Using the TSP on the TaskView Project

David Webb, ***Ogden Air Logistics Center, Software Engineering Division***
Watts Humphrey, ***Software Engineering Institute***

This article reports the first results of using the Team Software Process (TSP)TM on a software-intensive system project. The TSP was developed by the Software Engineering Institute (SEI) to guide integrated teams in producing quality systems on their planned schedules and for their committed costs. The TaskView team at Hill Air Force Base, Utah used the TSP to deliver the product a month ahead of its originally committed date for nearly the planned costs. Because the engineers' productivity was 123 percent higher than on their prior project, they included substantially more function than originally committed. Testing was completed in one-eighth the normal time, and as of this writing, the customer has reported no acceptance test defects.

T**HIS ARTICLE DESCRIBES** the experiences of a team that used the TSP to produce a software-intensive product for the U.S. Air Force. The Ogden Air Logistics Center, Software Engineering Division, Hill Air Force Base, Utah, has a long history of producing avionics and support software for the Air Force. The division had previously been assessed at a Capability Maturity Model (CMM)® Level 3 and has just recently been assessed at CMM Level 5. TaskView, one of the products they delivered, is a system to help Air Force pilots produce flight plans. Flight planning is labor-intensive and time-consuming; TaskView automates much of this work. It helps mission planners produce accurate flight plans with less labor and in less time than previously possible. The project was completed ahead of its original schedule and within its committed budget. The product is currently in customer acceptance testing with no defects reported to date. This article is the first published report of project results with the TSP.

Following a brief TSP overview, we describe the software organization, the TaskView project, and the team's experiences in introducing and using the TSP. Next, we cover the engineers' reactions to using this process. We conclude with a brief summary of the key findings from the TaskView experience. The division already had a high-maturity software process, so it had data available from prior work. We can thus compare the performance of the TSP team to previous projects. Although this article presents some of the data, we only show a few of the indicators that are potentially available for TSP projects.

The TSP

Although the concepts and methods for running integrated teams are well known, the specific steps often are not obvious to working engineers and managers. For example, to be effective, teams need precise goals, clearly stated roles, a defined engineering process, and a detailed plan for the work. They need a framework for periodic coordination and structured methods to review and track project risks and issues. Team measures must be defined and recorded, tracking mechanisms developed, and a reporting system established.

Although none of these items is particularly complex or difficult, the specific actions often are not obvious. Before engineers can work effectively in an integrated team environment, they need to know precisely what to do. If they have not done such work before or do not have a detailed process to guide them, they will generally defer the new or unfamiliar items until they know how to handle them. They then do the tasks they fully understand. As a result, many of the actions required for effective teaming do not get done. Teams can waste a great deal of time trying to establish goals, resolving their working relationships, and figuring out how to do the work.

How the TSP Works

The TSP defines the steps required to build and run software-intensive integrated product development (IPD) teams [1]. First, the engineers are trained precisely how to do quality work, use a defined process, and make and use process measurements. For engineers to use these methods on the job, they must have hands-on training, explanation of the methods, and experience using them on realistic project-like exercises. This training is provided by an intensive 120-hour course that teaches the Personal Software Process (PSP)^{SM} [2,3,4,5]. Figure 1 shows how the PSP training and the TSP process provide the capabilities for integrated teamwork.

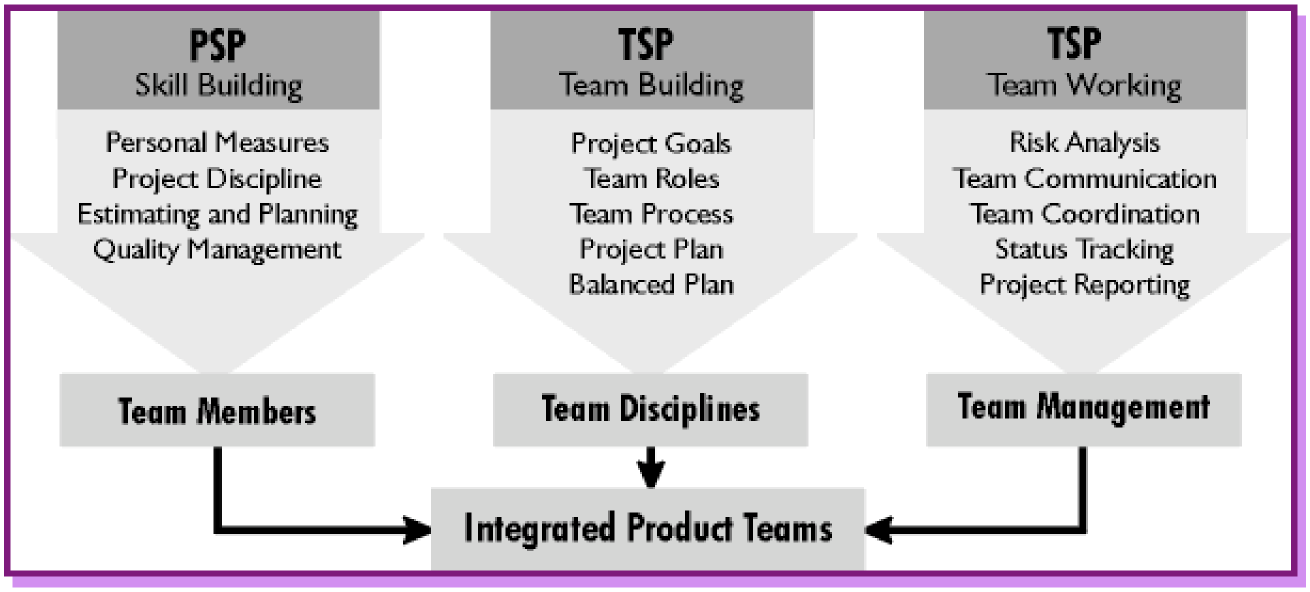


Figure 1. *How PSP and TSP provide IPD capabilities.*

After acquiring basic process, planning, and quality management skills, engineers have the prerequisites to use the TSP. Every project then starts with a three-day TSP launch workshop, where engineers develop teamworking practices, establish goals, select roles, define processes, and make plans. A shorter two-day relaunch workshop is then repeated at the start of every major project phase. Because team members work directly on their project during the launch, these three days are part of the job and are not a training exercise.

Finally, the TSP provides the mechanisms to maintain an effective teamworking environment. This is done with structured weekly team meetings and periodic relaunch workshops. The team meeting is much like the football huddle: all members participate, and they focus on precisely what to do next. If the plan is working, they follow it. If it is not, they may decide to change it. The team meeting not only maintains effective team communication but also facilitates precise status tracking, provides a context for team decision making, and supports continuous risk tracking and project reporting. As in football, periodic "huddles" are important; if teams did not huddle, they would do a lot of running around but not win many games.

The team relaunch is conducted at every principal project milestone. It serves to help the team evaluate and rebalance the project plan, reassess project risks, integrate new team members, reassign team roles, and re-emphasize the team's goals and charter. At the conclusion of each launch or relaunch, the team reviews its status, plans with management, and resolves any issues and problems.

What the TSP Provides

The TSP process provides a set of forms, scripts, and standards that lead the team through the process steps. Once they are PSP trained, engineers know how to develop and follow a defined process, and they understand how to use the process measures to consistently produce quality products. The PSP can be viewed as a language of process. Until engineers are reasonably fluent in this language, they generally are not able to follow the process and use its measures. PSP training provides the engineers the process fluency they need to use the TSP.

The TSP process also provides the guidance engineers need to work effectively in a team context. As shown in Figure 2, this is done during the three-day team launch. By following the launch process, the team members can quickly determine their own and everyone else's responsibilities, and they can readily track and coordinate their work with their teammates and other teams.

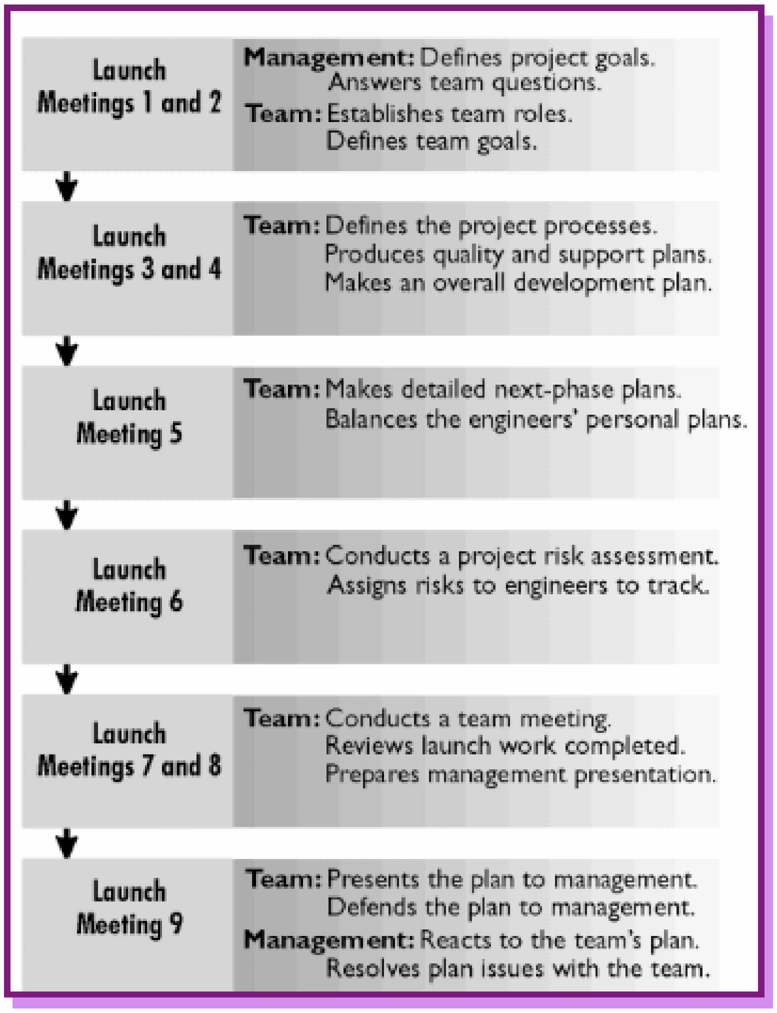

Launch Meetings 1 and 2
Management: Defines project goals. Answers team questions.
Team: Establishes team roles. Defines team goals.
Launch Meetings 3 and 4
Team: Defines the project processes. Produces quality and support plans. Makes an overall development plan.
Launch Meeting 5
Team: Makes detailed next-phase plans. Balances the engineers' personal plans.
Launch Meeting 6
Team: Conducts a project risk assessment. Assigns risks to engineers to track.
Launch Meetings 7 and 8
Team: Conducts a team meeting. Reviews launch work completed. Prepares management presentation.
Launch Meeting 9
Team: Presents the plan to management. Defends the plan to management.
Management: Reacts to the team's plan. Resolves plan issues with the team.

Figure 2. *The TSP launch process.*

Because the TSP produces a large volume of data, managing and tracking the data can become a burden. The SEI has developed a support tool that helps engineers record and track

TSP data. The initial tool support is in Microsoft Excel for Windows 95 and Windows NT. The TSP teams that have used this tool report that it substantially simplifies their data-gathering and reporting tasks. An enhanced tool is under development.

Engineering Support

During the launch and relaunch workshops, the team works as a unit to develop their process, quality, support, and project plans. These detailed plans identify and schedule the work for the next phase to the level of 10 task hours or fewer. Thus, the team members and their management know what tasks are to be done and when they are to be completed. In one example, Dave Webb, the TaskView team leader, needed to temporarily assign one engineer to help another project with a critical problem. By reviewing the detailed task schedule with the engineer, he precisely determined the impact of this reassignment and made workload adjustments to ensure that the project schedule was not affected.

The team as a unit also performs continuous risk management. In the launch and periodic relaunches, members do a complete project risk assessment. All risks are rated for likelihood and impact, and the more important risks are assigned to individual members for tracking. The assigned team members then develop mitigation plans for the immediate priority risks and monitor and report risk status in the weekly team meetings.

The TSP process helps working groups develop into cohesive and effective engineering teams. With defined and agreed-to goals and a process and plan to meet these goals, team members are more likely to submerge their personal problems and strive for the common objective. Efficiency is enhanced by the defined process, and communication is maintained by the weekly meetings of all team members. These meetings take less than one hour for teams of about 10 members. Team members review their role activities, planned vs. actual tasks completed, and risk status. Each member reports personal earned-value status, any needed team or management actions, and personal plans for the next period. These weekly meetings permit the team as a whole to periodically rebalance the workload, resolve issues, and make decisions.

TSP Status

The TSP process is being developed by the SEI, and it is currently under test by approximately 10 engineering groups and several dozen teams. Based on the experience to date, four TSP versions have been produced. The TSP has been used with teams as small as two engineers and with groups as large as 17. Some teams have been composed of software professionals, and others have also had hardware, systems, test, or other engineering participants. The project categories include maintenance, new product development, and product enhancement. System types have ranged from components of large commercial data-processing systems to embedded real-time controllers. TSP projects have covered proprietary product development, industrial software contracts, and military development and enhancement work.

Hill Air Force Base

The TaskView project was conducted by the Ogden Air Logistics Center, Technology and Industrial Support Directorate (TI), Software Engineering Division (TIS) at Hill Air Force Base, Utah. The TIS vision statement declares that they will provide "exceptional weapon system software and related hardware solutions and technology adoption expertise to enhance our nation's defense."

TIS is a high-maturity organization with a strong history of software process improvement. In March 1995, TIS was assessed as a CMM Level 3 organization, and the assessment conducted in July 1998 rated them at CMM Level 5. This is the first software organization in the Department of Defense (DoD) to receive this rating, and it is one of the few Level 5 software groups in the world.

The software products produced by TIS include operational flight programs for the F-16 Fighting Falcon aircraft, test program sets for F-16 automated test equipment, mission-planning software for a variety of aircraft, and avionics test-station software. TIS is also the home of the Software Technology Support Center (STSC), which provides technology adoption expertise to the DoD, sponsors the annual Software Technology Conference, and publishes *Crosstalk*.

During the summer of 1996, TIS introduced the PSP to a small group of software engineers. Although the training was generally well received, use of the PSP in TIS started to decline as soon as the classes were completed. Soon, none of the engineers who had been instructed in PSP techniques was using them on the job. When asked why, the reason was almost unanimous: "PSP is extremely rigorous, and if no one is asking for my data, it's easier to do it the old way."

Although the TIS Software Engineering Process Group (SEPG) believes that PSP training accelerated CMM improvement work, members were concerned that the PSP methods were not being used. They therefore asked the SEI how to get engineers to consistently use PSP practices on the job. Because the TSP was then being designed to address this exact problem, the SEI suggested that TIS become involved in TSP pilot testing. TIS decided to do so, and this project is the result.

The TaskView Project

TIS chose the TaskView project as the TSP pilot. TaskView is a UNIX-based tool that parses an Air Tasking Order (ATO), which is a set of battle instructions for all aircraft involved in a strike, including fighters, bombers, and refuelers. As shown in Figure 3, it describes the flight plans, aircraft armament, and specific mission roles and tasks. Once the battle has been planned, a complex set of computer programs generates an ASCII text file that contains the ATO information. This ATO is then delivered electronically to each of the units participating in the strike.

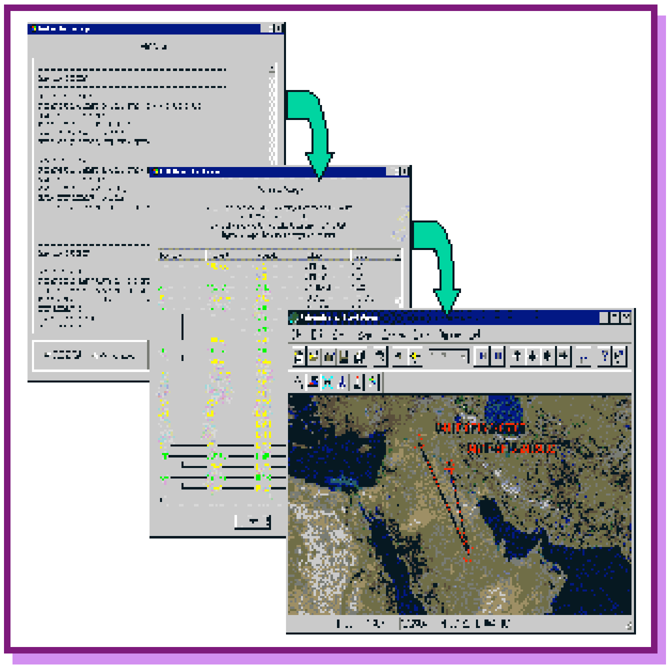


Figure 3. *TaskView converts complex ASCII text to tree structures to map routes.*

Currently, the ATO is "broken out" manually— interpreted, sorted, and restructured—by the participating groups, who use hard copies and highlighters to mark their specific instructions.

This is a laborious process that can take several hours. Once the information has been identified, the data must then be manually entered into mission-planning software tools for each unit, which provides ample opportunity for further mistakes. The TaskView tool parses the ATO and automatically "breaks out" (sorts and structures) the needed information in a few seconds. Additionally, TaskView can port data directly to mission-planning software tools, which greatly reduce the defects introduced during manual entry.

An initial prototype version of TaskView had been developed by another organization, and the TIS contract was to produce a product from this prototype, enhance it for a new ATO format, and port it from the UNIX environment to a PC Windows NT operating system.

TIS chose the TaskView project as a pilot for the TSP for several reasons:

- The team members were already PSP trained.
- TaskView was a small (under 20,000 lines of code [LOC]), short-duration (eight months) project from which results would be immediately apparent.
- The project manager for TaskView (Dave Webb) was an SEI-certified PSP instructor.

The TaskView project started a month before the introduction of TSP. The team had already been through the planning process required by TIS, and a detailed plan already existed before the first TSP launch. Since the TSP is designed to build on and augment an organization's existing process, the TaskView project could use the TIS Standard Engineering Process and tracking tools. When organizations do not have a fully defined process, the TSP launch process guides the team in defining and developing the needed process elements.

Using the TSP Process

The first TSP launch for the TaskView project was held at the end of February 1998. During the launch, we reviewed TSP concepts with the team and guided them through the project planning and tracking steps. The team spent about two and one-half days in this launch workshop.

Team Goals and Roles

During the project launch, the team members determined and documented the project goals. Some were high level, such as "delight our customers" and "be an effective pilot project for TSP in the Air Force and the DoD." More specific goals included "provide clean beta versions of TaskView to [the customer]" and "meet or exceed our quality plan." One important goal was to meet the customer's recent request that the TaskView project be delivered one month earlier than the original Sept. 30, 1998 commitment date.

Next, team members chose their personal team roles from among the TSP basic set: Customer Interface Manager, Design Manager, Implementation Manager, Planning Manager, Process Manager, Quality Manager, Support Manager, and Test Manager. Because of the limited size of the team, some members received more than one job. These roles were assigned so that when risks or issues arose, there would be a point of contact already designated and prepared to handle them. As usual, the official team leader had already been designated by management.

Detailed Planning

With the goals and roles determined, the team refined its existing project plan. The previously developed TaskView plan contained about three dozen work breakdown structure elements and tasks. During the TSP launch, the engineers produced a detailed list of more than 180 tasks. Using standard productivity rates, the team next estimated the task hours and the size of each task's product, usually in LOC. They also estimated each engineer's available task hours for each week of the project.

Task hours are hours spent working *only* on the tasks in the task list. Time spent in meetings, on the telephone, using E-mail, or engaged in any other activity that is not defined in the plan is not counted toward TSP task hours. Although these activities are necessary and are definitely work hours, they are not tracked as part of the project earned value. Based on the experiences of other TSP projects, the TaskView team estimated that in an engineer's standard 40-hour workweek, 20 hours would be an aggressive goal for task-related work.

The TSP Earned-Value Tool

TSP tools were then used to turn this top-down plan into an earned-value chart with a projected completion date. On the first run, the team and management were delighted to find that the new completion date projected by the top-down plan matched perfectly with the customer requirement for a one-month schedule acceleration.

Next, the software engineers were each given a copy of the task list and asked to estimate their personal work, using their own line of code and effort data. Such data are a product of the PSP course, which every engineer should complete before starting a TSP project. The TSP tool was then used to combine these individual estimates into a bottom-up estimate, also with earned value and a projected completion date. This estimate did not match the schedule requirements or the top-down estimate completed only a few hours earlier because some engineers were tasked more heavily than others. Because project schedules often slip if only one engineer is overburdened, the TSP launch process includes a workload-balancing step.

After workload balancing, the bottom-up schedule matched the top-down estimate and the customer's need. At this point, all engineers had a personal task and earned-value plan for which they individually had provided the estimates.

Risk Assessment and Mitigation

At the next TSP launch meeting, the TaskView team identified the risks associated with the project. They listed these risks in a brainstorming session, prioritized risk likelihood and impact, and assigned responsibility for mitigation and tracking. For example, the risk that "there will be a day-for-day slip in schedule if we do not receive the necessary header files by 3 March" was given a high likelihood and impact and assigned to the official team leader.

Fourteen risks were identified in this initial launch, of which seven were assigned to the team leader, and the balance were handled by team members. The team leader also agreed to share responsibility with the engineers to track and mitigate the other management-related risks.

Management Review

The final launch activity was a management review of the team's launch results. Normally, such meetings provide the forum to resolve serious scheduling or resource issues. For TaskView, however, the management review reaffirmed the existing project commitments.

Tracking the Work

After the two-and-one-half-day TSP launch, the team started on the job. Using the PSP, the engineers tracked, in minutes, the time they spent on each task and process phase, recorded the defects found at every phase, and measured the sizes of the products they produced. The data were stored in the engineers' data tracker and in the TSP tracking tools. Thereafter, the team met weekly to review earned-value status, goals, risks, issues, and action items.

Within the next few weeks, it was evident that the team had a problem. The engineers were not achieving the 20 task hours per week they had planned. Their earned-value data, however, showed them to be on or ahead of schedule. From the data, the team found that there were two offsetting factors: Tasks had generally been overestimated, and it was much harder to achieve 20 task hours per week than had been expected. Even though the schedule impact to date had been minimal, this new understanding helped the team make better plans, and it showed where to focus to improve performance.

The Team Relaunch

In May 1998, we guided the TaskView team in assessing their progress and conducting a relaunch. The relaunch was necessary because the project was moving into its second phase, and the engineers felt a new plan was needed. This new plan would reflect lessons learned from the prior phase, more realistically address task hours, and include new tasks.

Although relaunch workshops normally take two days, this team was able to accomplish it in only one day. During this period, they replanned the project, refined their size and time estimates, adjusted their schedule to reflect 15 weekly task hours per engineer, and reassessed risks. Based on the cost, schedule, risk, and quality data, the overall project was judged to be ahead of plan. Because tasks had been generally accomplished with less effort than originally planned, some functions were completed early, whereas one important function planned for Phase 1 had slipped to Phase 2.

Because of the project's progress, TaskView could either return some money to the customer or add new functionality. The customer interface manager worked with the customer and found that new functionality was more important than cost reduction. Management then agreed to add more tasks and more people to the project. These new functions caused a modest schedule delay, so the customer interface manager reviewed the new functionality and schedule with the customer for approval. Since the planned delivery was still months away, the customer decided to accept the small schedule change in order to get the added functions.

Project Results

To determine the benefits of the TSP, TIS compared the TaskView pilot with similar projects that followed the organization's standard process. The project manager and the software engineers were also asked how the TSP had helped or hindered their personal work. Because TIS projects already routinely meet schedules, commitment performance was not an important factor in the analysis.

Estimating Accuracy

Use of the TSP was found to substantially improve size and effort estimating accuracy. During the first launch, TaskView was estimated to be 14,065 LOC. By the second launch, with the new functions, the total estimated size grew to 19,105 LOC. When the TaskView project was completed, the final new and changed LOC for the project was 26,776, an underestimate of 40 percent. When the 9,455 LOC of added function were subtracted, the team's original 14,065 LOC size estimate had an error of 23 percent.

Table 1 shows the size estimates the engineers made during the second TSP launch. Module 7 took no new and changed code because the engineer reused an existing routine. Although some individual estimates were reasonably close, there was considerable variation. By using a sound statistically based method and their personal historical data, however, the engineers were able to make balanced estimates. This meant that, on average, they were as likely to estimate high as low. Because the errors in the individual estimates tended to compensate, the overall estimate was much more accurate than were the individual estimates. Team members believed that their large personal estimating errors were largely due to the lack of historical data for this kind of project. Future project estimates will benefit from the data gathered during this project and should be more accurate.

Module Number	Estimated New and Changed LOC	Actual New and Changed LOC	Percent Error*
1	1,500	1,656	10.40%
2	1,500	1,350	-10.00%
3	500	418	-16.40%
4	3,000	4,525	50.83%
5	1,000	973	-2.70%
6	500	1,067	113.40%
7	500	0	-100.00%
8	1,100	3,377	207.00%
9	1,500	848	-43.47%
10	500	956	91.20%
11	1,500	1,494	-0.40%
12	9	4	-55.56%
13	500	653	30.60%
14	unused	unused	unused
15	500	965	93.00%
16	1,177	2,973	152.59%
17	819	1,131	38.10%
18	3,000	4,386	46.20%
Total	19,105	26,776	40.15%

Table 1. *TaskView estimated vs. actual LOC. *Note that underestimates are positive, and overestimates are negative.*

The TaskView effort estimates were originally made before the introduction of the TSP. At the first launch, the effort was again estimated to determine if the costs were appropriate and if the load was properly balanced among the engineers. By the second launch, it was obvious that effort had been overestimated; the project was able to meet earned-value goals with fewer task hours than had originally been expected. After including the customer-requested new functionality, the final delivery date was only two days later than the accelerated schedule, and the cost error was negligible.

Productivity

The TIS software process database contains the average productivity in LOC per man-hour for this team's prior project, and the average productivity for every project that used the TIS organizational process. Although the exact numbers are proprietary, the TaskView project increased productivity to 16 percent above the TIS average. These particular engineers increased their productivity to 123 percent above their previous project, or more than two times. Data on the relative productivity in LOC per programmer-hour for TaskView, the team's prior project, and the average of all TIS projects are shown in Figure 4. The TIS average is shown as 100 and TaskView as 116.

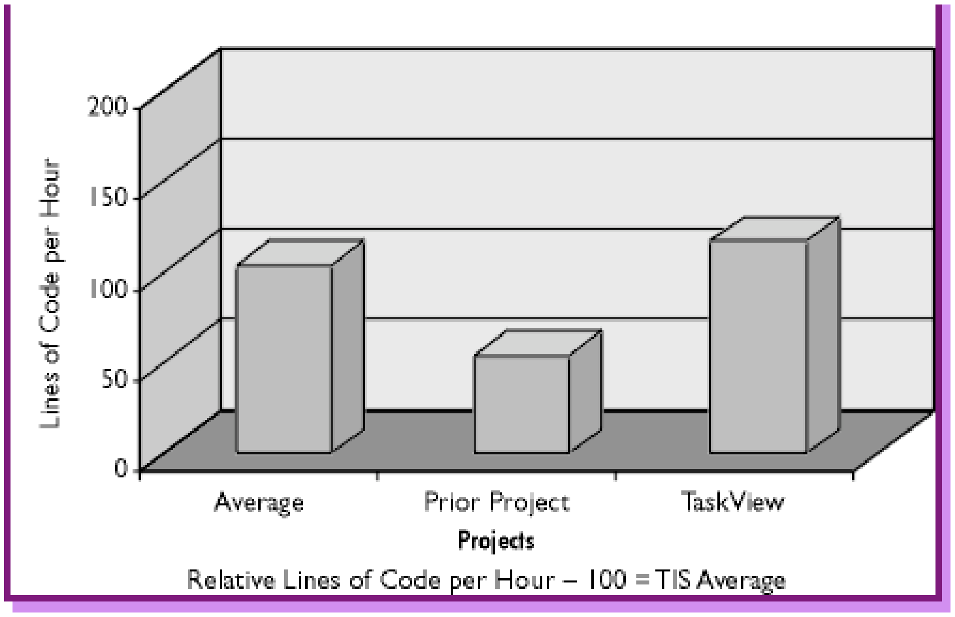


Figure 4. *Relative productivity.*

Productivity figures are impacted by many factors. Because TaskView and the team's prior project involved different languages, application domains, and development environments, the productivity improvement cannot be considered a measure of the TSP. The results do, however, suggest that the TSP improves productivity.

Quality Improvement

As shown in Table 2, the standard TIS process includes inspections (peer reviews) of all work products. The TSP adds a set of personal design and code reviews. One important question was whether the time spent doing these personal reviews was worthwhile. The TIS process typically removes about 13 defects for every thousand lines of code (KLOC) during design and code inspections. The rest must be found in test or by the user. With TSP, the TaskView project increased the yield of early defect removal by more than 60 percent by removing 21 defects per KLOC in both the reviews and the inspections. The benefits of this early attention to quality are apparent from the results of the later test phases.

Phase	**TIS**	**TSP**
Requirements Inspection	X	X
High-level design inspection	X	X
Detailed design personal review		X
Detailed design inspection	X	X
Personal code review		X
Compile	X	X
Code inspection	X	X
Functional test	X	X
Candidate evaluation (CPT&E)	X	X
System test (ERT)	X	X
Operational test and evaluation (acceptance	X	X
test)	X	X
Operational usage (external)		

Table 2. *TIS and TSP defect-removal process steps.*

Assuming the engineering process has rigorous testing criteria, an indicator of product and process quality is the time spent running tests. Generally, the fewer defects there are to be found, the less time is spent in test and the higher is the resulting product quality. The TIS process has three test phases, all with rigorous criteria, that must be completed before the product is passed to an external agency for operational testing: functional test, candidate evaluation, and system test. These phases are then followed by the customer's operational test and evaluation and then by operational usage. Typical TIS projects require 22 percent of the project schedule (in days) to perform the final two TIS test phases. The TaskView project, using TSP, sharply reduced this percentage to 2.7 percent. This is a schedule savings of nearly 20 percent. Only one high-priority defect was found in these last two test phases.

Data from the completed TaskView project show that the defect density at the functional testing phase was close to that normally achieved by other TIS projects only after all engineering testing phases have been completed. In TaskView, to find only one high-priority defect in the TaskView product during system and operational testing is unprecedented for any TIS project.

Because of the improved quality from the TSP process, TaskView testing time was sharply reduced, as shown in Table 3. Here, the test data for the TaskView project are compared with three comparable prior projects. Although one could reduce testing time by running incomplete tests, the fact that the customer has so far reported no defects during acceptance test suggests that this was not the case. By using the TSP, TaskView not only produced a higher-quality product, it also took only one-eighth the testing time normally required for similar projects.

	TaskView	**Project 1**	**Project 2**	**Project 3**
Program Size - LOC	26,776			
CPT&E Test Days	4	67,291	7,955	86,543
ERT (System Test) Days	2	22	10	33
Total Test Days	6	41	13	59
Test Days/KLOC	0.22	63	23	92
System Test Defects/KLOC	0.52	0.94	2.89	1.06
Acceptance Test	0*	2.21	4.78	2.66
Defects/KLOC		N/A	1.89	0.07

Table 3. *TaskView testing time. *Acceptance test is continuing but no defects have been reported to date.*

Qualitative Results

A critical question in introducing any new software engineering tool or technology is whether the engineers will use it. If the engineers do not like a tool or method, they will probably not use it, regardless of its effectiveness. To assess this issue, we privately asked all the TSP team members four questions:

- What do you believe are the advantages of the TSP?
- What do you believe are the disadvantages of the TSP?
- What about the TSP would you change?
- What about TSP would you keep the same?

Without knowing their teammates' responses, every team member said the TSP helped them form a closer, more effective team than any they had worked on before and that they would like to continue to use it. One team member said, "The TSP creates more group involvement.

Everyone feels like they're more part of a group instead of a cog in a wheel. It forces team coordination to talk about and solve problems—there's no pigeonholing." Another team member said, "This really feels like a tight team. I was on the same team for a year [while working on another project] and didn't know the team members as well as I do now."

Another qualitative advantage expressed by multiple team members was increased effectiveness in project planning and tracking. "TSP gives you better insight into your current state," said one software engineer. "It provides better focus for the software developer on tasks to be done." Another TaskView team member summed up the planning and tracking benefits of TSP in this way: "Measuring progress helps generate progress."

The principal weakness the TaskView team mentioned was the need for better TSP tool support. Several members said that the tracking and earned-value support needed to be improved, and another suggested more automated data gathering and analysis. Work on TSP tool improvement has already begun at the SEI, and a newer, better version of the planning and tracking tool will soon be available.

The lead software engineer gave perhaps the best testimonial to the qualitative results of the TSP. When asked what he would not change about the TSP, he said, "I've seen a lot of benefits [from the TSP]. I'd like to see us continue to use it."

Conclusions

One of the fears many have about process improvement initiatives like the TSP is that the cost of doing extensive planning, personal reviews, and data gathering will increase the overall cost of the project. It is evident from the TaskView data, however, that the time spent performing these activities is more than made up by improved planning accuracy and reduced test time. As Philip Crosby once noted, "Quality is free." [6]

Perhaps the greatest change with the TSP is in the relationship between management and the engineers. To be most effective, engineers must be motivated and energetic; they need to be creative and concerned about the quality of their products, and they should enjoy their work and be personally committed to its success. This can only be achieved if management trusts the engineers to work effectively and the engineers trust their management to guide and support them.

Although trust is an essential element of effective teamwork, it must rest on more than mere faith. The engineers must follow appropriate methods and consistently strive for quality results. They must report on their progress and rapidly expose risks and problems. Similarly, management must recognize that the engineers generally know more about their detailed work than the managers, and they must rationally debate cost and schedule issues. Management also needs to ensure that the engineers consistently follow disciplined methods and that the teams do not develop interpersonal problems.

The TSP is designed to address these issues and show engineers and managers how to establish an environment in which effective teamwork is normal and natural. Because this will often require substantial attitude changes for the engineers and the managers, to introduce the TSP is a non-trivial step. As the TaskView data show, however, the TSP can produce extraordinary results.

Acknowledgments

Being a leader and a coach for the TaskView team has been a rewarding experience for each of us. It would not, however, have been as rewarding or satisfying without a dedicated and hard-working team. For their support and cooperation, we thank Pattie Adkins, Keith Gregersen, Neil Hilton, Craig Jeske, Ken Raisor, Mark Riter, and Capt. David Tuma. We also enjoyed excellent support from Tresa Butler for configuration management, Pat Cosgriff for SEPG support, and Jim Van Buren of the STSC for PSP consultation.

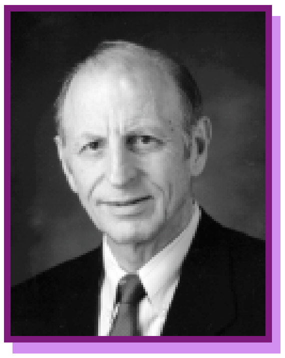
For quality engineering work, consistent and informed management leadership is essential. For their trust in us and their willingness to support us in pioneering the early use of TSP in practice, we thank Dan Wynn, Robert Deru, Don Thomas, LaMar Nybo, and Eldon Jensen. Lt. Col. Jacob Thorn, the TaskView program manager at Eglin Air Force Base, Fla., also supported our process improvement initiatives. His dedication to quality and informed oversight made the job possible.

We also thank those who reviewed this article. Their comments and suggestions were a great help. Our particular thanks to Rushby Craig, Walter Donohoo, Linda Gates, John Goodenough, and Bill Peterson. Finally, the professional help and guidance of the *Crosstalk* staff have, as always, been a great help.

About the Authors

David Webb has a bachelor's degree in electrical and computer engineering from Brigham Young University. He has worked for TIS for more than 11 years as a software engineer. Six of those years he spent as an F-16 Operational Flight Program software test engineer and system design engineer, three years as a member of the TIS SEPG, and two years as a technical program manager for TIS mission-planning software. He has participated in three CMM-Based Appraisals for Internal Process Improvement, including TIS's 1998 Level 5 assessment. He has also been certified by the SEI as a PSP course instructor.

OO-ALC/TISHD
6137 Wardleigh Road
Hill Air Force Base, UT 84056
Voice: 801-775-2916 DSN 775-2916
E-mail: webbda@software.hill.af.mil



Watts S. Humphrey is a fellow at the SEI at Carnegie Mellon University, which he joined in 1986. At the SEI, he established the Process Program, led initial development of the CMM, introduced the concepts of Software Process Assessment and Software Capability Evaluation, and most recently, the PSP and TSP. Prior to joining the SEI, he spent 27 years with IBM in various technical executive positions, including management of all IBM commercial software development and director of programming quality and process. He has a master's degree in physics from the Illinois Institute of Technology and in business administration from the University of Chicago. He is the 1993 recipient of the American Institute of Aeronautics and Astronautics Software Engineering Award and an honorary doctorate in software engineering from Embry Riddle Aeronautical University in 1998. His most recent books include *Managing the Software Process* (1989), *A Discipline for Software Engineering* (1995), *Managing Technical People* (1996), and *Introduction to the Personal Software Process* (1997).

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Voice: 412-268-6379
E-mail: watts@sei.cmu.edu

References

1. Humphrey, Watts S., "Three Dimensions of Process Improvement, Part III: The Team Process," *Crosstalk*, Software Technology Support Center, Hill Air Force Base, Utah, April 1998, pp. 14-17.
2. Ferguson, Pat, Watts S. Humphrey, Soheil Khajenoori, Susan Macke, and Annette Matvya, "Introducing the Personal Software Process: Three Industry Case Studies," *IEEE Computer*, May 1997, pp. 24-31.
3. Humphrey, Watts S., *A Discipline for Software Engineering,* Reading, Mass., Addison-Wesley, 1995.
4. Humphrey, Watts S., "Using a Defined and Measured Personal Software Process," *IEEE Software*, May 1996.
5. Humphrey, Watts S., "Three Dimensions of Process Improvement, Part II: The Personal Process," *Crosstalk*, Software Technology Support Center, Hill Air Force Base, Utah, March 1998, pp. 13-15.
6. Crosby, Philip B., *Quality Is Free: The Art of Making Quality Certain*, McGraw-Hill, New York, 1979.

The SEI's work is supported by the Department of Defense.

From the upper right corner, clockwise: Kevin Tjoland (TISFD), David Haakenson (TISFB), Ken Raisor (TISHD [TaskView TSP Project]), Mark Peterson (TISFD), David Webb (TISHD [TaskView TSP Project]).