



# COCOMO II模型讲解

作者：陈 云



# 主要内容

---

- **COCOMO II模型介绍**
- 规模估算
- 工作量估算方法
- 规模因子
- 成本驱动因子
- 成本、工期估算方法
- 工具演示



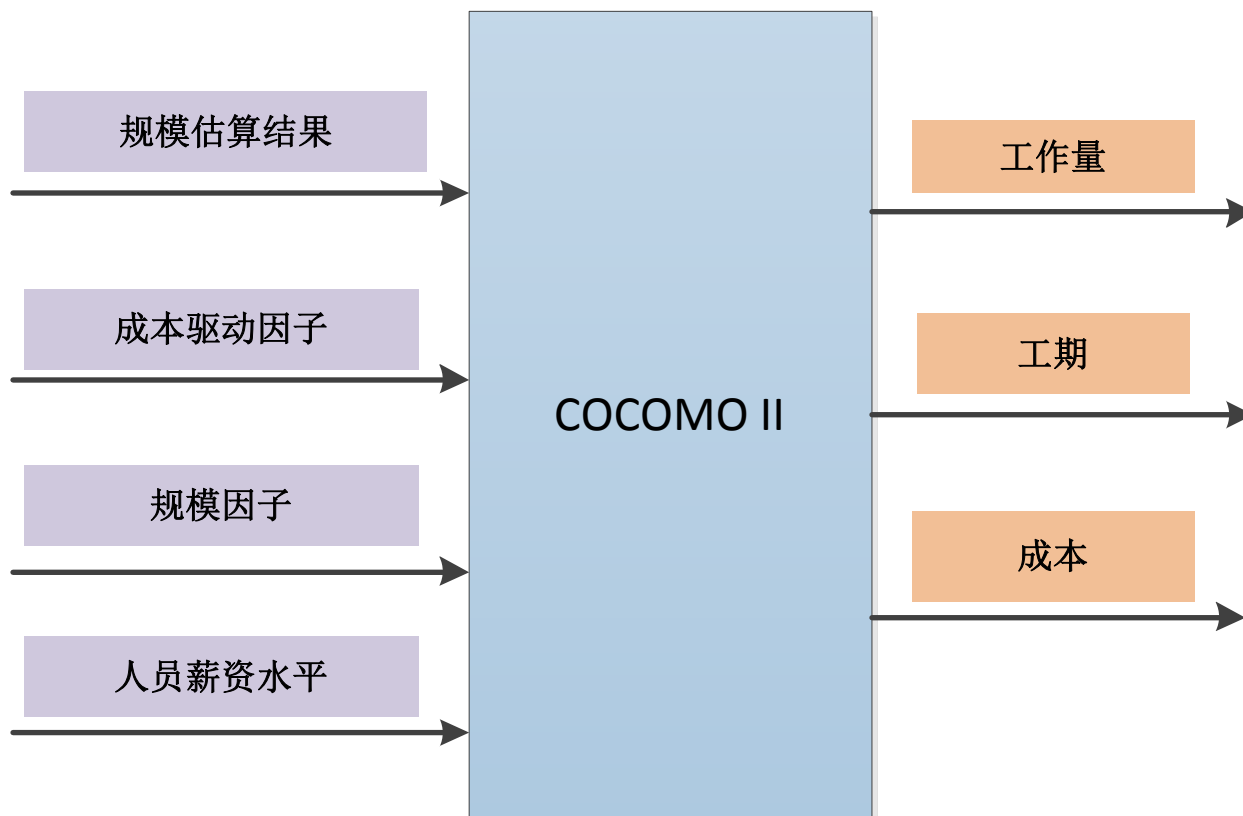
## COCOMO ( *COnstructive Cost MOdel* ) 模型发展介绍

- **COCOMO** (也叫**COCOMO81**) 最初发表于1981年巴里·勃姆《软件工程经济学》一书中, 作为一种估算工作量、成本以及工期的模型。
- 发展动因: 软件开发技术发生了翻天覆地的变化, 包括:
  - 基于大型机的批处理方式向基于台式机的实时方式的转变;
  - 更强调已有软件的复用和开发新系统时利用已有软件模块;
  - 用于软件设计管理和软件开发两个部分的开销应保持均衡。
- 1997年, **COCOMO II**开始研发, 并最终于2001年发表于《软件成本估算: **COCOMO II** 模型方法》一书中。

应用实践表明, 今天的**COCOMO II**模型已经有了相当的正确性, 其估算的软件开发成本与实际成本相差不到**20%**, 进度相差不到**46%**, 很好地满足了项目决策和管理的需要。



# COCOMO II模型示意图





# COCOMO II模型对软件的分类型

未来的软件开发可被分为以下几种：

- **终端用户编程（End-User Programming）**：指没有编程经验的用户可以开发一些具有简单用途或信息处理功能的软件，比如：**Excel**中可以由用户定义计算公式。
- **应用生成器（Application Generators）**：指可以根据用户自定义需求生成现有的解决方案。
- **应用组件（Application Composition）**：指将不同的可重用组件进行组合生成解决方案。
- **系统集成（System Integration）**：大规模软件需要更高层次的系统工程。
- **基础架构（Infrastructure）**：指涉及到操作系统、数据库管理系统等的开发方式。
- 第一类开发方式（终端用户编程）不需要使用**COCOMO II**来估算，因为该类应用程序具有低复杂度的特点，只需耗费几个小时就能完成。对于另外四种开发方式，**COCOMO II**提供了三种子模型用于估算。



# COCOMO II模型的划分

---

## COCOMO II模型包括三种子模型：

- 应用组合子模型（ **Application Composition Model** ）
- 早期设计子模型（ **Early Design Model** ）
- 后体系架构子模型（ **Post-Architecture Model** ）

## 应用组合子模型 (Application Composition Model)

- 模型适用于那些不能用应用生成器方式但能用应用组件方式开发的应用程序，比如GUI Builder、查询浏览器、数据库管理等。
- 模型使用对象点作为规模估算的单位。通过估计待估算系统的界面数、报表数和第三代编程语言（高级语言）模块得到对象点数目。
- 该子模型的工作量计算公式：

$$PM = \frac{NOP}{PROD}$$

- PM (Person-Month)：以人月数为单位表示的工作量
  - NOP (New Object Point)：新对象点数
  - PROD(Developers experience and capability)：开发人员经验和能力
- 该子模型应用范围比较局限，不是本次培训的重点，在此不做深入介绍。



## 早期设计子模型(Early Design Model)

- 该模型可用于以下几种开发方式的软件：
  - ✓ 应用生成器
  - ✓ 系统集成
  - ✓ 基础架构
- 模型使用**未调整功能点数(或源代码行数)**作为规模估算的输入。
- 早期设计子模型适用于**软件项目早期**，一般用于选择产品架构方案或增量迭代策略。
- 模型中使用**7种成本驱动因子**的工作量乘数来对工作量进行调整。





## 后体系架构子模型(Post-Architecture Model)

- 模型可用于以下几种开发方式的软件：
  - 应用生成器
  - 系统集成
  - 基础架构
- 以未调整功能点数或源代码行数作为规模估算输入。
- 该模型是COCOMO II模型中最为详细具体的模型，可用于实际开发过程或维护过程中。



## COCOMO II模型总结

- 简而言之，COCOMO II有三种子模型可用于应用生成器、系统集成和基础架构三种开发方式的软件的估算。
- 在螺旋模型开发的初始阶段，原型设计占多数，此时使用**应用组合子模型**最为恰当。
- 螺旋模型的下一阶段将探索所有可能的设计方案，项目信息更为详细，此时可使用**早期设计子模型**进行估算。
- 当项目即将进入开发阶段，项目信息最为详细，成本驱动因子的值均可确定，使用**后体系架构子模型**来估算最好。



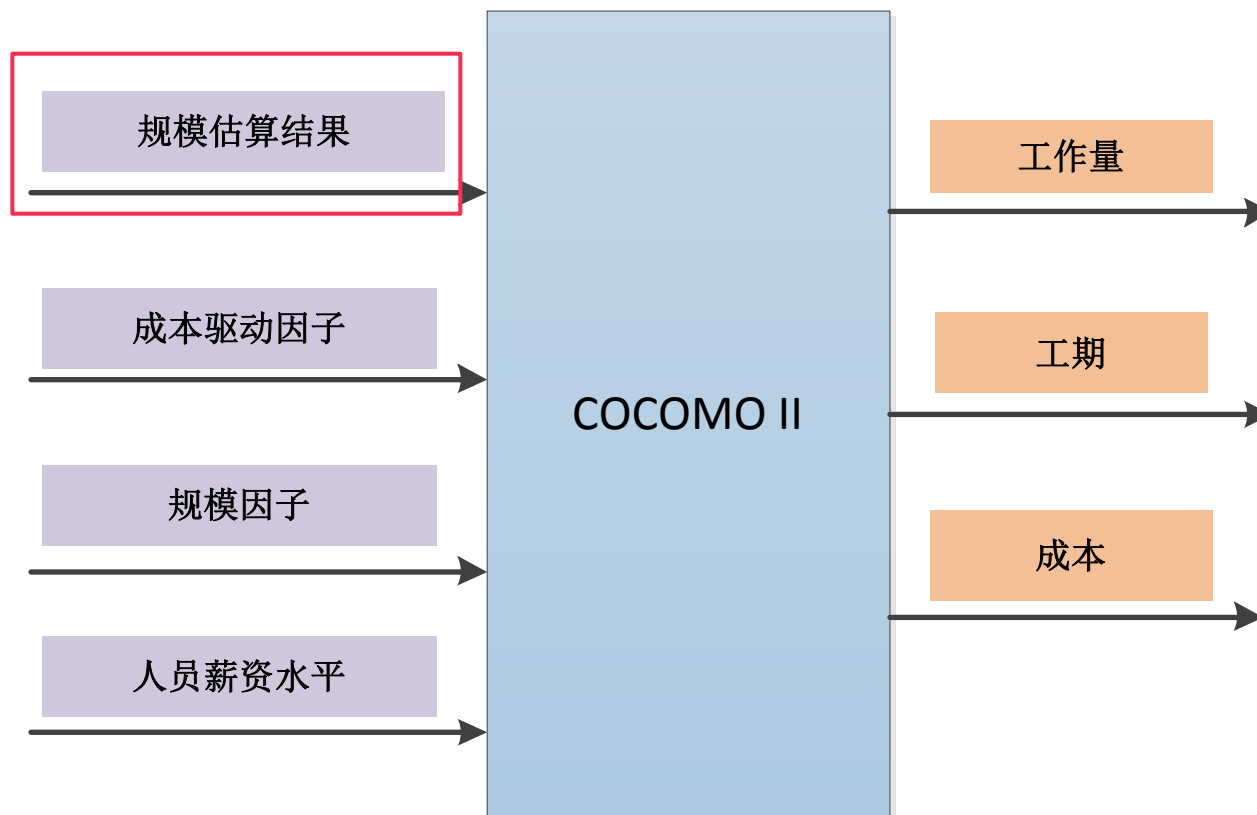
# 主要内容

---

- **COCOMO II模型介绍**
- **规模估算**
- 工作量估算方法
- 规模因子
- 成本驱动因子
- 成本、工期估算方法
- 工具演示



# COCOMO II模型示意图





# 规模估算

## 代码行估算:

- COCOMO II选择逻辑源代码行数（SLOC）作为代码行的度量方式。
- 美国软件工程研究所的逻辑源代码语句定义检查单作为代码行的定义标准。

## 功能点估算:

- 项目生命周期早期就可以估算出项目规模。
- 识别五种用户功能类型，根据复杂度进行分类，确定权重，累加得出未调整功能点数。

## 功能点转换为代码行

- COCOMO II的早期设计和后体系结构子模型使用代码行数作为工作量计算公式的输入，因此未调整功能点数要先转化为等量的代码行数。



# 规模估算

编程语言	源代码行数	编程语言	源代码行数
Basic Assembly	575	dBase III	60
JCL	400	Focus	60
Macro Assembly	400	Clipper	60
C	225	Oracle	60
Cobol 74 (Cobol I)	220	Sybase	60
FORTRAN	210	dBase IV	55
Cobol 85 (Cobol II)	175	Perl	50
Pascal	160	JavaScript	50
PL/1	126	VBScript	50
RPG I	120	Shell Script	50
C++	80	SAS	50
Java	80	APL	50



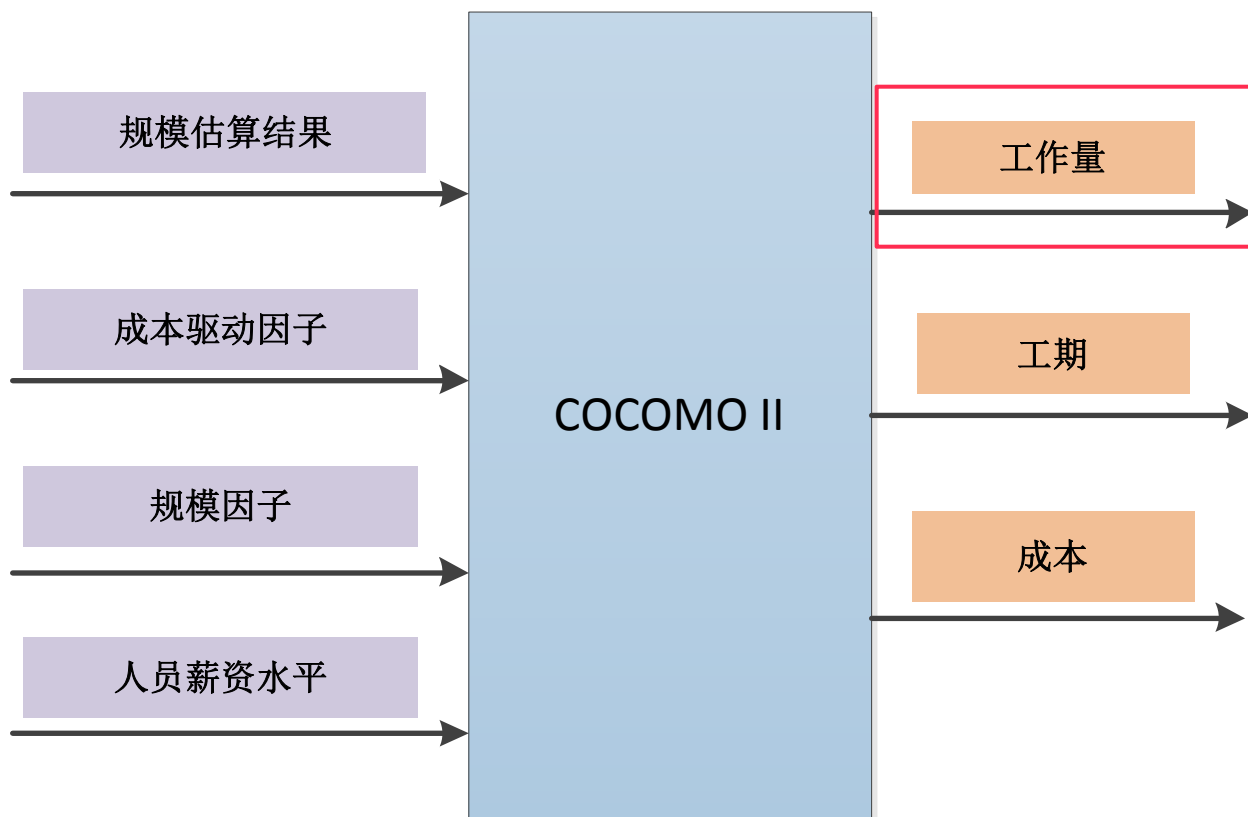
# 主要内容

---

- **COCOMO II模型介绍**
- 规模估算
- 工作量估算
- 规模因子
- 成本驱动因子
- 成本、工期估算方法
- 工具演示



# COCOMO II模型示意图







## 工作量估算

COCOMO II模型工作量计算公式为：

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

- $A=2.94$ ,  $B=0.91$ 。
- $Size$ 代表以千行源代码（KLOC）表示的项目规模。
- $E$ 是五个规模因子（Scale Factor）共同产生的结果。
- $EM$ (Effort Multiplies)是工作量乘数，不同子模型涉及的 $EM$ 数目不同，早期设计子模型涉及到7个成本驱动因子（Cost Driver），后体系架构子模型涉及17个成本驱动因子。



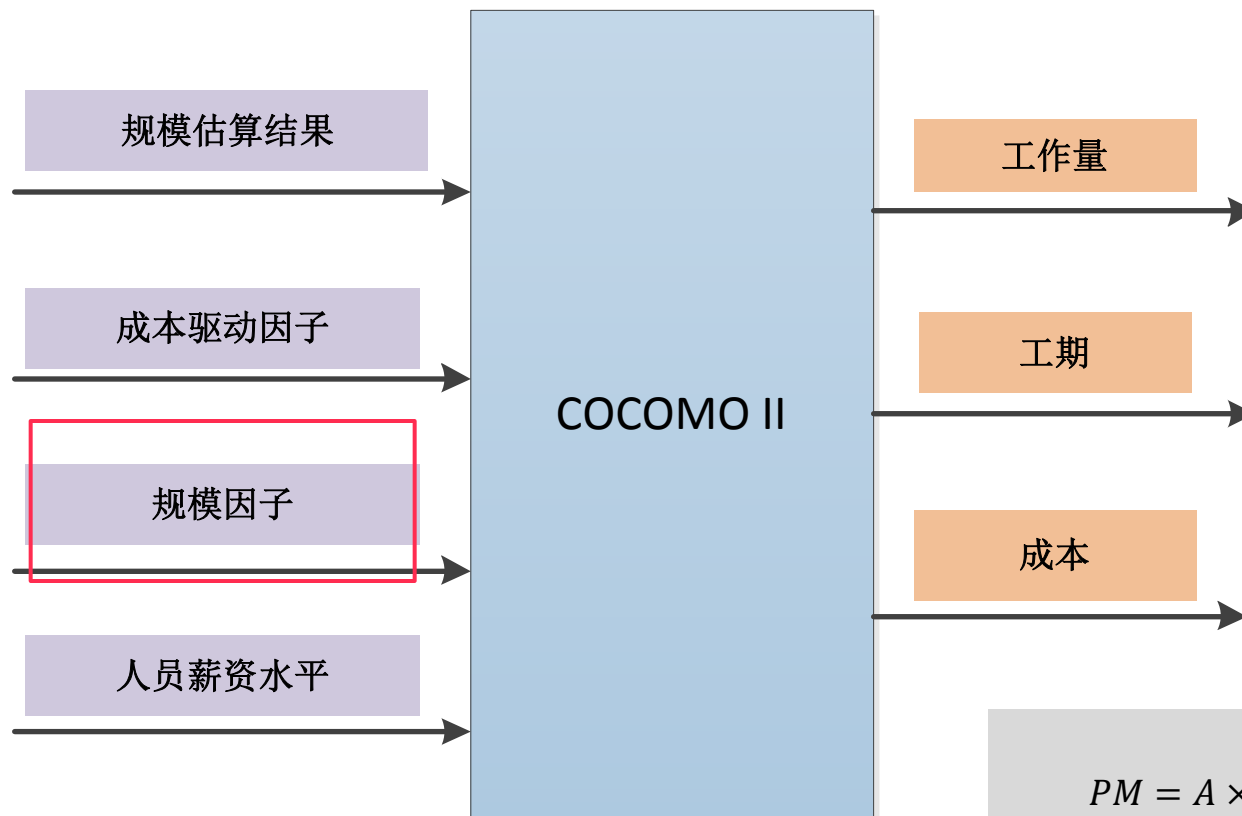
# 主要内容

---

- **COCOMO II模型介绍**
- 规模估算
- 工作量估算
- 规模因子
- 成本驱动因子
- 成本、工期估算方法
- 工具演示



# COCOMO II模型示意图



$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 规模因子(Scale Factors)

- 具有相同软件规模的项目之间存在着差异性，这就促使研究人员在描述项目时还要加入新的因素。
- 通过使用规模因子，具有相同规模的软件的估算结果会因为不同情况而出现差异。
- 早期设计子模型和后体系架构子模型中都要用到**5**种规模因子。

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 规模因子(Scale Factors)

- 5种规模因子决定了软件开发项目的规模经济和规模非经济特性
- COCOMO II判断项目是否存在规模经济或规模非经济:

$$E = B + 0.01 \times \prod_{i=1}^5 SF_i$$

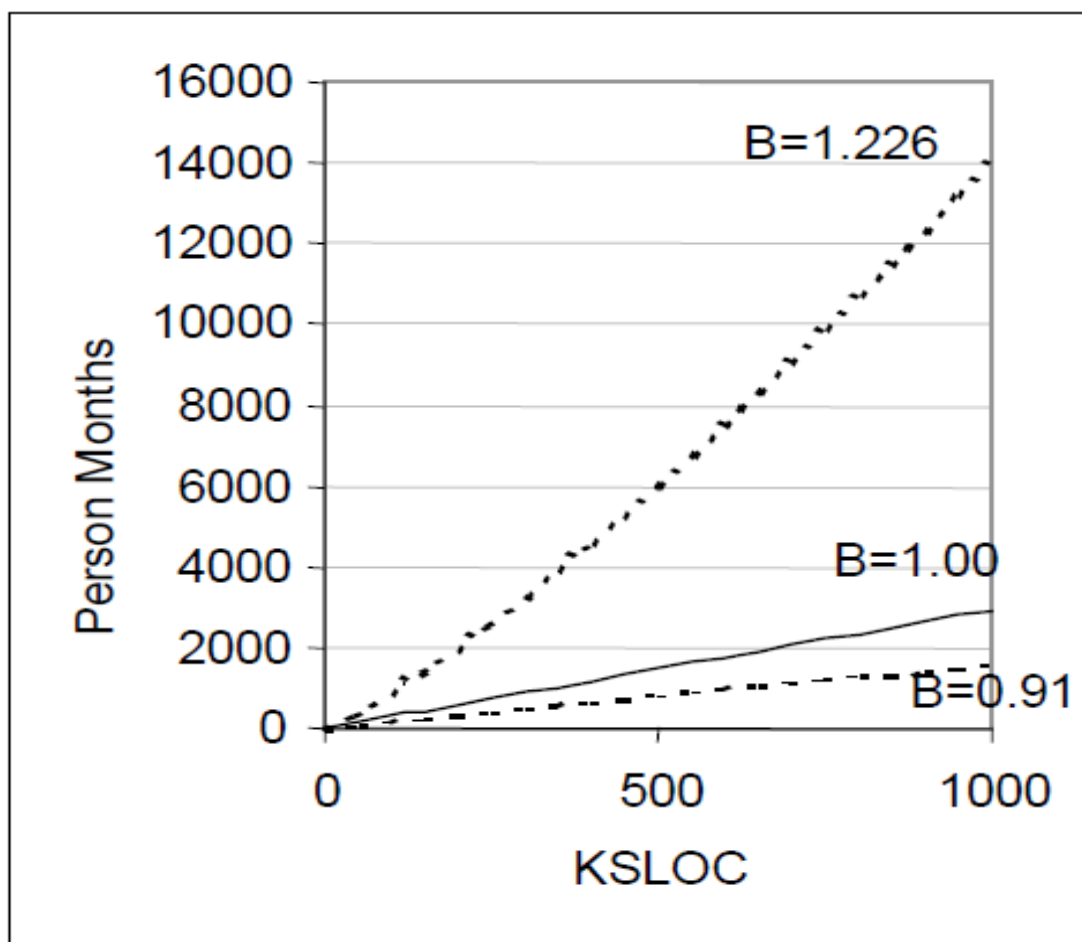
- 当E大于1时，所需工作量(PM)的增加速度大于软件规模(SIZE)的增加速度，体现出规模非经济性。反之，E小于1则表示规模经济性。
- COCOMO II计算结果的准确性关键在于识别项目比例因子的准确性。

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



# 规模因子(Scale Factors)



$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 规模因子(Scale Factors)

- **PREC- Precedentedness (先例性)**: 该因子描述了待估算项目对组织的新鲜程度。如果该项目和组织的几个已开发项目类似，则PREC为高。

规模因子	非常低	低	一般	高	非常高	极其高
先例性 (PREC)	完全无先例 6.20	大部分无先例 4.96	部分无先例 3.72	基本熟知 2.48	大部分熟知 1.24	完全熟知 0.00

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 规模因子(Scale Factors)

- FLEX- Development Flexibility (开发灵活性): 该因子表示待估算项目和需求说明书/外部接口规格说明书保持一致的严格程度。

规模因子	非常低	低	一般	高	非常高	极其高
开发灵活性 (FLEX)	严格 5.07	个别地方可以 放宽要求 4.05	部分放宽要 求 3.04	基本一致 2.03	部分一致 1.01	不偏离目 标即可 0.00

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$





# 规模因子(Scale Factors)

- RESL- Architecture / Risk Resolution (早期体系结构/风险化解) : 该因子包括设计完备性和风险消除能力。

规模因子	非常低	低	一般	高	非常高	极其高
早期体系结构/风险化解 (RESL)	很少(20%) 7.07	有时 (40%) 5.65	经常 (60%) 4.24	基本上 (75%) 2.83	大多数 (90%) 1.41	完全达到 (100%) 0.00

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 规模因子(Scale Factors)

- TEAM- Team Cohesion (团队凝聚力): 该因子表示了项目在统一所有相关人员意见方面的不确定程度, 相关人员包括: 软件用户、顾客、开发人员、维护人员和其他潜在人员。

规模因子	非常低	低	一般	高	非常高	极其高
团队凝聚力 (TEAM)	很难沟通 5.48	有些难沟通 4.38	基本沟通顺畅 3.29	大部分沟通 顺畅 2.19	配合高度协 调 1.10	无缝沟通 0.00

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



# 规模因子(Scale Factors)

- PMAT-Process Maturity (过程成熟度) :有两种方式可以确定PMAT:
  - 根据CMMI的评定结果;
  - 根据软件工程研究所 (SEI) 的能力成熟度模型的18个关键过程域 (KPA) 。只要确定18个KPA的百分比, 就能确定最终结果。

规模因子	非常低	低	一般	高	非常高	极其高
	自身估算出的过程成熟度等级或					
过程成熟度 (PMAT)	CMMI Level 1 (初级) 7.80	CMMI Level 1 (高级) 6.24	CMMI Level 2 4.68	CMMI Level 3 3.12	CMMI Level 4 1.56	CMMI Level 5 0.00

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



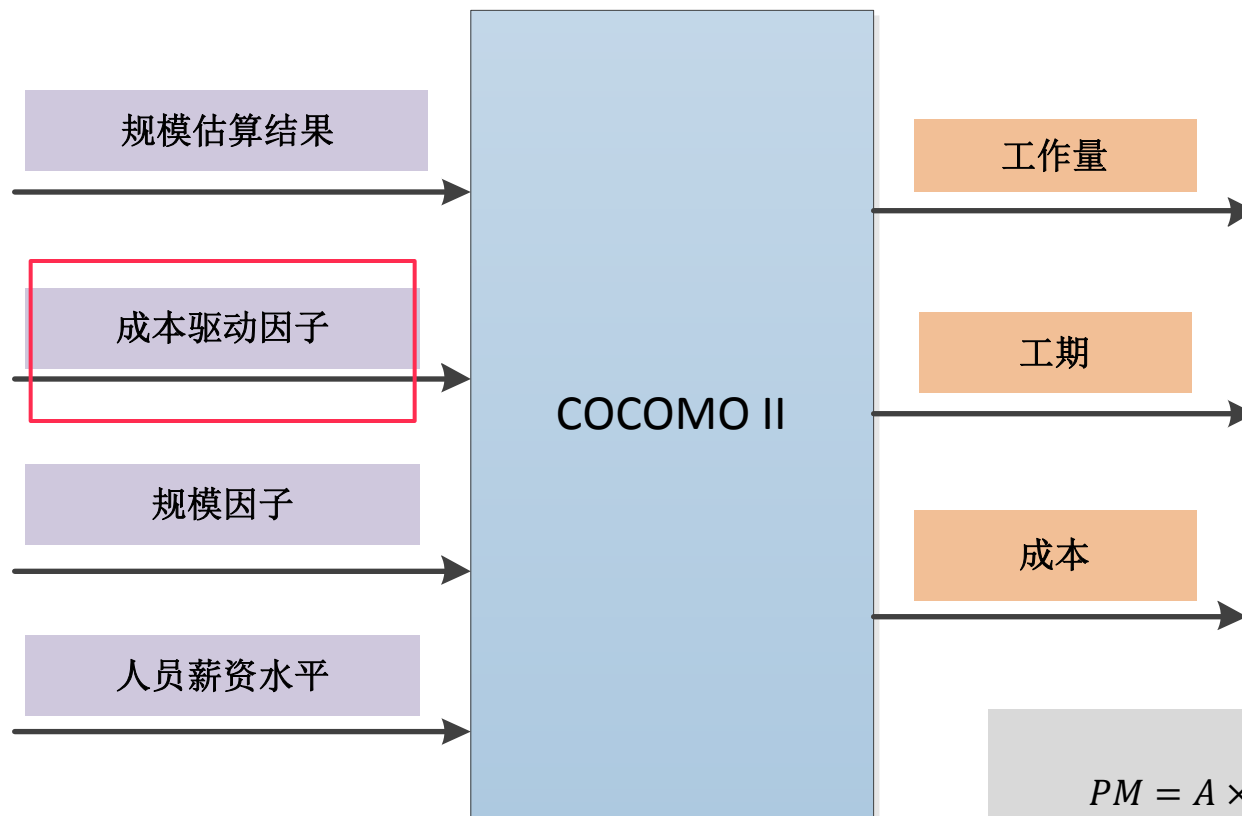
# 主要内容

---

- **COCOMO II模型介绍**
- 规模估算
- 工作量估算
- 规模因子
- **成本驱动因子**
- 成本、工期估算方法
- 工具演示



# COCOMO II模型示意图



$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 成本驱动因子 (Cost Drivers)

COCOMO II 有两类成本驱动因子适用于两种不同项目阶段。第一种是后体系架构子模型的工作量乘数。第二种是早期设计子模型的工作量乘数。

后体系架构子模型中包括17个工作量乘数，可被分为四类：**产品类、平台类、人员类及项目类**，每个类又包括一些子属性。

成本驱动因子的不同级别对应不同数值，级别包括：非常低、低、一般、高、非常高和极其高。

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 成本驱动因子-产品属性1

- 表示可以对工作量产生影响的产品特性。
- 软件可靠性需求（**RELY**）：该属性表示对软件可靠性的依赖程度。若软件失效带来的不便可忽略不计，则RELY低；若软件失效会影响人类的正常生活，则RELY为高。

RELY 描述:	只产生些 许不便	影响程度低， 破坏易修复	影响程度一般，产 生的破坏易修复	产生重大经 济损失	影响人类的 正常生活	
等级	非常低	低	一般	高	非常高	极其高
EM	0.82	0.92	1.00	1.10	1.26	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 成本驱动因子-产品属性2

- 应用数据库的大小(**DATA**): 该属性表示大量测试数据需求对产品开发的影响程度。因为产生的测试数据会在后续用于检验产品开发的正确性。
- 该属性通过**D/P**的值来度量, 即用于测试的数据除以源代码行数。

DATA描述		D/P ≤ 10		10 ≤ D/P ≤ 100	100 ≤ D/P ≤ 1000	D/P ≥ 1000	
等级		非常低	低	一般	高	非常高	极其高
EM		n/a	0.90	1.00	1.14	1.28	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$





## 成本驱动因子-产品属性3

- 产品复杂度(**CPLX**): 软件开发的复杂度涉及五个方面: 控制操作、计算操作、设备相关操作、数据管理操作和用户接口操作。
- 选择一个或多个方面的复杂度来刻画项目复杂度。
- 最终的复杂度等级是一个或几个方面的复杂度的加权平均。

等级	非常低	低	一般	高	非常高	极其高
EM	0.73	0.87	1.00	1.17	1.34	1.74

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 成本驱动因子-产品属性4

- 可重用性需求(**RUSE**): 该属性表示使模块可重用（用于现有项目或未来项目）所需的额外工作量。额外工作量主要用于完善软件的通用设计、更详细的文档和更全面的测试以确保模块能直接用于其他应用程序。

RUSE 描述		无	用于一个组织 内的单个项目	用于一个 组织内的 多个项目	用于产品线	用于多条生 产线
等级	非常低	低	一般	高	非常高	极其高
EM	n/a	0.95	1.00	1.07	1.15	1.24

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 成本驱动因子-产品属性5

- 文档与生命周期需求的匹配度(**DOCU**): 根据项目文档和项目所在生命周期的吻合程度而确定的。本项驱动因子从非常低（有很多项目生命周期的需求未被覆盖）到非常高（除满足项目生命周期的需求外，还有额外的文档）。

DOCU 描述	许多生命 周期需求 未被覆盖	有些生命 周期需求 未被覆盖	生命周期需 恰好被覆盖	覆盖生命周 期需求外， 还有一些其 他文档	覆盖生命周 期需求外， 还有很多其 它文档	
等级	非常低	低	一般	高	非常高	极其高
EM	0.81	0.91	1.00	1.11	1.23	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

- | TIME<br>描述 |     |     | ≤50% | 70%  | 85%  | 95%  |
|------------|-----|-----|------|------|------|------|
| 等级         | 非常低 | 低   | 一般   | 高    | 非常高  | 极其高  |
| EM         | n/a | n/a | 1.00 | 1.11 | 1.29 | 1.63 |

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

■ **内存约束(STOR):** 该属性表示花费在软件系统或子系统上的主存大小。若消耗的执行时间资源少于**50%**，则该属性的值为正常；若大于**95%**的执行时间资源，则属性值为非常高。

STOR		STOR 描述					
		≤50%		70%	85%	95%	
等级		非常低	低	一般	高	非常高	极其高
EM		n/a	n/a	1.00	1.05	1.17	1.46

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 成本驱动因子-平台属性3

- 平台稳定性(**PVOL**): 表示开发平台的稳定程度。
- 若要开发的软件是操作系统, 那么平台就是计算机硬件; 若要开发的软件是数据库管理系统, 平台则包括网络、计算机硬件、操作系统和分布式信息存储。平台还可能包括软件开发过程中使用到的编译器或汇编器。若每12个月发生一次平台革新, 该属性值为低; 若每两周生一次平台革新, 属性值为高。

POVL 描述	重大改变: 12个月一次 小改动: 1个月一次					
	重大改变: 6个月一次 小改动: 2个月一次		重大改变: 2个月一次 小改动: 1周一次		重大改变: 2周一次 小改动: 1天一次	
等级	非常低	低	一般	高	非常高	极其高
EM	n/a	0.87	1.00	1.15	1.30	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

## 成本驱动因子-人员属性1

- 分析能力(**ACAP**): 分析人员包括所有致力于开发软件需求、概要设计和详细设计的工作人员。该属性值涉及分析和设计能力、高效性和全面性以及交流或合作的能力, 不考虑分析人员的工作经验。
- 使用分析人员占总开发人员的比例来度量该属性值。该属性值小于**15%**, 则级别为非常低; 若不小于**95%**, 则级别为非常高。

ACAP 描述	15%	35%	55%	75%	95%	
等级	非常低	低	一般	高	非常高	极其高
EM	1.42	1.19	1.00	0.85	0.71	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

## 成本驱动因子-人员属性2

- 编程人员能力(**PCAP**): 目前软件开发的总趋势是重视高素质的分析人员。但是随着复杂公用架包的地位不断提升,生产力的提升与编程人员处理这些架包的能力变得息息相关,因此编程人员的能力变得越来越重要。
- 使用编程人员占总开发人员的比例来度量该属性值。

PCAP 描述	15%	35%	55%	75%	95%	
等级	非常低	低	一般	高	非常高	极其高
EM	1.34	1.15	1.00	0.88	0.76	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 成本驱动因子-人员属性3

■ 应用经验(**AEXP**): 该属性值由开发团队开发软件系统或子系统的团队经验的等级确定, 可由开发团队开发同类型软件系统的团队经验等级来定义。若团队项目经验少于两个月, 则该属性值为低; 若团队项目经验不少于六年, 则该属性值为高。

AEXP 描述	≤2个月	6 个月	1 年	3 年	6 年	
等级	非常低	低	一般	高	非常高	极其高
EM	1.22	1.10	1.00	0.88	0.81	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

## 成本驱动因子-人员属性4

- 平台经验(**PLEX**): 后体系架构子模型突出了平台经验属性对生产力的影响, 强调了对更强大平台的熟练程度的重要性, 平台包括: 更强大的图形用户界面、数据库、网络 and 分布式中间件。

PLEX 描述	≤2个月	6 个月	1 年	3 年	6 年	
等级	非常低	低	一般	高	非常高	极其高
EM	1.19	1.09	1.00	0.91	0.85	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

## 成本驱动因子-人员属性5

- 编程语言和工具经验(**LTEX**): 对编程语言和软件工具经验等级的概括。
- 软件开发要使用很多种工具, 用于需求和设计的展示和分析、配置管理、文档提取、库管理、程序风格管理、一致性检验等。
- 除了特定编程语言的使用经验, 开发工具也影响开发时间。若编程语言和工具的使用经验少于两个月, 则该属性值为低; 若编程语言和工具的使用经验不少于六年, 则该属性值为高。

LTEX 描述	≤2个月	6 个月	1 年	3 年	6 年	
等级	非常低	低	一般	高	非常高	极其高
EM	1.20	1.09	1.00	0.91	0.84	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

## 成本驱动因子-人员属性6

- 人员连续性(**PCON**): 该属性反映了项目每年人员革新的比例, 3%为非常高, 48%为非常低。

PCON 描述	48%/年	24%/年	12%/年	6%/年	3%/年	
等级	非常低	低	一般	高	非常高	极其高
EM	1.29	1.12	1.00	0.90	0.81	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

# 成本驱动因子-项目属性1

- 采用的软件工具(**TOOL**): 自从1970年使用项目来调整COCOMO模型以来, 软件工具的功能和性能都有了很大的提升。若工具只有简单的编辑代码功能, 则该属性值为低; 若工具集成有生命周期管理的功能, 则该属性值为高。

TOOL 描述	编辑、写 代码、调 试功能	简单, 有前端, 有后台辅助工 具, 存在一点 集成	包含基本的生命 周期, 中等规模 集成	功能强大, 包 括成熟的生命 周期, 中等规 模集成	功能强大, 有 预测性的生命 周期, 与过程 集成良好, 可 复用	
等级	非常低	低	一般	高	非常高	极其高
EM	1.17	1.09	1.00	0.90	0.78	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

## 成本驱动因子-项目属性2

- 多站点开发(**SITE**): 随着多站点开发的普及和影响程度的逐步加大, COCOMO II模型中新增了SITE属性。通过确定两个因素的平均值可得出SITE值, 这两个因素分别为: 站点分布方式(从完全并级式到跨国分布式)和通信支持方式(从信件和电话交流到全方位交互式多媒体)。

SITE:站点分布方式	跨国	城市和公司	跨城市或跨公司	城市内	同一楼座	无距离
SITE:通信支持方式	电话、邮件	个人电话、传真	窄带电子邮件	宽带电子通信	宽带电子通信, 适当条件下可以视频通话	交互式, 多媒体
等级	非常低	低	一般	高	非常高	极其高
EM	1.22	1.09	1.00	0.93	0.86	0.80

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

## 成本驱动因子-项目属性3

- 对开发时间的要求(**SCED**): 该属性表示项目开发的时间约束。
- 该属性值的定义为: 对于给定工作量的项目, 工期延后或提前的工作量的百分比。在工期提前的情况下, 由于开发早期没有花费足够多的时间而遗留下的很多缺陷, 可能导致开发后期工作量加大。工期被压缩**75%**时, 该属性值为非常低。工期延后主要因为在开发早起花费了过多的时间以进行充分的规划、规范和验证。工期被延长**160%**时, 该属性值为非常高。

SCED 描述	正常进 度的 75%	正常进度的 85%	正常进度的 100%	正常进度的 130%	正常进度的 160%	
等级	非常低	低	一般	高	非常高	极其高
EM	1.43	1.14	1.00	1.00	1.00	n/a

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$



## 早期设计子模型和后体系架构子模型的驱动因子之间的关系为:

早期设计驱动因子	后体系架构驱动因子
产品可靠性和兼容性 ( <b>RCPX</b> )	RELY, DATA, CPLX, DOCU
重用性需求 ( <b>RUSE</b> )	RUSE
平台困难程度 ( <b>PDIF</b> )	TIME, STOR, PVOL
个人经验 ( <b>PREX</b> )	AEXP, PEXP, PCON
工具 ( <b>FCIL</b> )	TOOL, SITE
日程安排 ( <b>SCED</b> )	SCED
个人能力 ( <b>PERS</b> )	ACAP, PCAP, PCON





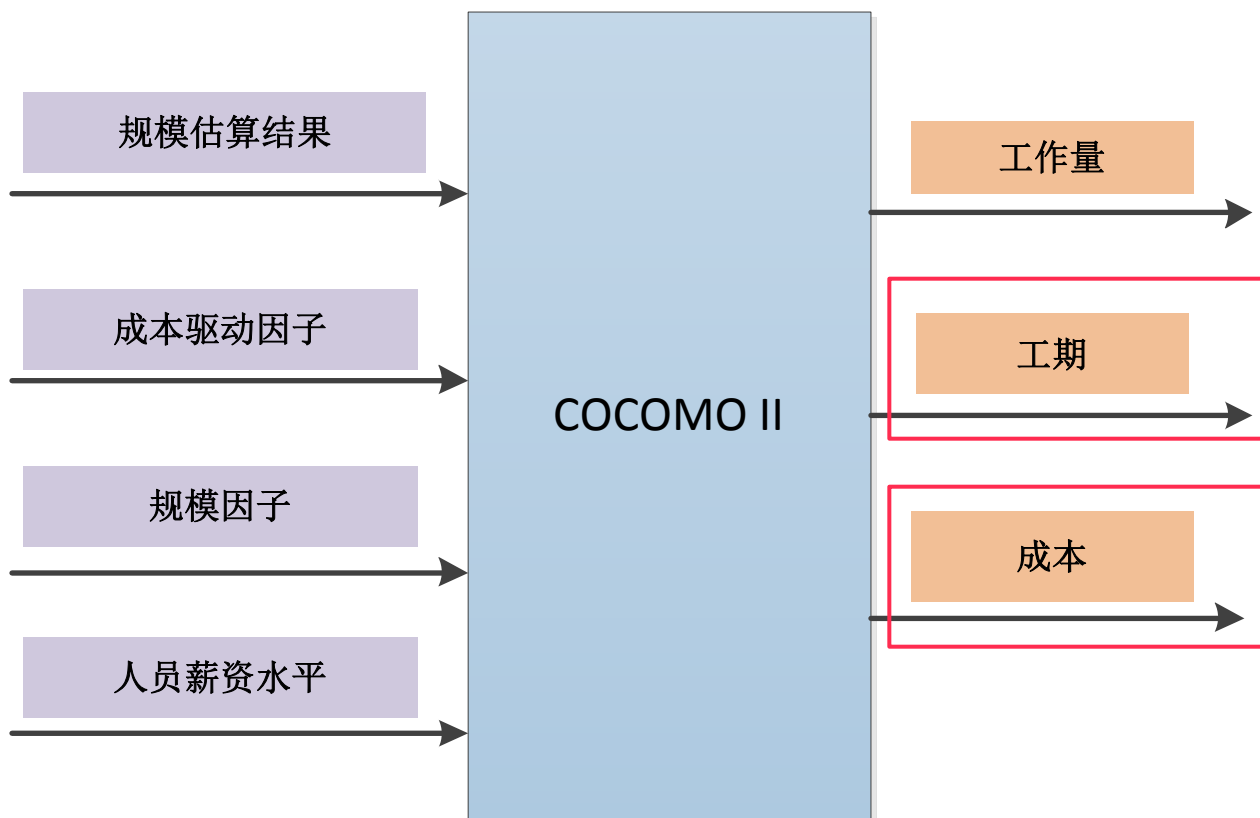
# 主要内容

---

- **COCOMO II模型介绍**
- 规模估算
- 工作量估算
- 规模因子
- 成本驱动因子
- **成本、工期估算方法**
- 工具演示



# COCOMO II模型示意图





## 工作量估算

**COCOMO II模型工作量计算公式为：**

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$
$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

- $A=2.94$ ,  $B=0.91$ 。
- $Size$ 代表以千行源代码（KLOC）表示的项目规模。
- $E$ 是五个规模因子（Scale Factor）共同产生的结果。
- $EM$ (Effort Multiplies)是工作量乘数，不同子模型涉及的 $EM$ 数目不同，早期设计子模型涉及到7个成本驱动因子（Cost Driver），后体系架构子模型涉及17个成本驱动因子。



## 工期估算

开发软件产品所需要的时间就是项目工期。估算项目工期的公式如下所示。

$$TDEV = C \times (PM)^F$$

$$F = D + 0.2 \times 0.01 \times \sum_{i=1}^5 SF_i$$

- C和D都是常量，分别为3.67和0.28。
- PM代表以人月为单位的工作量估算结果。
- TDEV表示以月为单位的项目工期。



## 成本估算

软件成本估算要考虑很多因素，比如硬件成本、劳动力成本、工具成本等等。要根据组织的特点确定成本估算的方法。软件开发的劳动力成本在总成本中所占的比重最大。得出工作量的估算结果（人月为单位）后，根据项目单个成员的薪资水平，就可以估算整个项目的成本。

COCOMO II中，工作量除以工期得到所需的人数。

$$Staffing = \frac{PM}{TDEV}$$

PM = 工作量的正常值 (人月)

TDEV = 项目工期



# 主要内容

---

- **COCOMO II模型介绍**
- 规模估算
- 工作量估算
- 规模因子
- 成本驱动因子
- 成本、工期估算方法
- **工具演示**



# Thanks!