Ch3 习题12：

用这种方法来测量生产率并不合适。

1. 采用不同的语言来进行设计，可能产生的代码行数不同。进行程序设计的时候最终代码应该体现出代码的可读性，这样就意味着代码不能有太多行，否则阅读起来非常费劲，同时也不能缩到一起，否则也无法阅读。况且代码风格比如花括号在哪一行、缩进用的是tab还是空格等也影响代码行数。最后不同的语言实现相同的功能所用的代码行数不一样，即使是同样的代码实现同一功能也有不同的实现方式，如果仅仅看代码行数那么会导致程序失去可读性。
2. 在实验开始之前不能用基于代码行的生产率来进行测量。程序开发的一个特性就是解耦性，同时大量的第三方库也让程序设计变得更简单，这样程序在设计之初就需要确定使用什么第三方库，第三方库实现内容比较复杂，而且不少设计基础架构与底层架构，当然有些时候还需要对第三方库进行一些修改保证能够实现某些特定的功能，那么估算代码量的时候无法判定这一部分第三方库的代码量如何。程序员真正实现的代码其实不多，但是他们需要了解基本架构来保证程序的实现，那么阅读代码是不是要定量算入生产的量中呢？
3. 程序员可能为了达到生产率的目标而堆积代码。如果以代码行数来判断生产量的话，那么程序员会进行大量的无效代码堆积，比如增加更多无用的注释来增加行数，甚至每一行都需要大量注释来做重复解释；将代码分开多行书写，极端情况甚至一个字母独占一行；加入大量不会引用的无效函数，这些函数并不会引用也不会影响最终结果但是却增加了很多代码量。以上种种方法使得最后的程序代码量无端增大且最终程序异常臃肿但是最后没有实现多少功能。

工作量估算：

在阶段1，我们主要分析规模、复用、需求变化、维护和名义工作量方程式中的比例（c），阶段1并不分析产品成本驱动因子、平台成本驱动因子、人员成本驱动因子、项目成本驱动因子。其中c可以取1.0，也就意味着它并不适合这种估算，复用和需求变化在模型中隐含，因此我们主要分析规模和维护，二者都需要计算应用点。规模测量根据kauffman和kumar提出的对象方法和banker、kauffman、kumar报告的生产率数据的扩充。我们发现程序中逻辑、UI、网络三个部分的部件数目中UI最多、网络和逻辑其次。此外开发人员能力明显不足，除去一小部分有开发经验的人士之外绝大部分人的开发经验几乎为零，大多数会使用基本的工具和适中的集成。