

Computer Programmer Productivity Differences

Prabhakar Deshpande

Date of Submission: 22-08-2022

Date of Acceptance: 05-09-2022

Abstract: Computer programmer productivity differences are substantial. However they can be exaggerated. But they should be ignored.

Being a Project Manager in Information Technology taught me that there are huge differences between one programmer and another even at same level. In terms of productivity one smart programmer could actually be almost 10 times as good as another not so savvy programmer. While one should avoid exaggerating these differences, it is important to not ignore these differences in productivity.

Of course humans are different. But one does not expect so much difference in human beings as much is found between one computer programmer and another computer programmer. After all most computer programmers are graduates - either in business or engineering or technology. Hence one would expect productivity difference between programmers to be at the most in order of 1 to 2. However, it turns out that the productivity difference between best programmers and worst programmers at the same level in a team can be of the order of 10:1.

A study on the difference in productivity levels of programmers began in the 1960s by Sackman, Erickson and Grant 1968. They studied programmer productivity differences with an average of 7 years of experience and found that the ratio of coding time between the best programmer and the worst programmer was of the order of 20:1, the ratio of debugging time was of the order of 25:1, program size was 5: 1 and program execution speed was 10:1.

A detailed study of Sackman, Erickson and Grant found some flaws in their methodology. However even after adjusting for flaws in methodology, data show that there still is a 10 fold difference between best programmers and worst programmers.

However, this difference some might argue is not limited to software engineering alone. In most professions - police work, writing, and invention - it could be said that 20% of workers produce 50% of output. Hence the difference between the top 20% and the bottom 20% could be almost 10 times.

Humans are vastly different. Some people find it difficult to run for 400 meters, whereas other humans can run 100 times that amount in a marathon of 42 km.

In one study by Boehm, Gray and Seewalt(1984) of identical projects, efforts expended varied by a factor of 3.4:1 and program size varied by a factor of 3:1. An earlier study in 1974 by Weinberg and Schulman found a 5:1 difference in program size and 2.6: 1 in program time to complete the same project.

Barry Boehm and others 2000 reviewed 20 years of study and found that developing a program with programmers ranked in the 15th percentile requires 3.5 times as much time as developing a program with somebody who was ranked in the 90th percentile.

It is interesting to compare productivity differences between program developers of Lotus 123 and Microsoft Excel. Excel took 50 staff years to produce 650,000 lines of code, whereas Lotus 123 took 260 staff years to produce 400,000 lines of code. Excel team produced 13000 lines of code per staff year. Whereas the Lotus team produced 1500 lines of code per staff year. Thus there is a productivity difference of an order of 8 between two project teams and not just two programmers.

However, Bill Nicholas challenges the extent of productivity difference in his article in IEEE Software - 'The End to the Myth of Individual Productivity Difference'

He found that there are very few programmers at the end of productivity difference. The same programmers were seldom best or worst. While the average programmer's performance varied between two programmers only half could be attributed to inherent programmer skills.

Thus while it is true that at extremes the programmer productivity difference may be as much as 10:1, most often productivity difference is as less as 2

Now what implication does the above study have for management.

If one were to strictly go by the earlier claims that programmer productivity difference is as high as 10:1 it would make sense to hire more productive programmers. After all, if you were to pay a bad programmer

50,000 dollars and a good programmer 100,000 dollars, given the fact that a good programmer is going to be 10 times more productive than a bad programmer then a good programmer is 5 times cheaper than a bad programmer.

However, this ignores the fact that these extremes in programmer productivity are rare. Most programmers fall into the average category. And even in the average category, the difference in programmer productivity attributable to skill differences is half the perceived difference. Again great programmers are few and hard to find. With better training, better practices, and better management, programmer productivity differences can be minimized.

Even so, the fact that programmer productivity skills when adjusted for all differences even in an average category after removing extremes still varies by a factor of 2. And that may mean a significant advantage. Hence it does follow that organizations must make effort to hire programmers with higher productivity even if it is at a slightly higher cost. However extreme differences in programmer's productivity are statistical aberrations that emphasize the marginal at the expense of the average.

- [1]. Nicholas, W., 2019. The End to Myth of Individual Programmer Productivity Differences. *IEEE Software*, 36(5), pp.71-75. <http://dx.doi.org/10.1109/MS.2019.2908576>
- [2]. Boehm, B., 1985. *Measuring Programmer Productivity and Software Quality*. John Wiley and Sons.

Prabhakar Deshpande. "Computer Programmer Productivity Differences." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 24(5), 2022, pp. 01-02.