

# 软件工作量估计方法比较

## 1. 算法模型与专家判断

算法模型依赖于历史数据和统计公式来预测工作量,适合于有大量历史项目数据可供参考的情况。COCOMO 模型(属于算法模型)通过公式  $Effort = c * size^k$  来估算工作量,其中  $c$  和  $k$  是模型参数,  $size$  是项目规模的度量。

专家判断依靠领域专家的经验 and 直觉,适用于缺乏历史数据或项目非常独特、难以量化的情况。Delphi 方法就是一种专家判断技术,通过匿名收集专家意见并迭代收敛到一致的估计结果。

## 2. 类比估计

类比估计基于与历史项目或类似项目的比较,适用于新项目与历史项目在多个特征上相似的情况。这种方法简单易行,但依赖于找到合适的类比项目。

## 3. 由低向上估计与自顶向下估计

由低向上估计从项目的最基本任务开始,逐步向上汇总工作量,适合于项目规划的后期,当项目细节较为清晰时。

自顶向下估计从项目整体出发,逐步细化到各个部分,适合于项目早期阶段,当对项目整体有较好把握但细节尚不明确时。

## 4. 功能点方法

功能点方法(如 Albrecht、COSMIC、Mark II)关注软件的功能规模,通过计算输入、输出、查询等软件的基本信息量,并结合环境复杂性因子来估算软件规模。这种方法适用于业务信息系统,强调数据方面的因素。

COSMIC 方法适用于实时系统或嵌入式系统,通过识别“数据移动”的个数来度量软件规模,每个数据移动计为一个 COSMIC 功能规模单位。

## 5. 对象点方法

对象点方法考虑应用所需处理的屏幕、报告和部件,将它们分为简单、中等或困难三个层次。这种方法类似于功能点方法,但更侧重于对象的复杂性。

## 6. NESMA 方法

NESMA 方法基于 IFPUG 发展而来,适用于需求不完善时的快速估算,分为指示法、估算法和详细法,适用于不同阶段的软件规模和成本估算。

## 7. FiSMA 方法

FiSMA 方法面向服务的方法,基本功能部件类 BFC,涵盖了互联网服务的各种系统,适用于新时代的软件开发。

## 8. 参数化模型 COCOMO II

COCOMO II 模型是 COCOMO 的改进版,适应了项目在执行过程中越来越确定的状态,是一种渐进性评估模型。它分为应用构成、早期设计和构造阶段,每个阶段使用不同的乘法算子和指数。

这些软件工作估量方法各有优势和局限，选择哪种方法取决于项目的具体需求、可用数据、项目阶段以及项目团队的偏好和经验。

对于有大量历史数据的项目，算法模型可能更为合适；而对于独特或创新的项目，专家判断可能更为可靠；功能点方法和对象点方法更适用于关注软件功能和对象复杂性的项目。