

Doing Disciplined Work

Watts S. Humphrey



Over the years, I have often been asked about how to get management support for process improvement. Typically, engineers want to use better software methods; but they have found that either their management doesn't care about the methods they use or, worse yet, some managers even discourage them from trying to improve the way they work. In addressing this subject, I have decided to break it into two parts. The first part concerns disciplined work: what it is, and what it takes to do it. Then, in later columns, I will address the problems of getting management support for process improvement.

In discussing disciplined work, I answer five questions. First, what do we mean by disciplined work? Second, why is disciplined work important? Third, what are the elements of disciplined engineering work? Next, if you have the basic training and motivation to do disciplined work, why can't you just do it? And finally, what kind of support and assistance do you need to consistently do disciplined work?

What is disciplined work?

Discipline is an aspect of behavior. It involves consistently using sound methods. Discipline is defined as an activity or regimen that develops or improves skill; acting in accordance with known rules and proven guidelines. Disciplined behavior is generally needed whenever human error can cause harm, substantial inconvenience, or expense. The disciplined behaviors are then designed to reduce human errors, prevent common mistakes, and improve the consistency of the work. Also, many people are surprised to find that disciplined behavior generally improves efficiency, saves time, and even facilitates creativity.

Why is disciplined work important?

In every advanced field, you get better and more consistent results by using proper methods and applying known and proven techniques. This is true in factories, development organizations, and even research laboratories. No one would agree to an operation by a doctor who had not finished medical school, spent years as a resident, and been board certified. Similarly, when hiring an accountant, you would not consider someone who did not have a CPA certificate and was properly licensed. If you had to go to court, you could not use a lawyer who had not passed the bar and been qualified to practice in the state. Anyone doing biomedical or nuclear research knows that disciplined behavior can be a matter of life and death.

While the software field is too new to have qualification mechanisms like those in many other fields, we now know the practices required for good software engineering. In addition, we also know that when software engineers use these methods, they consistently produce quality products on their committed schedules and for the planned costs. Unfortunately, however, these methods are not yet generally taught in university curricula. Thus, to advance in this field, engineers need to get their own training and to develop their personal skills.

What are the elements of disciplined engineering work?

Disciplined software engineering involves more than just producing good technical results. As is true in other advanced fields, you need to address every aspect of the job. While technical competence is essential, you also need to consider the customer's needs, handle business concerns, and coordinate with your teammates. If, for example, you handle the technical concerns but ignore those related to the customer, you will likely produce a product that solves the wrong problem. While you might not lose your job, it is never good for an engineer's career to be associated with a failed project.

Proper attention to customer-related issues requires understanding the requirements before starting the design, maintaining close customer contact throughout the work, and planning and negotiating every change with the customer. Important business issues involve planning, tracking, and reporting on the work; focusing on quality from the beginning; and identifying and managing risks. Key teamwork issues relate to agreeing on goals, making and meeting commitments, and reviewing and supporting teammates' work. Finally, your team needs a logical development strategy, a sound architecture, a comprehensive design, and a set of rigorously followed standards and methods.

Getting the needed skills

There are various ways to obtain the skills needed for disciplined work. While some of these skills can be learned in university programs¹, many must be learned through on-the-

¹ Available PSP courses teach planning, process, measurement, and quality methods, using my text: *A Discipline for Software Engineering*. If you can't get into such a course, you could learn the basics from the PSP introductory text, *Introduction to the Personal Software Process*.

job experience.² Instruction in disciplined personal and team software methods can also be obtained from the SEI and its transition partners, but that requires your management's support. While qualified training is the route that I would suggest for anyone who can follow it, a principal concern of many engineers is that they can't get management support.

Why not just do it?

While getting training is important, consistently using the methods that you know is even more important. Unfortunately, it is also much more difficult. A growing number of engineers are being trained in disciplined engineering methods, but many of them find that even though they know how to do good software work, they are unable to practice what they know. The reason is that disciplined work is very hard to do, particularly when you try to do it by yourself. Here, there are two contributing elements: lack of personal discipline and inadequate coaching and support.

Personal discipline

How many times have you decided to do something but never really did it? Like New Year's resolutions, there are lots of things we know we should do like quitting smoking, not eating between meals, exercising every day, and many others. We kid ourselves that we could do these things if we really had to, but somehow we never do.

The problem here is that most of us try to maintain strict personal disciplines all by ourselves. For example, can you visualize working for years to become a concert-quality pianist in a deaf world? When the quality of your work is invisible and nobody knows or cares how you perform, it is almost impossible to follow rigorous personal disciplines. This is why professional athletes and performing artists have coaches, trainers, conductors, or directors.

Even at the pinnacle of their fields, professional performers need the help and support of a cheering section, the constant push and motivation of a coach, and the demanding guidance of an informed and caring trainer. This is not just a nicety; it is absolutely essential. We humans are a group species. We work best in groups, and we have great difficulty performing alone. We need somebody who knows and cares.

² I have written a textbook for a new teamworking course. It is called *Introduction to the Team Software Process*, and it teaches the basics of the TSP. This book will be available from Addison Wesley in late 1999.

Coaching and support

I was fortunate to be on a marvelous team when I was in college. Our coach had been on the U.S. Olympic wrestling team; and he was an energetic, enthusiastic, and terribly demanding coach. Nobody wanted to disappoint him. We all worked harder than any of us had ever worked before. In our first year, he took a team of rookies to the AAU championship of 13 states. What was most interesting to me was that the next year I transferred to a different school. The wrestling coach was a nice guy, but he was not demanding or enthusiastic. Not only didn't the team do well, I didn't either. Superior coaching makes an extraordinary difference, and it is necessary for any kind of disciplined personal work.

What kind of support do we need?

The issues that we face in software engineering are severalfold. First, our field has not yet developed a tradition of disciplined work. Thus we must change an industry-wide culture. Second, coaching is not a common management style. Managers in software, as in other fields, feel more natural acting like straw bosses. Few know how to use, build, and develop the skills of their people. But this is the essence of management: helping and guiding people to do the best work that they are capable of producing. When people don't perform as well as they should, managers should help them develop their skills and motivate them to rigorously use the methods they know.

In software engineering, good work requires engineering discipline, and disciplined work requires coaching. In a subsequent column, I will discuss getting managers to act like coaches.

Acknowledgements

In writing papers and columns, I make a practice of asking associates to review early drafts. For this column, I particularly appreciate the helpful suggestions of Bob Cannon, Bill Peterson, and Mark Paulk. I also thank Jean-Marc Heneman for his questions and comments on this topic.

In closing, an invitation to readers

In these columns, I discuss software issues and the impact of quality and process on engineers and their organizations. I am, however, most interested in addressing topics that you feel are important. So please drop me a note with your comments, questions, or suggestions (email: watts@sei.cmu.edu). I will read your notes and consider them when I plan future columns. Thanks for your attention and please stay tuned.

About the author

Watts S. Humphrey founded the Software Process Program at the SEI. He is a fellow of the institute and is a research scientist on its staff. From 1959 to 1986, he was associated with IBM Corporation, where he was director of programming quality and process. His publications include many technical papers and six books. His most recent books are: *Managing the Software Process* (1989), *A Discipline for Software Engineering* (1995), *Managing Technical People* (1996), and *Introduction to the Personal Software ProcessSM* (1997). He holds five U.S. patents. He is a member of the Association for Computing Machinery, a fellow of the Institute for Electrical and Electronics Engineers, and a past member of the Malcolm Baldrige National Quality Award Board of Examiners. He holds a BS in physics from the University of Chicago, an MS in physics from the Illinois Institute of Technology, and an MBA from the University of Chicago.

The views expressed in this article are the author's only and do not represent directly or imply any official position or view of the Software Engineering Institute or Carnegie Mellon University. This article is intended to stimulate further discussion about this topic.

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.

SM IDEAL, Interim Profile, Personal Software Process, PSP, SCE, Team Software Process, and TSP are service marks of Carnegie Mellon University.

® Capability Maturity Model, Capability Maturity Modeling, CERT Coordination Center, CERT, and CMM are registered in the U.S. Patent and Trademark Office.