All the Right Behavior

David R. Webb
Software Division, Hill Air Force Base

Software projects using the Team Software Process^{SM} (TSP^{SM}) have an unusually high rate of on-time completion. One of several key factors contributing to this accomplishment is the effective way TSP teams make use of earned value techniques to iteratively refine their plan as they work it. Because earned value is reviewed weekly, and because no value is earned either at the personal or the team levels until a task is fully completed, software engineers are highly motivated to perform good earned value practices. This article expounds upon this principle and examines how TSP teams succeed with earned value.

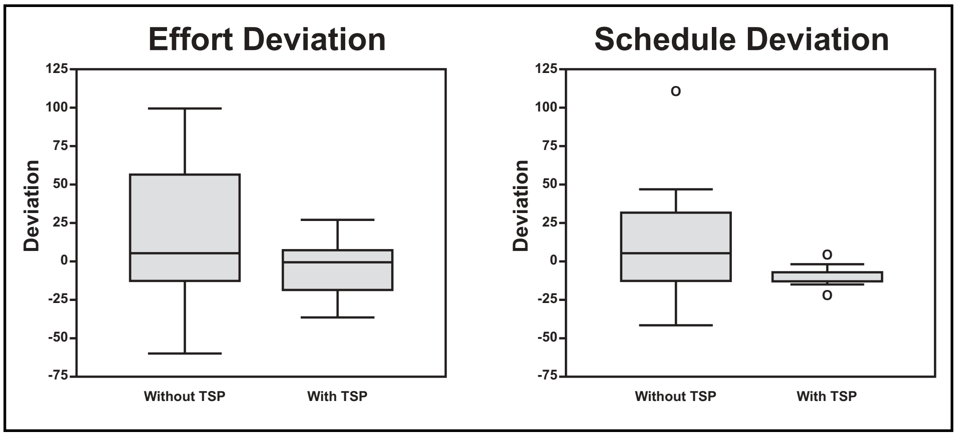
In a world where software projects can typically expect a 100 percent schedule slip, projects using the Team Software Process^{SM} (TSP^{SM}) have an unusually high rate of on-time completion (Figure 1). In fact, I have been on three TSP teams that have experienced tremendous success in meeting or exceeding schedule. One of several key factors contributing to this accomplishment is the effective way TSP teams make use of earned value techniques to iteratively refine their plan as they work it.

Unlike many teams using traditional earned value methods, TSP teams understand what their data mean, they trust their data, and they actually use their data to guide them in a way that most projects cannot. They succeed for the following five reasons:

1. TSP earned value is based upon properly decomposed tasks.
2. TSP earned value is measured at the personal level.
3. TSP earned value is based on true task completion.
4. TSP earned value is defined in terms of task hours, not dollars.
5. TSP teams review their earned value data and update their plans each week.

A well-known TSP coach recently summed this up when he said, "TSP earned value drives all the right behavior." This article will expound upon this simple, but profound statement, examining how TSP teams succeed with earned value, and why this approach does indeed drive all the right behavior.

Figure 1: *Comparison of Effort and Schedule Deviation with and without the TSP*



Breaking It Down

Earned value is simply a way of measuring progress. The following is a very simple (and unrealistic) example: If a project had 10 major tasks and each task was estimated to take 10 days to complete, the project would have a 100-day schedule and each task could be assigned a *value* equal to 10 percent of the whole (Table 1). As each task is completed, that value is *earned* by the project.

> **"Unlike many teams using traditional earned value methods, TSP teams understand what their data mean, they trust their data, and they actually use their data to guide them in a way that most projects cannot."**

Traditional earned value throws another curve at the project and equates each task to a dollar value. In the 100-day example, if the total project costs were estimated at $1,000, each task would have a value of – you guessed it – $100. This is called the Budgeted Cost of Work Scheduled (BCWS).

Assuming a month has 20 working days, you can quickly estimate that two tasks should be completed each month, and the entire project should take five months (Figure 2). Once you have that baseline estimate, you can begin to add actual data to the chart as each task is completed, displaying the Budgeted Cost of Work Performed (BCWP) and the Actual Cost of Work Performed (ACWP) (Figure 3). (If you are getting confused at the proliferation of acronyms, do not worry that your IQ level has dropped. This is a common problem with traditional earned value. Later, I will show how TSP makes this easier, or at least reduces the number of acronyms you will need to know.)

As stated earlier, this example is far too simple for the real world. Let us look at a more realistic situation. Figure 4 (see page 14) details the earned value progression of a fictional project called *Project Genesis* after one year of work. This is a software intensive project and has been using classic earned value since its inception. Project Genesis is a firm believer in true *all-or-nothing* earned value and does not put any value on the earned value chart until a task is completed. The project was originally scheduled to take 18 months to complete at a cost of just over $1 million. A few earned value calculations would tell you that the project, as depicted in Figure 4 is right on schedule and a bit over budget, which you can tell by looking at the chart.

Now, look at Figure 5 (see page 14). This is the same project four months later. By earned value definition, the project is exactly on schedule and, though somewhat over budget, is pretty much in line with what you would expect. Looking at this chart as a manager or customer, you may be tempted to say, "Well, they were behind for a few months, but now they've caught up."

Right here we have nailed one of the major problems with the misuse of traditional earned value, because that statement is dead wrong. In fact, Project Genesis is at least three months behind schedule and is in serious trouble. It will deliver months late at a very high cost. If you find this statement confusing, you are not alone. Many customers have told me they feel like they are being duped by doubletalk when they see earned value charts, because they know by experience that despite good-looking charts, projects often fail to meet their schedules. Let us go over a few reasons why Project Genesis's good-looking charts are unintentionally hiding the truth.

(Right now, any earned value gurus reading this are hopping up and down and shouting at the page. That is because there *are* earned value calculations – such as Estimate at Completion – that can tell you whether or not Project Genesis is truly behind schedule, even if the chart looks good. Unfortunately, you have to *be* an earned value guru to know this; most managers and customers are not gurus.)

One reason that our fictional Project Genesis has incorrectly determined schedule performance is that the team did not properly break down its tasks. An important rule of earned value is that you must plan to see progress *each time* you report. If you do not, you really cannot tell when you are getting behind.

Look at Figure 5 again. This project was in trouble way back in October, but did not feel it until January because the slope of the planned earned value line was zero. The result of this lack of proper planning is that they had no feedback on their progress for several months.

If Project Genesis had been using the TSP, it would have avoided this pitfall entirely. TSP teams do not (or should not) allow long stretches of zero slope on their planned earned value charts. They can break the tasks down into very small increments, usually less than one week in duration, so small that they can be measured much more frequently than a month at a time. In the next few paragraphs, I will explain how this works, and why it is so beneficial.

Personal Earned Value

One prerequisite a team must meet prior to beginning the TSP is that all software developers on the team must be trained in the Personal Software Process^{SM} (PSP^{SM}). There are numerous articles and books on the PSP [1], its tenets, and its numerous benefits. I will only mention here that PSP trainees learn how to plan and track their work at a personal level. PSP-trained software engineers know how to estimate in pieces, break their personal work down into measurable tasks, and gather minute-by-minute data on their progress (Table 2, see page 14). For people who have not had PSP training, this may seem like a ludicrous activity at the individual level. Those who have tried it out, though, have found it is really only a matter of personal engineering discipline and takes no more time than software development performed using the traditional ad hoc approach.

	Estimated Days	Value	Dollar Value
Task 1	10	10%	$100
Task 2	10	10%	$100
Task 3	10	10%	$100
Task 4	10	10%	$100
Task 5	10	10%	$100
Task 6	10	10%	$100
Task 7	10	10%	$100
Task 8	10	10%	$100
Task 9	10	10%	$100
Task 10	10	10%	$100
Totals	**100**	**100%**	**$1,000**

Table 1: *Simple Earned Value Breakdown*

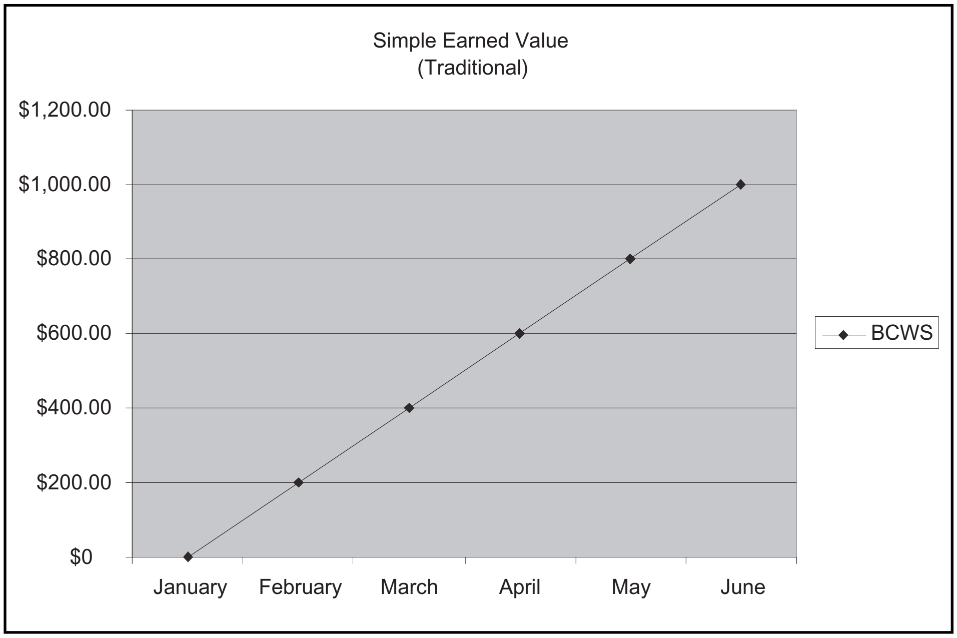


Figure 2: *Simple Earned Value Example – Estimates*

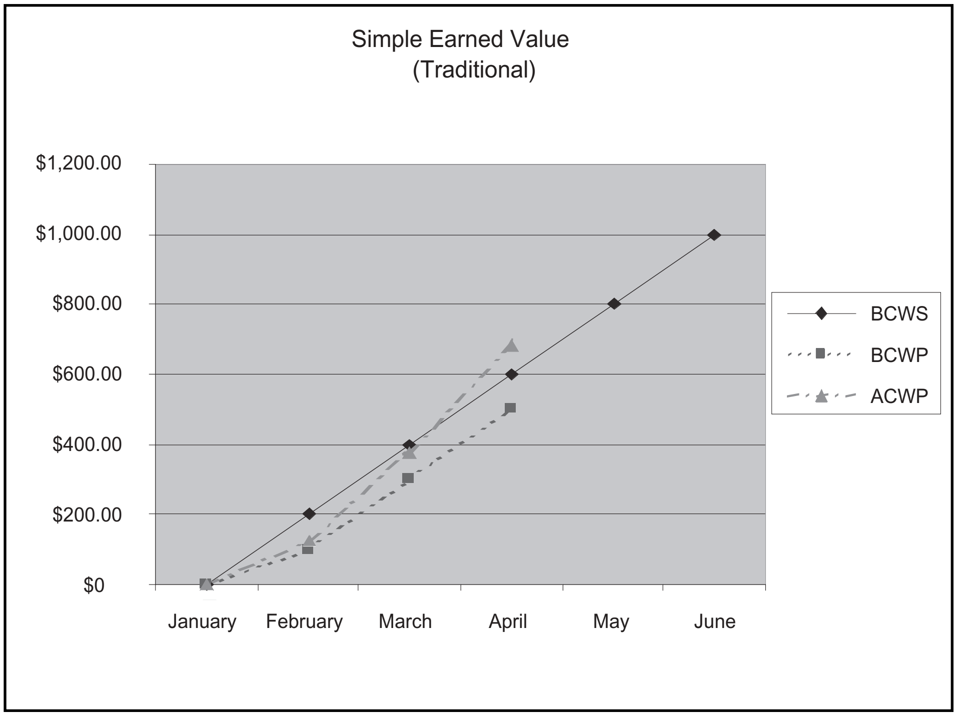


Figure 3: *Simple Earned Value Example – Actuals*

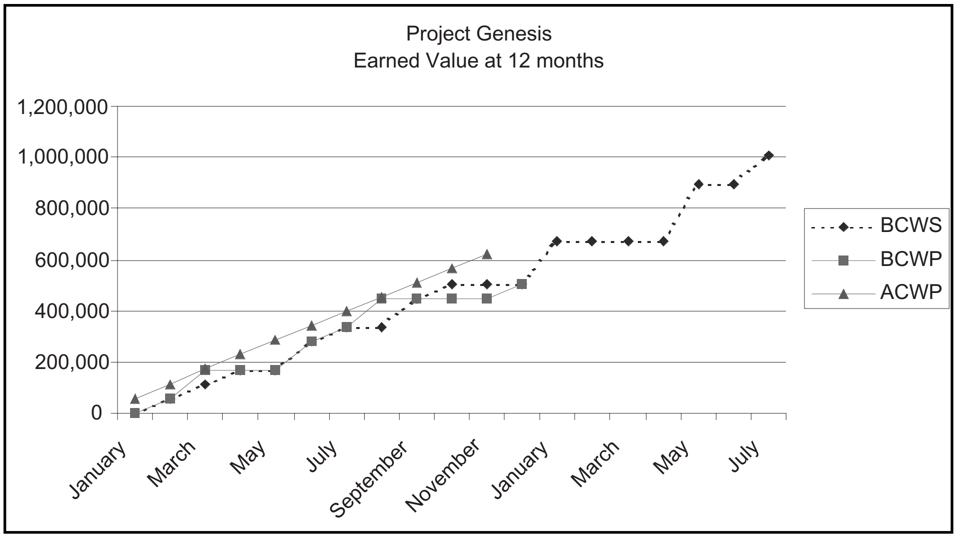


Figure 4: *Project Genesis Chart, Number One*

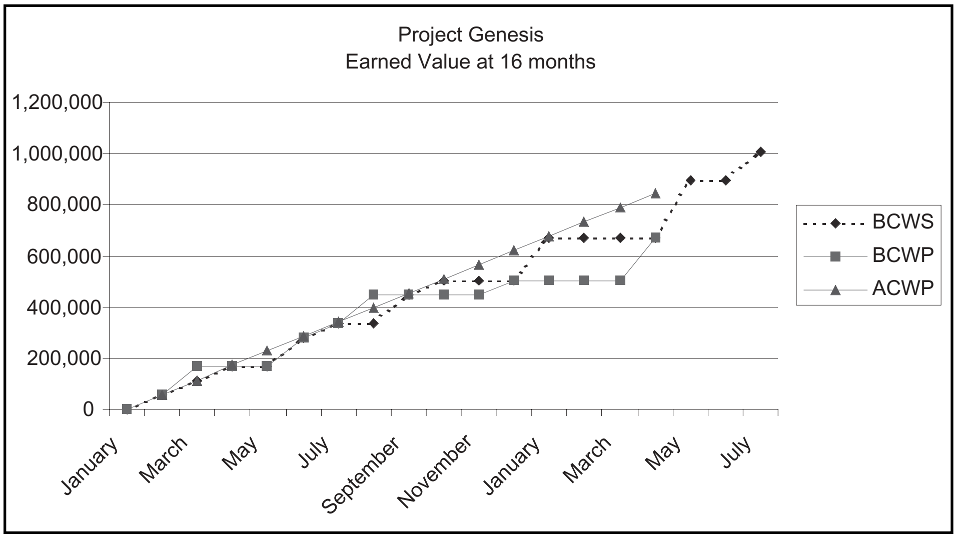
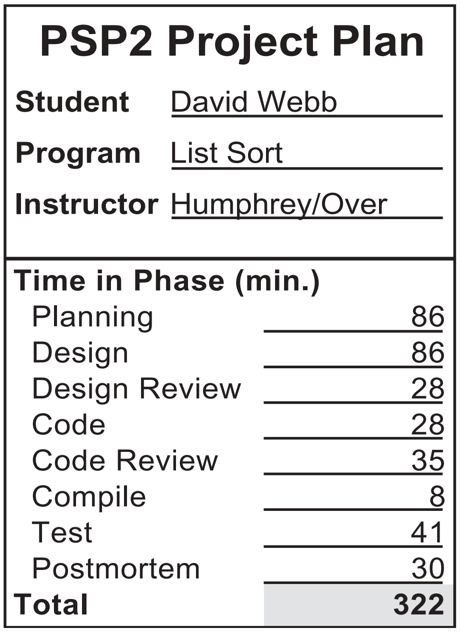


Figure 5: *Project Genesis Chart, Number Two*



PSP2 Project Plan

Student	David Webb
Program	List Sort
Instructor	Humphrey/Over

Time in Phase (min.)	
Planning	86
Design	86
Design Review	28
Code	28
Code Review	35
Compile	8
Test	41
Postmortem	30
Total	**322**

Table 2: *Portions of a PSP Planning Worksheet*

One of the many practical applications of this data-centric discipline is that an individual's tasks and estimates, broken down to a very fine granularity, are readily available to any team using the TSP. In addition, because of personal data gathering, TSP teams have real, measurable data on task completion, an absolutely essential element of tracking true earned value.

In other words, TSP team members can break tasks down and gather very accurate data on each task. This personal approach is what makes the type of earned value used by TSP teams possible.

Time Is Money, or Is It?

Another hallmark of TSP earned value is its lack of coupling between dollars spent and task completion. The reason for this ties back to the personal tracking methodology. TSP team members track *task time* in addition to *project time*. Task time is defined as the actual amount of time (in minutes) spent performing the specific tasks identified as key to project completion, minus any interruptions, which can consist of such mundane things as telephone calls, e-mails, meals, and trips to the restroom. However, it is important to note that any time spent on activities not typically identified on task lists also count as interrupts. This means meetings, work-related discussions, equipment setup, and even travel to other rooms or buildings are not counted as task time.

In a typical week, TSP engineers may track fewer than 20 hours of total task time. Of course, this is very different than the 40 hours of project time they have tracked since all of those *non-task items* must be done to support any project. As a result, the actual cost of a project in dollars is far different from the task hours earned. Rather than being a drawback, however, this personal tracking approach filters out unnecessary data and cleans up the earned value information. TSP team members know exactly how much time (not money) they are spending on the tasks that matter and exactly when those tasks are completed. This kind of precision leads to clearly understood earned value charts without reference to a budgeted or actual cost of anything (Figure 6).

Iterative Refinement

Finally, what sets TSP earned value apart from all other approaches – and frankly makes it work – is the frequency at which the data are reviewed[1]. Each TSP project begins with a *launch* (Figure 7). During the initial launch, tasks are defined at a very high level and estimated using gross measurements such as historical productivity (the number of lines of code a team typically can produce per hour). Using these high-level estimates, the team produces a detailed earned value plan.

This is typically the point when most software teams stop – that is as detailed as their plan becomes. The TSP team, however, then determines what the *next phase* of the project will be and uses PSP techniques to break that next phase into very detailed tasks of fewer than 10 task hours each. Using this level of detail, the team reviews progress against this earned value plan weekly. That is right, once a week, *not* once a month. To paraphrase TSP developer Watts S. Humphrey, projects do not slip a month at a time, they slip a day at a time, an hour at a time, and even a minute at a time.

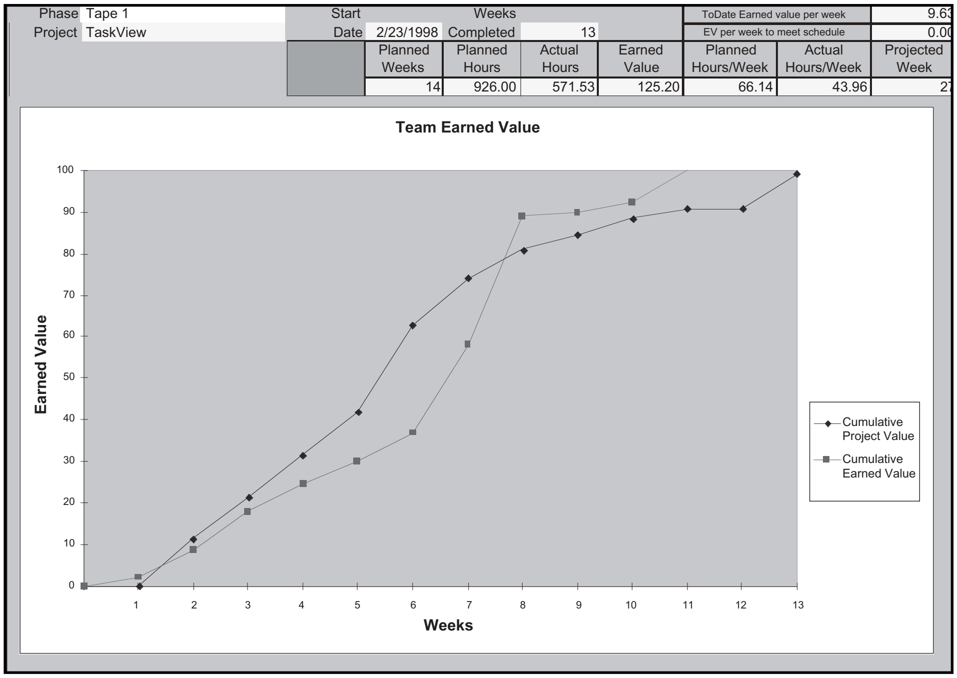
In order to get insight into project issues at the earliest possible moment, project data must be reviewed much more frequently than the traditional one-month milestone. During each weekly meeting, it is immediately obvious to TSP teams which tasks and team members are ahead of or behind schedule, and which tasks or team members need assistance. It is information like this that gives the team members the insight to make adjustments to task assignments, renegotiate functionality with the customer, or perform replanning activities to keep the project on track.

It is this combination of detailed planning, meticulous data gathering, and frequent reviews that makes the TSP's iterative refinement of project commitments possible. In fact, it is this ability, in combination with predictable test times due to the exceptionally high quality of the products they produce, that makes TSP teams so successful in on-time deliveries [2].

Let us take another look at Project Genesis, and this time let us assume the team had launched using the TSP. At first, the project earned value plan would look similar to the old plan they made before using TSP. Then, the team would determine that the next 10 weeks will constitute their *next phase*. This phase will consist of incorporating elements one through 10 as listed in Table 3. The team then would make an earned value plan for that period (Figure 8, see page 16).

Unfortunately, they still end up with a two-week period (Figure 8, circled) during which they cannot accurately determine their progress. This is due to a single program element – Element 2 – that is estimated to require more than the 30 task hours the project has estimated it will complete each week. Although this occurs early in the process, it could have a devastating impact on the outcome of the schedule. Using PSP techniques, the team members of Project Genesis break the large task down into its component elements (Table 4, see page 16).

Using this more refined estimate of Element 2, the earned value can be recalculated, producing the earned value chart in Figure 9 (see page 16). Notice that with this refinement, the *flat line* on the earned value chart has disappeared. Now the project has a detailed earned value plan that can be reviewed each week to determine if the project is actually meeting schedule.



Phase	Tape 1	Start		Weeks		ToDate Earned value per week	9.63
Project	TaskView	Date	2/23/1998	Completed	13	EV per week to meet schedule	0.00

Planned Weeks	Planned Hours	Actual Hours	Earned Value	Planned Hours/Week	Actual Hours/Week	Projected Week
14	926.00	571.53	125.20	66.14	43.96	27

Figure 6: *An Actual TSP Earned Value Chart from the TaskView Project*

All the Right Behavior

When the TaskView project first launched into the TSP at Hill Air Force Base in 1998 [3], the team did not have a lot of experience with this earned value methodology. Prior to the launch, the team had defined what they thought was a very thorough plan with more than 30 separate tasks. To do the launch and get the earned value chart using the TSP method, they doubled the number of tasks, which they felt was very thorough indeed. This did, however, leave three or four flat-line areas on their earned value chart similar to the one in Figure 8.

I was serving as project leader at the time and a few weeks after our launch the engineers came to me with a complaint: "Our tasks aren't broken down enough to earn value every week!" So, with the insistence of the engineers, we used PSP phases (as shown in Table 2) to further refine the plan until our tasks were small enough (10 task hours or fewer) to show earned value each week. This activity increased the number of tasks to 204, and the engineers were happy about it. Those are the kinds of engineers TSP teams produce!

The launch coach was correct when he said that TSP earned value "drives all the right behavior." Because it is reviewed each

> *"My experiences show that traditional earned value, while an effective tool, is rarely used correctly to predict and manage project performance and, as such, is usually incomplete."*

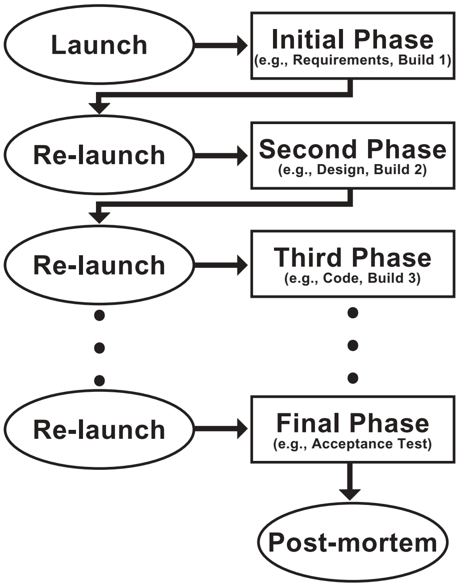


Figure 7: *The TSP Launch Process*

Task	Estimated Hours
Element 1	28
Element 2	85
Element 3	12
Element 4	26
Element 5	22
Element 6	5
Element 7	26
Element 8	29
Element 9	19
Element 10	23

Table 3: *Project Genesis Tasks for the "Next Phase"*

Task Element 2	Estimated Hours 85
Planning	10
Design	33
Design Review	17.5
Code	12
Code Review	6
Compile	0.5
Test	4
Post-mortem	2

Table 4: *Breakdown of Element 2*

week, and because no value is earned either at the personal or the team levels until a task is fully completed, software engineers are highly motivated to perform these good earned value practices:

- Follow a strictly defined process with very specific entry and exit criteria as well as well-defined tasks.
- Break large tasks into small pieces that can more easily be estimated and tracked and shows regular progress.
- Project forward to see if their progress will meet the current schedule.
- Re-plan when unplanned events arise.
- Do the right work, in the right order, at the right time.

My experiences show that traditional earned value, while an effective tool, is rarely used correctly to predict and manage project performance and, as such, is usually incomplete. In fact, many customers not only feel overwhelmed with acronyms like BCWS, BCWP, and ACWP, they do not trust charts that have all of those data and more splattered across them because so often they have seen results contrary to those charts. The TSP earned value techniques work because they collect data at the right level, they are simple, and they are measured each week.

TSP earned value works. It does indeed drive all the right behavior.◆

Figure 8: *Project Genesis Using TSP*

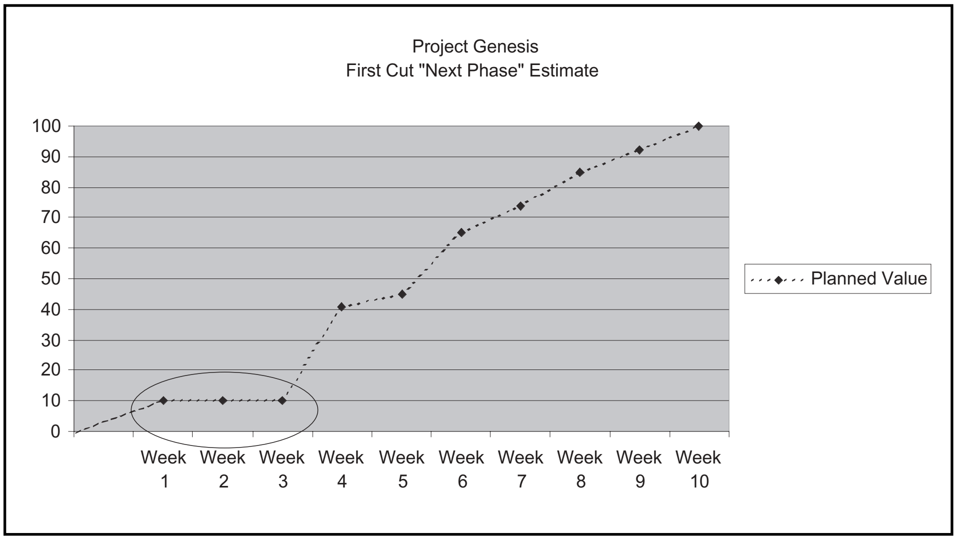
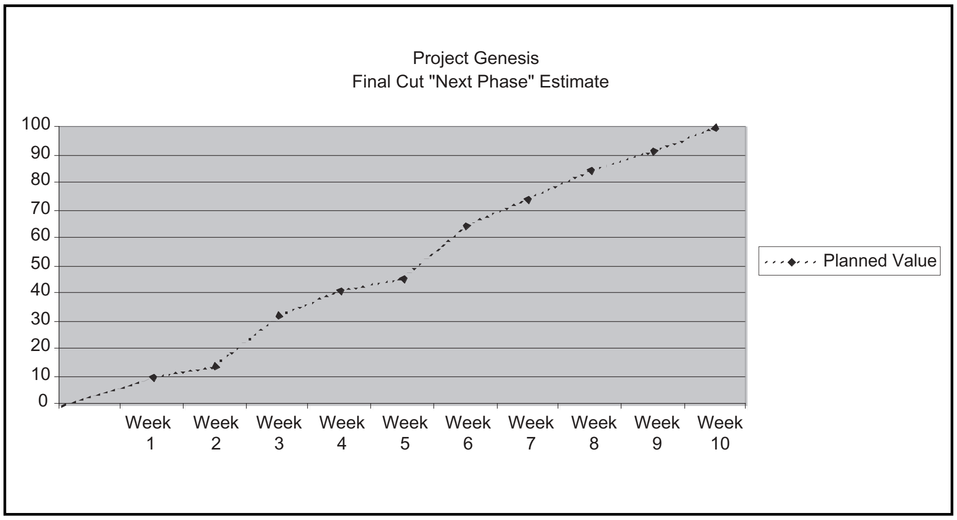


Figure 9: *Project Genesis "Final Cut"*



References

1. Humphrey, Watts S. "Making Software Manageable." CrossTalk Dec. 1996.
2. McAndrews, Donald R. The Team Software Process (TSP): An Overview and Preliminary Results of Using Disciplined Practices. Pittsburgh: Software Engineering Institute, Nov. 2000. 30.
3. Webb, David R., and Watts S. Humphrey. "Using the TSP on the TaskView Project." CrossTalk Feb. 1999: 3-10.

Note

1. Since schedules are living documents and must be renegotiated with the customer during development as new situations arise, this statement does not imply TSP teams always meet the original schedule set forth at project inception. I was involved with three TSP projects that met or exceeded the negotiated schedule. In one case, the customer shortened the original schedule; in another, the due date was extended; in the third, the acceptance test group was not ready to receive the product so more functionality was added during the *down* time. However, the ability to accurately renegotiate schedules *on the fly* and early in the process, to a customer's satisfaction, is one of the great strengths of TSP earned value and its iterative approach.

About the Author

David R. Webb is a project management and process improvement specialist for the Software Division of Hill Air Force Base in Utah, a Capability Maturity Model® for Software Level 5 software organization. He has 14 years of technical, program management, and process improvement experience with software in the Air Force. Webb is a Software Engineering Institute-certified instructor of the Personal Software Process^{SM} and a certified Team Software Process^{SM} launch coach. Webb has a bachelor's degree in electrical and computer engineering from Brigham Young University in Provo, Utah.

**7278 4th Street
Software Division, Bldg. 100
Hill AFB, UT 84056
Phone: (801) 777-9737
Fax: (801) 775-3023
E-mail: david.webb@hill.af.mil**