Submitted to *Agile Development*, June 25–28, 2003 Salt Lake City, Utah

Making Agile Development Work in a Government Contracting Environment

Measuring velocity with Earned Value

Glen B. Alleman and Michael Henderson

CH2M HILL
Golden, Colorado
glen.alleman@rfets.gov
michael.henderson@rfets.gov

Ray Seggelke

Envision Technology Partners
Lakewood, Colorado
raymond.seggelke@rfets.gov

Abstract: Before any of the current "agile" development methods, Earned Value Management provided information for planning and controlling complex projects by measuring how much "value" was produced for a given cost in a period of time. One shortcoming of an agile development method is its inability to forecast the future cost and schedule of the project beyond the use of "yesterdays weather" metrics. These agile methods assume the delivered value, "velocity" in the case of XP, is compared with the estimated value – this is a simple comparison between budget and actual cost resulting in a Cost Variance. No Schedule Variance process is directly available in XP. Earned Value Analysis provides a means of predicting future schedule and cost variances through three measurements – budgeted cost for work scheduled, actual cost for work performed, and budgeted cost for work performed (earned value). This paper describes the use of Earned Value in conjunction with Agile Development on a mission-critical, high-security, government project.

1. Introduction

Measuring progress to plan is important no matter what the business domain or software development method. Extreme Programming, SCRUM, DSDM, FDD, Crystal, etc. provide techniques for capturing requirements, estimating effort, developing high quality software, reporting progress to plan, and delivering value to the customer. [1] The effectiveness of any specific agile method in current business environments is not the topic of this paper. The topic is the introduction of agile methods into high–ceremony government contracting environments that use Earned Value Management Systems as their performance reporting and management.

In a government subcontractor software development environment formal "artifacts" are needed for many reasons, not the least of which is compliance with the contract. [4] Earned Value Management Systems (EVMS) are the means of complying with a "progress–to–plan" reporting requirements. The desire to use agile software development methods while still maintaining compliance with contract reporting needs may appear to be a conflict, but it is not. The two approaches provide similar solutions to measuring progress. These include:

Earned Value	Agile Development
A Big Picture View of the Project.	Continuous production of useable software.
Accurate Estimate at Completion.	Prediciton of the next iteration's effort.
End-to-end value tracking.	Iteration to iteration tracking

Figure 1 – Basic Earned Value versus Agile Processes

The remainder of this paper describes our experiences with embedded Extreme Programming practices in a high–ceremony Earned Value software development environment.

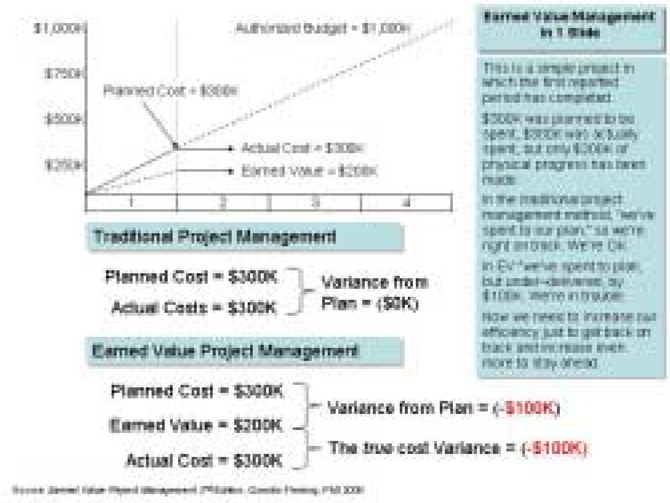
§ 2 Describes a brief overview of Earned Value
§ 3 Describes the realities of writing software in a government contracting environment.
§ 4 Positions Earned Value Management in the context of agile development
§ 5 Summarizes the steps needed and restates the context of XP in an EVMS process.

2. A Quick Tour of Earned Value

Earned value provides a balance of technical (performance), cost (resources), and schedule (time) measures for complex software projects, unlike traditional cost and schedule only techniques.

Earned Value is a project management technique that provides "leading" performance indicators that allow project managers to identify and control project problems before they become insurmountable. Traditional project management techniques compare planned expenditures with actual expenditures, which is equivalent to "driving in the rear view mirror." Earned Value adds a third measure – the actual work accomplishment as a result of the expenditure. Measuring the actual work accomplished provides greater insight into potential project risks.

risks. **Figure 2** describes a “simple” view of these metrics and their relationships.



Source: Earned Value Project Management 2nd Edition, Quentin Fleming, PMI 2000

Figure 2 – Earned Value is One Slide

Like any good methodology a set of terms unique to that method are needed. These Earned Value terms include:

- Budgeted Cost for Work Scheduled (BCWS) – this is the Plan and represents the total budgeted cost. It answers the question how much do we plan to spend?
- Budgeted Cost for Work Performed (BCWP) – this is the **Performance** or **Earned** Value and is the cost originally budgeted to accomplish the work that has been completed. It answers the question how much work has actually been completed?
- Actual Cost for Work Performed (ACWP) – this is the **Cost of the Performance** or the **Investment** and is the actual cost to accomplish all the work that was performed. It answers the question how much did we actually spend to deliver the Earned Value?
- Cost Variance – is the difference between planned cost and actual cost. CV= ACWP–BCWS.
- Schedule Variance – is the difference between the invested cost and the returned value. SV=BCWP–ACWP.
- Cost and Schedule Performance Indices – are the normalized performance indices. CPI=BCWP / ACWP, SPI=BCWP / BCWS
- Estimate at Completion and Estimate to Completion – are calculated values that are estimates of the total cost and cost to complete. EAC=Cost to Date + Estimated Cost of Remaining Work.

The essence of EVMS … is that at some level of detail appropriate for the degree of technical, schedule, and cost risk or uncertainly associated with the program, a target value … is established. — Paul Solomon [6]

3. REALITIES OF A GOVERNMENT PROJECT

3.1 What’s The Problem Here?

In the traditional government contracting environment a linear development process is common, usually based on high–ceremony work artifacts embedded in a CMMi compliant process. In many aspects these linear processes are valuable, since care and concern is needed when mission critical systems are developed. In nearly all cases the “project controls” for these projects are based on Earned Value measurements using the metrics shown in Figure 2.

When agility is introduced to the development process there is no real alternative for formally reporting progress–to–plan. The process of transforming a legacy environment to an agile development environment can take many paths. We took specific steps to not undo what had been put in place in the past, while at the same time moving forward. This includes:

- XP–like practices for software development [1] – these include most, but not all, of the XP practices See Figure 3.
- Balanced Scorecard – focuses the IT organization on strategies, objectives, and initiatives. This provides the “reason” for many of the development projects and process improvements. BSC starts at the top of the organization and moves down, so top management commitment is gained early in any improvement process.
- Project Portfolio Management – assembles collections of projects into a portfolio for decision making purposes.
- Team based delivery for all products and services – breaks the mold on the legacy “command and control” management style and replaces it with self–directed teams.

No matter what “work processes” are put in place, the contractual reporting of “progress–to–plan,” must be maintained. Reporting “velocity” to the Department of Energy was not an option when we started to introduce XP to the development process. Contract progress payments are based on “earned value” for the accounting period and therefore are considered our “rice bowl,” something you simply do not mess with.

3.2 OUR SOFTWARE DEVELOPMENT CHALLENGE

Most of the information and experience with earned value is centered on large programs with systems and

[1] We use the term XP–like to state that some of the XP practices aren’t in place. This also avoids explaining to the purist what practices we use or don’t use.

organizations in place explicitly to support project management and earned value. Our development is small compared to this environment. We have approximately 100 professional and technical support personnel that provide software development, infrastructure deployment and support, customer service, and program management. We currently hold a CMM Level 3 certification and are seeking CMMi Level 4. We would not be considered a "modern" development shop, since most of the work is done in Oracle forms, PeopleSoft's PeopleWare™ and other "low tech" tools.

Our mission is to provide applications and infrastructure to an $11B de-construction project of critical national importance. Rocky Flats Environmental Technology Site is the first Department of Energy (DOE) nuclear weapons production site to close on time on budget. Closure means all the nuclear materials used in production and their associated wastes will be removed from the 6,800–acre site, restoring it to its natural pristine high plains habitat. Our customer, Kaiser–Hill LLC, has a fixed price, fixed duration contract with DOE for the closure. Kaiser–Hill, DOE and our firm, CH2M HILL all live in the earned value mentality.

Introducing agile development processes into this environment is a challenge. Not because of the processes themselves, but because of the financial reporting, CMM compliance, and operational security requirements of the contract.

XP is our method of choice and Figure 3 summarizes the elements of XP currently in use in our project. Scrum is also a choice, but most of the "agile" developers and management have XP experience.

XP Practice	Our Implementation
Planning Game	Biweekly planning sessions.
Small Releases	Biweekly iteration releases with full integration with Configuration Management and IV&V.
Metaphor	Not used.
Simple Design	Forced on the team by the "time boxed" iteration process.
Refactoring	Not a major impact as yet.
Testing	Unit tests, integration tests, IV&V testing.
Pair Programming	Not usually allowed because of code access security requirements. Some PP within small groups.
Collective Ownership	Not usually allowed because of code access security.
Continuous Integration	Heavy investment in tools, but some progress for daily builds.
40 Hour Week	Mandated by our contract.
On–site Customer	Customer Service Managers (CRMs) with detailed process knowledge.
Coding Standards	Inherited from CMMI processes.

Figure 3 – XP Practices Deployed in a Government Environment

4. EVMS AND THE AGILE FRAMEWORK

> *Not everything that can be counted counts, and not everything that counts can be counted. – Albert Einstein*

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.

4.1 EVMS DEVELOPMENT VALUE MEASUREMENT

All of the methods described above depend either on a binary event or some subjective assessment of the progress that has been made during the reporting period. Both approaches fail the integrity test for software project management. This test asks the question – *how do we know that the software will behave as specified*? If it does behave as specified, then the development phase is complete. If not, then rework is needed. In the typical EVMS the budget for the task are used to accrue the value rather than the expected business value associated with the task's completion. [2]

4.1.1 Technical Performance Measurement

One approach to measuring value is to employ "Technical Performance Measurement." This approach is used on many engineering and development projects in defense systems. [5] 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 process for assigning value to BCWP. [5]

4.1.2 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

BCWS. [7, 8] 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).

4.2 Agile Development Value Measurements

In our use of EVMS, we took the approach of booking 0% or 100% of a task as BCWP and making the task durations sufficiently small. With this simple guideline, something then happens to the EVMS “value generating” approach – it looks similar, in many respects, to an Agile software development process. With this *fine–grained* task breakdown process, all the EVMS principles are still in place, but the behavior of the management system has many of the attributes of an agile process.

There are still gaps to be closed, but the two paradigms are now closer together than one would first imagine. In an XP environment “velocity” is the measure of the effort invested to produce software. The deployment of EV always starts with a chart of the EV components. Like the chart in the previous page, but now the XP components are integrated.

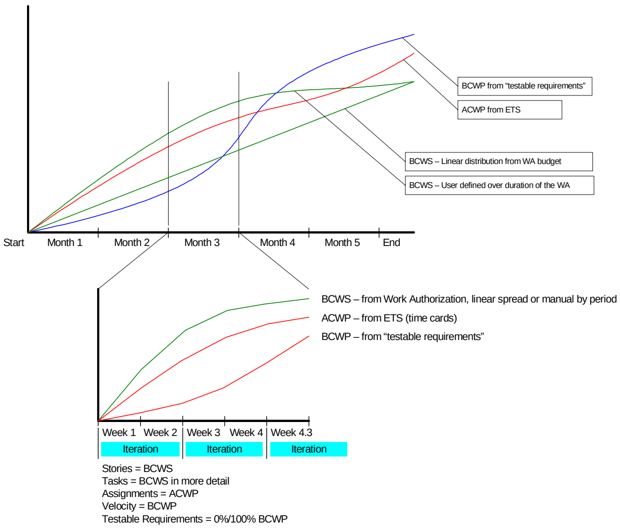


Figure 4 – replacing Velocity with earned value

For each iteration, BCWS and ACWP are acquired from the time card system. BCWP is defined through Stories, Tasks, and Testable Requirements. The testable Requirements are verified using Unit and Functional tests.

From the point of view of EVMS these processes are “normal,” with the exception of the “fine grained” deliverables. From the point of view of XP these processes are also “normal.”

An earned value management system is not a reporting system, contract administration, cost analysis, accounting, or a contractor's task management system. It is a measure of the value of physical progress in a project and as such adds additional effort to the work of managing a project. Beyond the additional effort of an EVMS, care must be taken to avoid hindering the project team’s ability to use its organic management systems.

With the Earned Value and Agile methods now outlined, let’s look at the similarities of each as ask *why can’t Agile methods be used in an EVMS environment*?

EVMS Methods	Agile Methods
Define the scope of work.	Scope defined in stories and tasks. Scope captured with 5 by 7 cards and held in a 3–ring binder by the project manager.
Develop an integrated bottom–up estimate for performing the scope of work.	Using stories, tasks, “velocity” estimates for completion and estimates at completion can be created.
Assign resources for each task in the plan.	Resources assigned during the bi–weekly planning session.

Measure the performance of these resources against the plan.	Use earned value as defined in EIA–748.
Measure the cost efficiency against the cost plan.	Use earned value as defined in EIA–748.
Forecast the final cost based on the current performance.	Use earned value as defined in EIA–748.
Manage the remaining work.	Use stories and tasks in the planning sessions.
Mange changes to the baseline.	Use stories and tasks in the planning session.

Table 1 – Comparison between EVMS and Agile Management

This table shows that many of the agile and EVMS processes share the same goal. It is likely though that each community has little understanding of the other's framework and motivations.

4.3 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.* [3]

5. Conclusion

Many would content that XP and government contracting Earned Value measurements based software development are like "gasoline and fire," never to be mixed. It turns out that Earned Value Management Systems are very similar to XP's velocity measurement. Using the activities in Table 1, an XP team can comply with EIA–748 planning, reporting, and cost/schedule management processes with ease.

We've created a development environment that performs many of the XP practices while maintaining our reporting deliverables for EIA–748 compliance. This involved:

- Replacing XP's velocity with Earned Value metrics.
- Creating fine–grained measures of BCWP using "testable requirements."
- Establishing the BCWS baseline at the beginning of each iteration.
- Capturing ACWP through a time keeping system.
- Computing Cost Variance, Schedule Variance from the three base 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. There are many other activities needed to fulfill the needs of a government contracting business, or any other business context.

Figure 5 shows how XP is positioned in this context…

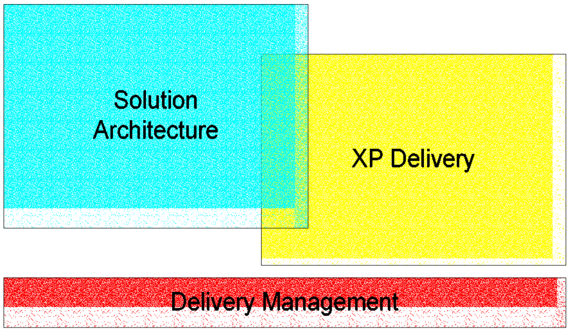


Figure 5 – Embedding XP in a Larger Context

In our environment we have embedded Extreme Programming in a large context. This context includes:

- *Solution Architecture* analyzes the business situation to accurately identify the core needs of the customer as well as the constraints imposed by the business environment.

- *Extreme Programming Delivery* provides the mechanisms to incrementally deliver value, address risks early in the development cycle, engage in continuous integration and test, and deploy the system to the customer in an incremental manner.

- Delivery Management is the key to success in the government contracting environment. Delivery Management provides:
  - Schedule Management through the creation and maintenance of the project schedule.
  - Budget and Financial Management through the creation and maintenance of the financial plan, the approved invoices for subcontractors and non-labor items, and the financial metrics. In our environment these are earned value metrics.
  - Scope Management through the creation and maintenance of updated schedules, architecture and system requirements.
  - Change Control Dispositioning and Integration manages change requests and the updated project baseline.

6. References

1. Abrahamsson, Pekka, Outi Salo, Jussi Ronkainen, and Juhani Warsta “Agile Software Development Methods: Review and Analysis,” *Proceeding of ESPOO 2002*.
2. Boehm, Barry and Kevin Sullivan, “Software Economics: A Roadmap,” *Proceedings on the Future of Software Engineering*, Limerick Ireland, 2000.
3. Fleming, Quentin and Joel M. Koppelman, *Earned Value Project Management 2nd Edition*, Project Management Institute, 2000.
4. Lett, Steve, “Earned Value Management for Self Directed Software Teams,” Software Engineering Process Group, Lockheed Martin.
5. Pisano, N. D., Commander, USN, “Technical Performance Measurement, Earned Value, and Risk Management: An Integrated Diagnostic Tool for Program Management.”
6. Solomon, Paul J., “Practical Software Measurement, Performance–Based Earned Value,” *Crosstalk*, September, 2002.
7. Wilson, P. B., “Testable Requirements – An Alternate Sizing Measure,” *The Journal of the Quality Assurance Institute* (October 1995):1
8. Wilson, P. B., “Sizing Software with Testable Requirements,” *Systems Development Management*, 34–10–04, Auerbach Publications, 2000

Glen B. Alleman is Vice President, Program Management, **CH2M** HILL. Prior to **CH2M** HILL, Glen was a member of a consulting firm specializing in ERP systems architecture and deployment.

Michael Henderson is Manager, Applications Development, **CH2M** HILL. Prior to **CH2M** HILL Michael was a Vice President of Development with Computer Associates.

Raymond Seggelke is Manager, Software Development for Environmental Technology Partners, a subcontractor to **CH2M** HILL. Prior to ETP, Ray was an engagement manager for BoldTech a Denver based Extreme Programming software development firm. Prior to Boldtech, Ray managed software development at Raytheon for TDRS.

CH2M HILL is a tier one contractor for the Rocky Flats Environmental Technology Site, Golden Colorado. Michael and Glen work in the Communications Group’s Information and Network Services department of **CH2M** HILL, providing applications and infrastructure support of the closure of Rocky Flats on or before December 15th, 2006.