

Personal Earned Value: Why Projects Using the Team Software Process Consistently Meet Schedule Commitments

David Tuma
Software Process Dashboard Initiative

David R. Webb
Hill Air Force Base

Data from dozens of projects using the Team Software ProcessSM (TSPSM) provide powerful proof of success at consistently meeting cost and schedule commitments. While disciplined engineering and high quality processes are important factors contributing to these successes, mathematical analyses of project data indicate that the most important factor is the proper management of earned value techniques at the team member level. In fact, this practice – unique to TSP teams – can produce a 10-times reduction in schedule variance by properly balancing team workload using personal data.

Projects using the Team Software ProcessSM (TSPSM) developed by the Software Engineering Institute (SEISM) have a phenomenal performance record, especially in terms of meeting schedule estimates. As noted by Watts Humphrey in this issue of CROSSTALK, current industry data show that more than one-third of all non-TSP software projects still fail [1]. In stark contrast, data gathered by the SEI from 20 TSP projects in 13 different organizations show that these TSP teams missed their schedules by an average of only 6 percent and had a very narrow schedule variance range, from 20 percent earlier than planned to 27 percent later than planned [2].

Why do projects using the TSP succeed at meeting schedule commitments so often and so well? Conventional wisdom suggests this world-class performance is due to two reasons: (1) TSP software engineers have become experts at using historical data to produce highly accurate estimates; and (2) TSP projects employ quality methods that drastically reduce or even eliminate defects found in later process phases (such as integration, system, and acceptance testing), rendering these typically volatile development activities consistent and predictable.

Years of real-world TSP project experience suggest that TSP's approach to earned value planning and tracking is also a significant factor in meeting schedule estimates. In fact, while the factors listed above are of great importance, our analysis indicates that the management of earned value at the team member level is more important than both these factors combined. To understand why, it is helpful to compare traditional earned value project management to the approach used in the TSP.

Traditional Earned Value Planning and Tracking

Many projects use a method called *earned*

value to plan and track progress. At the beginning of a project, teams using earned value will define a list of high-level project tasks, and estimate the time each task will require. As shown in Figure 1, a predicted completion date for each task can be estimated by determining when the project will have expended the requisite effort or Budgeted Cost of Work Scheduled (BCWS).

The earned value method then assigns each project task a *value* based upon its estimated cost or effort. As each task is actually completed by team members, the project *earns* the originally estimated value for the task; this is called the Budgeted Cost of Work Performed (BCWP). The real cost or effort to complete the task is also tracked as the Actual Cost of Work Performed (ACWP). The combination of these three values, arranged according to planned (BCWS) and actual (BCWP and ACWP) schedule performance, allows projects to determine both cost and schedule variances from their plan.

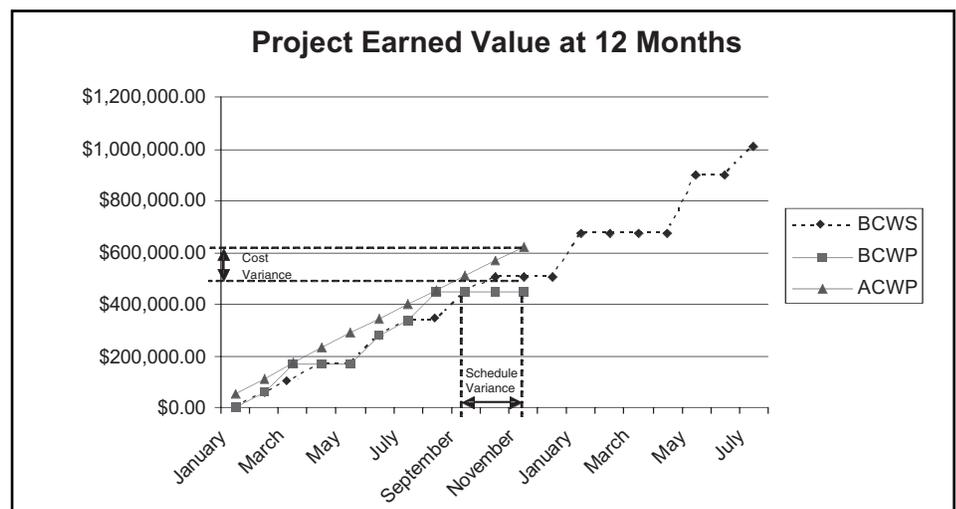
This traditional earned value technique, while an effective tool, is often incomplete because it is planned for the completion of high-level project tasks

(such as overall design, code, and test), is tracked only at the project level, and is reviewed only monthly. Since these large tasks often require more than one month to complete, there is a high likelihood of one or more *zero work* or *flat line* zones on a traditional earned value plan. (Note the BCWS for October to December and January to April in Figure 1.) Within these zero work zones, earned value metrics provide no insight into project progress; this can mask serious problems for months at a time. In Figure 1, a serious scheduling problem that was first encountered by the project in January does not show up on the earned value chart until May.

TSP Earned Value Planning and Tracking

TSP teams create and use earned value plans very differently from traditional teams. At the beginning of a TSP project, the team conducts a *launch* meeting. During the initial launch, tasks are defined at a very high level and estimated using gross measurements such as lines of code per hour. A rough plan is drawn up using these high-level estimates to determine an

Figure 1: Traditional Earned Value Plan



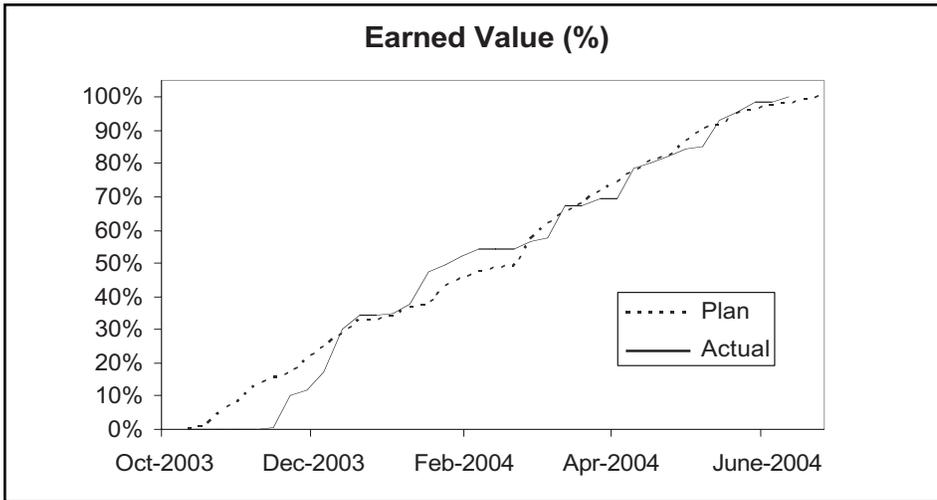


Figure 2: TSP Earned Value Plan

idea of the schedule the project will require to complete the assigned workload. Once this is complete, the TSP team subdivides the work into three-to-four month phases, breaking the first phase into detailed subtasks of fewer than 10 hours each. These tasks are assigned to individuals on the team, and personal earned value plans are created for each team member. These personal plans are then consolidated to create a team plan.

Once the launch is finished, the individuals immediately begin working to complete their assigned tasks, tracking their progress using their personal earned value plans. One of the SEI's stated entry criteria to launching a TSP project is that team members are trained in the Personal Software ProcessSM (PSPSM). Among other things, PSP students learn how to estimate in pieces, break down their personal work into measurable tasks, and gather minute-by-minute data on their progress to create detailed earned value plans. As a result, TSP teams have continuous access to real, measurable data on task completion, duration, and cost (effort). So, rather than the stair-step, month-to-month plan shown in Figure 1, TSP teams produce and live by much more detailed earned value plans like the one shown in Figure 2. The smoother look to this plan is due to a much higher granularity of measurement than is practiced or even possible on traditional

earned value projects. Note that there are basically no zero work zones in Figure 2, even during the weeks of Christmas and Independence Day!

Using this level of detail, the team holds weekly meetings where they review progress against the personal and team earned value plans. These weekly reviews include an examination of the forecast completion date; if the forecast differs significantly from the plan, the team produces corrective action plans to address the variance. Since individual team member data is available to supplement the rolled-up team measures, it is immediately obvious to TSP teams which tasks and team members are ahead of schedule and which need assistance. This information gives the team members the insight, on a weekly basis, to adjust task assignments, renegotiate functionality with the customer, or replan work to keep the project on track.

The Significance of Personal Earned Value

This individual earned value tracking methodology provides a very sound basis for planning and managing a team software project. Three fundamental behaviors are the key to this management approach:

1. Fine-Grained Estimation (subdividing project tasks before estimating).

2. Forecast Tracking (monitoring forecast cost and completion date).
3. Workload Balancing (reassigning workload between team members).

Although these seem to be fairly common-sense behaviors, they are radically affected by earned value metrics tracking at the personal level. At first glance, the first bullet would appear to be the most important of the three practices. Most TSP practitioners would be surprised to discover that the bullets are actually listed in *increasing order of importance*. Workload balancing, in fact, has the *most* significant impact by far on the reduction of project schedule variances. To understand why, it is helpful to examine these behaviors in light of a few simple and well-understood statistical phenomena.

Estimating Basics¹

Estimates, of course, are never perfect, and estimating errors are inevitable. The quality of a series of estimates can be characterized by two metrics: precision and accuracy. Estimating precision is the concept most people think of first. For example, an estimate that falls within 5 percent of the final value could be described as very precise. Most organizations have a strong business need to minimize cost and schedule overruns and overestimates; consequently, they focus on reducing the size of their estimating error.

Estimating accuracy, on the other hand, describes the bias in a series of estimates. If an organization were to consistently underestimate project cost, their estimates would not be considered very accurate. An even balance between overestimates and underestimates would characterize an accurate estimating process.

When estimating a large project, it is common to begin by breaking the work down into smaller tasks, estimating those tasks independently, and summing the results. Outlining tasks in greater detail can generally produce a more precise final estimate. Although this practice intuitively seems to be correct, statistical concepts explain this mechanism mathematically. Imagine, for example, you have a sequence of independent estimates for individual subtasks, like those in Table 1.

In Table 1, subtask 1 is estimated (with 70 percent certainty) to require between 75 and 125 hours, with 100 hours being the most likely cost. The individual task estimates would be summed to produce a total estimate of 550 hours. The ranges, however, cannot simply be summed (which would produce a range of ±125 hours). If these estimates are accurate (balanced between underestimates

Table 1: Example Subtask Estimates and Ranges

Task	Estimate (Hrs)	Range (70%)
Subtask 1	100	±25
Subtask 2	160	±30
Subtask 3	90	±20
Subtask 4	200	±50
Total	550	±67 (not summed)

and overestimates), it would be very unlikely for the actual project to complete every subtask at either the low or the high end. As a result, it can be assumed that over- and under-estimates will partially cancel each other out. Statistically, the estimated range for the overall project can be calculated by squaring each range, summing those values, then taking the square root. This approach yields a prediction range of ± 67 hours and makes the range around the sum of the estimates considerably tighter than the range around the estimate of each individual task.

This important concept is the basis for many industrial-strength estimating practices (including the cost estimation practices in the TSP). Although its application to cost estimation is well known, its implications for schedule estimation are much more profound (as will be explained).

Fine-Grained Estimating Defines 10-Hour Subtasks

When planning work for the next three- to four-month project iteration, TSP project teams create a plan that divides project work into subtasks of approximately 10 hours each. This behavior can be quickly understood as an example of the *estimating precision technique* described in the previous section. In practice, however, TSP teams rarely generate *independent* estimates for each subtask, which was a basic assumption for the sum-squares range calculation. Instead, larger tasks are estimated, and historical percentages are used to automatically subdivide those tasks into smaller parts. As a result, the individual estimates are not independent, and do not fully benefit from the statistical mechanisms described.

As mentioned, TSP teams require software engineers to be trained in the PSP. While it is true the PSP teaches engineers well-defined, statistically based methods for producing accurate estimates, TSP teams rarely use those methods to produce team plans. The PSP PROxy Based Estimating (PROBE) method requires abundant historical data at the personal level; teams rarely have access to that kind of data when they first launch. Even after archiving considerable data, teams generally use historical averages based on team-level metrics to produce their plans. Although the co-authors have collectively participated in more than a dozen TSP launches (including projects listed in the SEI studies cited earlier), we have actually never been part of a TSP launch that used PROBE methods for team planning purposes.

Task	Planned Hours	Actual Hours	Estimating Error %
Subtask 1	4	0.8	80%
Subtask 2	3	1.7	43%
Subtask 3	1.2	4.6	-283%
Subtask 4	10	4.3	57%
Subtask 5	5	7	-40%
Subtask 6	5	14.5	-190%

Note: Estimating Error % is calculated as (Plan-Actual)/Plan

Table 2: Example Task Data From Hill Air Force Base TSP Project

Furthermore, the need to produce such detailed estimates early in the project, with limited available estimating time, typically results in estimating errors that are much larger than those measured by engineers during the PSP training course. Consider the excerpt of data in Table 2 from a recent TSP project at Hill Air Force Base.

These subtasks, chosen at random, demonstrate the *significant* estimating errors that occur when work must be broken down to the 10-hour level during an initial project launch. The histogram in Figure 3 shows the estimating errors for all subtasks completed by the project during a 12-month period.

As Figure 3 indicates, estimating errors at the subtask level are large and widespread. More than two-thirds of the subtasks in this project were misestimated by 50 percent or more. This metrics trend is not unique to this project. As a result, the fine-grained estimating performed during a TSP launch rarely enables teams to see the cost estimating precision benefits that would be projected by a sum-squares range calculation, or by the estimating accuracy improvements described in PSP studies.

Without question, many TSP teams are able to finish projects with very small cost variances. These achievements, however, are not generally accomplished with the statistically precise estimating methods taught in the PSP course. Instead, TSP teams are able to manage cost variances with mid-course corrections, enabled by the forecast tracking behaviors described in the next section.

In fact, the project whose data is illus-

trated in Table 2 and Figure 3 was *highly* successful, completing with a cost variance of 17 percent (under planned cost) and schedule variance of only 2 percent (ahead of schedule). These phenomenal results were explained by a team member, who said, “Our project succeeded [on cost and within schedule] because *we made it succeed*.” The subtask data indicate that these results were not due to precise, fine-grained task estimates. Instead, it was accomplished by diligent forecast tracking and workload balancing.

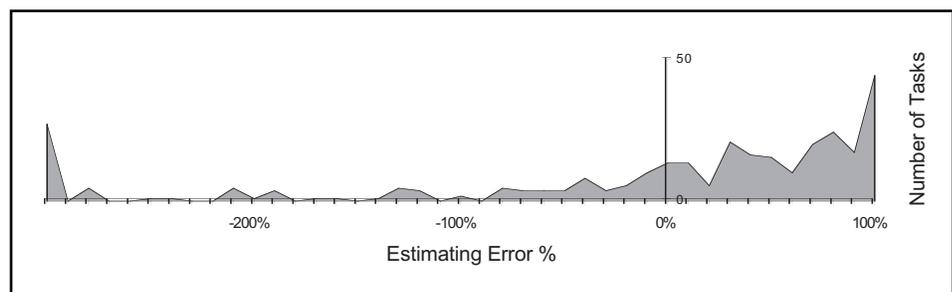
Fine-grained estimating, then, does not carry the full statistical significance suggested by the conventional wisdom. It does, however, provide an important tangible benefit: Defining tasks at the 10-hour level allows earned value progress to be tracked weekly. This helps the team to maintain a focus on continual progress, and facilitates early detection of problems.

Forecast Tracking Reveals Biases Early

Early detection of problems is the primary goal of forecast tracking. As described earlier, TSP teams collect earned value metrics daily and review them weekly, and produce corrective action plans when forecast cost and forecast completion dates differ significantly from the baseline. If these corrective action plans are unsuccessful, the team will quickly escalate the issue to management and to the customer.

Collecting earned value metrics at the personal level significantly increases the granularity of the resulting metrics, enabling TSP teams to discover cost and

Figure 3: Histogram of Task Estimating Errors From Hill Air Force Base TSP Project



Scenario	Team Member A	Team Member B	Unbalanced	Optimized
1	Early	Early	Early	Early
2	Early	Late	Late	On time
3	Late	Early	Late	On time
4	Late	Late	Late	Late

Table 3: Four Workload Balancing Scenarios

schedule discrepancies much earlier than usual. Early knowledge of these discrepancies allows the team to renegotiate scope and/or alter their technical direction, which can facilitate a significant reduction in final cost variances.

Although teams will strive for accuracy in their estimating process, significant estimating biases are still quite common. Forecast tracking provides a way for teams to discover these biases early when there is still time for the project to recover.

Workload Balancing

Workload balancing is the act of reassigning tasks from overburdened team members to under-tasked team members. The goal of workload balancing is to produce a plan in which all team members finish their assigned work on approximately the same date. Workload balancing is uniquely enabled by earned value tracking at the personal level.

Of the earned value management behaviors described, workload balancing is by far the most important. To understand why, it is helpful to consider the difference between two metrics:

- **Unoptimized Forecast Completion Date.** The date the project is forecast to complete, if progress continues at historical rates, and if team members perform tasks as assigned in the current project plan.
- **Optimized Forecast Completion**

Date. The date the project is forecast to complete, if progress continues at historical rates, and if tasks are reassigned to balance the workload optimally.

With earned value schedules for each individual on a team, it is simple to calculate these two metrics for the overall team. The optimized date can be calculated simply by summing up data values to the team level, and using traditional earned value equations². The unoptimized date can be calculated by looking at the personal schedules and seeing who finishes last.

Examining a very simple case can illustrate how these forecasts differ. Consider a team with only two individuals, and consider the various scenarios (shown in Table 3) where the individuals finish 20 percent early or 20 percent late.

Scenarios 1 and 4 show the presence of a consistent estimating bias. Workload balancing does not help in these scenarios, but fortunately these problems can be detected and corrected via the forecast tracking activity described in the previous section. Scenarios 2 and 3 show the simple effect of workload balancing; in these scenarios, the projects would finish late, but workload balancing helps them finish on time instead.

Table 3 makes clear a very simple observation: Workload balancing allows schedule variances to additively cancel each other out. This is an incredibly

important point because it allows project schedule variances² to benefit from the sum-squares reduction described earlier.

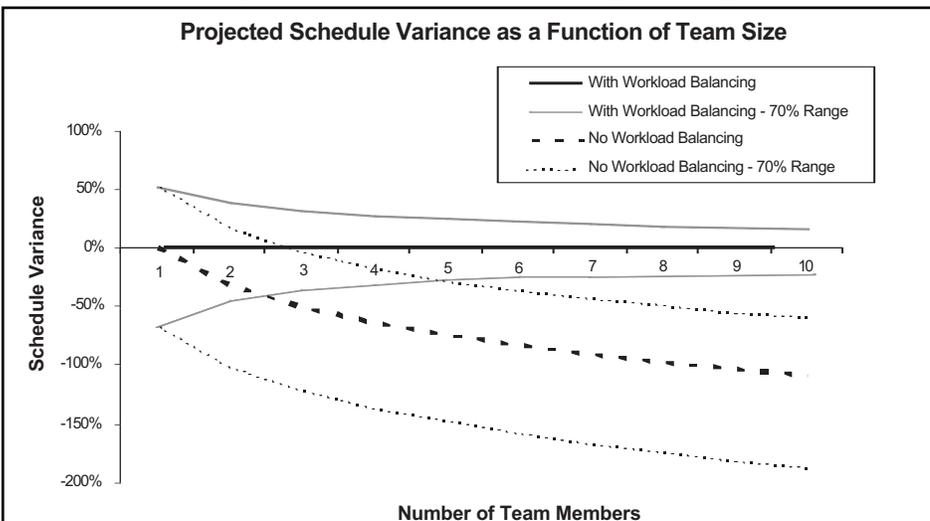
It is also possible to make this observation mathematically. Consider a project team with two individuals. Both individuals estimate, with 70 percent certainty, that they will complete the work assigned to them by the end of September. If the workload is not balanced, what is the likelihood that the overall project will finish by the end of September?

Since the workload is not reassigned, we can observe that the project will complete when the last person finishes. Since there is a 70 percent probability that Team Member A will finish by the end of September and a 70 percent probability that Team Member B will finish by the end of September, a simple calculation ($0.70 \times 0.70 = 0.49$) indicates that the overall probability is only 49 percent. This probability drops exponentially as individuals are added to the team: with eight team members, the projected likelihood of project completion by September 30 drops to less than 6 percent. This dismal percentage is due to the fact that a problem encountered by any individual can affect the project's completion date.

Of course, a confidence level of 6 percent is not useful when reporting forecast completion dates to management or to the customer; 70 percent prediction ranges are more in line with expectations. To estimate the project completion date with a team of eight people with 70 percent certainty, you need the 95 percent ranges for each individual! This graphically illustrates why most projects don't finish on time.

In fairness, the optimized and unoptimized metrics here are extremes. Even with the best workload balancing, a TSP team will never be able to perfectly optimize their plan; nevertheless, they are often able to come *very* close. And even on a non-TSP team, some workload balancing is likely to occur. But tracking earned value at the personal level has an undeniably significant impact on the effectiveness of workload balancing. By applying forecast tracking to the earned value plans of each individual, teams are able to notice imbalances early and reassign tasks that have not been started yet. In contrast, most non-TSP teams do not discover imbalances until late in the project. This awareness often comes too late to meet the originally committed completion dates, and the need to transfer knowledge from the overcommitted individual to other team members catastrophically impairs productivity [3]. For the

Figure 4: Simulation-Projected Schedule Variance as a Function of Team Size



Note: The schedule variances shown are based on a cost estimating error of ± 50% and a weekly task time estimating error of ± 25% (both for 70% certainty ranges).

entire period of the project before the imbalance is noticed, individual team members will have been pacing themselves, consciously or unconsciously. An under-tasked team member, seeing that they are comfortably meeting the dates required of them, will have most likely devoted a significant amount of time to non-project-essential tasks, unaware that their co-worker needed help. That time spent on non-essential work can never be recovered. In contrast, continual workload balancing helps to establish a shared level of urgency among team members.

Analysis With Numerical Methods

These simple mathematical analyses illustrate how unoptimized forecast dates become exponentially less reliable as individuals are added to a project. But when a workload is balanced based on personal earned value metrics, schedule overruns and underruns are able to cancel each other out, resulting in significantly smaller schedule variances for the overall project.

Using numerical methods, the authors of this article have succeeded in demonstrating this fact mathematically [4]. The results were striking: All other factors being equal, workload balancing predicted schedule variances that were orders of magnitude smaller than the schedule variances for unbalanced work. Figure 4 illustrates the results: With no workload balancing, a project is more and more likely to finish behind schedule as team size grows. In contrast, a project that balances workload optimally has more opportunities for workload balancing as team size grows, increasing the likelihood that the project will finish on time.

This analysis seems to suggest that workload balancing, enabled by personal earned value tracking as practiced in the TSP, can by itself account for a 90 percent reduction in the schedule variance of a project. These incredible results suggest that personal earned value tracking is *predominately* responsible for the tiny schedule variances seen by TSP projects.

Conclusions

The TSP includes many high-maturity behaviors that help teams produce superior results. While nearly all of these behaviors affect a team's on-time schedule performance, numerical analysis seems to indicate that proper application of earned value at the personal level is the largest single factor enabling the tiny schedule variances seen by TSP teams.

Curiously, engineers only receive

about an hour of earned value training in the PSP class, and they do not typically use these techniques during the course. Most engineers do not actually experience the practical application of earned value until they start their first TSP project (and historically there has not been any extra earned value training at that point)³.

These facts seem to beg these questions: "Could personal earned value tracking be used alone, without other PSP/TSP techniques, by teams in otherwise mature organizations?" "If so, what are the critical enabling success factors?" "What results might be expected?"

A six-sigma design of experiments seems warranted. While one would not expect to see the quality successes produced by TSP projects, the authors of this article feel that targeted earned value training and proper management support may allow non-TSP teams to enjoy significant cost and schedule benefits. ♦

References

1. Humphrey, Watts S. "Why Big Software Projects Fail: The 12 Key Questions." CROSSTALK Mar. 2005

<www.stsc.af.mil/crosstalk/2005/03/0503Humphrey.html>.

2. Davis, Noopur, and Julia Mullaney. The Team Software ProcessSM (TSPSM) in Practice: A Summary of Recent Results. Pittsburgh, PA: Software Engineering Institute, Sept. 2003.
3. Brooks, Frederick P. The Mythical Man-Month. 1st ed. Reading, MA: Addison-Wesley, 1995.
4. Tuma, David A. "A Statistical Model for Personal Earned Value, and Implications for Project Planning." 31 Dec. 2004. <<http://processdash.sourceforge.net/ev.html>>.

Notes

1. This article does not attempt to fully explain statistical estimating methods. It only describes these methods at a high level as background for the following discussion.
2. Variance is used in the project management sense, not the statistical sense.
3. TSP teams could potentially benefit from additional earned value training, to take full advantage of the powerful tools they have at their disposal.

About the Authors



David R. Webb is a senior technical program manager for the Software Division of Hill Air Force Base in Utah, a Capability Maturity Model®

for Software Level 5 software organization. He is a project management and process improvement specialist with 17 years of technical, program management, and process improvement experience with Air Force software. Webb is a Software Engineering Institute-authorized instructor of the Personal Software Process, a Team Software Process launch coach, and he has worked as an Air Force section chief, Software Engineering Process Group member, systems software engineer and test engineer. He is a frequent contributor to CROSSTALK and has a bachelor's degree in electrical and computer engineering from Brigham Young University.

OO-ALC/MASM
7278 Fourth ST
Hill AFB, UT 84056
Phone: (801) 777-9737
Fax: (801) 775-3023
E-mail: david.webb@hill.af.mil



David Tuma is the lead developer for the Software Process Dashboard Initiative, creating open source tools to support high-maturity software development processes. He first encountered open source software as a student at the Massachusetts Institute of Technology, and again later as a captain in the United States Air Force. As a strong supporter of open source, Tuma has been developing open source software on his own time for the past 10 years.

Software Process Dashboard Initiative
1645 E HWY 193 STE 102
Layton, UT 84040-8525
Fax: (801) 728-0595
E-mail: tuma@users.sourceforge.net