



Team Software Process Brings Project Success Over Time



While Personal Software ProcessSM (PSPSM) has proven benefits, it has nothing to offer if employees are not using it. The Ogden-Air Logistics Center faced this problem in 1998 when developing the TaskView software, which allows the user to view quickly an Air Tasking Order from low-level detail to high-level overview. More than 20 individuals in our software engineering division (MAS) had received PSP training, but six months after the class only one person was using it. As a result, MAS and TaskView became a test site for the Software Engineering Institute's (SEISM) Team Software ProcessSM (TSPSM), and Watts Humphrey, a SEI Fellow who led development of the TSP and PSP, worked directly with TaskView on this SEI pilot project.

Over the next four years, the team internalized the TSP/PSP concepts and delivered several remarkably successful releases of the TaskView product. As instructed by Humphrey, the TaskView team organized, used the scripts and templates, held weekly meetings, and updated its spreadsheets. Each team member also employed PSP processes, checklists, and spreadsheets personally. They also developed a *dashboard* application to collect semi-automatically PSP data. As a result, they reaped the TSP/PSP rewards: low defects, greater flexibility, more accurate estimates, team unity, quality focus, delivered on-time and within budget, and delighted the customer.

Of course, this wasn't an airtight success. There were misunderstandings in recording data, some PSP training was incomplete, PSP tailoring was needed, and other issues just like any project. In other words, TSP/PSP wasn't a quick fix. It was, however, the key to creating a very disciplined and self-guiding team. The success of the TaskView team led to not only the adoption of TSP in other mission planning and Command, Control, Communications and Intelligence projects in MAS, but helped guide many high maturity policy and process improvements in the division that are still in use today.

Randy B. Hill
Ogden Air Logistics Center, Co-Sponsor



TSP Has Multiple Uses



The Team Software ProcessSM (TSPSM) has proven successful in helping software projects and organizations adequately plan their efforts, track against those plans, and take action in a timely manner when things don't go according to plan. I'm not just parroting the information I have read in this month's articles: I used TSP for two years before coming to work for CROSSTALK. While I have seen its great advantages, I have also experienced its challenges. This month's issue contains articles that discuss both.

Alan S. Koch highlights the similarities between TSP practices and the practices advocated in the Capability Maturity Model[®] (CMM[®]) and CMM IntegrationSM (CMMI[®]), and how implementing TSP can actually facilitate your CMMI process improvement effort. Carol A. Grojean gives several examples of how TSP improved her development efforts and provides data to back her claims. Ray Trechter and Iraj Hirmanpour share some of the challenges that can come with TSP success. David Webb and David Tuma explain in their article that TSP itself does not cause success – the practices it contains do: TSP has a framework that guides developers to these practices.

TSP is also evolving to meet the needs of new applications and processes. Chris A. Ricketts' article discusses how his command adapted the published TSP process to meet the needs of its maintenance projects. In conclusion, Watts Humphrey presents an informative overview of why projects truly need TSP and insights into how it works. We received more good articles for this issue than we could possibly fit into it, so look for more TSP articles in later CROSSTALK issues.

Elizabeth Starrett
Associate Publisher