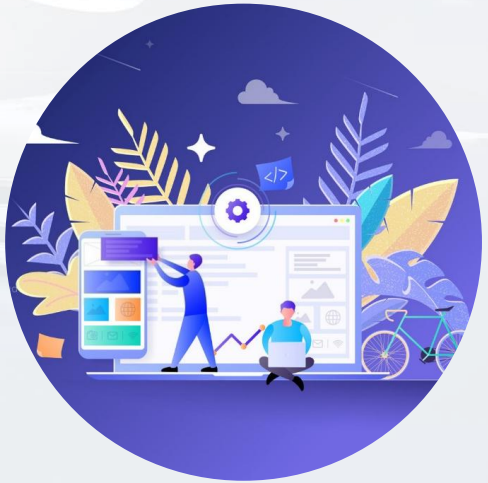
第二篇第7章

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

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**
- 软件项目进度计划确定
- 案例分析
- 课程实践

进度编制的基本方法

MIMA

- 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 资源平衡方法/资源平滑
- 敏捷项目进度编排

时间压缩法

MIMA

- 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
- 是一种数学分析方法

- 应急法 -- 赶工 (Crash)
  - 时间成本平衡方法
  - 进度压缩因子方法
- 平行作业法 -- 快速跟进 (Fast Tracking)

时间压缩法 → 应急法

MIMA

- **时间成本平衡方法**是一种线性关系方法，进度压缩与成本的增长是成比例的。也称为进度压缩单位成本方法

- 基于下面的假设：
  - 每个任务存在一个正常进度和可压缩进度、一个正常成本和可压缩成本。
  - 通过增加资源，每个任务的历时可以从正常进度压缩到可压缩进度。
  - 每个任务无法在低于可压缩进度内完成。
  - 有足够需要的资源可以利用。
  - 在“正常”与“可压缩”之间，进度压缩与成本的增长是成正比的。

$$\text{单位进度压缩的成本} = \frac{\text{可压压缩成} - \text{正常成本}}{\text{正常进常} - \text{可压压缩进}}$$

时间压缩法 → 应急法

MIMA

- 前提：活动的正常与压缩
  - 项目活动的正常值
    - 正常进度
    - 正常成本
  - 每个任务无法在低于可压缩进度内完成
  - $$\text{单位进度压缩的成本} = \frac{\text{可压缩成本} - \text{正常成本}}{\text{正常进度} - \text{可压缩进度}}$$

通过增加资源，每个任务可以从“正常”到“可压缩”

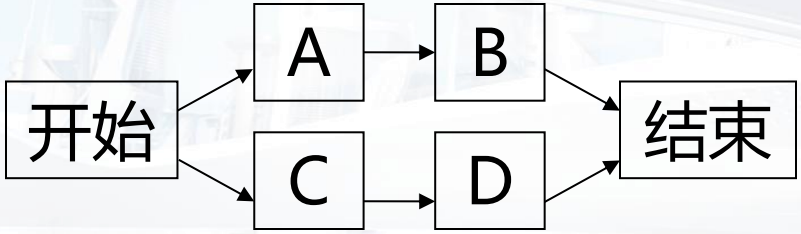
- 项目活动的可压缩值
  - 可压缩进度
  - 可压缩成本

- 线性关系，所以可以通过计算任务的单位进度压缩的成本来计算在压缩范围之内的进度压缩产生的压缩费用。

时间压缩法 → 应急法

MIMA

- 时间成本平衡方法 - 例



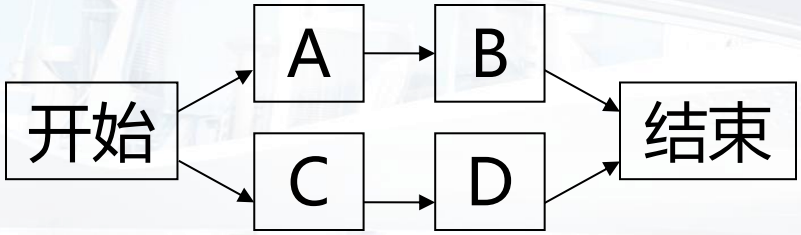
	A	B	C	D
正常进度	7周	9周	10周	8周
正常成本	5万元	8万元	4万元	3万元
可压缩进度	5周	6周	9周	6周
可压缩成本	6.2万元	11万元	4.5万元	4.2万元

问题：当前总工期是18周，若压缩到**17周、16周、15周**并保证每个任务在可压缩的范围内，应压缩哪些任务，并给出可压缩之后的总成本。

时间压缩法 → 应急法

MIMA

- 时间成本平衡方法 - 例



	A	B	C	D
正常进度	7周	9周	10周	8周
正常成本	5万元	8万元	4万元	3万元
可压缩进度	5周	6周	9周	6周
可压缩成本	6.2万元	11万元	4.5万元	4.2万元

开始→A→B→结束
Path:16周

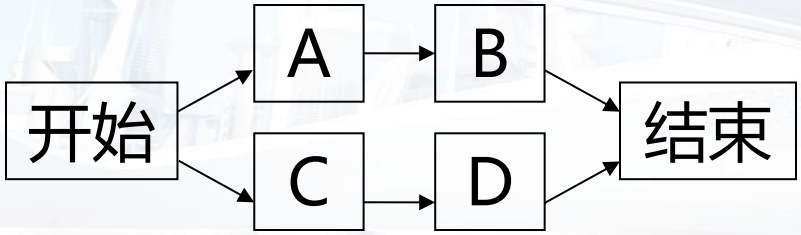
开始→C→D→结束
Path:18周
CP Path

总成本
5+8+4+3
20万元

时间压缩法 → 应急法

MIMA

■ 时间成本平衡方法 - 例



	A	B	C	D
正常进度	7周	9周	10周	8周
正常成本	5万元	8万元	4万元	3万元
可压缩进度	5周	6周	9周	6周
可压缩成本	6.2万元	11万元	4.5万元	4.2万元

$$\text{单位进度压缩的成本} = \frac{\text{可压缩成本} - \text{正常成本}}{\text{正常进度} - \text{可压缩进度}}$$

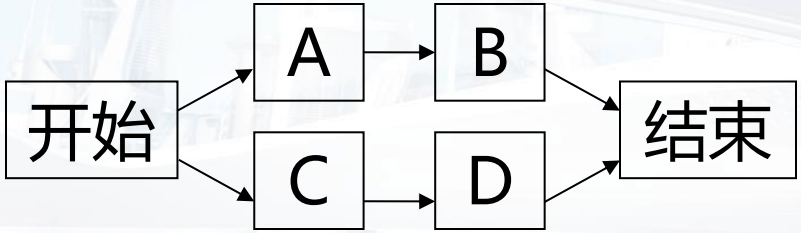
任务单位压缩成本	A	B	C	D
压缩成本(万/周)	0.6	1	0.5	0.6

时间压缩法 → 应急法

MIMA

- 时间成本平衡方法 - 例

压缩到17周



	A	B	C	D
正常进度	7周	9周	10周	8周
正常成本	5万元	8万元	4万元	3万元
可压缩进度	5周	6周	9周	6周
可压缩成本	6.2万元	11万元	4.5万元	4.2万元

开始→C→D→结束 Path:17周

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

原总成本+压缩成本

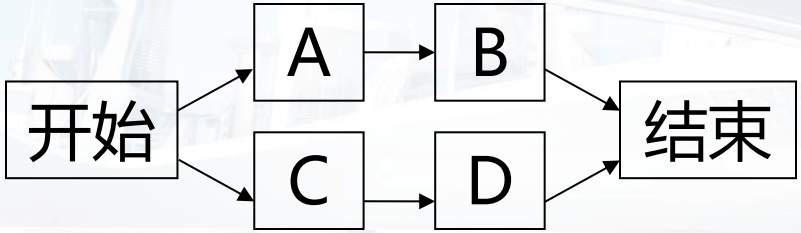
20+0.5

时间压缩法 → 应急法

MIMA

- 时间成本平衡方法 - 例

压缩到16周



	A	B	C	D
正常进度	7周	9周	10周	8周
正常成本	5万元	8万元	4万元	3万元
可压缩进度	5周	6周	9周	6周
可压缩成本	6.2万元	11万元	4.5万元	4.2万元

开始→C→D→结束 Path:16周

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

原总成本+压缩成本

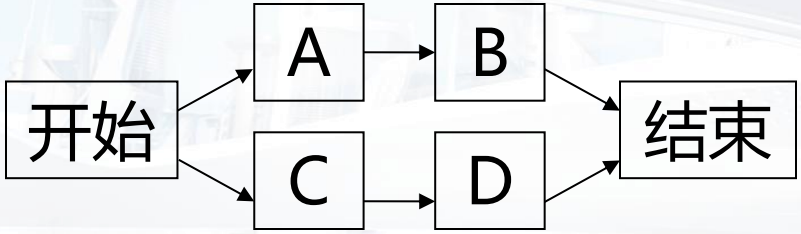
20+0.5+0.6

时间压缩法 → 应急法

MIMA

- 时间成本平衡方法 - 例

压缩到15周



	A	B	C	D
正常进度	7周	9周	10周	8周
正常成本	5万元	8万元	4万元	3万元
可压缩进度	5周	6周	9周	6周
可压缩成本	6.2万元	11万元	4.5万元	4.2万元

开始→A→B→结束 Path:15周

开始→C→D→结束 Path:15周

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

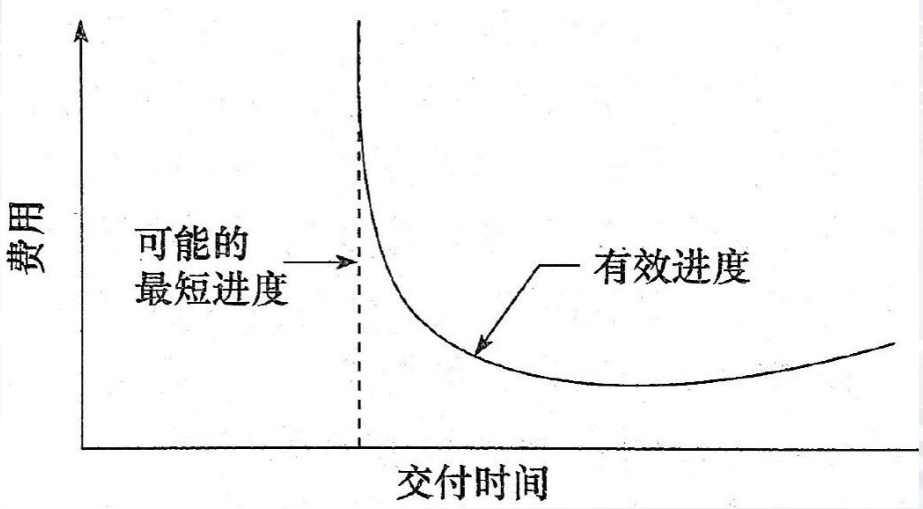
原总成本+压缩成本 C, D*2, A

20+0.5+0.6+0.6+0.6

时间压缩法 → 应急法

MIMA

- 实际中，**进度压缩与费用的上涨不是总能呈现正比关系**，如图所示



- 压缩进度因子方法 Charles Symons(1991)方法

时间压缩法 → 应急法

MIMA

- **压缩进度因子方法**
- 公式

$$\text{进度压缩因子} = \frac{\text{压缩进度}}{\text{正常进常}} \qquad \text{压缩进度的工作量} = \frac{\text{估算工作量}}{\text{进度压缩因子}}$$

- 例：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：进度压缩因子应大于等于0.75，最多可以压缩25%

时间压缩法

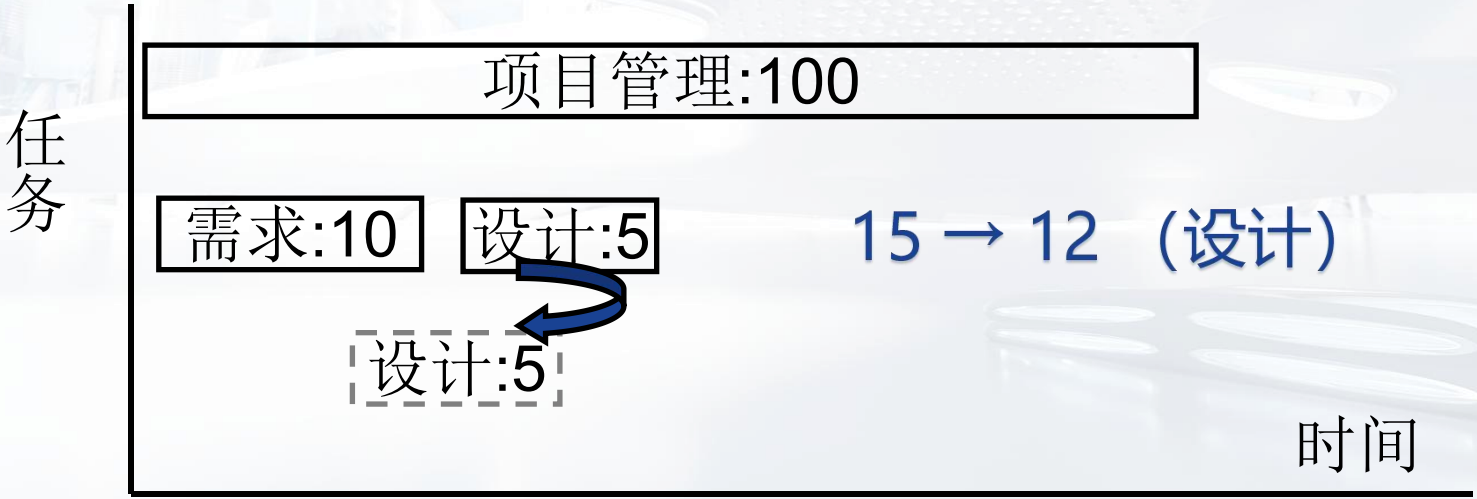
MIMA

- 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
- 是一种数学分析方法

- 应急法 -- 赶工 (Crash)
  - 时间成本平衡方法
  - 进度压缩因子方法
- 平行作业法 -- 快速跟进 (Fast Tracking)

时间压缩法 → 平行作业法

- 又称为快速跟进 (Fast Tracking)法，是平行地做活动



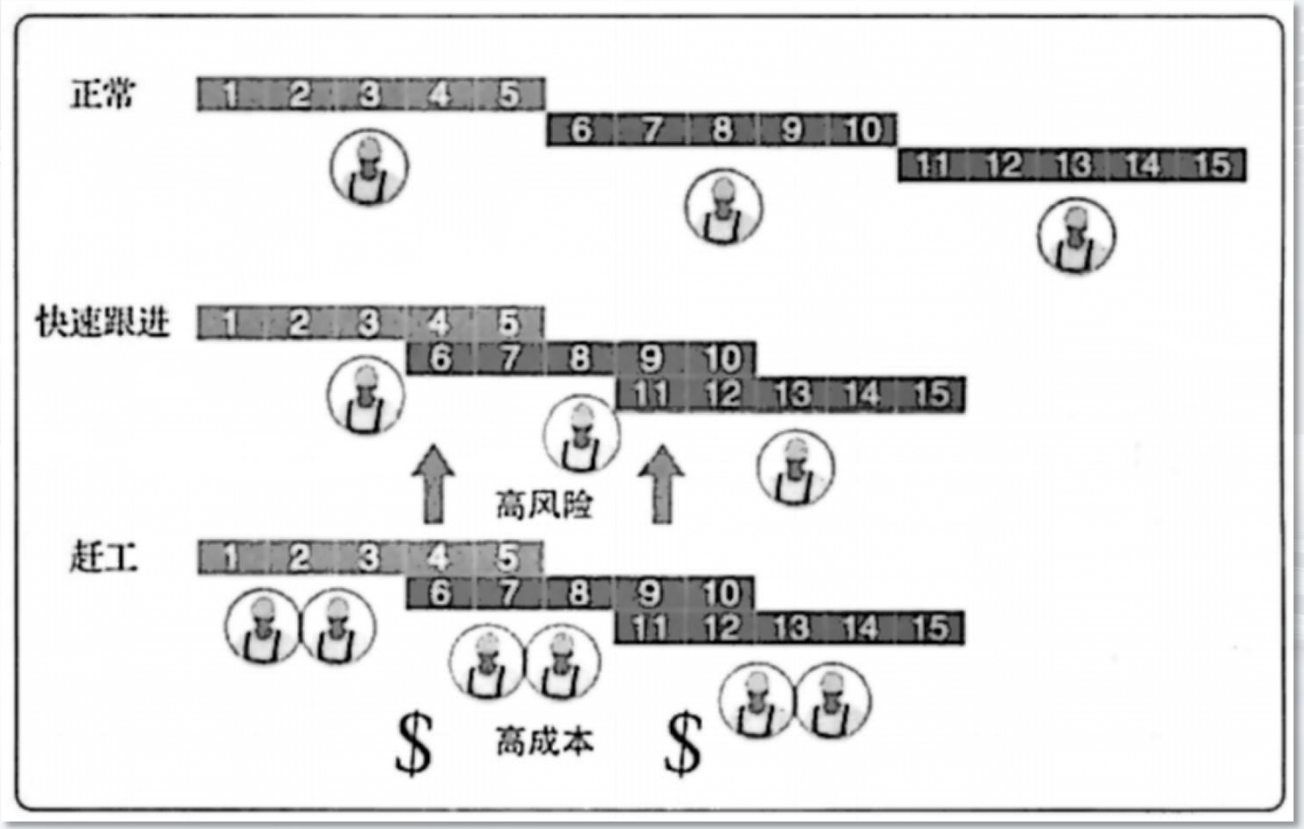
时间压缩法 应急法/平行作业法

MIMA

- **应急法不改变活动间的逻辑关系。**应急(赶工)法是一种通过增加资源，以最小的成本代价来压缩进度工期的技术。赶工只适用于那些通过增加资源就能缩短持续时间的且位于关键路径上的活动。但赶工并非总是切实可行的，因它可能导致风险或成本的增加。

- **平行作业法则可改变活动间的逻辑关系，并行开展某些活动。**平行作业法(快速跟进)将正常情况下按顺序进行的活动或阶段改为至少是部分并行开展的。平行作业法常导致返工和增加风险。使用提前量通常会增加相关活动之间的协调工作，并增加质量风险，快速跟进还有可能增加项目成本。

时间压缩法　应急法/平行作业法



进度编制的基本方法

MIMA

- 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 资源平衡方法/资源平滑
- 敏捷项目进度编排

资源优化

MIMA

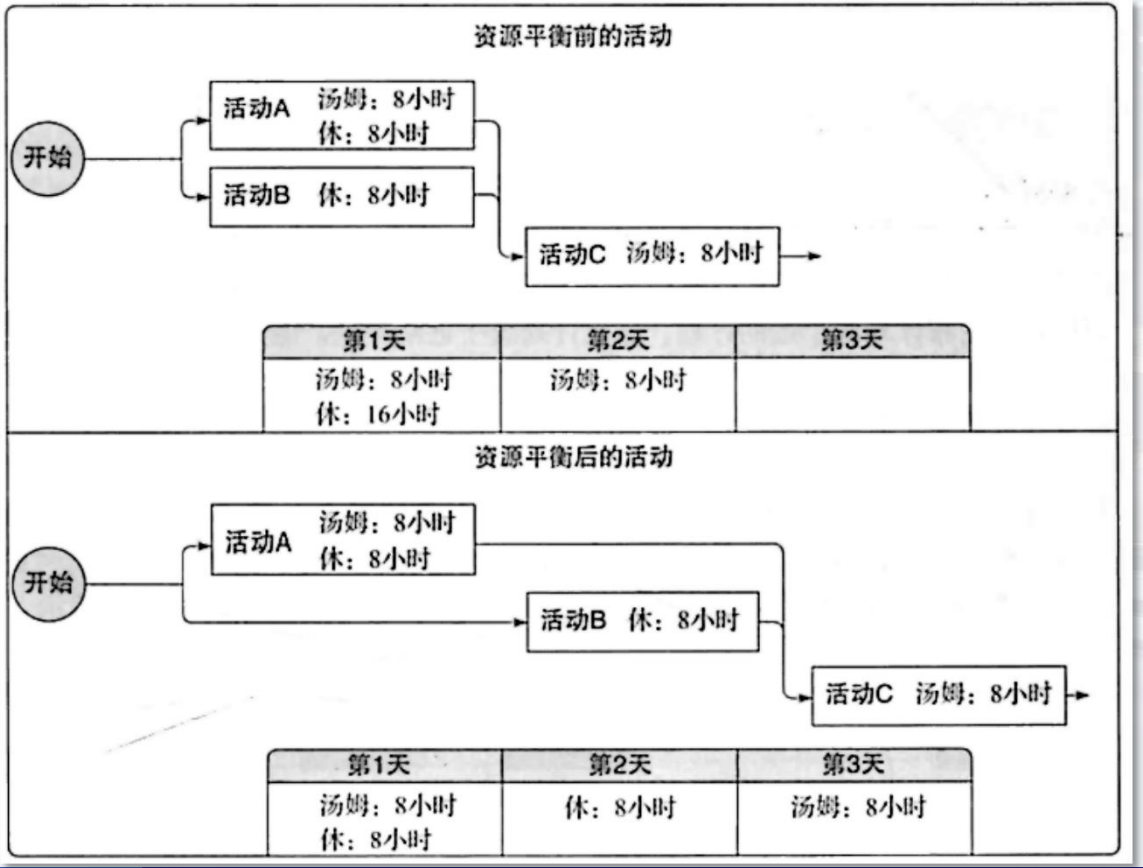
- 资源优化用于调整活动的开始日期和完成日期，以调整计划使用的资源，使其等于或少于可用的资源。
- 根据资源供需情况来调整进度模型
- 资源平衡和资源平滑都属于资源优化方法

资源平衡法

MIMA

- 资源平衡方法是为了在资源需求与资源供给之间取得平衡，通过调整任务的时间来协调资源的冲突，根据资源制约因素对开始日期和完成日期进行调整的一种技术。
- 主要目的是形成平稳连续的资源需求，最有效地利用资源，使资源闲置的时间最小化，同时尽量避免超出资源能力。
- 如果共享资源或关键资源只在特定时间可用，数量有限，或被过度分配，如一个资源在同一时段内被分配至两个或多个活动，就需要进行资源平衡。

资源平衡法 → 例


资源平衡前的活动
开始
活动A 汤姆：8小时 休：8小时
活动B 休：8小时
活动C 汤姆：8小时
第1天 第2天 第3天
汤姆：8小时 休：16小时
汤姆：8小时
资源平衡后的活动
开始
活动A 汤姆：8小时 休：8小时
活动B 休：8小时
活动C 汤姆：8小时
第1天 第2天 第3天
汤姆：8小时 休：8小时
休：8小时
汤姆：8小时

资源平衡法

MIMA

- **资源平衡往往导致关键路径改变**。关键路径法通常可以产生一个初始进度计划，而实施这个计划需要的资源可能比实际拥有的多。
- 在项目编排中进行资源的优化配置，可保证资源最优化、最有效。
- 利用资源平衡法可在资源有约束的条件下制定一个进度计划。

资源平滑

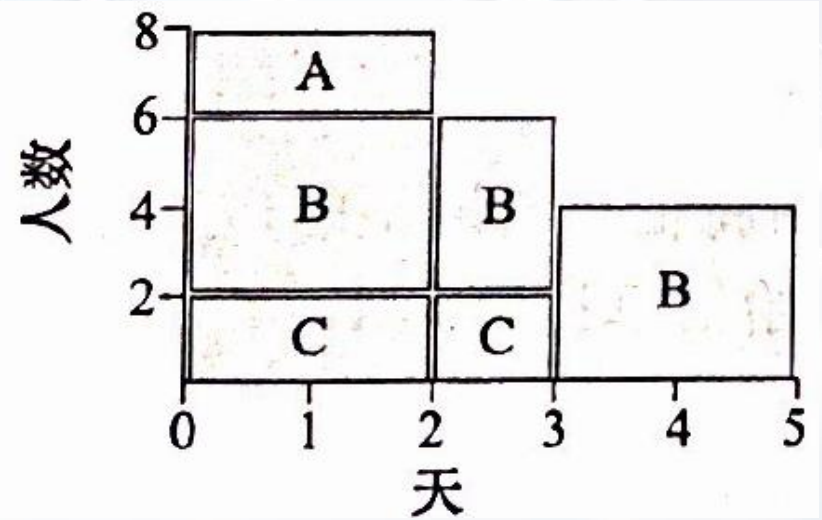
MIMA

- 资源平滑方法是对进度模型中的活动进行调整，从而使项目资源需求不超过预定的资源限制的一种技术。
- 相对于资源平衡而言，资源平滑不会改变项目关键路径，完工日期也不会延迟。
- 活动只在其自由和总浮动时间内延迟，但资源平滑技术可能无法实现所有资源的优化。

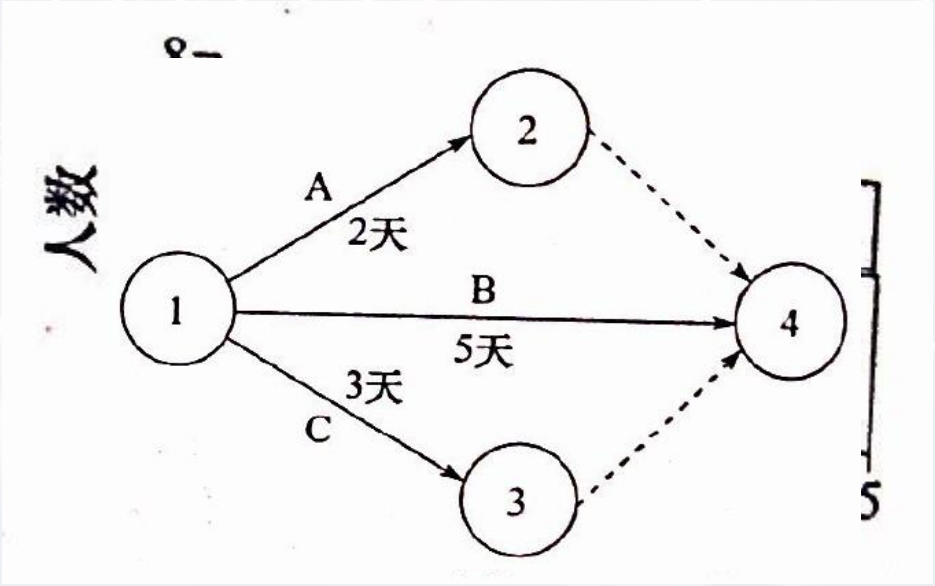
资源平滑 → 例

MIMA

- 如图，A任务需要2人2天，B需要4人5天，C需要2人3天。



3个任务同时开始



3个任务不同时开始

进度编制的基本方法

MIMA

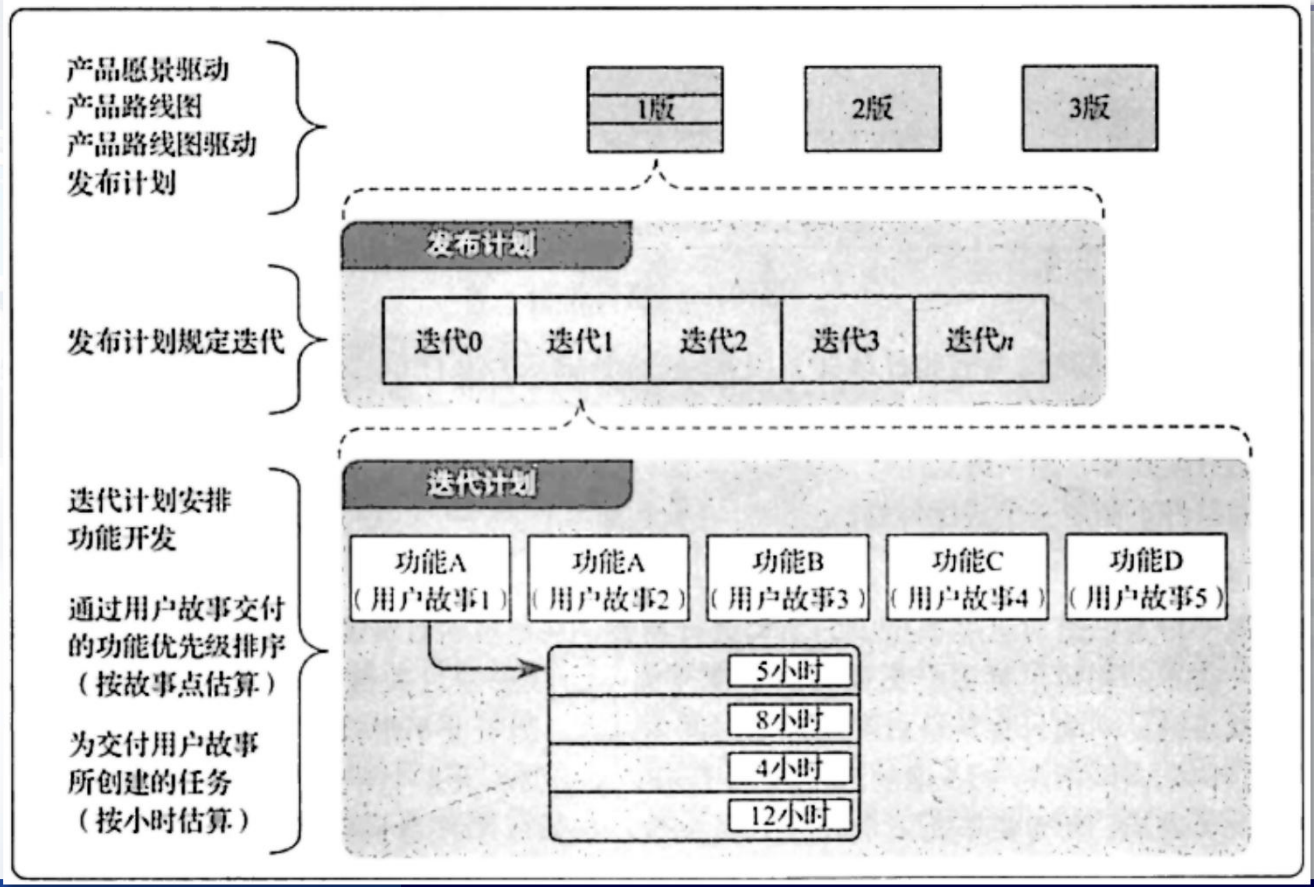
- 关键路径法 CPM Critical Path Method 正推法/逆推法
- 时间压缩法 应急法/平行作业法
- 资源优化 资源平衡方法/资源平滑
- 敏捷项目进度编排

敏捷项目进度编排

MIMA

- 敏捷项目一般采取远粗近细的计划模式
- 发布计划（远期计划）+迭代计划（近期计划）

敏捷项目进度编排

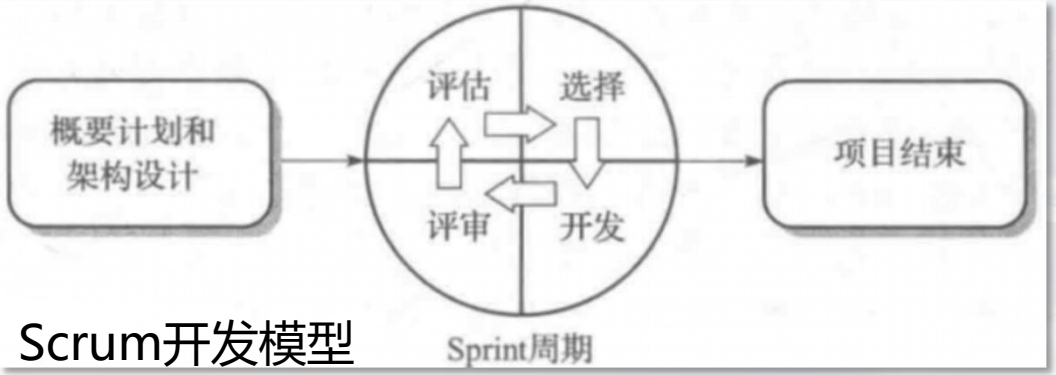


发布计划
迭代计划

敏捷项目进度编排

MIMA

- Scrum开发模型(典型的敏捷模型)



Scrum开发模型

- Scrum模型具有典型的两层项目计划，基于远粗近细的原则和项目渐进明细的特点。
- 将概要的项目整体规划和详细的近期迭代计划有机结合，有效提高计划的准确度、资源管理能力和项目的按时交付能力。

敏捷项目进度编排

MIMA

- Scrum模型是迭代式增量软件开发过程，通常用于敏捷软件开发

- 两层计划 — **远粗近细 渐进明细** 的特点
  - Product Backlog　产品待办事项
  - Sprint Backlog　Sprint待办事项

敏捷项目进度编排

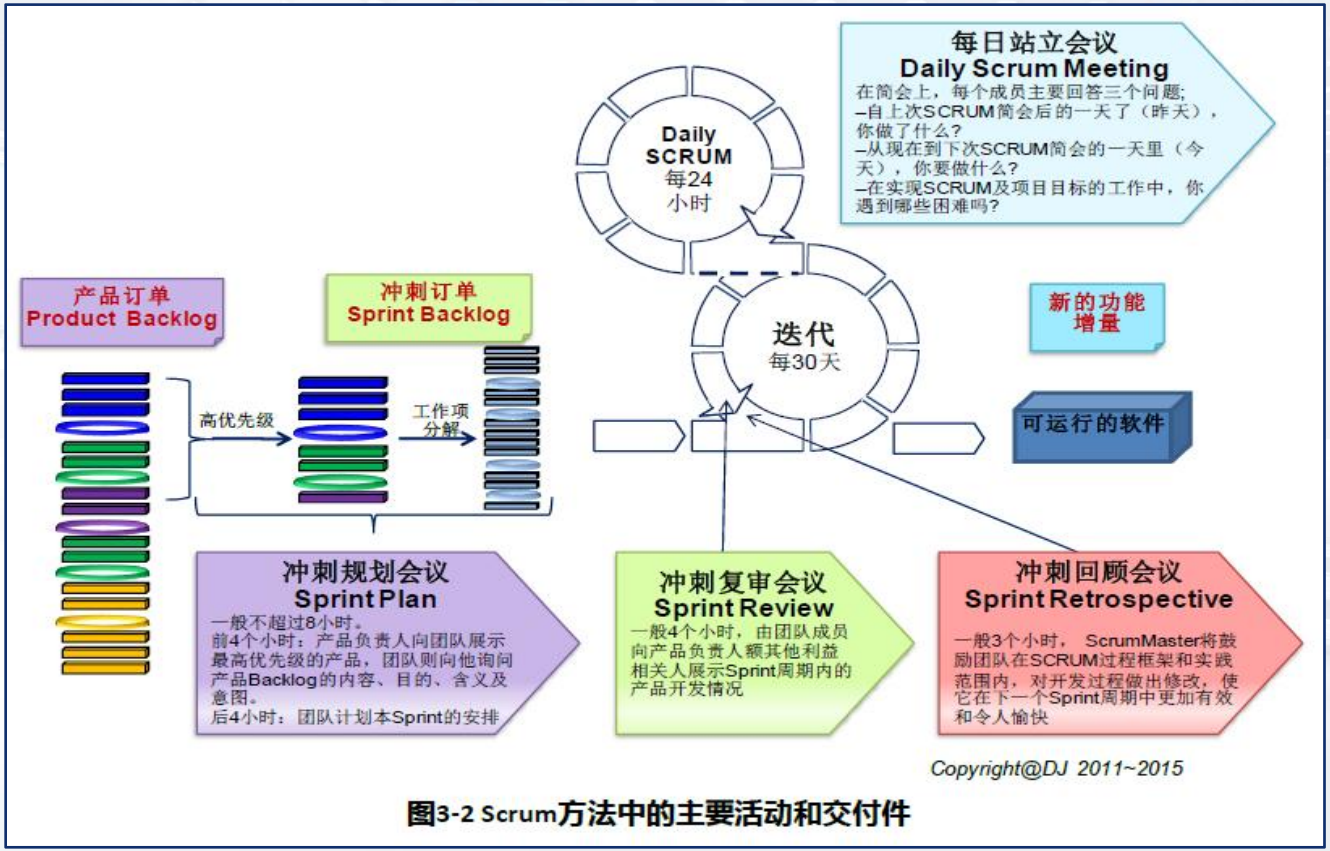


图3-2 Scrum方法中的主要活动和交付件

敏捷项目进度编排

MIMA

- 两层计划
- Product Backlog　产品待办事项
  - 产品待办事项列表存在于产品的整个生命周期，它是产品的路线图。
  - 任何时候，产品待办事项列表都是团队依照优先排列顺序完成工作的唯一、最终的概括。
  - 一个产品只有一个产品待 办事项列表，这意味着产品负责人必须纵观全局做出优先级排列的决策，以体现利益相关人（包括团队）的意愿。
- Sprint Backlog　Sprint待办事项

敏捷计划

MIMA

产品待办事项列表 Product Backlog

				每个Sprint的新估算					
优先级	事项	细节（wiki链接）	初始规模估算	1	2	3	4	5	6
1	作为买家，我想把书放入购物车（见wiki页面用户界面草图）	...	5						
2	作为买家，我想从购物车中删除书	...	2						
3	提高交易处理性能（见wiki页面目标性能指标）	...	13						
4	探讨加速信用卡验证的解决方法（见wiki页面目标性能指标）	...	20						
5	将所有服务器升级到Apache 2.2.3	...	13						
6	分析并修复订单处理脚本错误（错误号：14834）	...	3						
7	作为购物者，我想创建并保存愿望表	...	40						
8	作为购物者，我想增加或删除愿望表中的条目	...	20						

敏捷项目进度编排

MIMA

- 两层计划
- Product Backlog　产品待办事项

- Sprint Backlog　Sprint待办事项
  - 设定了 Sprint目标并挑选出这个Sprint要完成的产品待办列表项之后，开发团队将决定如何在Sprint中把这些功能构建成“完成”的产品增量。
  - 这个Sprint中所选出的产品待办列表项以及交付它们的计划统称为Sprint待办事项列表。

敏捷项目进度编排

优先级	事项
1	作为买家，我想把书放入购物车（见wiki页面用户界面草图）
2	作为买家，我想从购物车中删除书

Sprint待办事项 Sprint Backlog

				每日结束时所剩余工作量的最新估计					
产品待办事项列表事项	Sprint中的任务	志愿者	初始工作量估计	1	2	3	4	5	6
做为买家，我希望把书放到购物车中	修改数据库		5						
	创建网页(UI)		8						
	创建网页 (Javascript逻辑)		13						
	写自动化验收测试		13						
	更新买家帮助网页		3						
	. . .								
改进事务处理效率	合并DCP代码并完成分层测试		5						
	完成pRank的机器顺序		8						
	把DCP和读入器改为用pRank http API		13						

敏捷项目进度编排

MIMA

- 开发团队通常先由系统设计开始，设计把产品待办事项列表转换成可工作的产品增量所需要的工作。
- 在Sprint计划会议中，开发团队挑选出足够的工作量。
- 开发团队自发组织地领取Sprint待办事项列表中的工作，领取工作在Sprint计划会议和Sprint期间按实际情况进行。

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
- **软件项目进度计划确定**
- 案例分析
- 课程实践

软件项目进度问题(SPSP)模型

MIMA

- 软件项目进度问题（Software Project Scheduling Problem，SPSP）模型是在给定的项目任务工作量及其关系和资源限制下，对项目确定合适的人员安排，以保证项目的时间最短、成本最小。

- 最主要的资源是人力资源。人员需要具备项目需要的相应技能。

  - 项目需要的技能
  - 项目中的任务
  - 项目中的人员

软件项目进度问题(SPSP)模型

MIMA

- 技能

$$S = \{s_1, s_2, \cdots, s_n\}$$

- 活动

$$T = \{t_1, t_2, \cdots, t_m\} \quad (t_i, t_j) \in E$$

t_j^{sk} 该任务需要的技能

t_j^{eff} 该任务的工作量

- 人员

$$E = \{e_1, e_2, \cdots, e_e\}$$

e_i^{sk} 该人员具备的技能 $e_i^{sk} \subseteq S$

$e_i^{\max d}$ 该人员最大贡献度 $e_i^{\max d} \in [0,1]$

e_i^{rem} 人力成本

SPSP 模型 → 项目需要的技能

MIMA

- 人员技能(skills of employees)是完成项目需要的能力，项目人员需要具备这些技能。
- 例如，软件项目相关的技能有软件分析技能、软件设计技能、软件编程技能、领导技能等。
- 我们将这些技能定义为一个集合

$$S = \{s_1, s_2, \cdots, s_n\}$$

其中n是技能集合中的总技能数量。

SPSP 模型 → 项目中的任务

MIMA

- 任务是完成软件项目需要的所有活动，如分析、设计、编码、测试、写文档等活动。

- 软件项目由一系列这些软件活动组成，这些活动之间具有一定的关联关系。可以使用任务关系图示 *G(T, E)*，即PDM网络图表示，这个图示是单向的，*T* 代表有关联关系的一些活动，*E* 代表活动之间的关联关系，其中 $T=\{t_1,t_2,\cdots,t_m\}$ $(t_i,t_j)\in E$ 表示t_i是t_j的前置活动。

- 每个任务t_j有两个属性，一个属性是这个任务需要具备的技能 t_j^{sk} ，一个属性是这个任务的工作量 t_j^{eff}

SPSP 模型 → 项目中的人员

MIMA

- 软件项目中的人员是软件工程师，他们具备的技能是软件工程技能。
- 通常，每个软件项目 需要由具备一定软件工程技能的人员来完成项目。项目经理安排合适的人完成合适的任务。这样，我们就可以得到合适的任务进度安排。
- 项目中的人员可以定义为 $E = \{e_1, e_2, \cdots, e_e\}$
- 每个人员有3个属性，第一个属性是这个人员的技能集合 e_i^{sk} 并且 $e_i^{sk} \subseteq S$ 第二个属性是这个人员对项目的最大贡献度 $e_i^{\max d}$ 并且 $e_i^{\max d} \in [0,1]$ 如果 <1,表示这个人员是兼职的；第三个属性是人力成本 e_i^{rem}

SPSP 模型 → SPSP模型解决方案

MIMA

- 要想得到软件项目进度问题(SPSP)模型的解决方案，需要针对现有的项目任务、项目需要的技能、现有的人员、每个人员具备的技能、每个人员的成本，做到最合适的人员任务安排，以求达到项目进度最短、总成本最小。

- 所以，最终的解决方案可以表达为一个矩阵（维度是人员数与任务数的乘积）

$$\begin{vmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ m_{m1} & m_{m2} & \cdots & m_{mn} \end{vmatrix}$$

$$M = |E \times T|$$

SPSP 模型 → SPSP模型解决方案

MIMA

$$M = |E \times T|$$

$$\begin{vmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ m_{m1} & m_{m2} & \cdots & m_{mn} \end{vmatrix}$$

$\mathbf{m}_{ij} \in [0,1]$

第i个人对第j个任务的付出

$\mathbf{m}_{ij} = 0.25$

表示该人员花 25%的时间在此任务上

人员对任务的付出矩阵： 人员数乘以任务数

SPSP 模型 → SPSP模型解决方案

MIMA

- 通过贡献度矩阵可以将人员合理地安排到项目中，从而得到最短项目进度和最小项目成本，进而得到项目的进度安排计划。

- 为了得到最优的矩阵，需要满足限制条件：
  - 1.贡献度矩阵限制条件
  - 2.最优函数限制条件
  - 3.组合优化问题
  - 4.最佳方案确定

SPSP模型解决方案 - 1.贡献度矩阵限制条件

MIMA

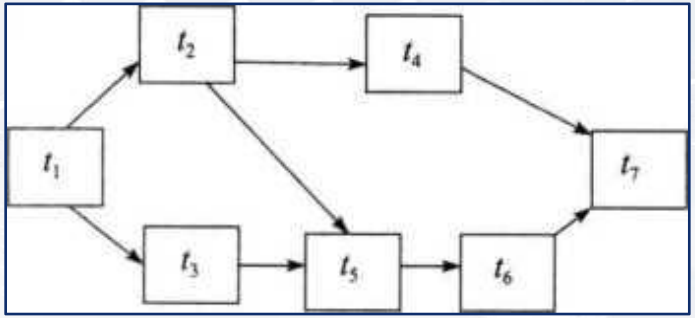
- 在人员项目贡献度矩阵中，$M = |E \times T|$ $\sum_{i=1}^{m} m_{ij} > 0$
- 即至少有一人分配到一个任务中，并且分配到项目中的人员应该具备相应的技能。
- 下式就满足这个限制条件。

$$\begin{vmatrix} 0.5 & 1 & 1 & 0 \\ 1 & 0 & 0.5 & 1 \end{vmatrix}$$

SPSP模型解决方案 - 2.最优函数限制条件

MIMA

- (1)构建项目网络图
- (2)人员模型
- (3)任务历时
- (4)项目时间
- (5)项目成本
- (6)最佳函数



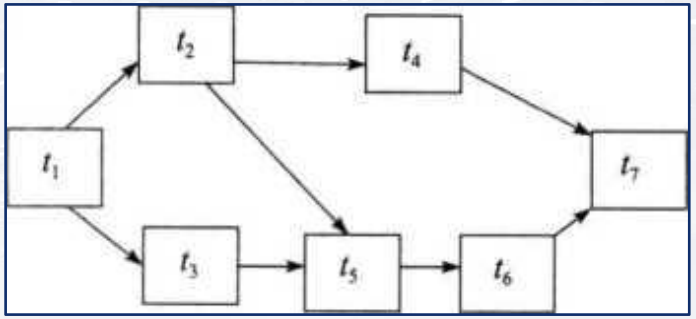
$$f(x) = W^{\cos} p^{\cos} + W^{\text{len}} p^{\text{len}}$$

$$p^{\text{overw}} = \sum_{i=1}^{|e|} e_i^{\text{overw}}$$

SPSP模型解决方案 - 2.最优函数限制条件

MIMA

- (1)构建项目网络图
- (2)人员模型
- (3)任务历时
- (4)项目时间
- (5)项目成本
- (6)最佳函数



$$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$$

$$S = \{s_1, s_2, s_3\}$$

$$t_1^{sk} = \{s_1, s_2, s_3\}, t_2^{sk} = \{s_2\}, t_3^{sk} = \{s_1, s_3\}, t_4^{sk} = \{s_1\}$$

$$t_5^{sk} = \{s_1, s_2, s_3\}, t_6^{sk} = \{s_1, s_2\}, t_7^{sk} = \{s_1\}$$

$$t_1^{eff} = 4, t_2^{eff} = 6, t_3^{eff} = 8, t_4^{eff} = 9, t_5^{eff} = 8, t_6^{eff} = 10, t_7^{eff} = 16$$

SPSP模型解决方案 - 2.最优函数限制条件

MIMA

- (1)构建项目网络图
- (2)人员模型
- (3)任务历时
- (4)项目时间
- (5)项目成本
- (6)最佳函数

项目人员共计4人，集合是 $E = \{e_1, e_2, e_3, e_4\}$

每个人员具备的技能和人力成本如下：

$$e_1^{sk} = \{s_1, s_2, s_3\}\ e_2^{sk} = \{s_1, s_2, s_3\}, e_3^{sk} = \{s_1, s_2\}\ e_4^{sk} = \{s_1, s_3\}$$

$$e_1^{rem} = \$6\,000, e_2^{rem} = \$6\,000, e_3^{rem} = \$5\,000, e_4^{rem} = \$5\,000$$

SPSP模型解决方案 - 2.最优函数限制条件

MIMA

- (1)构建项目网络图
- (2)人员模型
- (3)任务历时
- (4)项目时间
- (5)项目成本
- (6)最佳函数

$$M = |E \times T|$$

$$t_j^{len} = \frac{t_j^{eff}}{\sum_{i=1}^{m} m_{ij}}$$

SPSP模型解决方案 - 2.最优函数限制条件

MIMA

- (1)构建项目网络图
- (2)人员模型
- (3)任务历时
- (4)项目时间
- (5)项目成本
- (6)最佳函数

通过项目网络图，可以知道每个任务之间的关联关系，而且每个任务的历时也确定了，通过正推方法和逆推方法确定每个任务的开始时间和结束时间，最后得出关键路径，即项目的完成时间 p^{len}

SPSP模型解决方案 - 2.最优函数限制条件

MIMA

- (1)构建项目网络图
- (2)人员模型
- (3)任务历时
- (4)项目时间
- (5)项目成本
- (6)最佳函数

先要计算每个任务的成本 $t_j^{\cos}$

$$t_j^{\cos} = \sum_{i=1}^{|e|} e_i^{\text{rem}} m_{ij} t_j^{\text{len}}$$

其中，$|e|$ 是人员数。

$$p^{\cos} = \sum_{j=1}^{|t|} t_j^{\cos}$$

其中，$|t|$ 是任务数。

由于项目进度安排的最终目标是成本和时间的最小化，定义 W^{cos} 和 W^{len} 为 p^{cos} 和 p^{len} 的权重，这里（average cost）$^{-1}$ 可以作为 W^{cos} 的初始值，（average length）$^{-1}$ 作为 W^{len} 的初始值，这样，式（7-5）的函数单位可以取消，基本是一个数量级的。

$$f(x) = W^{cos} p^{cos} + W^{len} p^{len} \tag{7-5}$$

式（7-5）没有考虑到超额工作量，如果一个项目人员的在项目中有超额工作，也可能会延长时间，或者增加成本，为此，定义项目人员 e_i 对项目的贡献量函数 $e_i^w(t)$，如式（7-6）所示。

$$e_i^w(t) = \sum_{t_j^{init} \leqslant j \leqslant t_j^{term}} m_{ij}(t) \tag{7-6}$$

式（7-6）表示项目人员 e_i 在时间 t 内的贡献量大于这个人员的 e^{maxd}。为此，我们定义一个函数 ramp(x)，如式（7-7）所示。

$$\text{ramp}(x) = \begin{cases} x, & x > 0 \\ 0, & x \leqslant 0 \end{cases} \tag{7-7}$$

这样项目人员 e_i 的超额工作值如式（7-8）所示。

$$e_i^{overw} = \sum \text{ramp}(e_i^w(t) - e_i^{maxd}) \tag{7-8}$$

这样这个项目的超额工作值 p^{overw} 是所有项目人员的超额工作值之和，如式（7-9）所示。

$$p^{overw} = \sum_{i=1}^{|e|} e_i^{overw} \tag{7-9}$$

其中，$|e|$ 是人员数。

SPSP模型解决方案 - 2.最优函数限制条件

MIMA

- (1)构建项目网络图
- (2)人员模型
- (3)任务历时
- (4)项目时间
- (5)项目成本
- (6)最佳函数

$$f(x) = W^{\cos} p^{\cos} + W^{\text{len}} p^{\text{len}} \qquad (7\text{-}5)$$

$$p^{\text{overw}} = \sum_{i=1}^{|e|} e_i^{\text{overw}} \qquad (7\text{-}9)$$

因此为了得出 SPSP 最佳方案的最佳函数应该是式（7-5）和式（7-9）达到最小值。通过最佳函数可以得出 SPSP 最佳方案，然后就可以合理安排项目了。为此，有人研究了很多组合优化算法以求得最佳方案。

SPSP模型解决方案 - 3.组合优化问题

MIMA

- 组合最优化(combinatorial optimization)是通过对数学方法的研究去寻找离散事件的最优编排、 分组、次序或筛选等，是运筹学中的一个重要分支。组合优化问题的目标是从组合问题的可行解集中求出最优解。

- 典型的组合优化问题有旅行商问题、背包问题、装箱问题、最小度生成树问题、集合覆盖问题等。这些问题描述非常简单，并且有很强的工程代表性，但最优化求解很困难，其主要原因是求解这些问题的算法需要极长的运行时间与极大的存储空间，以致根本不可能在现有计算机上实现，即所谓的“**组合爆炸**”。

SPSP模型解决方案 - 3.组合优化问题

MIMA

- 蚁群算法(Ant Colony Optimization, AGO)已经成功应用到组合优化问题中。
- 其主要特点就是通过正反馈、分布式协作来寻找最优路径。这是一种基于种群寻优的启发式搜索算法。
- 扩展的蚁群优化算法有很多，最大最小蚂蚁系统(Max-Min Ant System, MMAS)就 是其中一个。

SPSP模型解决方案 - 4.最佳方案确定

MIMA

- 根据项目的各任务数据以及人员数据，结合最佳函数，采用MMAS算法获得最佳方案，即人员项目贡献矩阵。

$$\begin{vmatrix} 1 & 0.5 & 0.5 & 0 & 1 & 1 & 1 \\ 1 & 0.5 & 0.5 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0.5 & 0 & 0.5 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$$

SPSP模型解决方案 - 4.最佳方案确定

通过式（7-2）～（7-4）可以计算出每个任务的历时以及每个任务的成本。

任务历时：$t_1^{len}=2$，t_2^{len}，$=3$，$t_3^{len}=4$，$t_4^{len}=6$，$t_5^{len}=4$，$t_6^{len}=4$，$t_7^{len}=4$。

任务成本：$t_1^{cos}=24\,000$ 美元，$t_2^{cos}=33\,000$ 美元，$t_3^{cos}=44\,000$ 美元，$t_4^{cos}=45\,000$ 美元，$t_5^{cos}=48\,000$ 美元，$t_6^{cos}=58\,000$ 美元，$t_7^{cos}=88\,000$ 美元。

这样，通过正推方法和逆推方法确定了每个任务的开始时间和结束时间，由图 7-29 的 PDM 网络图得出关键路径是 $t_1\rightarrow t_3\rightarrow t_5\rightarrow t_6\rightarrow t_7$，因此关键路径的长度是 $p^{len}=18$，即完成项目时间是 18，通过每个任务的成本可以确定项目总成本为 $p^{cos}=340\,000$ 美元。

$T=\{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ 中的 7 个任务分别是 {Planning，High-level design，Low-level design，Testcase design，Coding，System Testing，Acceptance Testing}，因此，可以给出项目的甘特图，如图 7-30 所示。

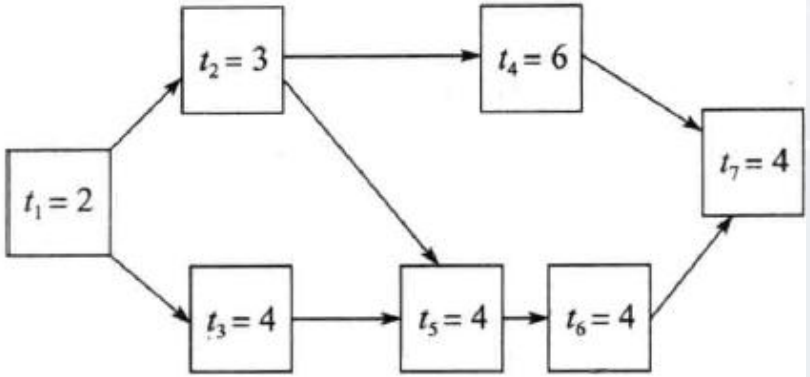


图 7-29　确定历时之后的 PDM 网络图

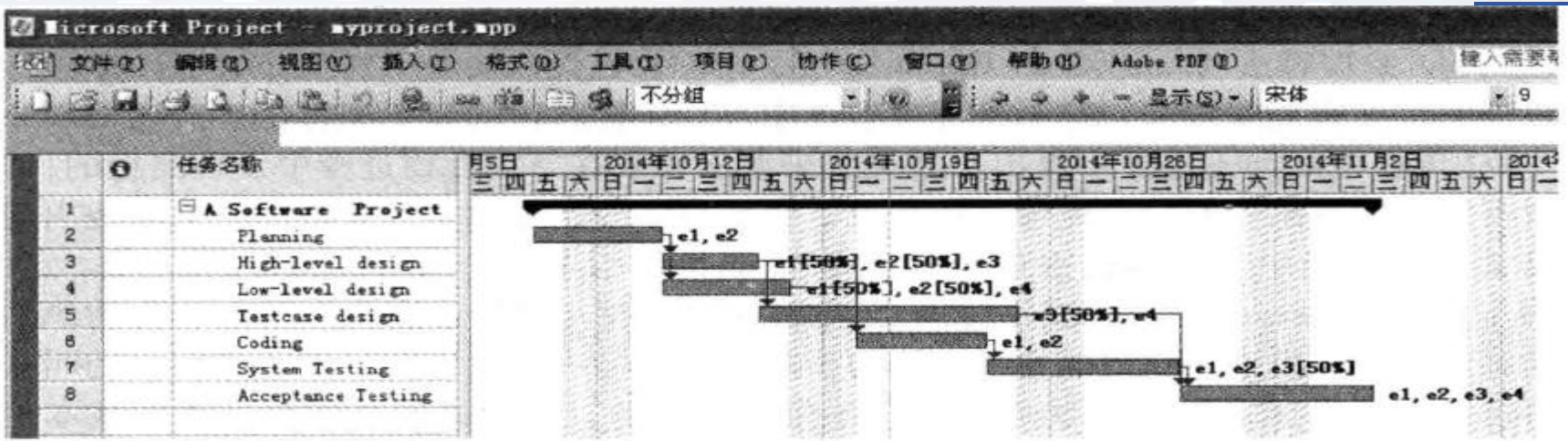


图 7-30　甘特图

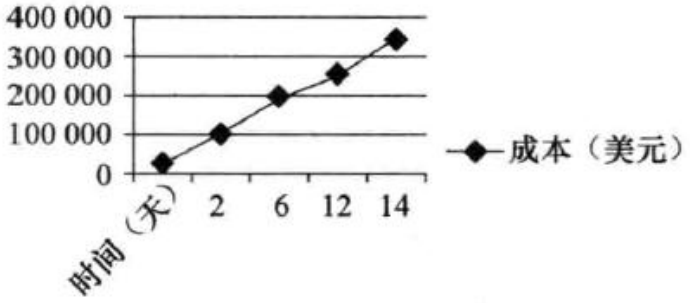


图 7-31　成本基线

进度计划的优化

MIMA

- 如果编制出来的进度计划与要求有差距，就要进行项目计划优化：调整资源，解决资源冲突；调整进度，缩短工期；调整成本预算，减少项目费用。
- 可使用网络分析这一项目进度优化的重要方法。
- 项目进度计划可以采用工具来协助编制。这些软件可自动进行数学计算和资源调整，可迅速地对许多方案加以考虑和选择，还可输出计划编制的结果。

进度计划的优化

MIMA

- 编制一个好的项目计划需要不断完善的过程，需要不断地优化、评审、修改、再评审、细化等，最后才可以确定成为基准的项目计划。
- 对于进度的安排，应该有适度的压力，让开发人员有适度的紧迫感，同时，不可以太过分强调进度，以至于让大家焦点总集中在进度上。
- 制订计划的过程就是一个对项目逐渐了解掌握的过程，通过认真地制订计划，项目经理可以知道哪些要素是明确的，哪些要素是要逐渐明确的，通过渐近明细不断完善项目计划。

项目进度计划的数据分析

MIMA

- 对完成的计划可以进行数据分析，常用的数据分析技术有假设情景分析、模拟等。

- 1) 假设情景分析：

- 假设情景分析是对各种情景进行评估，预测它们对项目目标的影响(积极或消极的)。
- 基于已有的进度计划，考虑各种各样的情景。例如，推迟某主要部件的交货日期，延长某设计工作的时间，或加入外部因素(如罢工或许可证申请流程变化等)。
- 可以根据假设情景分析的结果，评估项目进度计划在不同条件下的可行性，以及为应对意外情况的影响而编制进度储备和应对计划。

项目进度计划的数据分析

MIMA

- 对完成的计划可以进行数据分析，常用的数据分析技术有假设情景分析、模拟等。

- 2) 模拟：

  - 包括基于多种不同的活动假设、制约因素、风险、问题或情景，使用概率分布和不确定性的其他表现形式来计算多种可能的工作包持续时间，同时评估它们对项目目标的潜在影响。
  - 最常见的模拟技术是蒙特卡罗分析(MonteCarloanalysis),它利用风险和其他不确定资源计算整个项目可能的进度结果。

进度计划新兴实践简述

MIMA

- 项目具有渐进明细的特性，在高技术行业，日新月异是主要特点，因此项目计划需要在一定条件的限制和假设之下采用渐进明细的方式不断完善。
- 敏捷模型及迭代模型都能很好地解决这个问题。
- 一个好的项目计划的开发应该是渐进式的。

进度计划新兴实践简述

MIMA

- 新兴实践举例：
- 1.具有未完项的迭代型进度计划
  - 基于适应型生命周期的滚动式规划，如敏捷的产品开发方法。
  - 这种方法将需求记录在用户故事中，然后在建造之前按优先级排序并优化用户故事，最后在规定的时间盒内开发产品功能。
  - 通常用于向客户交付增量价值，或多个团队并行开发大量内部关联较小的功能。
  - 适应型生命周期在产品开发中的应用越来越普遍，很多项目都采用这种进度计划方法。这种方法的好处在于，它允许在整个开发生命周期期间进行变更。

进度计划新兴实践简述

MIMA

- 新兴实践举例：
- 2.按需进度计划

- 这种方法通常用于看板体系，基于制约理论和来自精益生产的拉动式进度计划概念，根据团队的交付能力来限制团队正在开展的工作。
- 按需进度计划方法不依赖于以前为产品开发或产品增量制定的进度计划，而是在资源可用时立即从未完项和工作序列中提取出来开展。
- 按需进度计划方法通常用于产品在运营和维护环境下以增量方式演进，且任务的规模或范围相对类似，或者可以按照规模或范围对任务进行组合的项目。

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
- 软件项目进度计划确定
- **案例分析**
- 课程实践

“医疗信息商务平台”项目进度案例分析

MIMA

- 迭代计划
- Sprint计划
- Sprint待开发事项列表
- Sprint预算

课程实践

MIMA

- 实践一：网络图和历时估算
- 实践二：项目进度编排

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
- 软件项目进度计划确定
- 案例分析
- 课程实践

感谢！

软 件 学 院　罗 昕
luoxin@sdu.edu.cn