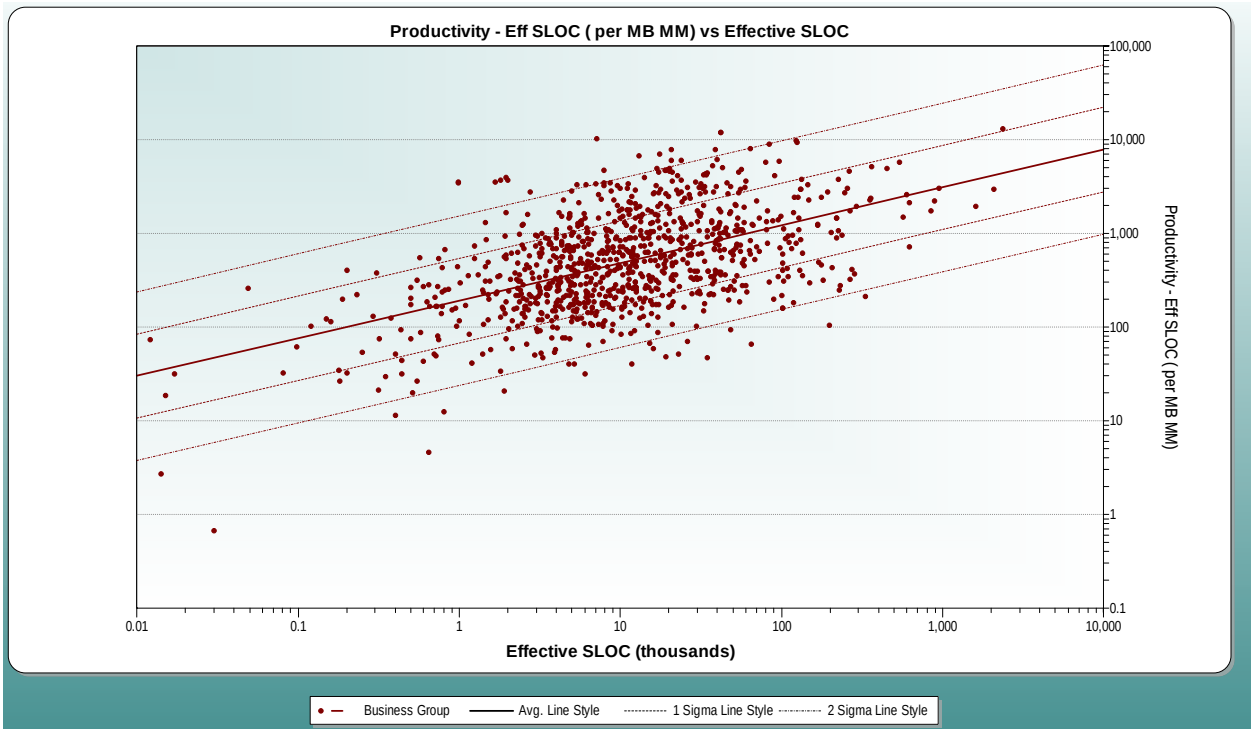
Productivity and Size: Part 2

If productivity increases with project size (and large projects use bigger teams) why are large-team projects generally *less productive* than small team projects?

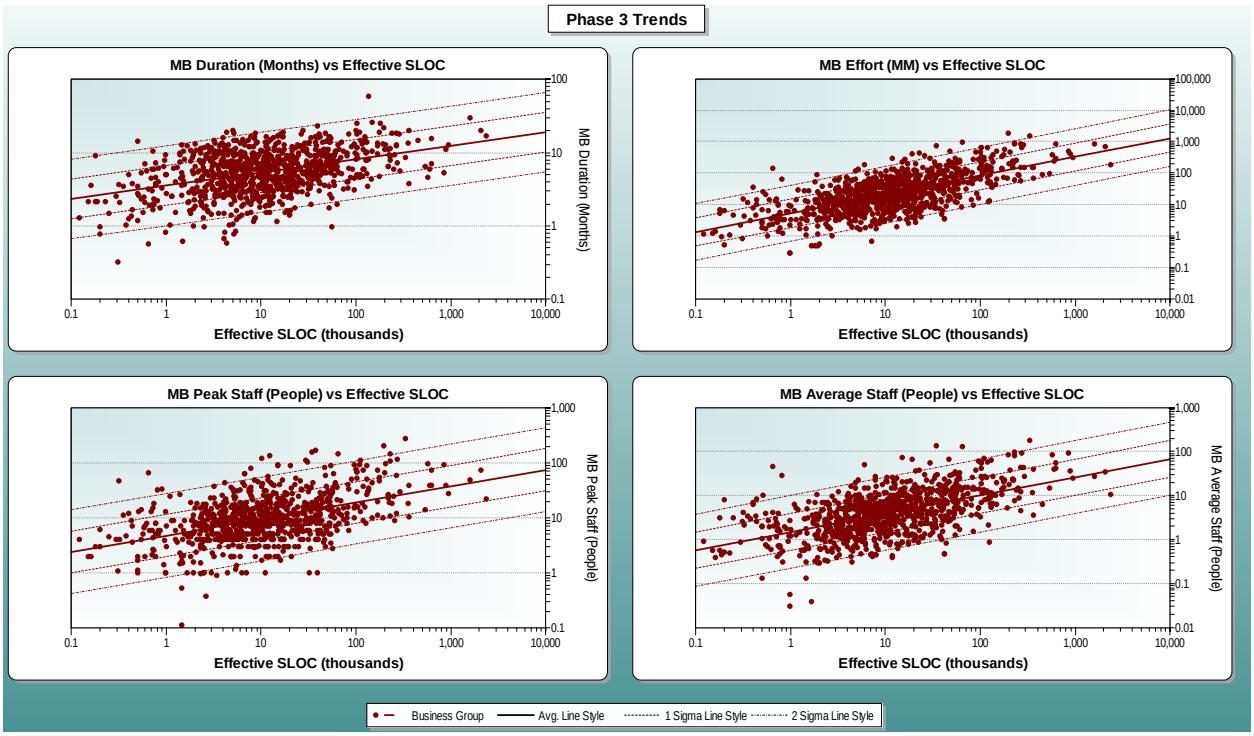
By Paul Below

In Part I of this series, we demonstrated that average productivity (defined as effective size/effort) increases with project size. This relationship holds true across the size spectrum whether we're talking about projects in the very small range or projects that deliver a million lines of code. Above this cutoff, the sample size is too small to be definitive.

Though these findings contradict the theories of some authors, the data is quite clear. Let's look at a graph of IT applications completed after January 1, 2000. Included effort covers detailed design through implementation. Average, one, and two sigma trend lines are derived from the displayed data points.

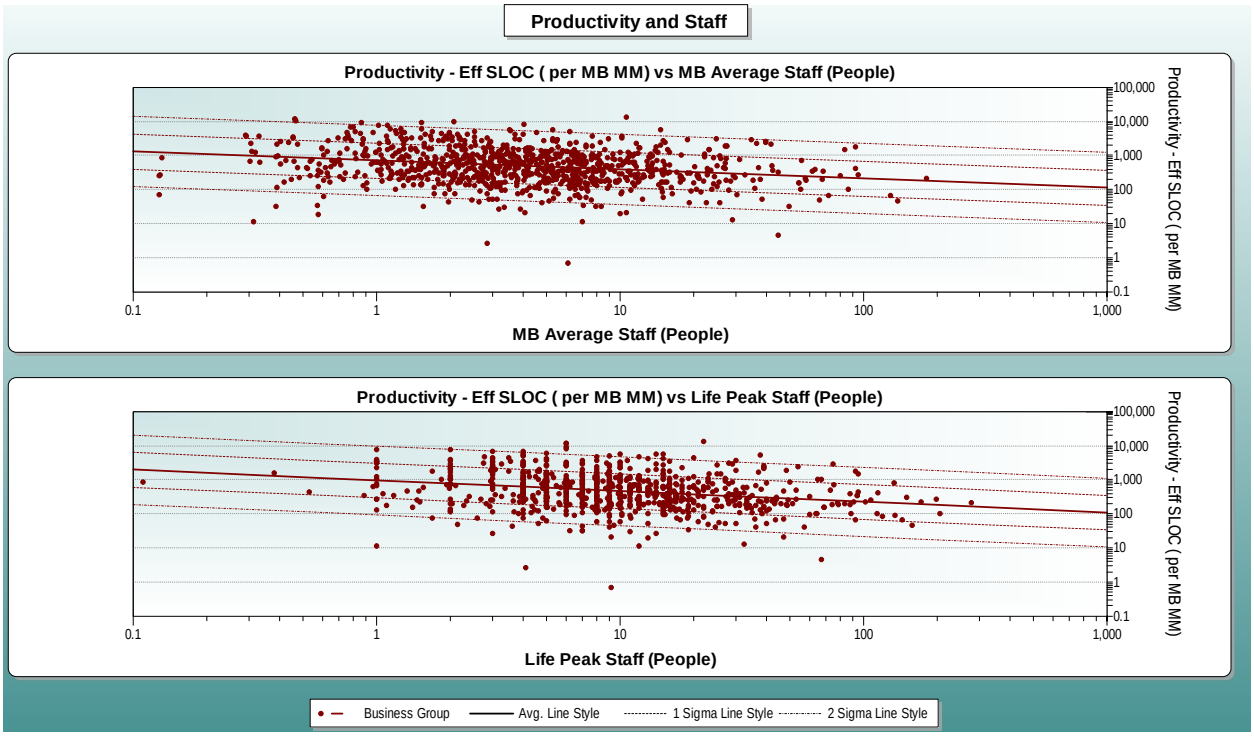


As the following four charts show, duration, effort and staff are all positively correlated with size. What this means is that on average, large projects use more effort, take longer, and use larger teams than small projects. At first glance this is not surprising: large jobs ought to require more people and time than small ones.

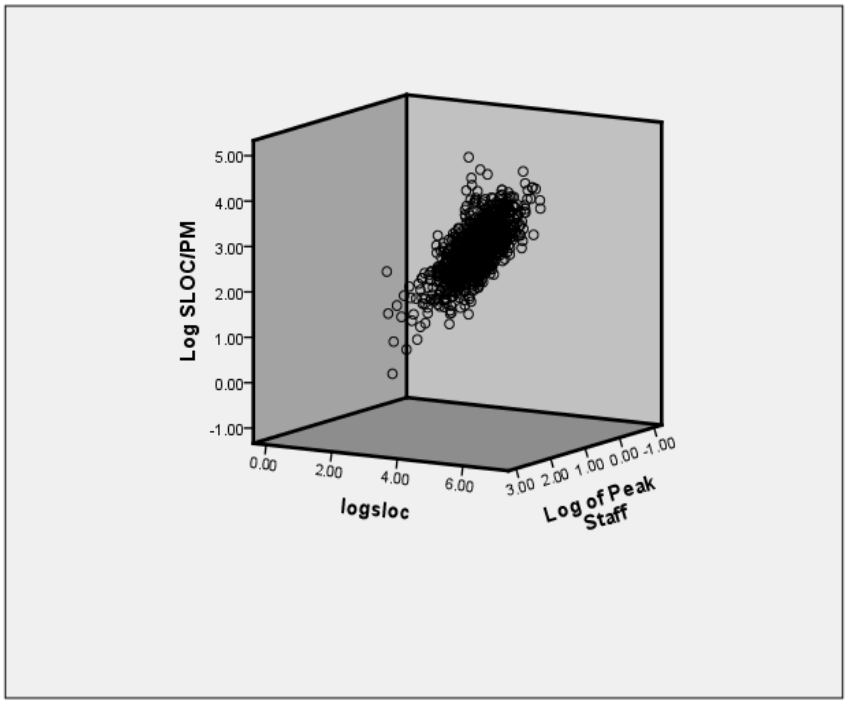


But how do we reconcile these results with previous studies that show large teams delivering lower average productivity? In part one of this series we learned that larger projects are *more* productive but use larger teams. What is going on?

To find out, we decided to look at the data a different way. We began by confirming that simple productivity declines as team size increases. As the next graph shows, the negative correlation holds true whether average staff or peak staff is used to measure of team size. Note the vertical clustering of the data points on the peak staff chart. This occurs because peak staff is frequently reported as a whole number. Average staff, which is calculated from effort and duration, is less likely to be recorded in integer form.



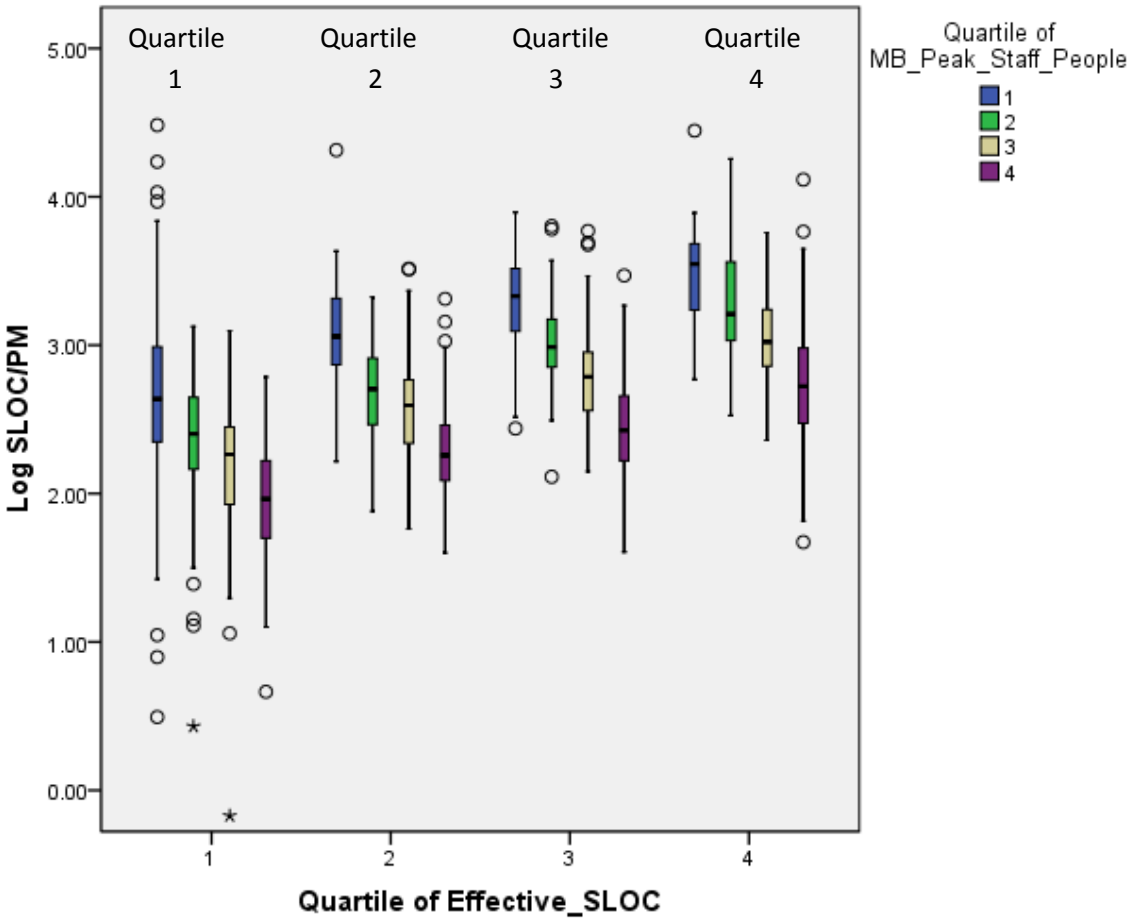
Though they do confirm that conventional productivity declines as project size increases, two variable charts don't tell us why this is true. To answer that question, we need a way to visually examine *three* variables at once: size, productivity, and team size. This can be done with a 3D plot but since we can't visually rotate the 3D cube in space, it is still hard to tell what is going on.



Clustered box plots provide more insight into the relationship between our three variables. Each box in the following box plot represents the interquartile range of the data:

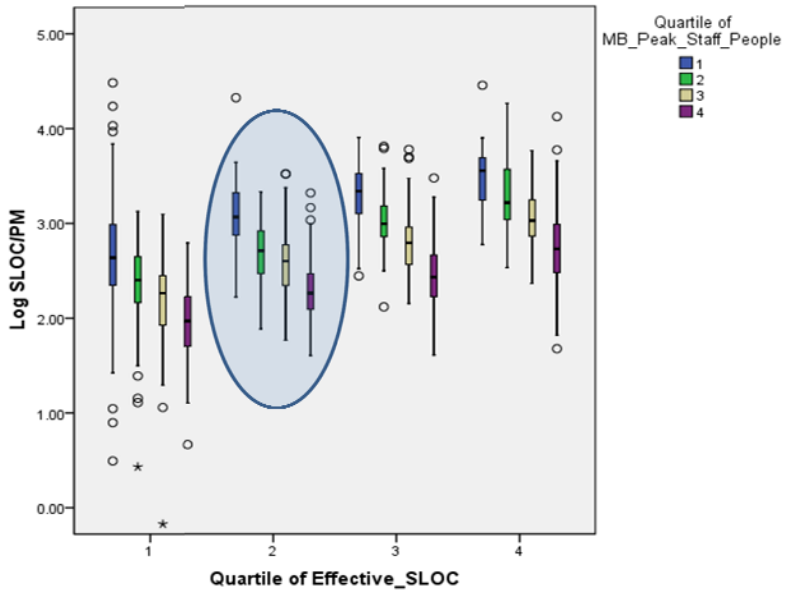
- The bottom of the box represents the first quartile (25th percentile).
- The horizontal line inside the box is the median (50th percentile).
- The top of the box represents the third quartile (75th percentile).
- "Whiskers" extending from either end of the box show the range of the values (min – max).
- Individual outliers show up as circles and extreme values as asterisks.

We divided our IT sample into size quartiles. Quartile 1 contains the smallest 25% of projects and quartile 4 contains the largest 25%. Productivity (SLOC/PM) is on the vertical axis.

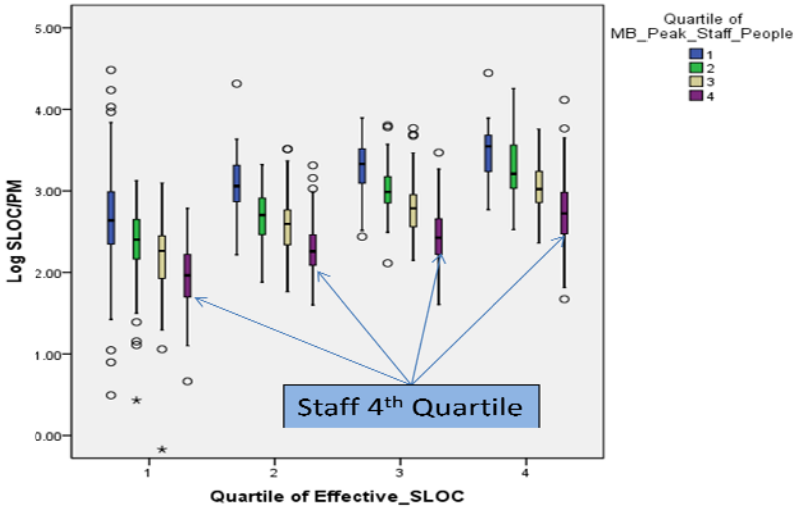


Each size quartile or "bin" was then divided into 4 staffing quartiles with the first quartile containing the smallest teams (the bottom 25%) and the 4th quartile containing team sizes at the 75th percentile or above.

To make the size/team size/productivity relationship easier to visualize, an oval has been drawn around the second size quartile in the next graph. Within each size "bin" or quartile the behavior is the same: productivity (the 4 colored boxes in each size quartile) declines as team size increases.



Next, let's compare the "largest teams" (top staff quartile) within each size bin. For large team projects in each size bin, productivity increases as project size increases. This holds true all along the "team size" spectrum: compare any 4 boxes of the same color (one from each size bin). For a given team size/project size bin, productivity is higher for the larger projects. In the following graph, the largest teams in each size bin are identified with arrows.



Summary

In the first post in this series, we determined that simple productivity increases with project size all along the size spectrum. In this installment we have shown that *in general, smaller teams are more productive.* But it is important to realize that **optimal team size is not independent of application size.** Though small teams are generally more productive, a team that is "too big" for a tiny project delivering only 1000 lines of code may be the perfect size for a project delivering 100,000 lines of code. The key is to *match the team size to the work* rather than overloading the project in hopes of achieving a dramatic reduction in schedule.

That might seem obvious, but what seems like common sense is a lot harder to argue with when the data makes your case for you!