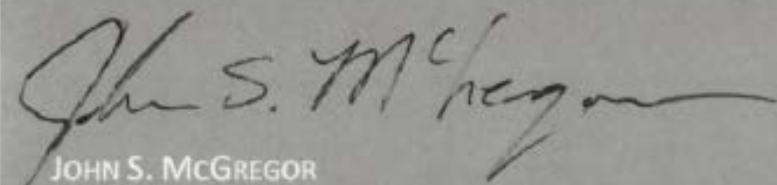


AGILE AND EARNED VALUE MANAGEMENT: A PROGRAM MANAGER'S DESK GUIDE



JOHN S. MCGREGOR

Deputy Director, Earned Value Management
Performance Assessments and Root Cause Analyses

OUSD AT&L (PARCA)
16 APRIL 2018



Table of Contents

Table of Contents	i
Foreword.....	1
Background	1
SECTION 1	3
Agile and EVM System Compliance	3
Organization and the WBS	3
Planning and Scheduling	5
Measuring Progress	7
Baseline Maintenance.....	8
Agile and Maintaining EVM System Compliance	10
Standard Terminology and Metrics	10
Agile Metrics and EVM Metrics.....	11
Traceability.....	11
SECTION 2	11
Integrated Baseline Review Guidance	11
Introduction	11
IBR Execution	13
IBR Preparation	14
PMB Assessment.....	16
Management Processes.....	18
Summary	20
SECTION 3	20
Agile Reports, Metrics and Analysis.....	20
Introduction	20
Delivered Functionality Metrics.....	20
Understanding Work in Process (WIP).....	24
Agile Metrics Related to EVM Metrics	26
Comparison of Agile and EVM Status Charts	27
Resources for Additional Information on Agile Metrics	29
APPENDIX.....	30
Example Agile and EVM Scenario	30
Agile Reference Terms	38

FIGURES

FIGURE 1. SW Development MIL-STD-881D Appendix K WBS breakout	4
FIGURE 2. Possible Agile SW Development MIL-STD-881D WBS breakout. Not prescriptive.	4
FIGURE 3. Features support the completion and delivery of Capabilities. In the product backlog, Capabilities directly relate to the Control Account level of the WBS.	5
FIGURE 4. A potential representation of an Agile product backlog hierarchy.....	6
FIGURE 5. “DoD Program Managers Guide to the Integrated Baseline Review Process”	12
FIGURE 6. Typical Agile Development Process.....	13
FIGURE 7. Product Increment Reviews follow a typical “time-boxed” planning process.....	14
FIGURE 8. Example Product Roadmap derived from NDIA “An Industry Practice Guide for Agile on Earned Value Management Programs, Version 1.1, dated March 31, 2017.....	17
FIGURE 9. Example Data Trace showing work decomposition and use of weighted stories to collect and provide Agile progress, at the Feature level, to the IMS.	18

FIGURE 10. Velocity Diagram for Two Different Teams.....	21
FIGURE 11. Increment Burn-Down Chart.....	22
FIGURE 12. Release Burn-Up.....	23
FIGURE 13. Capability Progress Charts (Derived from SAFe Metrics).....	24
FIGURE 14. Cumulative Flow Diagram Example.....	25
FIGURE 15. Example Data Trace showing work decomposition and use of weighted stories to collect and provide Agile progress, at the Feature level, to the IMS.	28
FIGURE 16. Cumulative Data and EAC Projections.....	28
FIGURE 17. EAC Projections EVM Gold Card Metrics with Agile-Informed BCWP.....	29

APPENDIX FIGURES

APP FIGURE 1. Shows alignment of EVM WBS elements to the Agile products hierarchy.....	31
APP FIGURE 2. Shows Feature Time-phasing using story points.	32
APP FIGURE 3. Shows how the Agile information can manifest itself as progress information in the IMS and BCWP for the control account.	33
APP FIGURE 4. Simple Scenario IMS.....	34
APP FIGURE 5. Simple Scenario Time Phased Budget.....	34
APP FIGURE 6. Agile Status by Sprint for this scenario	35
APP FIGURE 7. Calculation of BCWP by Sprint	36
APP FIGURE 8. Status IMS after Sprint 4	36
APP FIGURE 9. EVM Metrics by Sprint	37
APP FIGURE 10. EVM Metrics by Month	37

This document is intended as an informative resource for Department of Defense (DoD) personnel who encounter programs on which Agile philosophies and Earned Value Management (EVM) are applied. It is not an official policy, nor is it a step-by-step handbook for Agile implementation or the application of EVM.

Foreword

The DoD acquisition process supports the procurement and production of large, strategic, and deployable systems to address specific threats. Per the DoD Defense Acquisition System Directive 5000.01, an acquisition system is “a directed, funded effort that provides a new, improved, or continuing materiel, weapon or information system, or service capability in response to an approved need... [t]he primary objective...is to acquire quality products that satisfy user needs with measurable improvements to mission capability and operational support, in a timely manner, and at a fair and reasonable price.” Constantly evolving threats have presented a demand for an acquisition process that is able to respond quickly to emerging requirements and rapidly changing environments. To address this, the DoD has encouraged the following characteristics in acquisitions:

1. Flexibility: tailoring program strategies and oversight
2. Responsiveness: rapid integration of advanced technologies
3. Innovation: adapt practices that reduce cost and cycle time
4. Discipline: use of program baseline parameters as control objectives
5. Effective Management: decentralization to the extent practicable¹

These characteristics have led to an increased focus on flexible development approaches that include Agile philosophies and integrated program management tools such as Earned Value Management. With practice, integrating EVM and Agile has led to more success preparing for and managing the Integrated Baseline Review (IBR). By better understanding how the Agile methodology and the Earned Value Management program tool interact, programs are better able to measure status through Agile metrics and track success over time.

Background

Agile philosophies promote rapid incremental product deliveries, provide flexibility to respond to changing requirements, and advocate close customer collaboration. A major aspect of Agile is that changes to requirements, design details, or functional capabilities can be incorporated based on customer value, at any stage of the development cycle. While Agile is primarily used on software development projects, Agile methods are being used for complex system and hardware developments as well.

Agile for software development in the DoD is still an emerging product development approach. To be effective, the adoption of Agile methodologies must be integrated with existing DoD program management (PM) and system engineering (SE) processes. EVM is a disciplined integrated program management tool used to provide joint situational program awareness. EVM is used to measure technical progress against a baselined plan, independent of the consumption of resources. EVM is not tied to any specific development methodology and provides decision-makers with objective cost at completion forecasting as well as dollarized values of accomplishments and variances to the baseline plan. The requirement for EVM demands that performing contractors maintain an EVM System (EVMS) consistent with the 32 guidelines contained in the Electronic Industries Alliance Standard-748 (EIA-748) document. For DoD programs, EVMS implementations are evaluated in accordance with the DoD Earned Value Management System Interpretation Guide (EVMSIG)². As a result, a contractor’s EVMS provides the

¹ DoDI 5000.01, “The Defense Acquisition System,” May 2003

² Department of Defense Earned Value Management System Interpretation Guide, 2015.

structure to establish and maintain a credible Performance Measurement Baseline (PMB), which is the plan to accomplish a given contractual scope of work.

Agile and EVM are complementary when properly implemented together, and help enable a robust overall management process. In order to be effective, Agile must be evaluated for its applicability on a program-specific basis and tailored to best align with programmatic and contractual requirements.

Introduction

The origins of Agile Development can be traced back to 1957 to the incremental development of a large simulation by IBM's Service Bureau Corporation for Motorola.³ By the mid-1980s the DoD formally recognized the value of "adaptive software development" in the DoD's Military Standard for Software Development (DOD-STD-1679A). Throughout the 1990s, several other "lightweight" iterative development methods emerged including Dynamic Systems Development Method (DSDM), Scrum, and eXtreme Programming (XP). These methods, along with others, became collectively known as Agile methodologies. The tenets of Agile were codified with the creation of the Agile Manifesto in 2001. The Manifesto emphasizes the following major concepts:

Individuals and interactions	over	Processes and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

...while there is value in the items on the right, we value the items on the left more."⁴ For example, though the ability to respond to change is valued over following a plan, having and adhering to a plan remains essential for project completion.

In March of 2009, the Defense Science Board Task Force on DoD Policies and Procedures for the Acquisition of IT recommended that "The USD (AT&L) should lead an effort in conjunction with the Vice Chairman, Joint Chiefs of Staff, to develop new, streamlined, and agile capabilities (requirements) development and acquisition processes and associated policies for information technology programs."⁵ On October 28, 2009 Congress enacted the National Defense Authorization Act for Fiscal Year of 2010 which required the "Secretary of Defense [to] develop and implement a new acquisition process for [IT] systems."⁶ This included several principles of Agile development such as early and continual involvement of the user, multiple rapidly executed increments or releases of capability, early successive prototyping to support an evolutionary approach, and a modular open-systems approach. In 2013, the Deputy Assistant

³ C. Larman and V. R. Basili, "Iterative and Incremental Development: A Brief History," Computer, pp. 47-56, 11 June 2003

⁴ <http://www.agilemanifesto.org/>

⁵ Defense Science Board, "The Report of the Defense Science Board Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology," OUSD (AT&L), Washington, DC, 2009.

⁶ National Defense Authorization Act for Fiscal Year 2010, 2009.

Secretary of Defense for Systems Engineering stated in a briefing that DoD military systems must be “designed explicitly to have capacity to adapt and adjust to maintain relevance and operational advantage in an environment of change.”⁷

The origin of EVM can be traced back to the 1967 issuance of DoDI 7000.2 “Performance Measurement for Selected Acquisitions.”⁸ Since that time, EVM has been acknowledged by both Industry and the Federal Government as one of the most effective tools for planning, executing, maintaining, and reporting integrated cost, schedule, and technical status of an executing contract. EVM is a method for developing, baselining, and tracking a plan throughout execution and is based on pre-determined objective criteria. Although EVM is imposed as a contractual requirement, it does not mandate or prevent the use of other disciplined integrated program management methodologies.

In 2009, DoD established the Performance Assessments and Root Cause Analyses (PARCA), a directorate in the Office of the Assistant Secretary of Defense for Acquisition, to serve as the DoD focal point for all policy, guidance, and competency relating to EVM.⁹ In July of 2014, PARCA met with representatives from various DoD Services and Agencies to discuss the implementation of both EVM Agile development practices together on DoD programs. As a result, PARCA launched an initiative to explore the joint applicability of the two methodologies with stakeholders from the Office of the Secretary of Defense (OSD), the Services, the Intelligence Community (IC), Defense Acquisition University (DAU), and the National Defense University (NDU). This initiative included an action to examine issues, synergies, challenges, and best practices in the utilization of both Agile and EVM in industry.

SECTION 1

Agile and EVM System Compliance

The DoD EVMSIG, provides the overarching DoD interpretation of the 32 Guidelines where an EVMS requirement is applied. It serves as the authoritative source for EVMS interpretive guidance and is used as the basis for the DoD to assess EVMS compliance to the 32 Guidelines. The DoD EVMSIG provides the flexibility for contractors to utilize a disciplined development methodology. Agile, as a product development methodology, can exist within the disciplines required for EVMS compliance. However, certain considerations must be addressed in order for Agile and EVM to coexist.

Organization and the WBS

Within DoD, a Work Breakdown Structure (WBS) is an organized decomposition of a project’s work scope into manageable, product-oriented elements. The WBS facilitates communication between the government and the contractor and allows for the assignment of resources and subsequent tracking of progress. The DoD uses the MIL-STD-881D¹⁰ to pre-define the top-level program WBS, and provides for a common structure across product lines to allow for effective Cost and EVM reporting.¹¹ The primary purpose of the 881D WBS templates is to support standardized, historical data across similar program platforms.

- a. *WBS and 881D*: The WBS reported for EVM can align to a workflow-based waterfall

⁷ S. Welby, Deputy Assistant Secretary of Defense for Systems Engineering, “Thinking About Agile in DoD,” *AFEI Agile for Government Summit*, November 2013,

⁸ W. Abba, “How Earned Value Got to Prime Time: A Short Look Back and Glance Ahead,” *Measurable News*, p. 4, 1 March 2001.

⁹ <http://www.acq.osd.mil/parca/>

¹⁰ “DEPARTMENT OF DEFENSE STANDARD PRACTICE WORK BREAKDOWN STRUCTURES FOR DEFENSE MATERIEL ITEMS,” October 2011

¹¹ Cost and Software Data Reporting (CSDR) requires strict adherence to 881C (see the DoD 5000.04–M–1. Cost and Software Data. Reporting (CSDR) Manual. April 18, 2007)

development oriented hierarchy as found in MIL-STD-881D¹² (see **Figure 1**). However, an outcome-based Agile structure that focuses on customer driven deliverables (see **Figure 2**) is also acceptable. Both WBS hierarchies are product-based and supported by the DoD EVMSIG and MIL-STD- 881D.

1.1	Prime Mission Subsystem					
1.1.1		Computer Software Configuration Item A				
1.1.1.1		CSCI Requirements Analysis				
1.1.1.2		CSCI Design				
1.1.1.3		CSCI Code and Unit Test				
1.1.1.4		CSCI Integration and Test				
1.1.2		Computer Software Configuration Item B				
1.1.3	High level Integration, Assembly, Test, and Checkout					
1.1.4	...					

FIGURE 1. SW Development MIL-STD-881D Appendix K WBS breakout

1.1	Prime Mission Subsystem				
1.1.1		Capability A			
1.1.1.1		Feature A1			
1.1.1.2		Feature A2			
1.1.1.3		Feature A3			
1.1.1.4		Feature A4			
1.1.2		Capability B			
1.1.3	High level Integration, Assembly, Test, and Checkout				
1.1.4	...				

FIGURE 2. Possible Agile SW Development MIL-STD-881D WBS breakout. Not prescriptive.

- b. *The product backlog and the WBS:* Agile projects often utilize the product backlog to create the WBS. The product backlog contains 100% of the contract scope and is commonly defined as "...a list of features or technical tasks which the team maintains and which, at a given moment, are known to be necessary and sufficient to complete a project or a release."¹³ In the product backlog hierarchy, a capability¹⁴ directly relates to the Control Account level. To understand the relationship of the WBS to the backlog, the contractor should document the linkage of the WBS to the capabilities in both the backlog and EVM System Description.

¹² The waterfall model is a sequential design process in software development in which progress is seen as flowing steadily towards completion through the phases of design and development

¹³ Agile Alliance, "Guide to Agile Practices," 2015. [Online]. Available: <http://guide.agilealliance.org/guide/backlog.html>.

¹⁴ The term "capability" refers to a group of features that are traceable to the technical and operational requirements of the product being delivered

- c. *Requirements traceability*: As system requirements are decomposed into capabilities and features,¹⁵ the derivation should remain traceable and integrated with the contractor's proposed and extended Cost WBS, control accounts, work packages and planning packages, as applicable. Each hierarchical level of decomposition should have a set of clear and documented completion acceptance criteria to ensure that the basis of performance measurement in the EVMS is consistent and traceable.

Planning and Scheduling

The EVMSIG states that “The focus of...Planning, Scheduling, and Budgeting...is to develop plans and strategies to achieve the desired program cost, schedule, and technical objectives.” This focus includes how contractual work scope is decomposed and how the Integrated Master Schedule (IMS) is developed. The establishment of a time-phased Program Performance Measurement Baseline (PMB) is a critical aspect of EVM. The PMB is the foundation for integrating scope, schedule, and budgets into a plan against which accomplishments will be measured. The IMS establishes and maintains the relationship between technical achievement and progress status. EVM requires establishment of the baseline for all authorized work within the full period of performance, although detailed planning is only required for the near-term work.

The use of Agile tools or systems for this purpose is supported by the tailoring and system concepts allowed by EVM. In an Agile environment, a product roadmap¹⁶ generally is the basis from which a plan (IMS/PMB) is established.

Figure 3 provides a representation of the product backlog hierarchy for planning and scheduling. The capabilities and features define scope and have an assigned budget which is under baseline control.

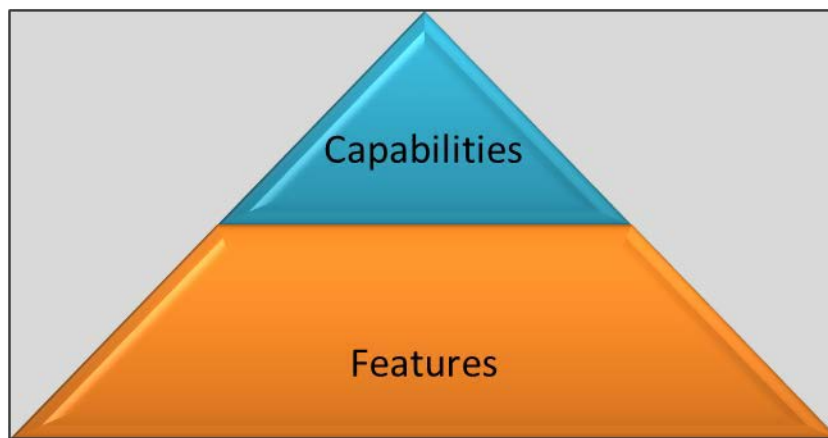


FIGURE 3. Features support the completion and delivery of Capabilities. In the product backlog, Capabilities directly relate to the Control Account level of the WBS.

In **Figure 4**, the stories¹⁷ (and any other hierarchical elements) below the dotted line describe the detailed means of accomplishing the scope of a feature (see section 2.b).

¹⁵ The term “feature” in Agile refers to a pre-determined functionality set that delivers value to the customer (see reference section for more information).

¹⁶ The product roadmap is a high-level plan for when capabilities/features will be accomplished over time (see <http://scaledagileframework.com/roadmap/>)

¹⁷ The term “story” is often used to refer to activities that contribute to the completion of a Feature

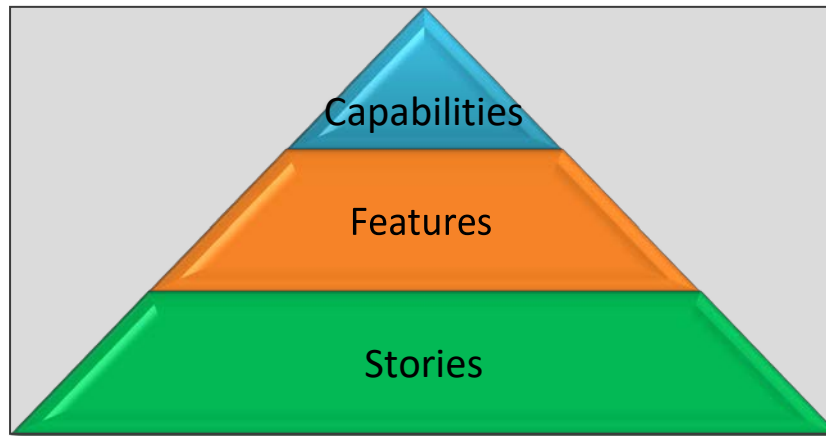


FIGURE 4. A potential representation of an Agile product backlog hierarchy

- a. *Decomposition of work scope:* In an Agile environment, contractual program requirements are broken down into capabilities, which in turn are further decomposed into features (features generally represent the point at which Agile execution and EVM progressing align). The product backlog also breaks down the feature into lower-level work items which are often referred to as stories. The completion of stories often contributes to the completion of a feature. It is a hierarchical relationship. The product roadmap identifies the sequence in which features and capabilities are completed and is the tool generally used as a mechanism to assist in the development of the baseline to accomplish the scope of work. The product roadmap supports and should be traceable to the work packages, planning packages, and summary level planning packages in the Program Performance Measurement Baseline (PMB)¹⁸, IMS, and the Control Account Plan (CAP).
- b. *Time-phasing of work:* EVM requires the time phasing of all work so that the Government and performing contractors have a common understanding of how the work is will be performed. The mechanism for developing the PMB may include the use of documented Agile techniques for plan creation. However, the PMB must capture all work scope and meet the intent of the EVMSIG. A typical Agile environment calls for a backlog of capabilities with an associated Agile roadmap that lays out the strategic delivery of the capabilities over time. Release timeframes are often determined from the roadmap and represent the time box to accomplish a given set of features¹⁹. Capabilities and features for future releases are placed in planning packages, per the contractor's EVM System Description. The government customer reviews the plan at an Integrated Baseline Review and during various events throughout the life of the contract.
- c. *Underpinning the IMS:* The EVMSIG allows for other tools, such as enterprise resource planning (ERP) or manufacturing resource planning (MRP) systems, to support higher level/summary tasks and milestones in an IMS. These MRP/ERP systems manage the finite details of labor assembly and material resourcing while balancing the timing of purchases to satisfy just-in-time principles. An Agile tool or system, in conjunction with

¹⁸ A PMB is a time-phased resourced plan against which the accomplishment of authorized work can be measured.

¹⁹ Agile planning and execution periods, often referred to as releases, do not necessarily correspond to the formal contractual delivery milestones. Though the term release has been used in the past to identify these delivery milestones, the delivery of functionality is now commonly referred to as a deployment.

the backlog, provides similar detailed task maintenance. Progress in the IMS can be summarized from status in the Agile management tools. The Agile process for statusing and forecasting, which supports IMS status updates, should be documented by the contractor in their EVM System Description. Note, the IMS must maintain a level of detail that provides the Government with actionable information and is sufficient to determine critical and/or driving paths through selected milestones.

- d. *Rolling wave planning*: EVM allows for an incremental planning process often referred to as rolling wave planning²⁰. The planning window is documented in the contractor's EVM System Description and the duration of the planning window is typically three to nine months. When rolling wave planning occurs, planning packages (and summary level planning packages) are detail-planned and the IMS is updated to reflect that detail planning. Agile release planning is similar to EVM rolling wave planning. The release cadence window is generally determined based on the nature of the work but is often three to six months in duration. During release planning, future capabilities are detail planned into features that can be accomplished during the subsequent release. The methods and timeframe for Agile planning activities should be documented in the EVM System Description and program procedures if applicable.
- e. *Freeze period*: The EVMSIG states that, "In order to solidify the PMB for accurate performance measurement, it is necessary to establish a freeze period. During the freeze period, changes to the PMB are limited to maintain its integrity. At a minimum, detail planning of planning packages must occur prior to the commencement of that work within the freeze period." The definition of a contractor's freeze period, including the mechanics and rules for controlling baseline changes during that timeframe should be documented in the contractor's EVM System Description.

Measuring Progress

Work scope should have clearly defined, objective, technical completion criteria that is documented prior to starting any specific effort. This is true regardless of the use of EVM and/or Agile. The criteria should be established either before or during the detail planning/release planning process at the appropriate level of the WBS.

- a. *Progress tied to scope completion*: In Agile, progress of a capability is based on the technical completion of each of its features, which in turn is based on the accomplishment of the feature's acceptance criteria. It is imperative that progress is tied to the completion of scope (technical progress) and not the completion of time boxed events such as sprints. As with EVM, the mechanism/technique used for taking performance against completion of a specific scope of work must be documented. To claim work as completed, the Agile system must support the EVM system in demonstrating that all of the objective technical completion criteria for that work has been met. Agile procedural documentation, particularly addressing processes that occur below the feature level, is an essential part of providing the trace that shows how Agile processes support or generate the EVM performance status reported to the Government program office.
- b. *Claiming performance*: EVM guidelines emphasize the use of appropriate performance measurement techniques based on the nature of the work. The EVMSIG states that the contractor must have information (Quantifiable Backup Data, or QBD) that supports the EV performance claims for each work package/control account. Similarly, Agile programs

²⁰ The continuous process of converting planning packages into control accounts and control account planning packages into work packages.

utilize QBD to substantiate performance claims. Stories are often assigned value based on size, complexity and/or risk. These values become the necessary underpinning QBD for claiming performance. The usage of stories to measure progress must be disciplined and consistent while following certain guidelines: all stories reflect technical accomplishment towards a feature; once established, story point values do not change; stories can be added or removed from the QBD through the development process to support technical completion of a feature. The process by which stories are used to in conjunction with the selected EVT must be documented and must not conflict with the contractor's EVM System Description.

EVM measures progress against the detailed planned activities for a given reporting period (i.e. accounting month). In Agile, features often span several months and the measure of progress is relative to the technical completion of a feature and not to the completion of a reporting period.

- c. *Agile and dollarization*: The contractor should define, within its EVM System Description, the level at which dollarization occurs in the system. This is true whether or not Agile is implemented and should be testable during routine surveillance. The relationship between Agile performance metrics and EVM status should be understood by all stakeholders.
- d. *Forecasting and Estimates at Complete*: The EVMSIG states that programs will maintain an Estimate at Complete (EAC) by developing "revised estimates of cost at complete based on performance to date ... and estimates of future conditions." This periodic reassessment of remaining requirements contributes to program success for both the Government and the contractor. The development of an EAC on a program that employs Agile philosophies is similar to the process required for traditional software programs and projects. The remaining work is analyzed to provide an assessment of the effort required to deliver the remaining capabilities and features.

The metrics generated from an Agile tool are typically used to establish a forecast Estimate to Complete (ETC). An Agile forecast reflects the attributes of a properly maintained EAC because it is continuously adjusted to reflect program progress. It enhances management value by tying projected costs for the remaining work to credible sources and ensures any decision regarding the allocation of future resources is based on valid data. Although it is not explicitly stated, the use of Agile tools for the estimation process is supported by the EVMSIG. The methods for the identification of forecasts and estimates between Agile tools and the EVMS should be documented in the EVM System Description.

Baseline Maintenance

- a. *Maintaining the backlog*: Backlog maintenance is critical to the effective management of an Agile program. It is a best practice to review the backlog and product roadmap during an Integrated Baseline Review (IBR) and periodically with the Government customer throughout the duration of the contract. It is imperative that the Government remain actively involved in the release planning process because of its potential to affect the PMB. Changes to work scope must follow the established rules for work authorization, baseline management, and change control as described in the EVMSIG. Items within the Agile product backlog, at the feature level (i.e., work package) or higher, have an assigned budget under baseline control. Removal or addition of any feature-level or higher item

from or to the backlog should be documented and performed in accordance with system baseline change processes.

b. *Scenarios for consideration:*

- i. *Work toward the completion of a particular feature is ongoing and is scheduled to finish at the end of the current sprint. Once the sprint ends, one of the planned stories for that feature has not been completed and the feature has not met its completion criteria.*

The team forecasts the story to complete in a future sprint. The result is the work package would have a negative schedule variance because the scope of work (feature) associated with the work package did not complete within the baselined period (assuming no other variances are affecting the work package).

- ii. *A feature will be moved from the current release to a subsequent release.*

A feature is part of the baseline and therefore if it changes, it must adhere to the baseline change control process. Baseline change could be processed in accordance with the contractor's approved change control process, taking into consideration whether the change is contractual or an internal management decision. If work has already begun, then the work package should be shut down by the generally accepted practice of setting BCWS to BWCP, and replan the remaining work package budget (BCWS) in the subsequent release identified.

- iii. *The team will need to complete additional stories in order to meet the completion criteria of a particular feature (i.e. there is no contractual scope change).*

If the additional stories are still consistent with the acceptance criteria of the feature, and simply provide greater granularity to how work will be performed, then no re-baselining/replanning is required. The originally planned scope (at the feature/work package level) is unchanged and the likely result will be a negative schedule variance as the work would be considered more complex than originally thought. The amount of performance claimed would remain the same, but the percent complete would change with an increase in the amount of effort required to complete the same scope of the feature.

- iv. *The team will be able to meet the objective technical completion criteria for a particular feature without having to complete all the planned stories.*

This discovered efficiency results in no change to the work scope, as the work would be seen as being less complex than originally thought. The likely result would be a positive schedule variance during the completion of the work scope and a positive float for the IMS upon completion. Also, with the associated hours not expended, a positive cost variance may occur.

- v. *A need for a new feature is identified and it must be completed in order to satisfy the completion criteria for a particular capability. It is an addition to the baselined work scope.*

If the added feature is not within the scope of the contract, the contractor shall receive contractual direction, and the associated contract modification with contract value and budget from the customer to begin work on the new feature. The new scope adds to the total amount of budget and performance that can be claimed, but does not affect the performance taken on already completed work. The Control Account where the feature was added and the overall contract percent complete value will decrease due to the added scope, but the amount of performance previously claimed does not change.

If the added feature is within the scope of the contract but not in the scope of the Control Account, then the contractor should follow their EVM System Description for newly identified in-scope effort, and likely distribute Management Reserve to plan and budget the resources for the new Feature. Again, the Control Account and overall contract percent complete value will decrease with the added effort, but this does not affect the amount of performance previously claimed.

- vi. *The team was able to meet planned objective technical completion criteria for all planned Features in the release and have additional capacity to perform more work given their observed productivity.*

A baseline change is required to allow the team to pull work out of the product backlog (in a planning package, not yet detail planned). The contractor would initiate a Baseline Change Request (BCR) and convert the planning package into a work package in accordance with their EVM System Description. Work pulled from the product backlog would be effort the team believed could be completed under existing constraints for upcoming releases and program milestones.

If the work to be performed as a result of the additional capacity is already in the baseline plan as a work package, a BCR would not be required and the portion of that work package completed in the current Release would result in a positive schedule variance.

Agile and Maintaining EVM System Compliance

The Integrated Program Management Report (IPMR) Data Item Description (DID)²¹ contains specific requirements for EVM data reporting to the Government. The IPMR allows tailoring to meet the needs of the Government program office. To enhance customer collaboration, consideration should be given for the buyer to have access to the supplier's Agile tools in the interest of transparency.

Standard Terminology and Metrics

Currently, there is no DoD standard terminology for Agile processes and artifacts. It is imperative for the government PMO and a contractor to define the Agile terms and process at the beginning of negotiations for a contractual scope of work. The EVMSIG and the IPMR, along with other DoD and Service/Agency policies, define EVM requirements and are included in RFPs where applicable. The performing contractor should provide information to the PMO regarding how Agile processes are incorporated in the implementation of their EVMS.

²¹ See http://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=278901

Agile Metrics and EVM Metrics

As with the discussion on definitions, there are no DoD standards for Agile methodology metrics. Metrics such as velocity, burn-down/burn-up, etc. must be agreed upon by the performing contractor and the government PMO. Agile metrics may be included in the IPMR Format 5 as supporting documentation for the status of work performed, but should not supplant the typical EVM metrics. As with the link between EVM status and technical performance, Agile metrics should provide status that supports and aligns with similar EVM metrics. Section 3 of this document discusses how a variety of Agile metrics can be used to show progress against product and business objectives; that metrics can be at the capability, feature, story, or story point level, and tracked by increment or sprint, as long as the measure is consistent and meaningful to the program.

Traceability

Generally, Agile processes influence segments of work below the reporting level for the IPMR. Performance status at the level where Agile execution occurs should underpin the performance information at the level where EVM is reported (usually through the features). Since they are typically the criteria for performance, if the stories and features for a given scope of work experience favorable performance, both the Agile metrics and the EVM metrics should reflect favorable performance. The same is true for unfavorable performance as well. Format 5 variance analysis is required at the IPMR reporting level; the performing contractor should provide information from the Agile system to help support variance explanations.

SECTION 2

Integrated Baseline Review Guidance

Introduction

The Department of Defense (DoD) maintains a “Program Managers’ Guide to the Integrated Baseline Review (IBR) Process”²² to improve the consistency of the IBR execution across the Department. The IBR is defined in DoD as a process to ensure the contractor’s plan covers the entire technical scope of the work under contract, that the work is realistically and accurately scheduled, the risks are captured, and a logical mix of resources are assigned. See Figure 5 below for an overview of the IBR process.

²² <https://www.acq.osd.mil/evm/docs/Program%20Managers'%20Guide%20to%20the%20Integrated%20Baseline%20Review%20Process.pdf>

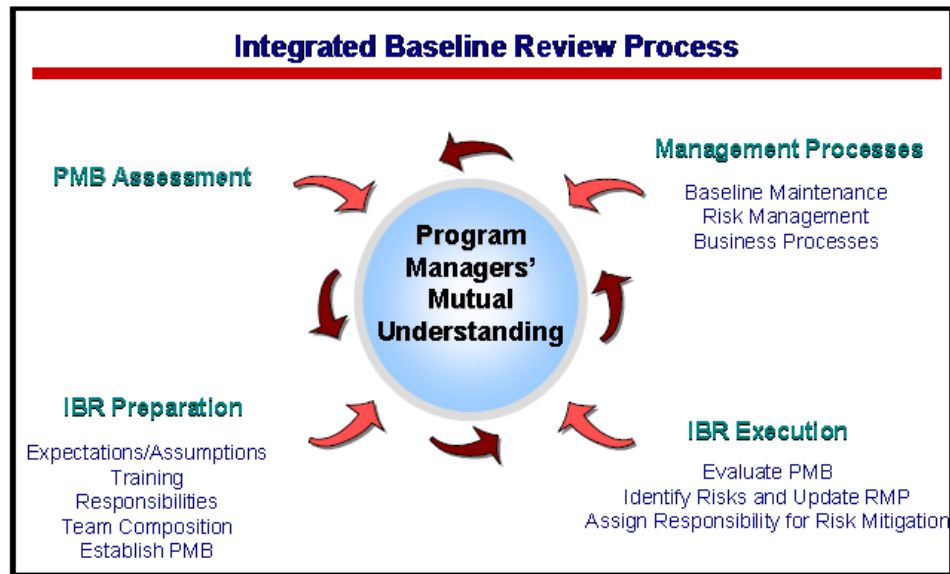


FIGURE 5. "DoD Program Managers Guide to the Integrated Baseline Review Process"

The DoD IBR guide helps Program Managers (PMs) understand the purpose, benefits, key elements, processes, and key attributes of an effective IBR process. Agile programs also benefit from the IBR process and this chapter will provide guidance and considerations for conducting IBRs in on programs using Agile development.

In the fall of 2017, PARCA conducted a survey across the DoD constituents to understand if and how IBRs are being conducted on Agile development programs. The survey completed in December 2017 and concluded that the IBR process and purpose does not change, but there are unique considerations that should be made to address Agile development. These unique considerations for conducting an IBR on

Agile Programs are described below:

Per Figure 5, the IBR process can be categorized in four major areas. Each area has specific additions or changes in an Agile environment:

- a. *IBR Execution*: The IBR execution should be consistent with the nature of the Agile planning processes.
- b. *IBR Preparation*: The IBR preparation and training will include Agile development familiarization training to ensure all parties have a mutual understanding of the Agile processes being used on the program.
- c. *PMB Assessment*: The Performance Management Baseline (PMB) assessment portion of the process will be reviewing additional artifacts and linkages between the Agile technical execution process and the EVM performance management system.
- d. *Management Processes*: The evaluation of the management processes will need to include a review of the Agile development processes and metrics that will support program decision making.

Each of these four areas of consideration are explained in the following sections.

IBR Execution

IBR Practice

The initial IBR occurs within the first 180-days of the contract award at any point during the life of the program. When significant changes to the planning or contract occur, an IBR may also take place. During the project execution program objectives (accomplishments, progress, and milestone completions) are addressed through monthly program management reviews and data item reviews. The IBR process continues with periodic review of the baseline during contract or engineering changes or planning events (major Engineering Change Proposal (ECP) or a formal baseline re-planning effort to address an Over Target Baseline (OTB)).

Agile Considerations

Agile methods focus on delivering demonstrable progress quickly to obtain customer feedback. Agile planning, execution, and feedback cycles occur in time-boxed increments. After the initial IBR during project execution alignment of program objectives and risk can be done during the product increment planning process.

When a contractor is using Agile development, increment planning sessions occur frequently, at set “time-boxed periods” to align with the Agile planning and execution process. There are a variety of Agile methods used, all of which have a delivery cadence much quicker than traditional methods. The length of the detailed planning period should be the same, for example, recurring increments every three months.

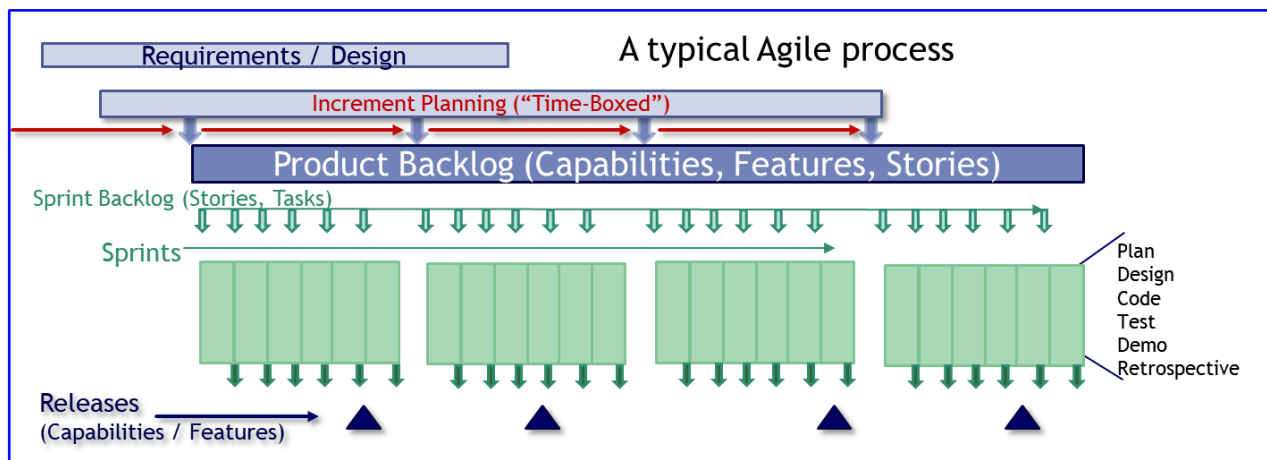


FIGURE 6. Typical Agile Development Process

Figure 6 shows a typical Agile process that has a continuous planning cadence to develop and groom the product backlog, scope of work to be done, prioritize and organize work into product releases. Each product release is defined by a fixed set of capabilities and features and is delivered when all the capabilities have been included. The team plans and implements the features into increments using a time-boxed rolling wave process. Each rolling wave increment is comprised of a fixed set of implementation time-boxes called sprints.

In Agile, as part of the Product Increment (PI) planning session, a review of work progress, work remaining, work priorities, risks, available staffing (teams), and project progress is performed to agree on the next Increment of work in the context of the overall program objectives, progress and risks. This includes decomposing capabilities into features to be completed during the increment, establishing

increment objectives, and decomposing the backlog to “sprint-able items”, typically referred to as stories. The sprint teams then implement the work through the sprint planning, execution, review and retrospective process.

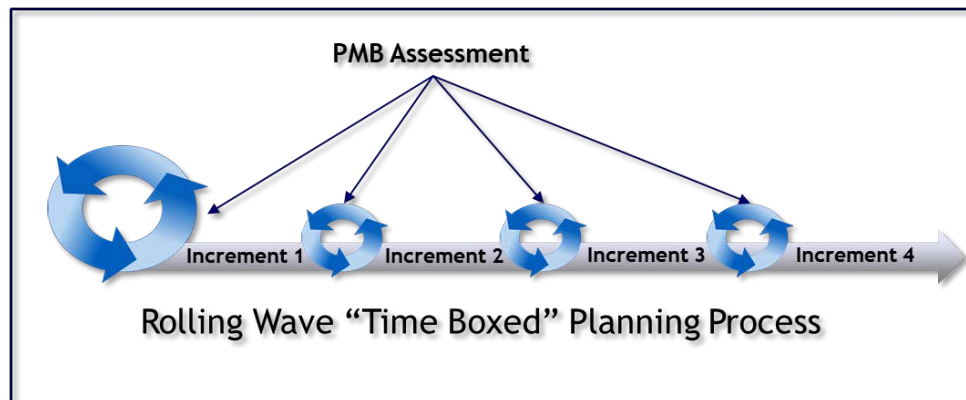


FIGURE 7. Product Increment Reviews follow a typical “time-boxed” planning process.

Figure 7 shows a continuous planning and execution process where the PMB is assessed periodically to verify IBR objectives as the plan is detailed for each of the rolling wave planning windows. In Agile these planning events are referred to as Product Increment (PI) reviews. The PI reviews are typically performed at the end of the planning events with full-time participation from the development team, the product owners, the customer, and the stakeholders.

IBR Preparation

Although there will be some preparation for each of the increment planning events, we separate the initial preparation to highlight the need to ensure the Government and the contractor have a common understanding of the Agile process in the context of conducting an IBR.

IBR Practice

Contractors begin detailed planning upon contract award. Typically, this planning is a refinement of the proposed plan provided during source selection. This is the time for Government Program Management Offices (PMOs) to engage in order to understand the assumptions and methods the contractor will be using to execute the entire program. This is particularly important when a contractor plans on using Agile development methods.

Agile Considerations

The most important element of the IBR preparation is for the Government and the contractor to jointly understand the Agile methods and management processes that will be used to manage the program. A best practice is for the Government to obtain Agile familiarization training from the contractor on the specific development methods the contractor will be using. The training should address at least the following kinds of topics:

- a. *Specific Agile Methods*
 - i. Overview of Agile and benefit to the program.

- ii. Description of the specific Agile methods being used (SCRUM, KANBAN, SAFe, and other Agile tools)
 - iii. Description of the Agile roles and responsibilities. Agile process requires Government resources particularly, the functional or operations personnel to participate.
 - iv. Agile cost estimation process and how it relates to establishing the time-phased budget.
 - v. The business cadence, for example, quarterly releases, 2-week sprints, daily standups.
- b. *Communication Plan*
- A communication plan should be developed that documents the Government's unique roles/responsibilities, interactions (events, meetings, etc.), and artifacts expected in the Agile process. Typical events in the Agile process include:
- i. Increment Planning
 - ii. Sprints
 - iii. Demonstration Events

The communications plan should identify what information the contractor will share from the Agile collaboration environment. In some cases, the government could get direct access to the Agile tools and in other cases periodic informal status reports from the Agile tools may suffice.

The communications plan should document the specific Agile roles that will be used to manage the program, and how these roles relate to the PM and Control Account Manager (CAM) roles. Typical Agile roles are identified below, not all of which would be required on every contract effort:

- a. Product Owner – Responsible for communicating, managing, and refining the backlog for Sprint teams. The Product Owner must have domain expertise to articulate backlog items clearly to the development team. The Product Owner has a very active Agile role with sometimes daily interaction with the team to address questions or issues.
- b. Development Team – Responsible for implementing the work identified in the backlog.
- c. Scrum Master – Responsible for ensuring the team, product owners, and stakeholders follow the pre-defined process. Responsible for working with the team to remove barriers.
- d. Stakeholder – Anyone who will be directly affected by the product.
- e. System Architect – Technical lead for documenting the technical vision and architectural decisions.
- f. Release Engineer – Technical person responsible for facilitating the product definition to meet the release objectives.
- g. Customer – Responsible for funding and accepting the product.

The Government typically assumes one or more of the following roles:

- Product Owner – The Government sometimes assumes the role of the Product Owner. The Product Owner, whether Government or Industry, should be involved in the release planning, product backlog definition and grooming, increment and sprint planning,

product acceptance and sprint review. The Government's role would most likely be the technical counterpart to the contractor's CAM.

- Stakeholder – In most cases the Government assumes the role of Stakeholder. As the Stakeholder, the Government is involved in release planning, and could also be involved in Increment Planning to review and approve the next phase of work. There would typically be multiple Government stakeholders including the end user. It is the responsibility of the Program Management Office (PMO) to communicate with all internal and external stakeholders.
- Customer – This role is typically the contracting program office.

PMB Assessment

IBR Practice

A key component of the IBR Process in assessing the baseline is to understand the integration of the scope, cost, and schedule. Some of the typical artifacts reviewed during an IBR include the Statement of Work (SOW), the Integrated Master Plan (IMP), the program and technical objectives, the time-phased budget, the Integrated Master Schedule (IMS), resources, the Control Account Plan (CAP) and other pertinent artifacts.

Agile Considerations

The objective for performing the PMB assessment on Agile programs is the same. The artifacts used to understand the integration of the cost, schedule and technical products and processes are different. Two important artifacts that will likely be different are the Product Roadmap and the Product Backlog. The Product Roadmap documents the strategic time-phased delivery objectives of the program, and the Product Backlog documents the work to be accomplished. These artifacts are described in more detail below. Agile programs should still have an IMS and a time-phased budget.

Product Roadmap: The Product Roadmap will identify the project delivery cadence, key decision and/or delivery milestones, and system or software (SW) releases which will provides a foundation for the creation of the Integrated Master Schedule (IMS). The Product Roadmap should capture, at a high level, all envisioned work scope.

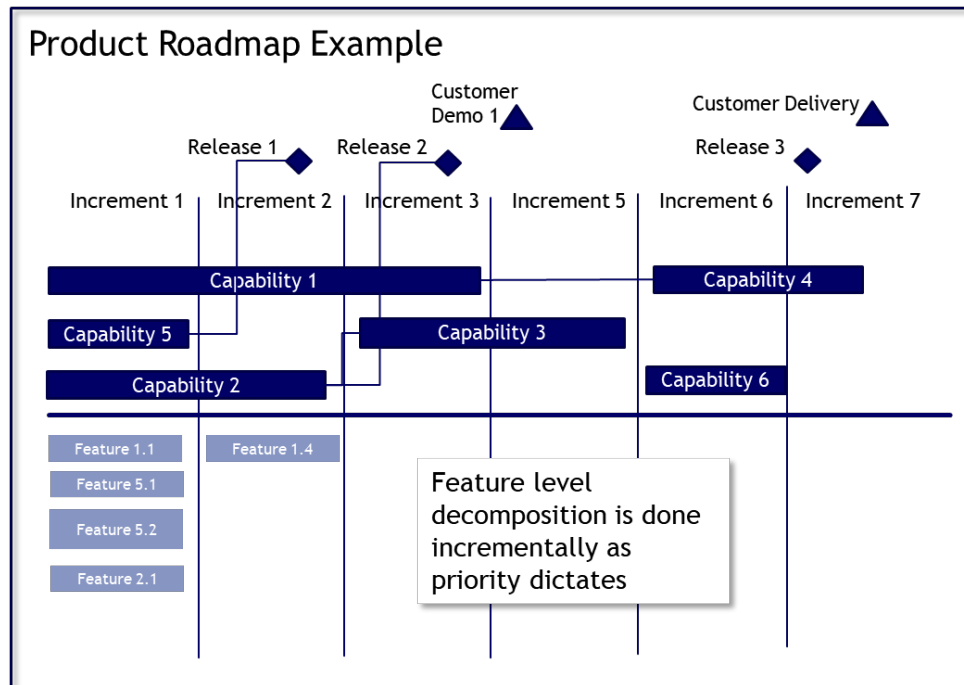


FIGURE 8. Example Product Roadmap derived from NDIA “An Industry Practice Guide for Agile on Earned Value Management Programs, Version 1.1, dated March 31, 2017

Product Roadmaps can be high level such as shown in Figure 8, or they could be a more detailed technical document that may look more like the detail you would find in an IMP.

Product Backlog: The Product Backlog will document the time-phased execution of work based on priority and support of the Roadmap objectives and milestones. The Product Backlog will be a comprehensive description of the work to be done on the program and is usually structured by listing the capabilities, features, and stories that represent all the envisioned work. The Product Backlog will be decomposed in detail for the items to be done in the first sprint and less detailed for the items in future sprints. The Product Backlog typically resides in the Agile development tool and is the basis for reviewing technical scope. The Product Backlog structure and relationship to the EVM artifacts such as the SOW, WBS, and the IMS must be clear. All scope from the SOW must be traceable back to the baseline product backlog. The flow of planning and execution status between these items must be clearly understood.

Considerations while reviewing the technical baseline include:

- Understand how the SOW, in the contract, relates to the product decomposition and structure of product backlog.
- Understand the mapping of the Product Backlog items to the WBS. Note that although stories can be shown in WBS format, stories are not part of the WBS but are implementation steps to develop products identified in the WBS, such as capabilities and features.
- Understand at what level within the Product Backlog is the EVM control account, work package, and planning package identified.
- Understand what methods will be used for collecting and rolling up progress to the IMS and then to the EVM System. See Figure 9.

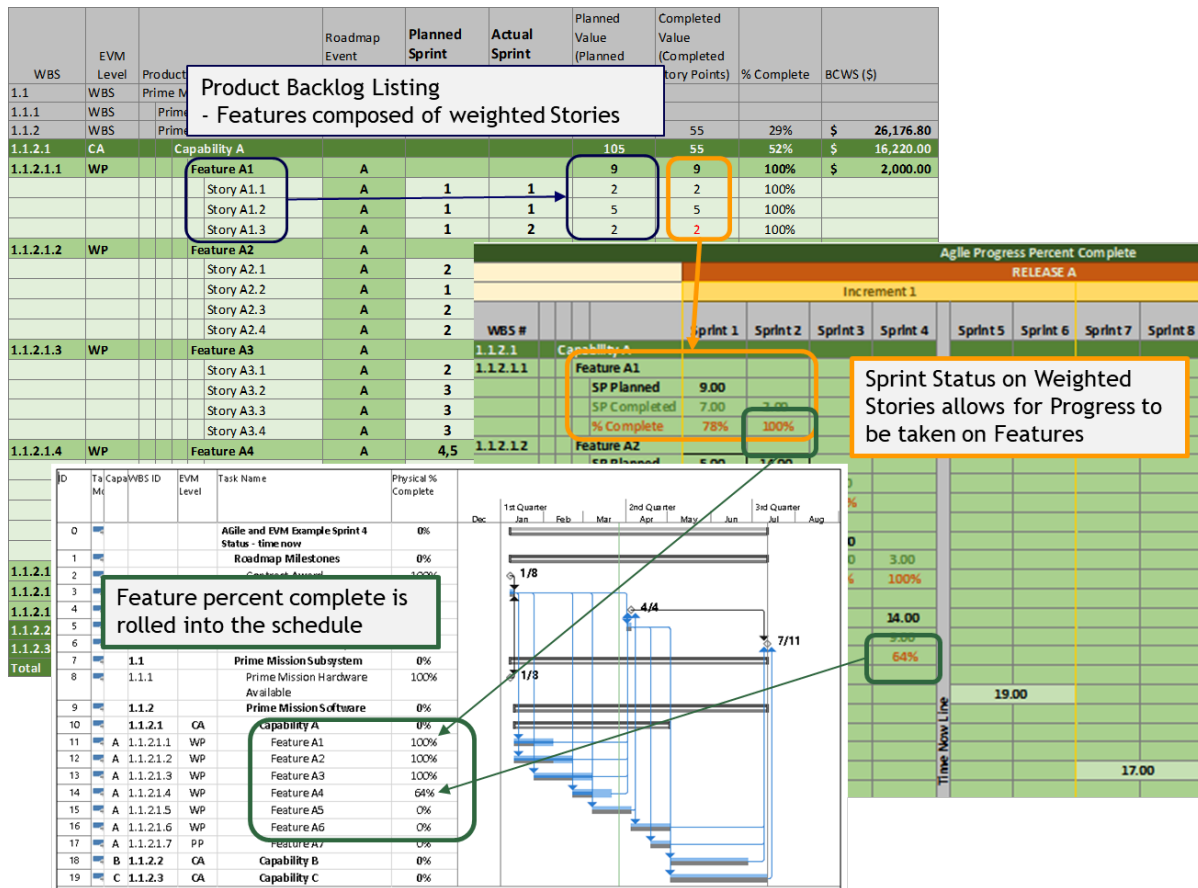


FIGURE 9. Example Data Trace showing work decomposition and use of weighted stories to collect and provide Agile progress, at the Feature level, to the IMS.

- Understand what level of the work being done in the Product Backlog flows into the IMS. The IMS includes the backlog items that show end-to-end dependencies to see an end-to-end critical path to program deliverables.
- Identify which Agile metrics track progress, help the contractor manage their resources, and the achievement of program level milestones.

Management Processes

IBRs on Agile contracts can benefit by leveraging the contractor's Agile planning cycle in conjunction with the natural collaboration that occurs between the Government and industry in program execution.

IBR Practice

- Baseline Maintenance Process**
Agile considerations for baseline maintenance includes reviewing the PMB on a regular basis. This process is in line with Business Processes from the PM IBR Guide, in insuring that the PMB accommodates changes in a consistent manner.
- Risk Management Process**
In each Agile planning cycle, risks are decomposed as the plan is made and entered and tracked in the Risk Register. This should be consistent with IBR Plan, the Program Management Plan, and the Program Risk Plan.

c. Management Processes

The following guidance is recommended for managing an Agile development effort: IBR Plan, Communication Plan, Earned Value System Description, Program Management Plan, and the Program Risk Plan. Each of these documents contains the process for managing, planning, executing, reporting performance and risk, and communicating in an Agile environment.

Agile Considerations

Specific management process considerations include:

- How does the contract create a unique approach for integrating project management practices with Agile methods?
- What is the Agile planning process for decomposing Statement of Work (SOW), technical, and contract requirements into Agile terms and product definition?
- What is the structure of the WBS (Product Backlog) and how do the EVM structures (CA, WP, PP, Summary PP) overlay?
- Synchronization of the EV Work Breakdown Structure (WBS) hierarchy with the Agile product hierarchy is one of the most critical aspects of implementing a combined EVM/Agile methodology and is key in understanding how these methodologies must work together.
- How will the work be prioritized and documented in the product roadmap and how does that flow into the product backlog as part of the planning process?
- Does the IMS maintain a level of detail that provides the Government with actionable information and is it sufficient to determine critical and/or driving paths through selected milestones?
- How will the authorized work be time-phased into the IMS to show alignment with the product roadmap and show end-to-end dependencies to see an end-to-end critical path?
- What are the methods that will be used to define the objective “definition of done” and to collect and roll up progress into the IMS and EVM cost reports?
- How does the EVMS “freeze period” align with the program’s Agile cadence for release planning and execution?
- Does the IMS accurately model the time phased authorized work represented in the product backlog?
- How is the Agile Management tool used to support PMB change control?
- How are Agile burn-up/down plans, velocity, and required remaining velocity related to the EVMS managerial analysis of related information?
- How is the ETC developed relative to the remaining work in the product backlog?
- How is subcontracted work being incorporated into the IMS and the baseline as well as Agile reporting?

Summary

The Integrated Baseline Review is a critical process that allows the Government and contractor to establish and maintain a mutual understanding of the PMB and program risk. As part of the IBR process it is important for the Government to understand the contractor's technical and management processes, roles and responsibilities, and communication artifacts and cadence. When contractors use Agile methods to perform their development it becomes one of the technical processes that must be understood as part of executing an IBR. Since Agile benefits from significant Government program office collaboration, it is particularly important for the Government to understand the details of the contractor's Agile process and how it underpins the reporting of progress through the EVM System.

SECTION 3

Agile Reports, Metrics and Analysis

Agile methodologies include metrics to track team, project, and program progress in support of continuous process and product improvement. This chapter will not address the traditional system or software metrics, such as technical performance measures, open defects, and effective lines of code, but instead will discuss specific Agile metrics and how they provide the underlying objective data to support EVM metrics. This chapter is intended to provide an overview of common metrics that programs may (or may not) use depending on the Agile methods used for program execution.

Introduction

Earned Value Management (EVM) provides effective and accurate performance data and metrics essential to Government contract oversight and program management decision making. Agile development has been gaining traction as a development methodology within Government contracting by enabling incremental and continuous product development and delivery. Agile methods offer disciplined planning, execution, and delivery processes that provide quantifiable data that aligns to the EVM requirement of objectively tracking technical progress.

Agile and EVM are complementary when properly implemented together. To those who have not practiced Agile, the methodology is perceived to have a lack of structured end-to-end program planning discipline and an inability to accurately forecast cost and schedule. But in fact, the rigor and attention to detail behind Agile development provides the objective evidence on technical progress that correlates to EVM metrics and BCWP (budgeted cost of work performed) reporting. In this chapter, the focus will be on metrics that not only support the Agile development methods, but also correlate with EVM metrics and program status.

Delivered Functionality Metrics

A variety of Agile metrics can be used to show progress against product and business objectives. It is important to note that there is no single metric that will answer all the questions related to program performance. Each metric has a specific purpose and all metrics need to be evaluated to get a picture of overall program status. Metrics can be at the capability, feature, story, or story point level, and tracked by increment or sprint, as long as the measure is consistent and meaningful to the program.

- a. *Velocity*: A team productivity measure of how much working product (i.e., tested software code) is delivered in each sprint. Average velocity is the number of story points completed per sprint and can be used to estimate a team's future efficiency to complete work. Since story points are defined within a team, velocity is used by team to measure, track, and

improve productivity and velocity over time.

Figure 10 below shows velocity over time for two different teams. The first diagram indicates a team that is gradually increasing its delivery of product over time. The second diagram indicates a team that is delivering an inconsistent amount of product in each sprint.

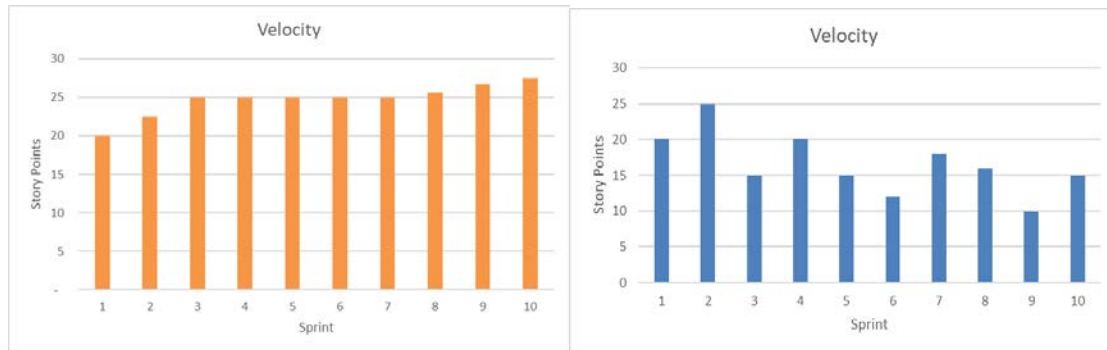


FIGURE 10. Velocity Diagram for Two Different Teams

The cause for inconsistent delivery could include a variety of reasons such as team dynamics, changing requirements, external impediments, etc. Program management leadership would want to understand the root causes and act to help the team improve. The Product Owner and ScrumMaster should work to troubleshoot and resolve any issues causing the inconsistent velocity.

Velocity is a team-specific metric and should not be used to compare one team against another. It is possible, however, to aggregate a total velocity across all Scrum teams to aid in forecasting at a total control account or program level (provided the aggregated Scrum teams are consistent over time).

- b. *Product Burn-Down*: The most common overall Agile metric is the product burn-down chart. This chart can be used to track progress of features, user stories, or story points completed versus points planned over time. Figure 11 is an example of a story point burn-down chart, which depicts the number of story points planned and completed by sprint for the current Increment.

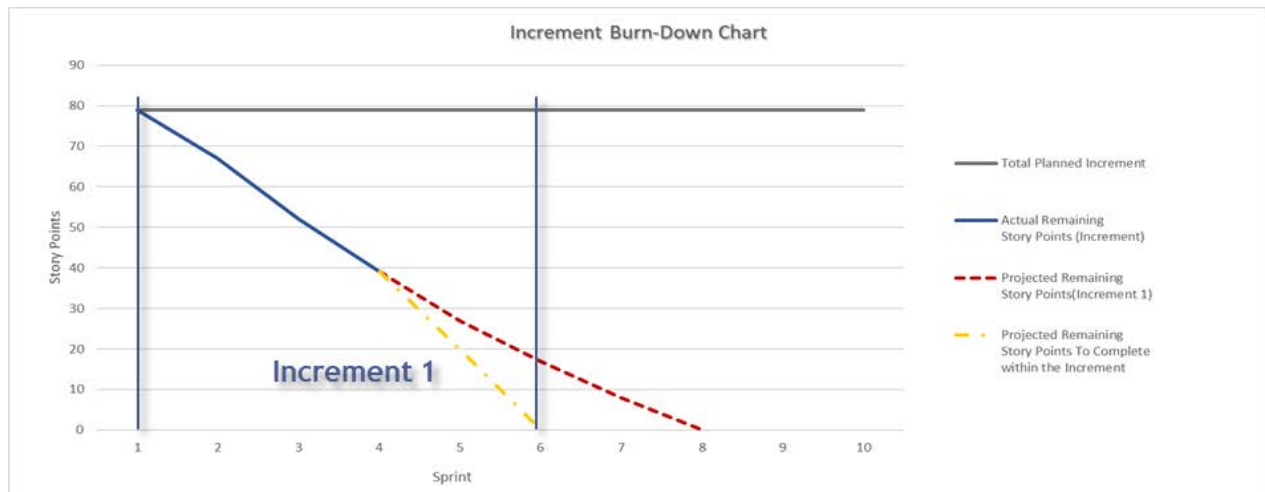


FIGURE 11. Increment Burn-Down Chart

This chart provides a depiction of the work remaining within the increment. The gray line across the top is the total cumulative Story points in the Increment. The blue line indicates the actual Story points remaining in the Increment and is an indicator of current efficiency. The yellow and the red dotted lines are used to forecast status at the end of the increment:

- i. **Optimistic:** The Projected Remaining Story Points line (yellow dotted line) indicates an increased level of efficiency (velocity), shown by the slope of the line, would be required to accomplish all Story points by the end of the Increment.
- ii. **Most Likely:** The Forecasted Features line (red dashed line) assumes remaining work to complete at the original planned velocity. Since Increments are time-boxed this line indicates that not all planned work will be accomplished within the Increment.

An Increment burn-down chart shows an overall view of progress, similar to a level 1 EVM WBS graph, but does not show progress on specific scope. To see the status in terms of scope (feature, capability, or release completion), progress is measured at lower levels (i.e. user stories) and rolled up to the capability/feature level.

Burn-down charts at the backlog level will show the aggregate of the total program and can be used to project the rate of completion of remaining work at a program level. Burn-down charts can also be developed to show the status of a specific capability.

- c. **Product Burn-Up:** Another common metric is a Product Burn-Up chart, which depicts the status and rate of progress (velocity) of scope completed. Agile units of measure may include user story counts, feature counts, or capability counts. Burn-Up charts typically show progress over time against a planned release, as in the example shown in Figure 12. Overall progress and technical performance should be tracked at the feature or capability level as that is the level tied directly to contract deliverable requirements.

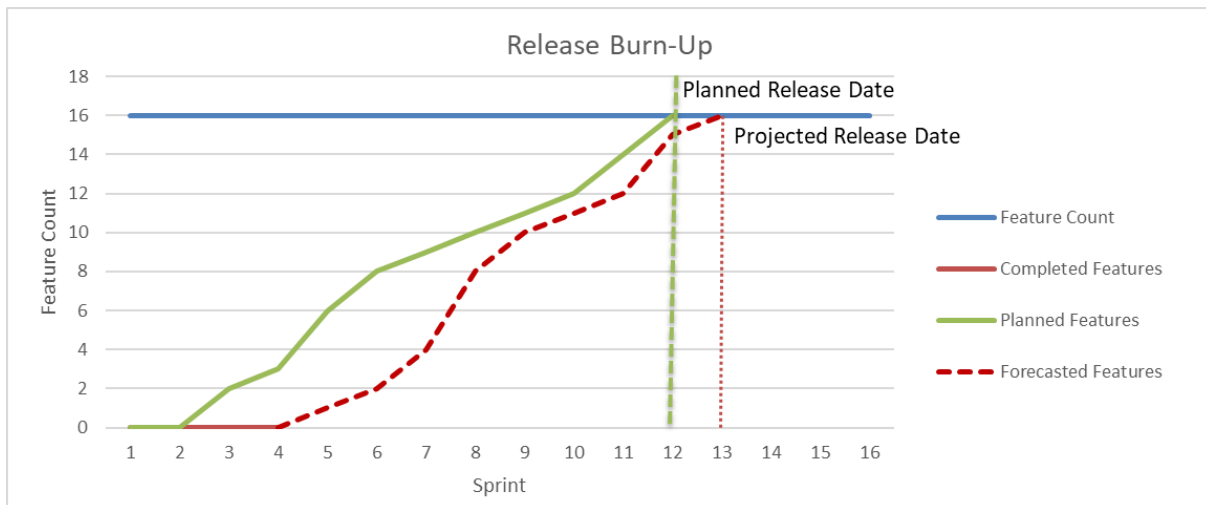


FIGURE 12. Release Burn-Up

The blue line across the top is the total amount of work in the backlog by feature count. The green line is the planned completion count and the red line is the actual count of completed features. The gap between the green line and the red line indicates there is a schedule variance and likelihood that all the scope may not be completed by the 12th sprint.

Considerations and questions that can be derived from a Burn-Up or Burn-Down Chart include:

- What is the overall status?
 - Are additional Agile teams needed to meet demands?
 - What is the plan for work across sprints? (In Figures 11 and 12, the planned work is constant across sprints).
 - How much work is being accomplished and at what rate?
 - Will all the stories be accomplished in the expected number of sprints? Was too much work planned in a sprint based on team capacity? In Figure 12, the gap between red and green lines suggests all features may not be completed by the end of the release.
- a. *Capability Progress Measure*: Provides status of completion of a capability. When combined with the other current capabilities, the chart, as shown in Figure 13, provides an at-a-glance view of all capabilities in the program portfolio at a point in time. It allows the reviewer to quickly see status of completed and in-progress story points by capability. Also, charts can provide a view of the current estimated total story points compared to the initial estimate to show changes in overall story estimates. This metric can also be adapted for capabilities in Scrum/Kanban or broken down to show features.

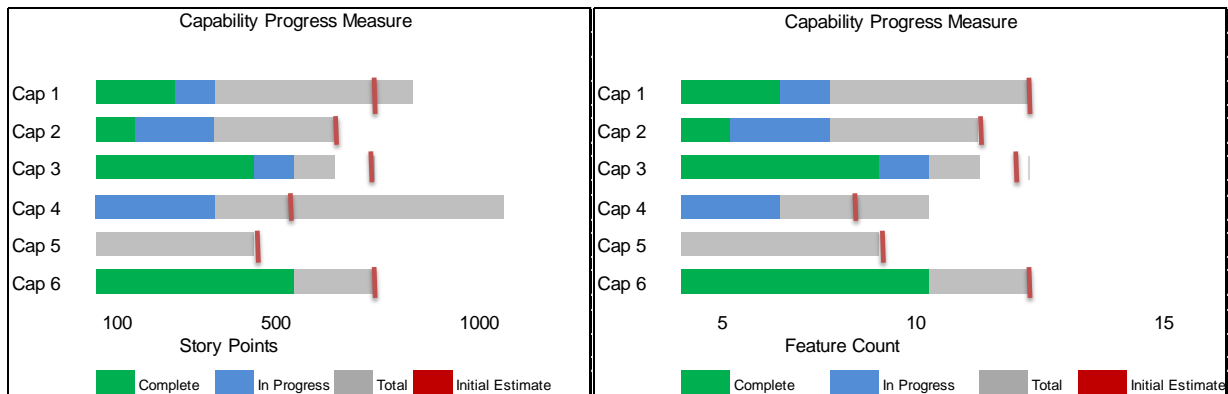


FIGURE 13. Capability Progress Charts (Derived from SAFe Metrics)

In the chart on the left, the bar length represents the total story points for each capability's child features and stories. The vertical red line indicates the initial estimate of story points for the capability, and the colored segments represent the status. The chart on the right is similar and represents the feature count of the same capabilities.

The story point view has the advantage of showing partial completion of features and provides perspective of "work" completed. The feature count view has the advantage of showing value from a customer perspective, i.e., completed features. They are complimentary and provide different and useful data.

Feature progress charts that show progress of features by story count are also sometimes used. When the EVM work package is at the feature level, the features in the schedule (IMS) and the progress from the agile tool are rolled into the schedule as feature percent complete. Because the IMS contains dependencies between features scheduled, the roll-up of agile status can support variance analysis and cross check against forecasted work.

The Capability Progress Measure charts in Figure 13 are useful as a program dashboard and can be mapped to the BCWP reported at the work package or control account level, reported variances in Format 5, and can be used to inform or support EAC analysis.

Understanding Work in Process (WIP)

Like the Baseline Execution Index (BEI), Counting Stories (when it includes counts for Stories complete, started, not started) can be useful in understanding and managing Work in Progress (WIP) and can be used to inform the downstream workload forecasts.

Kanban and Lean Development focus heavily on WIP, or rather minimizing WIP, which represents the amount of open work in a given point in workflow. The volume of WIP directly correlates to an amount of resources needed to manage and complete the in-process work (i.e., staff) and it represents time and money invested without completed scope. Too much WIP hides bottlenecks, masks efficiency issues, and carries the risk of potential rework.

WIP limits are typically used in conjunction with Kanban Boards to optimize throughput of work by optimizing the number of open tasks/items at a given stage to an upper limit. New work cannot be pulled into the queue until open work progresses sufficiently in the workflow to allow it. Appropriate WIP limits ensure a sufficient amount of work is being performed, bottlenecks are cleared, and there is a consistent workflow, even if there is idle or slack time in the schedule.

When teams have an amount of WIP that is above the level of efficiency, the overall feature delivery tends to be delayed and a bow-wave of work can be identified. It is important to note that at the beginning of a program, and especially for newly developed teams, the rate of work that can be accomplished is often lower than will be experienced later. Eventually the rate of completed features should achieve a maximum sustainable pace.

Workflow related metrics include:

- Lead Time: the total time it takes to deliver an item, measured from the time it is opened to the moment it is completed ($\text{Lead Time (days)} = \text{Date Completed} - \text{Date Added to Backlog}$)
- Response Time: the time that an item waits before it starts ($\text{Response Time (days)} = \text{Date Work Started} - \text{Date Added to Backlog}$)
- Cycle Time: the time required to process an item ($\text{Cycle Time (days)} = \text{Date Completed} - \text{Date Work Started}$)

All three of these measures can be applied at both the feature and story level.

Note that each feature is unique, so its corresponding cycle time will be unique. However, common sized features with similar complexity should have similar cycle times. Cycle time is useful to manage development efforts with external dependencies.

A cumulative flow diagram depicts work versus time to include Lead, Response, and Cycle times. A typical view as shown in Figure 14 has the tasks and work of the agile development team on the vertical Y-axis, while the horizontal X-axis represents time. Cumulative flow diagrams are typically used to track the progress and trends for the entire project at the feature level.

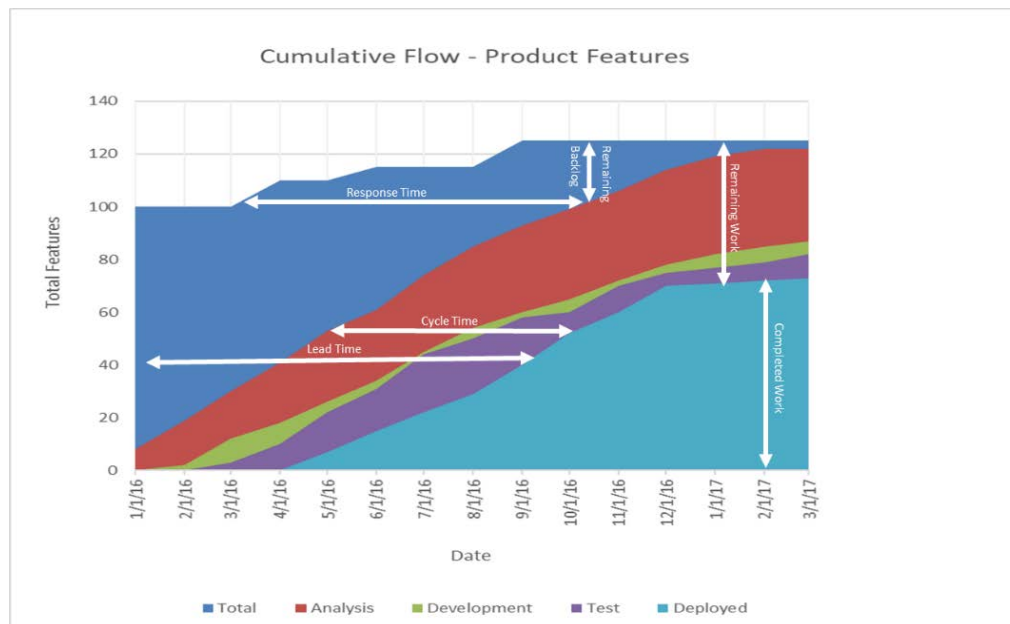


FIGURE 14. Cumulative Flow Diagram Example

In Figure 14, the remaining backlog (i.e., total number of features) is in blue at the top. The teal line nearest the X-axis are the completed features. The red, green, and purple lines represent the amount of

work (number of features) in various stages of development. In reading the chart, a surge in progress would be seen as a steep increasing slope of the completed line while a stall in completion would be identified by a flatter slope in the completed line. A widening gap between the number of In Progress and Completed Features would represent a bottleneck and increase in WIP at that stage, and a potential bow-wave of incomplete work.

The vertical thickness of the combined red, green, and purple bands represent total WIP, which is increasing over time in the figure, and an undesirable characteristic. One can see that this is caused by a large amount of work in Analysis.

Agile Metrics Related to EVM Metrics

Earned Value Management provides program management insight to manage the day to day aspects of a program and adjust the plan to complete the work as needed. It is a disciplined process for defining, planning, executing, and monitoring execution status against a plan. In reviewing problem areas and variances against the plan, the program management team evaluates the issue areas to establish root causes and develop mitigation steps. EVM provides the PM the metrics and information needed for situational awareness and Agile metrics are used to underpin and support the EVM metrics.

Three key values are used to measure Cost and Schedule variances and indexes.

- *Budgeted Cost of Work Scheduled (BCWS)*. The sum of the budgets for all work packages, planning packages, etc., scheduled to be accomplished (including in-process work packages), plus the amount of level of effort and apportioned effort scheduled to be accomplished within a given period.²³ For Agile, the time phased budget is established using similar cost estimating techniques as any traditional program. However, for Agile development it is important to keep the cost at a level that aligns with EVM reporting.
- *Budgeted Cost of Work Performed (BCWP)*. The value of completed work expressed as the value of the performance budget assigned to that work.²⁴ EVM Progress on Agile programs is established by measuring the accomplishment of work, typically identified as Features or Capabilities.
- *Actual Cost of Work Performed (ACWP)*. The costs incurred and recorded in the Earned Value Management System (EVMS) for accomplishing the work performed within a given accounting period.²⁵ Actual costs in an Agile program are collected the same as any traditional program where charge numbers are established for executing specific work scope.

The EVM metrics that are derived from the three key values above include the following:

- *Cost Variance (CV) = BCWP – ACWP*
- *Schedule Variance (SV) = BCWP – BCWS*
- *Cost Performance Index (CPI) = BCWP/ACWP*
- *Schedule Performance Index (SPI) = BCWP / BCWS*

The next section describes how these metrics are underpinned using Agile metrics.

²³ DoD EVMSIG, February 2013, Definition of “BCWS”

²⁴ DoD EVMSIG, February 2013, Definition of “BCWP”

²⁵ DoD EVMSIG, February 2013, Definition of “ACWP”

Comparison of Agile and EVM Status Charts

In Agile, progress of a capability is based on the technical completion of each of its features, which in turn is based on the accomplishment of its feature's acceptance criteria. While there are various approaches to collecting progress, the scenario described in Appendix A is an example utilizing weighted story values for Quantifiable Backup Data (QBD) to claim progress against feature level tasks. It shows how Agile and EVM metrics relate and how they can be used together to forecast cost and schedule.

Figure 15 is an example data trace from the product backlog, through a sprint status sheet, to the Integrated Master Schedule (IMS).

- The product backlog in this example shows alignment of EVM WBS elements to the Agile products hierarchy. The EV control account aligns with the Agile capability, and EV work packages align with Agile features and is an abstraction of a product backlog showing sprint status. Tracing the plan and progress of feature A1, the product backlog listing shows feature A1 consists of three stories (A1.1, A1.2, A1.3), all stories were planned for Sprint 1. In the actual sprint column, the listing shows only two of the three stories were completed in Sprint 1.
- The sprint status sheet shows, using the weighted values of Story A1.1 and A1.2, the feature percent complete is computed after Sprint 1 ($7/9 = 78\%$).
- The feature progress computed in the previous step manifests itself as progress information in the IMS (See Physical Percent Complete for Feature A.1).
- This is a critical thread, Agile technical progress informs the EVM progress claims in the IMS, once that occurs you have crossed the EVM boundary and the trace to the EVM System from the IMS is the standard practice.
- The percent complete at the feature level also informs the BCWP.

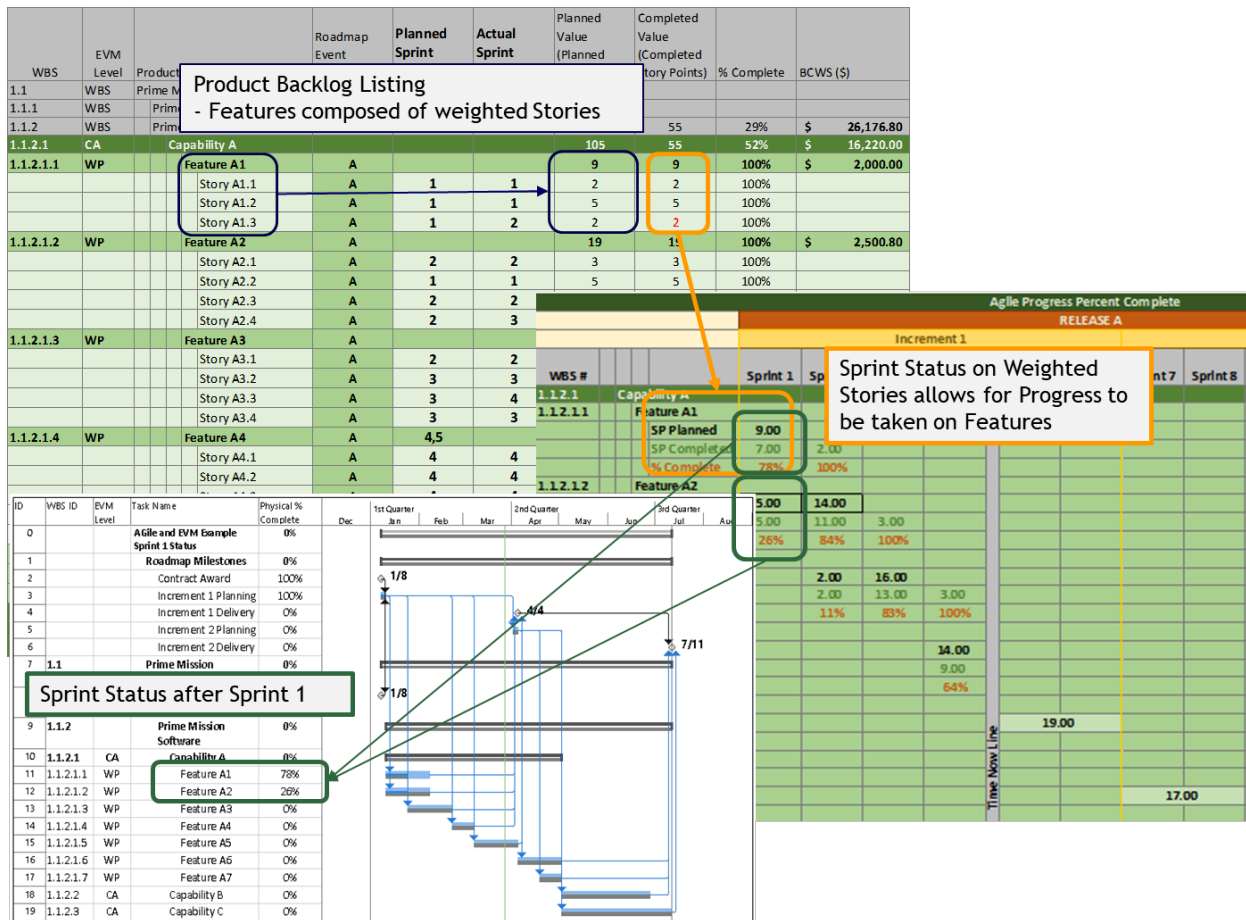


FIGURE 15. Example Data Trace showing work decomposition and use of weighted stories to collect and provide Agile progress, at the Feature level, to the IMS.

Using the calculated BCWP from the progress claims in the IMS Standard EVM practice can be used to calculate an EAC. In addition, the Agile data can be plotted using an Agile burn-up chart and shown side by side with the Level 1 EVM status chart, Figure 16. Caution, the PMO should not try to relate graphs shown in Figure 16 as an exact comparison but rather that the two graphs should be showing similar trends.

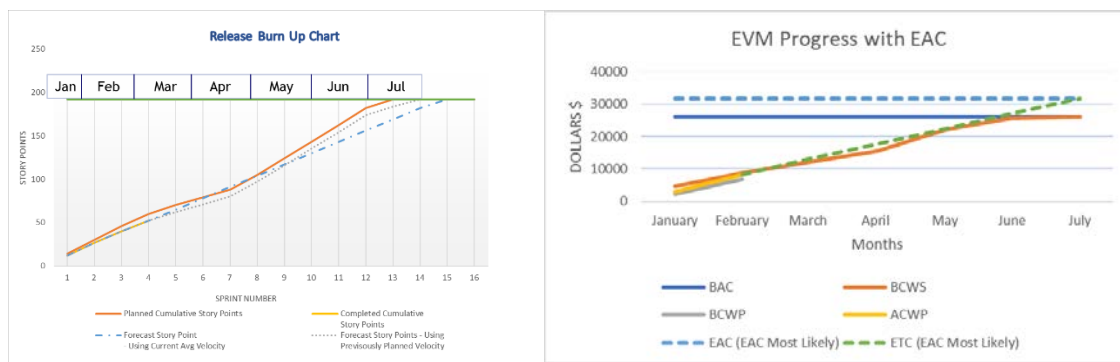


FIGURE 16. Cumulative Data and EAC Projections

Figure 17 provides the data and formulas used in the example charts above. Estimates at Complete are calculated using EVM Gold Card Formulas.

Like the way TCPI can be compared to CPI in evaluating the realism of achieving a cost target; the “Required Velocity to Complete to Baseline” can be compared to the Average Velocity to evaluate if the remaining backlog can be completed within a target number of sprints.

Sprint	1	2	3	4	Formula
Using EVM Gold Card Formulas					
EAC (worst case)	\$ 43,868.58	\$ 36,282.68	\$ 33,401.70	\$ 33,303.07	= ACWP + (BAC-BCWP)/(CPIxSPI)
EAC (most likely)	\$ 34,243.64	\$ 33,773.54	\$ 31,796.58	\$ 31,074.35	= BAC / CPIcum
EAC (best case)	\$ 26,879.55	\$ 27,476.63	\$ 27,659.67	\$ 27,893.00	= ACWP + (BAC - BCWPcum)
BAC	\$ 26,200.00	\$ 26,200.00	\$ 26,200.00	\$ 26,200.00	
BCWScum	\$ 2,893.00	\$ 4,811.00	\$ 7,300.00	\$ 10,100.00	
ACWPCum	\$ 2,893.00	\$ 5,693.00	\$ 8,293.00	\$ 10,793.00	
BCWPCum	\$ 2,213.45	\$ 4,416.37	\$ 6,833.33	\$ 9,100.00	= BCWS x (Story Points Completed / Story Points Planned)
CPIcum	0.77	0.78	0.82	0.84	= BCWP / ACWP
SPIcum	0.77	0.92	0.94	0.90	= BCWP / BCWS
TCPI (Baseline)	1.03	1.06	1.08	1.11	= (BAC - BCWP) / (BAC - ACWP)
Agile Metrics					
Total Story Points	192	192	192	192	
Completed Story Points (cumulative)	12	27	40	52	
% Complete	6.25%	14.06%	20.83%	27.08%	= (Story Points Completed) / (Total Story Points)
Average Velocity	12	13.5	13	13	= (Story Points Completed) / (# Sprints Completed)
Predicted Remaining Number of Sprints	15	12	12	11	= (Total SP - Completed SP) / Avg Velocity
Required Velocity to Complete to Baseline	16	17	17	18	= (Remaining SP - Completed SP) / (Baseline # Sprints - Completed # Sprints)

FIGURE 17. EAC Projections EVM Gold Card Metrics with Agile-Informed BCWP

When forecasting, teams use progress metrics, root cause analysis, and current risks to predict end program cost and schedule. Agile metrics can be another data source of information to inform cost and schedule forecasts and provide insight into root cause and risk analyses.

Resources for Additional Information on Agile Metrics

- “Agile Metrics: Progress Monitoring of Agile Contractors” Carnegie Mellon University Software Engineering Institute. 2014. Technical Note CMU/SEI-2013-TN-029.
- “Agile Practice Guide” by the Project Management Institute. 2017.
- “An Industry Guide for Agile on Earned Value Management Programs” Version 1.1, by NDIA Integrated Program Management Division. 31-MAR-2017.
<http://www.ndia.org/divisions/ipmd/division-guides-and-resources>.
- “Scaling Software Agility – Best Practices for Large Enterprises” by Dean Leffingwell. 2007. ISBN: 978-0-321-45819-3.

APPENDIX

Example Agile and EVM Scenario

This appendix describes a simple Agile and EVM scenario showing one method of how Agile can underpin EVM progress in support of tracking program cost and schedule status and metrics.

Scenario Description

The scenario is a generic SW program implementing an Agile development process. The scenario has defined a single product Release (Release A), which will be tracked using Earned Value Management. The product Release delivery date is scheduled after the 13th Sprint. The SW Release must include all the defined scope for the release and it will occur once all the scope is completed.

The program's Agile process includes Program Increment planning and execution. Each increment is a "Time-Boxed" period consisting of six Sprints. Each Sprint is 2 weeks long. The Scenario includes 2 Program Increments for a total of 12 Sprints. The program is managing the work using a SCRUM cycle for each Sprint.

Figure 1 is an abstraction of a product backlog showing status for Increment 1. The SW effort is structured around the major SW objectives (products), defined as Capabilities, decomposed into Features and Stories. Agile development will prioritize and time phase implementation of the Features and decompose those Features into Stories, assigning Stories to Sprints.

Figure 1 shows alignment of EVM WBS elements to the Agile products hierarchy. The SW effort structure is consistent with a typical MIL-STD-881D WBS. The figure also shows the WBS level where Control Accounts and Work Packages (WP) are represented forming the basis for EVM and Agile alignment. The EV control account aligns with the Agile capability, and EV Work Packages align with Agile Features. The features and their status will form the basis for calculating percent complete against the control account. Feature level BCWS is included from the Control Account Plan for reference.

Described below are each of the column headings shown in the figure.

WBS – Contains the WBS number used for traceability through the EVM System.

EVM Level – This column shows the EVM level of hierarchy. WBS level, Control Account Level, or Work Package Level. Planning Packages (PP) or Summary Planning Packages (SPP) would be identified in this column, this example does not have any PPs or SPPs

Product Item – WBS / Product Name. On a specific project this would be augmented with a product description at each level defining the work and acceptance criteria.

Roadmap Event (Release #) – This column identifies which Capabilities and Features are allocated to which Release. In this example the entire Product Backlog is allocated to Release A.

Planned Sprint Number – This column identifies which Sprint each Story is planned for implementation. Notice that Features are decomposed to Stories and Stories allocated to Sprints only in the near term. In this example the team has conducted four Sprints and is ready to start Sprint 5.

Actual Sprint Number – This column is filled in after the completion of each Sprint to show when Stories

were completed. Notice that Story A1.3 was planned for Sprint 1 but did not complete until Sprint 2. Story A2.4 was planned for Sprint 2 but completed in Sprint 3, Story A3.3 was planned for Sprint 3 but completed in Sprint 4, and finally Story A4.4 was planned for Sprint 4 but did not complete and will have to be planned for a future Sprint.

Planned Value (Planned Story Points) – This column shows the result of the story point estimates done by the team. Story Points are relative size estimates assigned to stories and will be used to calculate % complete at the Feature level. Teams take credit when each story is completed. The weighted Story (story point) values are rolled up to establish a total weight or value of the Feature.

Completed Value (Completed Story Points) – This column is populated at the end of each Sprint with the weighted value of each Story that was completed. The numbers highlighted in red indicate that this Story was done late, which you can see from the Sprint complete column.

Percent Complete – This column is a cumulative % complete over all completed Sprints. Notice that Stories are either 0 or 100 percent complete. You can see for Feature A4 that it is only 64 % complete since only 4 of the 5 stories have been completed to date. Using the weighted values (story points), you can calculate an overall % complete ($9/14 = .64$).

BCWS (\$) – This column shows the planned value for each of the Features. These numbers are not derived from the Agile process but will come from the EVM System. They are shown here to use later to calculate BCWP for the Features.

WBS	EVM Level	Product Item	Roadmap Event (Release #)	Planned Sprint Number	Actual Sprint Number	Planned Value (Planned Story Points)	Completed Value (Completed Story Points)	% Complete	BCWS (\$)
1.1	WBS	Prime Mission Subsystem							
1.1.1	WBS	Prime Mission Hardware							
1.1.2	WBS	Prime Mission Software				192	55	29%	\$ 26,176.80
1.1.2.1	CA	Capability A				105	55	52%	\$ 16,220.00
1.1.2.1.1	WP	Feature A1	A			9	9	100%	\$ 2,000.00
		Story A1.1	A	1	1	2	2	100%	
		Story A1.2	A	1	1	5	5	100%	
		Story A1.3	A	1	2	2	2	100%	
1.1.2.1.2	WP	Feature A2	A			19	19	100%	\$ 2,500.80
		Story A2.1	A	2	2	3	3	100%	
		Story A2.2	A	1	1	5	5	100%	
		Story A2.3	A	2	2	8	8	100%	
		Story A2.4	A	2	3	3	3	100%	
1.1.2.1.3	WP	Feature A3	A			18	18	100%	\$ 2,800.00
		Story A3.1	A	2	2	2	2	100%	
		Story A3.2	A	3	3	8	8	100%	
		Story A3.3	A	3	4	3	3	100%	
		Story A3.4	A	3	3	5	5	100%	
1.1.2.1.4	WP	Feature A4	A	4		14	9	64%	\$ 2,800.00
		Story A4.1	A	4	4	2	2	100%	
		Story A4.2	A	4	4	3	3	100%	
		Story A4.3	A	4	4	1	1	100%	
		Story A4.4	A	4		5		0%	
		Story A4.5	A	4	4	3	3	100%	
1.1.2.1.5	WP	Feature A5	A	5,6		19		0%	\$ 2,500.80
1.1.2.1.6	WP	Feature A6	A	6,7		17		0%	\$ 1,920.00
1.1.2.1.7	WP	Feature A7	A	7		9		0%	\$ 1,698.40
1.1.2.2	CA	Capability B	A			40		0%	\$ 4,768.00
1.1.2.3	CA	Capability C	A			47		0%	\$ 5,188.80
Total						192			

APP FIGURE 1. Shows alignment of EVM WBS elements to the Agile products hierarchy. The EV control account aligns with the Agile capability, and EV Work Packages align with Agile Features.

Figure 2 shows a Sprint by Sprint time phasing of the information from Figure 1. This view shows the Sprint by Sprint status of each of the Features. From Figure 18 it is shown that for Feature A1, only Story A1.1 and A1.2 were completed and for Feature A2, Story A.2 was completed as planned. Thus, after Sprint 1, Feature A1 was 78% complete and Feature A2 was 26% complete. These percentages are rolled into the IMS at the Feature level. Since Feature A1 was supposed to be done within the first Sprint, but did not finish until the second Sprint, a corresponding slip would be shown in the IMS.

Figure 1 and Figure 2 show the basis of how Agile progress underpins EVM status.

Agile Progress Percent Complete																				
		RELEASE A																		
		Increment 1				Increment 2														
WBS #					Sprint 1	Sprint 2	Sprint 3	Sprint 4		Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Sprint 11	Sprint 12	Sprint 13	Total (Sty Pts)	
1.1.2.1	Capability A								Time Now Line											
1.1.2.1.1	Feature A1																			
		SP Planned		9.00																9.00
		SP Completed		7.00	2.00															9.00
		% Complete		78%	100%															
1.1.2.1.2	Feature A2																			
		SP Planned		5.00	14.00															19.00
		SP Completed		5.00	11.00	3.00														19.00
		% Complete		26%	84%	100%														
1.1.2.1.3	Feature A3																			
		SP Planned			2.00	16.00														18.00
		SP Completed			2.00	13.00	3.00													18.00
		% Complete			11%	83%	100%													
1.1.2.1.4	Feature A4																			
		SP Planned					14.00													14.00
		SP Completed					9.00													9.00
		% Complete					64%													
1.1.2.1.5	Feature A5										19.00									19.00
		SP Planned																		0.00
		SP Completed																		
		% Complete																		
1.1.2.1.6	Feature A6												17.00							17.00
		SP Planned																		0.00
		SP Completed																		
		% Complete																		
1.1.2.1.7	Feature A7											9.00							9.00	
		SP Planned																	0.00	
		SP Completed																		
		% Complete																		
1.1.2.2	Capability B													40.00					40.00	
		SP Planned																		
		SP Completed																		
		% Complete																		
1.1.2.3	Capability C																			
		SP Planned											47.00						47.00	
		SP Completed																		
		% Complete																		
Total Program					14.00	16.00	16.00	14.00		10.00	9.00	9.00	17.00	19.00	19.00	19.00	20.00	10.00	192.00	
SP Completed Cur					12.00	15.00	16.00	12.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	55.00	
SP Completed Cum					12.00	27.00	43.00	55.00												
% Complete total					6%	14%	22%	29%												

APP FIGURE 2. Shows Feature Time-phasing using story points.

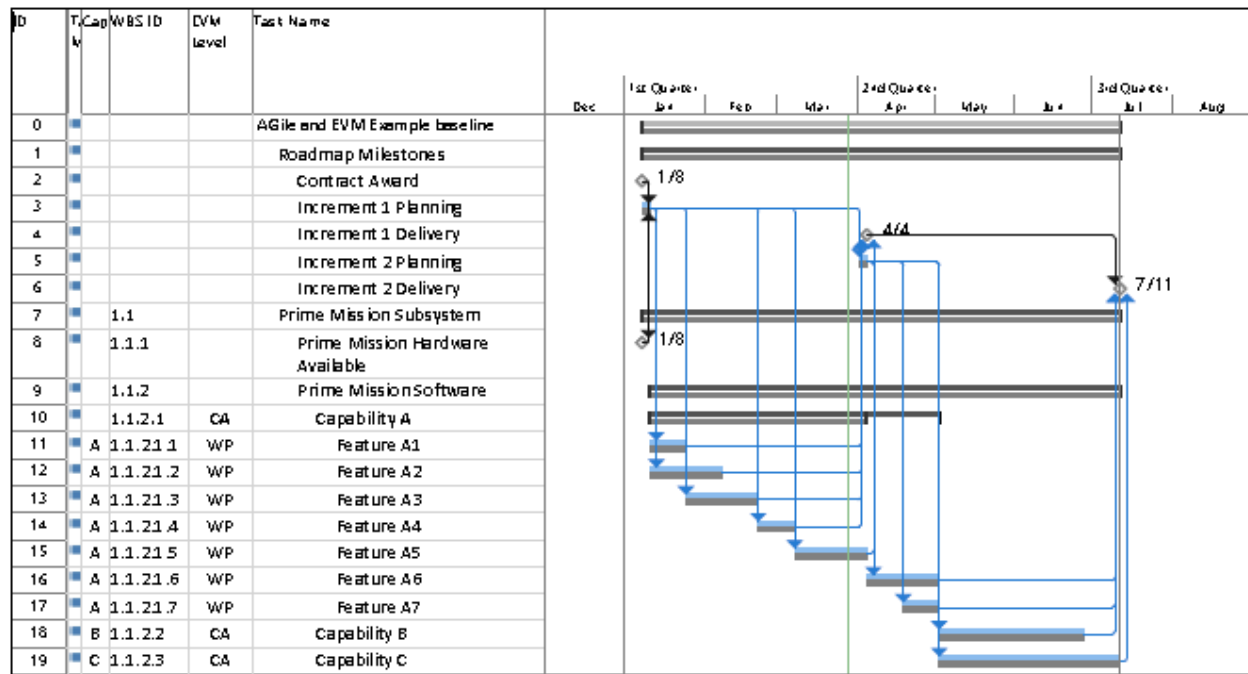
The figure shows the story points planned and completed for each feature. Feature % Complete is calculated to support EV progress shown in Figure 20.

Figure 3 shows how tracking Agile progress in Figure 2 underpins and supports Work Package progress in the EVMS. Feature A1 progress claimed of 78% after Sprint 1, shown in the previous diagram, is used to establish the claimed dollar value for Sprint 1. $(7/9) * \$2000 = \1556 .

EVM Progress Cost																	
WBS #			Sprint 1	Sprint 2	Sprint 3	Sprint 4		Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Sprint 11	Sprint 12		BCWS (\$)
1.1.2.1	Capability A																\$16,220.00
1.1.2.1.1	Feature A1																\$ 2,000.00
	BCWS		\$ 2,000														
	BCWP		\$ 1,556	\$ 444													
	ACWP		\$ 2,000	\$ 200													
1.1.2.1.2	Feature A2																\$ 2,500.80
	BCWS		\$ 1,250	\$ 1,250													
	BCWP		\$ 658	\$ 1,448	\$ 395												
	ACWP		\$ 893	\$ 1,500	\$ 350												
1.1.2.1.3	Feature A3																\$ 2,800.00
	BCWS			\$ 1,400	\$ 1,400												
	BCWP			\$ 311	\$ 2,022	\$ 467											
	ACWP			\$ 800	\$ 1,900	\$ 300											
1.1.2.1.4	Feature A4																\$ 2,800.00
	BCWS					\$ 2,800											
	BCWP					\$ 1,801											
	ACWP					\$ 2,200											
1.1.2.1.5	Feature A5																\$ 2,500.80
	BCWS							\$ 2,501									
	BCWP																
	ACWP																
1.1.2.1.6	Feature A6																\$ 1,920.00
	BCWS								\$ 1,920								
	BCWP																
	ACWP																
1.1.2.1.7	Feature A7																\$ 1,698.40
	BCWS									\$ 1,698							
	BCWP																
	ACWP																
1.1.2.2	Capability B																\$ 4,768.00
	BCWS																
	BCWP									\$ 4,768							
	ACWP																
1.1.2.3	Capability C																\$ 5,188.80
	BCWS																
	BCWP																
	ACWP																
Total																	
			</														

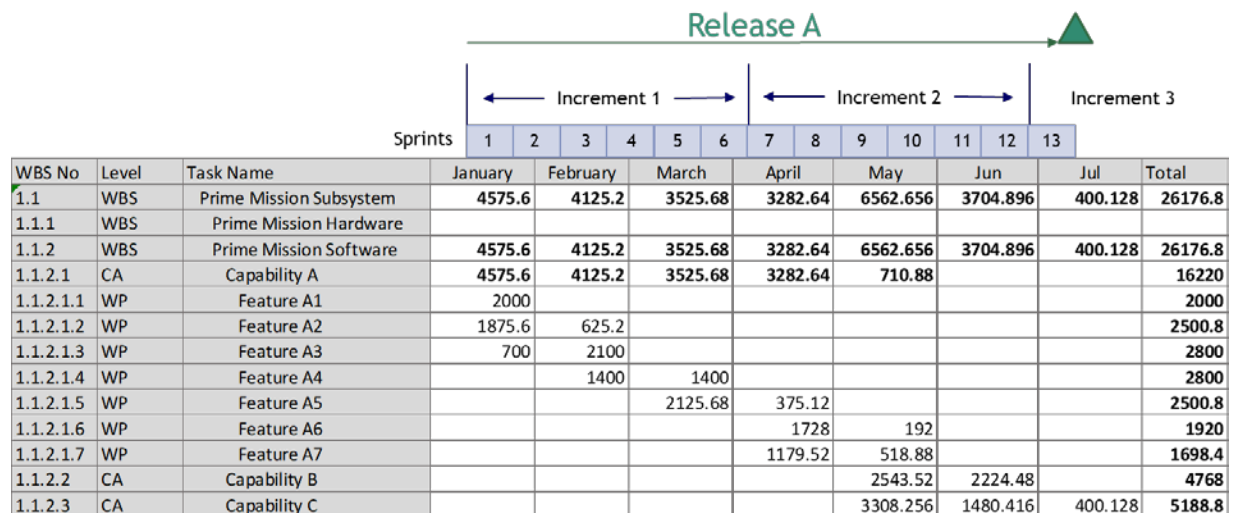
APP FIGURE 3. Shows how the Agile information can manifest itself as progress information in the IMS and BCWP for the control account.

The IMS for this scenario is shown in Figure 4. Notice that the Feature is the lowest level of the IMS. Progress from the Agile process is rolled into the IMS by updating the EVM % complete, in this example we are using Physical % complete.



APP FIGURE 4. Simple Scenario IMS

Figure 5 shows the monthly time phased budget for the program. This is also shown in this diagram is the alignment of the Sprint boundaries to the end of month boundaries.



APP FIGURE 5. Simple Scenario Time Phased Budget

Status After Sprint 4

The final section of this Appendix shows the Updated IMS and the Time Phased budget after Sprint 4 status has been input.

Agile Status

	Roadmap Event (Release #)	Planned Sprint Number	Actual Sprint Number	Planned Value (Planned Story Points)	Completed Value (Completed Story Points)
Capability A				105	55
Feature A1	A			9	9
Story A1.1	A	1	1	2	2
Story A1.2	A	1	1	5	5
Story A1.3	A	1	2	2	2
Feature A2	A			19	19
Story A2.1	A	2	2	3	3
Story A2.2	A	1	1	5	5
Story A2.3	A	2	2	8	8
Story A2.4	A	2	3	3	3
Feature A3	A			18	18
Story A3.1	A	2	2	2	2
Story A3.2	A	3	3	8	8
Story A3.3	A	3	4	3	3
Story A3.4	A	3	3	5	5
Feature A4	A	4		14	9
Story A4.1	A	4	4	2	2
Story A4.2	A	4	4	3	3
Story A4.3	A	4	4	1	1
Story A4.4	A	4		5	
Story A4.5	A	4	4	3	3
Feature A5	A	5,6		19	

Capability A	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Feature A1				
SP Planned	9.00			
SP Completed	7.00	2.00		
% Complete	78%	100%		
Feature A2				
SP Planned	5.00	14.00		
SP Completed	5.00	11.00	3.00	
% Complete	26%	84%	100%	
Feature A3				
SP Planned		2.00	16.00	
SP Completed		2.00	13.00	3.00
% Complete		11%	83%	100%
Feature A4				
SP Planned				14.00
SP Completed				9.00
% Complete				64%
Feature A5				

Sprint #	0	1	2	3	4
Total Planned Increment		79	79	79	79
Release Total		192	192	192	192
Planned Story Points		14	16	16	14
Planned Cumulative Story Points		14	30	46	60
Completed Story Points Current	0	12	15	13	12
Completed Cumulative Story Points		12	27	40	52
Forecast Story Point - Using Current Avg Velocity		12	27	40	52
Forecast Story Points - Using Previously Planned Velocity		12	27	40	52
Average Velocity		12	13.5	13	13
Predicted Remaining Number of Sprints to Complete Release		15	12	12	11
Required Velocity to Complete (13 Sprints)		16	17	17	18

APP FIGURE 6. Agile Status by Sprint for this scenario

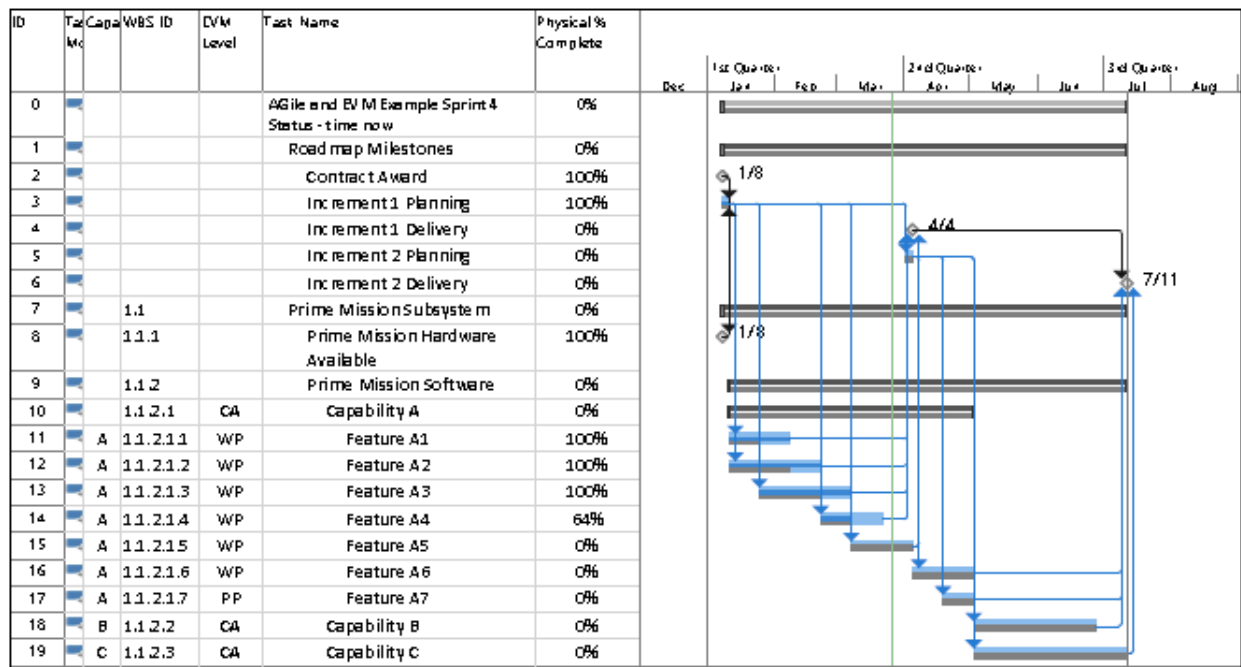
EVM BCWP Derived from Agile Percent Complete

		Sprint 1	Sprint 2	Sprint 3	Sprint 4
Capability A					
	Feature A1				
	BCWS	\$ 2,000			
	BCWP	\$ 1,556	\$ 444		
	ACWP	\$ 2,000	\$ 200		
	Feature A2				
	BCWS	\$ 1,250	\$ 1,250		
	BCWP	\$ 658	\$ 1,448	\$ 395	
	ACWP	\$ 893	\$ 1,500	\$ 350	
	Feature A3				
	BCWS		\$ 1,400	\$ 1,400	
	BCWP		\$ 311	\$ 2,022	\$ 467
	ACWP		\$ 800	\$ 1,900	\$ 300
	Feature A4				
	BCWS				\$ 2,800
	BCWP				\$ 1,801
	ACWP				\$ 2,200
	Feature A5				

APP FIGURE 7. Calculation of BCWP by Sprint

BCWP Rolled into the IMS

Notice the slip from Baseline by two weeks, Feature A1, A2, A3 and A4



APP FIGURE 8. Status IMS after Sprint 4

Corresponding EVM Metrics by Sprint

Once Agile Progress informs the schedule, BCWP (\$) can be calculated using the Feature percent complete. Once BCWP is known all the rest of the EVM metrics can be calculated to include EAC forecasts.

Sprint	1	2	3	4
Using EVM Gold Card Formulas				
EAC (worst case)	\$ 48,876.62	\$ 43,159.04	\$ 33,367.20	\$ 33,265.83
EAC (most likely)	\$ 34,209.80	\$ 33,738.23	\$ 31,764.46	\$ 31,042.11
EAC (best case)	\$ 26,855.94	\$ 27,452.55	\$ 27,635.47	\$ 27,868.29
BAC	\$ 26,176.60	\$ 26,176.60	\$ 26,176.60	\$ 26,176.60
BCWS _{cum}	\$ 3,250.40	\$ 5,900.80	\$ 7,300.80	\$ 10,100.80
ACWP _{cum}	\$ 2,893.00	\$ 5,693.00	\$ 8,293.00	\$ 10,793.00
BCWP _{cum}	\$ 2,213.66	\$ 4,417.05	\$ 6,834.13	\$ 9,101.31
CPI _{cum}	0.77	0.78	0.82	0.84
SPI _{cum}	0.68	0.75	0.94	0.90
TCPI (Baseline)	1.03	1.06	1.08	1.11

APP FIGURE 9. EVM Metrics by Sprint

EVM Metrics by Month – Notice that the BCWP after Sprint 1 is the same as the BCWP at end of January. This is because the only Sprint completed in January was Sprint 1. At the end of February Sprint 2 and Sprint 3 have been completed and informs the Monthly February Report. Sprint 4 was completed but progress is not reflected in this example even though some work has been done on Sprint 4. This is done in this example to highlight the timing idiosyncrasies between Sprint boundaries and Monthly EVM boundaries.

Earned Value Metrics (Cumulative)	January	February	March	April	May	June	July
BAC	26177	26177	26177	26177	26177	26177	26177
BCWS	4576	8701	12226	15509	22072	25777	26177
BCWP	2213.661	6834.133					
ACWP	2893	8293					
CV	-679.34	-1458.87					
SV	-2361.94	-1866.67					
CPI	0.77	0.82					
SPI	0.48	0.79					
EAC (EAC Most Likely)	31764.7	31764.7	31764.7	31764.7	31764.7	31764.7	31764.7
ETC (EAC Most Likely)		8293	12987.34	17681.68	22376.02	27070.36	31764.7
EAC (Best Case)	26856	27636					
EAC (Worst Case)		38175.73					

APP FIGURE 10. EVM Metrics by Month

Agile Reference Terms

Backlog	A backlog is a list of Features or technical tasks which the team maintains and which, at a given moment, are known to be necessary and sufficient to complete a project or a Release . The backlog is the primary point of entry for knowledge about requirements, and the single authoritative source defining the work to be done.
Burn Up/Burn Down Chart	A large graph relating the quantity of work remaining (on the vertical axis) and the time elapsed since the start of the project (on the horizontal, showing future as well as past). Two variants exist, depending on whether the amount graphed is for the work remaining in the iteration ("sprint burndown") or more commonly the entire project ("product burndown").
Cadence	A regular, predictable rhythm or heartbeat. Scrum Sprints of consistent duration establish a cadence for a development effort.
Capability	Term frequently used interchangeably with Epic to describe a high level system functionality defined by the government to meet a specific required need. All Capabilities should have clearly defined objective technical completion criteria. Capabilities are typically found at or above the Control Account level of the WBS and are usually composed of multiple Features .
Epic	See Capability .
Feature	Term used to describe a discrete system functionality defined by the government to help meet the specific completion criteria of a Capability . All Features should have clearly defined objective technical completion criteria. Features are typically found at the Work Package level of the WBS and can typically be completed in a single Release .
Iteration	See Sprint .
Release	Term used to describe a concrete time box or cadence used to complete Features . Release duration can vary, but is typically three to six months. Many practitioners use the Release cadence as their rolling wave planning period.
Roadmap	The overview of the entire Agile project.
Sprint	Term frequently used interchangeably with Iteration to describe a concrete time box or cadence used to complete Stories . Sprint duration can vary, but is typically two to four weeks.
Story (User Story)	Term used to describe activities that contribute to the completion of a Feature and can be completed within a single Sprint .
Velocity	At the end of each Iteration , the team adds up effort estimates associated with User Stories that were completed during that Iteration . This total is called velocity.