



# Ch.33估计



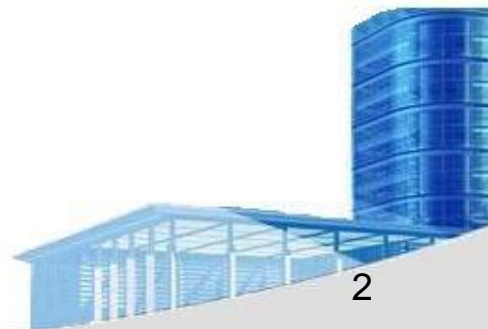


- 软件项目计划

项目计划的总体目标是建立一个实用的策略来控制、跟踪和监控一个复杂的技术项目。

为什么？

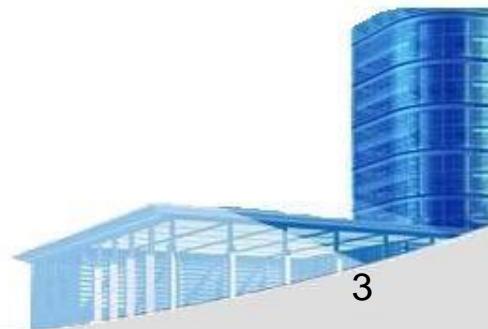
这样最终的结果就能按时、有质量地完成!





## • 项目规划任务一

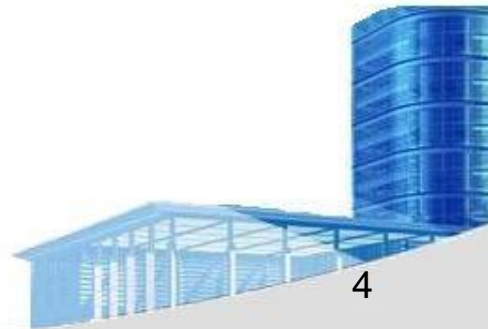
- 建立项目范围
- 确定可行性
- 分析风险
  - 第25章详细讨论了风险分析。
- 定义所需的资源
  - 确定需要人力资源
  - 定义可重用的软件资源
  - 确定环境资源





## • 项目计划任务集ii

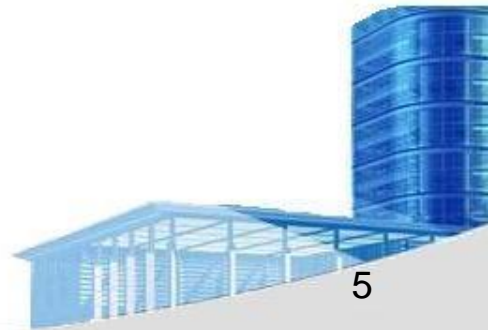
- 估算成本和工作量
  - 分解问题
  - 根据规模、功能点、过程任务或用例进行两个或更多的评估
  - 调和的估计
- 制定项目进度计划
  - 第34章详细讨论了进度安排。
    - 建立有意义的任务集
    - 定义一个任务网络
    - 使用安排工具来制定时间表
    - 定义计划跟踪机制





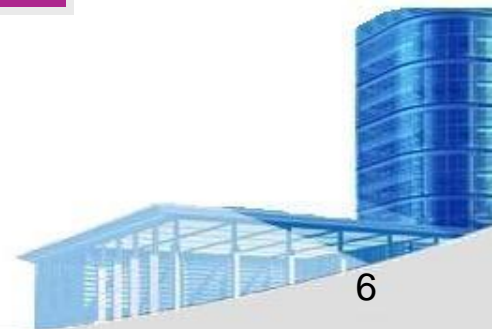
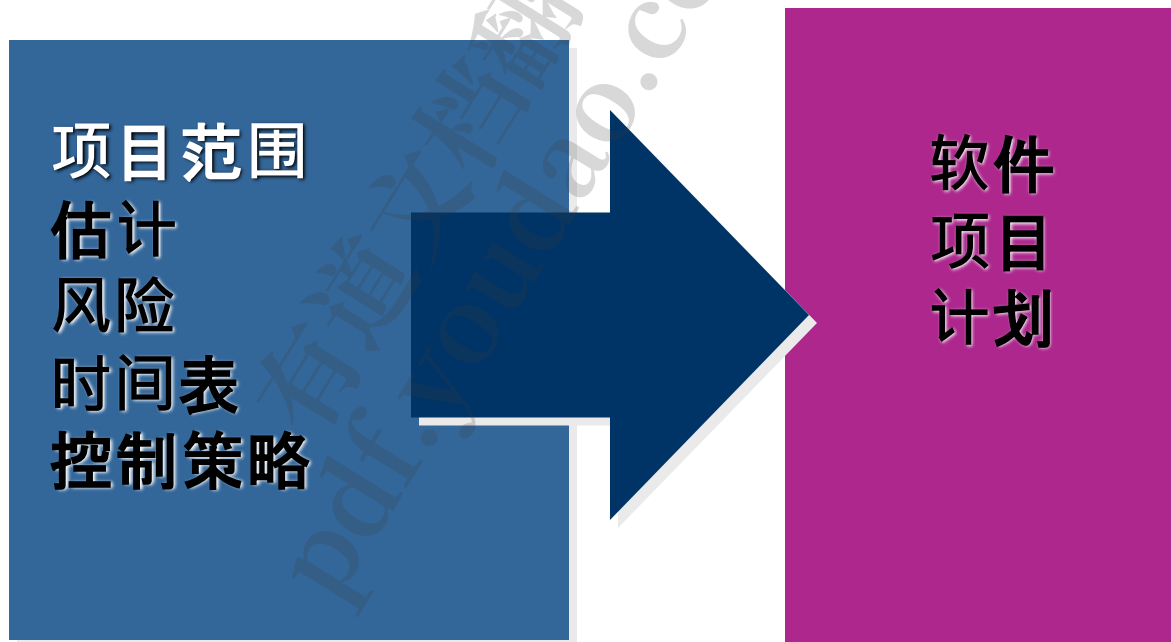
- 估计

- 对软件工程师工作所需的资源、成本和进度进行估计
  - 经验
  - 获得良好的历史信息(指标)
  - 在只有定性信息的情况下，敢于做出定量预测
- 估计带有固有的风险，而这种风险会导致不确定性





- 把它写下来!

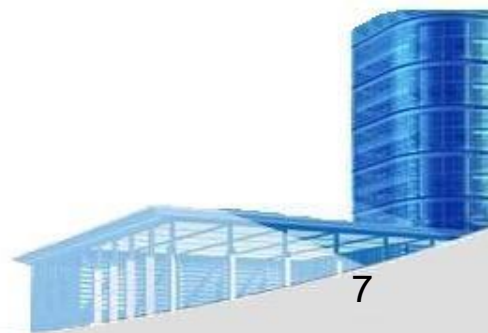




- 理解范围...

- 了解客户需求
- 了解业务背景
- 理解项目边界
- 了解客户的动机
- 了解可能的变化路径
- 明白.....

**即使你明白了,  
没有什么可靠的!**





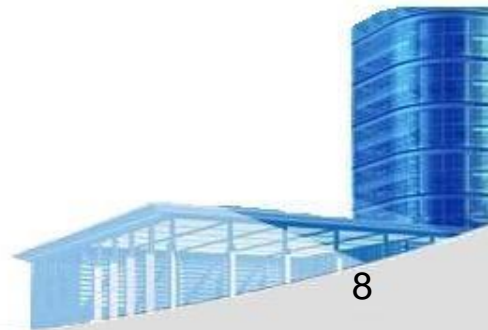
- 范围是什么？

- 软件范围描述

- 将要交付给最终用户的功能和特性
- 输入和输出的数据
- 作为使用软件的结果呈现给用户的“内容”
- 约束系统的性能、约束、接口和可靠性。

- 使用以下两种技术之一定义范围：

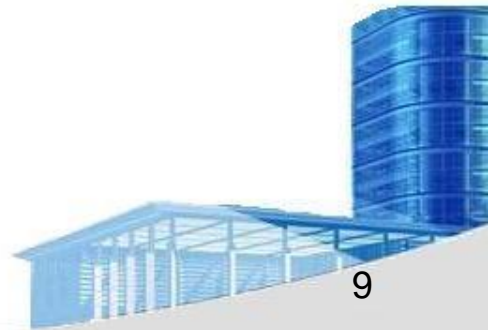
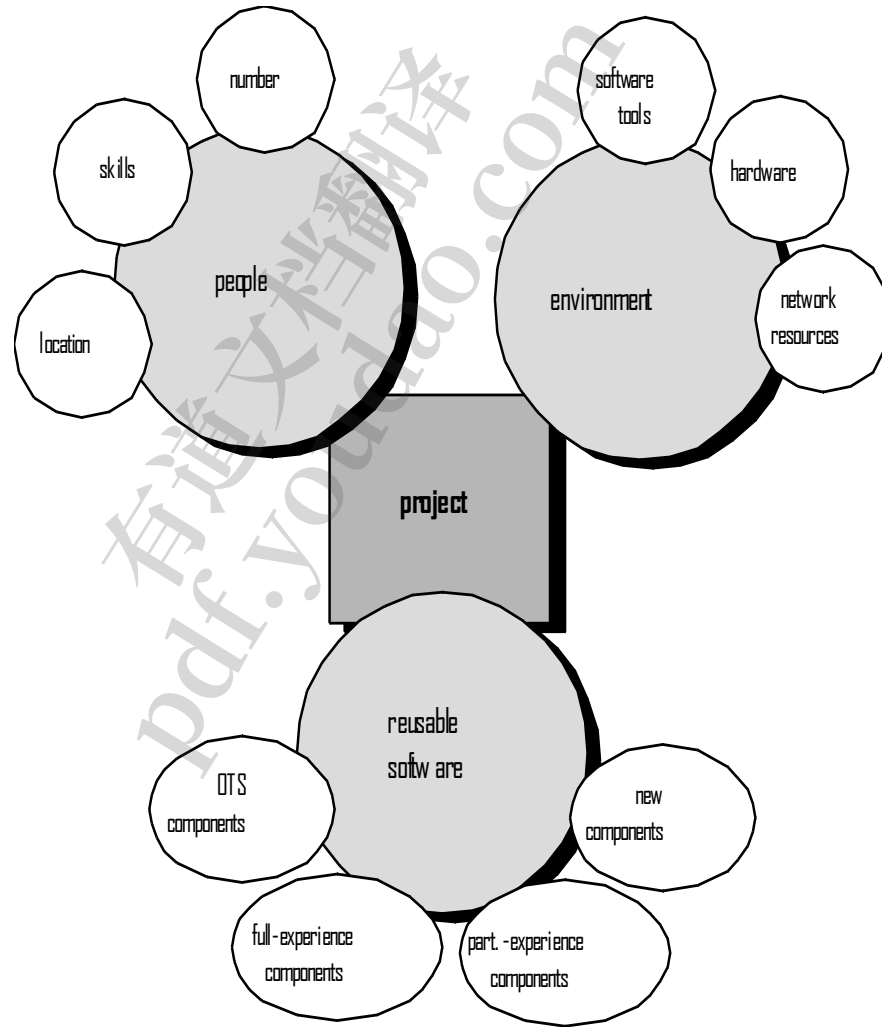
- 在与所有利益干系人沟通之后，开发出软件范围的叙述式描述。
- 一组用例由最终用户开发。







- 资源

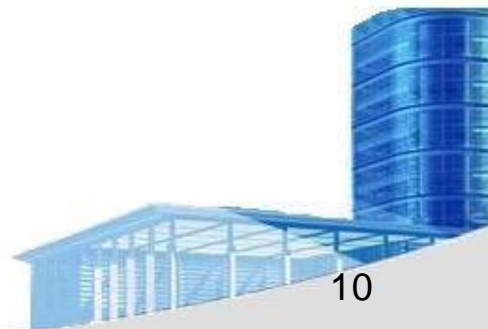




## • 项目评估



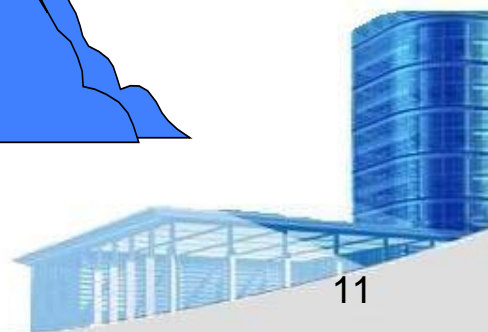
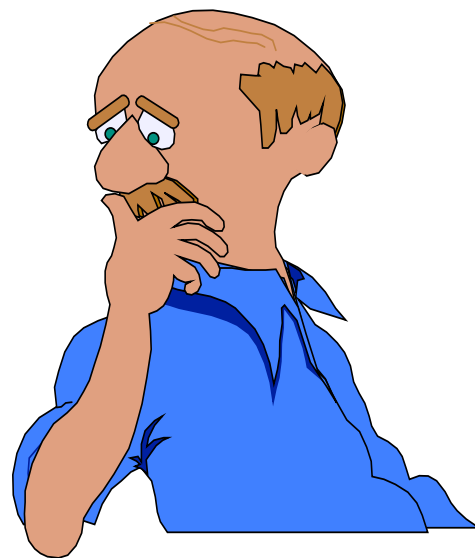
- 必须了解项目范围
- 精化(分解)是必要的
- 历史指标非常有用
- 至少应该使用两种不同的技术
- 不确定性是这个过程中固有的





## • 评估技术

- 过去(类似)的项目经验
- 传统的评估技术
  - 任务分解和工作量估算
  - 规模(例如, **FP**)估算
- 实证模型
- 自动化工具

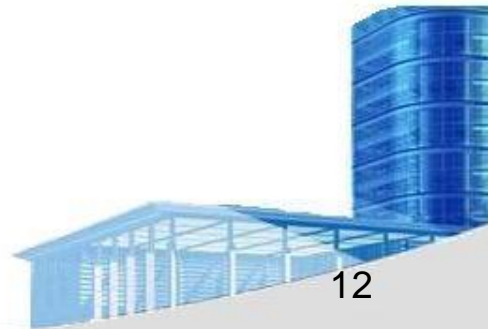




- 估计的准确性

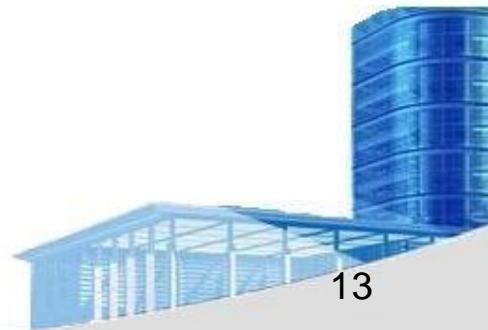
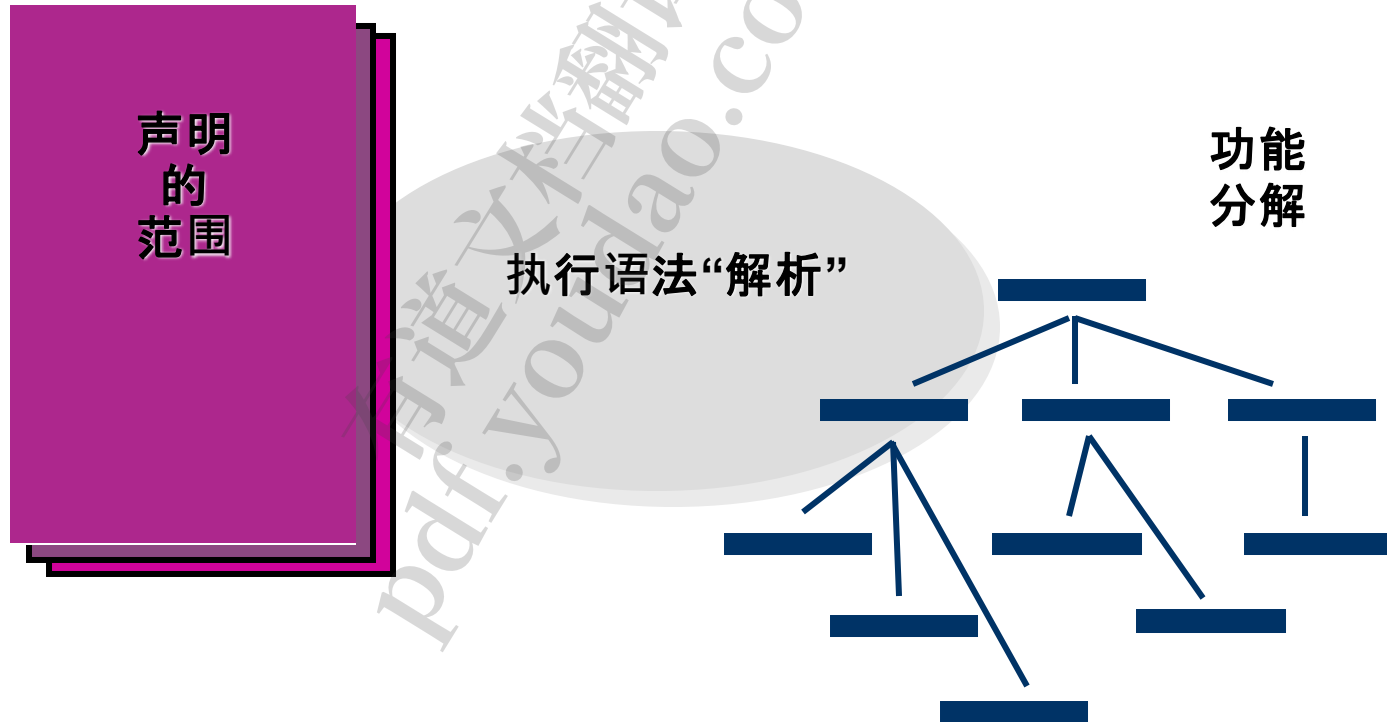
- 基于.....

- 策划者对将要建造的产品的大小做出正确估计的程度
    - 将规模估计转化为人力、日历时间和资金的能力(过去项目中可靠软件度量的可用性的函数)
    - 项目计划反映软件团队能力的程度
    - 产品需求的稳定性以及支持软件工程工作的环境。





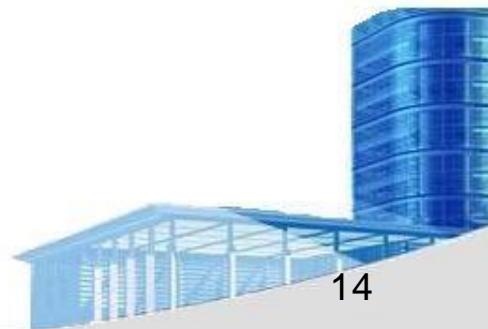
- 功能分解





- 常规方法:**LOC/FP**方法

- 使用估计的信息域值计算**LOC/FP**
- 使用历史数据对项目进行评估





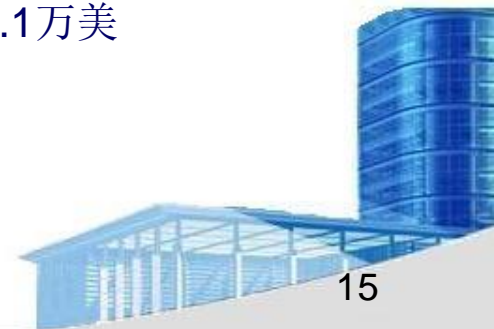
- 例子:疯狂的方法

Function	Estimated LOC
user interface and control facilities (UICF)	2,300
two-dimensional geometric analysis (2D GA)	8,300
three-dimensional geometric analysis (3D GA)	6,800
database management (DBM)	3,350
computer graphics display facilities (CGDF)	4,950
peripheral control (PC)	2,100
design analysis modules (DAM)	8,400
<i>estimated lines of code</i>	<b>33,200</b>

这种类型系统的平均生产力= 620 LOC/pm。

负担劳动率=\$8000 /月，每行代码的成本约为\$13。

根据LOC估算和历史生产力数据，项目总估算成本为43.1万美元，估算工作量为54人月。





- 例子:FP方法

Information Domain Value	opt.	lkely	poss.	est. count	weight	FP-count
number of inputs	20	24	30	24	4	97
number of outputs	12	18	22	16	8	78
number of inquiries	16	22	28	22	8	88
number of files	4	4	8	4	10	42
number of external interfaces	2	2	3	2	7	18
count-total						321

FP的估计数量是推导出来的:

$$\text{FP估计} = \text{count-total} \times [0.65 + 0.01 \times S(F_i)]$$

$$\text{FP预估} = 375$$

组织的平均生产力=6.5 FP/pm。

劳动负担率=每月8000美元，约为1230美元/FP。

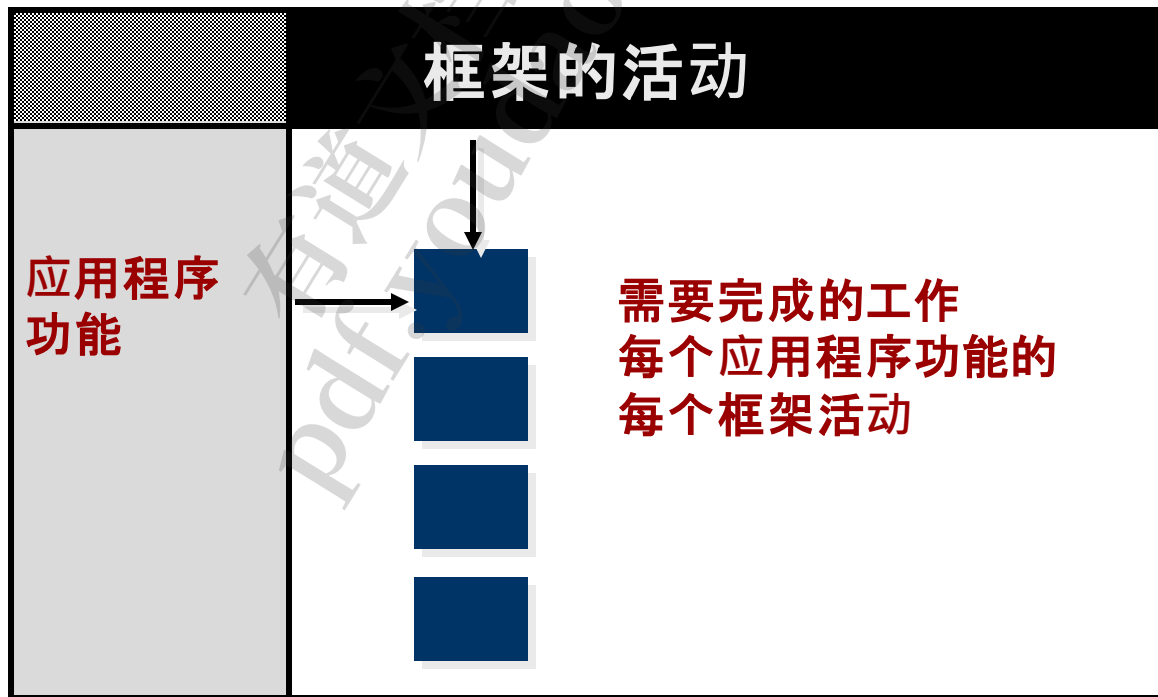
根据FP估算和历史生产力数据，项目总估算成本为46.1万美元，估算工作量为58人月。





- 基于流程的评估

从“过程框架”获得





- 基于流程的评估的例子

Activity →	CC	Planning	Risk Analysis	Engineering		Construction Release		CE	Totals
Task →				analysis	design	code	test		
Function ▼									
UICF				0.50	2.50	0.40	5.00	n/a	8.40
2DGA				0.75	4.00	0.60	2.00	n/a	7.35
3DGA				0.50	4.00	1.00	3.00	n/a	8.50
CGDF				0.50	3.00	1.00	1.50	n/a	6.00
DSM				0.50	3.00	0.75	1.50	n/a	5.75
PCF				0.25	2.00	0.50	1.50	n/a	4.25
DAM				0.50	2.00	0.50	2.00	n/a	5.00
Totals	0.25	0.25	0.25	3.50	20.50	4.50	16.50		46.00
% effort	1%	1%	1%	8%	45%	10%	36%		

CC = customer communication CE = customer evaluation

基于平均每月8000美元的劳动负担率，项目总成本估计为368000美元，工作量估计为46个人月。

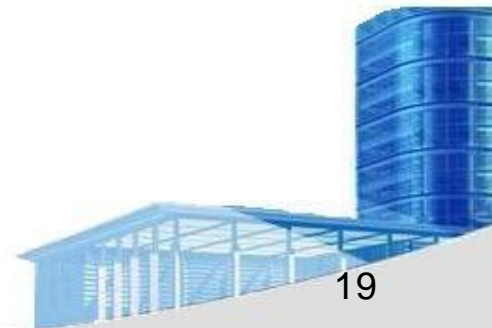
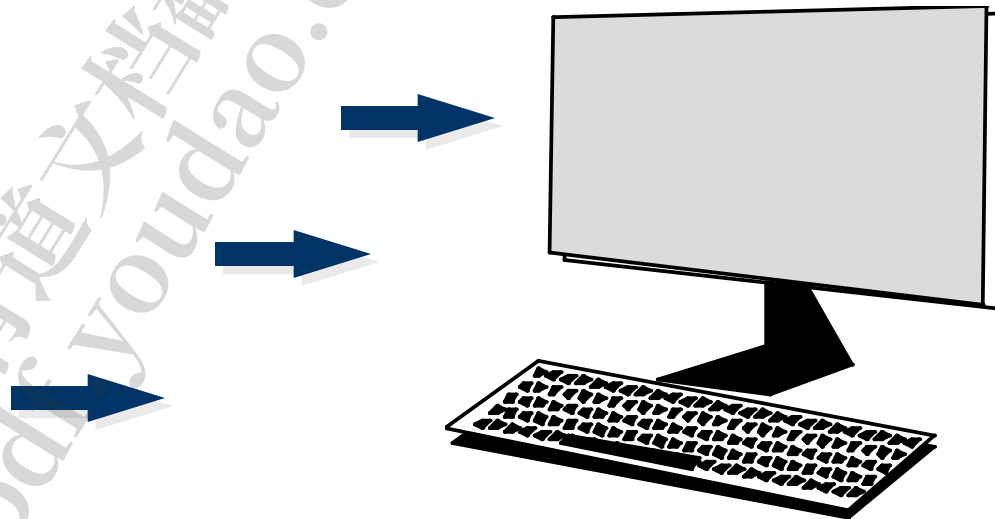


- 基于工具的评估

项目的特点

校准因素

LOC / FP数据

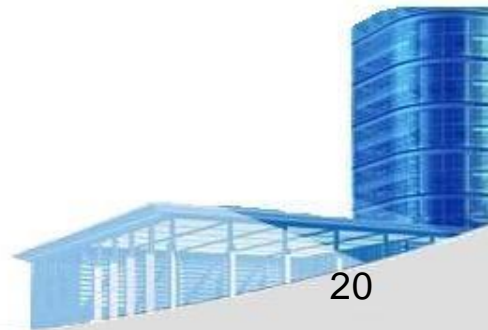




## • 估计与用例

	use cases	scenarios	pages	scenarios	pages	LOC	LOC estimate
User interface subsystem	6	10	6	12	5	560	3,366
Engineering subsystem group	10	20	8	16	8	3100	31,233
Infrastructure subsystem group	5	6	5	10	6	1650	7,970
Total							
LOC estimate							42,568

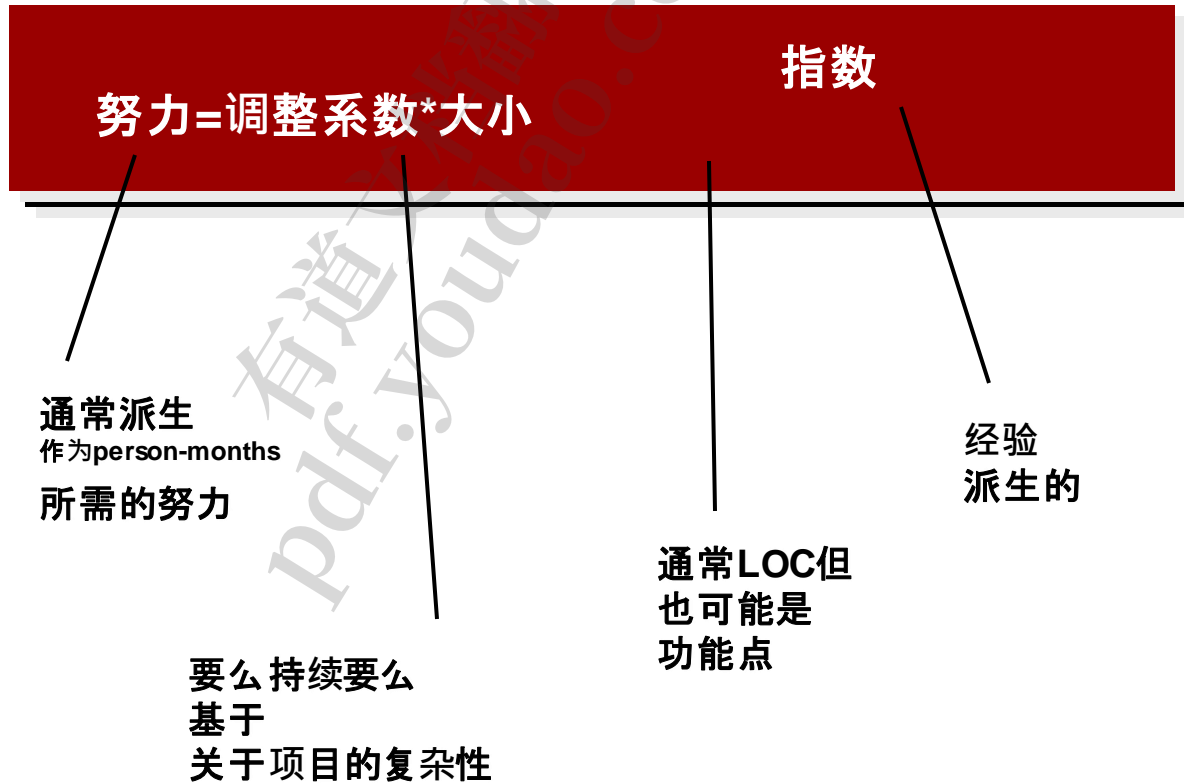
使用620 LOC/pm作为这种类型系统的平均生产力和每月8000美元的负担劳动率，每行代码的成本大约是13美元。基于用例估计和历史生产力数据，总的估计项目成本为55.2万美元，估计工作量为68人月。





## • 实证评估模型

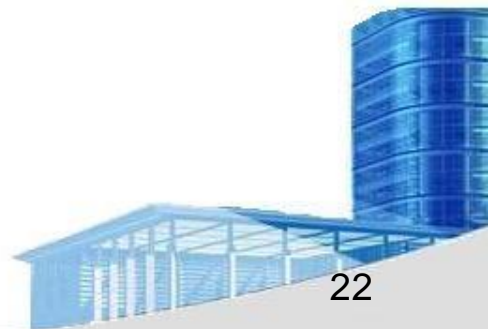
一般形式:





- **cocomo ii**

- **COCOMO II**实际上是一个评估模型的层次结构，它解决了以下领域：
  - **应用程序组合模型**。在软件工程的早期阶段使用，此时用户界面的原型设计、软件和系统交互的考虑、性能的评估和技术成熟度的评估是至关重要的。
  - **早期设计阶段模型**。一旦需求稳定下来，基本的软件架构已经建立起来，就可以使用。
  - **Post-architecture-stage模型**。在软件构建过程中使用。





- 软件方程

动态多变量模型

$$E = [LOC \times B^{0.333} / P]^3 \times (1/t^4)$$

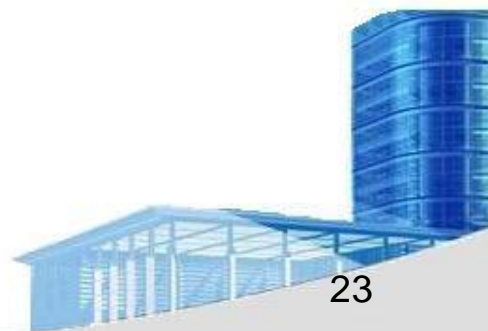
在哪里

E = 人月或人年付出的努力

T = 以月或年为单位的项目持续时间

B = “特殊技能因素”

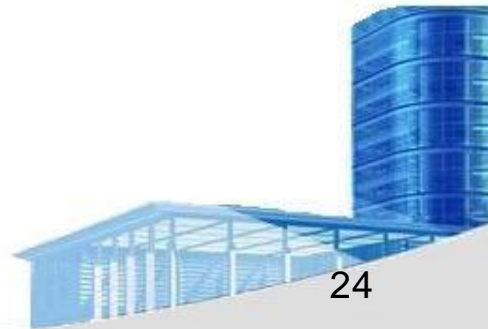
P = “生产力参数”





## • 面向对象项目的估计——i

- 使用工作量分解、FP分析和任何其他适用于常规应用的方法进行估算。
- 使用面向对象的需求建模(第6章), 开发用例并确定计数。
- 从分析模型中, 确定关键类的数量(在第6章中称为分析类)。
- 对应用程序的接口类型进行分类, 并为支持类开发一个乘数:
  - 接口类型的乘数
  - 没有GUI 2.0
  - 基于文本的用户界面2.25
  - GUI 2.5
  - 复杂gui 3.0

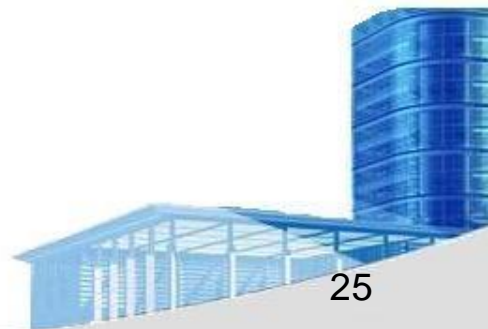






- 面向对象项目的估算- ii

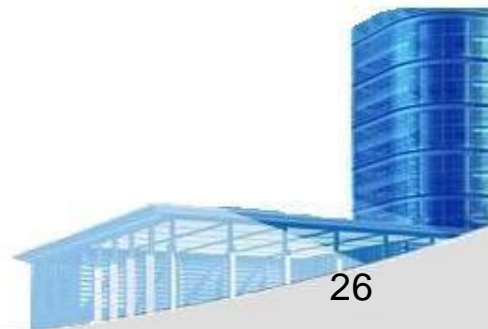
- 将关键类的数量(步骤3)乘以乘数, 得到支持类数量的估计。
- 将总类数(key + support)乘以每个类的平均工作单元数。洛伦兹和基德建议每个班15 ~ 20人日。
- 通过乘以每个用例的平均工作单元数, 交叉检查基于类的估计





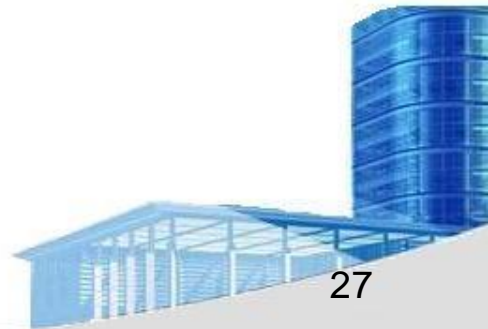
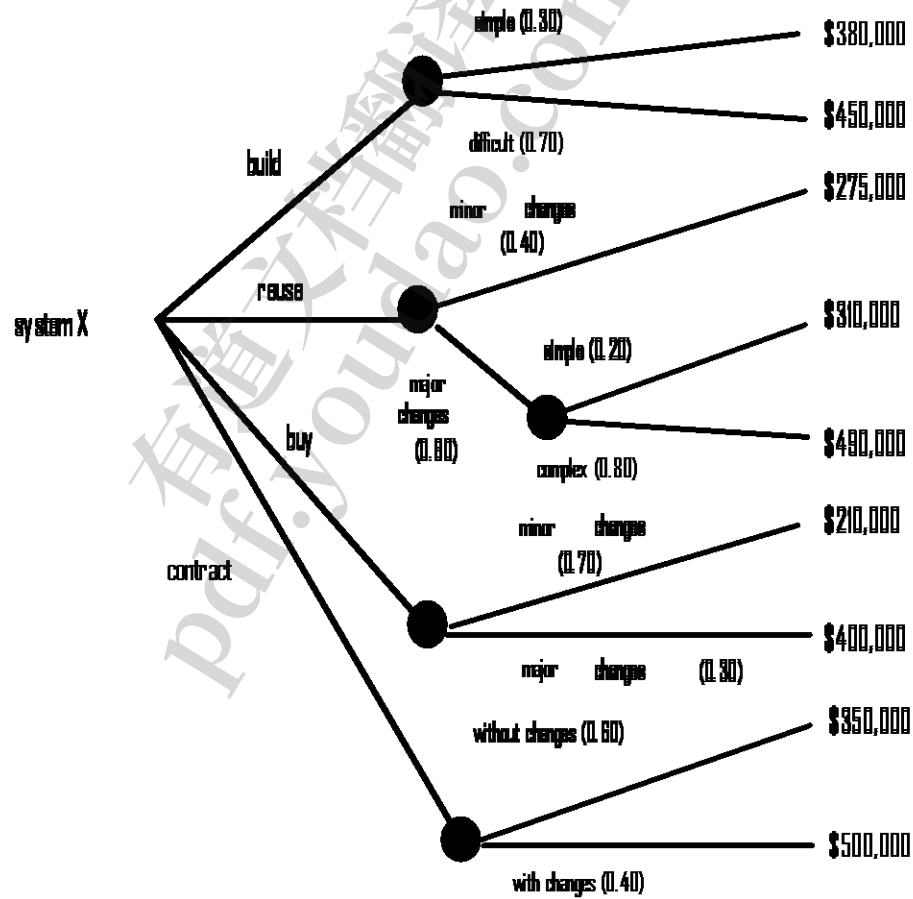
## • 敏捷项目的估算

- 为了估计的目的，每个用户场景(一个小型用例)都被单独考虑。
- 场景被分解为开发它所需的一组软件工程任务。
- 每个任务都是单独估计的。注意:估计可以基于历史数据、经验模型或“经验”。
  - 或者，场景的“体量”可以在LOC, FP或其他一些面向体量的度量(例如，用例计数)中进行估计。
- 将对每个任务的估计进行汇总，以创建场景的估计。
  - 或者，使用历史数据将场景的体积估计转化为工作量。
- 对于给定的软件增量所要实现的所有场景的工作量估计可以进行汇总，以开发增量的工作量估计。





## • Make-Buy决定





- 计算预期成本

预期成本=

$$\sum (\text{路径概率}) \times (\text{估计的路径代价})$$

我

我

例如，预期构建成本为:预期成本构建=  $0.30(38\text{万美元}) + 0.70(45\text{万美元}) = 42.9\text{万美元}$  同样，预期成本重用=  $38.2\text{万美元}$  预期成本购买=  $26.7\text{万美元}$   
预期成本控制=  $41\text{万美元}$

