

EVMS and the Agile Framework

Most software development methods, including Agile Methods, have a mechanism to measure progress to plan. But comparing actual cost with planned costs is simply measuring the “level of effort” consumed over a time period. This measurement does not describe the “value” delivered by the invested effort. The critical aspect of Earned Value Analysis is the determination of “value” delivered (BCWP) in exchange for hours or dollars invested (ACWP) for software projects.

This earned value is the basis for determining the cost and schedule performance for a task or project.

Technical Performance Measurement

Infer employs “Technical Performance Measurement” for measuring value. Technical Performance Measurement is the plan for expected technical achievement. The actual progress of the project is compared using periodic measurements or tests. The difference between the planned progress and the actual progress represents a technical variance. Technical Performance Measurement is an accepted Earned Value approach for assigning value to BCWP (Budgeted Cost for Work performed).

Testable Requirements

Within Earned Value’s Technical Performance Measurement our approach to measuring the “value” of a software component (BCWP) is the use of testable requirements as a completion criteria and a linearly adjusted monetary value for the component as a percent of Budgeted Cost for Work Scheduled (BCWS). A testable requirement can be decomposed to a collection of precise, unambiguous, and indivisible set of low-level requirements. These criteria are only met

if it is possible to write a test case that would validate whether the requirement has or has not been implemented correctly. This is the source of the term “testable requirement.”

Testable requirements provide several benefits for an EVMS based development method, including:

- ⌚ An overarching technical performance measure for identifying progress to plan.
- ⌚ The support of the contract measurement goals of a Performance Based Contract.
- ⌚ A uniform metric from the software conception phase through system acceptance
- ⌚ “Success oriented” metrics rather than “effort oriented.”
- ⌚ Integration of schedule and technical cost objectives in a single performance based metric. A testable requirement can be described in terms of:
- ⌚ The state of the system and the data elements that are inputs (e.g., customer number, product number)

- ⌚ The condition or action associated with the requirement (e.g., the user enters data, the order is validated, the check amount is deducted)

The expected or specified result described in terms of data elements (e.g., customer number must be 8 digit numeric, product quantity must be greater than zero).

Three Success Factors of Final Project Results

The success of using Earned Value Management to manage software development projects is dependent on three factors:

- ⌚ ***The quality of the baseline***

The establishment of a measurable baseline for work to be performed is difficult in the traditional software development effort.

Agile project methods focus much of their effort on defining and discovering the scope of work to be performance in iteration. Both XP and SCRUM have unique methods for capturing this scope of work.

- ⌚ ***The actual performance against the approved baseline***

Once the plan has been approved and implemented the second success factor comes into play -

The actual performance of the project activities.

- ⌚ ***Management's determination to influence the results given the performance indices.***

This is the most critical success factor for any project management method. Without a commitment from management to take aggressive actions based on the performance indicators to influence the outcome of the remaining work the project will fail to meet its desired outcomes.

Aggressive project management actions, if taken early, can often alter the final projected outcome for the project.

We've created a development environment that performs many of the XP practices while maintaining our reporting deliverables for EV compliance. This involved:

Replacing XP's velocity with Earned Value metrics.

Creating fine-grained measures of EV using "testable requirements."

Establishing the Budgeted Cost baseline at the beginning of each iteration.

Capturing Actual Cost through a time keeping system.

- Computing Cost Variance, Schedule Variance from the three base(s) earned value metrics

- Computing Estimate at Completion (EAC) and Estimate to Completion (ETC) from these base metrics as well.

Much of the "noise" about agile development, especially XP in the traditionalist environment has to do with how to position these processes in a larger context.

We've taken the approach that XP is for writing code, supporting the processes for writing code, and delivery code to the customer base.