

How to study *individual* programmers?

Maurizio Morisio

A.V. Williams Building
University of Maryland, College Park, MD 20742
voice: + 1 301 405 2740
fax: + 1 301 405 3691
email: morisio@cs.umd.edu, morisio@polito.it

Abstract

Software engineering is made by people, but empirical software engineering research often overlooks the effect of psychological characteristics. We report on a case study, the use of PSP in a company, where such a characterization would be useful to provide more insight in the results of the case study.

Keywords: PSP, Personal Software Process, psychology, empirical software engineering.

The problem

The Personal Software Process (or PSP) [5] is based on the hypothesis that the performance of individual programmers can be improved by applying sound techniques (such as reviews and effort estimation methods), within a defined process (plan, design, code, review, compile, test, post mortem). Further, performance should be measured constantly to verify improvements, if any. The measures used are effort spent in each phase of the process, size of the software developed, and defects.

Application of the PSP in training sessions [4] show improvements in estimation accuracy and quality (in the sense of defect density), while productivity does not change substantially. In this study individuals perform very differently, but no attempt is made to explain these differences. Little experience is available about the effects of the PSP in industry [3,6].

We recently run a case study dealing with the introduction and use of the PSP in an industrial setting [1,2,7].

The main findings are.

- The PSP cannot be used as-is, but should be adapted to match company practices (i.e. existing effort and defect collection, existing processes)
- Tool support for data collection is essential
- Privacy of data collected must be assured

From a researcher point of view, an important result of the case study was to understand the limitation of the research tools used. The PSP works at the level of individuals, while we were monitoring the case study with measures on software artifacts, such as defects and productivity. In other words we were overlooking the main factor in the experiment: people.

It is now clear that we should have used research tools from other disciplines, such as cognitive psychology, and psychology in general.

Questions

In the following we list the research questions that we were not able to treat. Some are not specific to the PSP study, but involve questions common to any experiment involving programmers and programming techniques.

- As from the title of Humphrey's book, PSP requires discipline. Methodical persons will be generally more prone to accept the PSP, while less organized persons will be less eager to adopt it. Is it possible to correlate the personality (as captured by Myers-Brigg tests and similar) and PSP acceptance? What personality characteristics are important?
- Apart the PSP, it is common experience that programmers program, and people approach problems, in different ways: top-down vs. bottom-up, organized vs. chaotic, by deadline vs. by priority, textually or graphically. How to characterize these ways of programming/thinking? Do these ways of programming impact external measures (such as productivity and defects) more or less than intelligence or experience? Are these ways of programming related with personality characteristics? Are some programming techniques (in)compatible, or indifferent, with personality characteristics? Is it possible to map programming roles (developer, analyst, tester) with personality characteristics?
- The company in which we run the case study observed, on previous technology transfer programs, that motivation of an individual in using a new technology is the real driver in its effectiveness, not the technology itself. In other words motivated users use the technology effectively, non motivated users use it with poor results. How to characterize motivation and its effect on the use of a technology?
- The PSP provides a process. Is it possible and useful to impose individual processes on individuals? Or individual processes should be defined by the individual?
- The PSP requires to log the effort spent in activities with minute level precision. Nobody in the case study accepted this level of precision, and used, at most, hours and half-hours. Is the perception of working time similar for everybody or not? What is the right level of precision, and 'right' in function of what factors?

References

- [1] Capra M., D. Escala, and M. Morisio, Adapting the PSP to an SME context, *Software Process Improvement Conference 1997*, Barcelona, December 1997.
- [2] Escala D., and M. Morisio, A metric suite for a team PSP, *Metrics98*, Bethesda, December 1998.
- [3] Ferguson, P., W.S. Humphrey, S. Khajenoori, S. Macke, and A. Matvya. Results of Applying the Personal Software Process, *IEEE Computer*, May 1997.

- [4] Hayes W., and J. Over, The Personal Software Process (PSP): An Empirical Study of the Impact of PSP on Individual Engineers, Technical report SEI-97-TR-001, December 1997.
- [5] Humphrey, W.S., *A discipline for Software Engineering*, Addison-Wesley, 1995.
- [6] Humphrey, W.S., Using a Defined and Measured Personal Software Process. *IEEE Software*, pages 77-88, May 1996.
- [7] Morisio M., Applying the PSP in Industry, *IEEE Software*, to appear, 2000.