



# 软件项目估算

---

清华大学软件学院 刘强



# 软件项目估算

---

项目估算是对完成项目交付物的时间和成本进行预算和估计的过程。



- 软件规模越大，复杂性越高，不确定性就越大
- 需求的不确定性会对项目估算产生很大影响
- 没有可靠的历史数据使项目估算缺少参照物

# 软件项目估算

---



项目成员不应该被估算所困扰，勇于面对软件项目估算的挑战，克服其中的困难，做出一个相对的有价值的估算。

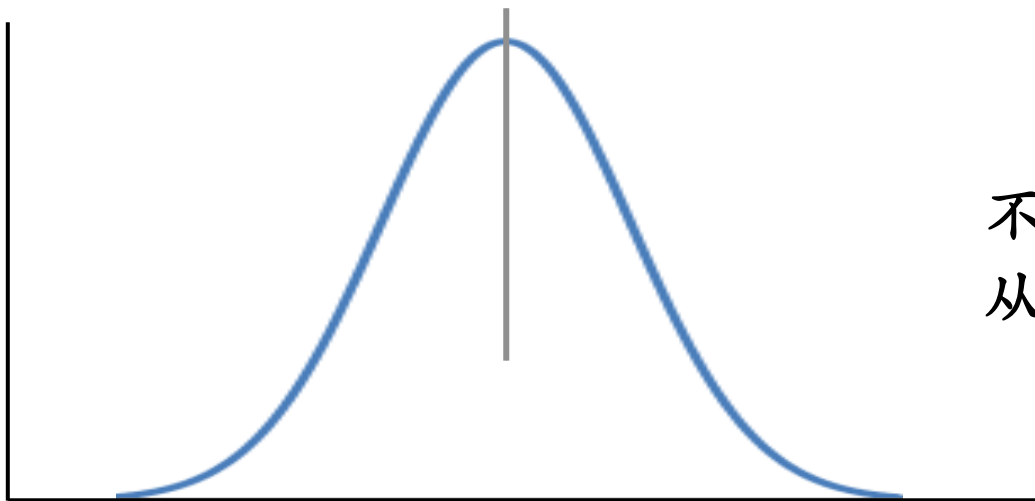


应该满足于事物的本性所能容许的精确度，当只能近似于真理时，不要去寻求绝对的准确。

—— Aristotle

# 软件项目估算

软件项目估算的首要原则：对结果进行估计，而不是活动。



不论用什么方法，所有估计  
从定义上来说都只是概率。

# 基本估算方法

---

**专家判断：**通过借鉴历史信息，专家提供项目估算所需的信息，或根据以往类似项目的经验，给出相关参数的估算上限。



# 基本估算方法

---

**参数估算**：通过对大量的项目历史数据进行统计分析，使用项目特性参数建立经验估算模型，估算诸如成本、预算和持续时间等活动参数。



功能点

COCOMO

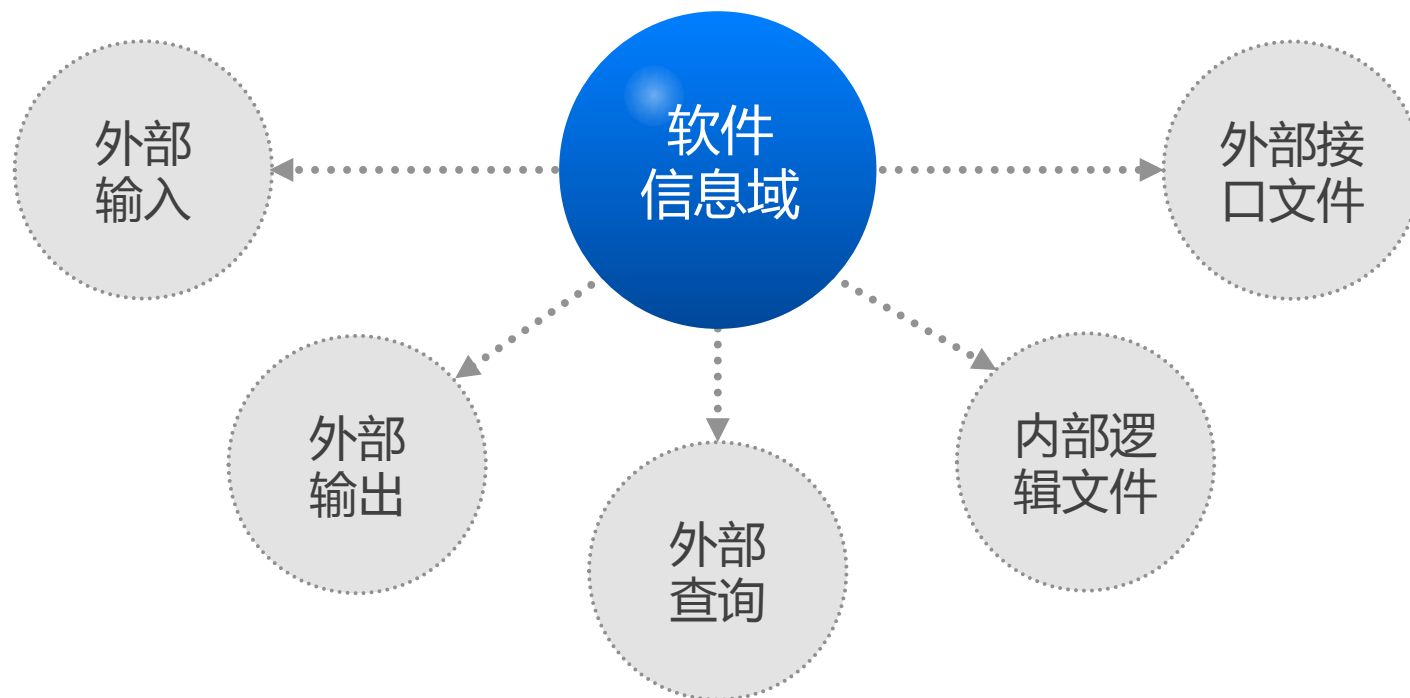
用例点

机器学习

# 功能点方法



功能点方法是依据软件信息域的基本特征和对软件复杂性的估计，估算出软件规模。这种方法适合于在开发初期进行估算，并以功能点为单位度量软件规模。



# 功能点方法



信息域加权因子：

信息域参数	加权因子			合计
	简单	中等	复杂	
外部输入	3	4	6	$\Sigma$
外部输出	4	5	7	$\Sigma$
外部查询	3	4	6	$\Sigma$
内部逻辑文件	7	10	15	$\Sigma$
外部逻辑文件	5	7	10	$\Sigma$
未调整功能点 UFC				$\Sigma$

系统复杂度调整值  $F_i$ ：取值 0..5

$F_1$	可靠的备份和恢复	$F_8$	在线升级
$F_2$	数据通信	$F_9$	复杂的界面
$F_3$	分布式处理	$F_{10}$	复杂的数据处理
$F_4$	性能	$F_{11}$	代码复用性
$F_5$	大量使用的配置	$F_{12}$	安装简易性
$F_6$	联机数据输入	$F_{13}$	多重站点
$F_7$	操作简单性	$F_{14}$	易于修改

功能点计算： $FP = UFC \times [ 0.65 + 0.01 \times \sum F_i ]$



# COCOMO模型

结构性成本模型 COCOMO ( COnstructive COst MOdel ) 是一种利用经验模型进行工作量和成本估算的方法。

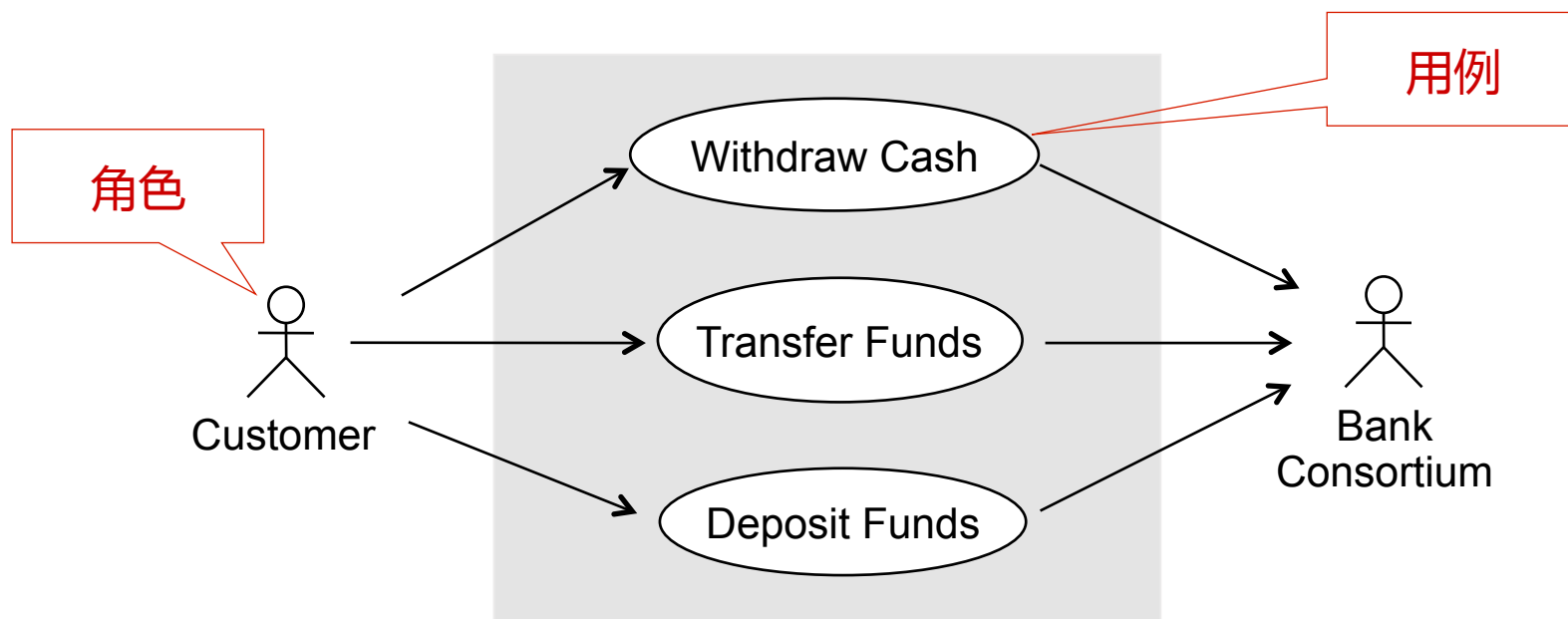
$$PM_{\text{nominal}} = A * (\text{Size})^B$$

- PMnominal : 人月工作量
- A : 工作量调整因子
- B : 规模调整因子
- Size : 规模, 单位是千行代码或功能点数

类型	A	B	说明
组织型	2.4	1.05	相对小的团队在一个高度熟悉的内部环境中开发规模较小, 接口需求较灵活的系统。
嵌入型	3.6	1.2	开发的产品在高度约束的条件下进行, 对系统改变的成本很高。
半独立型	3.0	1.12	介于上述两者中间

# 用例点估算

用例点估算是在面向对象软件开发项目中用于估计规模和工作量的方法，它比功能点方法要简单一些。



# 用例点估算



## 第 1 步：计算角色复杂度

角色 复杂度	说明	权重	总计
简单	角色是通过API或接口与系统进行交互的其他系统	1	$\Sigma$
一般	角色是通过协议（如TCP/IP）与系统进行交互的其他系统	2	$\Sigma$
复杂	角色是通过GUI或Web界面与系统进行交互的人	3	$\Sigma$
角色复杂度UAW			$\Sigma$

# 用例点估算



## 第 2 步：计算用例复杂度

用例 复杂度	说明	权重	总计
简单	仅涉及1个数据库实体；操作不超过3步；实现用到5个类以下	5	$\Sigma$
一般	涉及2个或以上数据库实体；操作在4-7步；实现用到5-10个类	10	$\Sigma$
复杂	复杂的用户界面或涉及3个或以上数据库实体；操作超过7步；实现用到超过10个类	15	$\Sigma$
用例复杂度UUCW			$\Sigma$

# 用例点估算

第 3 步：计算未平衡用例点  $UUCP = UAW + UUCW$

第 4 步：用技术复杂度因子TCF和环境复杂度因子ECF进行调整，得到用例点

$$UCP = TCF * ECF * UUCP$$

**说明：**技术复杂度因子TCF和环境复杂度因子ECF计算见后面表格

第 5 步：估算项目开发工作量

只要给出基于每个UCP完成的时间，就可以计算出项目开发工作量

$$工作量 = UCP * 生产率$$

**建议：**每个UCP为16-30人时，均值为20人时

# 用例点估算



技术复杂度因子  $TF_i$  : 取值 0..5

TFC	说明	权重	TFC	说明	权重
$TF_1$	分布式系统	2	$TF_8$	可移植性	2
$TF_2$	系统性能要求	1	$TF_9$	可修改性	1
$TF_3$	终端用户使用效率要求	1	$TF_{10}$	并发性	1
$TF_4$	内部处理复杂度	1	$TF_{11}$	特殊的安全性	1
$TF_5$	可重用性	1	$TF_{12}$	提供给第三方接口	1
$TF_6$	易安装性	0.5	$TF_{13}$	需要特别的用户培训	1
$TF_7$	易用性	0.5			

$$TFC = 0.6 + 0.01 \times \sum TF_i$$

# 用例点估算

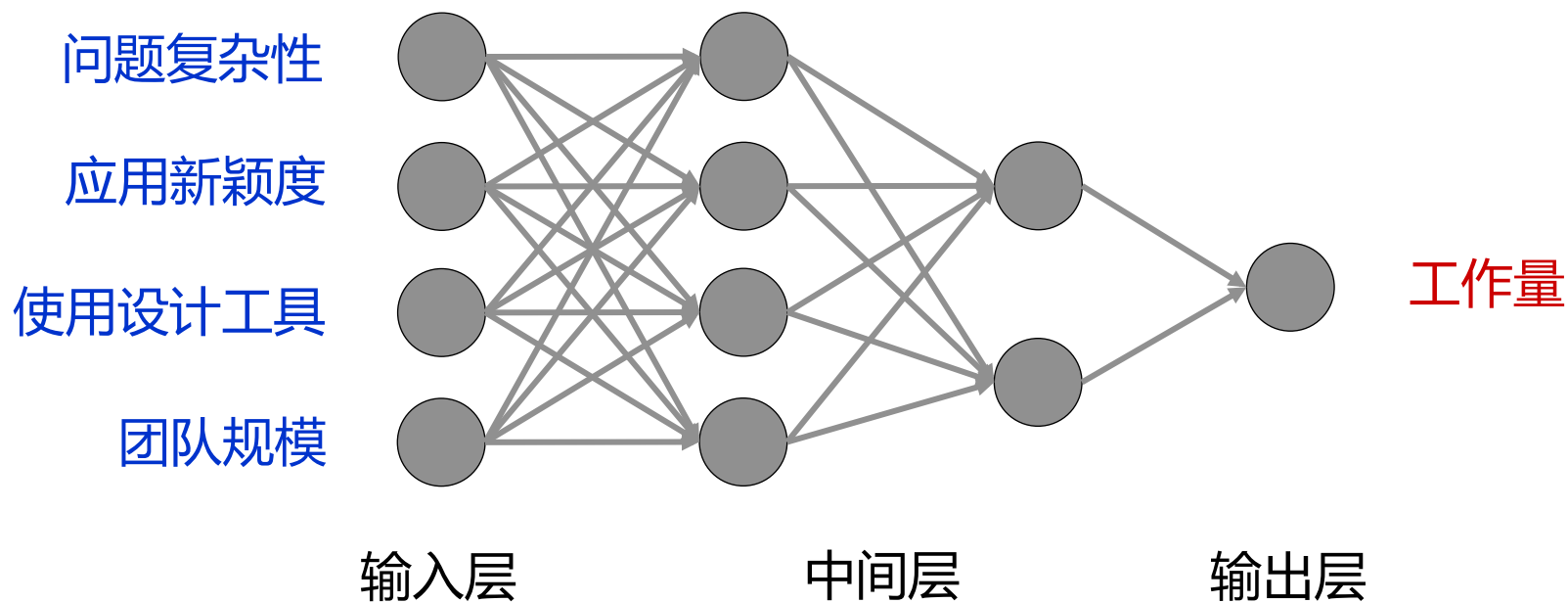
环境复杂度因子  $EF_i$  : 取值 0..5

EFC	说明	权重	TFC	说明	权重
$EF_1$	UML精通程度	1.5	$EF_5$	团队士气	1
$EF_2$	开发应用系统经验	0.5	$EF_6$	需求稳定性	2
$EF_3$	面向对象经验	1	$EF_7$	兼职人员比例	-1
$EF_4$	系统分析员能力	0.5	$EF_8$	编程语言难易程度	2

$$EFC = 1.4 + ( -0.03 \times \sum EF_i )$$

# 机器学习方法

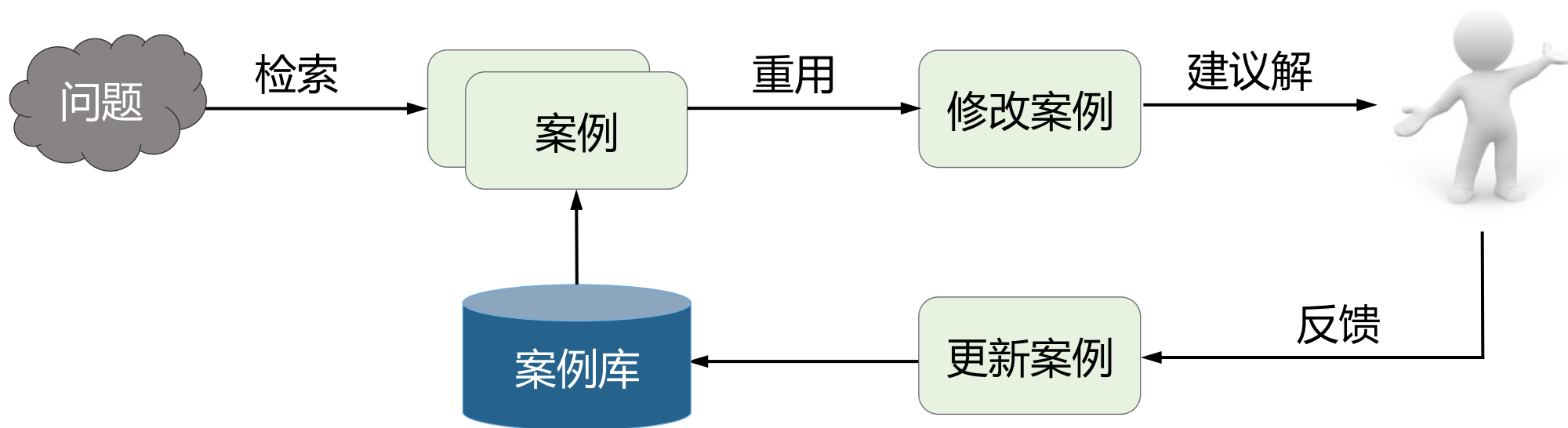
神经网络是采用一种学习方法导出一种预测模型，这种方法使用历史项目数据训练网络，通过不断学习找出数据中的规律，再用其估算新项目的工作量。





# 机器学习方法

基于案例的推理方法可以用于基于类推的估算，即识别出与新项目类似的案例，再调整这些案例，使其适合新项目的参数。





# 谢谢大家！

---

## THANKS

