# 开发工作量估计主要方法与比较

## 1. 软件工作量估计的定义与重要性

定义：软件工作量评估是指对软件开发项目进行工作量的量化估计，用于确定开发项目所需的资源和时间投入。

重要性：贴合实际的估计对商业活动和项目负责人至关重要，有助于系统按时、按预算交付，并满足要求的质量。

## 2. 软件工作量估计的难点

规模大：软件系统的规模越大，估计的难度越高。

复杂性高：软件系统的复杂性增加，层次多，不可见性高，导致估计困难。

估计的主观性：人们容易低估小项目的工作量，而过分夸大大项目的工作量。

技术变化快：技术更新迅速，缺乏一致的项目经验。

## 3. 软件工作量估计的方法

#### 3.1 算法模型

COCOMO模型：由Barry Boehm提出，包括基本的COCOMO模型和COCOMO II模型。COCOMO II模型考虑了项目的不确定性和风险，是一种渐进性评估模型。

参数化模型：如COCOMO模型，通过公式`Effort = c size^k`来估算工作量，其中`c`和`k`是系数，`size`是软件规模。

#### 3.2 专家判断

Delphi法：一种专家评估技术，通过专家们的反复讨论以达成一致的估算结果。

Wideband Delphi：Delphi法的一种变体，通过多轮评估来细化估计。

#### 3.3 类比估计

基于案例的推理：评估者寻找已经完成的项目，这些项目与需要开发的新项目在许多特征上必须是类似的。

#### 3.4 功能点方法

IFPUG：国际功能点用户小组提出的方法，通过计算软件的基本信息量和环境复杂性因子来估算功能点。

COSMIC：全功能点方法，通过识别“数据移动”的个数来度量软件规模。

Albrecht方法：计算方法的基本思想为首先计算软件的五个基本信息量，然后对其通过环境复杂性因子作修正。

#### 3.5 对象点方法

Object Points：类似于功能点方法，但是更容易计算，考虑应用所需要处理的屏幕、报告和部件。

#### 3.6 NESMA方法

NESMA：基于IFPUG发展而来，2003年升级为ISO/IEC国际标准。

#### 3.7 FiSMA方法

FiSMA：面向服务的方法，基本功能部件类base functional component class; BFC。

## 估计方法优缺点

#### 4.1 算法模型（如COCOMO）

优点：

提供了一个基于经验的量化框架。

COCOMO II模型能够适应项目进展，进行渐进式估计。

缺点：

需要大量的历史数据来校准模型。

对于小型或者独特项目可能不够准确。

#### 4.2 专家判断（如Delphi法）

优点：

利用专家的知识和经验，适合于缺乏历史数据的情况。

通过多轮迭代，可以减少个体偏见的影响。

缺点：

依赖于专家的可用性和质量。

可能受到专家主观性的影响。

#### 4.3 类比估计

优点：

简单易行，适合早期项目估计。

可以快速得出一个粗略的估计。

缺点：

依赖于类似项目的可用性和准确性。

对于与历史项目差异较大的新项目，估计可能不准确。

#### 4.4 功能点方法（如IFPUG、COSMIC、Albrecht方法）

优点：

基于功能的用户价值，更加客观。

可以跨项目和组织进行比较。

缺点：

需要专业的功能点分析师，对分析师的依赖性高。

对于非功能性需求的考虑可能不足。

#### 4.5 对象点方法

优点：

相对于功能点方法，对象点方法更易于理解和应用。

适合于面向对象的软件开发环境。

缺点：

可能不如功能点方法那样成熟和广泛认可。

对于非面向对象的项目可能不适用。

#### 4.6 NESMA方法

优点：

基于IFPUG，但更适合于计划阶段，尤其是在需求不完善时。

提供了不同的估算方法（指示法、估算法、详细法）以适应不同阶段的需求。

缺点：

需要对NESMA方法有深入的理解才能准确应用。

对于小型项目可能过于复杂。

#### 4.7 FiSMA方法

优点：

面向服务的方法，适合于现代服务导向架构。

考虑了互联网服务的多样性。

缺点：

相对较新，可能缺乏广泛的行业接受度。

对于传统项目可能不适用。

**综合比较：**

数据依赖性：算法模型和功能点方法依赖于历史数据，而专家判断和类比估计则依赖于专家经验和类似项目。

适用性：功能点方法和对象点方法适用于功能驱动的项目，而NESMA和FiSMA方法适用于服务导向的项目。

复杂性：功能点方法和NESMA方法在应用上可能较为复杂，需要专业的培训和理解。

准确性：在有足够历史数据和专家支持的情况下，这些方法可以提供较为准确的估计，但在数据不足或专家意见不一致时，准确性会受到影响。

## 5. 结论

软件工作量估计是一个复杂的过程，涉及多种方法和技术。选择合适的估计方法需要考虑项目的特点、需求的详细程度以及可用的历史数据。通过综合运用不同的估计方法，并结合专家经验和历史数据，可以提高软件工作量估计的准确性和可靠性。