Quantifying Soft Factors

Steve McConnell

The role that soft, human-oriented factors play in software effectiveness sometimes gets lost in discussions of best practices, process models, and other more complex topics.

Limited Importance of Process Maturity

One of the articles in this issue, Brad Clark's "Quantifying the Effect of Process Improvement," has a perhaps startling message for more process-oriented readers: a one-CMM-level improvement by itself accounts for only an 11% increase in productivity.[1] In comparing medium-size projects (100,000 lines of code), the one with the worst process will require 1.43 times as much effort as the one with the best process, all other things being equal. In other words, the maximum influence of process maturity on a project's productivity is 1.43.

What Clark doesn't emphasize is that for a program of 100,000 lines of code, several human-oriented factors influence productivity more than process does. According to the regression studies performed to calibrate the Cocomo II estimation model,[1] analyst experience (AEXP in Cocomo II) exerts an influence of 1.51. Communications factors (SITE, which refers to colocation of personnel and communication support such as e-mail and networks) exert an influence of 1.52. Personnel continuity (PCON) has an influence of 1.59. Programmer capability (PCAP) has an in-

fluence of 1.77, and the capability of the requirements analysts (ACAP) has an influence of 2.00. Language and tool experience (LTEX) exert the same influence as process maturity (1.43), and, trailing only slightly, programmer experience (PEXP) exerts an influence of 1.40.

The effect of process maturity varies with project size; it has less influence in small projects. For a project of 10,000 lines of code, process maturity affects productivity less than any of the other factors mentioned.

The seniority-oriented factors alone (AEXP, LTEX, and PEXP) exert an influence of 3.02. The seven personnel-oriented factors (ACAP, AEXP, LTEX, PCAP, PCON, PEXP, and SITE) collectively exert a staggering influence range of 25.8! This simple fact accounts for much of the reason that non-process-oriented organizations such as Microsoft, Amazon.com, and other entrepreneurial powerhouses can experience industry-leading productivity while seemingly shortchanging process.

Physical Environment

One soft factor that the Cocomo II analysis doesn't quantify is the effect of the office environment. Tom DeMarco and Timothy Lister sponsored a now-famous programming competition in which 166 developers competed on the basis of quality and speed.[2] The competitors provided information on the characteristics of their physical work environments, and it turned out that the developers who performed in the top 25% had bigger, quieter, more private offices and fewer interruptions from people and phone calls than the other

75%. The differences in physical space weren't especially dramatic. The top 25% of performers had an average of 78 square feet of dedicated floor space; the bottom 25% had 46 square feet.

The differences in productivity were more dramatic. Developers in the top quartile had productivity 2.6 times better than developers in the bottom quartile. In Cocomo II terms, the influence of office environment is 2.6, which is significantly greater than that of process maturity.

Motivation

Motivation is usually thought to be the greatest influence on how well people perform, and most productivity studies agree.[3]

Whatever else its critics might say about Microsoft, everyone agrees that it has succeeded in motivating its developers to an extraordinary degree. Stories of 12-, 14-, even 18-hour days are common, as are stories of people who live in their offices for weeks at a time. I know of one developer who had a Murphy bed custom-made to fit his office. Locally, Microsoft is known as "The Velvet Sweatshop," which suggests that, if anything, the company might be doing too good a job of motivating its employees.

Microsoft's approach to achieving this high level of motivation is simple. It focuses explicitly on morale. Each work group has a morale budget it can use for anything it wants. Some groups buy movie-theater style popcorn poppers, some go skiing or bowling or have a cookout, and some make T-shirts. Some groups rent a whole movie theater for a private screening of their favorite movie.

Microsoft also uses nonmonetary rewards extensively. I spent a year at Microsoft working on Windows 3.1. During that time, I received three team T-shirts, a team rugby shirt, a team beach towel, and a team mouse pad. I also took part in a team train ride, a nice dinner on the local "Dinner Train," and another dinner at a nice restaurant. If I had been an employee, I also would have received a few more shirts, a Microsoft watch, a plaque for participating in the project, and a big Lucite "Ship-It" award for shipping the product. This stuff's total value is probably only $300 or $400, but its psychological value is much greater.

Microsoft doesn't ignore developers' personal lives, either. During the time I was there, the developer whose office was next to mine had his 10-year-old daughter come by every day after school. She did her homework quietly in his office while he worked. No one at the company even raised an eyebrow.

In addition to providing explicit support for morale, Microsoft gladly trades other factors to keep morale high, sometimes in ways that would make other companies shudder. I've seen them trade methodological purity, programming discipline, control over the product specification, control over the schedule, management visibility—almost anything to benefit morale. Regardless of the other effects, the motivational efficacy of this approach speaks for itself.

Staff Seniority

Falling in line with Cocomo's emphasis on staff seniority, many leading organizations recognize the importance of senior staff. Many years ago, Microsoft's director of development pointed out to me that he had identified senior personnel as a critical success factor. He said that one of the keys to success of a product such as Microsoft Excel was to have at least two senior staff members continue over from the product's previous release.

In a study of runaway projects in the UK, managers identified "insufficient senior staff" as a contributing cause of difficulties in approximately 40% of the projects that significantly overran their schedules or budgets.[4]
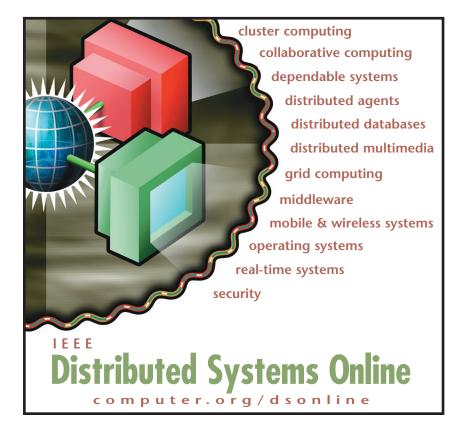
Even organizations that focus strongly on software processes recognize the important role of human factors. The NASA Software Engineering Laboratory was the first organization to win the IEEE Computer Society's award for software process achievement. In their "Recommended Approach to Software Development, Revision 3," one of their top nine recommendations is "Do start the project with a small, senior staff."[5]

The Bottom Line

It turns out that trading process sophistication for staff continuity, business domain experience, private offices, and other human-oriented factors is a sound economic tradeoff. Of course, the best organizations achieve high motivation and process sophistication at the same time,[6] and that is the key challenge for any leading software organization.

References

1. B. Clark, "Quantifying the Effect of Process Improvement," IEEE Software, Vol. 17, No. 6, Nov./Dec. 2000, pp. 65–70.
2. T. DeMarco and T. Lister, "Programmer Performance and the Effects of the Workplace," Proc. 8th Int'l Conf. Software Eng., ACM Press, New York, 1985, pp. 268–272.
3. B.W. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Cliffs, N.J., 1981.
4. A. Cole, "Runaway Projects—Cause and Effects," Software World, Vol. 26, No. 3, 1995, pp. 3–5.
5. "Recommended Approach to Software Development, Revision 3," Doc. No. SEL-81-305, NASA Goddard Space Flight Center, Greenbelt, Md., 1992.
6. S. Ahuja, "Laying the Groundwork for Success" (Interview), IEEE Software, Vol. 16, No. 6, Nov./Dec. 1999, pp. 72–75.

IEEE Software