软件项目进度计划

进度的定义

- 进度是对执行的活动和里程碑制定的工作计划日期表

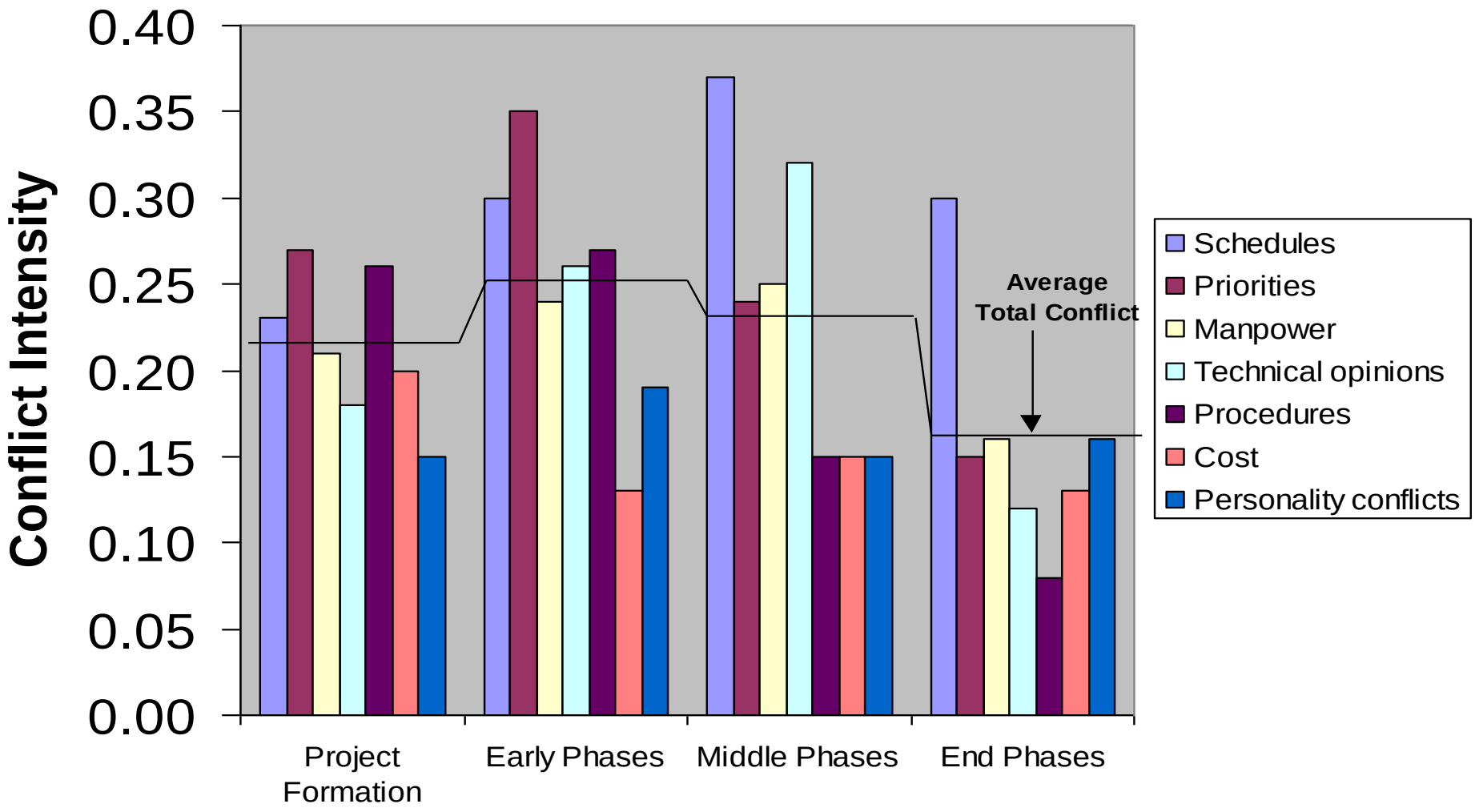
进度管理定义

- 进度管理是为了确保项目按期完成所需要的过程.

进度管理的重要性

- 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。

进度管理的重要性


Conflict Intensity
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Project Formation
Early Phases
Middle Phases
End Phases
Average Total Conflict
Schedules
Priorities
Manpower
Technical opinions
Procedures
Cost
Personality conflicts

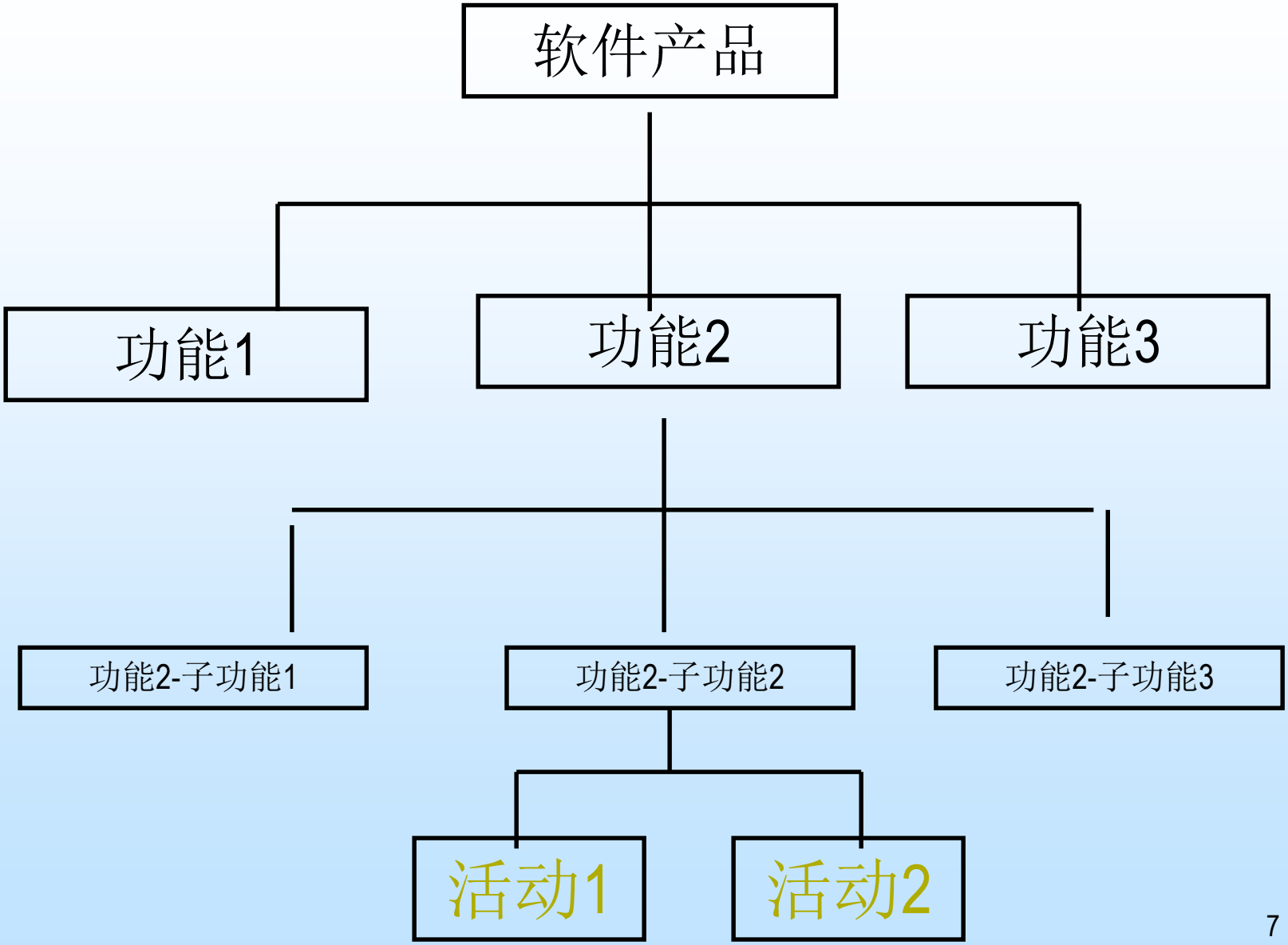
软件项目进度(时间)管理过程

- 活动定义（Activity definition）
- 活动排序（Activity sequencing）
- 活动历时估计（Activity duration estimating）
- 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

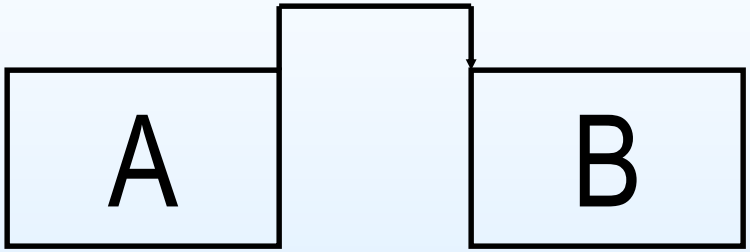


项目活动排序

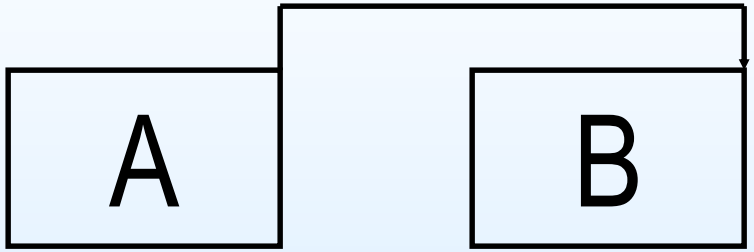
- 项目各项活动之间存在相互联系与相互依赖关系，
- 根据这些关系进行适当的顺序安排

前置活动（任务）---〉后置活动（任务）

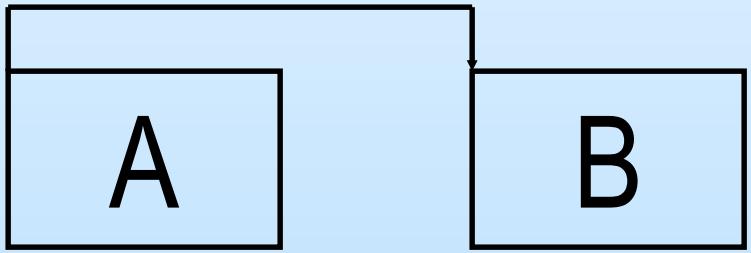
任务(活动)之间的关系



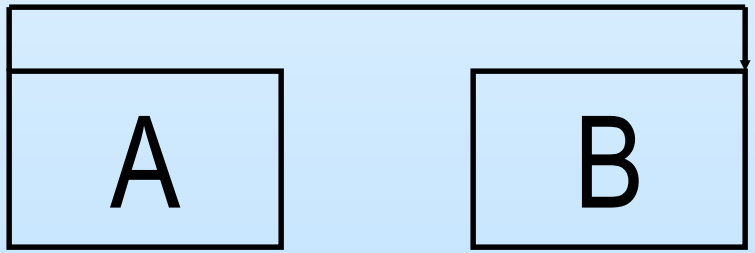
结束-开始



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

You *must* determine dependencies in order to use critical path analysis

进度管理图示

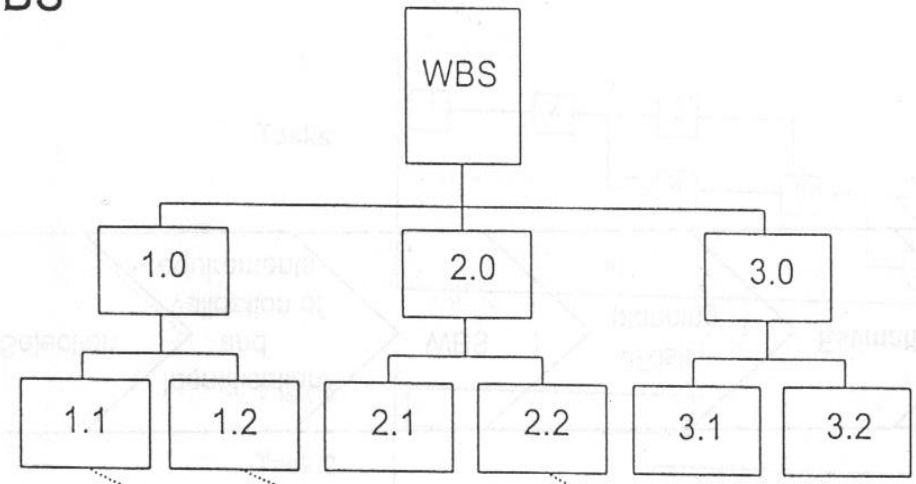
- 网络图
- 甘特图
- 里程碑图
- 资源图

网络图

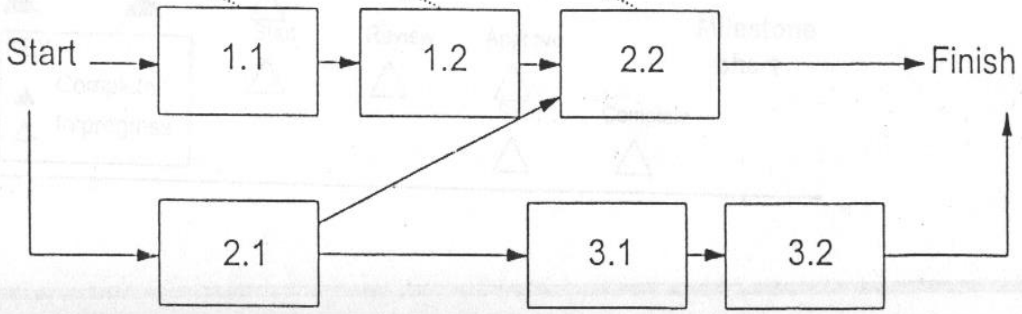
- 网络图是活动排序的一个输出
- 展示项目中的各个活动以及活动之间的逻辑关系
- 网络图可以表达活动的历时

Network Relationship to WBS

WBS



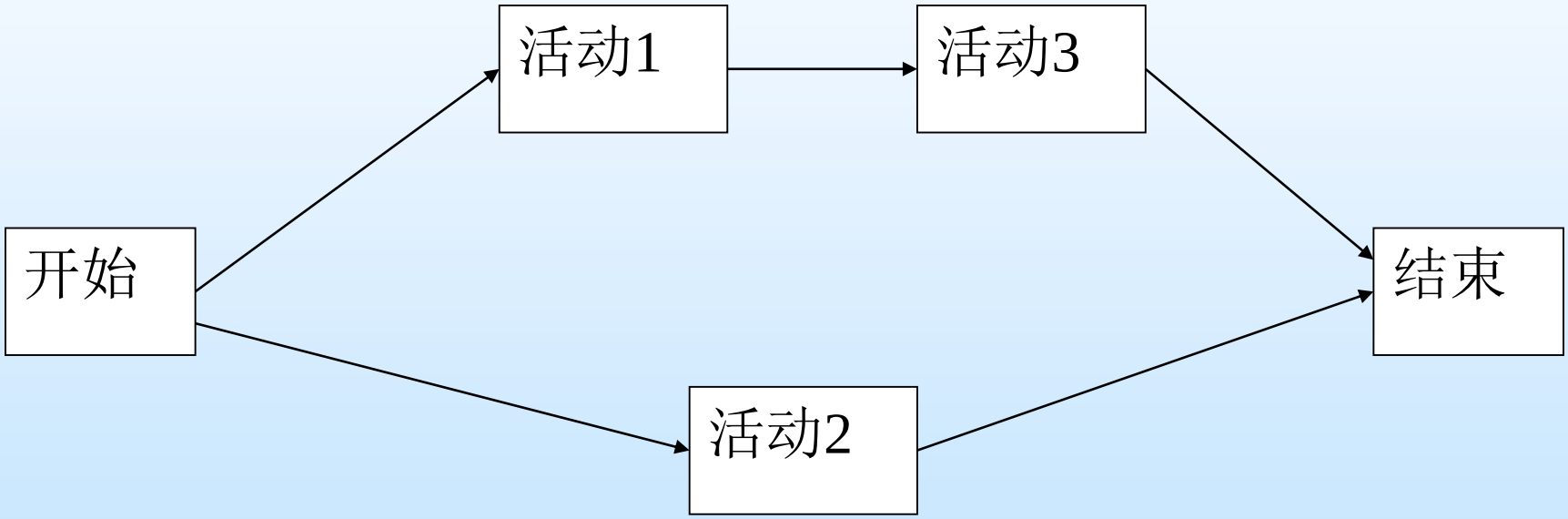
Network



常用的网络图

- PDM （Precedence Diagramming Method ）
  - 优先图法，节点法（单代号)网络图
- ADM （Arrow Diagramming Method ）
  - 箭线法（双代号)网络图

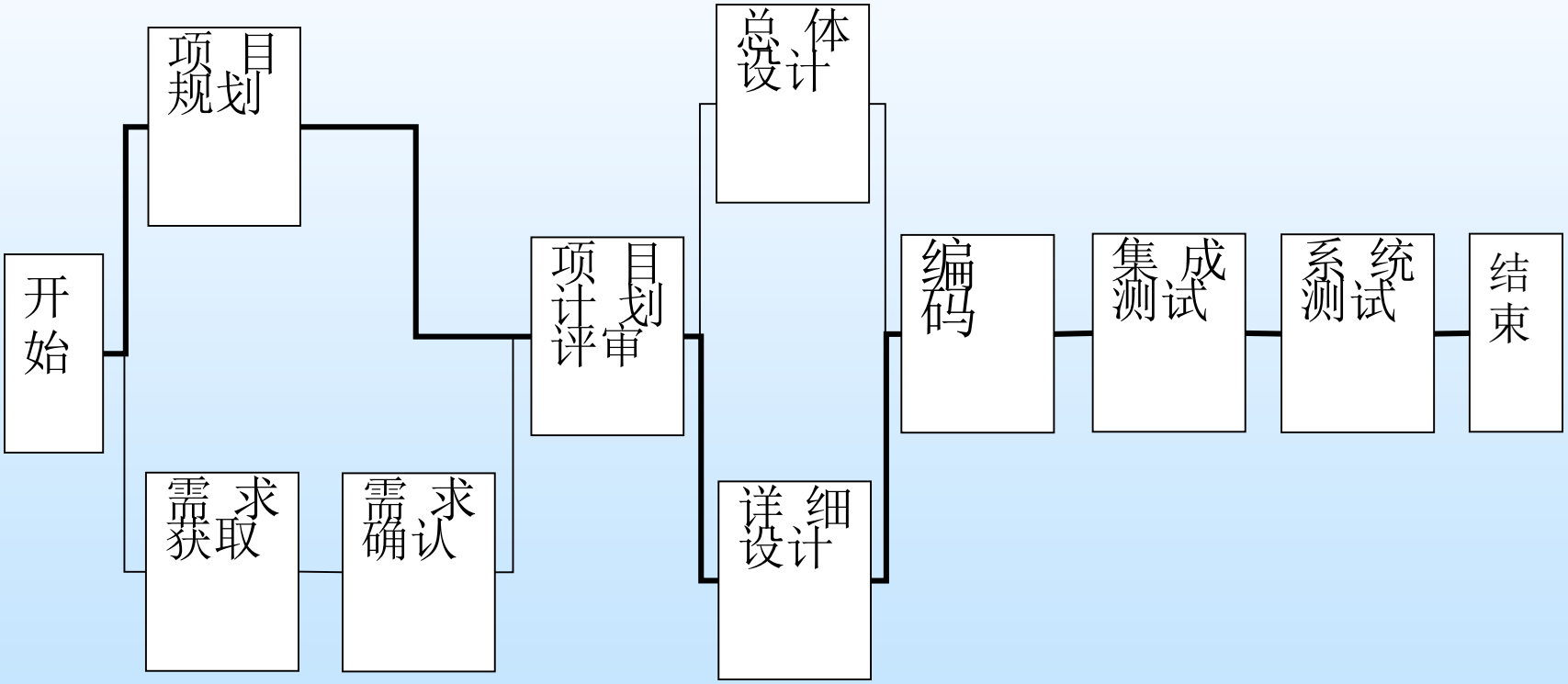
PDM图例



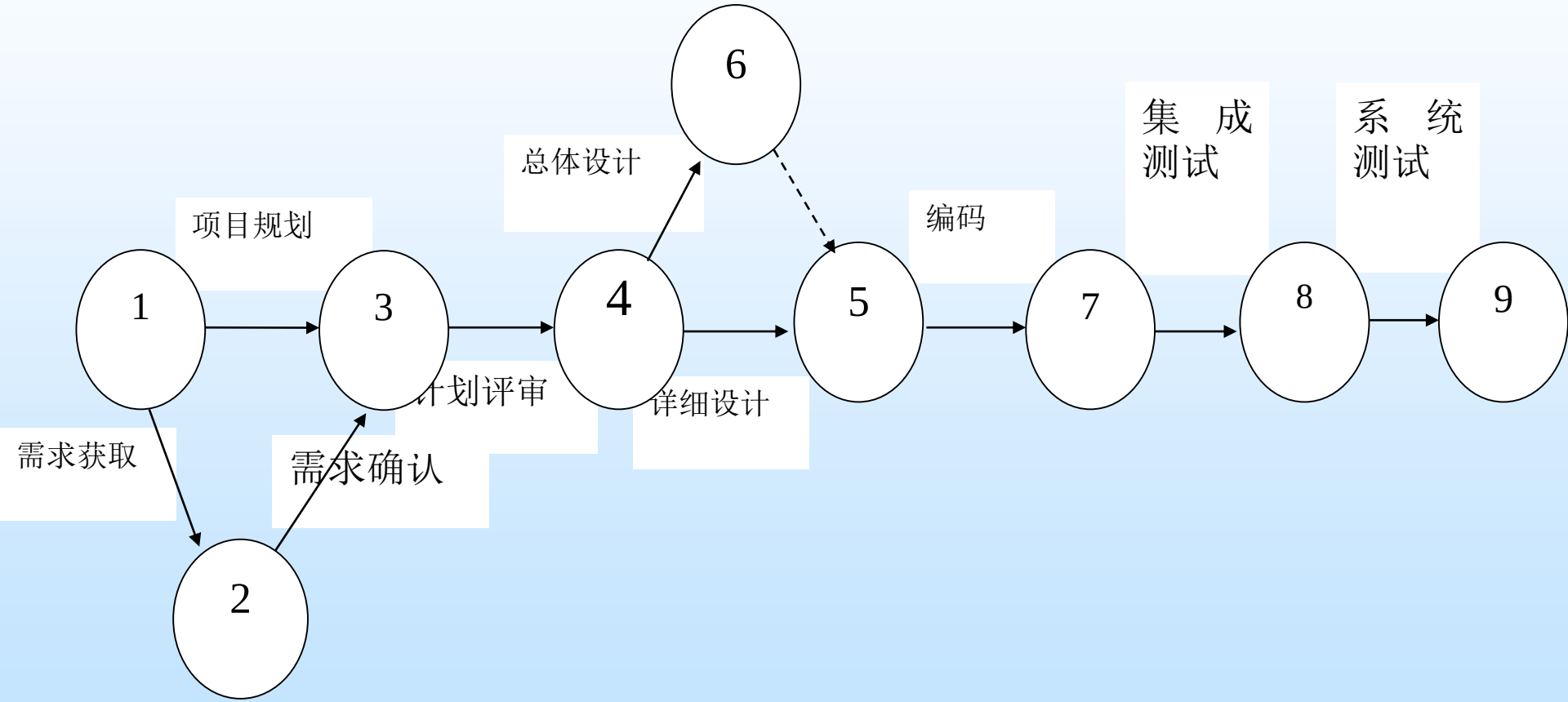
PDM(Precedence Diagramming Method)

- 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（Precedence Diagramming Method）-优先图法图例



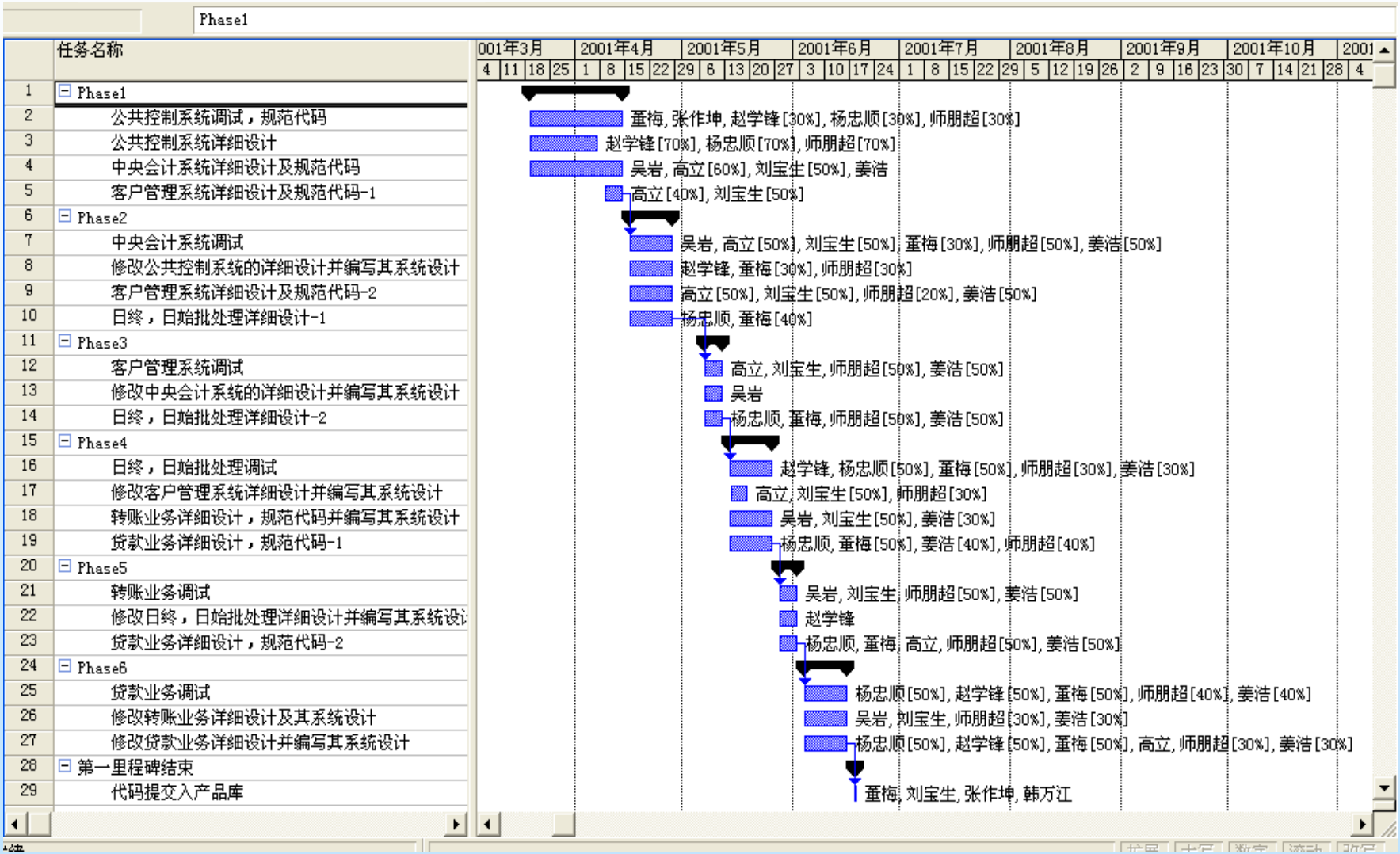
ADM图例



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

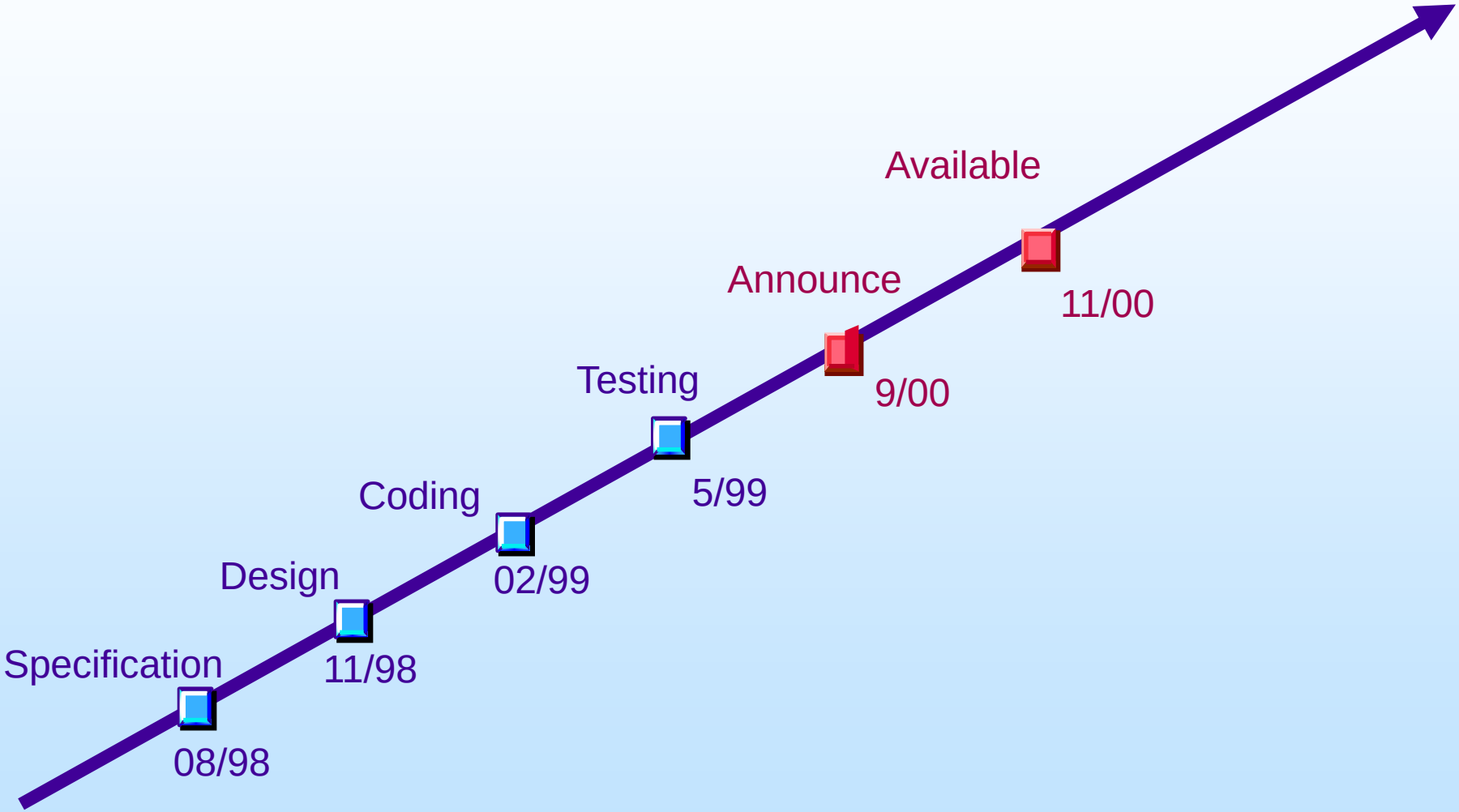
甘特图-实例


Phase1
任务名称
001年3月
2001年4月
2001年5月
2001年6月
2001年7月
2001年8月
2001年9月
2001年10月
1 Phase1
2 公共控制系统调试，规范代码
3 公共控制系统详细设计
4 中央会计系统详细设计及规范代码
5 客户管理系统详细设计及规范代码-1
6 Phase2
7 中央会计系统调试
8 修改公共控制系统的详细设计并编写其系统设计
9 客户管理系统详细设计及规范代码-2
10 日终，日始批处理详细设计-1
11 Phase3
12 客户管理系统调试
13 修改中央会计系统的详细设计并编写其系统设计
14 日终，日始批处理详细设计-2
15 Phase4
16 日终，日始批处理调试
17 修改客户管理系统详细设计并编写其系统设计
18 转账业务详细设计，规范代码并编写其系统设计
19 贷款业务详细设计，规范代码-1
20 Phase5
21 转账业务调试
22 修改日终，日始批处理详细设计并编写其系统设
23 贷款业务详细设计，规范代码-2
24 Phase6
25 贷款业务调试
26 修改转账业务详细设计及其系统设计
27 修改贷款业务详细设计并编写其系统设计
28 第一里程碑结束
29 代码提交入产品库
董梅,张作坤,赵学锋[30%],杨忠顺[30%],师朋超[30%]
赵学锋[70%],杨忠顺[70%],师朋超[70%]
吴岩,高立[60%],刘宝生[50%],姜浩
高立[40%],刘宝生[50%]
吴岩,高立[50%],刘宝生[50%],董梅[30%],师朋超[50%],姜浩[50%]
赵学锋,董梅[30%],师朋超[30%]
高立[50%],刘宝生[50%],师朋超[20%],姜浩[50%]
杨忠顺,董梅[40%]
高立,刘宝生,师朋超[50%],姜浩[50%]
吴岩
杨忠顺,董梅,师朋超[50%],姜浩[50%]
赵学锋,杨忠顺[50%],董梅[50%],师朋超[30%],姜浩[30%]
高立,刘宝生[50%],师朋超[30%]
吴岩,刘宝生[50%],姜浩[30%]
杨忠顺,董梅[50%],姜浩[40%],师朋超[40%]
吴岩,刘宝生,师朋超[50%],姜浩[50%]
赵学锋
杨忠顺,董梅,高立,师朋超[50%],姜浩[50%]
杨忠顺[50%],赵学锋[50%],董梅[50%],师朋超[40%],姜浩[40%]
吴岩,刘宝生,师朋超[30%],姜浩[30%]
杨忠顺[50%],赵学锋[50%],董梅[50%],高立,师朋超[30%],姜浩[30%]
董梅,刘宝生,张作坤,韩万江

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

里程碑图示


Specification
08/98
Design
11/98
Coding
02/99
Testing
5/99
Announce
9/00
Available
11/00

里程碑图示

里程碑图

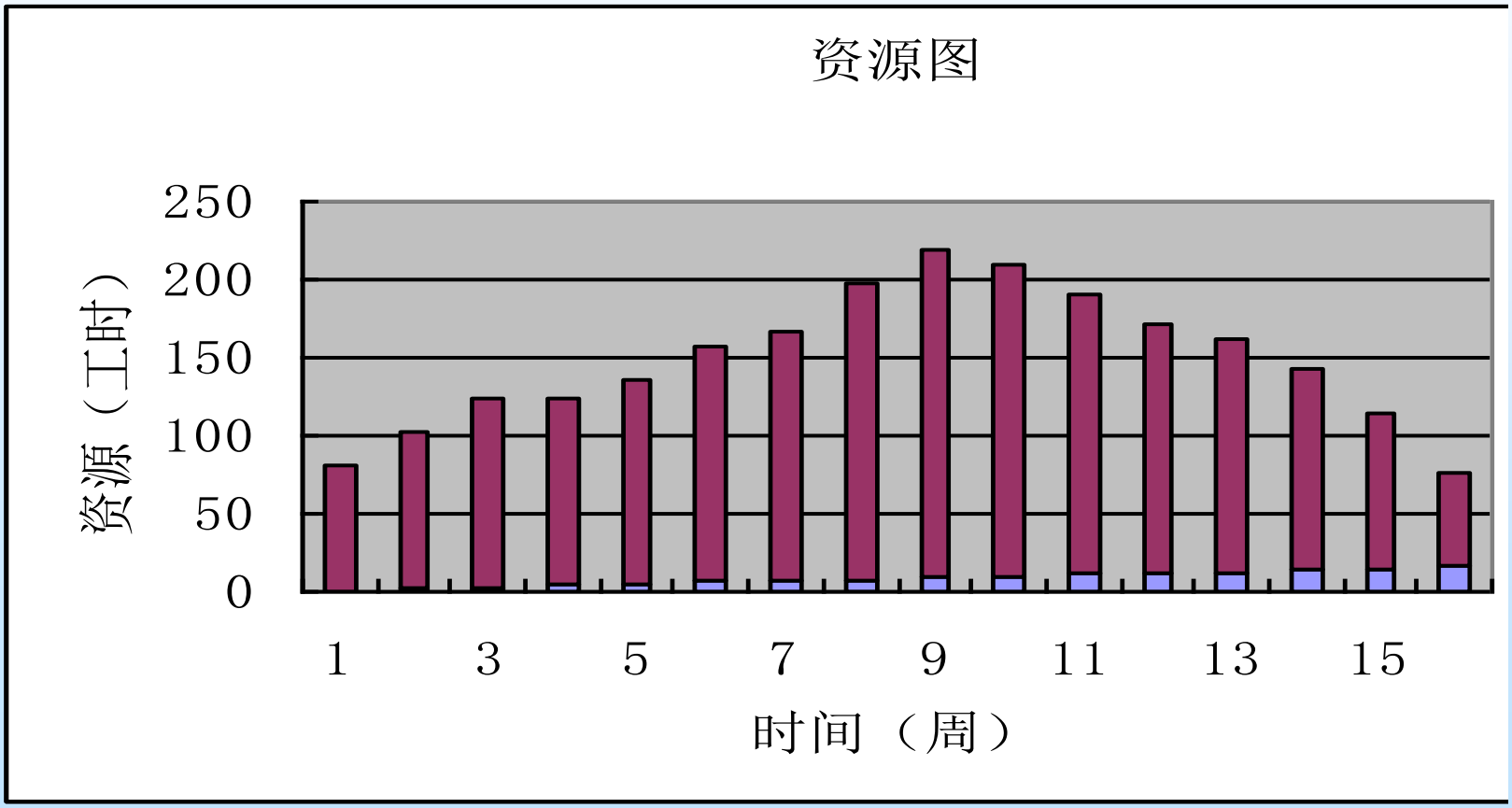
图6-7

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

资源图



本章要点

一、进度管理的基本概念及过程

二、进度估算的基本方法

三、编制进度计划

四、案例分析

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

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
- Jones的一阶估算准则
- 其它策略

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—4
  - b:1/3左右:依赖于项目的自然属性

建议掌握模型

- Walston-Felix(IBM)： D=2.4*E exp(0.35)
- 基本COCOMO: D=2.5(E)exp(d_b)， d_b：0.32-0.38

方式	d_b
有机	0.38
半有机	0.35
嵌入式	0.32

举例

- 采用基本COCOMO模型估算的规模E＝152ＰM
- 采用基本COCOMO模型估算的进度
- D=2.5＊E＾0.35
  =2.5*152＾0.35＝14.5 M

经验导出其它模型举例

- 如果：E=65人月，并且a=3，b=1/3
- 则：D= 3 * 65 exp(1/3)=12月

项目进度估算的基本方法

- 基于规模的进度估算
- CPM
- PERT
- 基于进度表的进度估算
- 基于承诺的进度估计
- Jones的一阶估算准则
- 其它策略

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

项目进度估算的基本方法

- 基于规模的进度估算，
- CPM
- PERT
- 基于进度表的进度估算
- 基于承诺的进度估计
- Jones的一阶估算准则
- 其它策略

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。

工程评价技术（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
- 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 Formula and Example

Example:

PERT weighted average =

$$\frac{8 \text{ workdays} + 4 \times 10 \text{ workdays} + 24 \text{ workdays}}{6}$$

= 12 days

where 8 = optimistic time, 10 = most likely time, and 24 = pessimistic time

PERT的评估进度风险

- 标准差δ =(最大估算值-最小估算值)/6
- 方差δ^2 =[(最大估算值-最小估算值)/6]^2

- 例如上图： δ =(24－8) /6=2.67

PERT/CPM区别

- PERT
  - 计算历时采用的算法:加权平均（O+4m+P)/6
  - 估计值不明确
- CPM
  - 计算历时采用的算法:最大可能值m
  - 估计值比较明确

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
- CPM
- PERT
- 基于进度表的进度估算
- 基于承诺的进度估计
- Jones的一阶估算准则
- 其它策略

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，
- 有几年应用编程语言和编程环境的工作经验，
- 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的工作
- 采用矩形员工模式

可能的最短进度表–工具支持

- 有先进的软件开发工具
- 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

可能的最短进度表

表 2－2－12　　可能的最短进度

系统规模（代码行）	系统软件 进度（月）	系统软件 工作量（人月）	商业软件 进度（月）	商业软件 工作量（人月）	封装商品产品 进度（月）	封装商品产品 工作量（人月）
10 000	6	25	3.5	5	4.2	8
15 000	7	40	4.1	8	4.9	13
20 000	8	57	4.6	11	5.6	19
25 000	9	74	5.1	15	6	24
30 000	9	110	5.5	22	7	37
35 000	10	130	5.8	26	7	44
40 000	11	170	6	34	7	57
45 000	11	195	6	39	8	66
50 000	11	230	7	46	8	79
60 000	12	285	7	57	9	98
70 000	13	350	8	71	9	120
80 000	14	410	8	83	10	140
90 000	14	480	9	96	10	170
100 000	15	540	9	110	11	190
120 000	16	680	10	140	11	240

可能的最短进度表

续表

系统规模（代码行）	系统软件 进度（月）	系统软件 工作量（人月）	商业软件 进度（月）	商业软件 工作量（人月）	封装商品产品 进度（月）	封装商品产品 工作量（人月）
140 000	17	820	10	160	12	280
160 000	18	960	10	190	13	335
180 000	19	1 100	11	220	13	390
200 000	20	1 250	11	250	14	440
250 000	22	1 650	13	330	15	580
300 000	24	2 100	14	420	16	725
400 000	27	2 900	15	590	19	1 000
500 000	30	3 900	17	780	20	1 400

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年应用编程语言和编程环境的工作经验，
- 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

- 有效的编程工具
- 主动的风险管理
- 优良的物理环境
- 沟通工具方便

有效进度表

表 2－2－13　　有效进度

系统规模（代码行）	系统产品 进度（月）	系统产品 工作量（人月）	商业产品 进度（月）	商业产品 工作量（人月）	封装产品 进度（月）	封装产品 工作量（人月）
10 000	8	24	4.9	5	5.9	8
15 000	10	38	5.8	8	7	12
20 000	11	54	7	11	8	18
25 000	12	70	7	14	9	23
30 000	13	97	8	20	9	32
35 000	14	120	8	24	10	39
40 000	15	140	9	30	10	49
45 000	16	170	9	34	11	57
50 000	16	190	10	40	11	67
60 000	18	240	10	49	12	83
70 000	19	290	11	61	13	100
80 000	20	345	12	71	14	120
90 000	21	400	12	82	15	140
100 000	22	450	13	93	15	160
120 000	23	560	14	115	16	195
140 000	25	670	15	140	17	235

有效进度表

续表

系统规模（代码行）	系统产品		商业产品		封装产品	
	进度（月）	工作量（人月）	进度（月）	工作量（人月）	进度（月）	工作量（人月）
160 000	26	709	15	160	18	280
180 000	28	910	16	190	19	320
200 000	29	1 300	17	210	20	360
250 000	32	1 300	19	280	22	470
300 000	34	1 650	20	345	24	590
400 000	38	2 350	22	490	27	830
500 000	42	3 100	25	640	29	1 100

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
- 与编程语言和编程环境一般熟悉
- 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

- 编程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

普通进度表

表 2－2－14　　普通进度

系统规模（代码行）	系统产品		商业产品		封装产品	
	进度（月）	工作量（人月）	进度（月）	工作量（人月）	进度（月）	工作量（人月）
10 000	10	48	6	9	7	15
15 000	12	76	7	15	8	24
20 000	14	110	8	21	9	34
25 000	15	140	9	27	10	44
30 000	16	185	9	37	11	59
35 000	17	220	10	44	12	71
40 000	18	270	10	54	13	88
45 000	19	310	11	61	13	100
50 000	20	360	11	71	14	115
60 000	21	440	12	88	15	145
70 000	23	540	13	105	16	175
80 000	24	630	14	125	17	210
90 000	25	730	15	140	17	240
100 000	26	820	15	160	18	270
120 000	28	1 000	16	200	20	335
140 000	30	1 200	17	240	21	400
160 000	32	1 400	18	280	22	470
180 000	34	1 600	19	330	23	540
200 000	35	1 900	20	370	24	610
250 000	38	2 400	22	480	26	800
300 000	41	3 000	24	600	29	1000
400 000	47	4 200	27	840	32	1 400
500 000	51	5 500	29	1 100	35	1 800

三种进度比较

- 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
- Jones的一阶估算准则
- 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
- 要求开发人员做出进度承诺，非进度估算

基于承诺的进度估计–优点

- 有利于开发者对进度的关注
- 有利于开发者在接受承诺之后的士气高昂

基于承诺的进度估计–缺点

- 开发人员估计的比较的乐观
- 易于产生大的估算误差

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
- Jones的一阶估算准则
- 其它策略

Jones的一阶估算准则

- 取得功能点的总和
- 从幂次表中选择合适的幂次将它升幂

Jones的一阶估算准则–幂次表

软件类型	最优级	平均	最差级
系统软件	0.43	0.45	0.48
商业软件	0.41	0.43	0.46
封装商品软件	0.39	0.42	0.45

Jones的一阶估算准则实例

- 如果
  - FP=350
  - 平均水平的商业软件公司
- 则
  - 粗略的进度= 350exp(0.43)=12月

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
- Jones的一阶估算准则
- 其它策略

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

估算不确定—表示

- 见下例子：把握性因素估算例子

交付日期	按期或者提前交付的概率
4月5日	5%
5月5日	50%
6月5日	90%

本章要点

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

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
- 监控项目实施的基础，它是项目管理的基准
- 计划是三维的，考虑时间，费用和资源

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

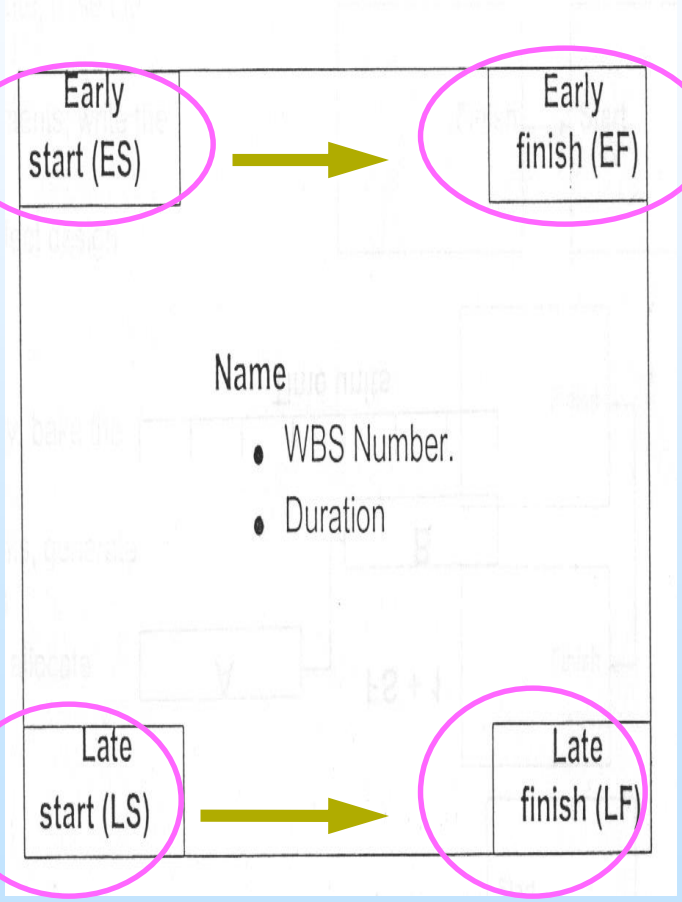
- 关键路径法
  - 正推法
  - 逆推法
- 时间压缩法
  - 赶工（Crash）
  - 快速跟进（Fast tracking:搭接）

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

- 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)



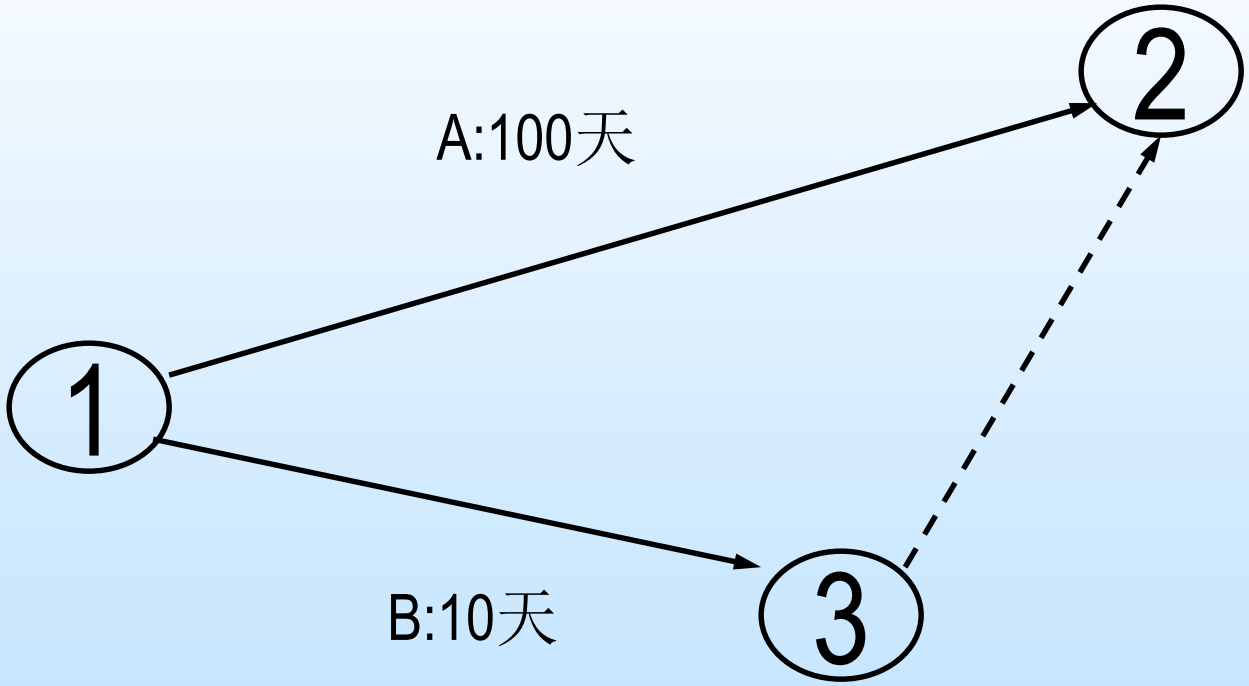
浮动时间(Float)

- 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

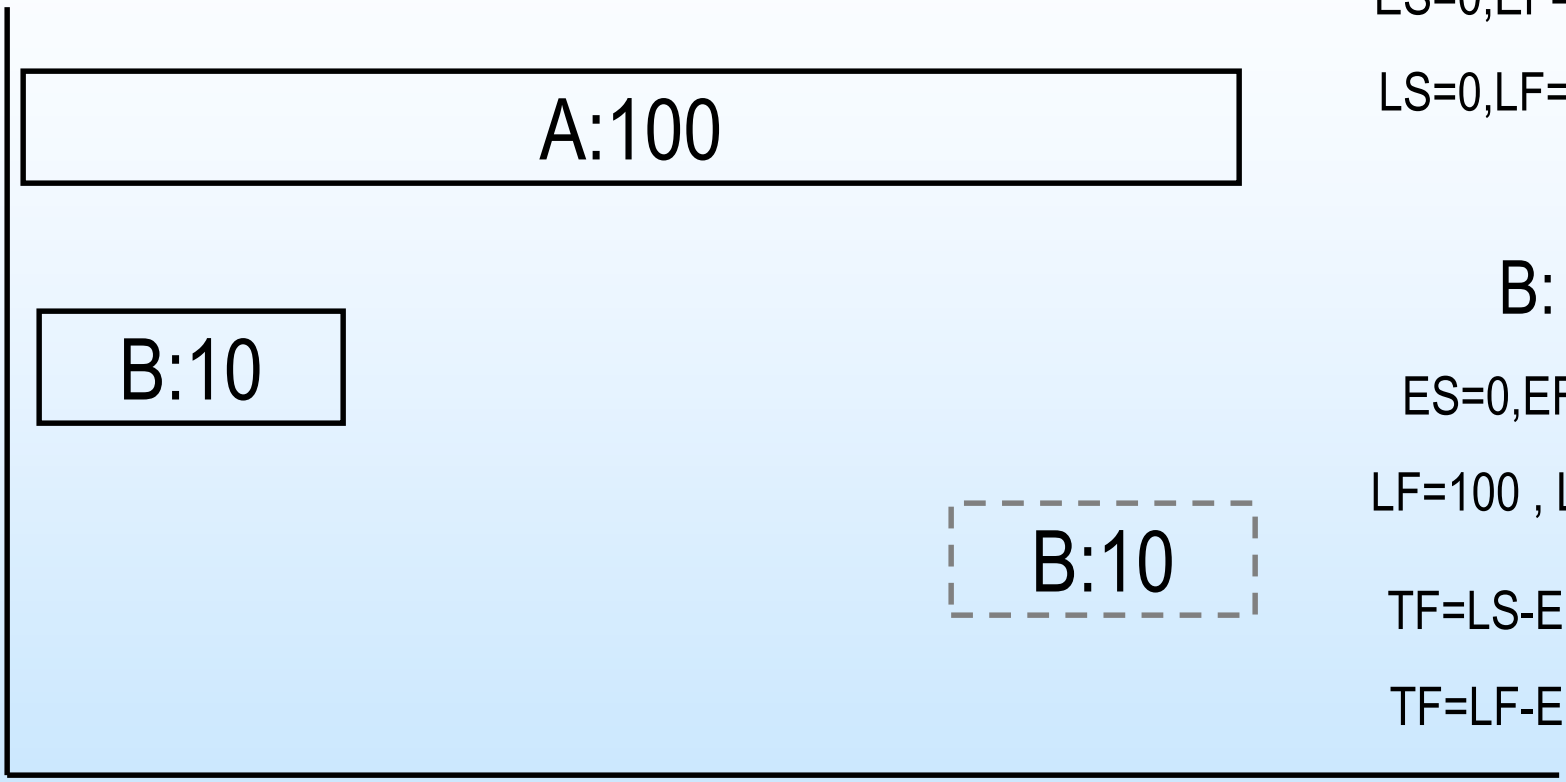
自由与总浮动时间

- 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
- 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

CPM估计



进度时间参数



A:

ES=0,EF=100

LS=0,LF=100

B:

ES=0,EF=10

LF=100 , LS=90

TF=LS-ES=90

TF=LF-EF=90

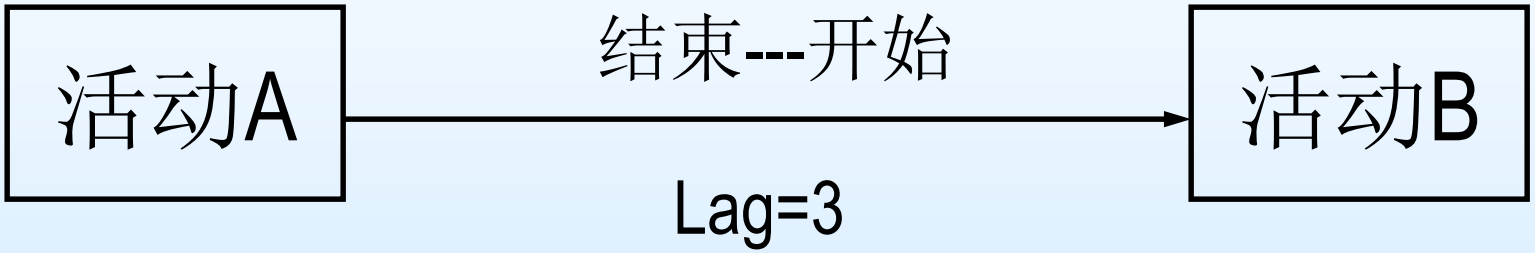
公式: EF= ES+duration

LS=LF- duration

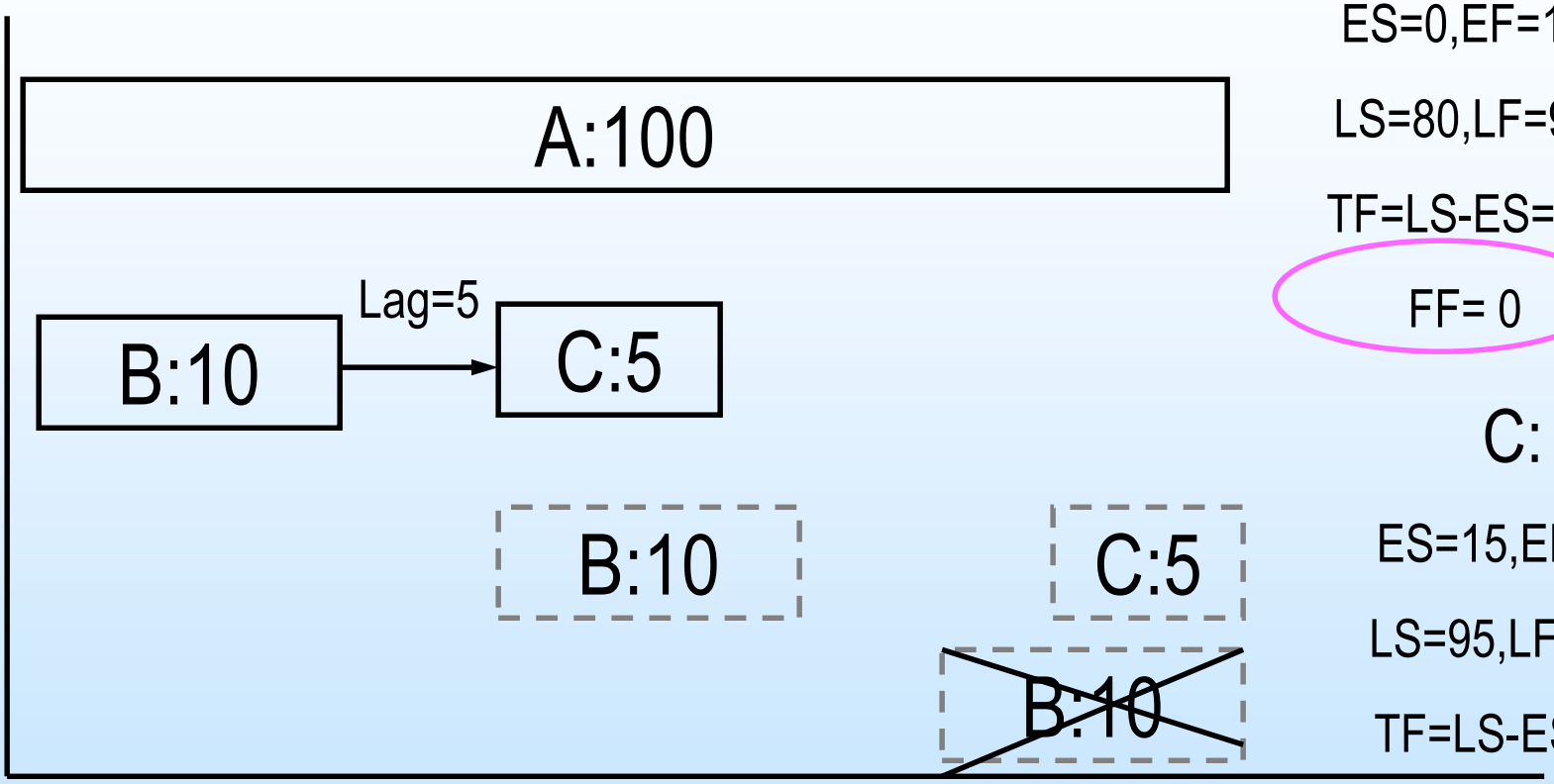
TF=LS-ES =LF-EF

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=100

TF=LS-ES=80

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

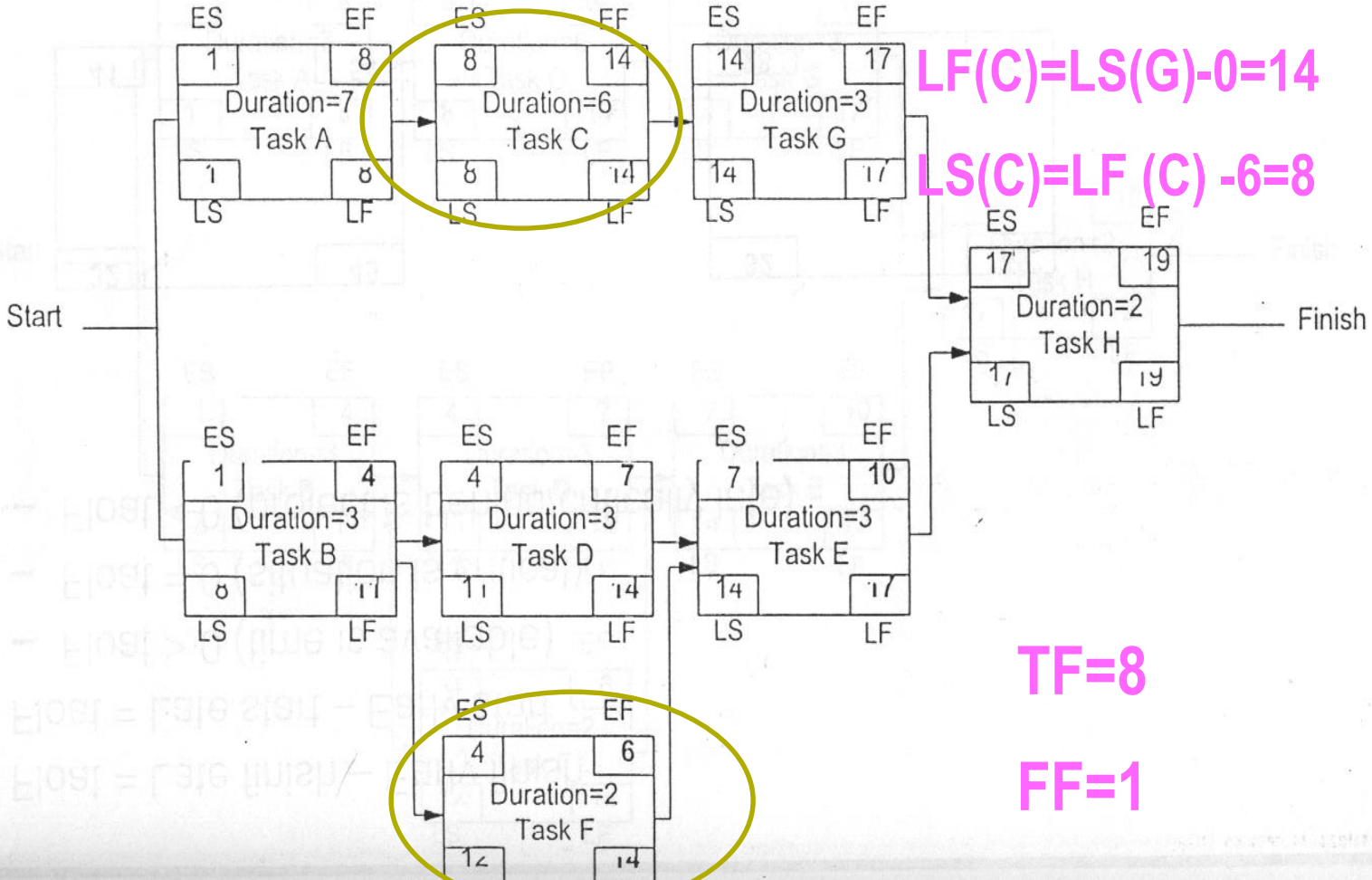
TF=LS-ES,FF= ES(S)-EF(P)- Lag

Float Example

- Free float = *ES* (successor) – *EF* (predecessor)
- Total float = *LS* – *ES* or *LF* – *EF*

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

ES(G)=EF(C)+0=14



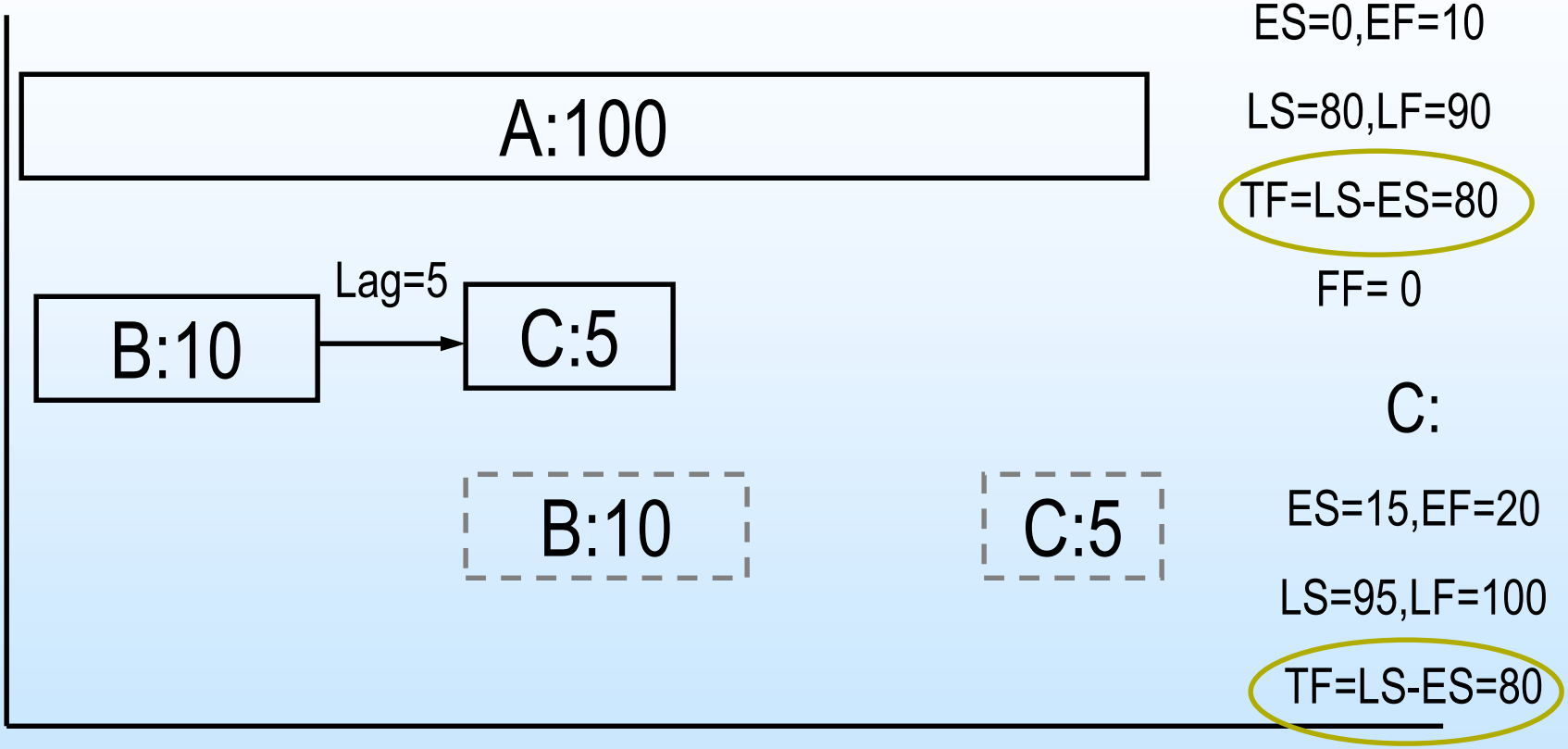
LF(C)=LS(G)-0=14

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

TF=8

FF=1

同时浮动？


A:100
B:10
Lag=5
C:5
B:10
C:5
B:
ES=0,EF=10
LS=80,LF=90
TF=LS-ES=80
FF= 0
C:
ES=15,EF=20
LS=95,LF=100
TF=LS-ES=80

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×（1-2/3）
- C赔偿：　100×（1-1/3）

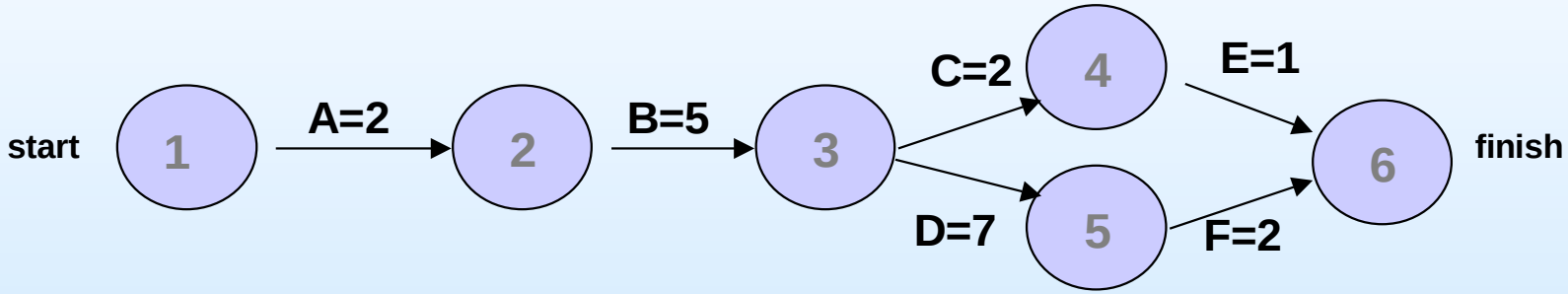
- 作为项目经理应该避免一些对项目不利的因素
  - 严禁不应该的浮动
  - 避免损失

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

Simple Example of Determining the Critical Path

- Consider the following project network diagram. Assume all times are in days.



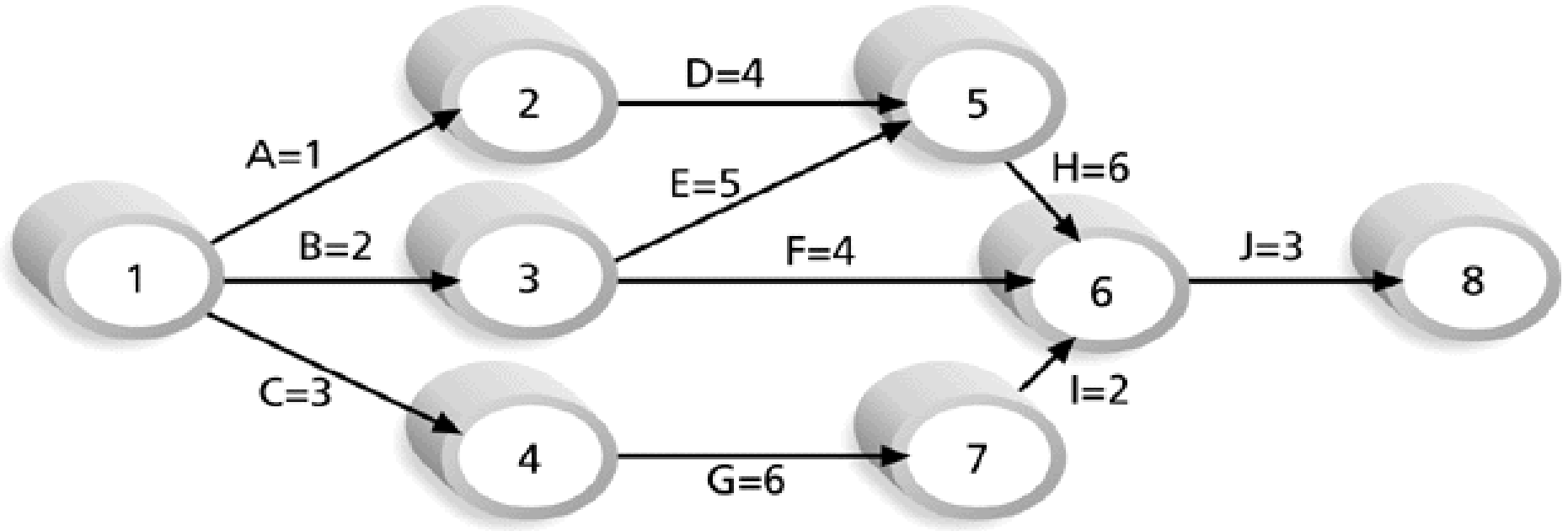
a. How many paths are on this network diagram?

b. How long is each path?

c. Which is the critical path?

d. What is the shortest amount of time needed to complete this project?

Determining the Critical Path for Project X



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

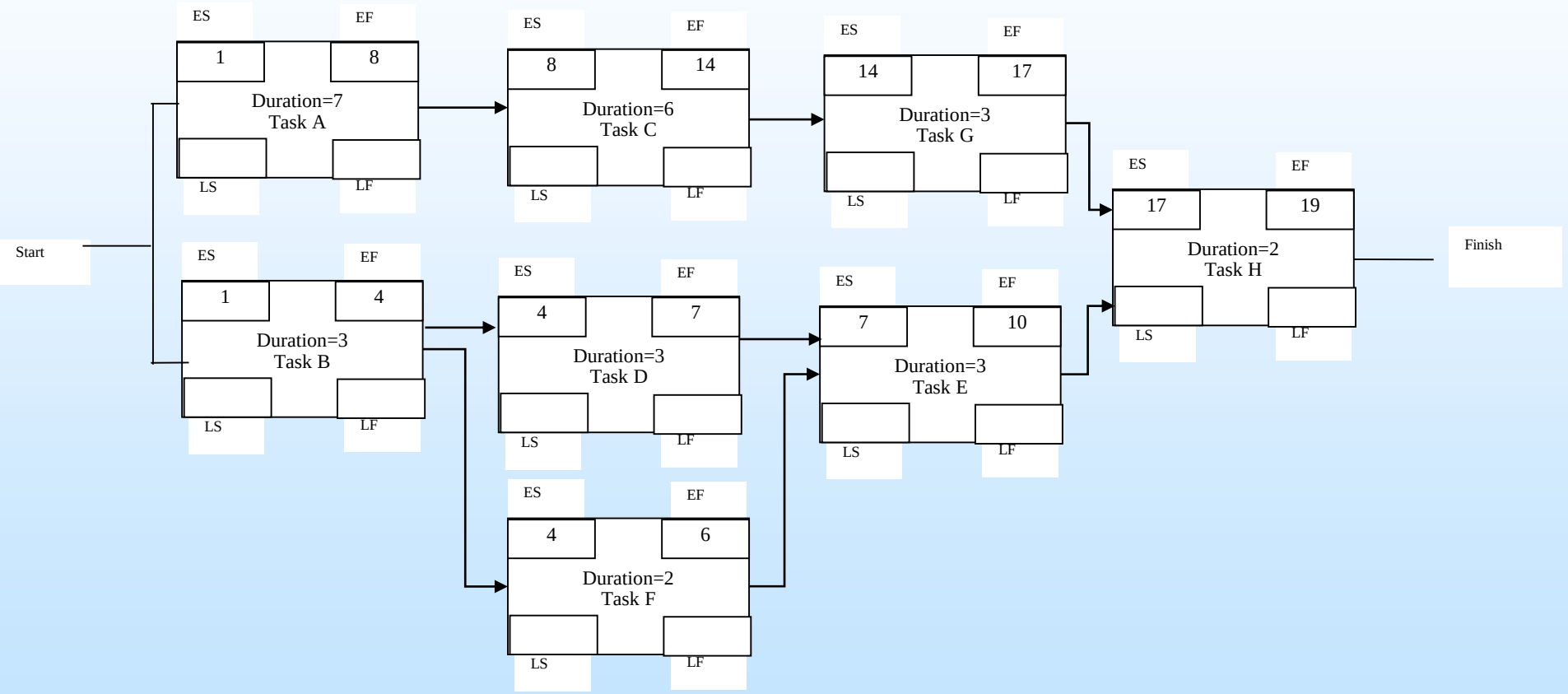
- 明确关键路径后，你可以合理安排进度
- 关键路径可能不止一条
- 在项目的进行过程中，关键路径可能改变的

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=ESs

正推法实例



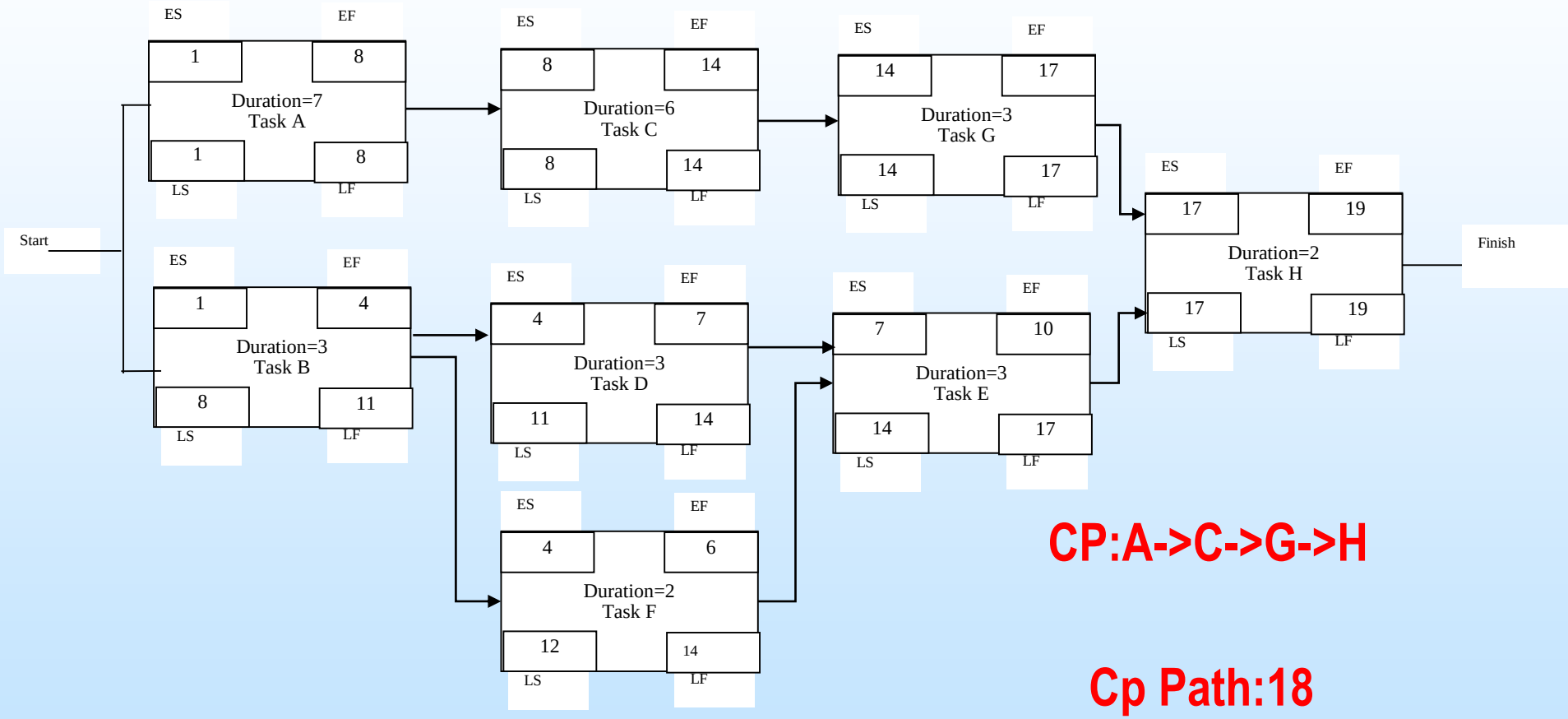
当一个任务有多个前置时，选择其中最大的最早完成日期作为其后置任务的最早开始日期

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=LFp

逆推图示



CP:A->C->G->H

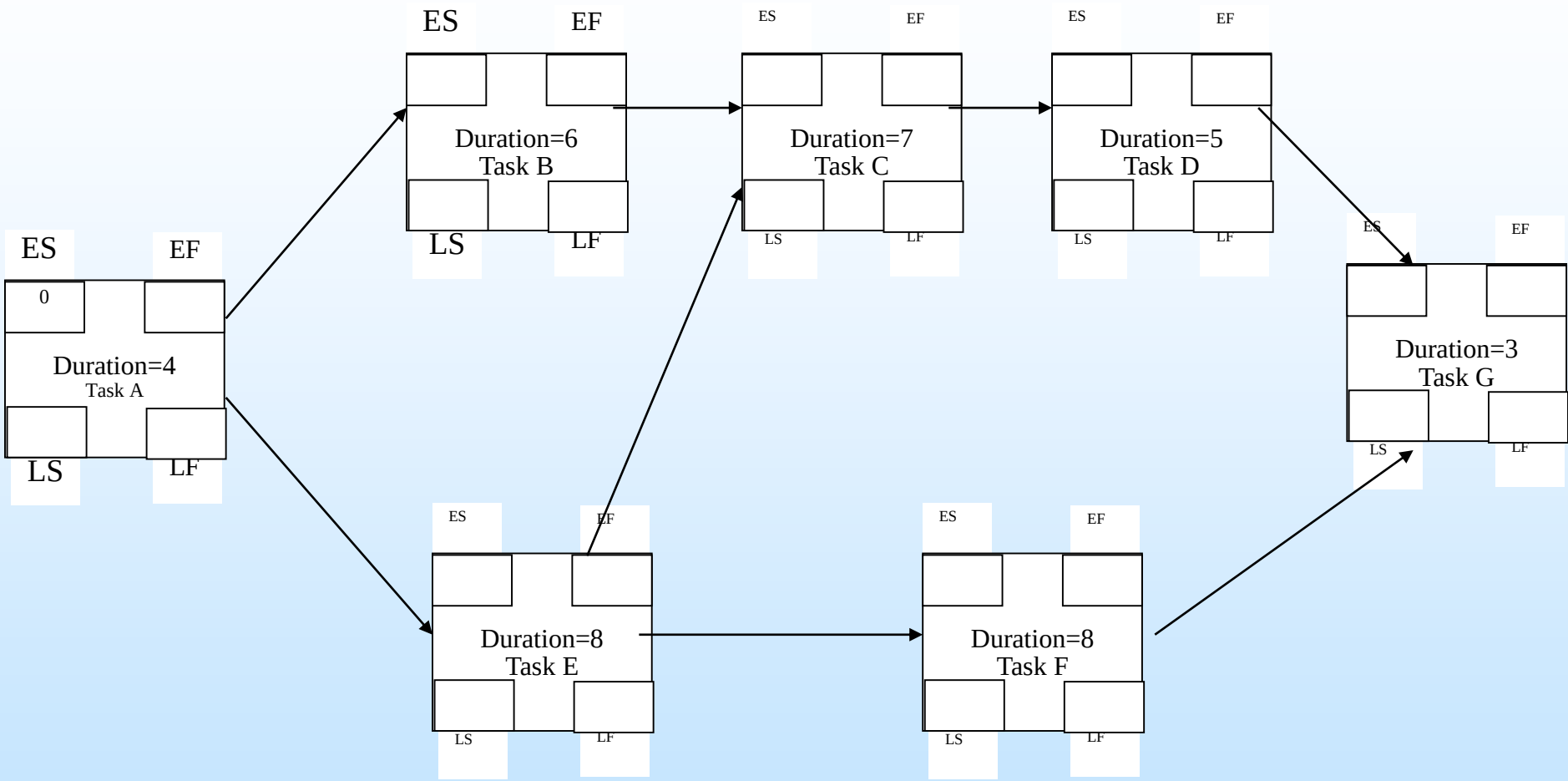
Cp Path:18

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

课堂练习

- 作为项目经理，你需要给一个软件项目做计划安排，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



1.确定ＣＰ以及ＣＰ的长度？

2.Ｆ的自由浮动和总浮动？

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）

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，压缩关键路经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

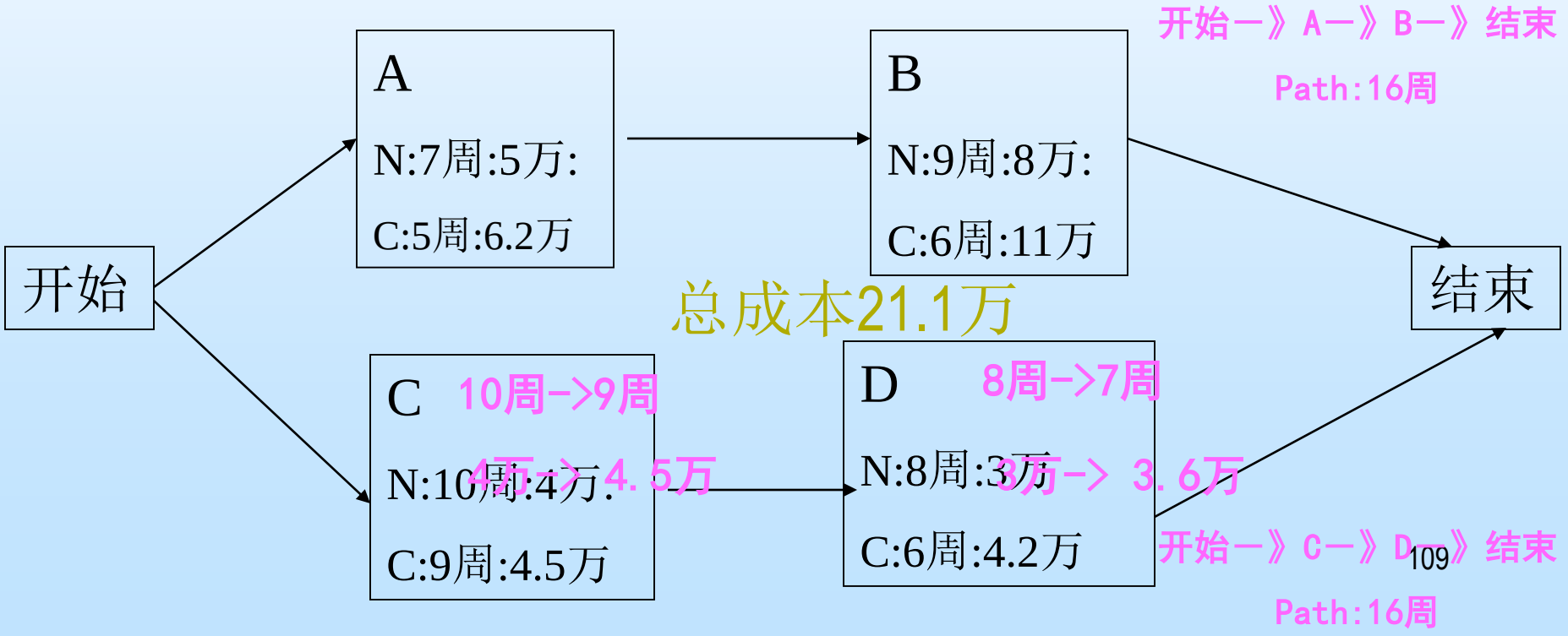
任务 单位压缩成本	A	B	C	D
压缩成本(万/周)	0.6	1	0.5	0.6

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？

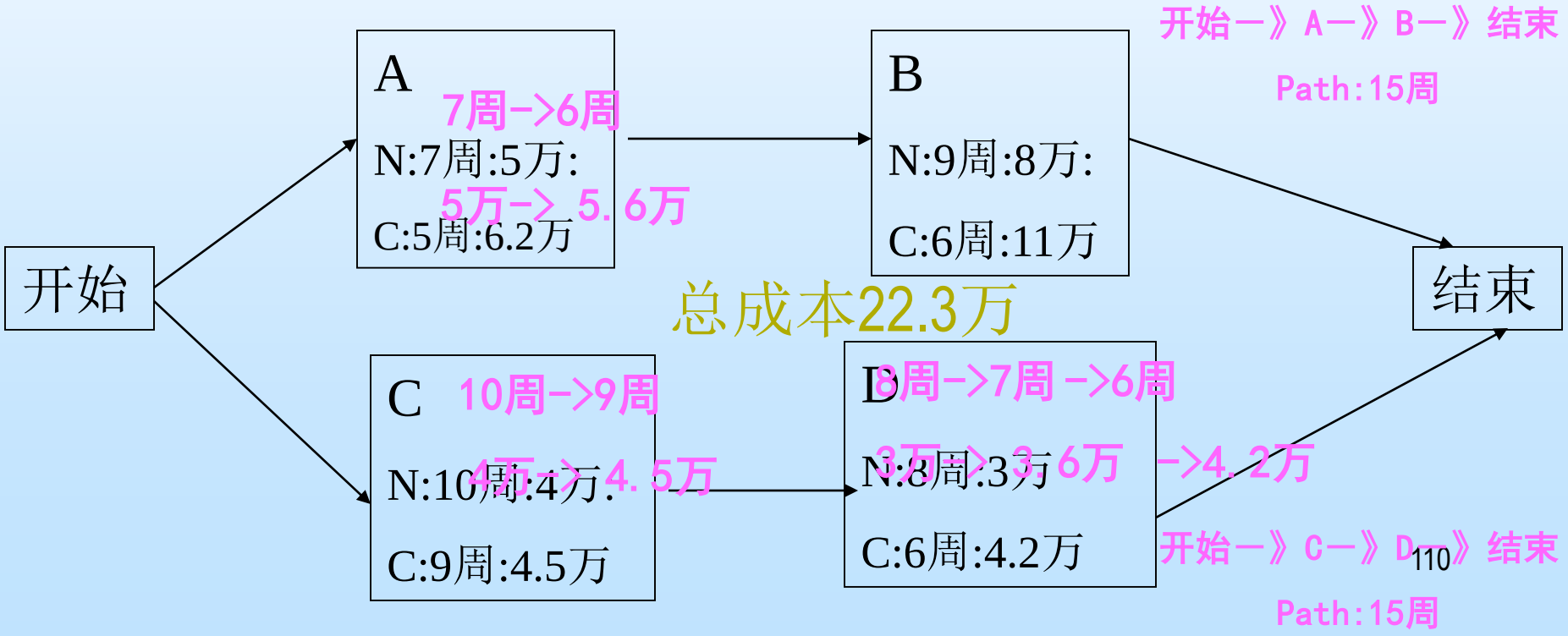
时间压缩例题

- 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

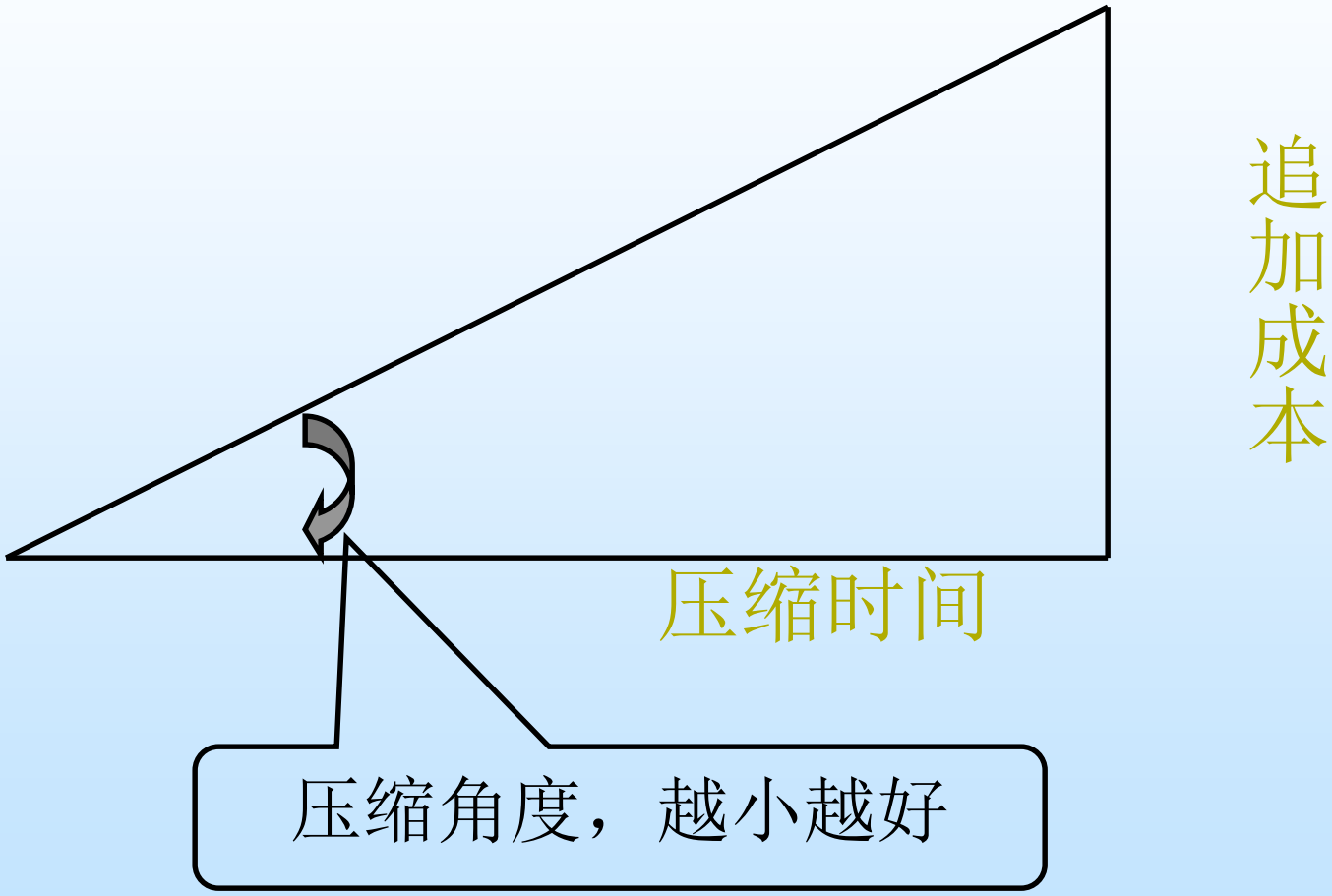
- 将工期压缩到15时应该压缩的活动和最后的成本？



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



关于进度的一些说明

- 项目存在一个可能的最短进度

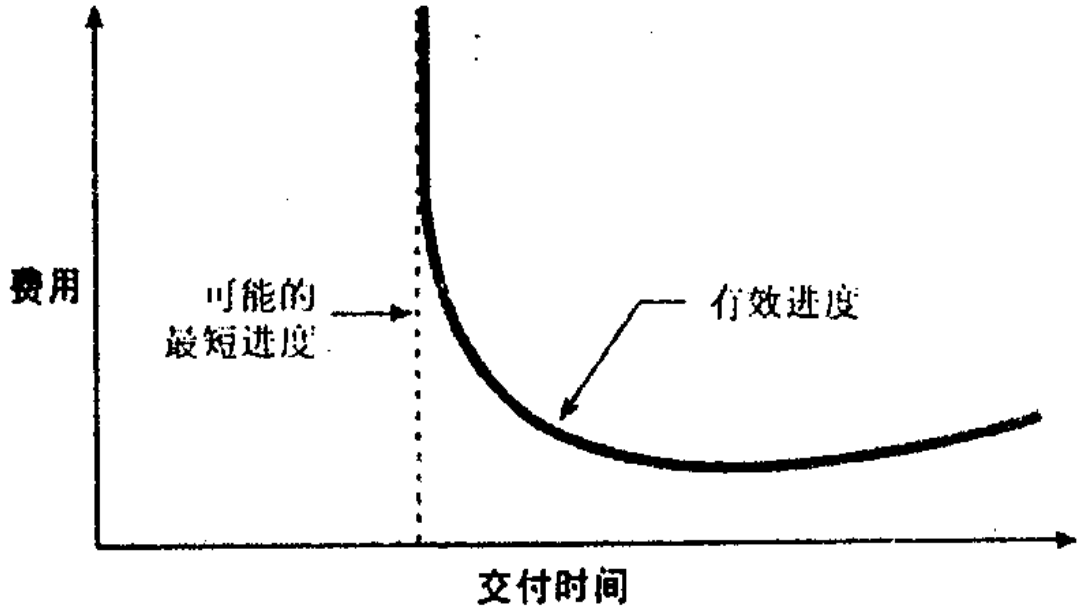


图 2－2－5　软件项目费用与进度之间的关系（完成有效进度的费用大大低于完成最短可能进度的费用）

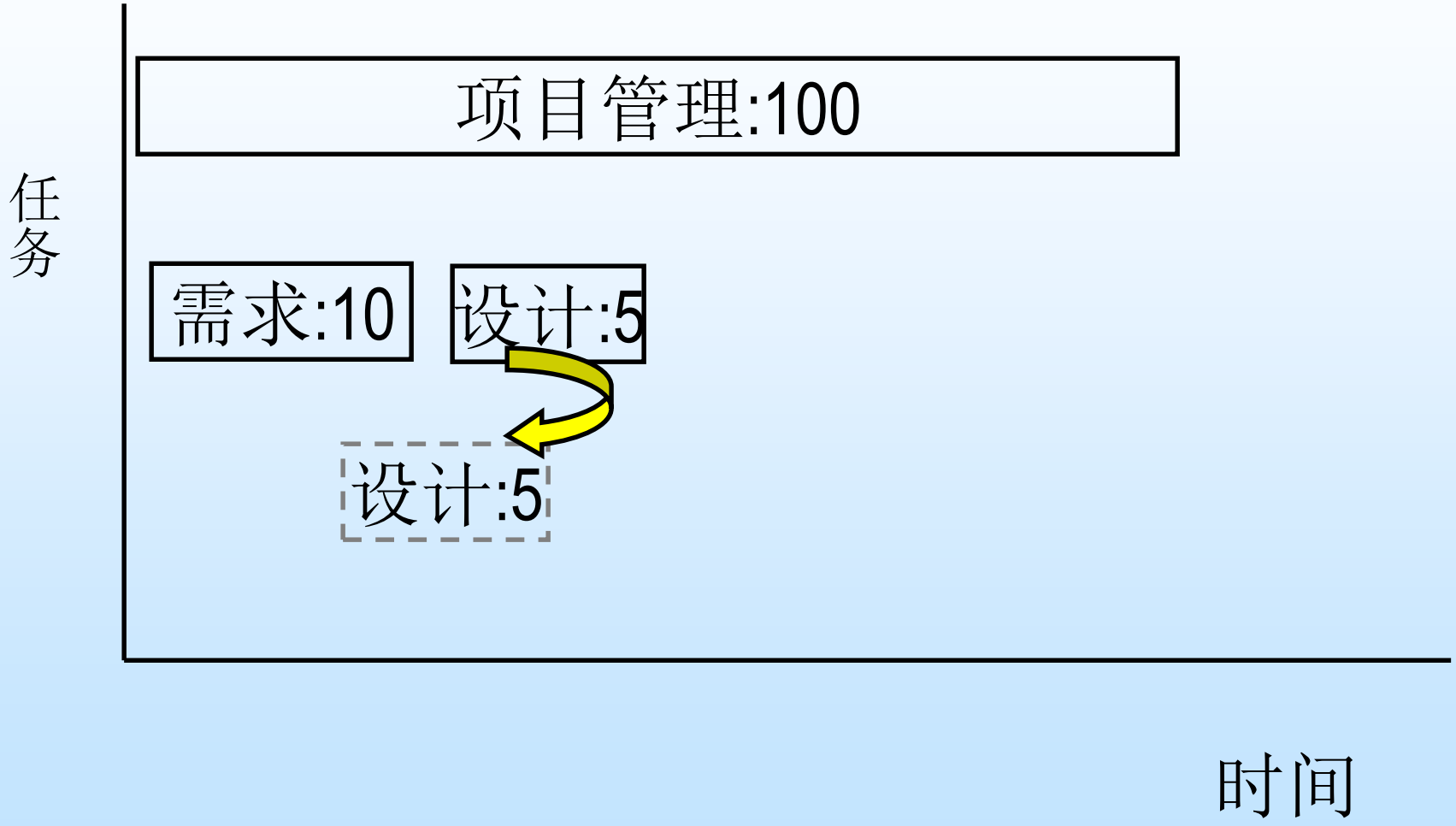
Charles Symons(1991)方法

- 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）

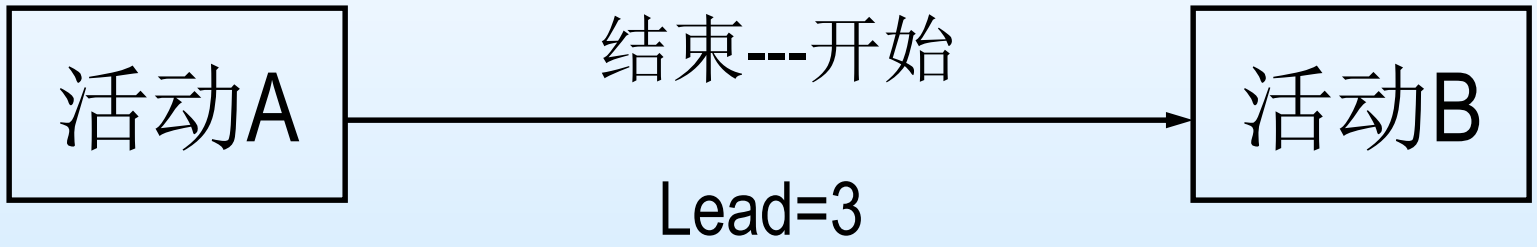
- 是在改变活动间的逻辑关系，并行开展某些活动

进度时间参数



任务超前(Lead)

A完成之前3天B开始



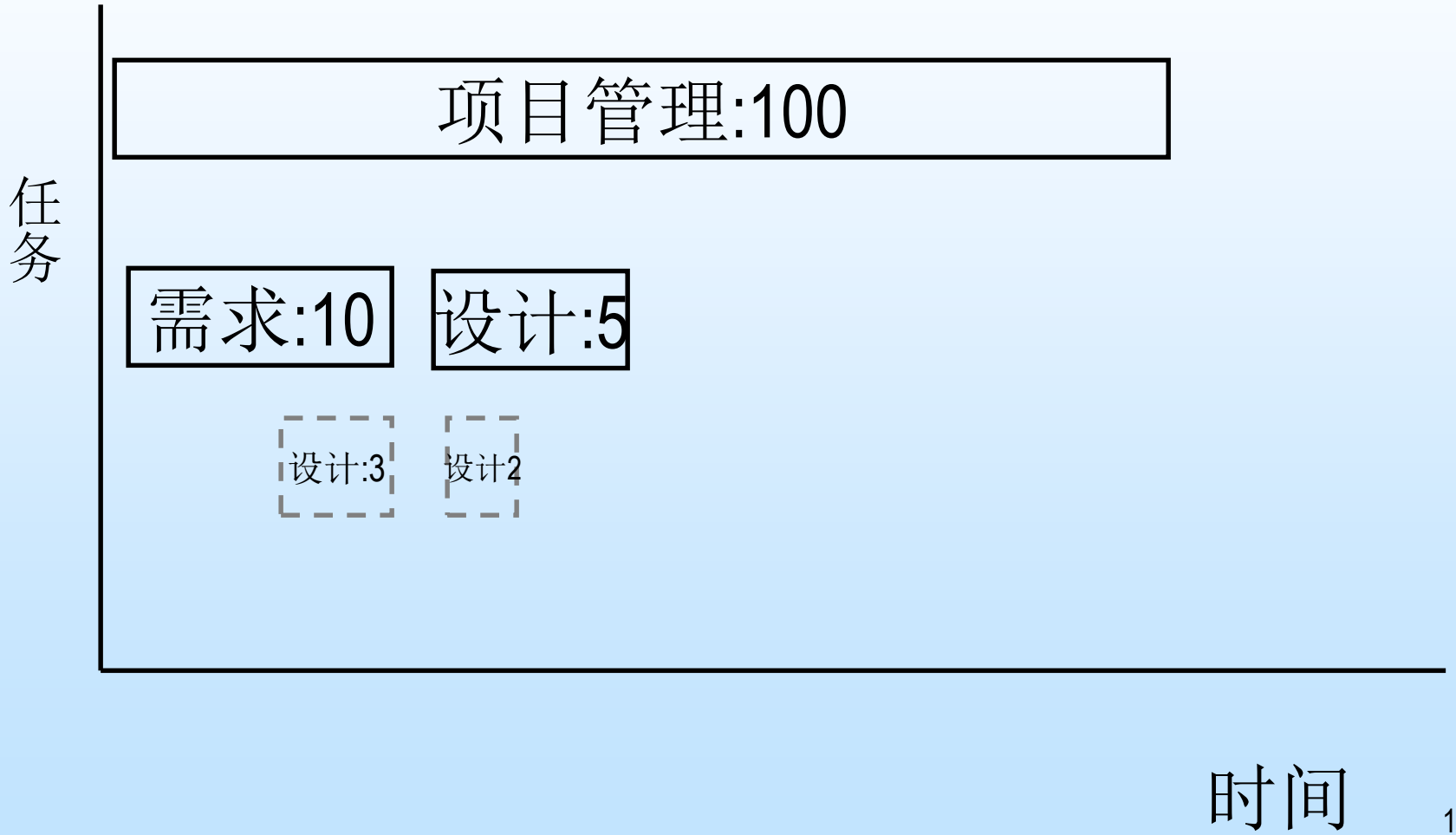
作用：

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



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

资源调整尝试法

- 资源优化配置
- 通过调整进度计划，形成平稳连续的资源需求
  - 最有效的利用资源
  - 使资源闲置的时间最小化
  - 尽量避免超出资源能力
- 方法
  - 资源平衡，维持工期不变，使资源强度尽可能平衡
  - 在满足资源约束条件下，使工期最短

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

项目成本预算

分配项目成本（预算）包括三种情况：

1. 分配资源成本
2. 给任务分配固定资源成本
3. 给任务分配固定成本

1分配资源成本

- 资源成本与资源的基本费率紧密相连
- 设置资源费率
  - 标准费率
  - 加班费率
  - 每次使用费率
  - 。 。 。 。 。 。

2分配固定资源成本

- 当一个项目的资源需要固定数量的资金时，用户可以向任务分配固定资源成本。
- 例如：需要的硬件设备

3分配固定成本

- 有些任务是固定成本的类型的任务，也就是说，用户知道某项任务的成本不变，不管任务的工期有多长，或不管任务使用了那些资源。在这种情况下，用户向任务直接分配成本。
- 例如：培训任务

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

计划优化调整

1. 调整资源, 解决资源冲突
2. 调整进度, 优化项目, 缩短工期
3. 调整项目成本预算, 以便减少项目费用.

调整资源，解决资源冲突

资源冲突(过度分配)主要有两种表现：

1、分配给一个资源的工时总量大于它的最大可用工时量。

2、同一种资源被分配给时间上重叠的几个任务或项目中。

解决资源冲突的方法

- 资源调配
- 推迟资源开始工作时间
- 替换资源
- 设置资源加班时间
- 调整资源日历
- 只使用资源的一部分工作时间

优化进度，缩短工期

- 项目中各任务的执行时间是否合理，有无冲突现象
- 尽可能缩短项目工期

优化进度，缩短工期

1. 分解关键任务
2. 给任务增加资源
3. 缩减关键任务的工期
4. 重叠关键任务
5. 设置日历增加工作时间
6. 通过分配加班工时来缩短关键任务

缩减项目工期

1、分解关键任务

注意：通过“分解关键任务”可以缩短任务工期，但有时候，受资源量的限制，有些任务是不能同步进行的，所以这时任务分拆也无助于缩短项目周期。

缩减项目工期

2、给任务增加资源

注意：

- 增加的资源数量不能大于资源的最大可用量。
- 增加资源必须是主导项目工期的关键路径上。
- 关键任务的缩短可能会变成非关键任务，因此，此时增加过多的资源是无法达到继续缩短总工期的目的的。

缩减项目工期

3、缩减关键任务的工期

注意：在任务已分配了资源的情况下，缩短任务工期意味着增加资源的工作量，可能导致资源的过度分配。

缩减项目工期

4、重叠或延迟链接任务

方法有两种：

- 改变任务的链接关系
- 在链接任务之间增加负延迟

缩减项目工期

5、设置日历增加工作时间

可以通过改变资源的日历来调整工期，比如将资源原来的休息时间改变成工作时间来实现。这样通过增加资源的工作时间来缩短任务的工期。

缩减项目工期

6、通过分配加班工时来缩短关键任务

需要在关键任务上为资源设置加班时间，以缩短任务工期。

调整项目成本预算

降低预算成本的方法:

1. 降低资源的费率
2. 减少任务的工时
3. 减少加班
4. 替换资源
5. 减少任务的固定成本
6. 删除任务

减少项目成本

1. 降低资源的费率

降低资源的费率往往会打击工作人员的积极性，但可以通过降低其他资源的费率来实现，比如降低能源消耗、设备费用等。

2. 减少任务的工时

适当的减少工时，可以降低任务的费用。但减少工时同时也影响项目的工期。

减少项目成本

3．减少加班

加班需要支付加班费率，这通常要高于资源费率，所以减少加班可以有效的减少任务成本。

减少项目成本

4. 替换资源

用廉价的资源替换比较高价的资源，但有一个前提，那就是替换的资源同样能胜任这项任务。

5. 减少任务的固定成本

固定成本就是任务本身所需要的成本。

减少项目成本

6．删除任务

确认删除改任务对项目没有影响或影响在可控制范围内才可采用。

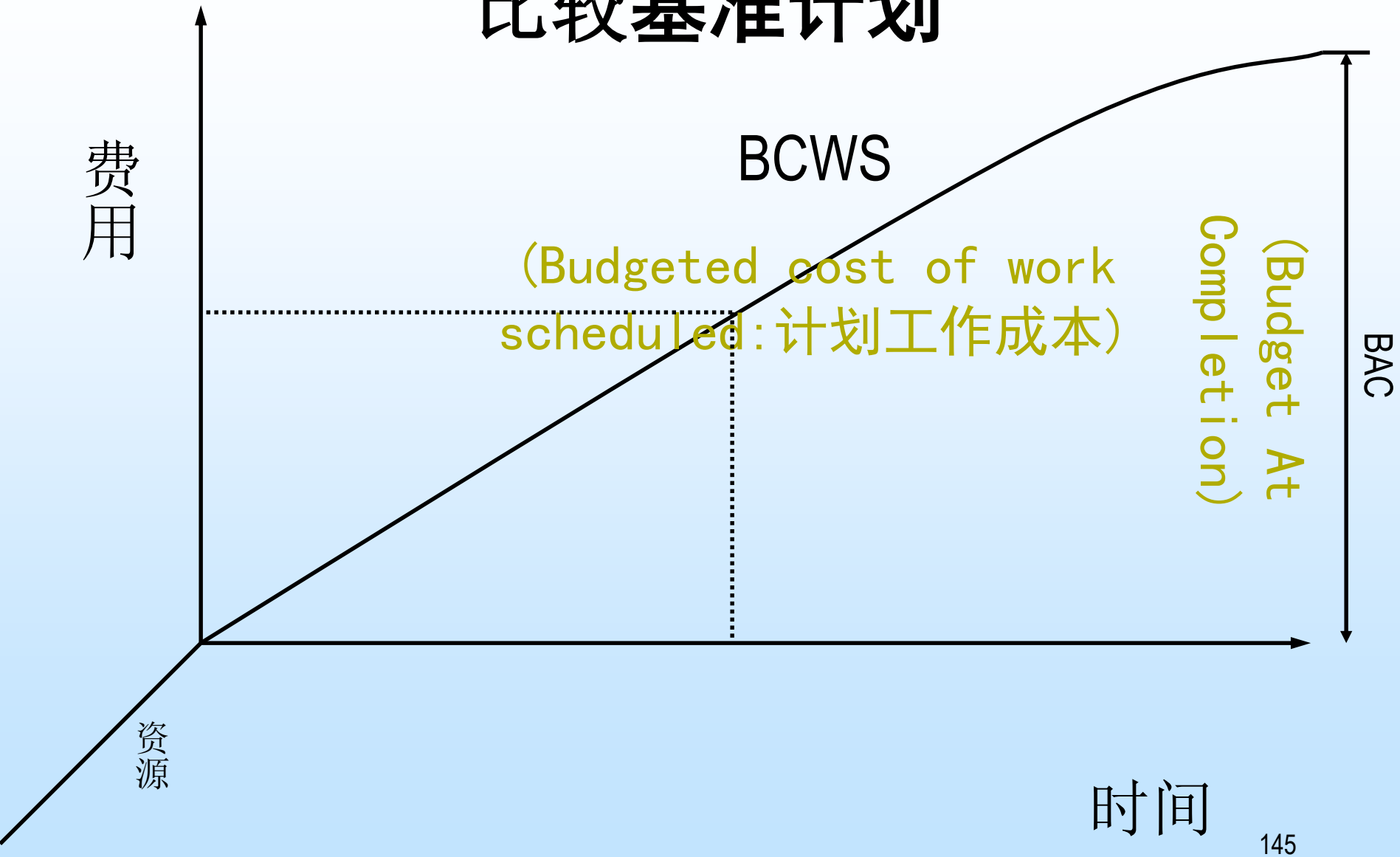
最后审查

- 角色
  - 是否每个任务都有完整的角色，如果需要就增加角色
- 人员
  - 这些角色都做什么？必要的分配任务
- 技能
  - 分配任务的人是否具备应有的能力
- 可行性
  - 什么时候真的需要这些人

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

比较基准计划


费用
BCWS
(Budgeted cost of work scheduled:计划工作成本)
(Budget At Completion)
BAC
资源
时间

Microsoft Project

- Microsoft的项目管理软件产品
- 占领通用项目管理软件市场的大量份额
- 软件项目管理, 主要采用Microsoft Project工具
- 版本：98、2000、2002、2003、2013、2016

用Project工具编制项目计划

1. 创建项目文件
2. 创建项目的任务
3. 确定任务之间的关系
4. 为任务分配资源
5. 安排任务的工期
6. 分配项目成本，进行成本预算
7. 调整优化计划