



****

**软件项目管理课程作业**

**题 目:**  软件工作量估计方法的比较

**学 院 ：** 软件学院

**学 号：**  2021141090183

**姓 名：**  付文君

**联系电话：** 17313194059

### 1. ****专家估计法（Expert Judgment）****

专家估计法依赖于具有相关经验的专家，根据他们对类似项目的经验进行判断，估算出项目的工作量。这种方法常用于项目初期，或者当缺乏足够的数据支持时。

**优点**：

* **快速**：适用于项目初期或需求不明确时，能够快速得出估算。
* **经验丰富**：对有经验的专家来说，能够通过类比、直觉和经验提供有价值的估算。
* **灵活性强**：可以在任何阶段使用，并且可以根据项目的具体需求进行调整。

**缺点**：

* **主观性强**：容易受到专家个人偏见、经验的局限性或过度乐观的影响。
* **缺乏一致性**：不同的专家可能给出不同的估算结果，缺乏标准化。
* **依赖于专家经验**：对于新领域或缺乏经验的团队来说，这种方法可能不够准确。

**适用场景**：

* 项目初期需求不明确或不完整的情况下。
* 小型项目或不确定性较高的项目。

### 2. ****类比估算（Analogous Estimating）****

类比估算是一种通过参考历史类似项目的数据来估算当前项目工作量的方法。它依赖于过去类似项目的经验，使用这些历史数据来进行估算。

**优点**：

* **快速**：能够基于过往经验迅速得出估算。
* **简单直观**：通过对比历史项目，易于理解和使用。
* **低成本**：不需要复杂的工具或方法。

**缺点**：

* **依赖历史数据**：如果历史项目与当前项目差异较大，类比估算的准确性就会降低。
* **不适用新领域**：对于没有类似项目经验的新领域或技术，类比估算效果不好。
* **可能误差较大**：不同项目的规模、技术环境等差异，可能导致误差较大。

**适用场景**：

* 在有丰富历史数据支持的项目中使用。
* 类似的过去项目可以提供可靠的估算依据。

### 3. ****功能点估算法（Function Point Estimation）****

功能点估算法基于软件的功能需求来估算工作量。功能点的数量和复杂度决定了项目的规模和工作量。功能点通常包括输入、输出、查询、内部逻辑文件（ILF）和外部接口文件（EIF）。

**优点**：

* **客观性强**：通过明确的标准化方法度量功能需求，相对客观。
* **技术无关**：不依赖于开发平台或编程语言。
* **适用于复杂应用**：适用于复杂的系统开发，特别是具有明确功能需求的系统。

**缺点**：

* **依赖需求分析**：需要准确的需求分析文档，对于需求不明确的项目不适用。
* **对非功能需求忽略较多**：功能点方法主要关注功能需求，非功能需求（如性能、安全性等）没有得到充分考虑。
* **需要专业知识**：功能点的计数和评估需要专业的知识和经验。

**适用场景**：

* 需求明确、功能导向的项目。
* 大型企业级应用、数据库密集型项目。

### 4. ****COSMIC-FFP（Common Software Measurement International Consortium - Functional Flow Points）****

COSMIC-FFP 是一种基于功能流（Functional Flow）进行度量的方法。它通过衡量系统功能数据的流动来估算工作量，考虑了输入、输出、数据存储和数据处理四个维度，尤其适用于复杂的、数据流密集型的系统。

**优点**：

* **精细化度量**：相比传统的功能点方法，COSMIC-FFP考虑了数据流和功能处理能力，更适用于复杂和实时系统。
* **适应性强**：适用于多种开发环境，尤其在实时系统和嵌入式系统中表现良好。
* **全面性**：除了功能，还考虑了数据流的复杂度，能够更全面地反映系统工作量。

**缺点**：

* **学习曲线较陡**：COSMIC-FFP需要对数据流的定义和度量有深刻理解，相较于传统功能点法更复杂。
* **需要详细的需求文档**：需要有详细的功能描述和数据流定义，需求不明确时难以使用。

**适用场景**：

* 实时系统、嵌入式系统和复杂的数据库密集型应用。
* 需要对功能流和数据处理进行详细度量的项目。

### 5. ****MK II（Mark II Function Point Method）****

MK II 是功能点分析的一个改进版，提供了一种更精细化的功能点计数方法。它对功能点进行更细致的复杂度评估，以便更准确地估算软件工作量。

**优点**：

* **精细度高**：对每个功能点的复杂度有细致的评估标准，能够提供更精确的估算。
* **标准化**：相比传统功能点方法，MK II 提供了更为标准化的度量。

**缺点**：

* **复杂度较高**：对计数人员的要求较高，需要较长的培训时间和经验积累。
* **需要详细的需求**：需要对系统功能的需求进行详细分析，不适用于需求不明确的项目。

**适用场景**：

* 大型企业级应用，尤其是需求明确并且对复杂度要求高的系统。

### 6. ****NESMA（Netherlands Software Metrics Association）****

NESMA 是一种标准化的功能点估算方法，提供了对功能点计数的明确指导，强调标准化和一致性，旨在减少人为偏差。

**优点**：

* **标准化**：提供了统一的计数方法，确保不同计数者得到一致的结果。
* **广泛使用**：NESMA 在企业级应用中得到广泛应用，具有较高的可靠性。

**缺点**：

* **复杂性高**：需要对系统功能点的定义和复杂度进行全面理解。
* **灵活性较差**：过于依赖标准化，可能会导致某些特定项目的估算失真。

**适用场景**：

* 企业级应用，特别是需要高度标准化和一致性的项目。

### 7. ****FiSMA（Functional Size Measurement Method）****

FiSMA 是另一种功能尺寸度量方法，它通过度量软件系统中的功能模块的大小和复杂度来估算工作量。FiSMA 强调软件质量、复杂度等维度，灵活性较强。

**优点**：

* **灵活性高**：可以根据项目的不同需求进行灵活的调整。
* **质量控制**：除了功能点外，FiSMA 还考虑了系统的可维护性、扩展性等质量指标。

**缺点**：

* **学习曲线陡峭**：FiSMA 比较复杂，需要一定的领域知识和技术背景。
* **适用性较窄**：虽然适应性强，但不适合所有类型的项目。

**适用场景**：

* 质量控制较为严格的项目，如需要考虑软件的可维护性和复杂度较高的项目。

### 总结对比

| **方法** | **优点** | **缺点** | **适用场景** |
| --- | --- | --- | --- |
| **专家估计法** | 快速，灵活，适用初期阶段 | 主观性强，准确性差 | 项目初期，需求不明确，快速估算 |
| **类比估算** | 快速，简单，基于历史数据 | 依赖历史数据，差异较大时不准确 | 有历史数据可参考的项目，规模相似的项目 |
| **功能点估算法** | 客观、标准化，技术无关 | 需求分析要求高，忽略非功能需求 | 需求明确的大型系统、企业级应用 |
| **COSMIC-FFP** | 适应性强，精细化度量，实时系统和复杂系统适用 | 学习曲线陡峭，要求详细的需求分析 | 实时系统、嵌入式系统、数据库密集型应用 |
| **MK II** | 细化评估，适用于大型企业应用 | 复杂度高，需要经验积累 | 大型企业级应用，复杂系统 |
| **NESMA** | 标准化，减少人为偏差 | 复杂，灵活性差 | 企业级应用，需求明确的大型项目 |
| **FiSMA** | 灵活，注重质量控制，适用于复杂项目 | 学习曲线陡峭，适用性较窄 | 质量控制严格、复杂项目 |

总体而言，**专家估计法**和**类比估算**适用于项目初期阶段或快速估算，而**功能点估算法**、**COSMIC-FFP**、**MK II**、**NESMA**和**FiSMA**适用于需求明确、大型或复杂系统的工作量估算。不同方法的选择应根据项目的性质、需求明确程度和开发团队的经验来做出。