**比较课堂上所讲到的软件工作量估计方法**

**1. 算法模型**

1. **描述**：算法模型基于数学公式来估算工作量，依赖于输入的规模和环境参数。例如COCOMO模型使用系统规模（如代码行数KLOC）和经验参数来计算工作量。
2. **常见模型**：
   1. **COCOMO**：经典模型，分为基本COCOMO、名义成本修正COCOMO、COCOMO II。
   2. **功能点与生产率结合模型**：通过功能点计算工作量。
3. **优点**：
   1. 提供可量化的标准。
   2. 适用于长期项目和复杂项目。
4. **缺点**：
   1. 依赖高质量的历史数据。
   2. 参数调优需要专业知识。
5. **应用场景**：长期、大规模项目，如企业级软件开发和嵌入式系统。

**2. 专家判断**

1. **描述**：专家基于自身经验和领域知识，对项目的工作量进行定性或定量估计。
2. **过程**：
   1. 个别专家或小组提供估计。
   2. 使用Delphi法时，专家独立提供估算结果，再通过多轮匿名讨论达成共识。
3. **优点**：
   1. 快速适应新项目和未知领域。
   2. 在经验丰富的领域能提供较高准确性。
4. **缺点**：
   1. 主观性强，依赖专家的个人经验。
   2. 对无法获得经验的项目估算可能误差较大。
5. **应用场景**：
   1. 无历史数据支持的项目。
   2. 原型开发和新领域的探索性项目。

**3. 类比估算**

1. **描述**：通过寻找与当前项目相似的已完成项目，基于两者之间的特性差异调整估算工作量。
2. **过程**：
   1. 选择与当前项目特征（如输入、输出、逻辑复杂性）相似的历史项目。
   2. 使用类似性度量（如欧几里得距离）计算差异。
3. **优点**：
   1. 借助历史经验，易于实施。
   2. 对相似性高的项目估算准确。
4. **缺点**：
   1. 需要高质量的历史数据库。
   2. 项目差异较大时，准确性降低。
5. **应用场景**：
   1. 企业内部标准化流程开发的系统。
   2. 相似模块复用的项目。

**4. 功能点方法**

1. **描述**：通过计算软件的五个基本信息量（如外部输入、输出等）并加权，评估功能点（FP）。FP与生产率结合后可推导工作量。
2. **步骤**：
   1. 确定信息量：外部输入、输出、查询、内部逻辑文件、外部接口文件。
   2. 计算功能点：调整功能点的复杂性权重。
   3. 转换为代码行数或生产率。
3. **扩展**：
   1. **IFPUG方法**：面向信息系统。
   2. **COSMIC-FFP**：适用于实时系统。
4. **优点**：
   1. 与技术无关，具有较高独立性。
   2. 容易和标准化指标结合。
5. **缺点**：
   1. 不适合嵌入式系统或注重行为的系统。
   2. 对环境因子敏感，分析需经验支持。
6. **应用场景**：企业信息系统和业务系统。

**5. 帕金森法**

1. **描述**：基于“工作总是占满可用时间”（帕金森定律），以可用时间和资源反推工作量。
2. **优点**：易于实施，特别是在资源有限的情况下。
3. **缺点**：未必反映实际需求，可能导致低效。
4. **应用场景**：时间或资源严格限制的项目。

**6. 自顶向下估算**

1. **描述**：从整体项目出发，逐层细化到子任务，再估算子任务的工作量。
2. **优点**：
   1. 在项目初期快速给出总量估算。
   2. 易于适应变更。
3. **缺点**：
   1. 依赖高层次假设，误差较大。
   2. 可能忽略细节导致低估工作量。
4. **应用场景**：项目策划和预算阶段。

**7. 由低向上估算**

1. **描述**：从底层任务开始估算每个任务的工作量，再汇总为整体工作量。
2. **优点**：
   1. 精度高，考虑了所有细节。
   2. 更适用于复杂项目。
3. **缺点**：需要任务明确，耗时较长。
4. **应用场景**：项目计划的细化阶段。

**8. 对象点方法**

1. **描述**：统计项目中屏幕、报告和组件的数量和复杂性，计算出对象点。
2. **过程**：
   1. 对象分为简单、中等、复杂三类。
   2. 计算新对象点（NOP），结合开发生产率估算工作量。
3. **优点**：
   1. 简单易行。
   2. 特别适用于面向对象的软件。
4. **缺点**：
   1. 依赖对系统对象划分的质量。
   2. 不适合非面向对象的系统。
5. **应用场景**：面向对象开发环境和快速原型开发。

**9. 参数化模型（COCOMO及其变种）**

1. **描述**：基于系统规模、生产率和乘法算子，分阶段估算工作量。
2. **特点**：
   1. **COCOMO I**：基础模型，分有机模式、嵌入式模式和半分离模式。
   2. **COCOMO II**：增加了阶段性模型（如原型设计、后期设计）。
3. **优点**：
   1. 提供渐进式估算，适应项目变化。
   2. 融入了多种环境因子，模型成熟。
4. **缺点**：
   1. 参数复杂，难以调整。
   2. 对新手不友好。
5. **应用场景**：各种规模的软件项目。

通过综合分析，企业可根据项目阶段、复杂性和数据可用性选择合适的方法。例如：

* 项目初期：自顶向下估算或帕金森法。
* 中期规划：算法模型或功能点方法。
* 后期精度：由低向上估算或对象点方法。

### 方法比较

| **方法** | **精度** | **简便性** | **数据依赖** | **应用场景** |
| --- | --- | --- | --- | --- |
| 算法模型 | 高 | 中 | 高 | 大型复杂项目 |
| 专家判断 | 中 | 高 | 低 | 新领域，无历史数据 |
| 类比估算 | 中 | 中 | 高 | 相似项目丰富场景 |
| 功能点方法 | 高 | 中 | 中 | 信息系统、管理软件 |
| 帕金森法 | 低 | 高 | 低 | 资源约束型项目 |
| 自顶向下估算 | 中 | 高 | 中 | 项目初期 |
| 由低向上估算 | 高 | 低 | 中 | 项目后期 |
| 对象点方法 | 中 | 高 | 中 | 面向对象系统 |
| 参数化模型 | 高 | 中 | 高 | 各阶段项目 |