Chapter 39
Implementation of the Personal Software Process in Academic Settings and Current Support Tools

Mohd Hairul Nizam Md Nasir
University of Malaya, Malaysia

Shukor Sanim Mohd Fauzi
Universiti Teknologi Mara, Malaysia

Nur Aalyaa Alias
Two Sigma Technologies, Malaysia

Mohd Hashim Massatu
Two Sigma Technologies, Malaysia

ABSTRACT

The Personal Software Process (PSP) is a structured software development framework that includes defined operations, measurements, and analysis techniques designed to help software engineers understand and build on their personal skills and improve their performance. The PSP provides a defined personal process, guides software engineers in collecting and recording data, and defines ways to analyse data and make process improvements. This chapter reviews the previous literature on the implementation of the PSP in academic settings. Lessons learned from PSP implementations are then highlighted. We found that there were mixed outcomes due to several issues that later become a barrier to the adoption of the PSP. Several tools have been developed to overcome these barriers and constitute a helping hand for both students and engineers in adopting the PSP.

BACKGROUND

The Personal Software Process (PSP) is a self-improvement process that helps software engineers develop high-quality and predictable software. The PSP is intended to control, manage, and improve the way engineers work. Additionally, the PSP helps to improve software quality during the development process because engineers can systematically track and plan their work accordingly and manage any defects they encounter in completing the project. These features ultimately facilitate the production of high-quality software and also help to assure product delivery within time constraints.

DOI: 10.4018/978-1-4666-4301-7.ch039

The PSP was developed by Watts Humphrey in April 1989. Despite the circumstances, he managed to develop a total of 62 programs and 15 distinct versions of PSP implementations in three years. Humphrey utilised approximately 25,000 lines of code (LOC) in the Pascal, Object Pascal and C++ programming languages during PSP development. Before Humphrey developed the PSP framework, he led the initial development of the Software Engineering Institute's Capability Maturity Model (CMM) and applied CMM to many software development projects. However, many questions remained, such as how to apply CMM principles to small organisations or to the work of a small software team, because the CMM principles focused more on assuring that the management system provided full support and assistance to the development engineers. Initially, Humphrey intended to apply CMM principles to write a small program. However, in a small group setting, Humphrey conceded that is not generally possible to have dedicated process specialists; hence, every engineer must participate at least part-time in process improvement.

Engineers can certainly benefit from a more detailed process, and it is thus required to deal overtly with actual practices in software development and to show engineers precisely how to apply the CMM process principles. As a result, the PSP was invented to convince software engineers that improvement requires change, and changing the behaviour of software engineers is a nontrivial problem. Humphrey also asserted that the PSP design is based on the principles of consensus planning and quality, providing an internal reference point for software quality before its public release.

One of the principles also espoused by Humphrey is that every engineer is different; therefore, engineers must plan their work based on their own personal data. Occasionally, an engineer may be *a doubting Thomas*; i.e., one who is sceptical about changes to their work habits. Although engineers may be willing to make a few minor changes, engineers generally hold fairly closely to what has worked for them in the past, unless they can be convinced that a new method is more effective. The second principle implied in the PSP is that engineers should use well-defined and measured processes to consistently improve their own performance. Altogether, engineers need to understand their personal performance, measures of the time they spend on each job step, defect injection and removal as well as the size of the product they produce. Subsequently, engineers are required to produce quality products and, to do this, they must plan, measure, and track product quality; the overall message is to focus on quality from the outset of the project. Additionally, engineers need to find and fix defects earlier during the process rather than later, as the latter tends to increase project expenses. In other words, prevention is better than cure; hence, defects are more effectively prevented and fixed earlier rather than later, when the project is nearly ready for release for public use. Equally important, the results of each job should be analysed for future reference, as any findings may be helpful in improving their personal processes, bearing in mind that the right way to complete the project is to find the fastest and cheapest way to do the job.

Structures and Processes

The first step in the PSP is planning, which consists of a planning script that is used to guide work and also a plan summary which records planning data. After gathering and analysing all the requirements for the project, engineers follow each step in the planning script and perform time recording and defect logging. In the planning phase, the overall plan is created with a gross estimate, wherein the engineer uses process scripts in defining each step required for each part of the process. Using this structure, engineers are able to quantify their work (Figure 1).

Initially, engineers should know about the work to be done; thus, they should plan their tasks based on their own historical time, size and defect data.