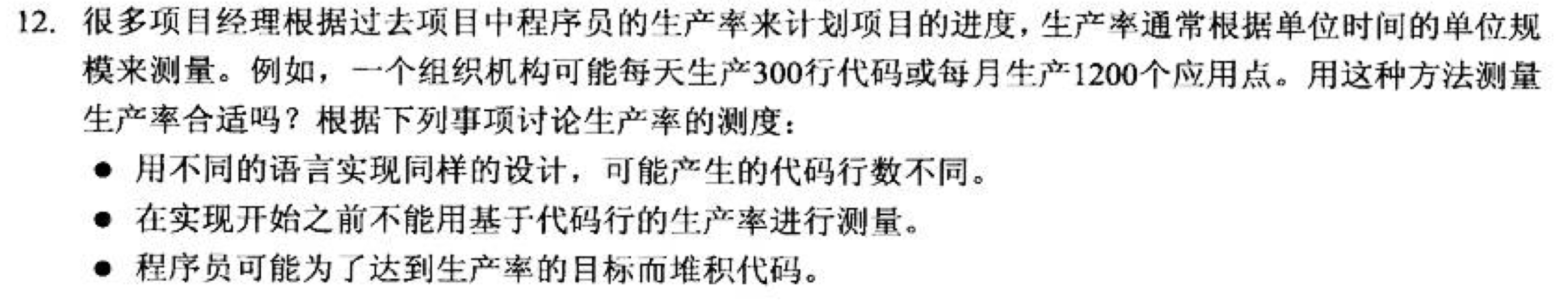
Ch3 习题12：



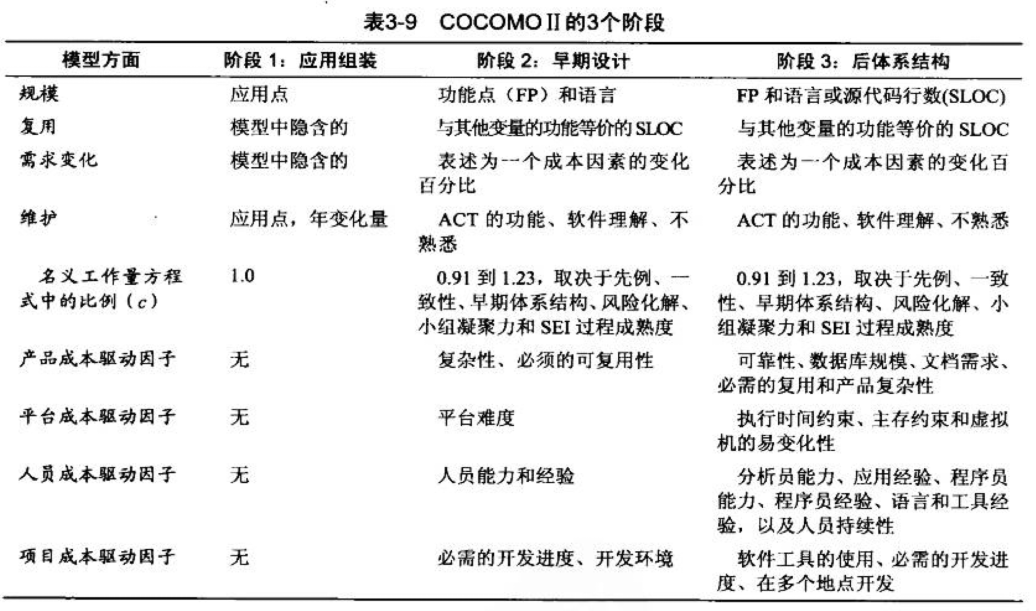
答：根据单位时间的单位规模来测量生产率这种方法不合适。

具体原因如下：我们的项目使用不同的语言实现，会导致项目的代码量差距很大。比如同样的一个项目使用rust、haskell、ocaml编译器的代码量差不过，但是使用c++的编译器的代码量会多30%，而使用python的代码量则会少一半。同样的对于这种方式，还有一个问题是在项目的开发前期，还没有进行代码量的工作，这个方法将会不适用，这对于前期的项目开发会产生负面影响，导致效率低下。在程序员的方面，这种方法没有重视人的能动性，只是用很简单的代码工作量标示生产率，会导致程序员进行水代码的性为，而不会促进程序员去开发更合理的代码结构和算法来使代码更加精简，提高项目的成产率。因此，制定一个能够克服所有这些缺点的生产率指标是非常有帮助的，此外，我们还需要了解些无论如何测量都与之相关的周边问题，比如生产力目标，管理风格和程序员专业知识对生产力的影响等。

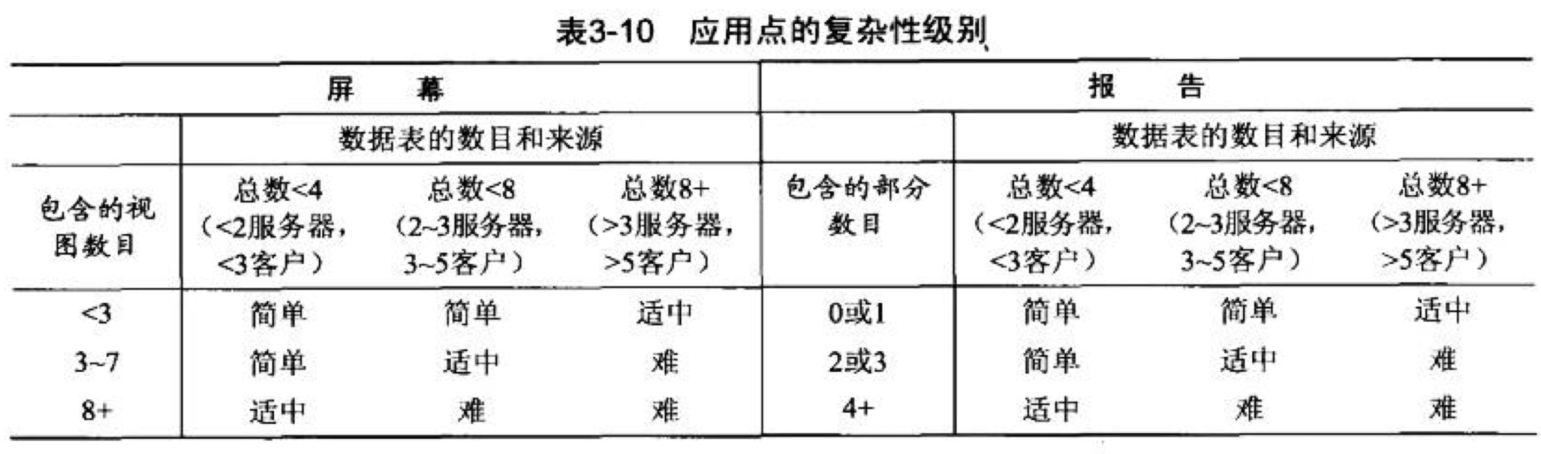
工作量估算技术：专家判断；算法方法；机器学习法方法。

工作量估算（算法方法）：

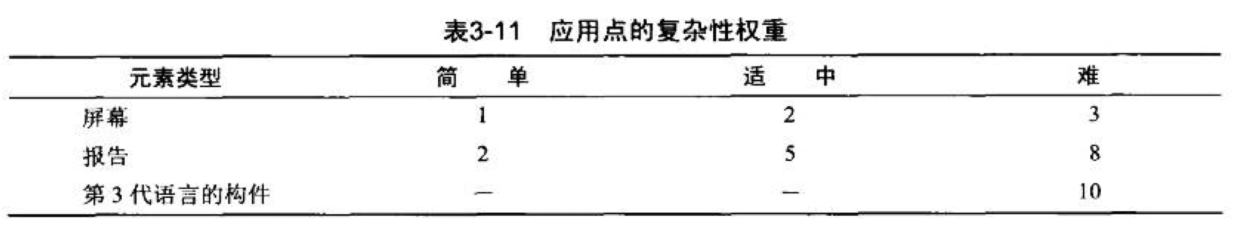
COCOMOII模型反映了软件开发充分发展后的各个方面。COCOMOII模型的估算过程体现了任何开发项目都包含的3个主要阶段。



在阶段1，项目通常构建原型以解决包含用户界面、软件和系统交互、性能和技术成熟性等方面在内的高风险问题。由于此时人们对于正在创建的最终产品的可能规模知之甚少，因此COCOMOII用应用点来估计规模。这种技术根据高层的工作量生成器（如屏幕数量和报告数量、第三代语言构建数）来获取项目的规模。要计算应用点，首先要计算应用中将要包含的**屏幕、报告和第三代语言的构建数目**，之后将每个应用元素分类为简单、适中或难3各级别。



简单、适中或难的应用点所对应的数值在下表可找到的复杂性权重。这些权重反应那些复杂性级别的高爆或屏幕所需的相对工作量。



本项目所需的影片信息、相关演员信息以及影院信息均通过合理爬虫获得；

我们通过计算应用点确定项目可能的规模，使用汇总后的影片、相关演员、影院数据库表格以及注册用户表作为屏幕数据表的数目和来源；使用爬虫所得原始影片、相关演员、影院数据库表格作为报告数据表的数目和来源；已知本项目没有复用已有代码。

综上，我们需要预测本项目所需使用的屏幕与报表数目。假定初始估算需要从三个源获取原始影片信息（获取原始影片信息的同时，可以构建演员与相关影片的联系），需要注册用户表1；假定将所获取信息汇总后构建为一个影片信息表，一个演员信息表，一个演员信息关联表，一个用户表。

* 原始影片信息：由爬虫获得，假定需要三个源（即从三个源头获取信息）
* 原始演员信息：爬虫原始影片信息的同时获得
* 影片与演员关系表：可以单独构建一张关联表
* 影院信息：由爬虫获得，包含影院环境评价以及上映的影片
* 用户表：注册用户相关信息（如历史购票信息、历史评价信息）

对于每个信息表，使用COCOMOII提供的指导以及所需数据表数目的估算来生成该网页或账单的描述，预约网页可能需要使用3种数据表：影片与演员关系表、影院与影片上映关系表、影片评价表、用户表。因此数据表数目小于等于4，这样我们必须决定是否需要8个以上的试图，通过规划计算，需要的试图足够8个，因此根据书中3-11应用点复杂性权重表，把影片与演员关系表，评定为“适中”。类似的把影院与影片上映表评定为“简单”，影片评价表评定为“适中”。用户表评定为“适中”。之后给各屏幕分配复杂性评分，简单的屏幕为1，适中的屏幕为2，适中的报表为5，汇总表如下：

表：项目网页和报告的评估

|  |  |  |  |
| --- | --- | --- | --- |
| 应用点 | 屏幕或报表 | 复杂性 | 权值 |
| 影片与演员关系表 | 屏幕 | 适中 | 2 |
| 影院与影片上映表 | 屏幕 | 简单 | 1 |
| 影片评价表 | 报表 | 适中 | 5 |
| 用户表 | 报表 | 适中 | 5 |

加和最右边一列的所有权值，生成新的应用点的计数（NOPS），其值为13，开发人员为山大在校学生，具有最低的经验和最低的CASE成熟度。通过书中表3-12显示了这种情况的生产率评分为4.因此，用COCOMO模型可以等到笔墨丹青项目系统的估算工作量是NOP除以生产率评分，即3.25人月。