Applying Earned Value Management to Software Intensive Programmes

~ By Bob Hunt

Many information technology projects have been declared too costly, too late and often don’t work right. Applying appropriate technical and management techniques can significantly improve the current situation. The principal causes of growth on these large-scale programmes can be traced to several causes related to overzealous advocacy, immature technology, lack of corporate technology road maps, requirements instability, ineffective acquisition strategy, unrealistic programme baselines, inadequate systems engineering, and work-force issues. This article provides a brief summary of four processes to resolve these issues.

Establishing a Process for Requirements Definition and Developing the Technical, Cost and Schedule Baselines

We all realise the importance of having a motivated, quality work force but even our finest people can’t perform at their best when the process is not understood or not operating at its best. A well defined process is critical to defining the requirements and completing the initial cost and schedule estimate. The proper use of Performance-Based Earned Value (PBEV) provides for integration of project technical scope, schedule, and cost objectives; and the establishment of a baseline plan for performance measurement. Additionally, the use of an analytic application to project likely cost and schedule based on actual performance provides for realistic projections of future performance. Success of the project can be aided by defining the best objectives, by planning resources and costs which are directly related to those objectives, by measuring accomplishments objectively against the plan, by identifying performance trends and problems as early as possible, and by taking timely corrective actions.

In the book, "Software Sizing, Estimation and Risk Management" (Dan Galorath and Michael Evans, 2007) a ten step process is presented for programme requirements generation and estimation. The 10 steps are:

1. Establish Estimate Scope
2. Establish Technical Baseline, Ground Rules, and Assumptions
3. Collect Data
4. Estimate and Validate Software Size
5. Prepare Baseline Estimates
6. Review, Verify and Validate Estimate
7. Quantify Risks and Risk Analysis  
8. Generate a Project Plan  
9. Document Estimate and Lessons Learned  
10. Track Project Throughout Development

The key here is to establish an auditable, repeatable set of steps to establish the requirements and develop the baseline estimate of cost and schedule.

**Identifying Critical Software Management Metrics**

That most large software programmes get into trouble is a demonstrated phenomenon. Therefore selecting the correct set of software metrics to track is critical to programme success. Practical Software Measurement (McGarry, Card, Jones; Addison-Wesley, 2002) identifies seven information categories and expands these information categories into measurable concepts and then prospective metrics.

For Earned Value purposes, the most effective software metrics are those that relate to product size, schedule, quality, and progress. For software intensive programmes, measures of quantity (e.g. number of lines of code completed) do not accurately reflect the quality aspects of the work performed on neither the programme nor the actual progress since items such as lines of code completed do not capture items such as integration, testing, etc.

Size is often measured as Source Lines of Code (SLOC) or Function Points and used as a sizing measure for budgets and for earned value using a percent of completion method. There are two critical problems with this approach. First, there has traditionally been a significant error in estimating SLOC. And, the number of lines of code completed does not necessarily reflect the quality or total progress toward a performance goal. Therefore, any progress metric based solely on SLOC is highly volatile. Whether SLOC, function points, Use Cases, or some other size artefact is selected, a careful process must be utilised to establish a credible size metric. It is recommended that in addition to tracking progress toward a goal, size growth should also be tracked.

Schedule metrics and procedures normally relate to completion milestones are also a common tracking metric. Sometimes these milestone definitions and completion criteria lack quantifiable objectives. Often an incremental build is released that does not incorporate all the planned functional requirements or a developer claims victory after just testing the nominal cases.

Progress metrics can be very difficult for large software programmes. It is generally agreed that no software is delivered defect free. Software engineers have hoped that new languages and new processes would greatly reduce the number of delivered defects. However, this has not been the case. Software is still delivered with a significant number of defects. The physical and practical limitations of software testing (the only way to determine if a program will work is to write the code and run it) ensure that large programs will be released with undetected errors. Therefore, defects discovery and removal is a key metric for assessing program quality.

**Applying Performance-Based Earned Value (PBEV)**

Performance-Based Earned Value (PBEV) is an enhancement to the Earned Value Management Systems (EVMS) standard. PBEV overcomes the standard shortcomings with regard to measuring technical performance and quality (quality gap). PBEV is based on standards and models for systems engineering, software engineering, and project management that emphasise quality. The distinguishing feature of PBEV is its focus on the customer requirements. PBEV provides principles and guidance for cost effective processes that specify the most effective measures of cost, schedule, and product quality performance.
Programme managers expect accurate reporting of integrated cost, schedule, and technical performance when the supplier’s EVMS procedure complies with the EVMS Standard. However, EVM data will be reliable and accurate only if the following occurs:

- The indicated quality of the evolving product is measured
- The right base measures of technical performance are selected
- Progress is objectively assessed

Using EVM also incurs significant costs. However, if you are measuring the wrong things or not measuring the right way, then EVM may be more costly to administer and may provide less management value.

Because of the quality gap in the EVMS standard, there is no assurance the reported earned value (EV) is based on product metrics and on the evolving product quality. First, the EVMS standard states that EV is a measurement of the quantity of work accomplished and that the quality and technical content of work performed are controlled by other processes. A software manager should ensure that EV is also a measurement of the product quality and technical maturity of the evolving work products instead of just the quantity of work accomplished. Second, the EVMS principles address only the project work scope. EVMS ignores the product scope and product requirements. Third, the EVMS standard does not require precise, quantifiable measures of progress. It states that objective EV methods are preferred but it also states that management assessment (subjective) may be used. In contrast, other standards specify objective measurement. Fourth, EVM is perceived to be a risk management tool. However, EVMS was not designed to manage risk and provides no guidance on the subject.

PBEV is a set of principles and guidelines that specify the most effective measures of cost, schedule, and product quality performance. It has several characteristics that distinguish it from traditional EVMS, by augmenting EVMS with four additional principles and 16 additional guidelines.

PBEV supplements traditional EVMS with the best practices. Its principles and guidelines enable true integration of project cost, schedule, and technical performance. The distinguishing feature of PBEV is its focus on the customer requirements. Measures of product scope and product quality are incorporated into the project plan. Progress is measured against a plan to fulfil all customer requirements. Measuring the wrong things does not dilute management attention. Consequently, management is able to take rapid corrective actions on deviations that threaten customer satisfaction and business enterprise objectives.

**Using an Analytic Process to Project Cost and Schedule Based on Actual Performance**

Once the requirement definition is complete; the cost and schedule baseline has been established; the appropriate metrics have been selected; and a PBEV system is in place, the final challenge is to implement a process that quickly and accurately estimates final cost and schedule based on actual performance. This analysis is best accomplished using an analytic/parametric process. Galorath Incorporated calls this process SEER Control. The purpose of SEER Control is to provide an understanding of the project’s progress so that appropriate corrective actions can be taken when the project’s performance deviates significantly from the plan. SEER Control provides a "dashboard" that includes a health and status indicator for the project related to: schedule variance, time variance, cost variance, size growth, and defects discovery and removal.

At the heart of SEER Control is the ability to forecast the final project outcome based on actual performance to date. One of the primary goals of SEER Control is to provide adequate supporting documentation (charts and reports) to support the software project management process and to satisfy stakeholder needs.
Conclusion

Management of Software Intensive Programmes should be based on the foundation of establishing the requirements, developing a reliable baseline estimate for cost and schedule, selecting effective software metrics, applying Performance-Based Earned Value (PBEV), and using analytic processes to project cost and schedule based on actual performance.

Author's Note: This article was written with contributions from Paul Solomon, co-author of the book, Performance-Based Earned Value and Dan Galorath, CEO of Galorath Inc. and co-author of the book, Software Sizing, Estimation, and Risk Management.

Bob Hunt is V.P., Services for Galorath Inc. He has performed software programme assessments, SEI Checklist evaluations, software sizing analyses, and software cost estimating. As a civil servant, he was Deputy Director of Cost Analysis for Automation and Modeling, Cost Analysis Division, U.S. Army, The Pentagon.