1. Quality and productivity were chosen to evaluate the level of a programmer because they are easy to measure. According to a section of "Measuring Software Productivity" by Steve McConnell [1], it is difficult to precisely define the input and output of a programmer, which means that it is difficult to choose the right metric. The study created by Sackman and his teammates in 1968 and similar studies [2] have demonstrated that productivity varies greatly, whether it is a single person or a team. Therefore, I believe that it is inaccurate to evaluate programmers simply by evaluating quality and productivity, because they are susceptible to process, practice or environmental influences.
2. I would like to use a bug rate to make some simple conclusions, which is quality divide by productivity. Bug rate shows the probability of a programmer's bug in unit 1 of productivity, the higher the bug rate is, the more it can reflect part of the programmer's strength. A is 0.2, B is 0.12, C is 0.333, and D is 0.35. So D has the highest bug rate and B has the lowest bug rate.

[1] "Measuring Software Productivity," Steve McConnell <https://www.youtube.com/watch?v=Umj-I1CS4YU>

[2] What’s Wrong With Measuring Developer Performance (+ 10 Best Metrics) https://www.actitime.com/productivity/developer-performance-metrics