"Why Measure Performance"

Coding War Games: Observed Productivity Factors

Beginning in 1977, we have conducted some sort of a public productivity survey each year. So far, more than three hundred organizations worldwide have participated in these studies. From 1984 on, we have run our annual survey as a sort of public competition in which teams of software implementors from different organizations compete to complete a series of benchmark coding and testing tasks in minimal time and with minimal defects. We call these competitions *Coding War Game*s. Here's how they work:

♦   The basic competing unit is a pair of implementors from the same organization. The pair members do not work together, but in fact members work against each other as well as against all the other pairs.

♦   Both pair members perform exactly the same work, designing, coding, and testing a medium-sized program to our fixed specification.

♦   As they go through the exercise, participants record the time spent on a time log.
♦   After all participant testing is completed, the products are subjected to our standard acceptance test.

♦   Participants work in their own work areas during normal work hours using the same languages, tools, terminals, and computers that they use for any other project.

♦   All results are kept confidential.

From 1984 to 1986, more than 600 developers from 92 companies have participated in the games.

Productivity Non-Factors

In our analysis of game results, we discovered that the following factors had little or no correlation to performance:

• *Languag*e: Those who coded in old languages like COBOL and Fortran did essentially as well as those who coded in Pascal and C. The spread within each language group was much like the overall spread of performance. The only exception to this observation about language was assembly language: The assembly language participants got badly left behind by all the other language groups. (But, then, people who use assembly language are used to being left behind.)

*Years of experienc*e: People who had ten years of experience did not outperform those with two years of experience. There was no correlation between experience and performance except that those with less than six months' experience with the language used in the exercise did not do as well as the rest of the sample.

*Number of defect*s: Nearly a third of the participants completed the exercise with zero defects. As a group, the zero-defect workers paid no performance penalty for doing more precise work. (In fact, they took slightly less time, on the average, to complete the exercise than those who had one or more defects.)

*Salar*y: Salary levels varied widely over the sample. There was a very weak relationship between salary and performance. The half above the median made less than ten percent more than the half below, but they performed nearly twice as well. The performance spread at any given salary level was nearly as wide as over the whole sample.

Again, nothing very astonishing, as most of these effects have been noted before. Slightly more surprising were some of the factors that we found *did* have a substantial effect on performance.

You May Want to Hide This from Your Boss

Among our findings of what did correlate positively to good performance was this rather unexpected one: It *mattered a lot who your pair mate was.* If you were paired with someone who did well, you did well, too. If your pair mate took forever to finish, so did you. If your pair mate didn't finish the exercise at all, you probably didn't either. For the average competing pair, the two performances differed by only 21 percent.

Now, why is that so important? Because even though the pairs didn't work together, the two members of the pair came from the same organization. (In most cases, they were the only ones from that organization.) They worked in the same physical environment and shared the same corporate culture. The fact that they had nearly identical performances suggests that the wide spread of capabilities observed across the whole sample may not apply within the organization: Two people from the same organization tend to perform alike. That means the best performers are clustering in some organizations while the worst performers are clustering in others. This is the effect that Harlan Mills predicted in 1981:

> **While this [10 to 1] productivity differential among programmers is understandable, there is also a 10 to 1 difference in productivity among software organizations.**
>
> -Software Productivity

Our study found that there were huge differences between the 92 competing organizations. Over the whole sample, the best organization (the one with the best average performance of its representatives) worked 11.1 times faster than the worst organization. In addition to their speed, all competitors from the fastest organization developed code that passed the major acceptance test.

This is more than a little unsettling. Managers for years have affected a certain fatalism about individual differences. They reasoned that the differences were innate, so you couldn't do much about them. It's harder to be fatalistic about the clustering effect. Some companies are doing a lot worse than others. Something about their environment and corporate culture is failing to attract and keep good people or is making it impossible for even good people to work effectively.

Effects of the Workplace

The bald fact is that many companies provide developers with a workplace that is so crowded, noisy, and interruptive as to fill their days with frustration. That alone could explain reduced efficiency as well as a tendency for good people to migrate elsewhere.

The hypothesis that qualities of the workplace may have a strong correlation to developer effectiveness is an easy one to test. All you have to do is devise a set of fixed benchmark tasks, similar to those that developers do in their normal work, and observe how well they perform these tasks in different environments. The Coding War Games were designed with exactly that purpose in mind.

In order to gather some data on the workplace, we had each war game participant (prior to the exercise) fill out a questionnaire about the physical quarters in which the work was to be performed. We asked for some objective data (measurements of the dedicated space provided and height of partitions, for example) and for answers to some subjective questions, like "Does your workplace make you feel appreciated?" and "Is your workplace acceptably quiet?" Then we correlated their answers to their performance in the exercise.

An easy way to spot the trend is to look at the workplace characteristics of people who did well in the exercise (based on a composite performance parameter) against those of participants who didn't do so well. We chose to compare the top quarter of finishers with the bottom quarter. Average performance of those in the top quarter was 2.6 times better than that of those in the bottom quarter. The environmental correlations are summarized in Table 8.3:

Table 8.3
**Environments of the Best and Worst Performers
in the Coding War Games**

Environmental Factor	Those Who Performed in 1st Quartile	Those Who Performed in 4th Quartile
1. How much dedicated work space do you have?	78 sq. ft.	46 sq. ft.
2. Is it acceptably quiet?	57% yes	29% yes
3. Is it acceptably private?	62% yes	19% yes
4. Can you silence your phone?	52% yes	10% yes
5. Can you divert your calls?	76% yes	19% yes
6. Do people often interrupt you needlessly?	38% yes	76% yes

The top quartile, those who did the exercise most rapidly and effectively, work in space that is substantially different from that of the bottom quartile. The top performers' space is quieter, more private, better protected from interruption, and there is more of it.

What Did We Prove?

The data presented above does not exactly prove that a better workplace will help people to perform better. It may only indicate that people who perform better tend to gravitate toward organizations that provide a better workplace. Does that really matter to you? In the long run, what difference does it make whether quiet, space, and privacy.

help your current people to do better work or help you to attract and keep better people?

If we proved anything at all, it's that a policy of default on workplace characteristics is a mistake. If you participate in or manage a team of people who need to use their brains during the work day, then the workplace environment is your business. It isn't enough to observe, "You never get anything done around here between 9 and 5," and then turn your attention to something else. It's dumb that people can't get work done during normal work hours. It's time to do something about it.

Peopleware: Productive Projects and Teams – *Second Edition-pages 44-50 Dorset House Publishing Co. 1999 - Reprinted with the permission of the authors Tom DeMarco & Timothy Lister*
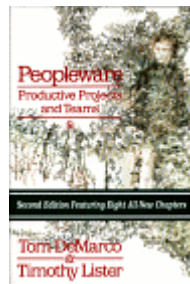
Peopleware : Productive Projects and Teams, 2nd Ed.
by Tom Demarco, Timothy Lister

Editorial Reviews
Amazon.com
Peopleware asserts that most software development projects fail because of failures within the team running them. This strikingly clear, direct book is written for software development-team leaders and managers, but it's filled with enough commonsense wisdom to appeal to anyone working in technology. Authors Tom DeMarco and Timothy Lister include plenty of illustrative, often amusing anecdotes; their writing is light, conversational, and filled with equal portions of humor and wisdom, and there is a refreshing absence of "new age" terms and multistep programs. The advice is presented straightforwardly and ranges from simple issues of prioritization to complex ways of engendering harmony and productivity in your team. *Peopleware* is a short read that delivers more than many books on the subject twice its size. *--This text refers to an out of print or unavailable edition of this title.*

The publisher, Dorset House Publishing, dhpubco@aol.com, 212) West 12th St., New York, NY 10014 , March 19, 1999
Peopleware Is Now Updated with Eight New Chapters!

Two of the computer industry's best-selling authors and lecturers return with a new edition of their 1987 classic on the management of software development.

With humor and wisdom drawn from years of management and consulting experience, DeMarco and Lister demonstrate that the major issues of software development are human, not technical -- and that managers ignore them at their peril.

Now, with a new Preface and eight new chapters, the authors enlarge upon their previous ideas and add fresh... read more

---

Customer Reviews of the Day

★★★★★ **Classic text for middle managers**, August 23, 2000
Reviewer: Gary Powell (see more about me) from Renton, WA USA
If you are moving into new office space and want to justify giving everyone an office, this is the book for you. The classic answer is "cubicals are cheaper." Peopleware refutes this bad idea with hard data you can use. Studies at reputable companies, white papers and cost analysis. Things even an accountant will understand. Its a book for "moles" in the corporate organization. People who are trying to humanize the workplace and yet still get something done.

Have a new manager? Need to give them some help in managing you? Give them this book. It will make your life a lot easier. The other target audience is the H.R. Dept. Unfortunately they usually don't know about it, nor can they affect the changes necessary to make things better. However if they are on the same band wagon it will make the fight for decent work space easier.

Its a classic along with "The Mythical Man Month", and the now out of print "The 59 Second employee." So if you are up late wondering how your projects got so late, why your staff is complaining/leaving, read this book and get a clue.

★★★★★ **Anyone managing software projects should read this!**, March 30, 2000
Reviewer: Joel Spolsky (see more about me) from New York, NY
As summer interns at Microsoft, my friends and I used to take "field trips" to the company supply room to stock up on school supplies. Among the floppy disks, mouse pads, and post-it notes was a stack of small paperback books, so I took one home to read.

The book was Peopleware, by Tom DeMarco and Timothy Lister. This book was one of the most influential books I've ever read. The best way to describe it would be as an Anti-Dilbert Manifesto.

Ever wonder why everybody at Microsoft gets their own office, with walls and a door that shuts? It's in there. Why do managers give so much leeway to their teams to get things done? That's in there too. Why are there so many jelled SWAT teams at Microsoft that are remarkably productive? Mainly because Bill Gates has built a company full of managers who read Peopleware. I can't recommend this book highly enough. It is the one thing every software manager needs to read... not just once, but once a year.

★★★★★ **Hard numbers on good work environments**, February 25, 2001
Reviewer: David Walker (see more about me) from Melbourne, Victoria, Australia
Summed up in one sentence, Peopleware says this: give smart people physical space, intellectual responsibility and strategic direction. DeMarco and Lister advocate private offices and windows. They advocate creating teams with aligned goals and limited non-team work. They advocate managers finding good staff and putting their fate in the hands of those staff. The manager's function, they write, is not to make people work but to make it possible for people to work.

Why is Peopleware so important to Microsoft and a handful of other successful companies? Why does it inspire such intense devotion amongst the elite group of people who think about software project management for a living? Its direct writing and its amusing anecdotes win it friends. So

does its fundamental belief that people will behave decently given the right conditions. Then again, lots of books read easily, contain funny stories and exude goodwill. Peopleware's persuasiveness comes from its numbers - from its simple, cold, numerical demonstration that improving programmers' environments will make them more productive.

The numbers in Peopleware come from DeMarco and Lister's Coding War Games, a series of competitions to complete given coding and testing tasks in minimal time and with minimal defects. The Games have consistently confirmed various known facts of the software game. For instance, the best coders outperform the ten-to-one, but their pay seems only weakly linked to their performance. But DeMarco and Lister also found that the best-performing coders had larger, quieter, more private workspaces. It is for this one empirical finding that Peopleware is best known.

(As an aside, it's worth knowing that DeMarco and Lister tried to track down the research showing that open-plan offices make people more productive. It didn't exist. Cubicle makers just kept saying it, without evidence - a technique Peopleware describes as "proof by repeated assertion".)

Around their Coding Wars data, DeMarco and Lister assembled a theory: that managers should help programmers, designers, writers and other brainworkers to reach a state that psychologists call "flow" - an almost meditative condition where people can achieve important leaps towards solving complex problems. It's the state where you start work, look up, and notice that three hours have passed. But it takes time - perhaps fifteen minutes on average - to get into this state. And DeMarco and Lister that today's typical noisy, cubicled, Dilbertesque office rarely allows people 15 minutes of uninterrupted work. In other words, the world is full of places where a highly-paid and dedicated programmer or creative artist can spend a full day without ever getting any hard-core work. Put another way, the world is full of cheap opportunities for people to make their co-workers more productive, just by building their offices a bit smarter.

A decade and a half after Peopleware was written, and after the arrival of a new young breed of IT companies called Web development firms, it would be nice to think DeMarco and Lister's ideas have been widely adopted. Instead, they remain widely ignored. In an economy where smart employees can increasingly pick and choose, it will be interesting to see how much longer this ignorance can continue.

★★★★★ **Rethinking the Way We Work and Manage**, February 12, 2001
Reviewer: **njbookworm (see more about me)** from New Jersey
My boss gave this to each member of our team as a Christmas gift. He admitted that his motivation was quite selfish as he wanted us to adopt the principles outlined in the text. It turned out to be a great gift!

Have you often wondered why you've felt stifled and unmotivated? Have you ever wondered why, despite all of your efforts, you can't get a project completed until after hours? Do you find yourself arriving earlier and earlier and getting less and less done? This book speaks to those issues by describing how the modern, cubicle-studded workplace and unenlightened managers make it difficult to work, produce, and succeed. The authors' insights are easy-to-digest as they give practical, real-life examples of their observations. Additionally, they offer tips on how to change personal strategies and your environment to heighten personal fulfillment and productivity.

On a personal note, I had wondered why my productivity has not been great the past several weeks. While reading this book, I traced the change back to a switch in our office's chemistry. A

new team member recently moved into our 4-person office. (This guy is in sales. This requires him to make lots of boisterous phone calls and frequently entertain visitors. I'm a writer who requires quiet and concentration with occasional collaboration. In short, it's a recipe for disaster!) For a while, I contemplated asking for a desk reassignment but thought I was just being picky. This book taught me that I wasn't being oversensitive. I was wasting my time and costing the company money! The result? I have scheduled an appointment with our Office Manager to request a change.

This book profoundly changed the way I think about management. Very highly recommended.

P *d* K Consulting

PdK Consulting focuses on **Workplace** performance optimization and on measuring the resulting performance improvements.

Using the "**Balanced Scorecard**", PdK identifies work group performance variables and establishes baseline performance measures. PdK develops a Workplace Strategy and a detailed Action Map with accountabilities and schedules aimed at:

♦ Implementing strategies and tactics for high performance workplaces
♦ Developing processes for continuing improvements
♦ Deploying knowledge supporting technologies
♦ Enabling behavior changes increasing performance

PdK measures the resulting **performance improvements** (business results) on a simple linear scale via multi-variate statistical or simple heuristic models.

Gérald de Kerchove P *d* K Consulting	Tel: +1 (415) 453 6902 Fax: +1(415) 456 9041 Mobile: +1 (415) 246 4156
50 Palm Avenue San Rafael, CA 94901 U.S.A.	mailto:GdeKerchove@PdKConsulting.com http://www.PdKConsulting.com/