Maximizing Value for Cost in Software Development

Paul Below

paul_below@qsm.com

Agenda

Introduction

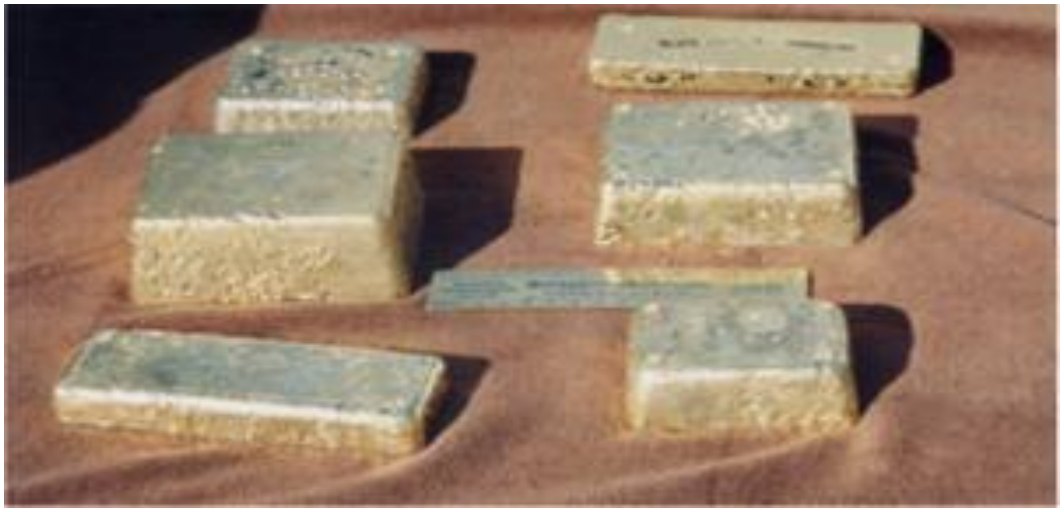
Productivity Paradox

Reliability

Summary

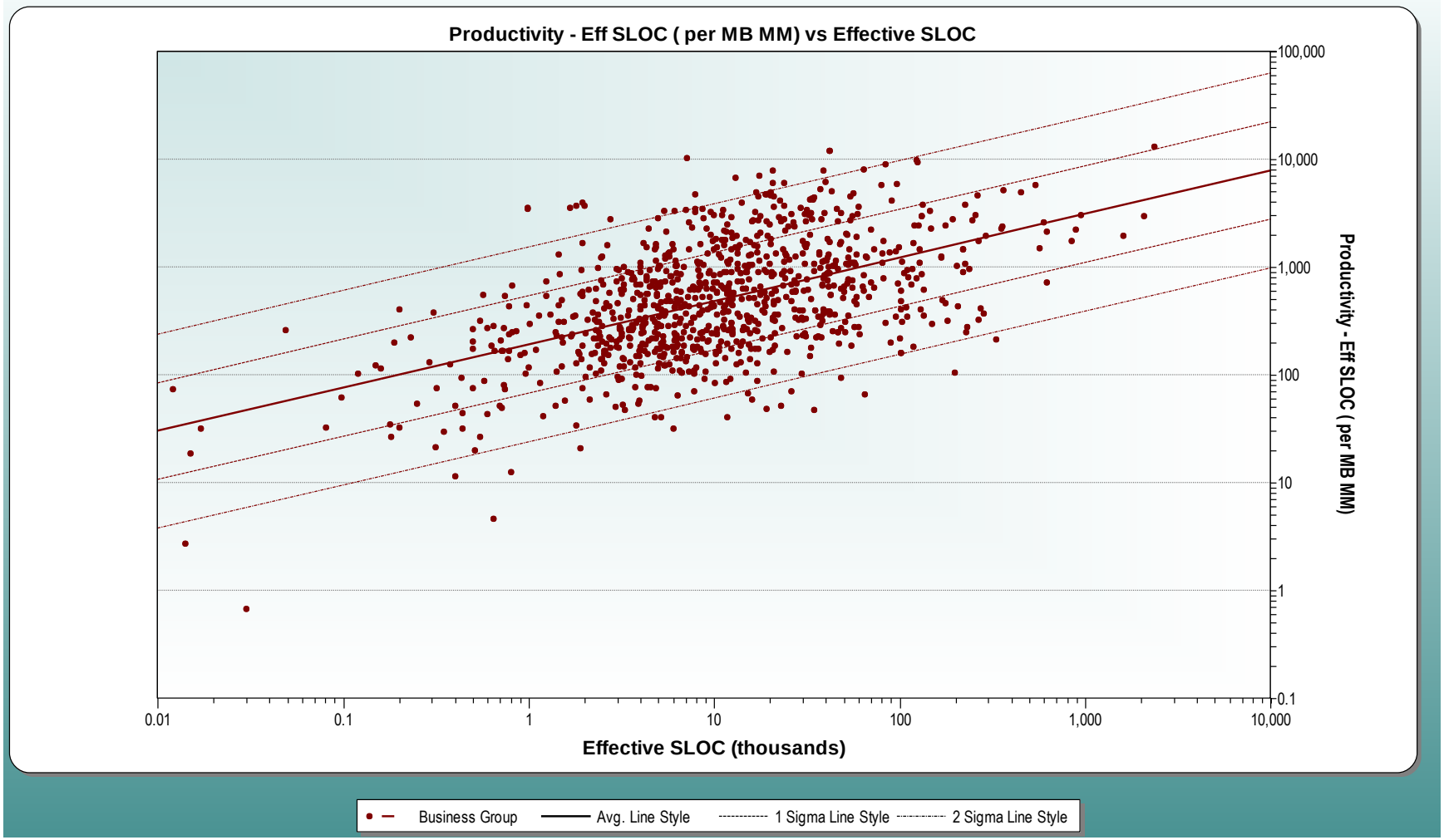
Value for Cost

- Now, more than ever, software projects need to efficiently deliver reliable software.
- However, many development plans unintentionally guarantee a less than optimal result.
- In addition, the plans either do not specify required system reliability or, if they do, the plan is not based on forecasts of when the minimum acceptable reliability will be reached.



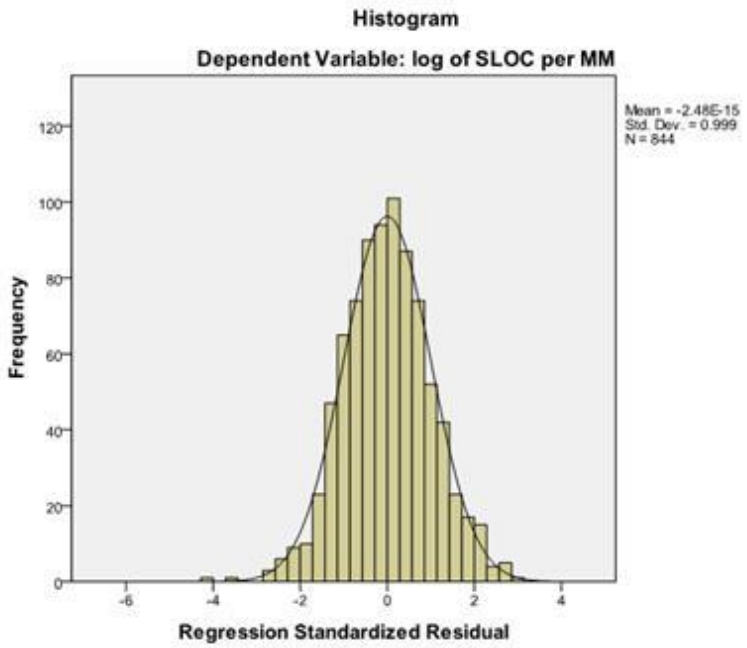
"Too many Government IT projects cost hundreds of millions of dollars more than they should, take years longer than necessary to deploy, and deliver technologies that are obsolete by the time they are completed."
OMB memo, June 28, 2010

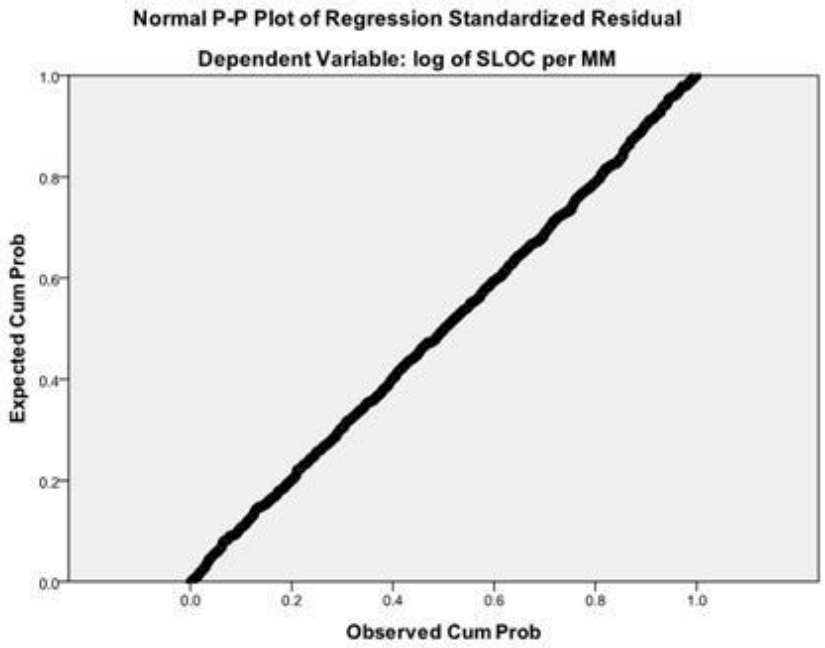
Productivity and Size



Productivity Paradox: Residuals of Productivity versus Size

Consistent relationship across the entire size range of projects





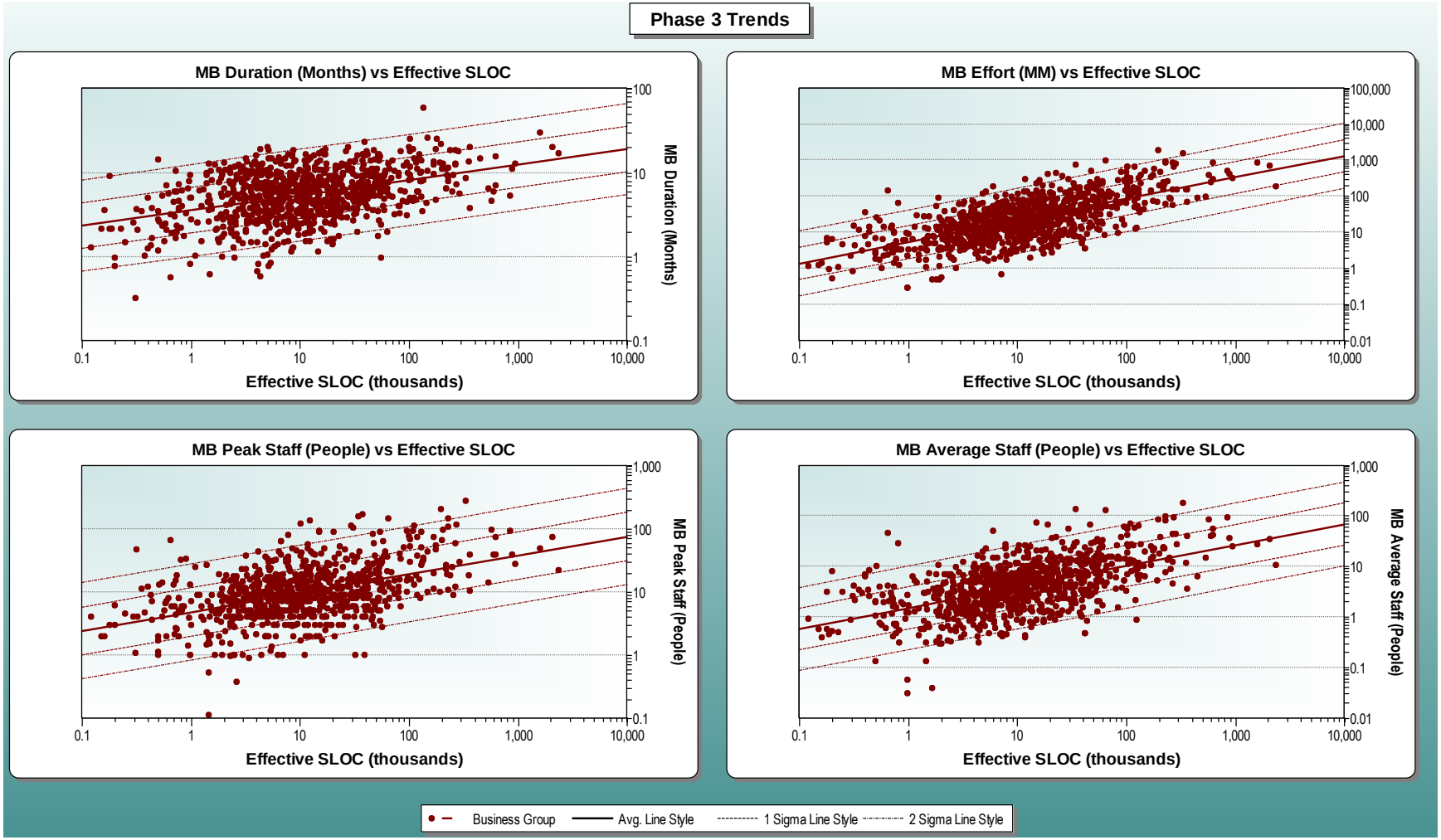
Yes, it is true!

Productivity Size Paradox

- On average, productivity is higher on larger projects (!)
- This is true for both ratio based productivity measures (size / effort) as well as exponential measures (PI, Productivity Index)
- Simple linear productivity (size over effort) is insufficient to plan or estimate projects

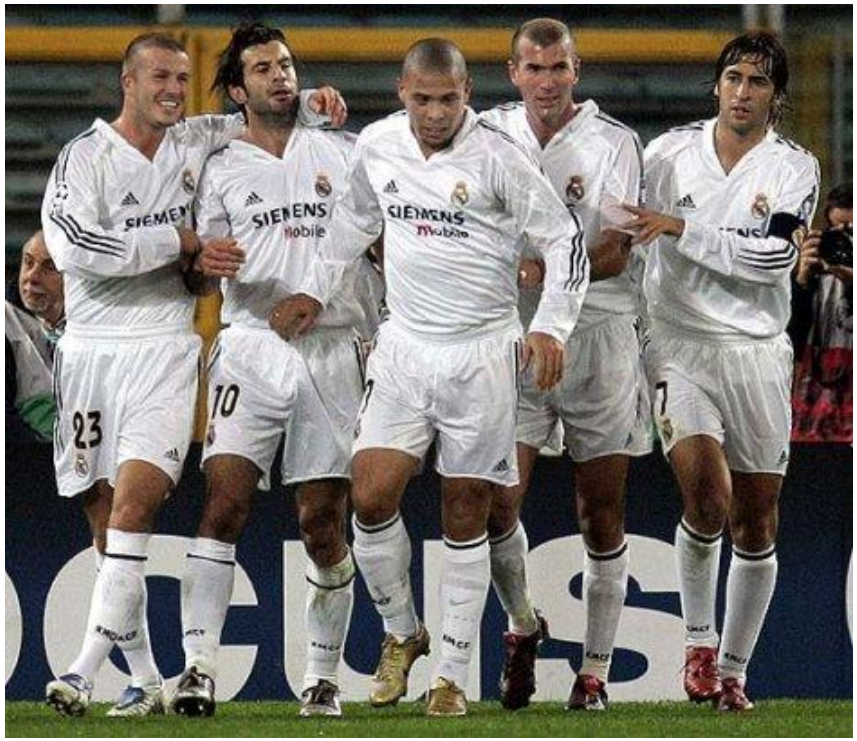
"…Our estimating techniques fallaciously confuse effort with progress, hiding the assumption that men and months are interchangeable."
– Fred Brooks

Trends: Duration, Effort, and Staff

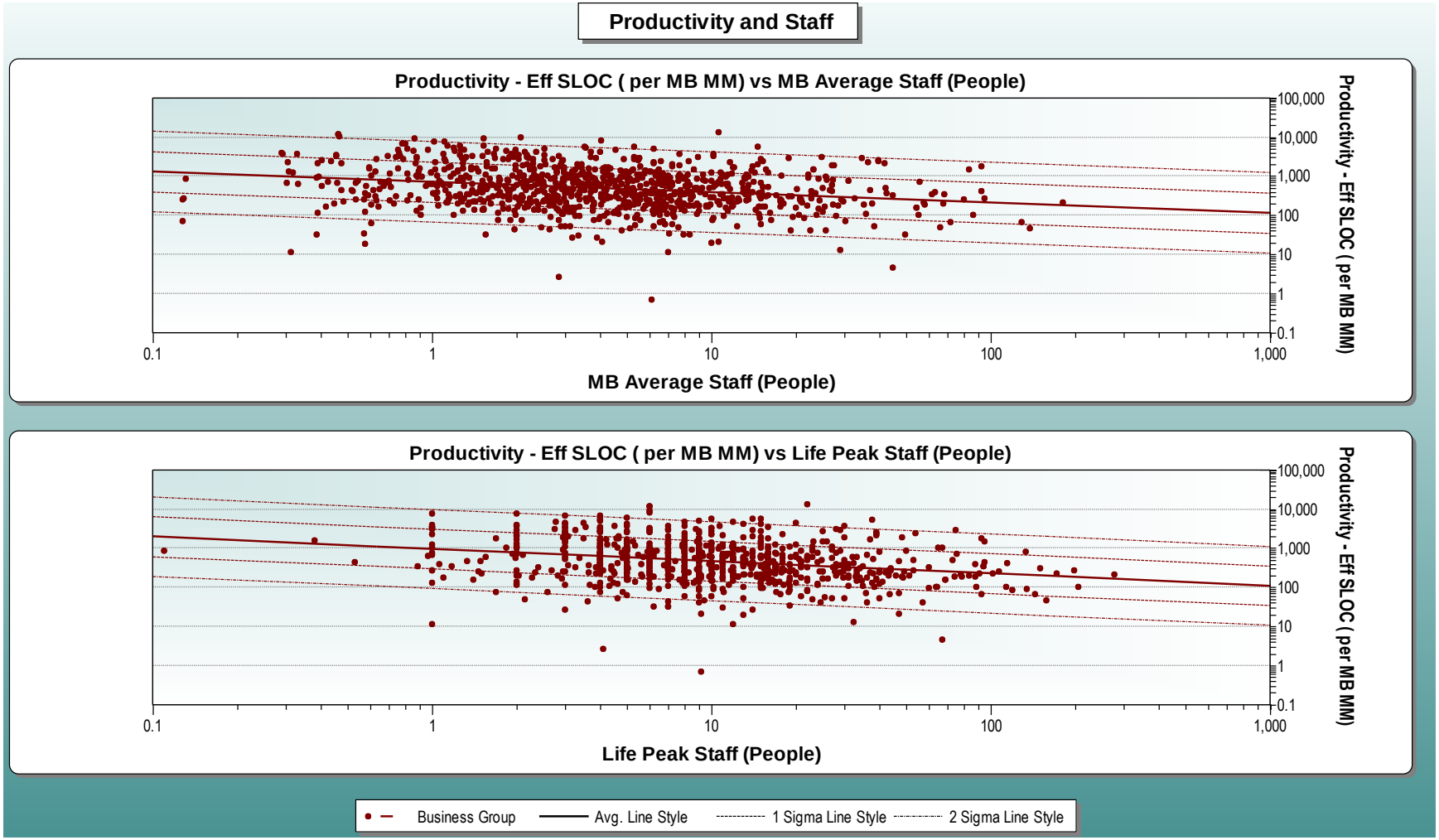


Productivity: But, wait!

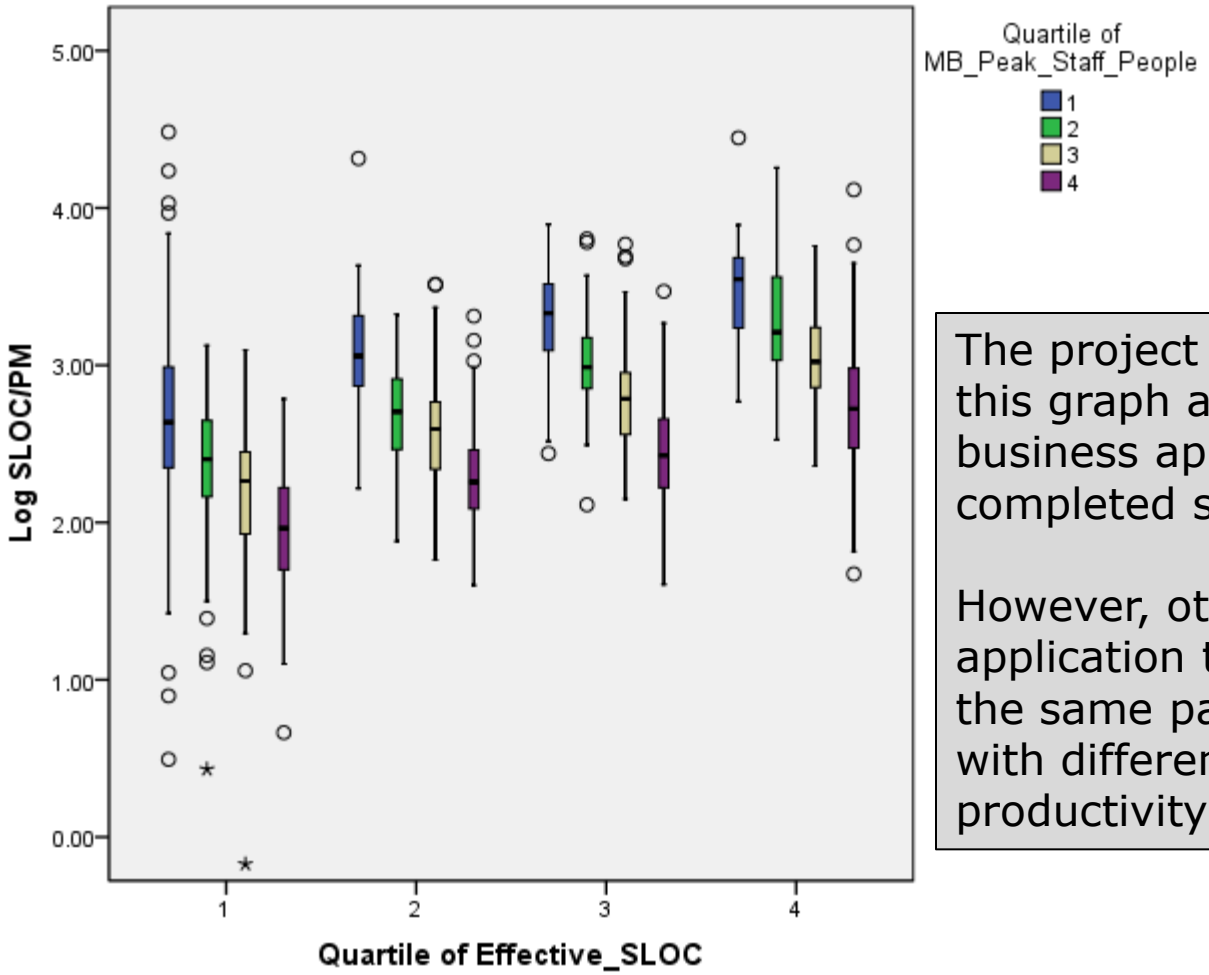
- If productivity increases with project size, why are projects with large teams generally less productive than projects with smaller teams?
- Let's look at the data in a different way…



Productivity and Staff



Staff, Productivity, and Size



The project metrics in this graph are from business applications completed since 2000.

However, other application types exhibit the same pattern (albeit with different productivity ranges).

Team Size

- In general, smaller teams are more productive. However: Optimal team size is not independent of application size
- Match the team size to the work, rather than overstaff the project in hopes of achieving a dramatic schedule reduction

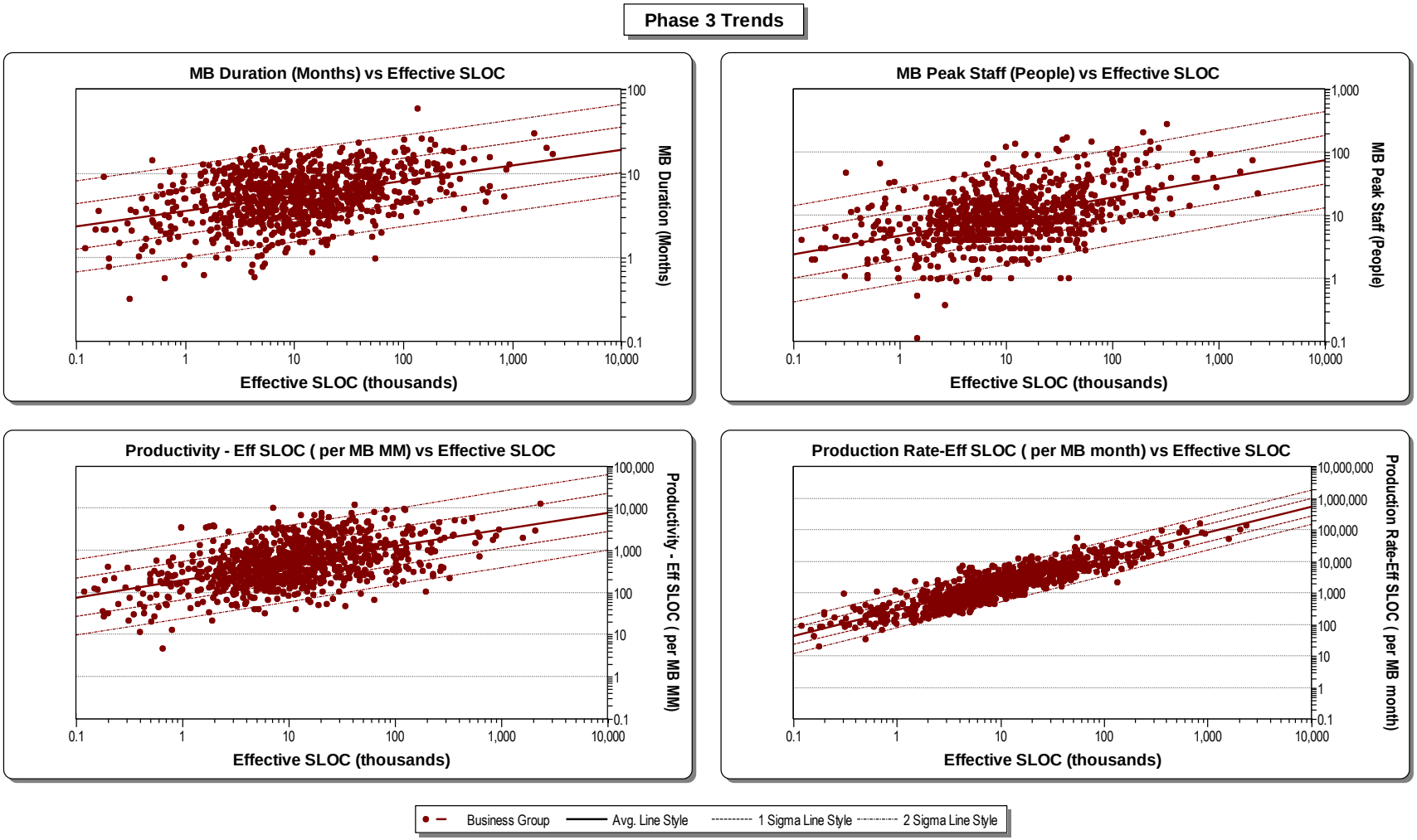
"The pattern that management follows in adding staff to a project affects the development time, effort and cost." – Larry Putnam, Sr.

Duration

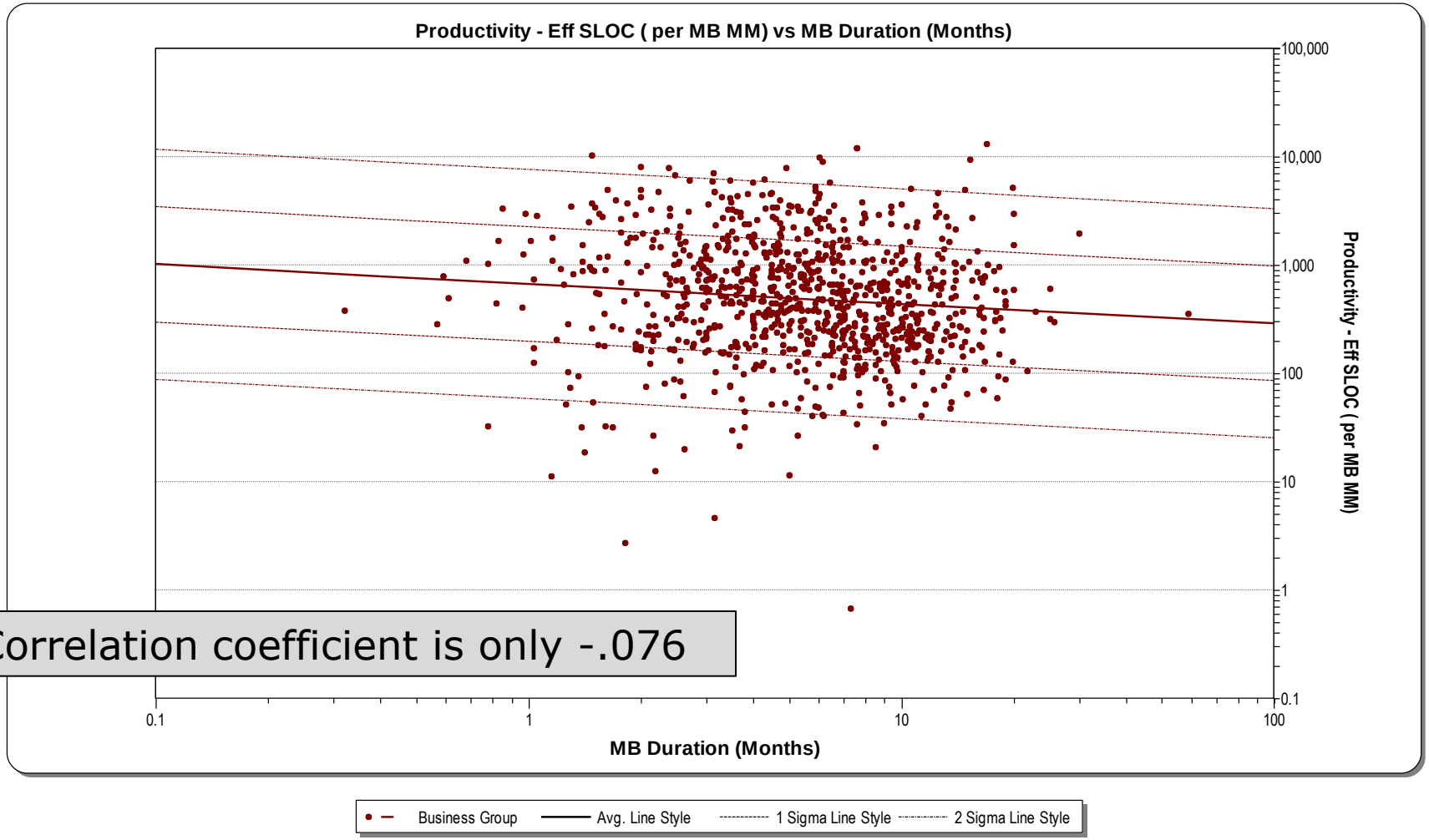
- How does project duration relate to productivity?
- Duration, Productivity, Staff, Production Rate all have a positive correlation with size, but the relative rates are important

"More software projects have gone awry for lack of calendar time than for all other causes combined." – Fred Brooks

Duration, Staff, Productivity, Production Rate

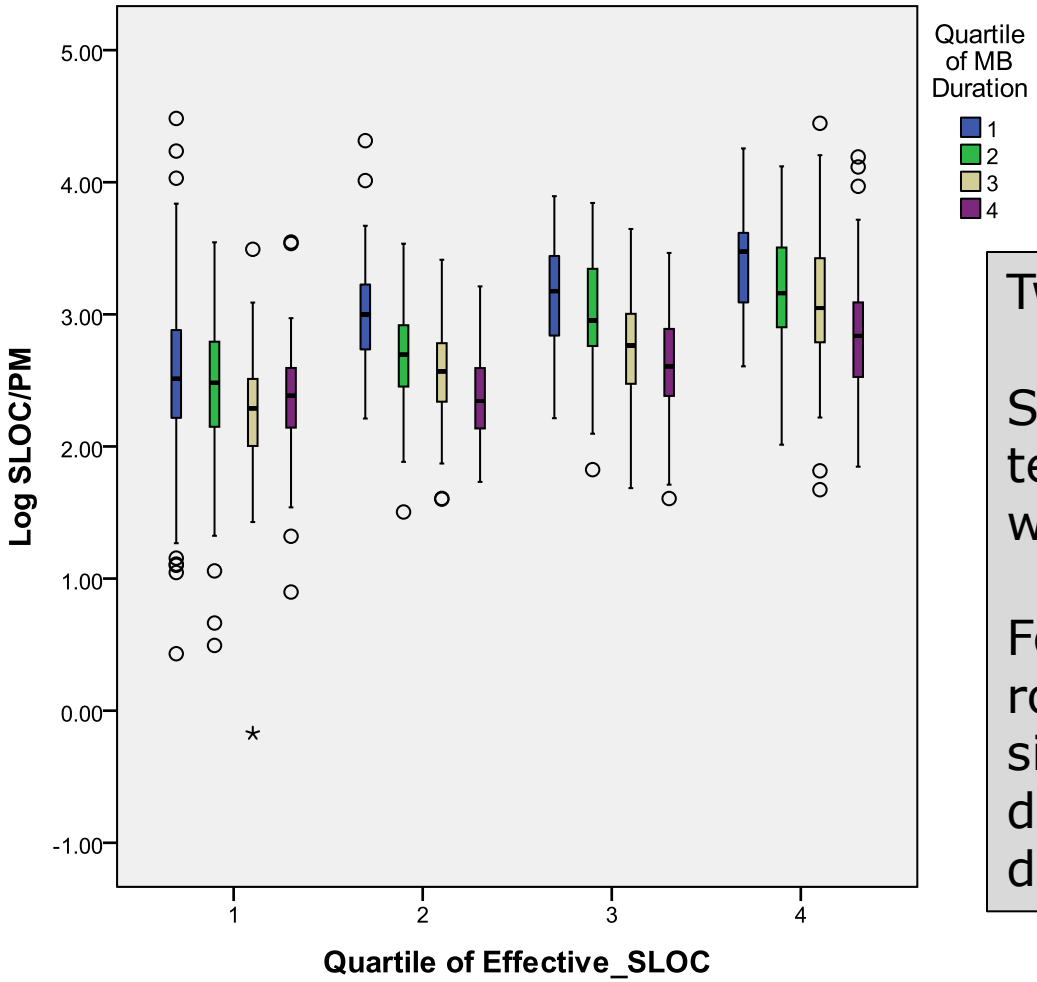


Productivity and Duration: Maybe there is no relationship after all?



Correlation coefficient is only -.076

Duration, Productivity and Size



Two observations:

Simple productivity tends to increase with size.

For projects of roughly the same size, productivity decreases as duration increases.

Putting it All Together

- The equation for simple productivity (size over effort) does not take duration into account, and yet duration has an effect on productivity
- Given the importance of schedule to most projects, duration needs to be factored into the productivity equation
- Recall that not all staffing strategies are equal. The most productive projects tend to use smaller teams.
- Note that it is difficult to determine causation, projects may have short duration because they had high productivity

Reliability

- Software is often part of a hardware system (device, airframe, ground vehicle, etc.) that has an overall reliability target
- Reliability needs to match the mission profile
- Software Target Reliability = System Target Reliability / (software defects / total defects)
- As we saw with productivity, defect ratios (defects per size) do not adequately reflect the non linear relationships

A Note on Application Types

- We often group application types into three "super groups", as project performance is similar within each of these groups:
  - The Business group includes business (IT) systems.
  - The Engineering group includes: command & control; system software; telecommunications; scientific; and process control.
  - The Real time group includes: avionics; real time; microcode & firmware.

MTTD

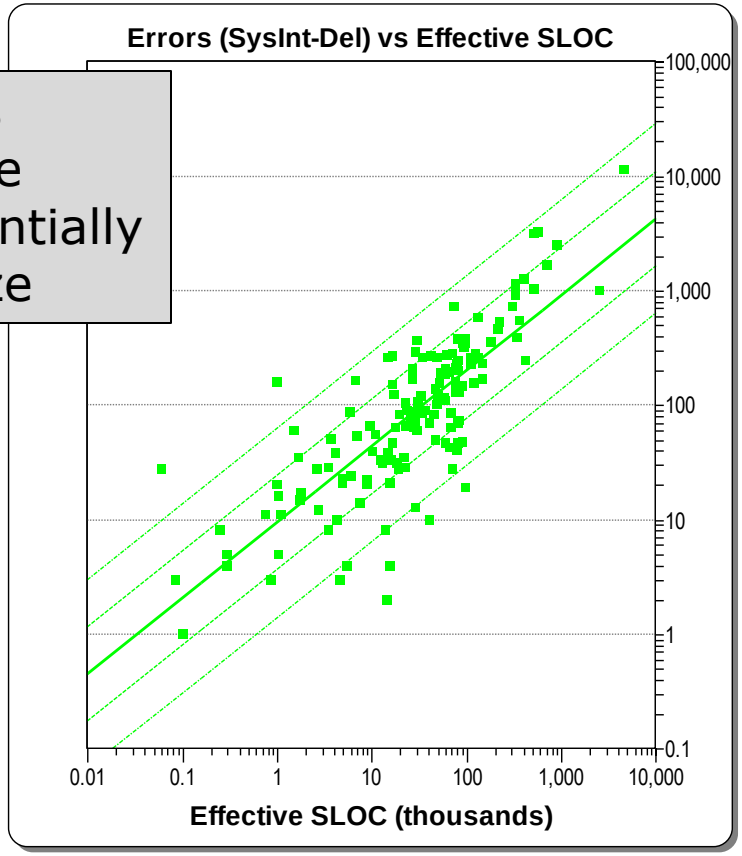
- The Mean Time to Defect is simply the reciprocal of the defect discovery rate.
- MTTD is particularly useful as a reliability metric. Average time between discoveries of defects can be compared to the software's required mission profile to decide when the software is "reliable enough" to be put into production.
- Mission critical or high reliability software will have a higher minimum MTTD than a typical IT application.
- An alternative is to use Mean Time to Failure (MTTF) where only high severity defects (e.g. Critical and Serious) are counted.

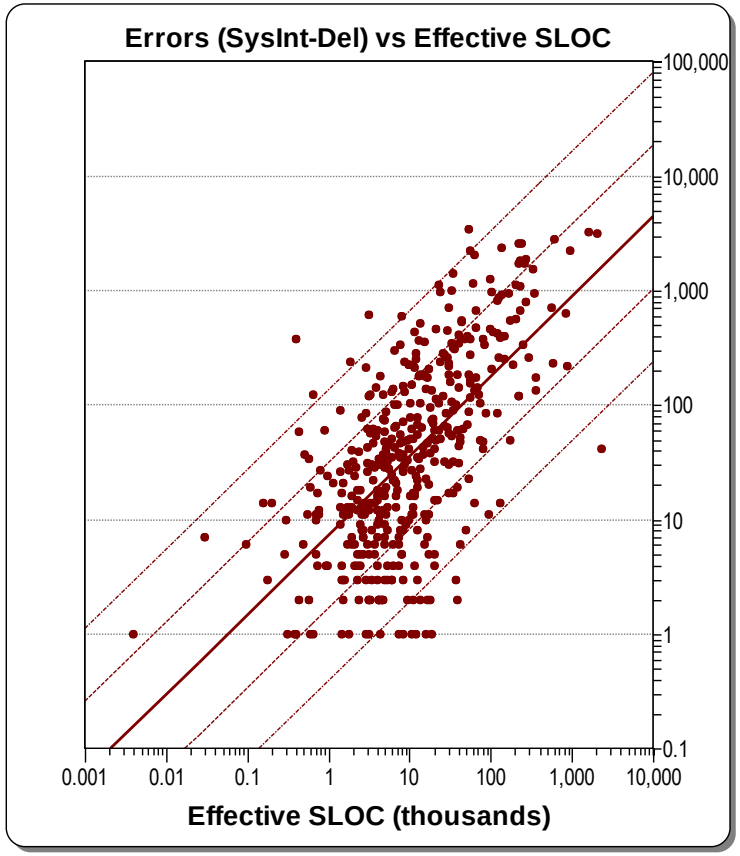
Defects and Size

Graphs include only Systems delivered after year 2000

Quality Trends

Defects increase exponentially with size





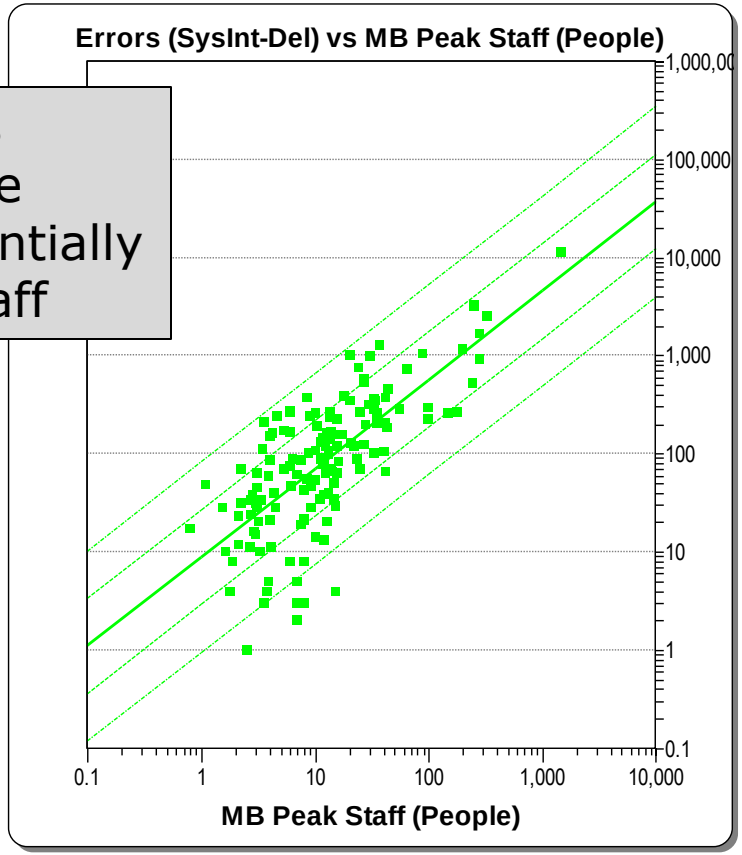
Engineering Group · Business Group — Avg. Line Style ---------- 1 Sigma Line Style ---------- 2 Sigma Line Style

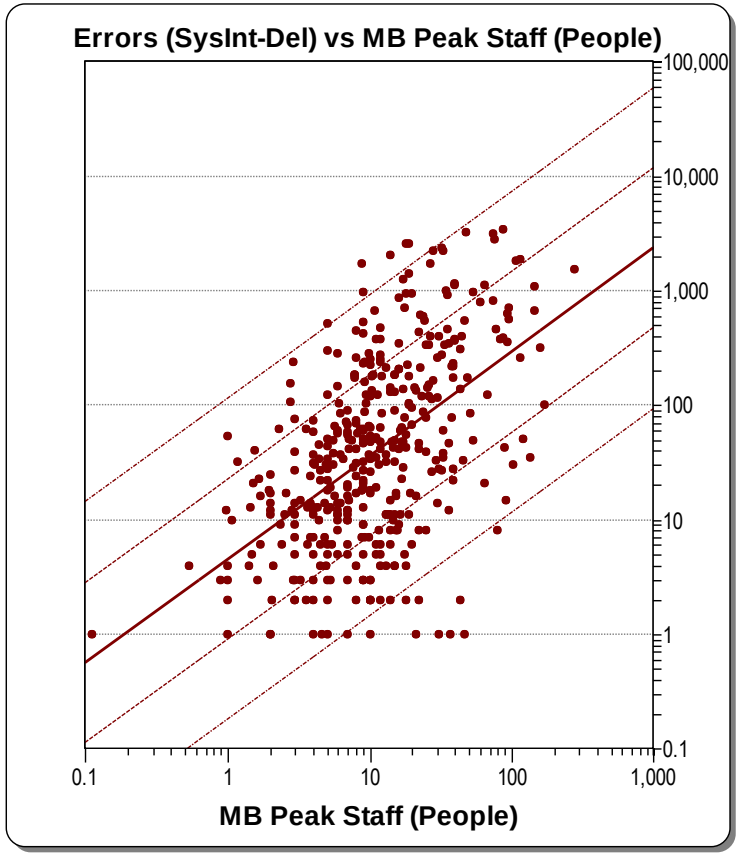
Defects and Peak Staff

Graphs include only Systems delivered after year 2000

Quality Trends

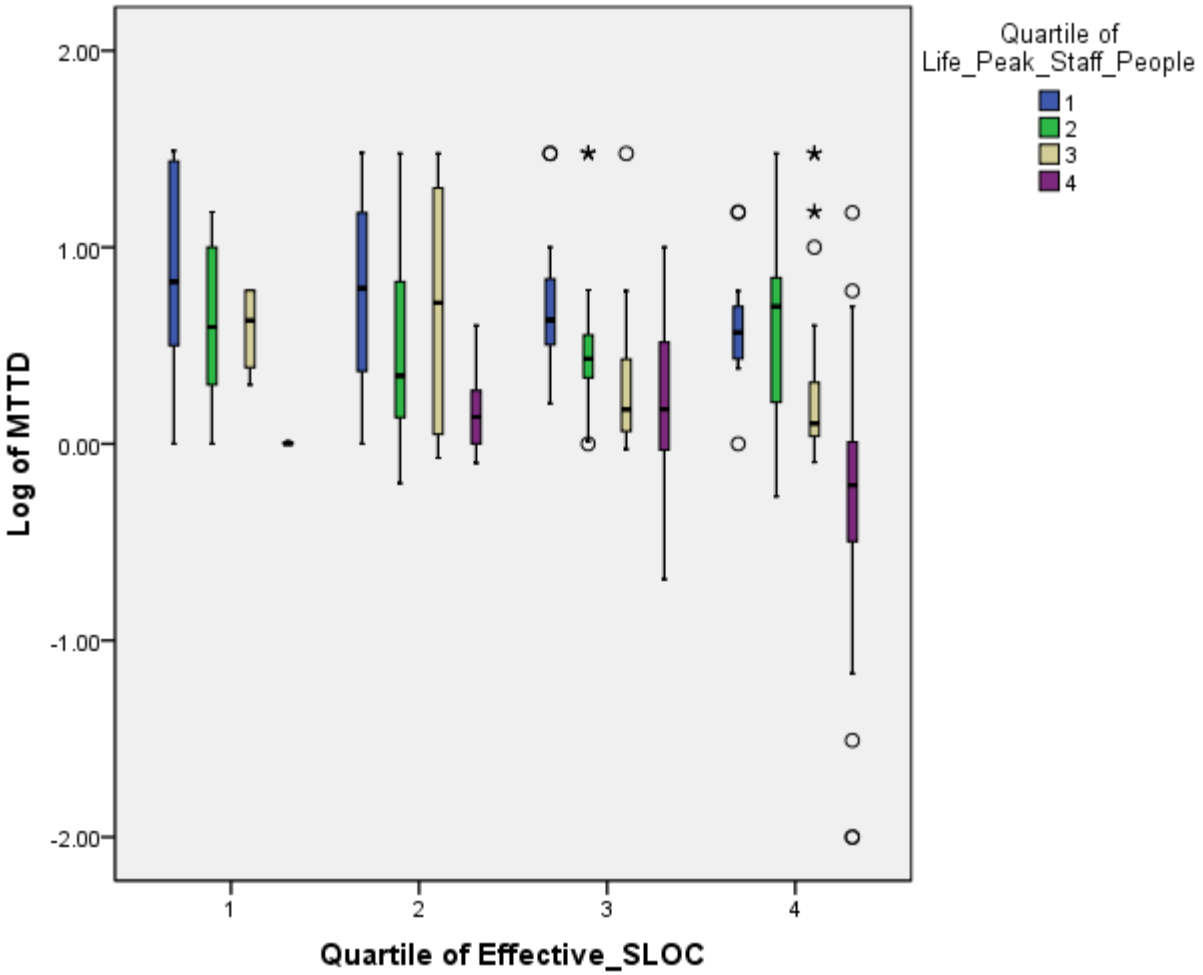
Defects increase exponentially with staff





Engineering Group · Business Group — Avg. Line Style ---------- 1 Sigma Line Style ---------- 2 Sigma Line Style

So which is more important?



MTTD (Days)

Size (ESLOC)	Business	Engineering	Real Time
1000	8.1	7.6	35.1
10000	3.5	3.1	6.7
100000	1.5	1.3	1.3
500000	0.9	0.7	0.4

Size (FP)	Business
100	6.5
500	2.9
1000	2.0
5000	0.9
10000	0.6

Compare the target reliability to historical results for feasibility.

Estimate reliability based on project schedule.

Projects completed after the year 2000. MTTD during the first 30 days of system operation.

Summary

- The relationships between productivity, size, duration and staff are not simple, but they can be understood. The *relative* rates at which the factors change is very important.
- Reliability is related to size and staff (as well as productivity). Estimating and tracking reliability is important.
- Selecting an appropriate balance of all these factors for a project schedule is important to minimize cost and produce a reliable system

Resources

- Performance Benchmark Tables: http://www.qsm.com/resources/performance-benchmark-tables/index.html
- *Measures for Excellence*, Putnam and Myers, 1992.
- *The Mythical Man-Month: Anniversary Edition*, Brooks, 1995.
- *Five Core Metrics*, Putnam and Myers, 2003.
- QSM High Performance Benchmark Consortium, http://www.qsm.com/

Bonus Slide: Duration Prediction from Earliest Phase

Projects assigned to one of two groups for each phase: Group 1 has duration less than predicted, Group 2 has duration greater than predicted. The top table compares Feasibility with Code/Test phases. The second table compares Requirements with Code/Test phases.

Percentile Group of CTDev * Percentile Group of FeasDev Crosstabulation

			Percentile Group of FeasDev		
			1	2	Total
Percentile Group of CTDev	1	Count	175	83	258
		Expected Count	129.0	129.0	258.0
	2	Count	88	180	268
		Expected Count	134.0	134.0	268.0
Total		Count	263	263	526
		Expected Count	263.0	263.0	526.0

68% of projects that took longer than expected in Feasibility (column 2) also took longer than expected in Code/Test (row 2).

Percentile Group of CTDev * Percentile Group of RqtsDev Crosstabulation

			Percentile Group of RqtsDev		
			1	2	Total
Percentile Group of CTDev	1	Count	359	138	497
		Expected Count	248.5	248.5	497.0
	2	Count	141	362	503
		Expected Count	251.5	251.5	503.0
Total		Count	500	500	1000
		Expected Count	500.0	500.0	1000.0

72% of projects that took longer than expected in Requirements (column 2) also took longer than expected in Code/Test (row 2).

Chi Square test result indicates that a pattern this extreme would occur by chance less than 1 time in a thousand.