## 1.工作量估算

### 习题12.1：

**Q:很多项目经理根据过去项目中程序员的生产率来计划项目的进度，生产率通常根据单位时间的单位规模来测量。例如，一个组织机构可能每天生产300行代码或每月生产1200个应用点。用这种方法测量生产率合适吗？**

A: 这种方法不合理，因为：

不同编程语言实现同一功能所需的代码量存在较大差异，例如，实现网络请求的发送与接收功能，JAVA和Python的代码量差别较大，可能相差数十行。

当代码复用率高时，代码量会相对减少，但是代码效率和可维护性会显著提高。

代码量与项目使用的设计模式密切相关，MVVM模型相对于MVC模型的代码量可能会增加几倍，特别是在小型项目中更为明显。

在开始实施整个项目之前，无法准确估计代码行数，因此使用基于代码行数的生产率计算失去了意义。

如果任务分配不合理，程序员可能会为完成目标而堆积代码，从而造成人力浪费。”

**Q1：根据下列事项要论生产率的测度：用不同的语言实现同样的设计，可能产生的代码行数不同：**

A：用单位时间的单位规模来测量生产率在某些情况下是合适的，但也有一些限制和局限性。

首先，用代码行数或应用点数等类似的指标来测量生产率可能存在误导性。因为不同的编程语言和技术可以实现相同的功能，但生成的代码行数可能会有很大的差异。因此，仅根据代码行数或应用点数等指标来衡量生产率可能会产生误导，因为这些指标并不能准确地反映实际的工作量或质量。

其次，单位时间的单位规模也不能反映出程序员的工作效率和质量。程序员的工作效率和质量可能受到多种因素的影响，包括代码复杂度、开发工具的质量、项目的复杂度、团队合作等因素。因此，仅仅通过代码行数或应用点数等指标来测量生产率是不够准确的，需要综合考虑多种因素来评估程序员的工作效率和质量。

综上所述，用单位时间的单位规模来测量生产率可能存在一些限制和局限性。为了更准确地评估程序员的工作效率和质量，需要考虑多种因素，包括代码复杂度、开发工具的质量、项目的复杂度、团队合作等因素。

**Q2: 根据下列事项讨论生产率的测度：在实现开始之前不能用基于代码行的生产率进行测量**

A: 在实现开始之前使用基于代码行的生产率进行测量可能会产生误导性，因为在项目开始之前并没有具体的代码实现。此时，生产率的测量应该基于其他因素，例如项目的预算、资源、进度和范围等。

**Q3: 根据下列事项讨论生产率的测度：程序员可能为了达到生产率的目标而堆积代码**

A3: 程序员为了达到生产率的目标而堆积代码可能会产生一些负面影响。如果程序员只关注生产率目标，而忽略了代码的可维护性和可读性等因素，就可能会产生堆积代码的情况。堆积代码可能导致代码复杂度增加，代码可读性和可维护性降低，从而增加代码出错的风险。

此外，如果程序员仅关注生产率目标而忽略了质量控制和测试等过程，就可能会出现代码质量低下的情况，这会对项目的进度和质量产生负面影响。

因此，生产率的测量应该考虑多个因素，例如代码质量、可维护性、可读性、代码测试等因素。同时，还应该鼓励程序员采用有效的编程技术和工具来提高他们的工作效率和代码质量，例如自动化测试、代码审查、重构等技术。