

软件项目管理

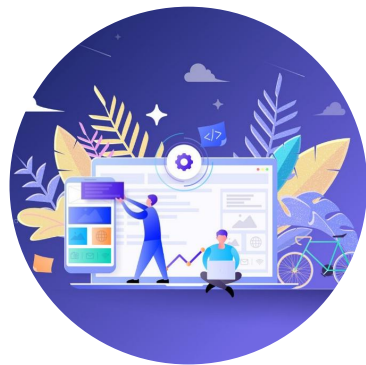
第二篇 项目计划

软件学院
罗昕



 luoxin@sdu.edu.cn

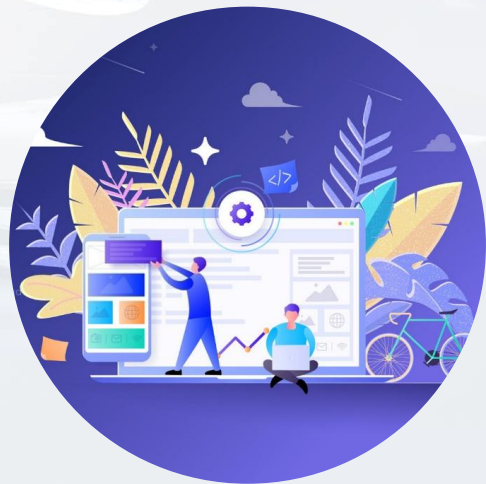
 软件学院办公楼-425



第二篇 第7章 软件项目进度计划

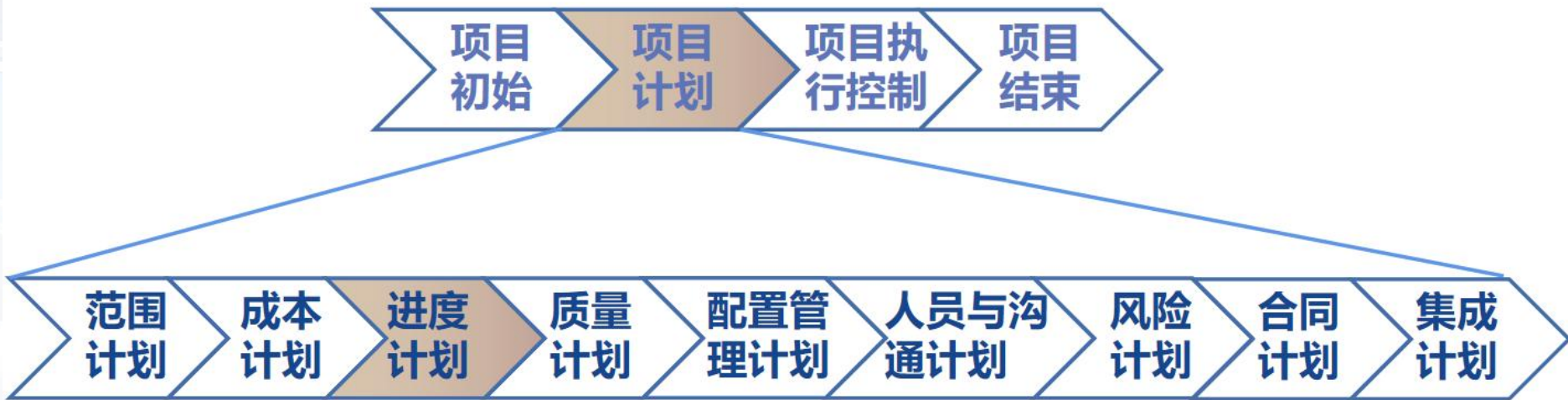


软件学院 罗昕
luoxin@sdu.edu.cn



《软件项目管理》 - 路线图

MIMA



- **进度管理基本概念**
- 任务资源估计
- 任务历时估算
- 进度计划编排
- 项目进度模型
- 案例分析
- 课程实践

- 时间是一种特殊的资源，以其单向性、不可重复性、不可替代性而有别于其他资源
- 进度是对执行的活动和里程碑制定的工作计划日期表
- 一个项目管理者应该定义所有的项目任务，识别出关键任务，跟踪关键任务的进展情况，同时能够及时发现拖延进度的情况。为此，项目管理者必须制定一个足够详细的进度表，以便 监督项目进度并控制整个项目

■ 进度计划的重要性

- 按时完成项目是项目经理最大的挑战
- 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

■ 主要过程：

- 根据WBS分解出主要的任务（活动）；
- 确立任务（活动）之间的关联关系；
- 然后估计每个任务（活动）需要的资源、时间；
- 最后编制出项目的进度计划。

7.2.1 任务定义

- WBS的每个工作包需要被划分成所需要的活动（任务），每个被分配的活动（任务）都应该与一个工作包相关，通过任务（活动）定义这一过程可使项目目标体现出来。
- 任务定义：确认和描述一些特定的活动的过程。
- 完成了这些活动意味着完成了WBS中的项目细目和子细目。
- 对WBS做进一步地分解

7.2.2 项目任务的关联关系

- 任务定义之后，接下来需要确定任务之间的关系。为了进一步制定切实可行的进度计划，必须对活动（任务）进行适当的顺序安排。它是通过分析所有的任务，项目的范围说明以及里程碑等信息来确定各个任务之间的关系。
- 项目各项任务之间存在相互联系与相互依赖关系
- 根据这些关系安排任务之间的顺序
- 前置活动（任务）---> 后置活动（任务）
- 排序可由计算机执行（利用计算机软件）或手工排序

7.2.2 任务(活动)之间的关系

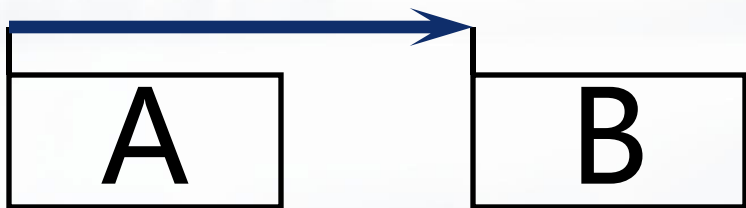
MIMA



结束-开始 FS



结束-结束 FF



开始-开始 SS



开始-结束 SF

7.2.2 任务(活动)之间的关系 - 示例

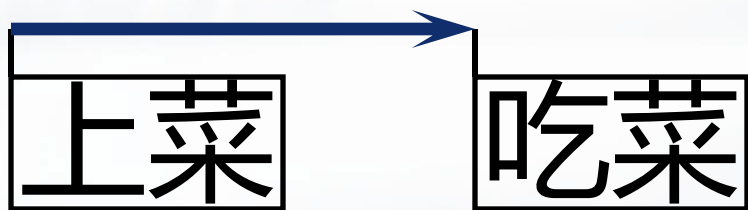
MIMA



结束-开始 FS



结束-结束 FF



开始-开始 SS



开始-结束 SF

7.2.2 任务(活动)间关系的依据

- 确定任务（活动）之间关联关系的依据：
- **硬逻辑关系** （强制性依赖关系）
固有的、不可违背的
- **软逻辑关系**
人为的、主观的
- **外部依赖关系**
项目活动与非项目活动之间

7.3 进度管理图示

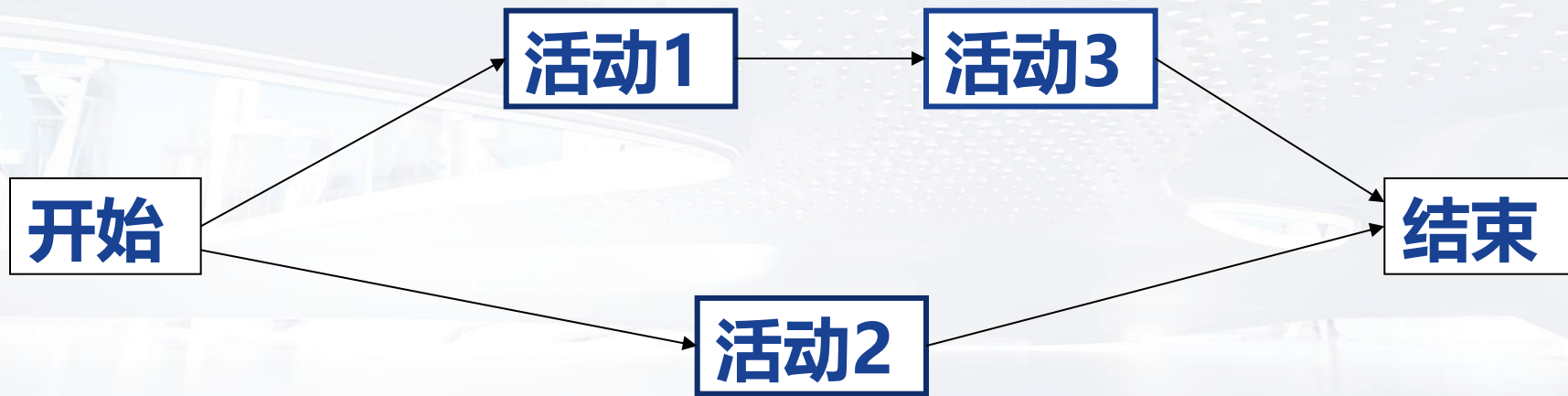
- 项目进度管理的图示:
- **网络图** network diagramming
 - 节点法/单代号网络图 PDM (Precedence Diagramming Method)
 - 箭线法/双代号网络图 ADM (Arrow Diagramming Method)
- **甘特图** Gantt图
 - 棒状甘特图
 - 三角形甘特图
- **里程碑图**
- **资源图**
- **燃尽图/燃起图**

7.3 进度管理图示 → 网络图

- 网络图是活动排序的一个输出
- 展示项目中各个活动以及活动之间的逻辑关系
- 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

7.3 进度管理图示 → 网络图 → PDM

MIMA

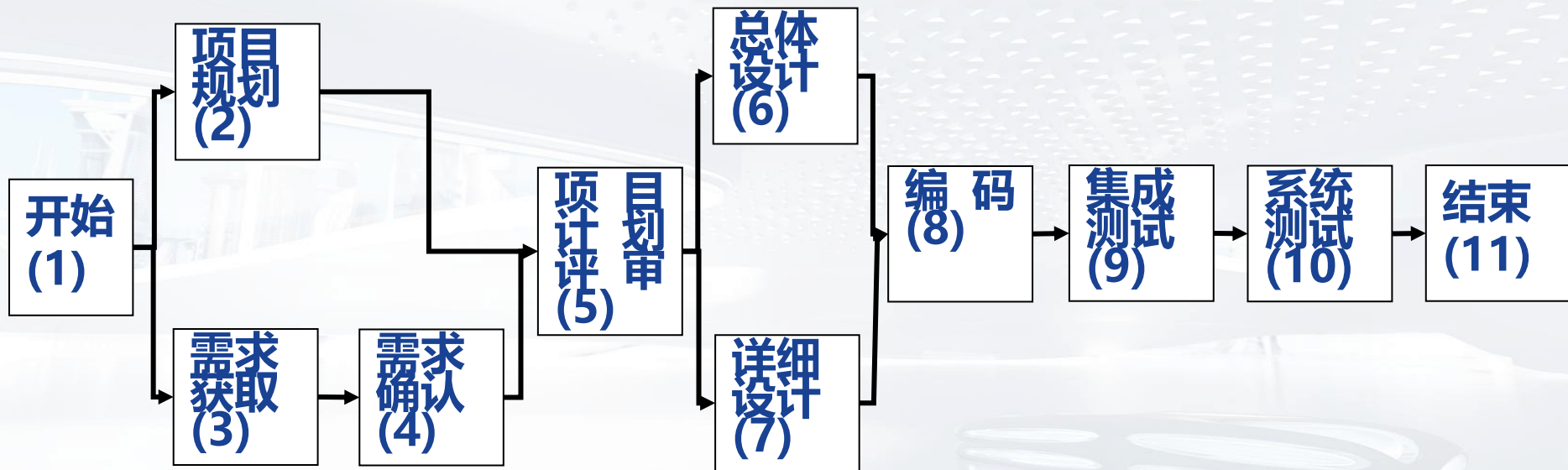


优先图法/节点法/单代号网络图
Precedence Diagramming Method

节点表示活动(任务)
箭线表示各活动(任务)
之间的逻辑关系

7.3 进度管理图示 → 网络图 → PDM

MIMA

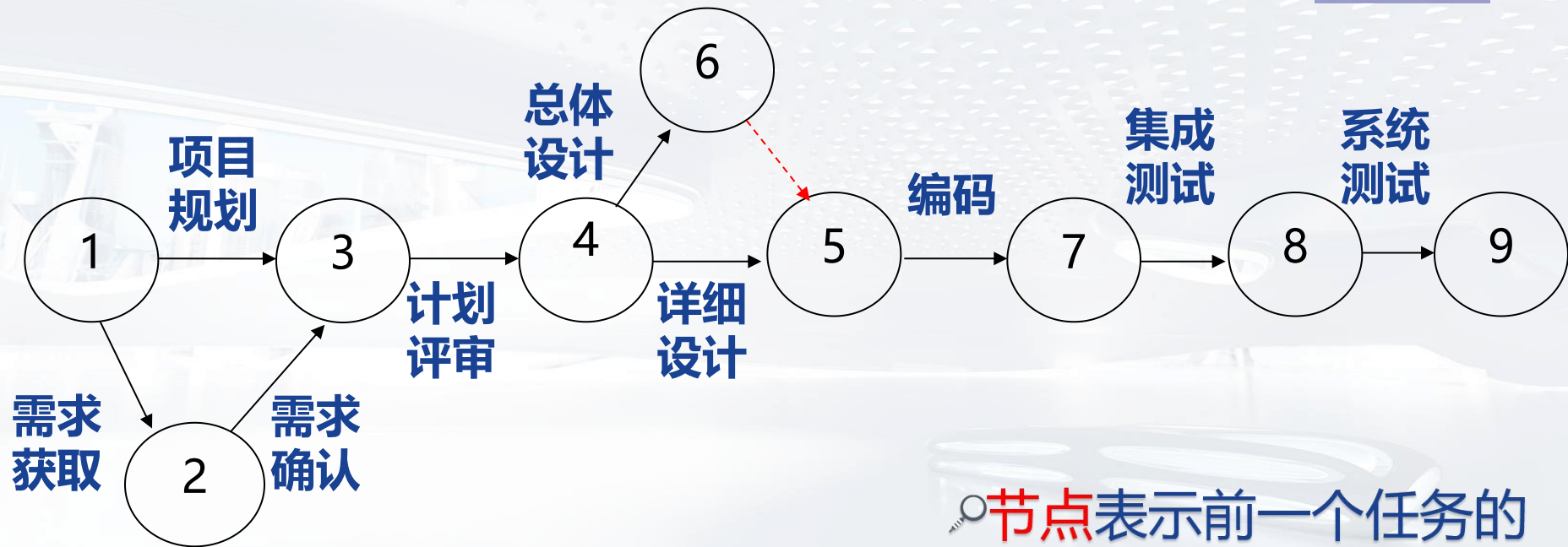


优先图法/节点法/单代号网络图
Precedence Diagramming Method

节点表示活动(任务)
箭线表示各活动(任务)
之间的逻辑关系

7.3 进度管理图示 → 网络图 → ADM

MIMA



箭线法/双代号网络图

Arrow Diagramming Method

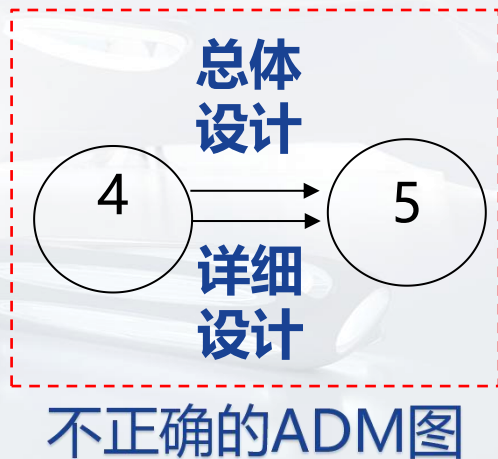
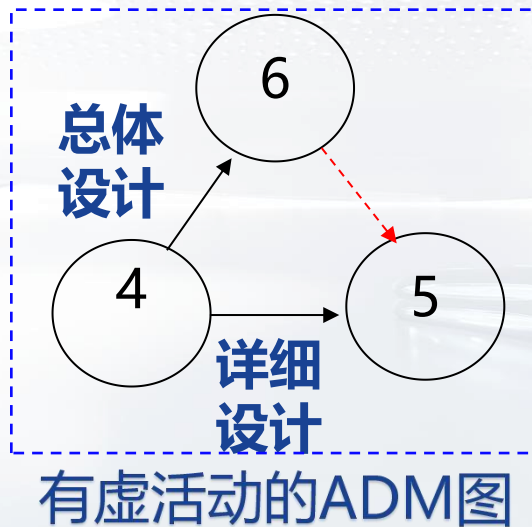
节点表示前一个任务的结束以及后一个的开始
箭线表示任务

7.3 进度管理图示 → 网络图 → ADM

- 双代号表示网络图中两个代号**唯一**确定一个任务

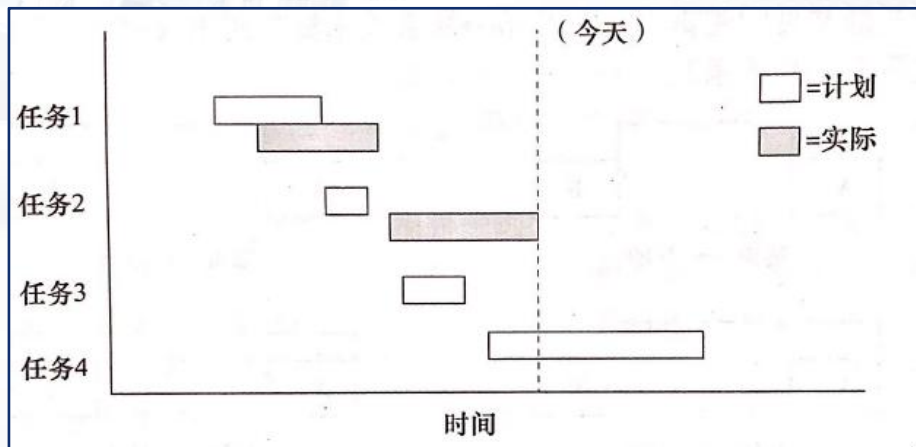
- 虚活动

- 为了定义活动
- 为了表示逻辑关系
- 不消耗资源的

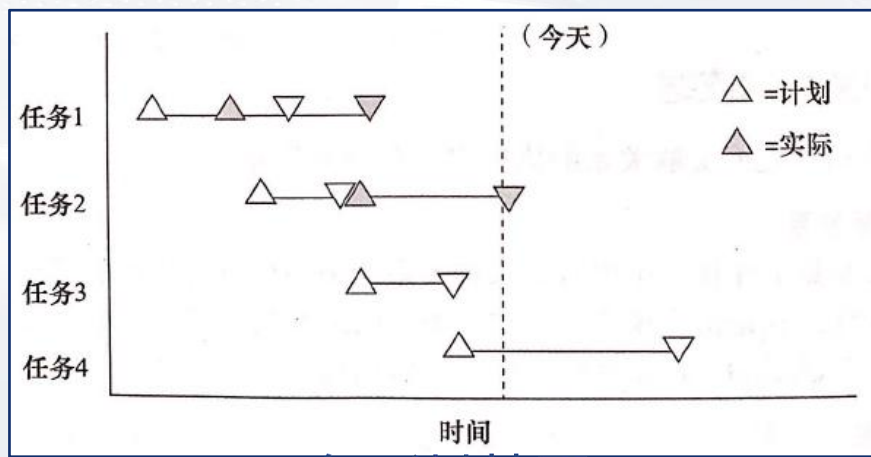


7.3 进度管理图示 → 甘特图

- 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



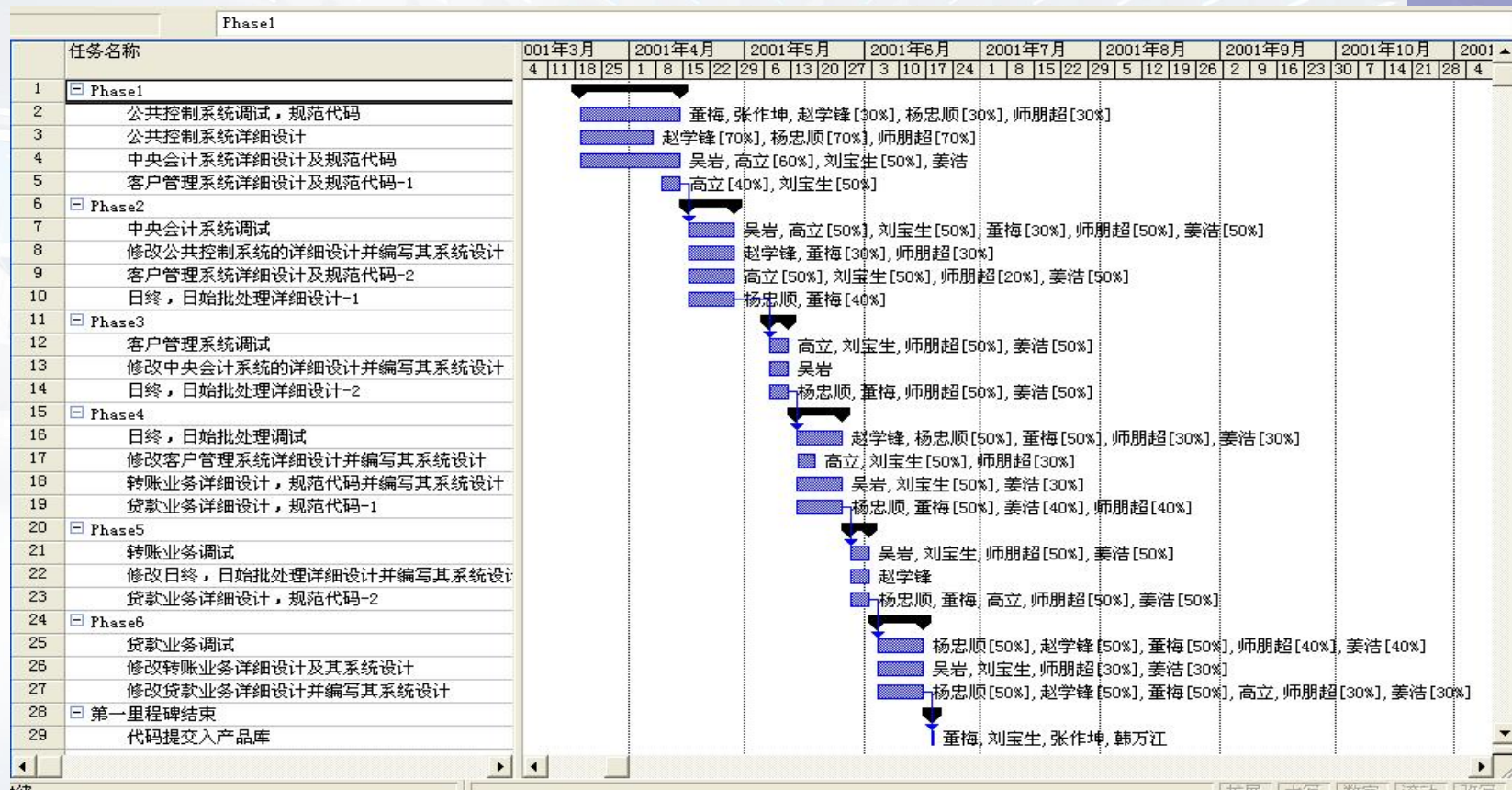
棒状甘特图



三角形甘特图

7.3 进度管理图示 → 甘特图

MIMA



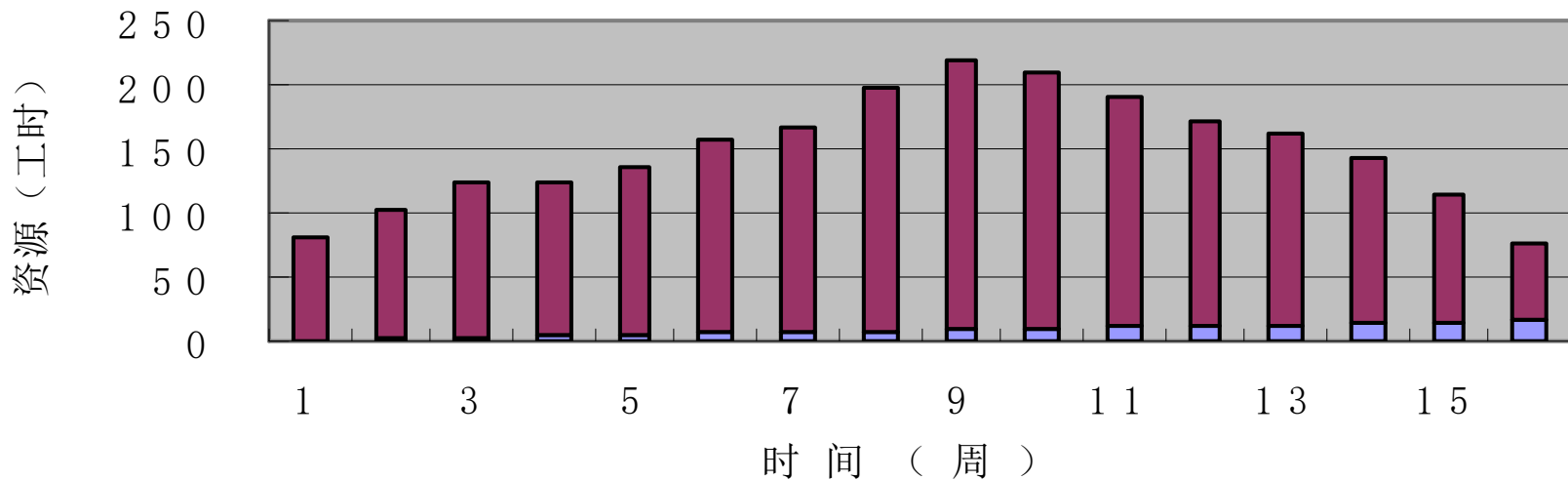
7.3 进度管理图示 → 里程碑图

- 是由一系列的里程碑事件组成的
- 里程碑图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



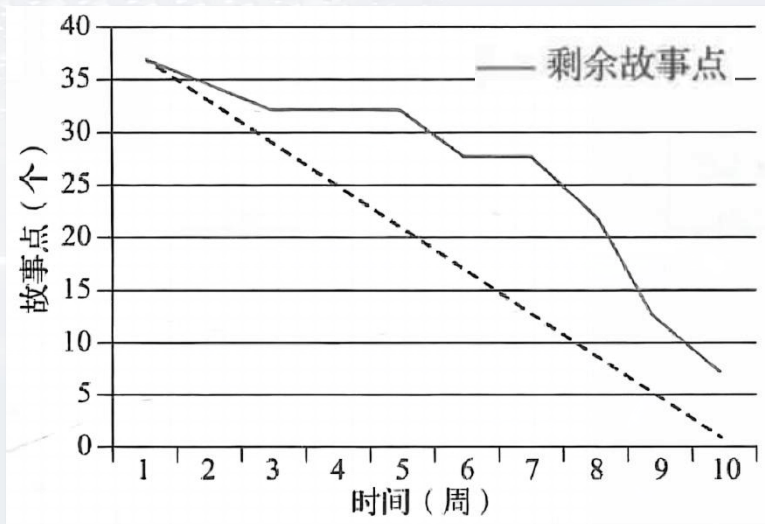
7.3 进度管理图示 → 资源图

资 源 图



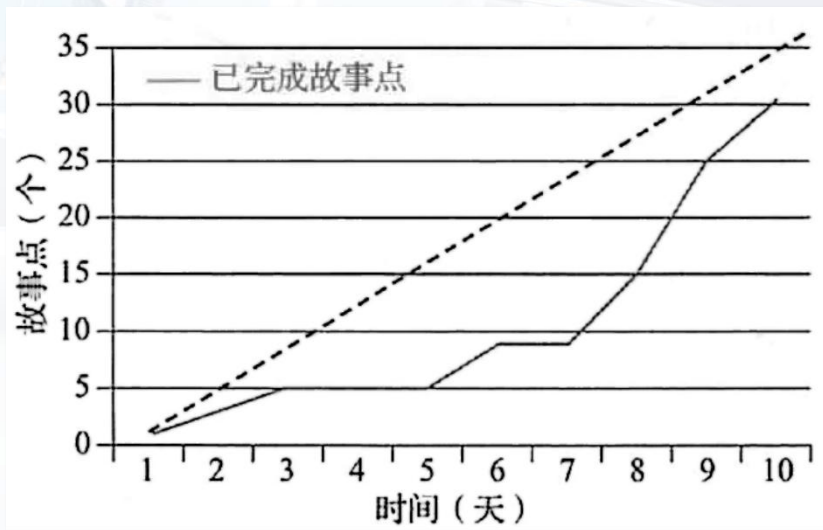
7.3 进度管理图示 → 燃尽图

- 燃尽图是在项目完成之前，对需要完成的工作的一种可视化表示。
- 在燃尽图中，Y轴表示工作量，X轴表示项目时间。
- 燃尽图描述随着时间的推移而剩余的工作数量，可用于表示开发速度。
- 理想情况下，燃尽图是一个向下的曲线，随着剩余工作的完成，“烧尽”至零。

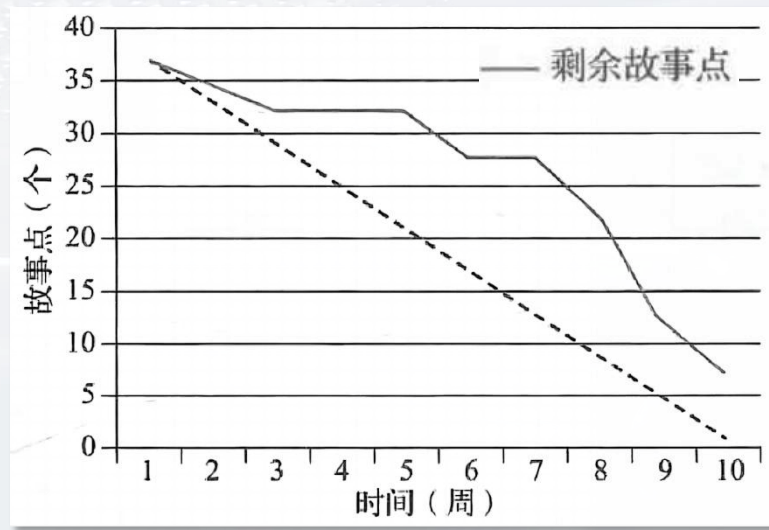


7.3 进度管理图示 → 燃起图

- 燃起图显示已完成的工作。Y轴表示工作量，X轴表示项目时间。



燃起图



燃尽图

7.3 进度管理图示 → 燃尽图 对比 燃起图

燃尽图 (Burn Down Chart) :

- 仅展示了从项目开始到终点的剩余工作量的减少
- 不直接显示工作量的增加, 如果项目范围发生变化需要重新绘制图表
- 通常不直接显示范围变化, 如果增加任务, 剩余工作量会突然增加, 可能使图表读者难以理解原因
- 理想情况下, 图表是一个向下的曲线, 随着剩余工作的完成, 烧尽至零

燃起图 (Burn Up Chart) :

- 展示了随时间变化的完成工作量以及项目的总工作量
- 通过两条线 (完成工作量和总工作量) 来展示进度与目标, 使得目标非常明确
- 能够清晰地表现出项目范围的变化, 因为总工作量线会随着变更而调整
- 如果项目范围有所变化, 总工作量线会上升或下降, 从而直观反映范围的变动

软件项目进度计划 学习要点

MIMA

- 进度管理基本概念
- **任务资源估计**
- 任务历时估算
- 进度计划编排
- 项目进度模型
- 案例分析
- 课程实践

- 在估计每个任务的历时之前，首先应该对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、设备资源及其他资源等。

1)对于特定的任务，它的难度如何？ 2)是否有唯一的特性影响资源的分配？ 3)企业以往类似项目的状况如何？个人的成本如何？ 4)企业现在是否有完成项目合适的资源(人、设备、资料等)？企业的政策是否能够影响这些合适的资源？ 5)是否需要更多的资源来完成这个项目？是否需要外包人员等？

- 进度管理基本概念
- 任务资源估计
- **任务历时估算** 7.5
- 进度计划编排
- 项目进度模型
- 案例分析
- 课程实践

- 任务历时估计是估计任务的持续时间
 - 项目中每个任务的历时估计：分别估算项目各个活动所需要的时间
 - 项目总历时估计：根据项目活动的排序来确定整个项目所需要的时间
- 过长或过短的估计都是不利的
 - 过短，则会使项目团队处于被动紧张的状态
 - 过长，则会延迟项目的完成，可能使项目失去大好的获利机会

■ 一般地，在历时估计的时候，还应该考虑如下信息。

- 1) 实际的工作时间。
- 2) 项目的人员规模。
- 3) 生产率。
- 4) 有效工作时间。
- 5) 连续工作时间。
- 6) 人员级别。
- 7) 历史项目。

■ 介绍几种软件项目历时估计常用的估算方法。

历时估算的基本方法

- 定额估算法
- 经验导出模型
- PERT(工程评估评审技术)
- 专家判断法
- 类比估计法
- 基于承诺的进度估计
- Jones的一阶估算准则
- 预留分析
- 敏捷历时估算

基本方法 → 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，小于10000LOC（代码行）或者说小于6个月的项目

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

T:活动历时

Q:任务规模(工作量)

R:人力数量

S:工作效率(贡献率)

Q=6人天, R=2人, S=1

则: T=3天

Q=6人天, R=2人, S=1.5

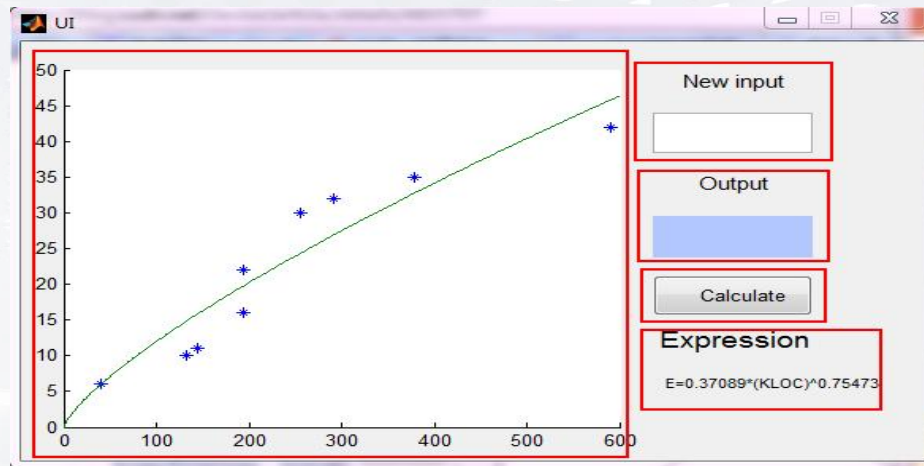
则: T=2天

基本方法 → 经验导出模型

MIMA

■ 经验导出模型: $D = a * E^b$

- D:进度(以月单位)
- E:工作量(以人月单位)
- a:2-4之间
- b:1/3左右
- a、b是依赖于项目的自然属性



■ IBM模型/基本COCOMO模型

$$D = 2.4 * E^{0.35}$$

$$D = 2.5 * E^b \quad (b \text{ 在 } 0.32 \sim 0.38 \text{ 之间})$$

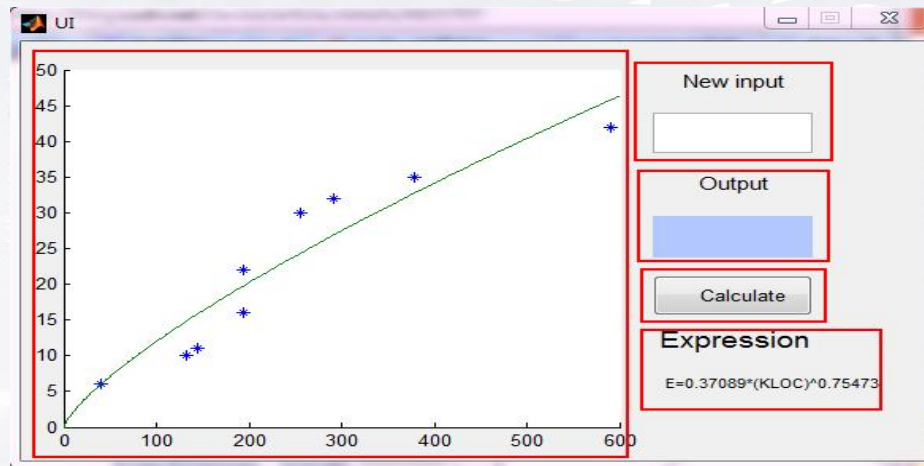
有机	0.38
半有机	0.35
嵌入式	0.32

基本方法 → 经验导出模型

MIMA

■ 经验导出模型: $D = a * E^b$

- D:进度(以月单位)
- E:工作量(以人月单位)
- a:2-4之间
- b:1/3左右
- a、b是依赖于项目的自然属性



■ IBM模型/基本COCOMO模型

$$D = 2.4 * E^{0.35}$$

$$D = 2.5 * E^b \quad (b \text{ 在 } 0.32-0.38 \text{ 之间})$$

有机	0.38
半有机	0.35
嵌入式	0.32

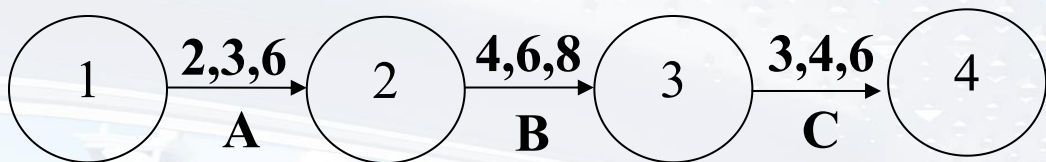
基本方法 → 工程评估评审技术

- PERT: Program Evaluation and Review Technique
- 于1958年，为适应大型工程的需要，并取得不错效果
- 利用网络顺序图的逻辑关系和加权算法估算任务历时

基本方法 → PERT → 加权算法

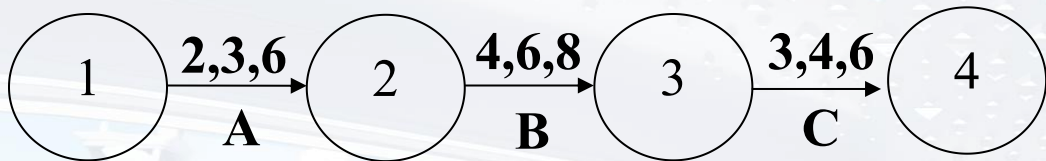
- 它是基于对某项任务的乐观，悲观以及最可能的概率时间估计
- 采用加权平均得到期望值
$$\text{PERT历时} = (O + 4m + P) / 6$$
 - O是最小估算值:乐观(Optimistic Time)
 - P是最大估算值:悲观(Pessimistic Time)
 - M是最可能估算(Most Likely Time)

基本方法 → PERT → 例



$$\text{PERT历时} = (O+4m+P) / 6$$

基本方法 → PERT → 例



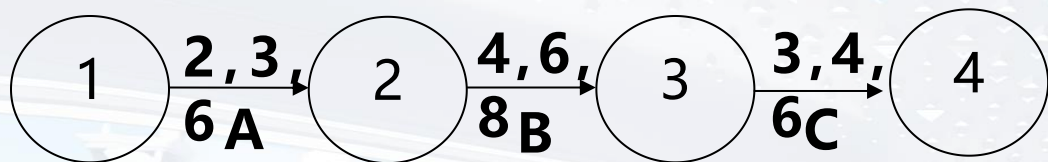
活动项	O,M,P	PERT估计值
A	2,3,6	3.33
B	4,6,8	6
C	3,4,6	4.17
估计项目总历时		

$$\text{PERT历时} = (O + 4m + P) / 6$$

← $\text{PERT历时} = (4 + 4 \times 6 + 8) / 6 = 6$

基本方法 → PERT → 例

MIMA



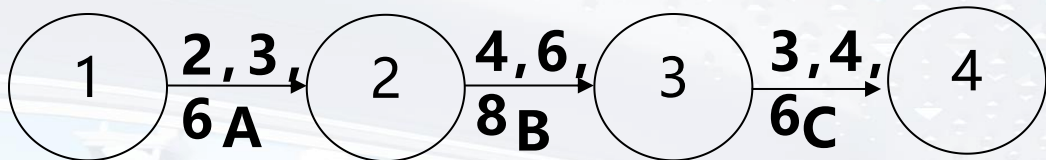
$$\delta = \frac{P - O}{6} \quad \text{标准差}$$

$$\delta^2 = \left(\frac{P - O}{6} \right)^2 \quad \text{方差}$$

活动项	O,M,P	PERT估计值	δ	δ^2
A	2,3,6	3.33	4/6	16/36
B	4,6,8	6	4/6	16/36
C	3,4,6	4.17	3/6	9/36
估计项目总历时				

$$\delta_B^2 = \left(\frac{8 - 4}{6} \right)^2 = \frac{16}{36}$$

基本方法 → PERT → 例



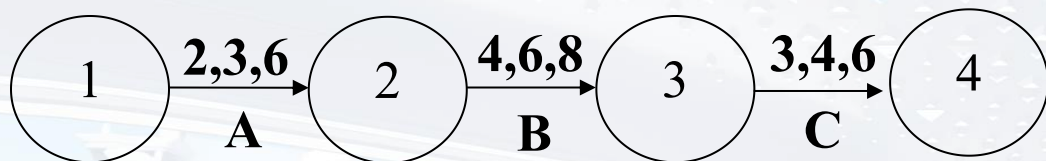
活动项	O,M,P	PERT估计值	δ	δ^2
A	2,3,6	3.33	4/6	16/36
B	4,6,8	6	4/6	16/36
C	3,4,6	4.17	3/6	9/36
估计项目总历时		13.5	1.07	41/36

$$E = E_A + E_B + E_C$$

$$\delta^2 = \delta_A^2 + \delta_B^2 + \delta_C^2$$

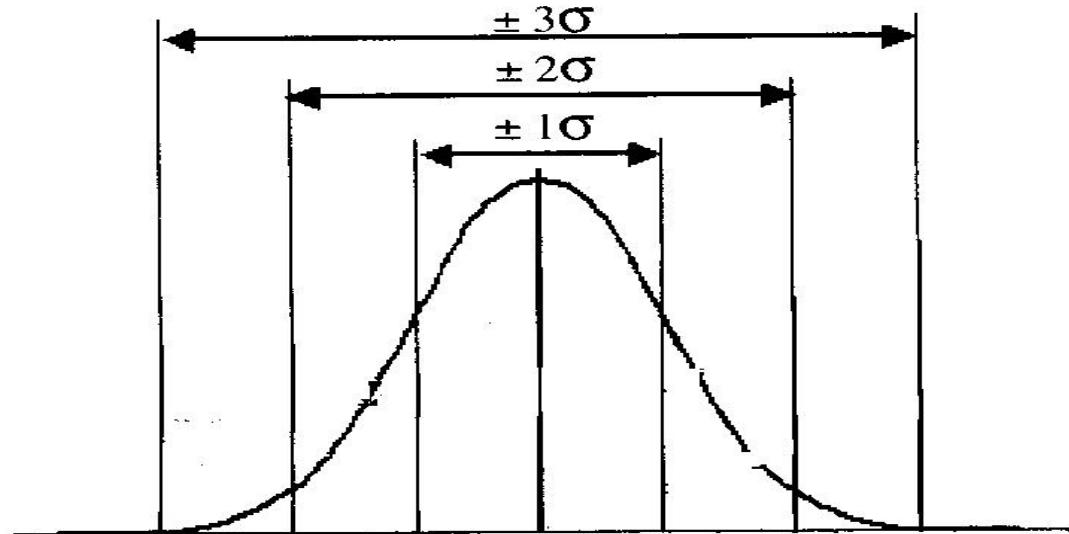
$$\delta = \sqrt{\delta_A^2 + \delta_B^2 + \delta_C^2}$$

基本方法 → PERT → 例



$$E = 13.5$$

$$\delta = 1.07$$



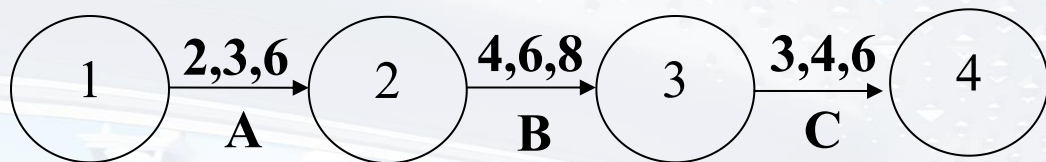
$$E \pm \delta \quad 68.3\%$$

$$E \pm 2\delta \quad 95.5\%$$

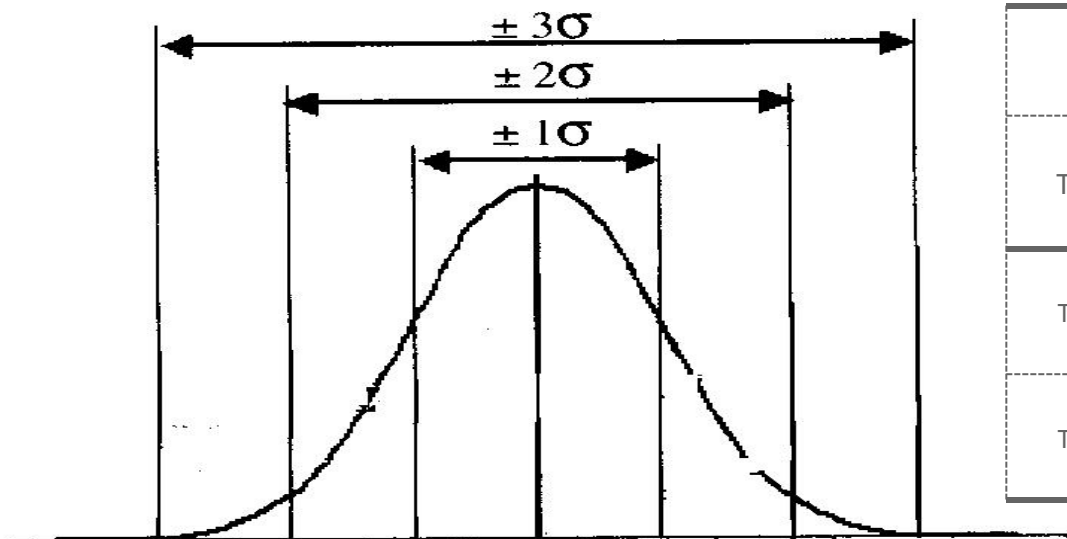
$$E \pm 3\delta \quad 99.7\%$$

基本方法 → PERT → 例

MIMA



$$E = 13.5 \quad \delta = 1.07$$



范围		概率	从	到
T_1	$\pm \delta$	68.3%	12.43	14.57
T_2	$\pm 2\delta$	95.5%	11.36	15.64
T_3	$\pm 3\delta$	99.7%	10.29	16.71

基本方法 → 专家判断法

- 专家判断是指基于某应用领域、知识领域、学科和行业等的专业知识而做出的关于当前活动的合理判断，这些专业知识可来自具有专业学历、知识、技能、经验或培训经历的任何小组或个人。
- 专家判断法是通过专家依靠过去资料信息进行判断，以估算进度的方法。
- 如果找不到合适专家，则估计结果往往不可靠且具有较大风险。

基本方法 → 类比估计方法

- 类比估计也称为类推估计，使用相似活动或项目的历史数据来估算当前活动或项目的持续时间的技术。
- 是一种粗略的估算方法，项目详细信息不足时，就经常使用类比估计方法。通常成本较低、耗时较少，但准确性也较低。对于软件项目，利用企业的历史数据进行历时估计是常见的方法。
- 以下情况的类比估计是可靠的：
 - ① 先前活动和当前活动在本质上类似而不仅仅是表面相似；
 - ② 专家有所需专长。

基本方法 → 基于承诺的进度估计法

- 基于开发人员做出的进度承诺进行
- 不进行中间的工作量（规模）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
- 缺点
 - 易于产生大的估算误差

基本方法 → Jones的一阶估算准则

- 基于估算项目功能点
- 从幂次表中选择合适的幂次将功能点升幂
- 例如，一个商业软件的功能点 $FP=350$ ，若一个平均水平的软件公司来承担，粗略的进度估算为 $350^{0.43}=12$ 月。

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

一阶幂次表

基本方法 → 预留分析

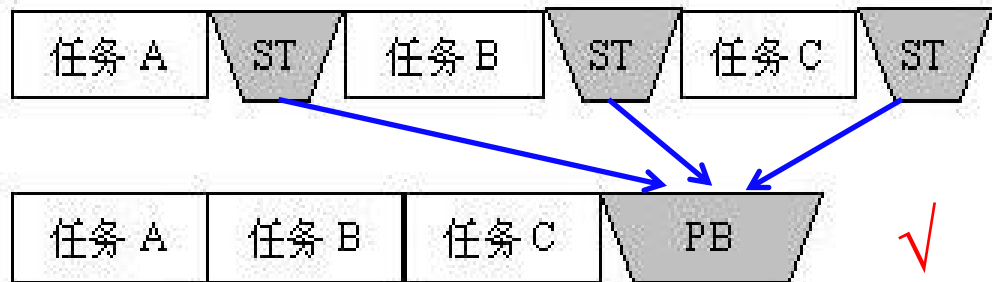
- 帕肯森定律(Parkinson's law)指出工作总是拖延到它能够允许最迟完成的那一天。
- 如果工作允许拖延、推迟完成，往往这个工作总是推迟到它能够最迟完成的那一刻，很少有提前完成的。
- 如果一项任务需要花费10小时完成，可能任务执行者自己知道只需要6小时就可以完成，但不可思议地花费了10小时。这样他们会不珍惜时间，可能会找一些其他事情来做或者简单地等待，直到预留时间花完，才开始正常的工作以期待将项目成功完成。

基本方法 → 预留分析

- 在进行持续时间估算时，需考虑应急储备(有时称为“进度储备”)，以应对进度方面的不确定性。
- 预留分析也称为储备分析，用于确定项目所需的应急储备(预留)量和管理储备(预留)量。
- 应急储备是包含在进度基准中的一段持续时间，用来应对已经接受的已识别风险。应急储备可取活动持续时间估算值的某一百分比(如10%~15%)或某一固定的时间段。

基本方法 → 预留分析

- 预留的应急储备应该是将每一项任务的预留时间累加在一起放在关键路径末端，而不要增加每一项任务时间，即把应急储备从各个活动中剥离出来并汇总。



Safety Time 安全时间
Project Buffer 缓冲时间

- 当一项任务超出了分配的时间时，超出的部分可以使用这个预留时间。

基本方法 → 敏捷历时估算

- 在敏捷项目中，团队的估算最多限于未来几周时间。
- 项目发起人通常想知道项目什么时候能够完成。一旦团队建立了稳定的开发速度或平均周期时间，团队就能够预测项目将花费多长时间。
- 敏捷历时估算可以分为：开发速度稳定前和开发速度稳定后两种

1. 开发速度稳定前的估算方法

- 开发速度稳定前，可以采用决策技术，如举手表决。
- 采用这种技术时，项目经理会让团队成员针对某个决定示意支持程度，举拳头表示不支持，伸五个手指表示完全支持，伸出三个以下手指的团队成员有机会与团队讨论其反对意见。
- 项目经理会不断进行举手表决，直到整个团队达成共识(所有人都伸出三个以上手指)或同意进入下一个决定。

2.开发速度稳定后的估算方法

- 团队通过观察历史表现来更准确地规划下阶段的能力，可能需要4~8个迭代才能达到稳定的速度。
- 根据自己在一个迭代中完成工作的能力(多少故事或故事点)来建立下一个迭代的能力衡量指标。
- 这样，产品负责人与团队一起重新规划，团队就更有可能在下一次迭代中成功交付。

软件项目进度计划 学习要点

MIMA

- 进度管理基本概念
- 任务资源估计
- 任务历时估算
- **进度计划编排**
- 项目进度模型
- 案例分析
- 课程实践

- 进度计划编制是决定项目开始和结束日期的活动。
- 主要目的是控制和节约项目的时间，保证项目在规定的时间内能够完成。
- 最终目标是建立一个现实的项目进度计划，为监控项目的进度进展情况提供一个基础。
- 项目进度计划过程反复多次，最后才能确定项目进度计划

进度编制的基本方法

- 关键路径法 CPM Critical Path Method 正推法/逆推法
- 时间压缩法 应急法/平行作业法
- 资源优化
- 敏捷项目进度编排

基本方法 → 关键路径法

- 关键路径法是根据指定的网络图逻辑关系进行的单一的历时估算。
- 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；计算网络图中最长的路径。以便确定项目完成时间估计。
- 关键路径是网络图中最长的路径
- 关键路径可以确定项目完成时间

基本方法 → 关键路径法

■ 进度编制相关术语：

- 最早开始时间(Early Start, ES) 表示一项任务（活动）的最早可以开始执行的时间
- 最晚开始时间(Late Start, LS) 表示一项任务（活动）的最晚可以开始执行的时间
- 最早完成时间(Early Finish, EF) 表示一项任务（活动）的最早可以完成的时间
- 最晚完成时间(Late Finish, LF) 表示一项任务（活动）的最晚可以完成的时间

基本方法 → 关键路径法

■ 进度编制相关术语:

- 超前(Lead) 表示两个任务的逻辑关系所允许的提前后置任务的时间
 - 滞后(Lag) 表示两个任务的逻辑关系所允许的推迟后置任务的时间
 - 浮动(Float) 表示一个任务的机动性, 是其在不影响项目完成的情况下可以推迟的时间量
 - 自由浮动(Free Float, FF) 在不影响后置任务最早开始时间, 本任务可以延迟的时间
 - 总浮动(Total Float, TF) 表示在不影响项目最早完成时间, 本任务可以延迟的时间
- $$FF = ES(\text{后置任务}) - EF - lag \quad TF = LS - ES \quad / \quad TF = LF - EF$$

基本方法 → 关键路径法

■ 进度编制相关术语：

➤ 关键路径 (Critical Path)

- 时间浮动为0 (Float=0) 的路径 $TF=LS-ES$ / $TF=LF-EF$
- 网络图中最长的路径
- 关键路径是决定项目完成的最短时间
- 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
- 关键路径可能不止一条

基本方法 → 关键路径法 → 例

MIMA



$ES(A)=LS(A)=0$
 $EF(A)=LF(A)=100$

$ES(B)=0$ $LS(B)=90$
 $EF(B)=10$ $LF(B)=100$

$TF(A)=LS(A)-ES(A)=LF(A)-EF(A)=0$
 $TF(B)=LS(B)-ES(B)=LF(B)-EF(B)=90$

基本方法 → 关键路径法 → 例

MIMA



$ES(C)=0$ $ES(D)=15$
 $EF(C)=10$ $EF(D)=20$

$FF(C)=ES(D)-EF(C)-lag=0$

$LF(D)=100$ $LS(D)=95$
 $LF(C)=90$ $LS(C)=80$

基本方法 → 关键路径法

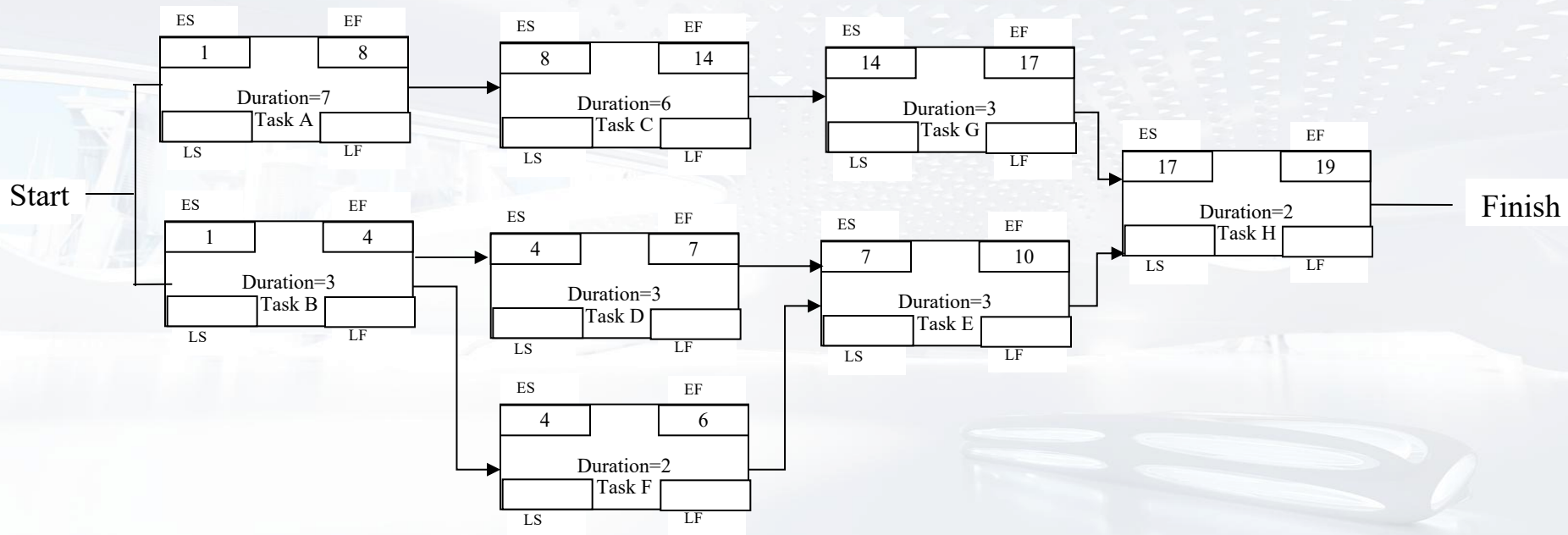
■ 任务图示

ES 最早 开始时间		EF 最早 完成时间
任务名称:工期 Duration Task		
LS 最晚 开始时间		LF 最晚 完成时间

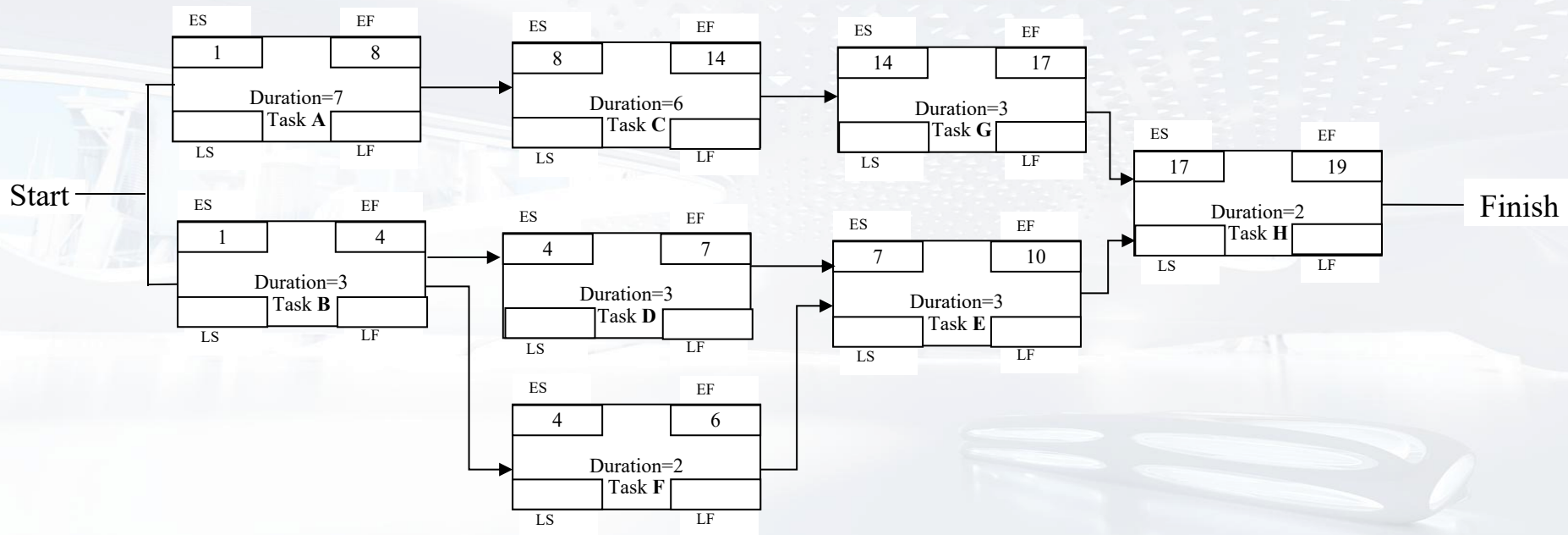
基本方法 → 关键路径法 → 正推法

- **正推法**：按照时间顺序计算各个任务（活动）的最早开始时间和最早完成时间的方法
- 过程如下
 - 确定项目的开始时间
 - 从左到右，从上到下进行任务编排
 - 计算每个任务的最早开始时间ES和最早完成时间EF：
 - 网络图中第一个任务的最早开始时间是项目的开始时间
 - $ES + Duration = EF$
 - $EF + Lag = ES(s)$
 - 当一个任务有多个前置任务时，选择前置任务中最大的EF加上Lag作为其ES

基本方法 → 关键路径法 → 正推法



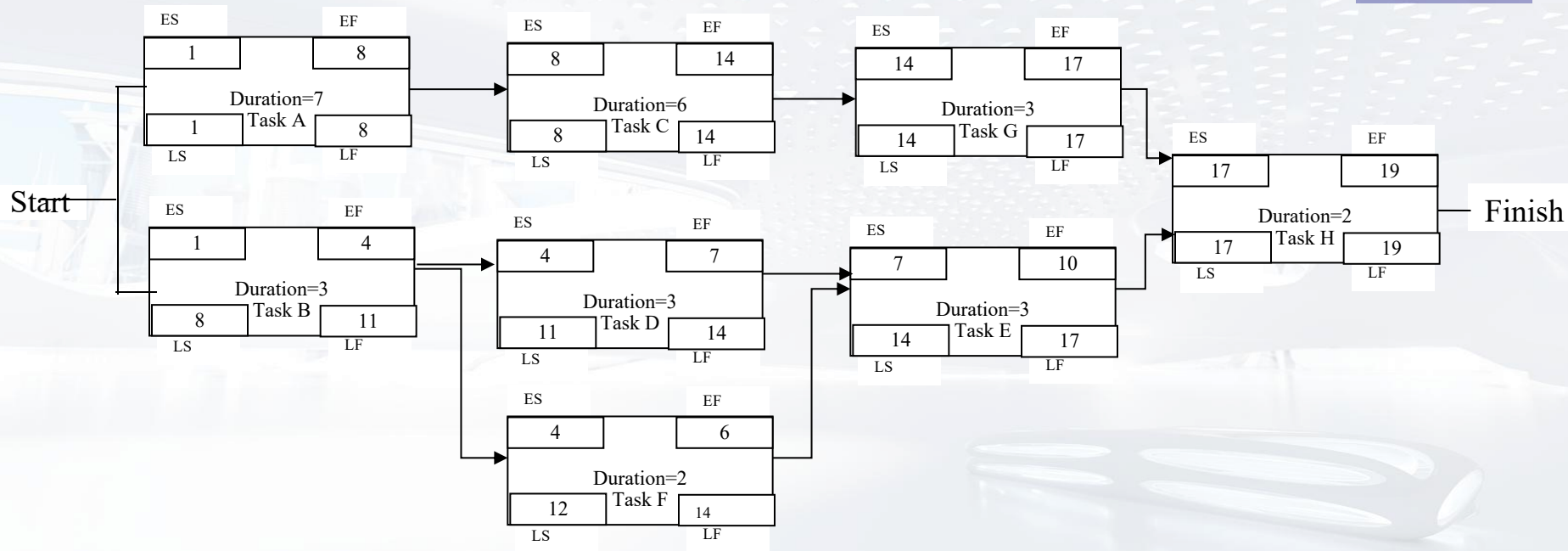
基本方法 → 关键路径法 → 正推法



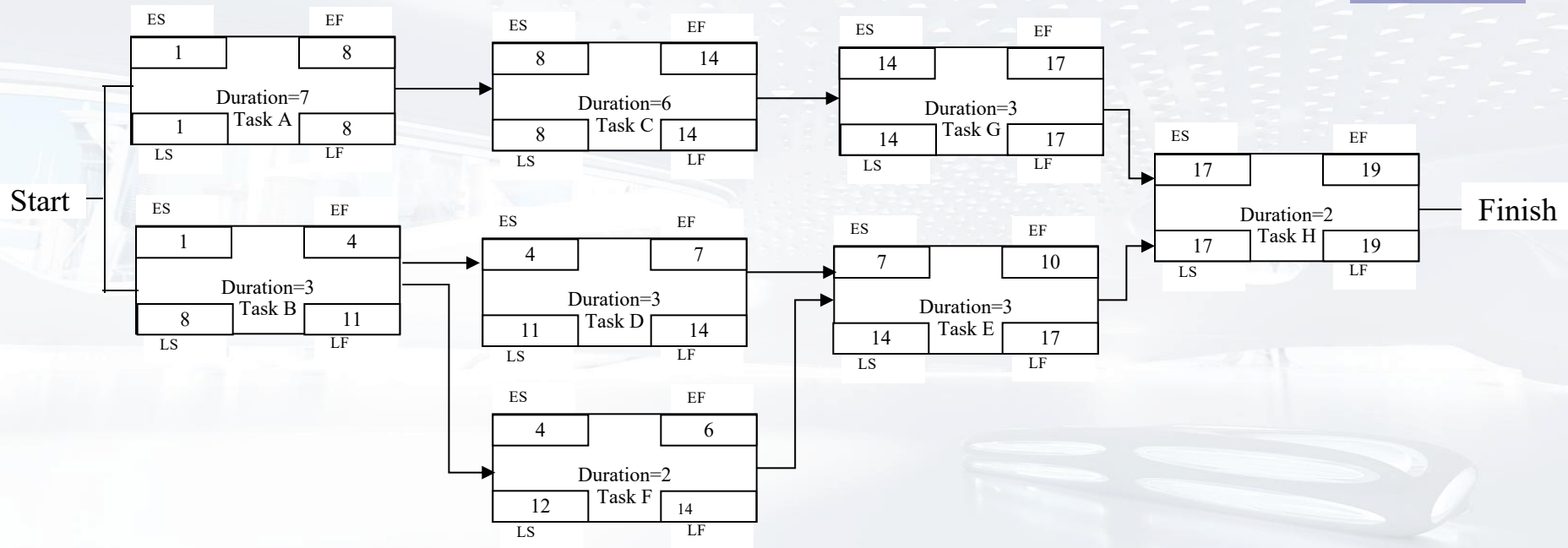
基本方法 → 关键路径法 → 逆推法

- **逆推法**：按照时间顺序计算各个任务（活动）的最晚开始时间和最晚完成时间的方法
- 过程如下
 - 确定项目的结束时间
 - 从右到左，从上到下进行任务编排
 - 计算每个任务的最晚开始时间LS和最晚完成时间LF：
 - 网络图中最后一个任务的最晚完成时间是项目的结束时间
 - $LF - Duration = LS$
 - $LS - Lag = LF(p)$
 - 当一个任务有多个后置任务时，选择其后置任务中最小LS减Lag作为其LF

基本方法 → 关键路径法 → 逆推法

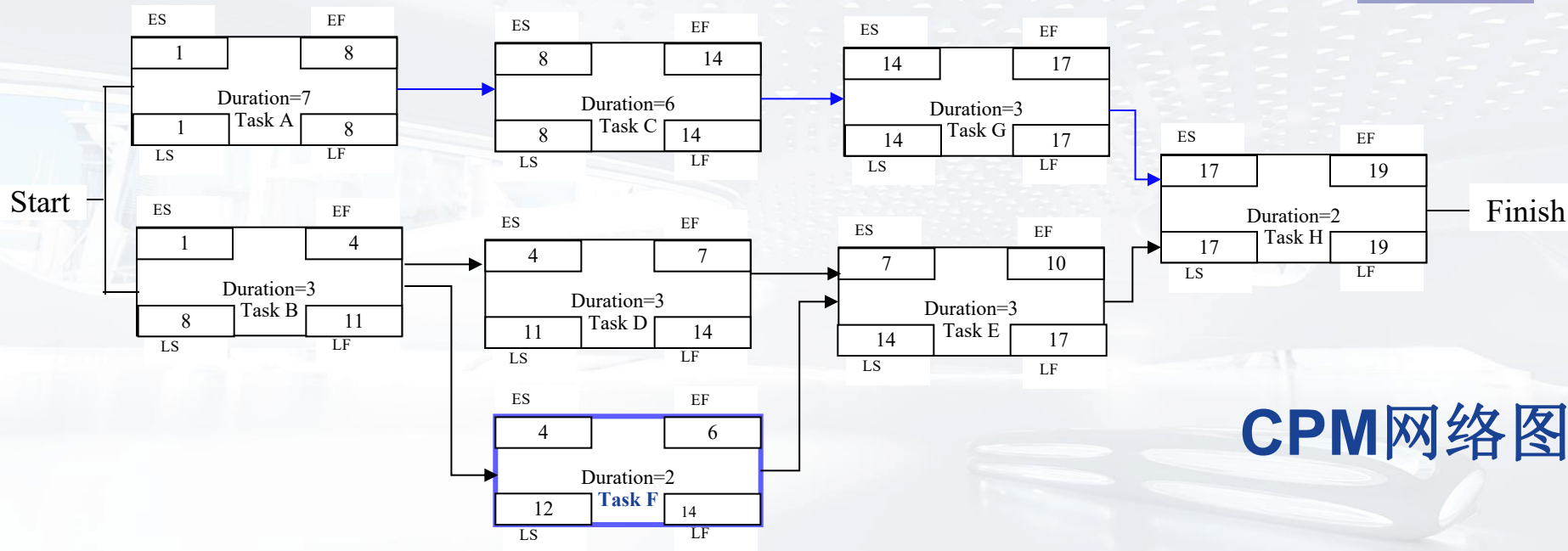


基本方法 → 关键路径法 → 逆推法



基本方法 → 关键路径法 → 逆推法

MIMA



$$FF(F) = ES(E) - EF(F) - lag = 7 - 6 - 0 = 1$$

$$TF(F) = LS(F) - ES(F) = LF(F) - EF(F) = 8$$

关键路径: A → C → G → H

Cp Path: 19

- 进度管理基本概念
- 任务资源估计
- 任务历时估算
- 进度计划编排
- 项目进度模型
- 案例分析
- 课程实践

感谢！



软件学院 罗昕
luoxin@sdu.edu.cn

