课上提到的几种软件工作量估计方法各有特点，适用于不同的项目需求和开发环境。

1. **由低向上估计**：这种方法通过将项目分解成具体的任务，并估算每个任务的工作量，最终汇总得到整体工作量。它适用于项目规划的后期，尤其是当项目已经有了明确的需求和任务划分时。如果项目是全新的且缺乏历史数据，这种方法能提供较为精确的估算，因为每个任务的工作量都是逐一考虑的。然而，缺点是，若在项目初期使用，需要对系统进行很多假设，可能导致估算不准确。
2. **自顶向下估计和参数模型**：这类方法通过对比历史数据，基于系统规模和生产率等参数来估算工作量。它适用于项目初期的快速估算，尤其是没有详细需求或项目结构不明确时。通过公式（例如：effort = 系统规模 \* 生产率）进行估算，通常使用KLOC（千行代码）来衡量系统规模，生产率则以每KLOC所需的天数来表示。优点是快速、简便，但缺点是依赖于有效的历史数据，并且参数选择不当可能导致估算误差。
3. **专家判断**：专家判断通过依赖具备领域知识的人员来进行估算。Delphi法是一种典型的专家评估方法，专家通过讨论达成一致。这种方法适用于缺乏历史数据或项目需求不明确的情况，能够为估算提供专业的见解。然而，它的准确性依赖于专家的经验和技术水平，可能受到主观因素影响，且对于非常复杂的系统可能难以准确评估。
4. **类比估计**：类比估计通过寻找与目标项目相似的历史项目进行对比，从而推测工作量。该方法依赖于已完成项目的相似性，因此它要求找到足够相似的项目进行比较，评估者使用欧几里得距离等数学公式来量化相似度。这种方法的优点是能够在项目初期快速得到估算，缺点是相似项目的选择至关重要，不同项目之间的细微差别可能导致估算误差。
5. **功能点方法**：功能点方法根据系统的功能需求来计算工作量，考虑外部输入、输出、查询等因素，并通过环境复杂性因子进行修正。该方法更加注重系统功能的复杂度，适用于需求明确的软件项目，能够提供较为系统的估算。但功能点的计算需要详细的需求分析，且对复杂的环境和系统需求可能不够灵活。
6. **对象点**：对象点方法与功能点方法类似，但它更加注重系统中需要处理的“对象”，如屏幕、报告等。每个对象被分为简单、中等和困难三级，依据开发者的经验和工具使用情况来估算工作量。该方法相较于功能点更加简单和直观，适用于面向对象的方法，但也需要具备一定的经验和项目背景支持。

总的来说，**低向上估计**适合于后期的详细规划，能够提供精确的估算，但对早期项目规划不适用；**自顶向下和参数模型**适合于项目初期，可以迅速给出估算，但依赖于历史数据；**专家判断**方法适用于缺乏历史数据或需求不明确时，但可能受到主观因素影响；**类比估计**适合于已有类似项目经验时，但需要确保相似项目的准确选择；**功能点方法**适用于需求明确且功能复杂的项目，而**对象点方法**则更加简洁、易用，适用于面向对象的开发环境。选择合适的工作量估算方法需结合项目的实际情况和需求进行权衡