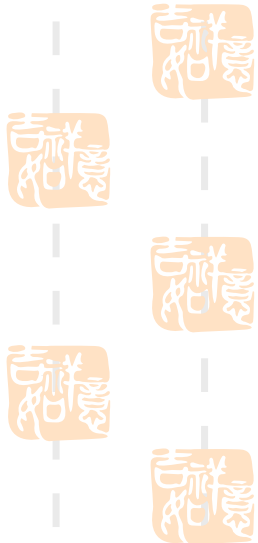


软件协同设计课程之

## 5.3 工作量估算



2025年1月



# 1. 软件项目的估算 (1/4)

- 估算：时间、成本、大小
- 方法：
  - 功能点、特性点扩展方法
    - 功能点：MIS适合
    - 特性点：其它类型的软件。如计算分析软件、嵌入式软件
  - Cocomo 构建性成本模型
    - 成本驱动因子属性：产品属性、硬件属性、人员属性、项目属性。
  - 宽带Delphi方法
    - 几个有经验的人估计同样的任务
    - 需要细致的分解工作
    - 基于一致意见的迭代过程

# 功能点计算举例，适合MIS（2/4）

度量	计数	*	权重			=	
			简单	平均	复杂		
用户输入数	3		3	4	6		9
用户输出数	2		4	5	7		8
用户查询数	2		3	4	6		6
文件数	1		7	10	15		7
外部接口数	4		5	7	10		20
总功能点							50

如果每个功能点100LOC, 则共有  
 $50 * 100 \text{LOC} = 5 \text{kloc}$



## 2. COCOMO模型

- **构造性成本模型**Constructive Cost Model
- Boehm1981年提出，被不断研究改进，是一种参数化的项目估算方法
- COCOMO有3套模型：
  - **基本型**：用估计出的代码行LOC为参数，估算项目工作量
  - **中间型**：考虑产品、硬件、人员、项目等参数的影响，调整项目工作量
  - **详细型**：以上全部特征，增加软件工程中分析、设计过程等影响

# CoCoMo模型

## ■ 将软件项目类型划分为三类：

项目类型	定 义
组织型	相对较小、较简单的软件项目，对需求不苛刻，开发人员对开发目标理解充分，相关的工作经验丰富，对使用环境熟悉，受硬件约束较少，程序规模不大（<5万行），如多数应用软件、早期的操作系统和编译程序等
嵌入型	软件在紧密联系的硬件、其他软件 and 操作的限制条件下运行，通常与硬件设备紧密结合在一起，对接口、数据结构、算法要求较高，软件规模任意。如大而复杂的事务处理系统、大型/超大型操作系统、航天用控制系统、大型指挥系统等
半独立型	介于组织型和嵌入型之间，软件规模和复杂性属中等以上，最大可达30万行。如多数事务处理系统、操作系统、数据库管理系统、大型库存/生产控制系统、简单的指挥系统等

# Cocomo 构建性成本模型举例 (3/4)

## (基本型, 只考虑KLOC)

### ■ 项目的人月

$$E = a * KLOC^b$$

### ■ 项目开发时间

$$D = c * E^d$$

项目复杂度	a	b	c	d
组织性	2.4	1.05	2.5	0.38
半独立性	3.0	1.12	2.5	0.35
嵌入型	3.6	1.2	2.5	0.32

# Cocomo 构建性成本模型举例 (4/4)

- 如果KLOC=5, 组织型的项目

$$E = a * KLOC^b$$
$$= 2.4(5)^{1.05} = 13.0 \text{ (人月)}$$

$$D = c * E^d$$
$$= 2.5(13)^{0.35} = 2.5 * 2.45 = 6.13 \text{ (月)}$$

■ 人数

$$N = E/D = 13/6.13 = 2.12$$

### 3. 中间CoCoMo模型

- 在基本CoCoMo模型基础上考虑了15种影响软件工作量的因素
- 通过工作量调节因子(EAF)修正对工作量的估算，从而使估算更合理
- 人月计算公式如下： $E=a(KLOC)^bEAF$ 
  - 其中：KLOC是软件产品的目标代码行数，单位是千行代码数，a、b是常数，取值如下表所示

项目类型	a	b
组织型	3.2	1.05
半独立型	3.0	1.12
嵌入型	2.8	1.20



# 工作量调节因子的计算

- 每个调节因子 $F_i$ 的取值分为很低、低、正常、高、很高、极高六级，正常情况下 $F_i=1$
- 当15个 $F_i$ 选定，可得：
$$EAF = \prod_{i=1}^{15} F_i$$

工作量因素 $F_i$		很低	低	正常	高	很高	极高
产品因素	软件可靠性	0.75	0.88	1.00	1.15	1.40	
	数据库规模		0.94	1.00	1.08	1.16	
	产品复杂性	0.70	0.85	1.00	1.15	1.30	1.65
计算机因素	执行时间限制			1.00	1.11	1.30	1.66
	存储限制			1.00	1.06	1.21	1.56
	虚拟机易变性		0.87	1.00	1.15	1.30	
	环境周转时间		0.87	1.00	1.07	1.15	
人员的因素	分析员能力		1.46	1.00	0.86		
	应用领域实际经验	1.29	1.13	1.00	0.91	0.71	
	程序员能力（软硬件结合）	1.42	1.17	1.00	0.86	0.82	
	虚拟机使用经验	1.21	1.10	1.00	0.90	0.70	
	程序语言使用经验	1.41	1.07	1.00	0.95		
项目因素	现代程序设计技术	1.24	1.10	1.00	0.91	0.82	
	软件工具的使用	1.24	1.10	1.00	0.91	0.83	
	开发进度限制	1.23	1.08	1.00	1.04	1.10	

## 4. 项目进度管理

- **目标：确保软件项目在规定的时间内完成**
- **项目进度管理任务**
  - 定义所有的项目任务以及它们之间的依赖关系
  - 规划每个任务所需的工作量和持续时间
  - 制订项目的进度安排
  - 在项目开发过程中不断跟踪项目的执行情况，发现那些未按计划进度完成的任务对整个项目工期的影响，并及时进行调整

## 4. 项目进度管理

### ■ 制定进度计划的两种情况

- 客户方都**规定了明确的交付日期**，此时进度安排必须在此约束下进行
- **只规定了大致的时间界限**，最终的交付日期由开发组织确定，此时的进度安排可以比较灵活，工作量的分布可考虑对资源的充分利用

# 5 如何制定软件项目进度

## ■ 任务划分

- 项目划分成若干可以管理的活动和任务，为了实现项目的划分，对产品和过程都需要进行分解

## ■ 相互依赖性分析

- 确定各个被划分的活动或任务之间的相互关系，有些任务必须是串行的，有些可能是并行的
- 采用网络图分析的方法（数据结构中有介绍）

# 5 如何制定软件项目进度

## ■ 时间分配

- 必须为每个被调度的任务分配一定数量的时间，此外还必须为每个任务制定开始和结束日期，这些日期是相互依赖的

## ■ 工作量确认

- 确保在任意时段中分配给任务的人员数量不会超过项目组中的人员数量

## 5 如何制定软件项目进度

### ■ 定义责任

- 每个被调度的任务都应该指定某个特定的小组成员来负责

### ■ 定义结果

- 每个被调度的任务都应该有一个确定的输出结果

### ■ 定义里程碑

- 每个任务或任务组都应该与一个项目里程碑相关联(当一个或多个工作产品经过质量评审并且得到认可时，标志着一个里程碑的完成)