**孙明宇：**

用这种方法来测量生产率是不合适的。

①首先不同的编程语言生产的代码，长度是明显不同的。例如可能在C语言中需要很长一段实现的一个函数，在Python语言中，调用一个函数库就可以实现这个功能。

②在项目开展之前，其实不能明确地对某部分的代码长度有一个较为准确的估算，对于一些美观装饰类的代码，其长度可能很长，但作用却不是那么的大。但是对于某些关键算法，可能算法十分简短，但是功能却极其强大。同时，对于同一个问题，可能有多种不同的算法思路进行实现，而这样也会使代码长度会有一些差异。

③同时，如果以这种测量方式去进行代码的书写，可能导致某些程序明明可以很简便、明了的可以用几行代码实现，却为了生产率，而用一大长段无用代码进行替代，造成资源浪费和无用功的出现。

④而且，不同程序员的代码书写风格和格式也是略有差异的。有些程序员希望函数与函数之间有许多空行，方便看的清除；有些程序员希望函数与函数之间可以紧凑一些，方便整体美观。同时负责不同部分的程序员代码工作量是不同的，

**宋璎航：**

用单位时间的单位规模来衡量生产率显然欠缺科学性。

首先，采用不同的语言实现同样的代码，代码行数是不相同的。比如Python进行网络编程只需要调用如requests等库即可，JAVA网络编程同样简洁，但如果用C或C++进行网络编程，则需要从构造套接字开始进行复杂的系列操作，产生的代码长度可能是Python的成百上千倍，但是这两种代码所产生的功用是相同的。

其次，代码的行数很大程度上取决于代码的复用率，一个优秀程序的代码复用率可能极高，这就导致了这类程序代码较短，而完成相同功能的复用率较低的代码长度可能很长。

再次，代码的风格也导致了一定程度的代码长度差异。比如C语言的如下语句

A[++a1][++a2] = B[++b1][++b2];

如果是另一种代码风格：

a1++;

a2++;

b1++;

b2++;

A[a1][a2]=B[b1][b2];

这样代码长度就会长很多，但两者的功能完全相同。

最后，如果采用以代码数量衡量生产率的方式，程序员可能会为了达到生产率而堆积代码，给后期的维护运营带来不便，同时部分冗余代码甚至会拖慢机器性能，降低产品质量。

**皮亚杰：**

* 一个项目中可能会用到多种编程语言,而不同的编程语言实现同样的功能所需要的代码行数会有较大的差别
* 程序员可能会因为希望增加代码行数而选择换行较多的代码格式,甚至在可以使用循环的情况下使用多次循环或直接不使用循环而罗列大量的条件分支语句
* 在项目开始之前由于不会产生任何代码,因此这个阶段程序员的工作量无法通过测量代码行数进行衡量
* 不同的任务实现起来的难度会有很大的不同,完成难度大的任务的程序员在与完成难度低的程序员进行代码行数比较时会有很大的劣势，而前者完成的工作可能实际上更多、更重要

**小组整合意见：**

（1）不同编程语言，产生同样功能的代码长度不同，不同程序员的代码风格也会影响代码长度，所以这种生产率的测度是不合理的。

（2）不是所有的任务分配所带来的工作量都会与代码长度成正相关，很多不产生代码的工作，其工作量也很大，无法测量这部分的工作量。

（3）如果一味地去追求生产率目标，可能会造成代码阅读性差，或者开销大，进而导致效能低下，项目延时或质量下降。