



PSP & TSP—The Necessary Approach



In the fall of 1996, approximately 20 software engineers from the Software Engineering Division at Hill Air Force Base were trained in the “Personal Software Process (PSPSM).” I was one of them. The division of approximately 500 scientists and engineers decided to try a pilot course to see if there was value in PSP training. The course consists of a very structured and disciplined approach for writing software. Ten computer programs and five reports are assigned. The objective is to track time, lines of code used, defects inserted and removed, and the process block where it occurred. PSP techniques reduce the number of defects inserted into the code by using process reviews at early stages of computer programming and closing the gap between estimates and actuals.

“It is better to eliminate the insertion of defects instead of using the compiler to catch problems in the code.” Sounds easy, right? This course is anything but easy. Among the 20 engineers in the course, most had undergraduate and master's degrees in electrical engineering or computer science. Most were leads of projects that required an enormous amount of time. Finally, most of these engineers were helping define the processes used and needed for the division's work into CMM Level 5. Those 20 agreed that it was the hardest course they had ever taken. The PSP course consumed all other responsibilities and time. Elizabeth Starrett's article, *PSP: Fair Warning*, on page 14 explains some of the difficulties in learning PSP.

After the grueling few weeks of the course, few were able to adopt the techniques taught in PSP and most did not. I was able to use many of the concepts in my Software Quality Assurance team. The PSP concepts were crucial for understanding CMM[®] Level 5 principles in helping the division. Those who were not able to use PSP in their projects had a harder time understanding true defect prevention and true teamwork.

Since the pilot, the division has trained many more engineers in PSP. The mission planning software project, TaskView, has brought PSP-trained engineers together into a special team, where the Team Software Process (TSPSM) is being used. In *Managing Risk with the Team Software Process* on page 7, David Webb, the Technical Program Manager of TaskView, explains how TSP is an effective method of managing software project risks by using a common-sense approach with nearly defect-free code.

Great teams, whether in sports or in business, share common commitments and goals. Teamwork is essential for most successful software engineering projects, as explained in Watts Humphrey's article, *Building Productive Teams*, on page 4.

Being associated with a CMM[®] Level 5 organization, I believe that PSP and TSP are essential for understanding true Level 5 concepts. If your organization is striving for CMM[®] Level 5, it should look into investing time, money, and effort into training key players in your organization. At the very least, SEPG and SQA members should be trained in PSP and TSP.

Lynn P. Silver
Associate Publisher

SM The Team Software Process and TSP are service marks of Carnegie Mellon University.

[®] The Capability and Maturity Model and CMM are Registered in the U.S. Patent and Trademark Office to Carnegie Mellon University.

SM The Personal Software Process and PSP are service marks of Carnegie Mellon University.