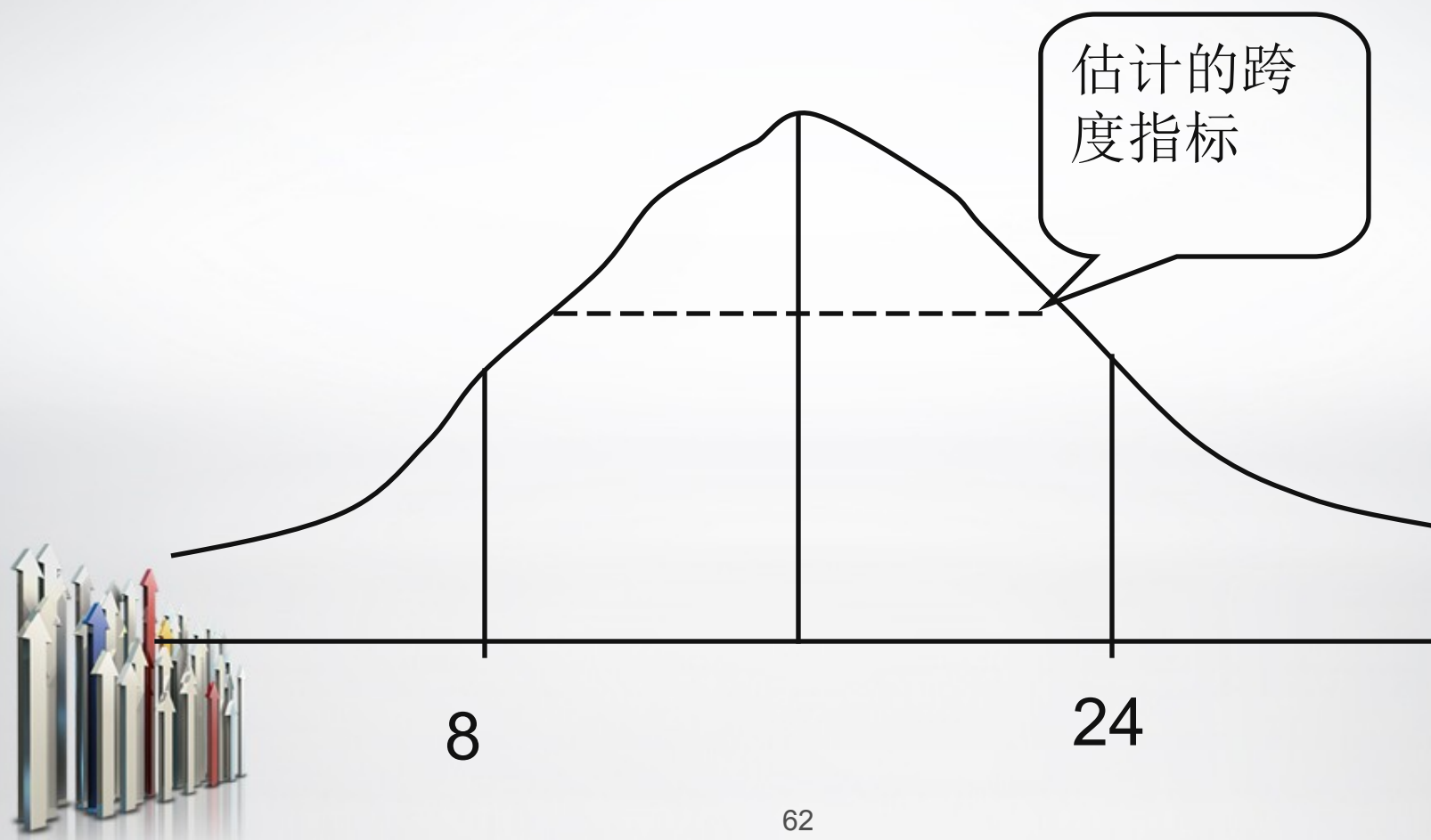
日 指工作日。



PERT 的保证率



PERT 的度量指标



PERT 的评估进度风险

- 标准差 $\delta = (\text{最大估算值} - \text{最小估算值}) / 6$
 - 方差 $\delta^2 = [(\text{最大估算值} - \text{最小估算值}) / 6]^2$

 - 具体介绍见课外资料
-



PERT/CPM 区别

□ PERT 计划评审技术

- 计算历时采用的算法：加权平均 $(O+4m+P)/6$
- 估计值不明确

□ CPM 关键路径法

- 计算历时采用的算法：最大可能值 m
- 估计值比较明确



定额估算法

$$T=Q/(R*S)$$

- T: 活动持续时间
- Q: 活动的工作量
- R: 人力或设备的数量
- S: 产量定额，以单位时间完成的工作量表示



定额估算法

□ 例如

□ $Q=6$ 人月 , $R=2$ 人 , $S=1$

□ 则: $T=3$ 月

□ 例如

□ $Q=6$ 人月 , $R=2$ 人 , $S=1.5$

□ 则: $T=2$ 月



定额估算法

- 方法比较的简单，容易计算。
- 适合项目的规模比较小，比如说小于 10000LOC 或者说小于 6 个月的项目



经验导出模型

- 经验导出模型： $D = a * E \exp(b)$:
 - D: 月进度
 - E: 人月工作量
 - $a = 2 \sim 4$
 - $b: 1/3$ 左右: 依赖于项目的自然属性



估算的其他策略

- ❑ 专家估算方法
- ❑ 类推估计
- ❑ 模拟估算
- ❑ 利用估算软件估算进度
- ❑ 利用企业的历史数据

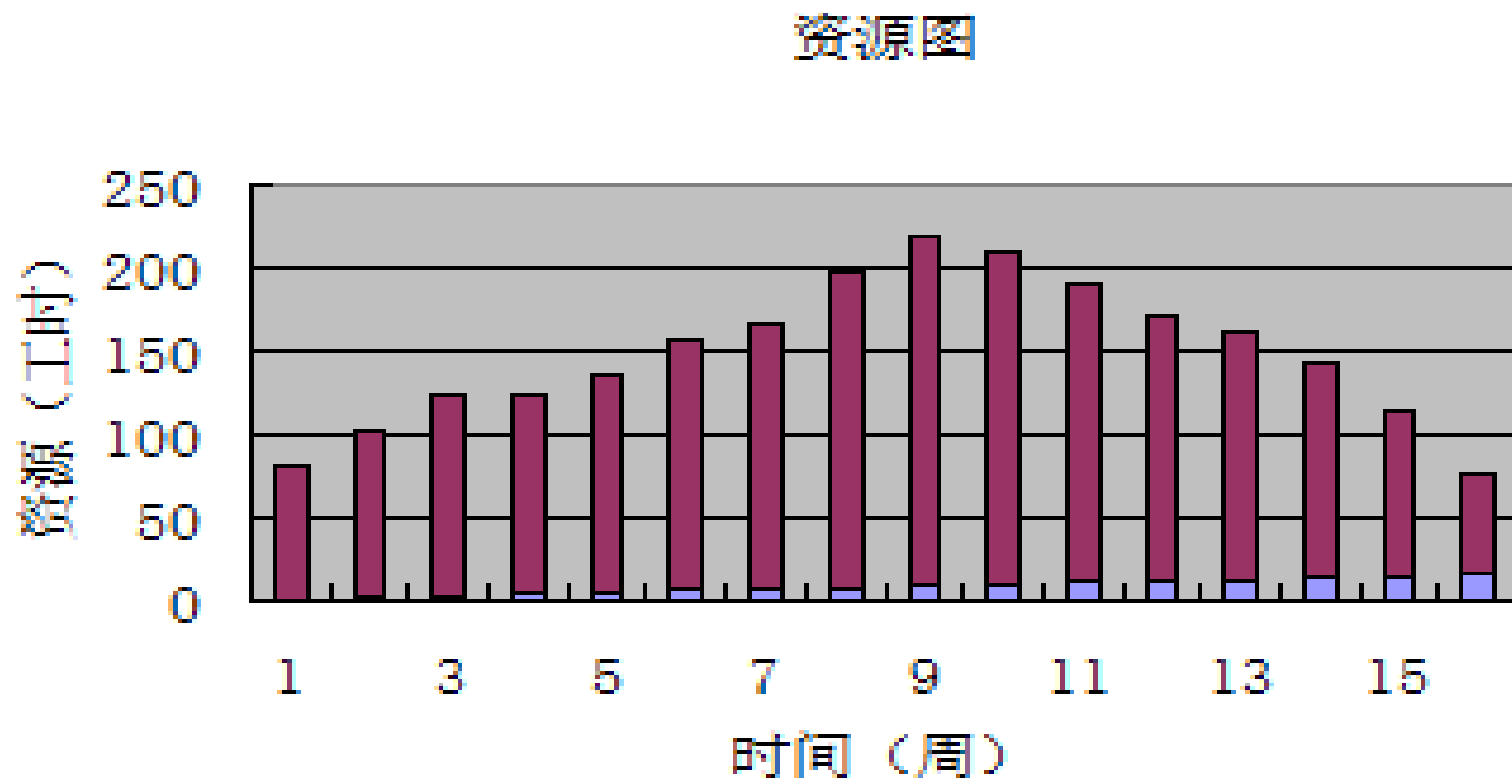


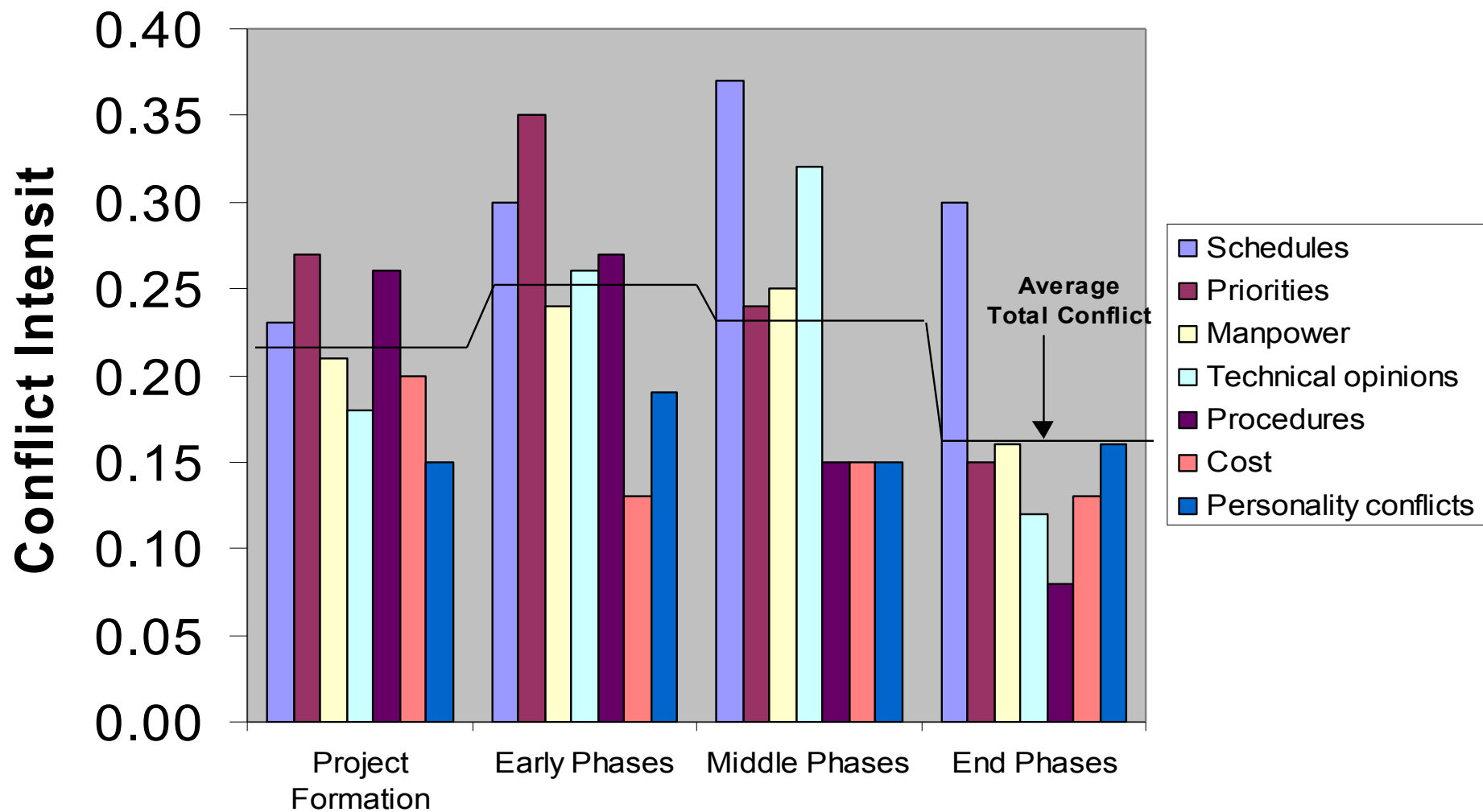
三、任务资源估计

- **每个任务需要的资源类型和数量有一定的考虑，这些资源包括，人力资源，设备资源，以及其它资料资源等**



资源图





本单元要点

- 一、进度管理的基本概念及过程
- 二、进度估算的基本方法
- 三、任务资源估计
- 四、编制进度计划



编制项目进度计划

- 确定项目的所有活动及其开始和结束时间
- 计划是三维的，考虑时间，费用和资源
- 监控项目实施的基础，它是项目管理的基准



编制项目进度计划步骤

1. 进度编制
2. 资源调整
3. 成本预算
4. 计划优化调整
5. 计划基线



进度编制的基本方法

- 关键路径法
 - 正推法
 - 逆推法
- 时间压缩法
 - 赶工（Crash）
 - 快速跟进（Fast tracking: 搭接）
- 关键链法



关键路径法

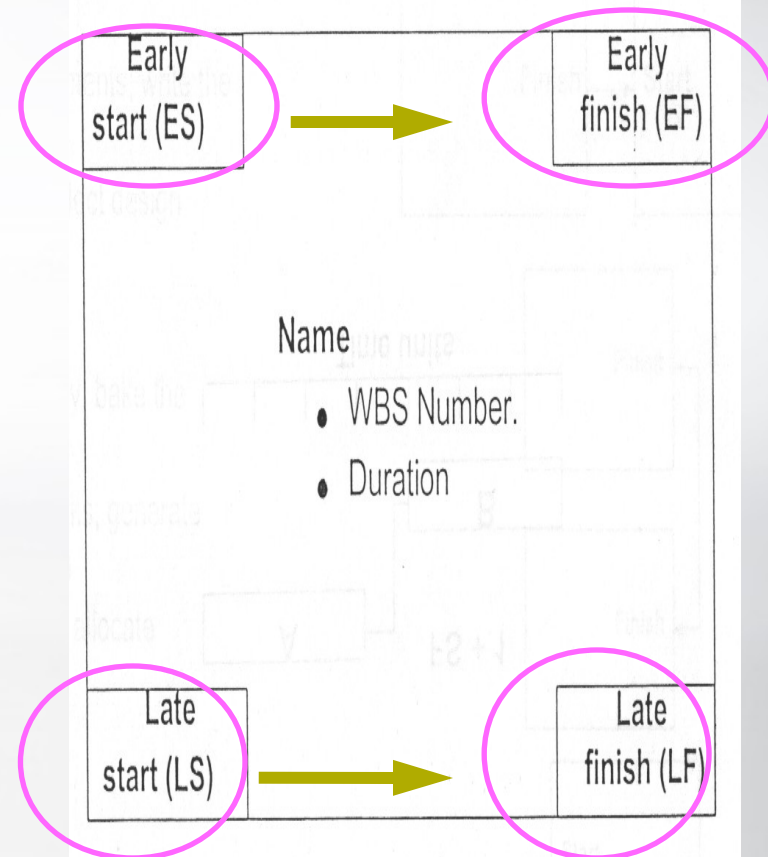
CPM : Critical Path Method)

- 根据指定的网络图逻辑关系和单一的历时估算，计算每一个活动的单一的、确定的最早和最迟开始和完成日期。
- 计算浮动时间。
- 计算网络图中最长的路径。
- 确定项目完成时间



网络图中任务进度时间参数说明 (p74)

- ❑ 最早开始时间 (Early start)
- ❑ 最晚开始时间 (Late start)
- ❑ 最早完成时间 (Early finish)
- ❑ 最晚完成时间 (Late finish)
- ❑ 自由浮动 (Free Float)
- ❑ 总浮动 (Total Float)
- ❑ 超前 (Lead)
- ❑ 滞后 (Lag)



浮动时间 (Float)

- 浮动时间是一个活动的机动性，它是一个活动在不影响其它活动或者项目完成的情况下可以延迟的时间量

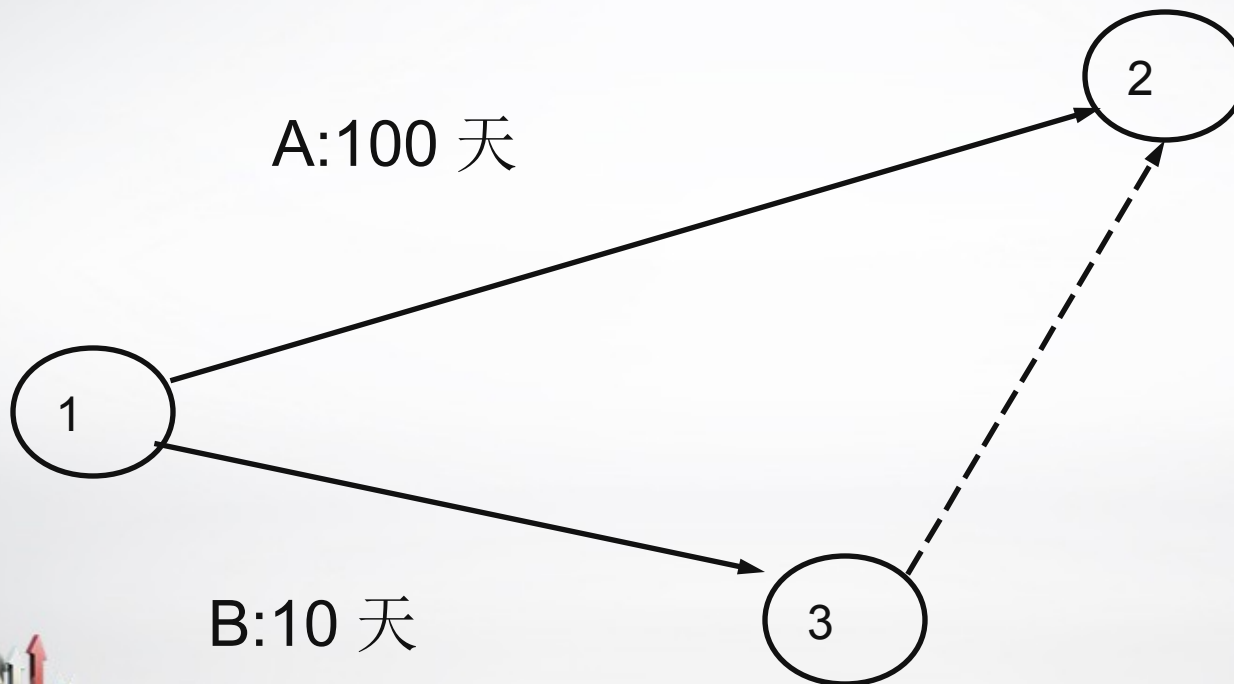


自由与总浮动时间

- **总浮动（ Total Float ）**
 - 在不影响项目最早完成时间本活动可以延迟的时间
- **自由浮动（ Free Float ）**
 - 在不影响后置任务最早开始时间本活动可以延迟的时间



CPM 估计



进度时间参数

A:

ES=0,EF=100

LS=0,LF=100

A:100

B:

ES=0,EF=10

LS=90 ,

LF=100

TF=LS-

ES=90

TF=LF-

EF=90

B:10

B:10

公式： $EF = ES + \text{duration}$

$LS = LF - \text{duration}$

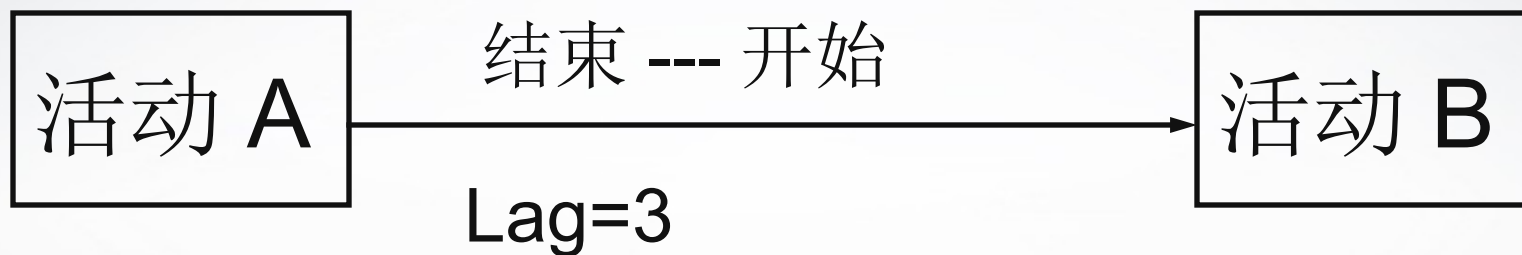
$TF = LS - ES = LF - EF$

提示： E 最早 L 最晚
S 开始 F 结束



任务滞后 Lag

A 完成之后 3 天 B 开始



进度时间参数

B:

ES=0,EF=10

LS=80,LF=90

TF=LS-ES=80

FF= 0

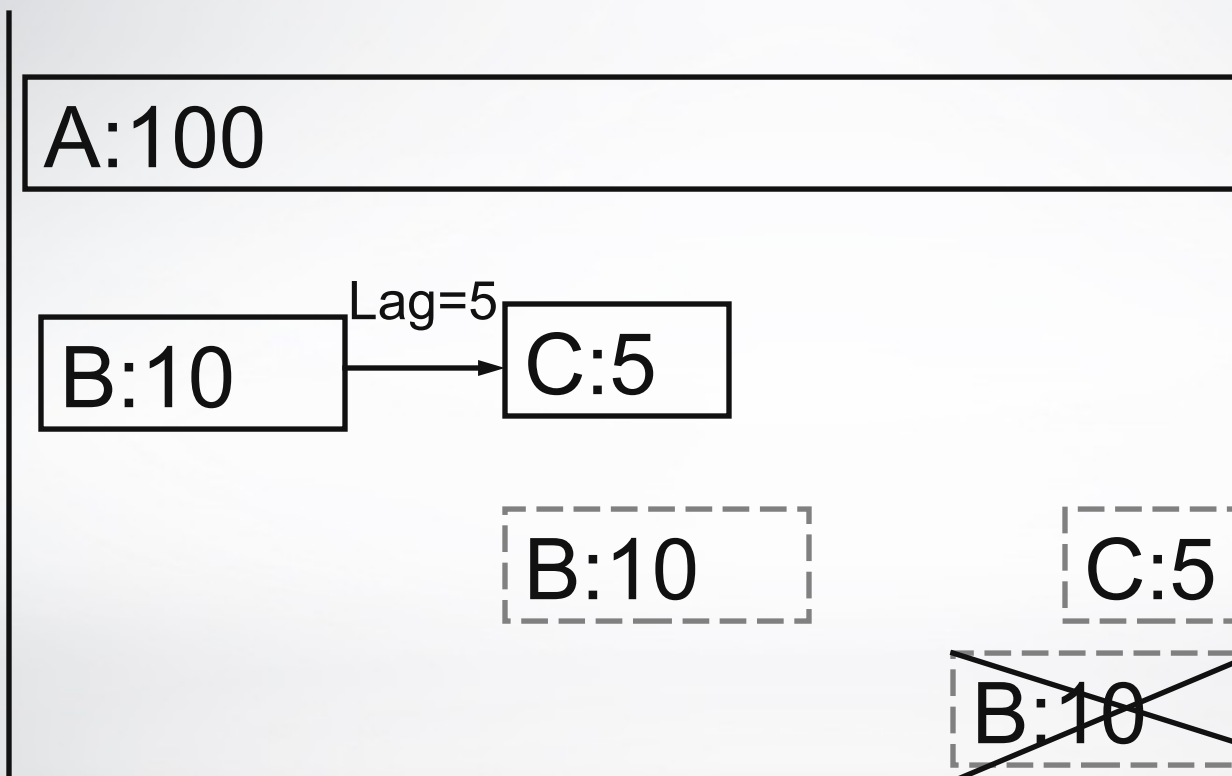
C:

ES=15,EF=20

LS=95,LF=10

0

TF=LS-ES=80



公式: $ES(S) = EF(P) + Lag$, $LF(P) = LS(S) - Lag$

$TF = LS - ES$,

$FF = ES(S) - EF(P) - Lag$

提示: E 最早 L 最晚
S 开始 F 结束

Float Example

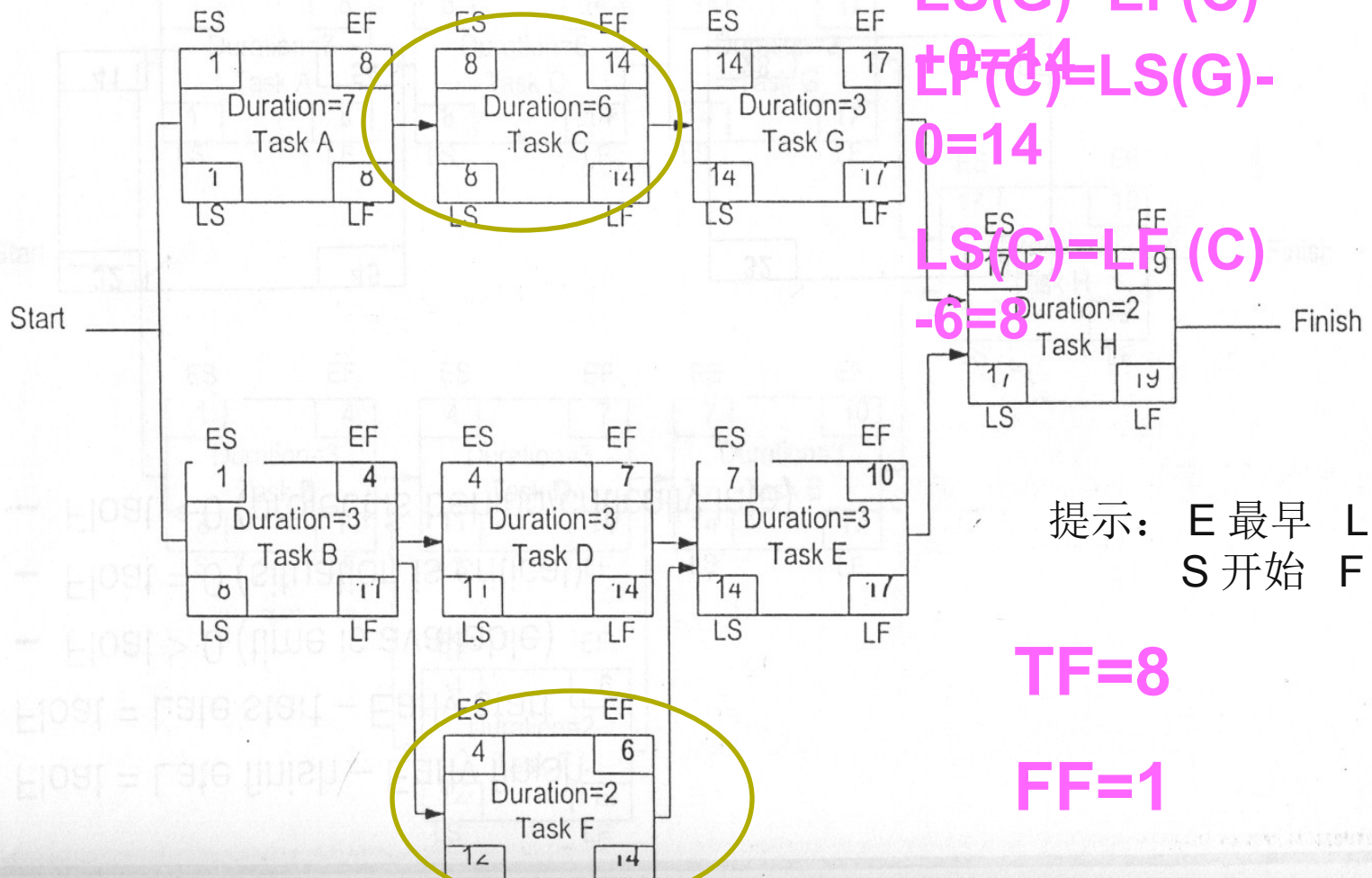
- Free float = $ES(\text{successor}) - EF(\text{predecessor})$
- Total float = $LS - ES$ or $LF - EF$

$$EF(C) = ES(C) + 6 = 14$$

$$ES(G) = EF(C) + 1 = 14$$

$$LF(C) = LS(G) - 0 = 14$$

$$LS(C) = LF(C) - 6 = 8$$



提示: E 最早 L 最晚
S 开始 F 结束

$$TF=8$$

$$FF=1$$

同时浮动？

B:

ES=0,EF=10

LS=80,LF=90

TF=LS-ES=80

FF= 0

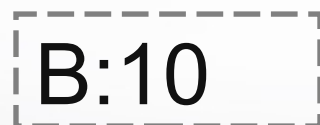
C:

ES=15,EF=20

LS=95,LF=10

0

TF=LS-ES=80



提示： E 最早 L 最晚
S 开始 F 结束

同时浮动时间

- B 可以浮动的时间: $80*10/15=53$
- C 可以浮动的时间: $80*5/15=27$
- 问题: 如果由于 B, C 分别延误 80 天, 造成 100 万损失, 应该如何赔偿?



同时浮动赔偿

- B 赔偿: $100 \times (1 - 2/3)$
- C 赔偿: $100 \times (1 - 1/3)$

- 作为项目经理应该避免一些对项目不利的因素
 - 严禁不应该的浮动
 - 避免损失



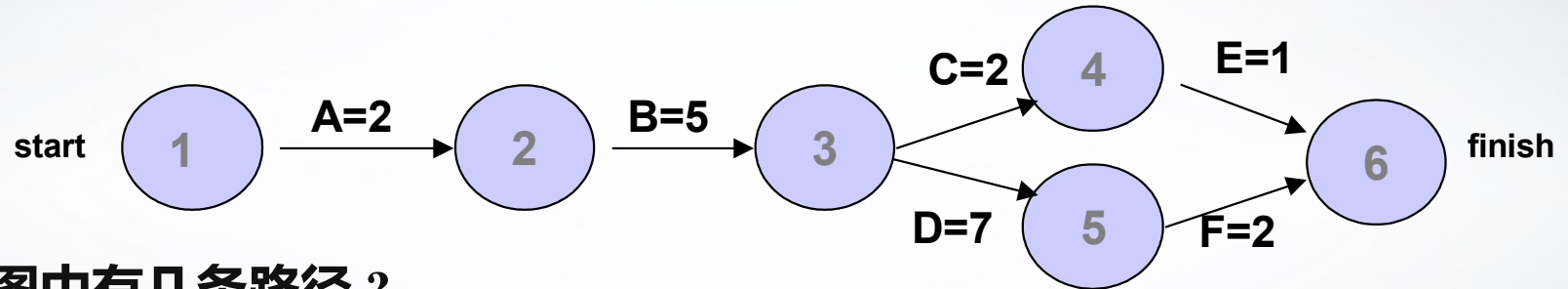
关键路径（Critical Path）

- ❑ 关键路径是决定项目完成的最短时间。
- ❑ 是时间浮动为 0（Float=0）的路径
- ❑ 网络图中最长的路径
- ❑ 关键路径上的任何任务都是关键任务
- ❑ 关键路径上的任何活动延迟，都会导致整个项目完成时间的延迟



关键路径举例

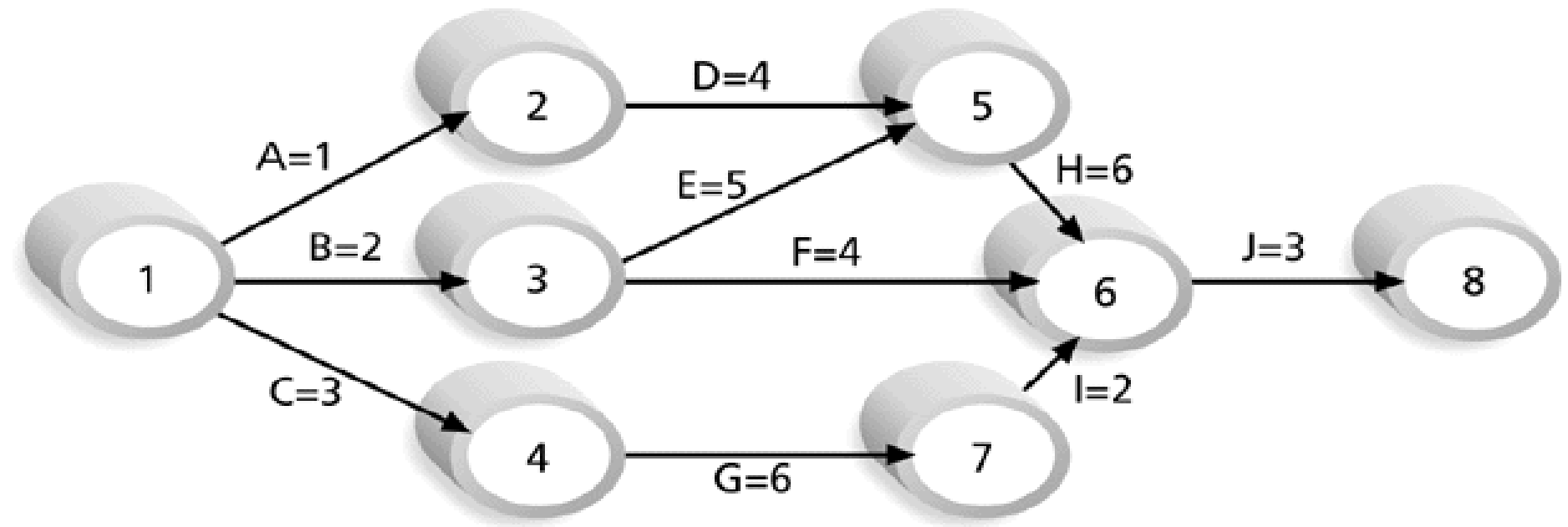
- 研究网络图，确定全程时间：



- 图中有几条路径？
- 每条路径几天？
- 哪条是关键路径？
- 整个项目最少需要多少天？



关键路径计算



Note: Assume all durations are in days.

- | | | |
|---------|---------|------------------------------|
| Path 1: | A-D-H-J | Length = $1+4+6+3 = 14$ days |
| Path 2: | B-E-H-J | Length = $2+5+6+3 = 16$ days |
| Path 3: | B-F-J | Length = $2+4+3 = 9$ days |
| Path 4: | C-G-I-J | Length = $3+6+2+3 = 14$ days |

Since the critical path is the longest path through the network diagram, Path 2, B-E-H-J, is the critical path for Project X.

关键路径的其他说明

- ❑ 明确关键路径后，你可以合理安排进度
- ❑ 关键路径可能不止一条
- ❑ 在项目的进行过程中，关键路径可能改变的



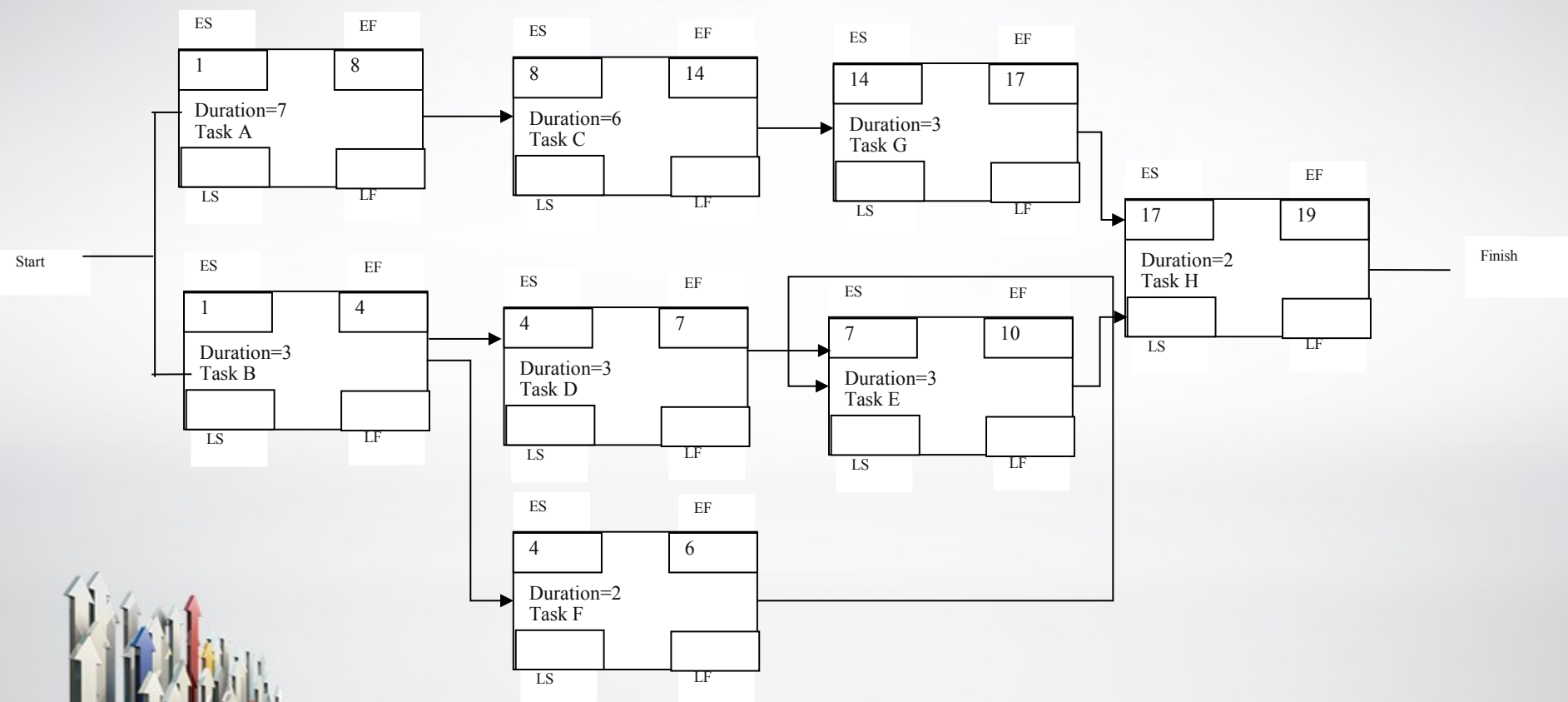
正推法 (Forward pass)

按照时间顺序计算最早开始时间和最早完成时间的方法，称为正推法。

- ❑ 首先建立项目的开始时间
- ❑ 项目的开始时间是网络图中第一个活动的最早开始时间
- ❑ 从左到右，从上到下进行任务编排
- ❑ 当一个任务有多个前置时，选择其中最大的最早完成日期作为其后置任务的最早开始日期
- ❑ 公式：
 - ❑ $ES + Duration = EF$
 - ❑ $EF + Lag = ES_s$



正推法实例



当一个任务有多个前置时，选择其中最大的最早完成日期作为其后置任务的最早开始日期



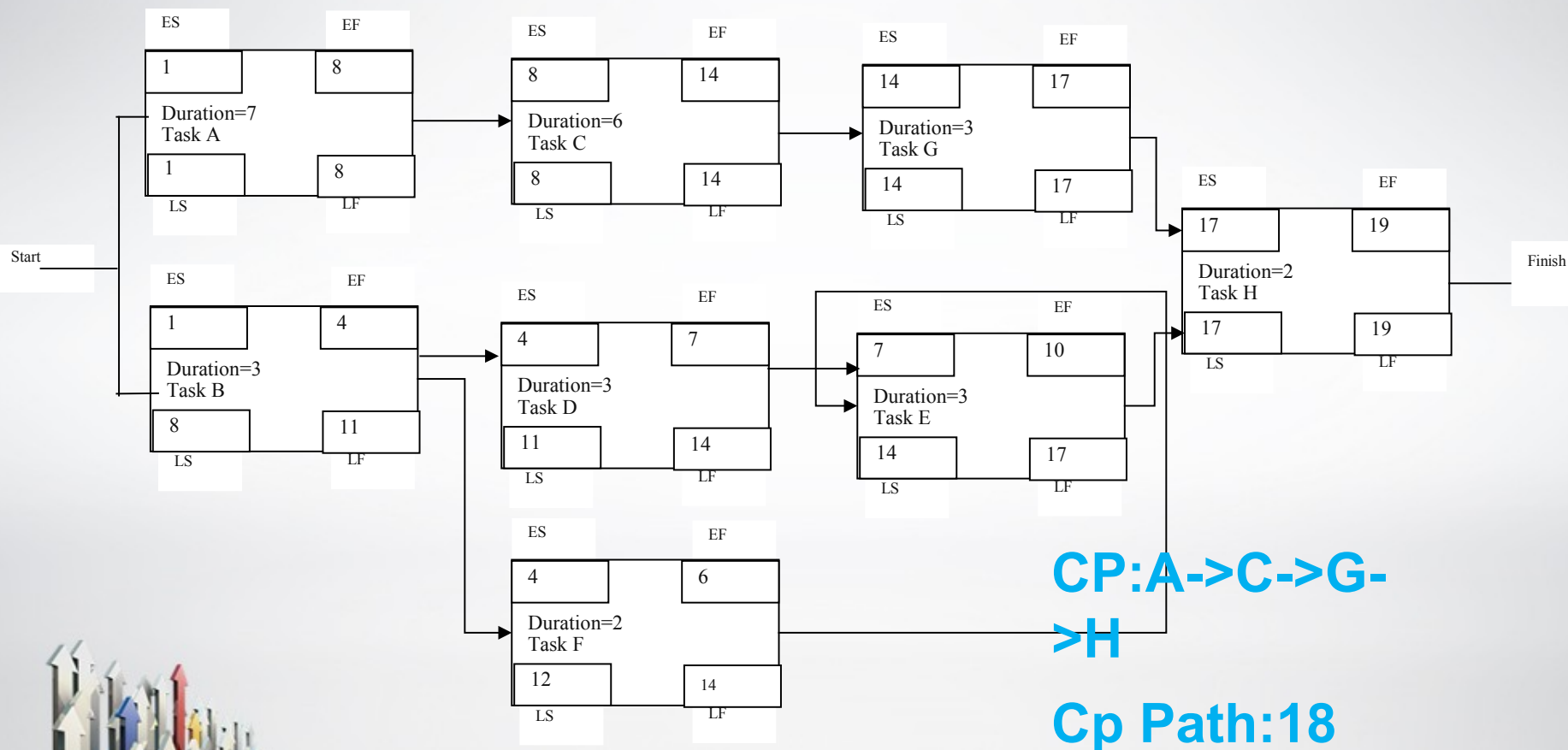
逆推法 (Backward pass)

按照逆时间顺序计算最晚开始时间和最晚结束时间的方法，称为逆推法。

- ❑ 首先建立项目的结束时间
- ❑ 项目的结束时间是网络图中最后一个活动的最晚结束时间
- ❑ 从右到左，从上到下进行计算
- ❑ 当一个前置任务有多个后置任务时，选择其中最小最晚开始日期作为其前置任务的最晚完成日期
- ❑ 公式：
 - ❑ $LF - Duration = LS$
 - ❑ $LS - Lag = LF_p$



逆推图示



当一个前置任务有多个后置任务时，选择其中最小最晚开始日期作为其前置任务的最晚完成日期

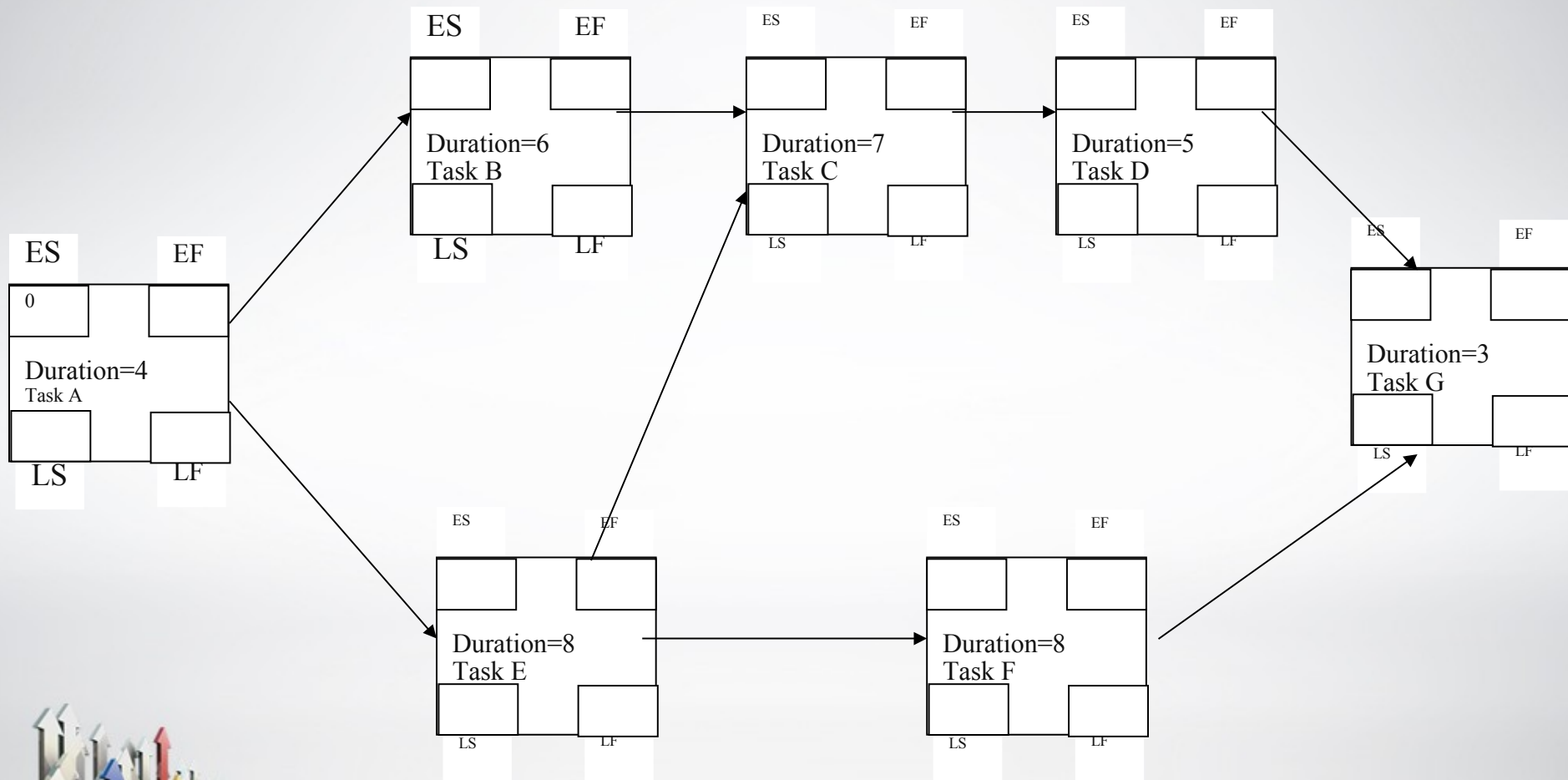
练习

- 作为项目经理，你需要给一个网站项目做计划安排，经过任务分解后得到任务 A，B，C，D，E，F，G，假设各个任务之间没有滞后和超前，下图是这个项目的 PDM 网络图。通过历时估计已经估算出每个任务的工期，现已标识在 PDM 网络图上。假设项目的最早开工日期是第 0 天，请计算每个任务的最早开始时间，最晚开始时间，最早完成时间，最晚完成时间，同时确定关键路径，并计算关键路径的长度，计算任务 F 的自由浮动和总浮动。



课堂练习

项目管理



1. 确定 C P 以及 C P 的长度？

2. F 的自由浮动和总浮动？



公式：ES(S)= EF(P) + Lag, LF(P) = LS (S) – Lag

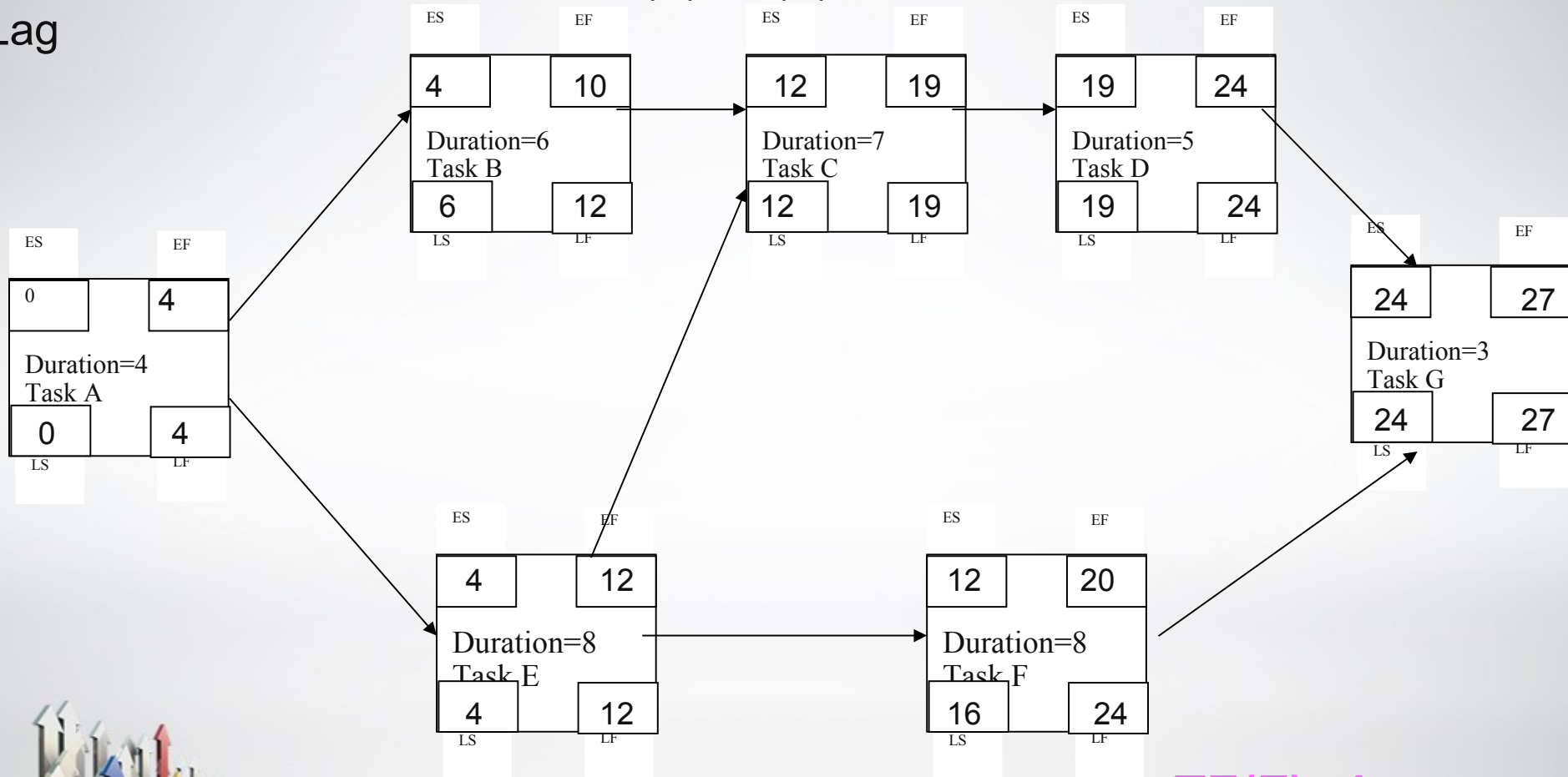
课堂练习 - 答案

项目管理

TF=LS-ES,

FF= ES(S)-EF(P)-

Lag



CP:A->E->C->D->G CP

TF(F)=4

FF(F)=4



进度编制的基本方法

- 关键路径法
 - 正推法
 - 逆推法
- 时间压缩法
 - 赶工（Crash）
 - 快速跟进（Fast tracking: 搭接）
- 关键链法



时间压缩法

时间压缩法是在不改变项目范围的前提下缩短项目工期的方法

- 应急法 — 赶工（Crash）
- 平行作业法 — 快速跟进（Fast tracking: 搭接）



应急法 – 赶工（Crash）

- 赶工也称为时间 – 成本平衡方法
- 在不改变活动的前提下，通过压缩某一个或者多个活动的时间来达到缩短整个项目工期的目的
- 在最小相关成本增加的条件下，压缩关键路径上的关键活动历时的方法



关于进度压缩的费用

- 进度压缩单位成本方法：
 - 线性关系：
- Charles Symons (1991) 方法
 - 进度压缩比普通进度短的时候，费用迅速上涨



进度压缩单位成本方法

前提：活动的正常与压缩

□ 项目活动的正常值

- 正常历时
- 正常成本

□ 项目活动的压缩值

- 压缩历时
- 压缩成本



进度压缩单位成本方法

- 进度压缩单位成本 = (压缩成本 - 正常成本) / (正常进度 - 压缩进度)
- 例如：
 - 任务 A: 正常进度 7 周, 成本 5 万; 压缩到 5 周的成本是 6.2 万
 - 进度压缩单位成本 = $(6.2 - 5) / (7 - 5) = 6000$ 元 / 周
 - 如果压缩到 6 周的成本是: 5.6 万



时间压缩例题

- 下图给出了各个任务可以压缩的最大限度和压缩成本，请问如果将工期压缩到 17，16，15 周时应该压缩的活动和最后的成本？



计算单位压缩成本

任务 单位压缩成本	A	B	C	D
压缩成本 (万 / 周)	0.6	1	0.5	0.6



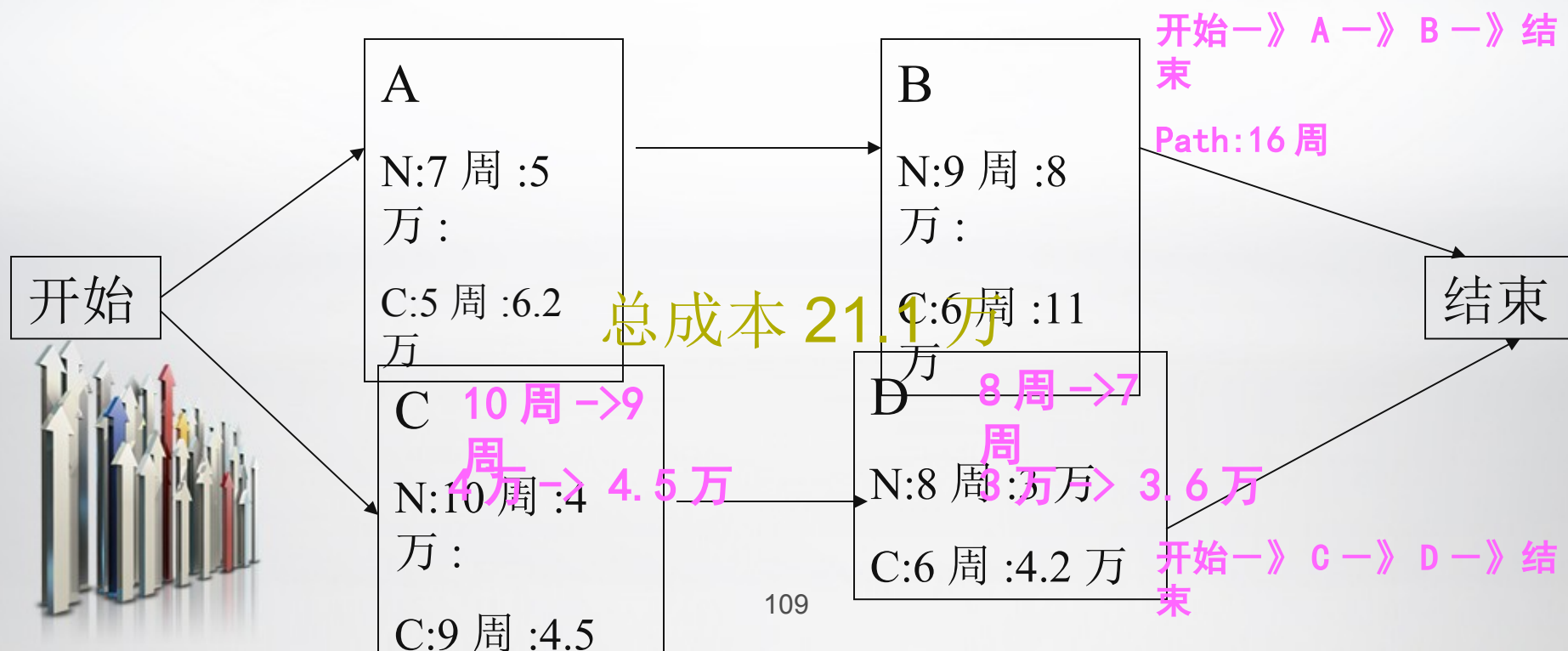
时间压缩例题

- 将工期压缩到 17 时应该压缩的活动和最后的成本？



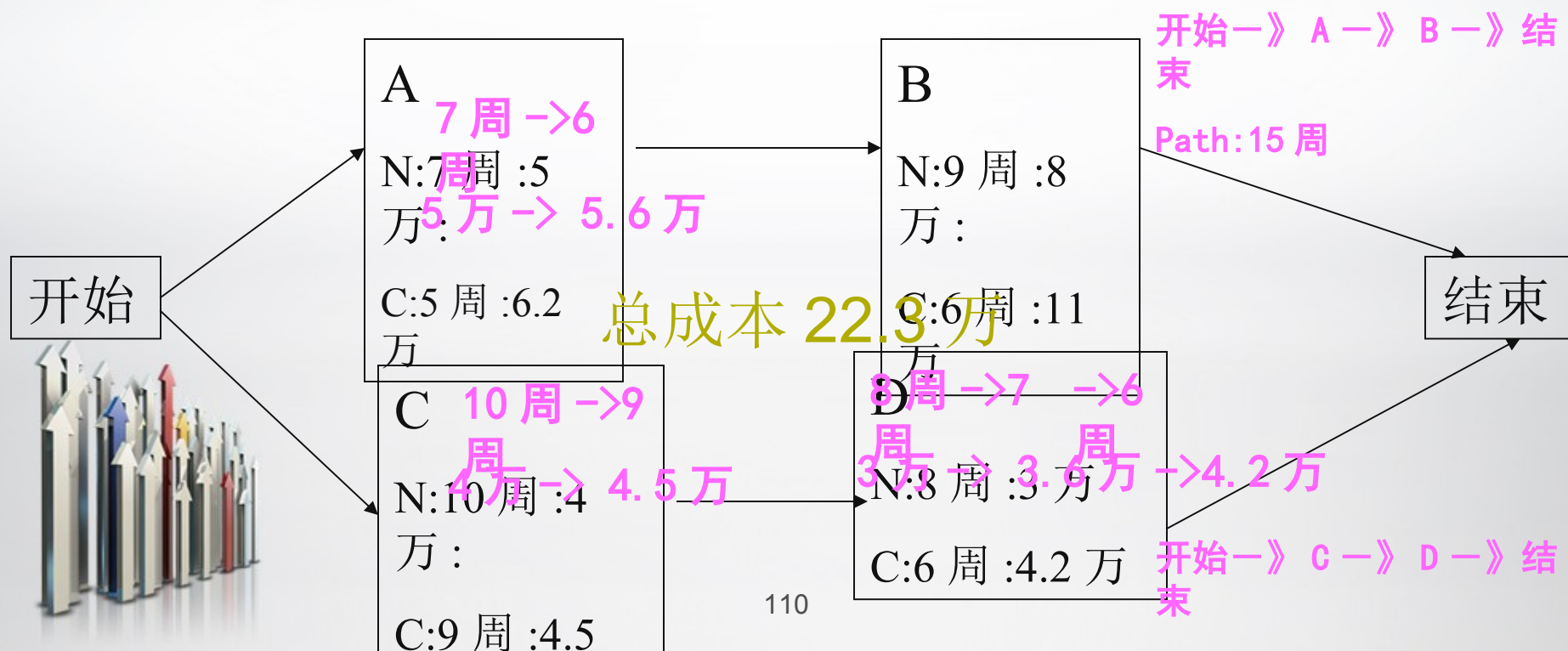
时间压缩例题

- 将工期压缩到 16 时应该压缩的活动和最后的成本？



时间压缩例题

- 将工期压缩到 15 时应该压缩的活动和最后的成本？



时间压缩答案

	可以压缩的任务	压缩的任务	成本计算 (单位:万)	项目成本 (单位:万)
18			$5+8+4+3$	20
17	C,D	C	$20+0.5$	20.5
16	C,D	D	$20.5+0.6$	21.1
15	A,B,C,D	A,D	$21.1+0.6+0.6$	22.3



赶工时间与赶工成本关系图

追加成本

压缩时间

压缩角度，越小越好



关于进度的一些说明

- 项目存在一个可能的最短进度

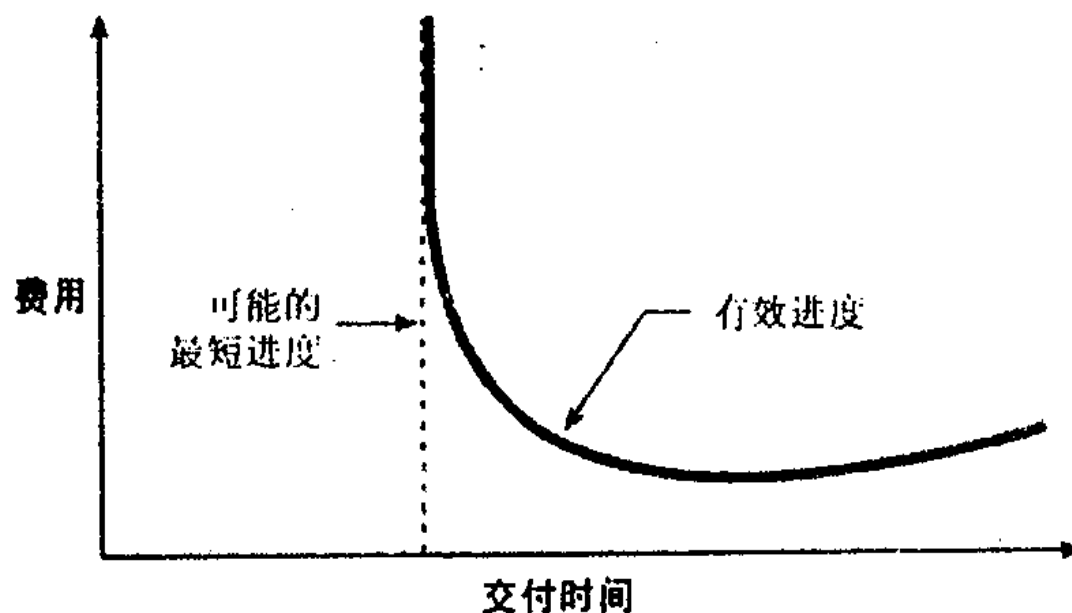


图 2-2-5 软件项目费用与进度之间的关系（完成有效进度的费用大大低于完成最短可能进度的费用）

Charles Symons (1991) 方法

- 进度压缩因子 = 压缩进度 / 正常进度
- 压缩进度的工作量 = 正常工作量 / 进度压缩因子
- 例如：
 - 初始进度估算是 12 月，初始工作量估算是 78 人月，
 - 如果进度压缩到 10 月，进度压缩因子 = $10/12=0.83$ ，
 - 则进度压缩后的工作量是： $78/0.83=94$ 人月
 - 总结：进度缩短 17%，增加 21% 的工作量
- 研究表明：进度压缩因子 > 0.75 ，最多可以压缩 25 %



平行作业法 - 快速跟进 (Fast tracking: 搭接)

- 是在改变活动间的逻辑关系，并行开展某些活动



进度时间参数

任务

项目管理 :100

需求 :10 设计 :5

设计 :5

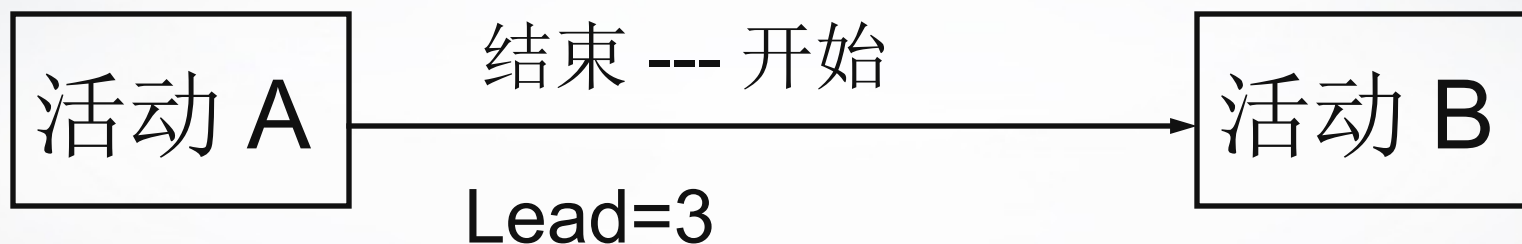


时间



任务超前 (Lead)

A 完成之前 3 天 B 开始



作用：

- 1) 解决任务的搭接
- 2) 对任务可以进行合理的拆分
- 3) 缩短项目工期



任务拆分

任务

项目管理 :100

需求 :10 设计 :5

设计 :3

设计 :2

时间



进度编制的基本方法

- 关键路径法
 - 正推法
 - 逆推法
- 时间压缩法
 - 赶工（Crash）
 - 快速跟进（Fast tracking: 搭接）
- 关键链法



关键链法的预备知识

项目管理

- 管理预留
- 约束理论



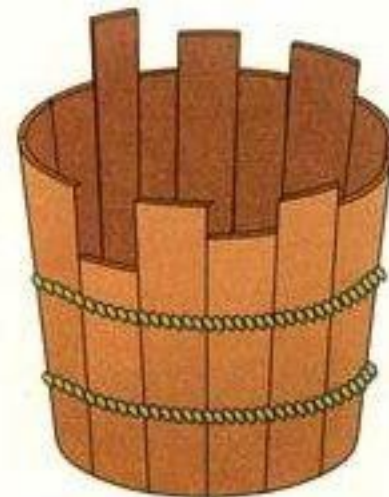
- 管理预留是一项加在项目末端的人为任务
- Parkinson 法则声明，“扩展工作是为了填补时间以便能够完成。”

缓冲



约束理论

1. 所有现实系统都存在约束。
2. 约束的存在表明系统存在改进的机会。



“木桶理论”

木桶理论是由美国管理学家彼得提出的。组成木桶的木板如果长短不齐，那么木桶的盛水量不是取决于最长的那一块木板，而是取决于最短的那一块木板。构成组织的各个部分往往是优劣不齐的，而劣势部分往往决定整个组织的水平。



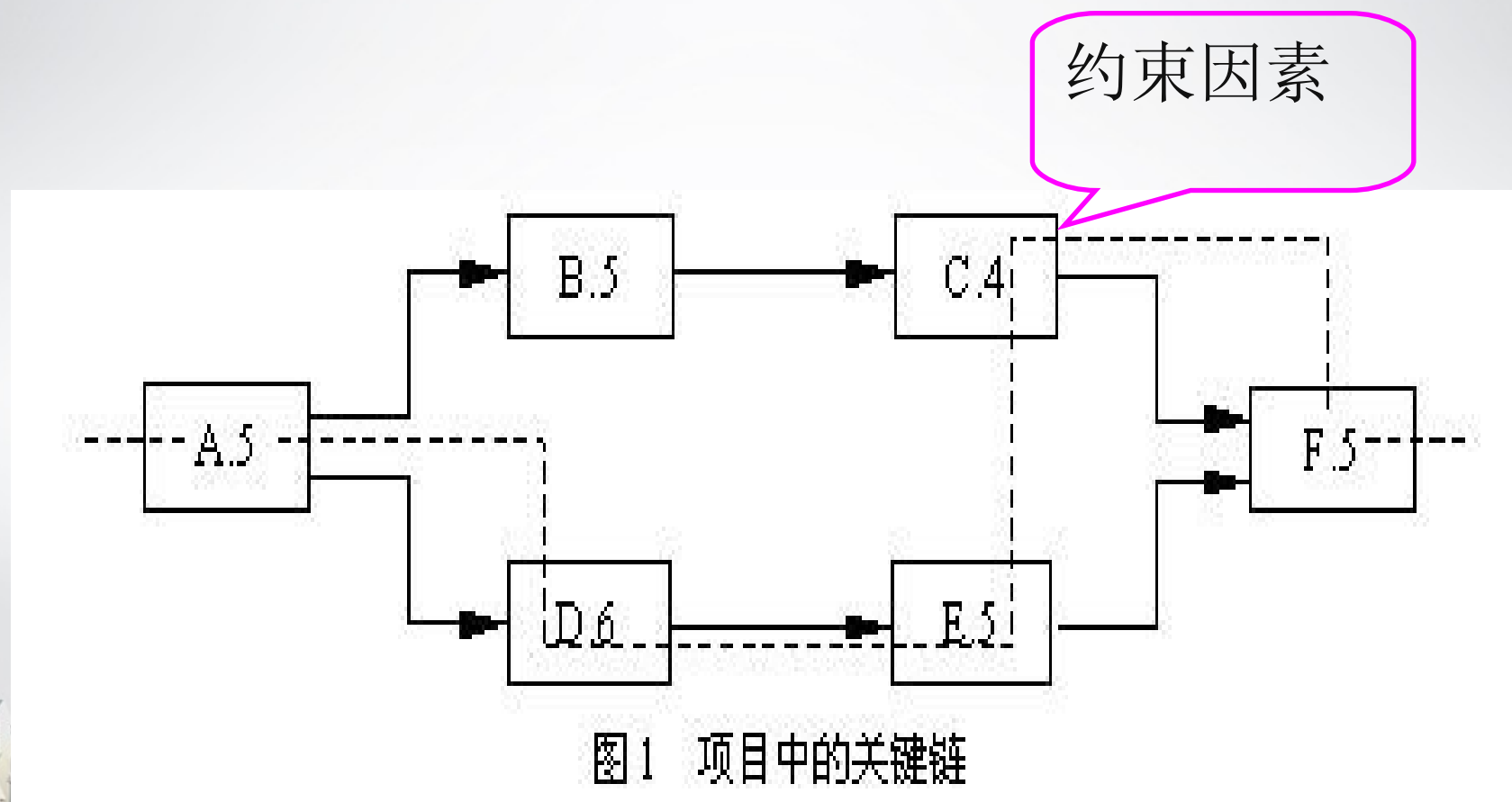
约束理论五大关键步骤

1. 找出系统中的约束因素；
2. 决定如何挖掘约束因素的潜力；
3. 使系统中所有其他工作服从于第二步的决策；
4. 提升约束因素的能力；
5. 若该约束已经转化为非约束性因素，则回到第一步，否则回到第二步，要注意不要让思维惯性成为新的主要约束因素。



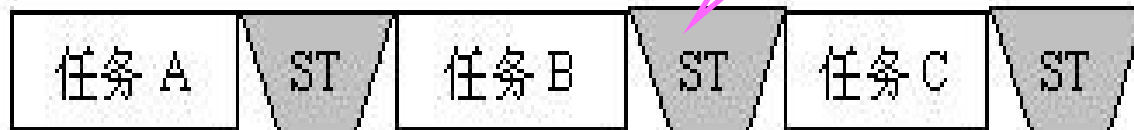
关键链法（自学，了解）

项目管理



安全时间与缓冲时间

关键路线法:



关键链方法:

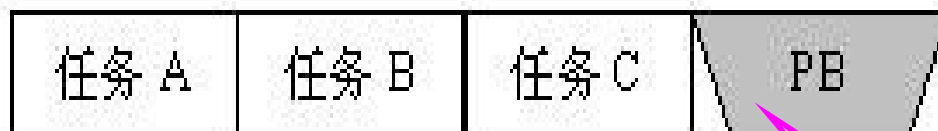


图2 关键路线法和关键链方法在风险管理上的差异



非关键链缓冲时间

非关键链:



Feeding Buffer

关键链:



图 3 非关键链的缓冲时间

