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。

综上。