Ch3 -12

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?根据下列事项讨论生产率的测度:

• 用不同的语言实现同样的设计，可能产生的代码行数不同。

• 在实现开始之前不能用基于代码行的生产率进行测量。

• 程序员可能为了达到生产率的目标而堆积代码。

用这种方法测量生产率是不合适的。

不同语言的影响：

不同的编程语言具有不同的语法、语义和抽象级别，因此同样的功能可能需要不同数量的代码行来实现。有的语言会需要更多的代码来实现相同的功能，而另一些语言可能具有更高的抽象级别，相对简洁，因此能够以更少的代码行实现相同的功能。比如使用高级编程语言如Python或Ruby比使用低级编程语言如C或汇编语言，可能会产生更少的代码行数，因为高级语言通常具有更高的抽象，能够更快速地实现相同的功能。另外，使用某些框架或库也可以减少编写的代码量。因此，单纯以代码行数来测量生产率可能会导致误导，特别是在跨不同编程语言或技术栈的项目中。

无法预测实现前的生产率：

在项目开始之前很难预估最后会产生多少行代码，要知道代码行数只能等项目实现。首先代码行数会受到设计的算法等影响，后期经过修改和调试，会发生变动，在项目完成之前也很难预判，比如说指定某一功能后，在实现之前也很难知道会出现多少行代码。

堆积代码的风险：

堆积代码是指程序员为了达到生产率目标而不顾代码质量、可维护性和整体设计的情况下写出的低质量代码。这种做法可能会带来一系列问题。包括：

代码质量下降：堆积代码往往意味着牺牲了代码质量，可能导致出现bug、性能问题或安全漏洞等质量问题；维护困难：低质量的代码通常难以理解和维护，这会增加未来修改和更新代码的成本和风险；技术债务增加：堆积代码会增加技术债务，即未来需要花费额外时间和资源来解决由于低质量代码而产生的问题；团队工作积极性下降：过度强调生产率目标而忽视代码质量可能会导致团队成员的工作积极性下降，因为他们无法保证编写高质量的代码。