Using Earned Value for Performance Measurement on Software Development Projects

Using Earned Value for Performance Measurement on Software Development Projects

Major David S. Christensen and Daniel V. Ferens

The earned value method is an internationally recognized project managment tool for measuring progress on projects. Despite its popularity, it has not been widely applied on software development projects. This paper proposes the use of earned value on software development projects. After a brief description of the earned value method, seven software metrics appropriate for earned value application are described. The use of these metrics should facilitate more effective management of software development projects.

INTRODUCTION

Measuring progress on software development projects is a difficult but important challenge for project managers. In the Department of De-

Major David S. Christensen is a tenured associate professor of accounting at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. He received a Ph.D. in accounting from the University of Nebraska in 1987. He is a member of the American Accounting Association, the American Society of Military Comptrollers, the Performance Management Association, and the Society of Cost Estimating and Analysis.

Daniel V. Ferens is a tenured associate professor of systems management at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. He has a Master's Degree in Electrical Engineering from Rensselaer Polytechnic Institute, and a Master's Degree in Business from University of Northern Colorado. He is the author of the book *Defense System Software Project Management*.

fense (DoD), computer software costs are a substantial and growing portion of the budget. In 1993, for example, DoD software development costs are estimated at over $30 billion (Defense Systems Management College [DSMC], 1990). Similar trends are apparent in high-tech commercial projects.

Since 1967, the DoD has used a performance measurement technique known as "earned value" to monitor performance on defense contracts. In the DoD, the earned value method is usually implemented with Cost/Schedule Control Systems Criteria (C/SCSC). However, the method does not require C/SCSC. As a result, earned value is rapidly becoming an internationally recognized tool in project management, with both defense and nondefense applications.

Despite the established utility of the earned value method, its use on software development projects has not been widespread. Based on discussions with DoD project managers and analysts, software development progress is often assumed to be unmeasurable, and software projects are classified as "level-of-effort." Given the relative importance and cost of these projects, arbitrarily classifying them as "level-of-effort" is extremely unfortunate.

This paper proposes the use of the earned value method to measure progress on software development projects. After a brief description of the earned value method and related topics, seven software metrics are described and evaluated for their appropriate application in a performance measurement system that is based on the earned value method.

BACKGROUND

Performance Reporting and C/SCSC

To facilitate the effective cost management of defense acquisitions, the DoD requires standardized cost management reports from defense contractors. Two reports that specifically focus on cost and schedule performance are the *Cost Performance Report* (CPR) and the *Cost/Schedule Status Report* (C/SSR). The CPR is normally submitted on contracts which require compliance with the Department of Defense Cost/Schedule Control Systems Criteria (C/SCSC). For contracts not required to comply with C/SCSC, the C/SSR is usually required.

Compliance with the C/SCSC is required on "significant" contracts and subcontracts within all acquisition programs, including those that require software development. DoD Instruction 5000.2 defines significant contracts as research, development, test, and evaluation contracts with an estimated cost of $60 million or more (in fiscal year 1990 constant dollars), or procurement contracts with an estimated cost of $250 million or more (in fiscal year 1990 constant dollars). For contracts

below these thresholds, compliance with C/SCSC may also be required when contract risk is judged to be high. Compliance with C/SCSC on firm fixed price contracts is not normally required (Department of Defense [DoD], 1991, February).

Cost/Schedule Control Systems Criteria are not a management system imposed by the government on the contractor. Instead, the criteria establish minimal standards for the contractor's existing planning, scheduling, budgeting, accounting, and analysis systems, collectively termed the contractor's "internal management control systems." In total, there are 35 rather generic standards. One criterion, for example, requires a comprehensive budget for all the authorized work on the contract. Another criterion requires that all the authorized work be scheduled.

The DoD specifies two objectives for the criteria: (a) for contractors to use effective internal cost and schedule management control systems; and (b) for the government to be able to rely on timely and verifiable data produced by those systems for determining product-oriented contract status (Department of the Air Force [DAF], 1989, October). Implicit in these objectives is the assumption that if the contractor's management control systems comply with the criteria, then the data generated by those systems are reliable.

The cost management report summarizes the contract's cost and schedule performance using the key data elements shown in Figure 1. The Budgeted Cost of Work Scheduled (BCWS) is the sum of budgets allocated to time-phased elements of work on the contract, known as work packages. The cumulative expression of these budgets, termed the "Performance Measurement Baseline," takes on a characteristic S-shaped curve. The end point of the baseline, termed the "Budget at Completion" (BAC), represents the total budget of all the identified work on the contract.

Another key data element is the Budgeted Cost of Work Performed (BCWP). BCWP, also known as "earned value," is the same number as BCWS. They are both the budgeted cost of work. The only difference is when they are recorded. BCWS is recorded when work is planned to be completed; BCWP is recorded when work is actually completed. If work is completed at a different time from when it was planned to be completed, then a "schedule variance" is identified. Figure 1, for example, illustrates an adverse schedule variance because cumulative BCWS exceeds cumulative BCWP. When all of the work on the contract is completed, cumulative BCWP will equal cumulative BCWS.

Figure 1 illustrates two other variances: cost variance and variance at completion. A cost variance is the difference between BCWP and Actual Cost of Work Performed (ACWP). In this example, the cost vari-

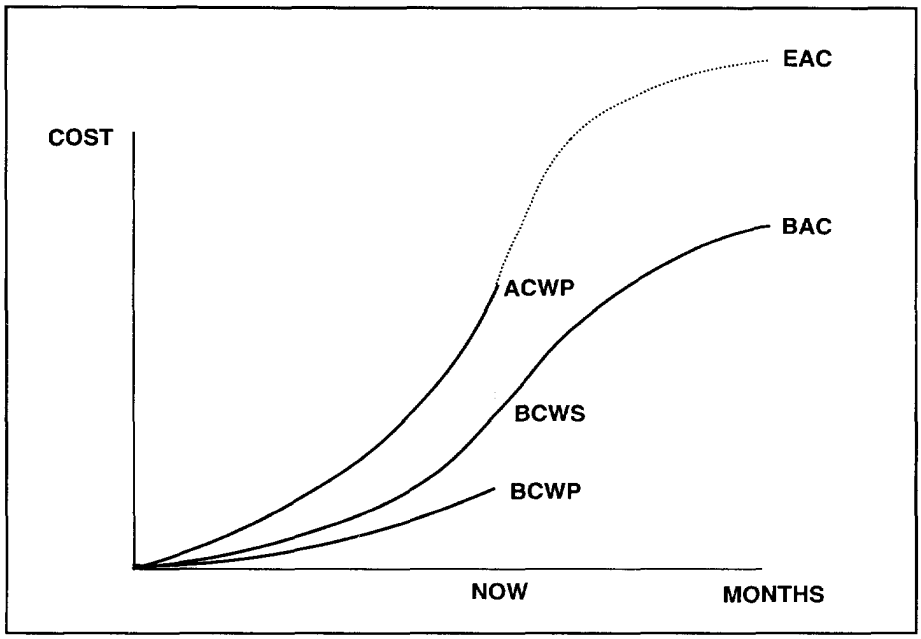

Figure 1. **The Performance Measurement Baseline (PMB)**

ance is unfavorable because the actual cost exceeds the budgeted cost of the completed work. The variance at completion is the difference between the Estimate at Completion (EAC) and the Budget At Completion (BAC).

The EAC is simply the actual cost of completed work plus an estimate of the cost to complete the remaining work on the contract. This estimate is reported by the contractor on the cost management report and reviewed for reasonableness by the government. When this projected final cost exceeds the budget, the contractor is effectively predicting an overrun, termed an adverse "variance at completion." Figure 1 illustrates the usual condition of a defense acquisition contract: behind schedule and overrunning the budget (Christensen, Antolini, and McKinney, 1992).

The C/SCSC require that all "significant" variances on the contract be analyzed. By definition, a significant variance is one that breaches a threshold (DAF, 1987, October, pps. 3-17). Thresholds are usually expressed as a percentage and in dollars. If, for example, a threshold for a work package was ±10 percent and $10,000 dollars, then any variance that breached thiss threshold would be investigated and it is to be hoped, corrected. The intent is that though disciplined variance analysis, prob-

lems can be corrected before they become serious.

Clearly, for variance analysis to be effective, the proper planning and measurement of earned value is essential. One of the purposes of the criteria (C/SCSC) is to assure that the earned value method is properly planned and implemented. Earned value (BCWP) is the key number on the cost management report. If it is inaccurate, then the three variances and the EAC are also wrong. It is possible, however, to use the earned value method without the criteria. When this is the case, controls similar to those described by the criteria should be enforced. Otherwise, BCWP will not be a reliable indicator of progress on the project. This paper will now describe how BCWP is planned and measured.

Planning and Measuring Earned Value

As described earlier, the criteria require that all the work on a contract be budgeted and scheduled. To accomplish this, the contractor will first develop a product-oriented family tree of hardware, software and services that successively subdivides all of the authorized work on the contract. This detailed breakdown of the work, termed the "Contract Work Breakdown Structure" (CWBS), typically extends to levels where work is to be performed, called "work packages." There may be over 100,000 work packages on a large defense acquisition contract.

A work package has three characteristics: technical content, schedule, and budget. Once the contract is subdivided into work packages, each work package is arranged in the order that it has to be accomplished, assigned start and stop dates, and assigned a budget. The budget for each work package is then spread through the life of the work package according to the technical requirements of the work. These "time-phased" budgets for all work packages become the basis for monthly BCWS, monthly BCWP, and the Performance Measurement Baseline. The proper time-phasing of the budget is thus critical to the planning of BCWS and the subsequent measurement of BCWP.

There are many "earned value methods" to time-phase the budget for BCWS and BCWP (Fleming, 1992, pps. 119-127). As indicated in Table 1, earned value methods depend upon the nature of the work that is being measured. Progress on the contract should ideally be measured by assessing discrete tasks which have a specific end product or end result. Work of this kind is termed "discrete effort." Common earned value methods appropriate for discrete effort include weighted milestones, interim milestones, and percent complete.

Work that can be directly related to other identified discrete tasks, such as quality control or inspection, is termed "apportioned effort." Support type activities, such as sustaining engineering or coordination,

TABLE 1
EARNED VALUE METHODS

Category of Work	Earned Value Method
Discrete Effort	Weighted Milestones (e.g., 50-50)
	Interim Milestones
	Percent Complete
Apportioned Effort	"Factored" on Discrete Effort
Level of Effort	BCWP set equal to BCWS

that does not result in a final end product is termed "level of effort" (LOE). On criteria-compliant contracts, these categories are mutually exclusive and collectively exhaustive. All work must be classified into one of these categories.

Although the criteria allow the contractor to use any one or any combination of these earned value methods, there are some general requirements. These requirements are intended to insure the usefulness of the performance measurement data.

To be useful for performance measurement, the data must be verifiable and objective. Therefore, the contractor must document the earned value method used in developing BCWS before the work begins, and then use the same method for measuring BCWP when work is being performed. Because BCWS and BCWP are the same number, it's absolutely essential that the same method be used for each. In addition, allowing one method for BCWS and another for BCWP would allow the contractor to distort performance measurement and the variances reported on the cost management report.

In addition to being verifiable and objective, the numbers for BCWP must be valid; namely, BCWP must clearly reflect performance. Therefore, the use of arbitrary measurement methods, such as the weighted milestone method, are limited to short-span work packages. An example of an arbitrary weighted milestone method is the "50-50" method, where one half of the budget for the work is "earned" (recorded as BCWP) when the work begins, and the other half is earned when the work is completed. To minimize the distortion created by such an arbitrary performance measurement, the method is generally restricted to work packages with durations of two months or less.

For longer work packages, "interim milestones" are required, where a portion of the budget for the work is assigned to each milestone. When that milestone is accomplished, the budget for that milestone is recorded

as BCWP. As long as the milestones are tangible and integral parts of the work, this interim milestone method will properly reflect performance on long-span work packages.

For some work packages, identifying interim milestones may not be possible. In this case, the contractor may simply estimate the percentage of the work planned to be completed for planning BCWS, and later estimate the percentage of work actually completed for recording BCWP. It is to be hoped that the contractor will employ some objective parameter of progress as a basis for estimating the percent complete. In any case, the criteria require that the contractor's method for determining BCWP be rational. The contractor should, therefore, be able to explain the basis for determining the estimates of BCWS and BCWP.

Another requirement related to earned value methods involves the proper matching of ACWP with BCWP. To facilitate the timely analysis of cost variances, ACWP should be recorded in the same period that BCWP is recorded. When BCWP for a work package is recorded but the actual cost is not yet known, ACWP may be estimated. Later, when the actual cost is known, ACWP can be adjusted.

EARNED VALUE AND SOFTWARE DEVELOPMENT

It has been difficult to use earned value methods on software development projects. Models that predict the amount and timing of software development costs, and metrics for accurately measuring work accomplishment have been inadequate. An obvious metric, percentage of code written, is both deficient and misleading. For earned value methods to be effective, BCWS and BCWP must be reflect the timing and technical requirements of the work. Software development involves much more than writing code, and the most difficult coding is often accomplished last. Therefore, using the percentage of code written as an arbitrary method to plan BCWS and record BCWP would not be an appropriate application of the earned value concept.

Fortunately, there are more appropriate methods or metrics for planning and measuring software development costs. Some of these can be used to adequately plan BCWS, and measure BCWP and ACWP. Regardless of the metric, the general approach is to divide the work into portions, establish a schedule and a budget for each portion, and then use this time-phased budget as baseline against which performance is measured.

Figures 2 and 3 illustrate how software projects are planned. Figure 2 represents a typical hierarchical breakdown of a system into hardware configuration items (HWCIs) and computer software configuration items (CSCIs). CSCIs are divided into computer software components (CSCs)

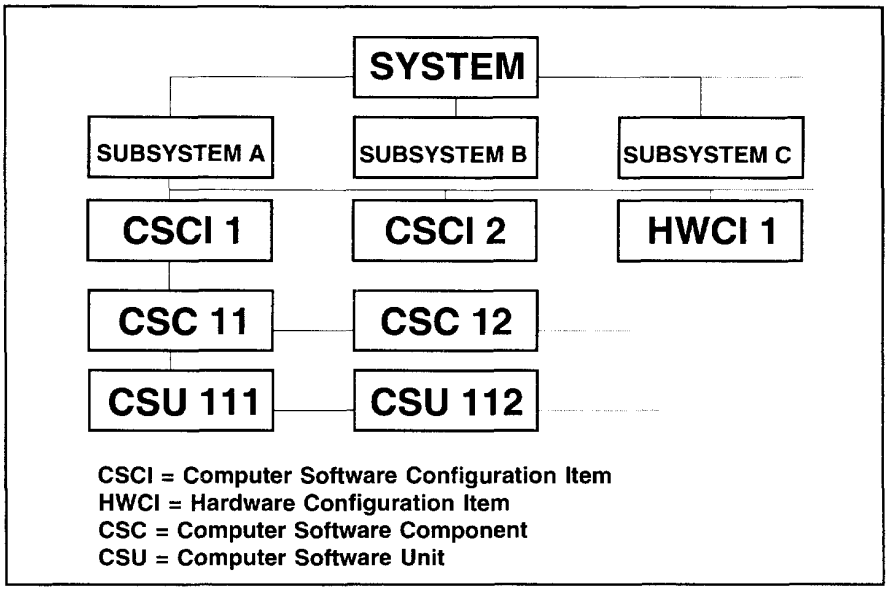

Figure 2. **A System Hierarchy for Software Development**

and computer software units (CSUs), which represent the lowest-level subfunctions of the software (DoD, 1988, February). For performance measurement to be meaningful, performance and actual costs should be planned and measured where work is being performed. For software development projects, this should be at the CSCI level or below. At higher levels, the planning of BCWS and the measurement of BCWP and ACWP would require rather arbitrary and subjective estimates of actual progress and costs.

To facilitate the objective measurement of progress and costs, earned value methods typically require the use of work packages. Figure 3 illustrates the typical software development process, known as the "waterfall" model described in DOD Standard 2167A (DoD, 1988, February). Each phase of this process may be considered a work package, appropriate for earned value application. The second through seventh phases are performed at the CSCI level. Coding does not begin until the fifth phase. In the waterfall model, a coded product is not available until CSCI testing is completed; however, the completion of earlier phases is extensively documented and includes reviews and audits to assure adequacy.

Using the phases of software development as work packages for earned value application appears to be a viable approach, especially if the cost and schedule of each phase can be estimated with reasonable accuracy,

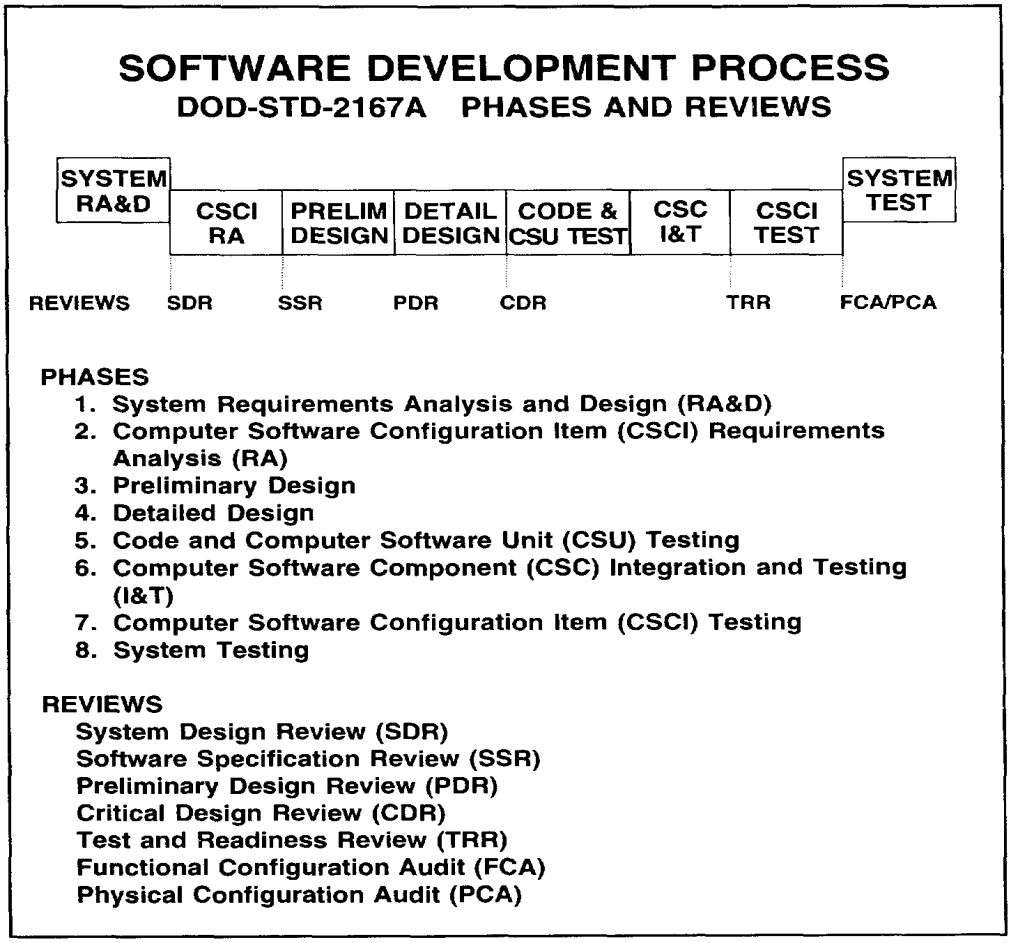

Figure 3. **The Software Development Process**

and appropriate metrics for measuring technical progress and cost within each phase are available. The earned value method generally requires that the cost and schedule for each phase (work package) be estimated. Next, an appropriate metric to measure cost and technical progress is identified and used to develop the time-phased budget (BCWS). Finally, as work is accomplished for that work package, the time-phased budget for the accomplished work is recorded as BCWP and its cost is recorded as ACWP.

Several models are available for predicting the cost and schedule for each phase of a software system or CSCI, including the Constructive Cost Model (COCOMO), PRICE-S, SEER, SLIM, SoftCost-R, and

Checkpoint (Ferens, 1990). Although the accuracies of these models have not been validated for a broad range of programs, they are generally suitable for rough estimates. For a review of the accuracy of these models, see Institute of Electronic and Electrical Engineers (IEEE) (1988).

Once the budget and schedule for each work package have been estimated, software metrics may be used to plan BCWS and measure BCWP and ACWP. Although much research has been performed on software metrics, there is currently very little standardization. Therefore, a manager must determine which metric is appropriate for each phase of the project.

There are several desirable properties of software metrics (Conte, Dunsmore, and Shen, 1986; DeMarco, 1982; Humphrey, 1990; Jones, 1991). To be useful, the metric should be (a) **relevant** to the work being measured; (b) **explicit** (directly measurable); (c) **objective**; (d) **absolute** (able to be assessed without reference to an average); (e) **timely** (available early in the project); and (f) **independent** from the influence of personnel performing the project. Of these, Ayres and Rock (1992) found relevance to the most important property. Accordingly, the metrics appropriate for BCWS, BCWP and ACWP were chosen with this property in mind. The first two metrics are appropriate for earned value measurement, and the third is most appropriate for ACWP. The remaining four metrics are more useful in investigating variances than in the direct measurement of earned value or actual costs. Each metric and its relevance to the earned value approach is now be briefly described. A more detailed description of these metrics is provided elsewhere (Ayres and Rock, 1992; DoD, 1991, February).

1. **Requirements and Design Progress**. This metric is based on the number of CSCI requirements determined during the first two phases of software development. The requirements are detailed in several documents (System/Segment Design Document, Software Requirements Specification, Software Design Document) written during these phases. As illustrated in Figure 4, the planned and actual CSCI requirements are used for determining BCWS and BCWP, respectively. Figure 4 also illustrates that the total CSCI requirements may change. In addition, counting the requirements can be difficult. If these limitations can be overcome, this metric is a viable tool for earned value application, especially early in the project.

2. **Code and Testing Progress**. This metric is based on the number of CSUs that have been designed, coded, and tested. As illustrated in

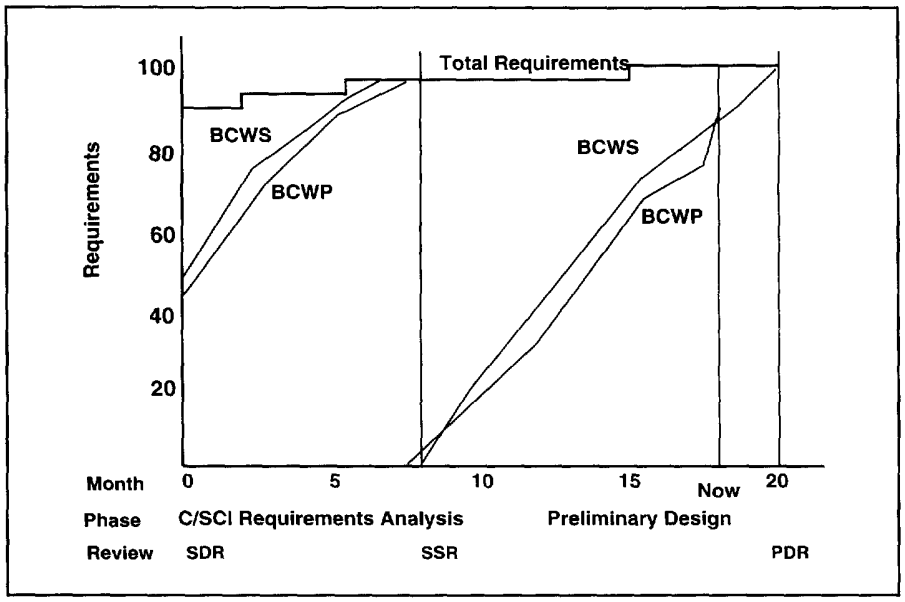

Figure 4. **The Requirements and Design Process Metric**

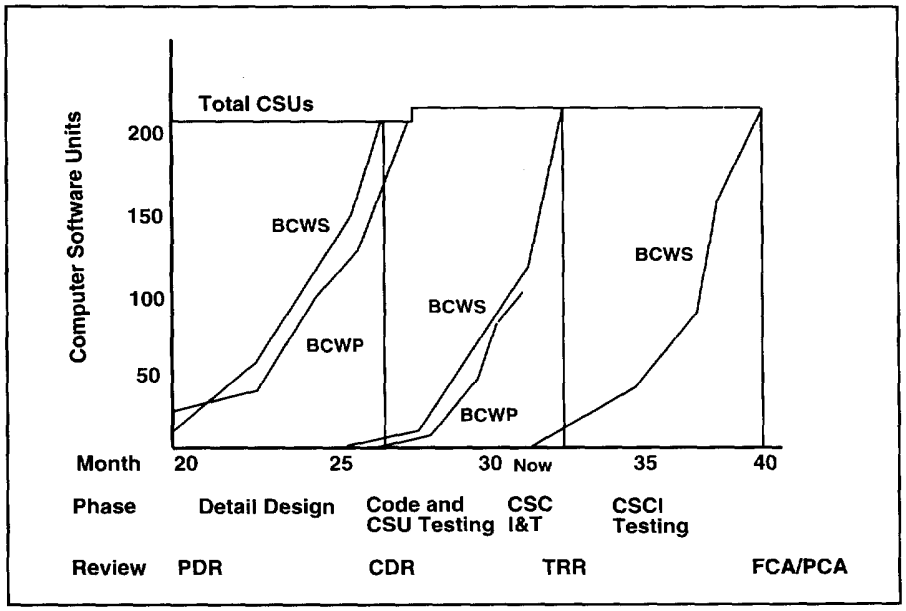

Figure 5. **The Code and Test Progress Metric**

Figure 5, it is appropriate after the second phase of the software development project. Like the previous metric, the planned and actual CSUs represent BCWS and BCWP. In addition, the total number of planned CSUs for each phase represents the end point of the performance measurement baseline for that phase. Generally, this metric is easier to measure than the previous one. CSU progress can be measured using a unit development folder or similar technique. Also, more detailed information is known about the software project in these later phases.

3. **Person-months of Effort**. As illustrated in Figure 6, this metric is based on person-months throughout the project. As such, it is particularly useful for measuring ACWP because the costs of software development are almost entirely labor-related. Using planned person-months for BCWS and BCWP is probably inappropriate because available estimation methods may be inaccurate, and the time spent on the project may not correlate to progress. Nevertheless, this metric is useful, if only because it is the single metric in this collection that directly reflects ACWP.

4. **Software Size**. This metric tracks the size of the software during the entire project. Usually, size is expressed in source lines of code

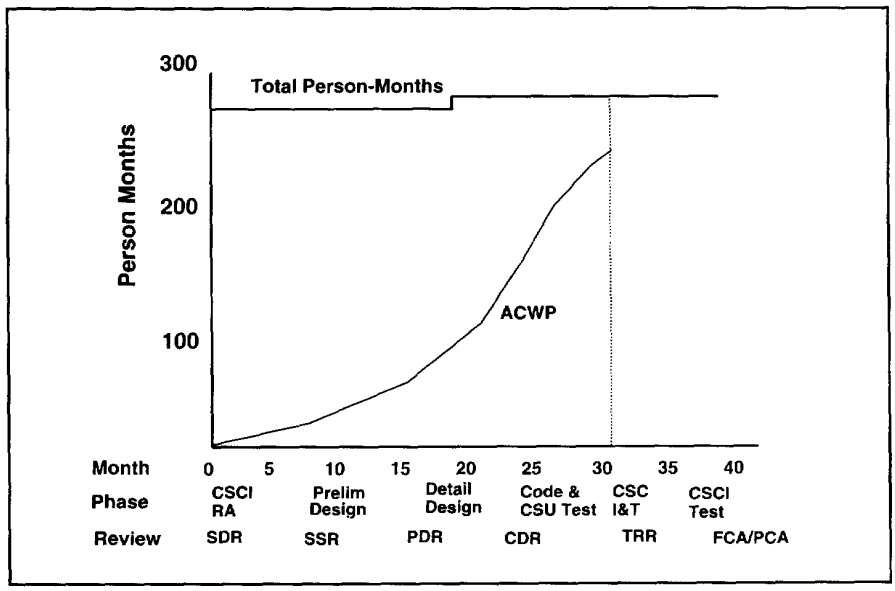

Figure 6. **The Person-Months Progress Metric**

(SLOC). The total size may be divided into categories of new, modified, and reused code. Since there is a direct relationship between size and effort required, this metric may be helpful in estimating actual cost. However, effort required and actual progress may not correlate; accordingly, the method may be inadequate as an earned value metric, and should be used as a technical parameter to investigate the cause of cost variances based on the other metrics.

5. **Computer Resource Utilization**. This metric is a measure of the available computer hardware timing, memory, and input/output (I/O) resources consumed by the software. It is closely related to the software size metric described above in that increases in total size will result in a greater percentage of hardware resources utilized. Like software size, this metric can be helpful early in the program for determining the causes of variances.

6. **Requirements Stability**. This metric has similarities to the requirements and design progress metric. Like that metric, requirements stability tracks total requirements; however, it also tracks the number of changes (additions, deletions, and modifications) made to requirements throughout the entire development process. Numerous or frequent changes will result in additional effort required, and may explain unfavorable cost and schedule variances.

7. **Design Stability**. This metric is like requirements stability in that it tracks the number of changes to the detailed design (CSUs). Like the code and testing progress metric, it is primarily useful later in the program, after preliminary design is completed. Frequent lower-level design changes will result in additional effort required.

CONCLUSION

Table 2 lists the seven metrics described in this paper, and indicates the role that each metric could have in an earned value performance measurement system. The table also indicates our judgment of how well the metric satisfies the seven desirable properties of software metrics. Because these properties are nearly identical to the goals for earned value measurement that are described in C/SCSC, they appear to be viable candidates for earned value application, especially the first three listed in the table.

Of course, the seven metrics described in this paper are not the only ones. Especially worthwhile are "quality metrics" that track defects, com-

Table 2
SOFTWARE METRICS FOR EARNED VALUE APPLICATION

Metric	Desirable Attributes of Software Metrics						Role in Earned Value System	Phase(s) in Software Development Process
	Relevant	Explicit	Objective	Absolute	Timely	Independent		
Requirements and Design Progress	Yes	Yes	Somewhat	Yes	Yes	Somewhat	BCWS and BCWP	CSCI RA and Prelim Design
Code and Test Progress	Yes	Yes	Somewhat	Yes	No	Somewhat	BCWS and BCWP	Detail Design, Code and CSU Testing, CSC I&T, and CSCI Testing
Person-Months Progress	Yes	Somewhat	Yes	Yes	Yes	Somewhat	ACWP	All phases
Program Size	Yes	Somewhat	Yes	Yes	Yes	Somewhat	Technical indicator for variance analysis	All phases
Computer Resource Utilization	Yes	No	Somewhat	Yes	Somewhat	Yes	Technical indicator for variance analysis	All phases
Requirements Stability	Yes	Yes	Yes	Somewhat	Yes	Yes	Technical indicator for variance analysis	All phases
Design Stability	Yes	Yes	Somewhat	Somewhat	No	Yes	Technical indicator for variance analysis	All phases

plexity and modularity (Jones, 1991). While these metrics may not directly relate to earned value measurement, they do help measure quality, which is the *sine qua non* of software projects today; using them in tandem with the ones recommended for earned value application is highly recommended.

REFERENCES

Ayres, Bradley J., and William M. Rock (1992, September). *Development of a Standard Set of Software Indicators for Aeronautical Systems Center* (AFIT Thesis GSS/ENC/92D-1), Dayton, Ohio, Air Force Institute of Technology.

Christensen, David S., Richard Antolini, and John W. McKinney (1992, July). A Review of Estimate At Completion Research, *Proceedings of the Society of Cost Estimating and Analysis Society.*

Conte, S. D., H. E. Dunsmore, and V. Y. Shen (1986). *Software Engineering Metrics and Models.* Menlo Park, California: Benjamin-Cummings.

Defense Systems Management College (1990). *Mission Critical Computer Resources Management Guide.* Fort Belvoir, Virginia: Government Printing Office.

DeMarco, Tom (1982). *Controlling Software Projects*, Englewood Cliffs, New Jersey: Prentice-Hall.

Department of the Air Force (1987, October). *Cost/Schedule Control Systems Criteria Joint Implementation Guide* (Air Force Systems Command Pamphlet 173-5). Washington DC: Headquarters Air Force Systems Command.

Department of the Air Force (1989, September). *Guide to Analysis of Contractor Cost Data* (Air Force Systems Command Pamphlet 173-4). Washington DC: Headquarters Air Force Systems Command.

Department of Defense (1991, February). *Defense Acquisition Management Policies and Procedures* (Department of Defense Instruction 5000.2). Washington DC.

Department of the Air Force (1991, November). *Software Management Indicators.* (Air Force Pamphlet 800-48). Washington, DC.

Department of Defense (1988, February). *Defense System Software Development* (Department of Defense Standard 2167A). Washington, DC.

Ferens, Daniel V. (1990). *Defense System Software Project Management.* Dayton, Ohio: Government Printing Office.

Fleming, Quentin W. (1992). *Cost/Schedule Control Systems Criteria - The Management Guide To C/SCSC.* Chicago, Illinois: Probus Publishing Company.

Humphrey, Watts S. (1990). *Managing the Software Process.* Reading, Massachuttes: Addison-Wesley.

Institute of Electronic and Electrical Engineers (1988, June). *Standard Dictionary of Measures to Produce Reliable Software* (IEEE Standard 982.1-1988). New York: Institute of Electronic and Electrical Engineers.

Illinois Institute of Technology Research Institute (1989). *Estimating the Cost of Ada Software Development*, Lanham, MD: IIT Research Institute.

Jones, Capers (1991). *Applied Software Measurement.* New York, McGraw-Hill.