Anecdotal observations on software engineering productivity

Wayne K. Yamamoto

Engineering productivity

- How is productivity distributed across the organization?
- How are labor costs (salaries) distributed across the organization?
- What is the most efficient way to increase productivity through headcount adjustments?
- How does this affect your hiring?

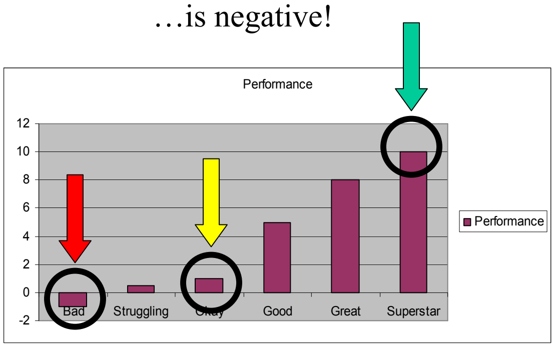
General Productivity

Productivity of the top performer…

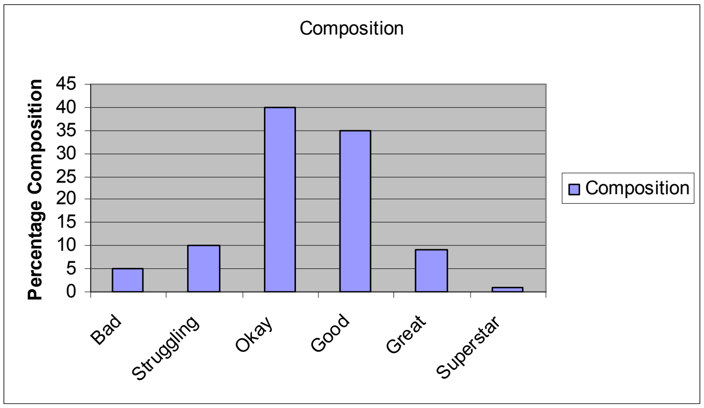
…is 10X that of the “average” performer!

Productivity of the worst performer…

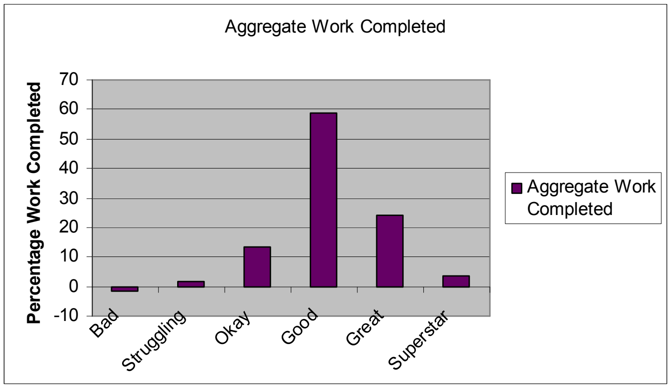
…is negative!



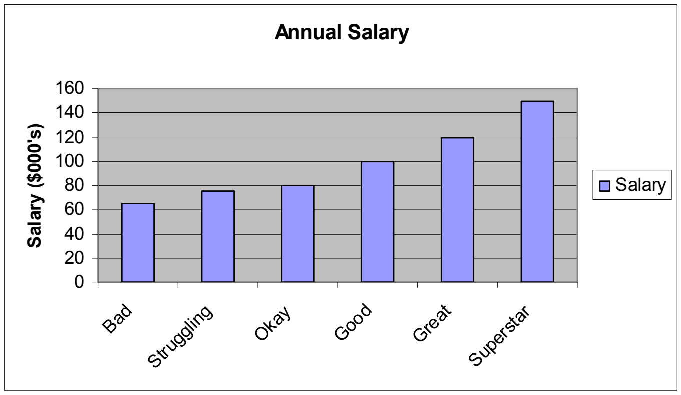
Personnel Mix

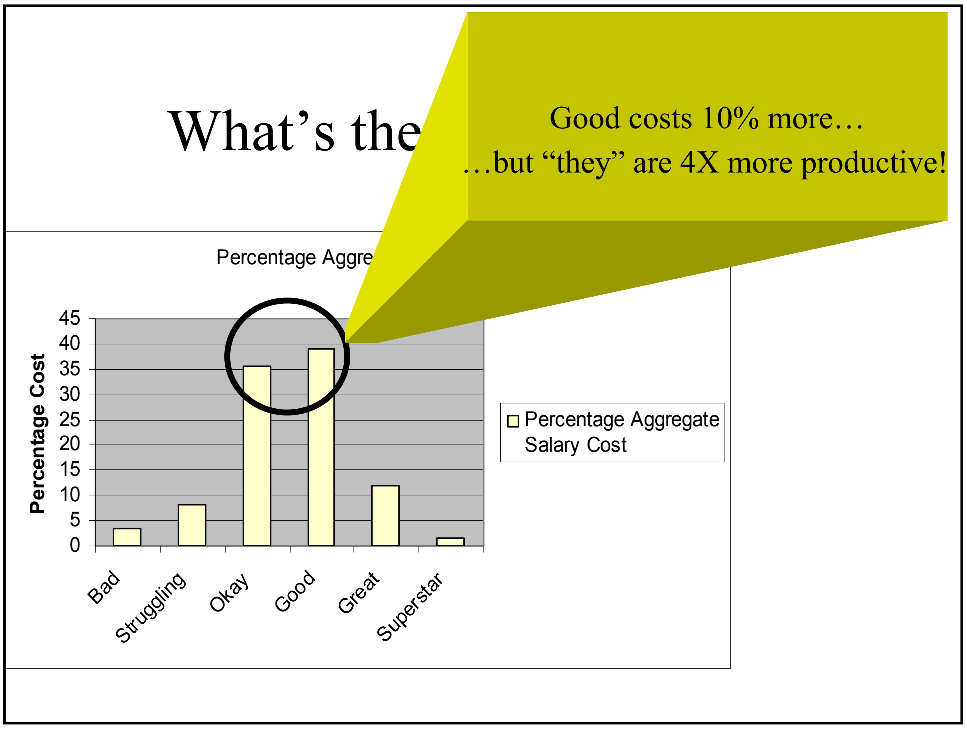


Who does the work?



What do they cost?

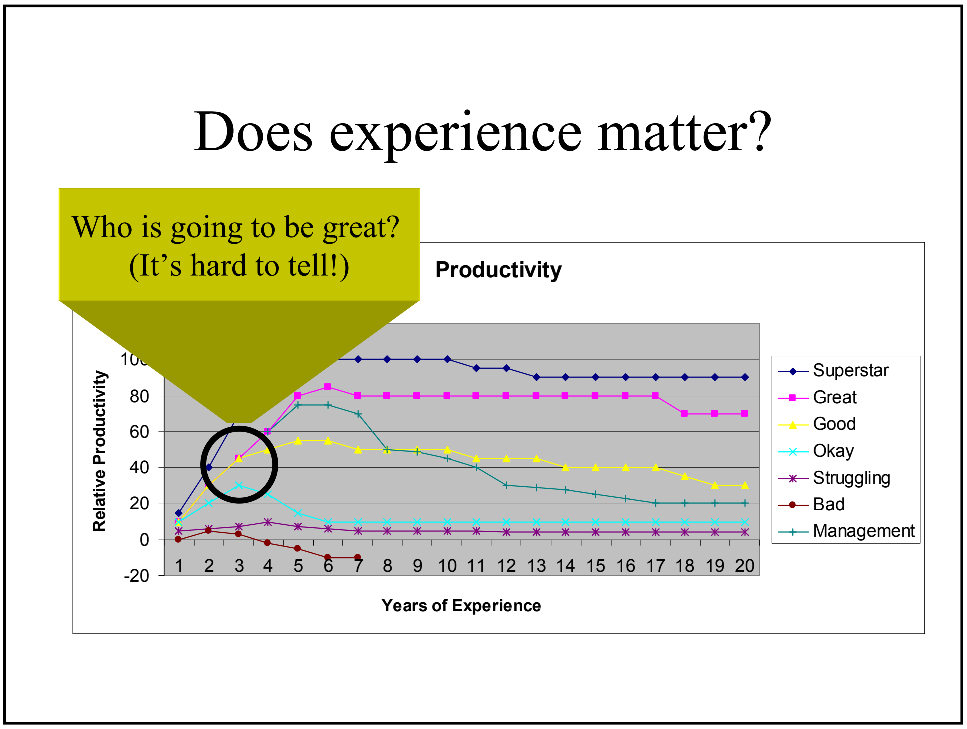




What's the

Good costs 10% more…
…but "they" are 4X more productive!

Percentage Aggre

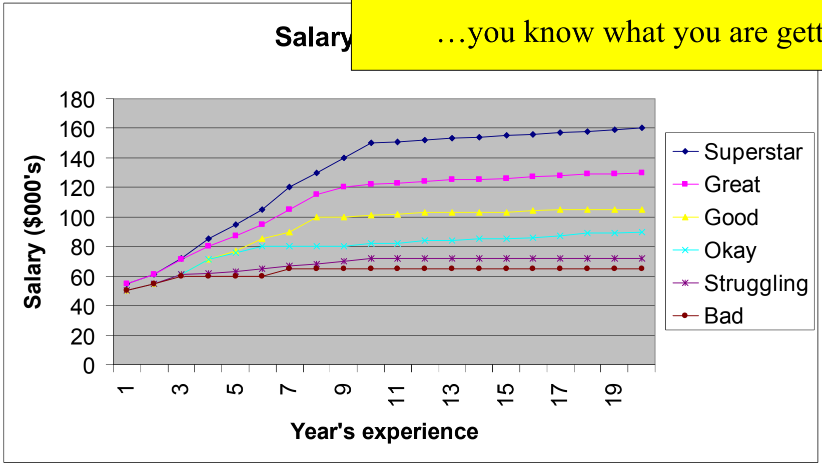


Does experience matter?

Who is going to be great?
(It's hard to tell!)

Productivity

The cost/v



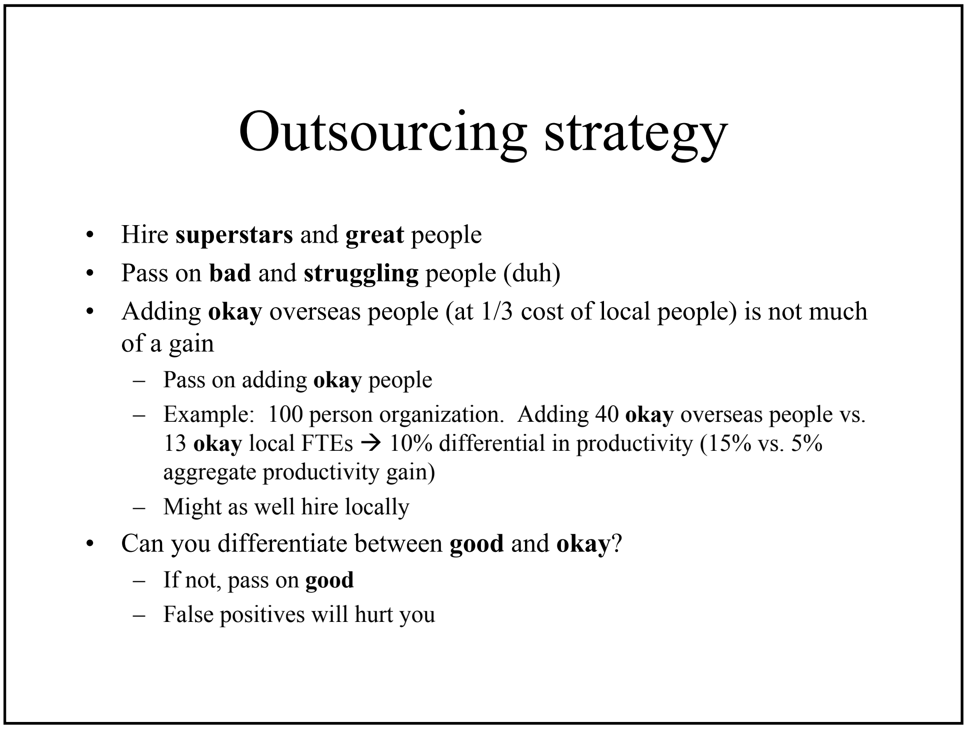
Experienced "good" personnel costs 25% more than "okay"…

…but you get 5X the productivity

…you know what you are getting

Conclusions

- Individual productivity is widely variant
- Compensation is less widely varying
- Hire/Retain **superstars**, **great**, and **good**
- Pass on/Release **bad** and **strugglers**
- Jury is out on **okay**
- "Overweight" compensation towards **superstars**, **great**, and **good**
- Differentiation between **good** and **okay** is tough → but critically important



Outsourcing strategy

- Hire **superstars** and **great** people
- Pass on **bad** and **struggling** people (duh)
- Adding **okay** overseas people (at 1/3 cost of local people) is not much of a gain
  - Pass on adding **okay** people
  - Example: 100 person organization. Adding 40 **okay** overseas people vs. 13 **okay** local FTEs → 10% differential in productivity (15% vs. 5% aggregate productivity gain)
  - Might as well hire locally
- Can you differentiate between **good** and **okay**?
  - If not, pass on **good**
  - False positives will hurt you