National Defense Industrial Association
Integrated Program Management Division

An Industry Practice Guide for Integrating Agile and Earned Value Management on Programs

December 9, 2022

Version 1.4

National Defense Industrial Association (NDIA)
2101 Wilson Blvd., Suite 700
Arlington, VA 22201
(703) 522-1820
Fax (703) 522-1885
www.ndia.org

Table of Contents

*** Disclaimer: This guide assumes basic knowledge and understanding of both Agile and EVM, it is not meant for practitioners new to Agile or EVM. Seek alternate foundational Agile and EVM training resources prior to utilization of this document. Nor does this guide supersede any organizational documented Agile or EVM policy.*

Introduction: EVM for Agile Programs

The growing importance of quickly and affordably delivering business outcomes has led to an increased focus on capability-based planning and iterative product development and delivery. To swiftly react to the changing demands of an operational environment requires programs adapt new delivery methods for software and hardware products and systems. Planning and execution focus on delivering the highest priority system functionality to the stakeholders as quickly and affordably as possible. To meet this demand, Program Managers need a planning and execution method that can quickly and efficiently react to changes at the necessary relevant speed across all levels of the program. Program Managers require Corporate Leadership and Contracting Representative support to ensure contracts are executed at the required pace. Agile has emerged as the leading industry product development methodology and has seen growing adoption across the Department of Defense (DoD) and other federal agencies. Agile implements the needed method by focusing on small, frequent releases, working software through demonstration of capabilities, responding rapidly to changes in operations, technology, and budgets, and actively involving users throughout development to ensure high operational value.[1]

While Agile principles have been applied more often to software development efforts, these methodologies and the EVM implementation described herein are applicable to a wide range of development and production efforts.

The demand for responsiveness, efficiency, and collaboration extends to all aspects of system development and delivery, starting with negotiation of the contract, applicable Contract Data Requirements Lists (CDRLs), and effective implementation of Earned Value Management (EVM). Both EVM System (EVMS) and agile methods need to consider flexibility for these changing demands while enabling schedule and cost performance measurement and timely change control to the Performance Measurement Baseline (PMB). This need creates an opportunity for embracing the application of EVM with agile methods on product and system development and delivery programs. The intent of this Guide is to describe best practices for integrating Agile principles while maintaining compliance with the EIA 748 Standard for EVM Guidelines. In addition, this Guide describes best practices of integrating Agile principles with scheduling and performance measurement to proactively manage the total expected cost at completion of the contract even if EVM is not explicitly required.

Agile methods provide a disciplined process for defining work and tracking the progress of this work. Integrating Agile performance data with the EVM system provides a vertical integrated view of scope, schedule, and cost, from development activities to program performance measures.

This Guide discusses practices drawn from lessons learned by multiple aerospace and defense firms successfully integrating Agile and EVM. None of the best practices discussed in this Guide negate any of the fundamental practices described in EIA 748, the Department of Defense (DoD) Earned Value Management System Interpretations Guide (EVMSIG) or a corporate system description. The best practices in this Guide are meant to be details of clarity typically documented in an EVM system description and/or as supplemented in a Program Management Plan / Program Procedure. The content in the Guide is organized into the following sections and appendices outlined in the table below.

Agile Guide Section	Description
1. Agile Program Planning	Overview of the Agile planning process and levels. Includes an illustration of the Agile planning levels and their relationship to EVM processes.

[1] Defense Agile Acquisition Guide, Pete Modigliani and Su Chang, Mitre Corporation, March 2014

Agile Guide Section	Description
2. The Performance Measurement Baseline (PMB) and Agile Methods	Discusses recommended approach for the Work Breakdown Structure (WBS), Integrated Master Plan (IMP), and Integrated Master Schedule (IMS) for Agile programs. Also discusses freeze period considerations.
3. Structures for Performance Metrics	Discusses best practices to plan and then measure work package earned value performance using Agile progress measures. Also discusses using Agile metrics to forecast the estimate to complete.
4. Managing Baseline Change on Agile Programs	Discusses best practices to manage baseline changes on Agile development programs also using EVM. Provides example baseline and forecast change scenarios to illustrate recommended approaches.
5. Contracting for Agile and EVM	Discusses contracting best practices when Agile and EVM apply.
Appendix A – Agile Data Dictionary	Provides Agile terminology definitions.
Appendix B – Examples of Progress Tracking Charts with Agile and EVMS	Illustrations of charts.
Appendix C – References	A list of Agile and EVM references for more information about the topics in this Guide.
Appendix D – Product Roadmap, Release Planning, and Rolling Wave Planning Products	Elaborates on the Product Planning (Section 2.1), Release Planning (Section 2.2), and Integrated Master Schedule (IMS) (Section 3.3) discussions.
Appendix E – IBR Considerations	Provides a framework for conducting an Integrated Baseline Review on an Agile program.
Appendix F – Agile RFP Language	Sample language to include in RFP for agile development
Appendix G – Using Agile Metrics to Support Analysis and Forecasting	Examples of standard metrics used to track agile information
Appendix H – Agile/EV Guide Contributors	Acknowledgements
Appendix I - Acronyms	Acronyms unique to this Guide not found in the NDIA Master Definitions List for IPMD Guides

1 Agile Program Planning

Work planning in Agile development is preceded and guided by prioritization of business value defined collaboratively by the customer, contractor, and other stakeholders. Business value is expressed in terms of the contract goals and functionality needed by the customer. The functionality is typically extended from the CWBS into the decomposition of a hierarchical scope structure of broadly defined Epics/Capabilities, each of which is further decomposed into more specific Features. In turn, each Feature(s) is further decomposed into a set of lower work items or detailed User Stories (Stories).[2] This hierarchical decomposition of product functionality provides broad definition of the project scope at program startup while reserving details to subsequent, periodic planning events. The hierarchy of Epics/Capabilities, Features and User Stories (Stories) are listed in prioritized order in the Product Backlog. The Product Roadmap displays the delivery of the Product Backlog in a series of build releases.

1.1 Product Planning

Product Planning is a continuous control activity that establishes the Product Backlog and Product Roadmap in accordance with the awarded contract scope. The focus of Product Planning is the creation and maintenance of the Product Backlog and Product Roadmap at the Epic/Capability level. The Product Backlog is the master list of functionality that is desired in the product and any other elements needed to produce the product, even if not in the final product. The Product Roadmap reflects the prioritization of the items on the Product Backlog by the customer, contractor, and other stakeholders based on business value and dependencies. The Product Roadmap may precede, inform, or supplant the development of an IMP, and informs the top-level plan of the IMS. Due to its architectural significance, Product planning (agile) and planning, scheduling and budgeting (EVMS) starts with an integrated product and scope hierarchy that extends from a product oriented WBS to the Epics/capabilities in the product backlog as prioritized in the product roadmap. Product Planning is performed throughout the life of the program to refine and update the Product Backlog based on changes adopted from regular, periodic assessment of customer needs, within the current scope of the authorized contract. The Product Owner (PO) role is responsible for managing Product Planning through the life of the contract, in collaboration with Customer representatives. The outcome of continual Product Planning throughout the lifecycle of a contract is the final product and corresponding refinements to the contractual requirements.

1.2 Release Planning

Release[3] Planning is the activity most closely related to developing the Integrated Master Schedule (IMS) and subsequent Rolling Wave planning represented in the IMS. Release Planning encompasses the product goals for the next planning horizon or time-block of work, typically a 3 to 6-month window of time. During Release Planning the team refines the Product Backlog and decomposes Epics/Capabilities into Features and candidate Stories that are to be delivered in the next Release based on Customer priority, dependencies, and available capacity.

[2] For the purposes of this Guide, a general framework of decomposition will be used to include the tiers of Epic/Capabilities, Features, and User Stories. Other decomposition approaches exist, and care should be taken to understand a program's specific lexicon and decomposition approach.

[3] For the purposes of this Guide, "Release" is a concept and generic reference to a block of time containing multiple sprints. Each program will have a specific definition of Release documented in the Agile Implementation Plan. In practice, a "release" can be an internal release across environments or a formal release to an operational system for users, which will be defined by each program. Assume the term "Release" is followed by "to" for clarification of the purpose.

The IMS activities / tasks planned and scheduled should be 'feature-based' to ensure that the IMS is product driven rather than modelled as time-boxes. The Product Backlog and Product Roadmap identifying required product functionality are inputs to the Release Plan. Selected Features define what the product must do and when the functionality will be delivered within the Release. A Feature is typically sized to be completed within one release[4] (consistent with the specific Agile implementation for internal releases for test, to operations or release on demand). The candidate Stories associated with a feature suggest how the functionality of the feature will be completed. It is within Release Planning that IMS planning occurs. As a result of release planning and detail planning, the prioritized feature(s) in the roadmap comprise the work package scope and the corresponding feature-driven IMS activity/task(s). During Release Planning either Features or higher-level Epics/Capabilities should comprise the Planning Package(s) scope, consistent with the Product Roadmap and the program's Product Hierarchy. The Agile framework implemented for decomposition does not change the requirement that Work Packages are comprised of work scope, baseline period of performance, budget, Earned Value Technique (EVT) and objective exit criteria. The work scope of the Work Package is directly traceable to the product hierarchy as detail planned within the Control Account scope.

On large-scale programs with multiple Scrum teams, the Release Planning meeting includes coordination of Feature planning among the various POs to achieve a release plan that supports the required product deliveries and overall goals of the program.

The Control Account Manager(s) (CAM(s)) may participate in the Release Planning event to complete the IMS planning. The purpose is to validate the detail plan of the next increment of work or rolling wave in parallel with Release Planning to implement any change management immediately upon completion of the Release Planning Event. CAMs will be compliant with the Earned Value Management System Description (EVMSD) Freeze Period. Work Packages typically align with individual Features, logical groups of related Features, or Epics/Capabilities. Dependencies across Features are identified and documented. The budget for each Work Package is allocated from the authorized budget for the Planning Package/Control Account in terms of hours and resources.

1.3 Sprint Planning

Sprint Planning is the activity in which product goals are defined for the next Sprint. In support of these goals, Scrum teams commit to the completion of specific Stories representing lower-level work items. A Sprint is a fixed time period, typically two to four weeks in duration. The Release duration is expressed as a number of Sprints of equal length, aligning with the start of the first Sprint in the Release and the end of the last Sprint in the Release.

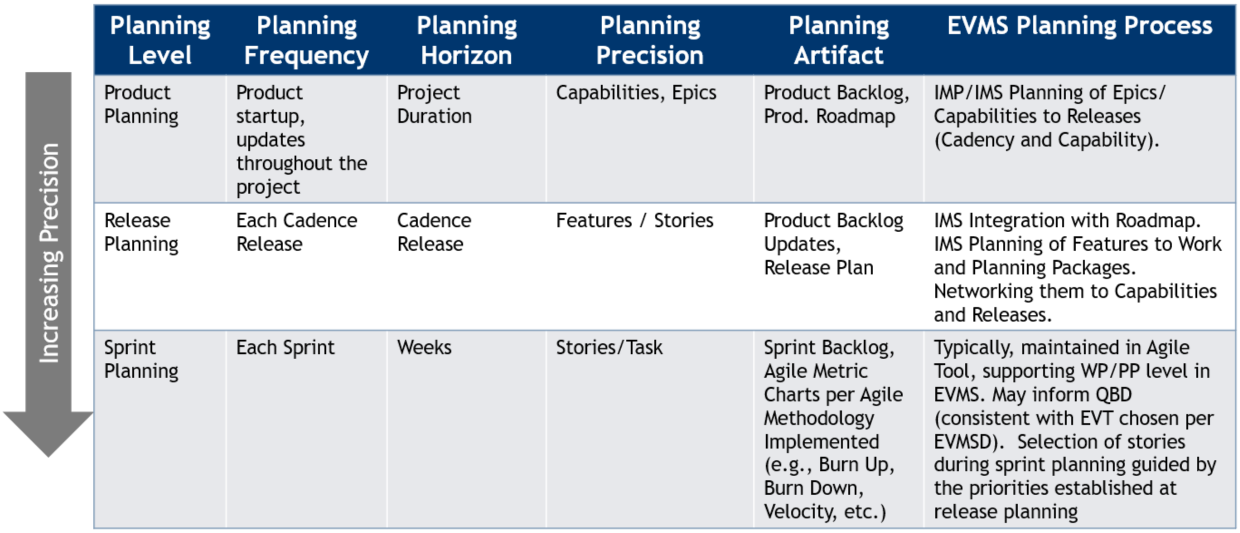
Features and initial Sprint stories are prioritized and sized at release planning and decomposed into Stories which are planned prior to the start of Sprints. In accordance with a Corporate EVMSD and during rolling wave planning that coincides with Release Planning, a CAM selects the applicable EVT for a work package(s) comprised of Feature(s) and subordinate Stories. If / when the EVT chosen requires Quantifiable Backup Data (QBD) (as uniquely defined by the Corporate EVMSD), then a CAM may use the decomposed Stories in a QBD. Stories are typically sized to complete within one sprint. During the Sprint Planning event, the Stories may be refined. These Stories are prioritized by the Product Owner. The Sprint Planning process is attended by the CAM and is completed before work starts on the implementation details of Features in the current Sprint. The CAM will evaluate the impacts of the refinement of the stories on the work package and complete any change actions required by the EVMSD. The Scrum team's list of

[4] Appendix D will expand on this discussion

Stories from that Sprint comprises the Sprint Backlog. The Scrum Master is responsible for facilitating Sprint Planning.

Stories are completed during each Sprint; progress is determined by the completion status of the planned Stories for the Feature in accordance with the CAM's assigned work package EVT. (See Section 3.0 for more information on progress determination.) Once a Sprint begins, the Stories (and if applicable, story points) within that Sprint do not change. At the next Sprint Planning event, starting at the Sprint Planning horizon, all remaining work to complete the feature is reevaluated and dispositioned. It is expected Sprint over Sprint that candidate Stories from Release Planning (the implementation details to complete feature scope) will evolve and change as the team continually learns, adapts and documents Feature completion.

The tiered Agile planning levels are shown in Table 1-1. The hierarchy of the Planning Artifacts is described in more detail in Section 2.2. The Work Breakdown Structure, used for Agile programs, is described in more detail in section 2.1.


Increasing Precision

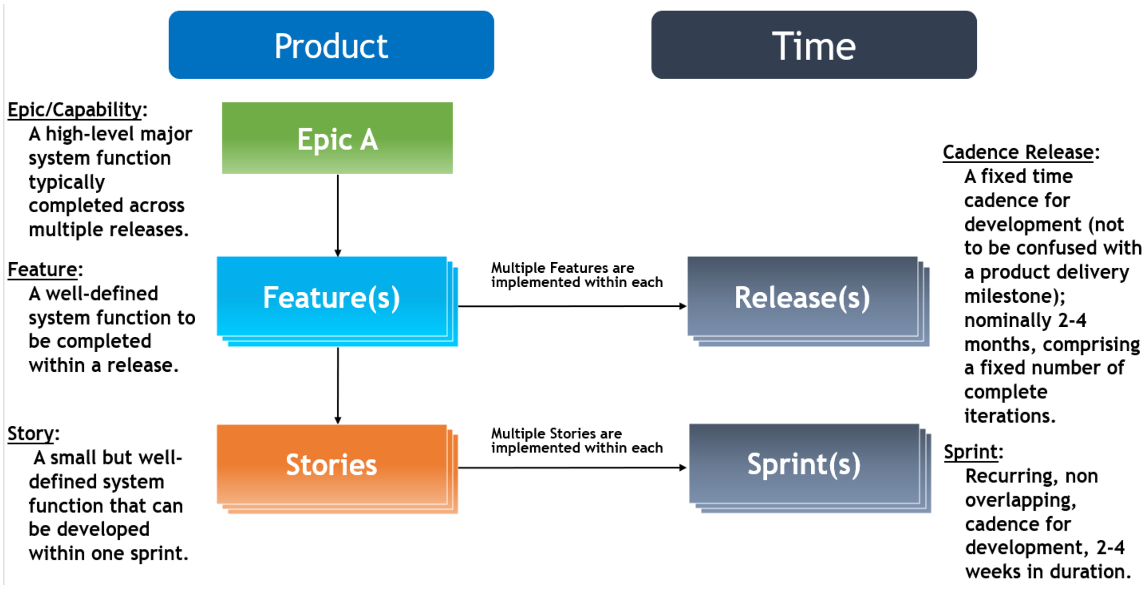
Planning Level	Planning Frequency	Planning Horizon	Planning Precision	Planning Artifact	EVMS Planning Process
Product Planning	Product startup, updates throughout the project	Project Duration	Capabilities, Epics	Product Backlog, Prod. Roadmap	IMP/IMS Planning of Epics/ Capabilities to Releases (Cadency and Capability).
Release Planning	Each Cadence Release	Cadence Release	Features / Stories	Product Backlog Updates, Release Plan	IMS Integration with Roadmap. IMS Planning of Features to Work and Planning Packages. Networking them to Capabilities and Releases.
Sprint Planning	Each Sprint	Weeks	Stories/Task	Sprint Backlog, Agile Metric Charts per Agile Methodology Implemented (e.g., Burn Up, Burn Down, Velocity, etc.)	Typically, maintained in Agile Tool, supporting WP/PP level in EVMS. May inform QBD (consistent with EVT chosen per EVMSD). Selection of stories during sprint planning guided by the priorities established at release planning

Table 1-1: Agile and EVMS Planning levels and inter-relationships.

When Agile is a methodology conveyed on DoD contracts, it may be appropriate to align an engineering change process with the agile change process. The integration of the program management (including EVMS), engineering and agile methods should ensure that the program cadence, planning, pace of change and change control are aligned to ensure that all types of change are implemented across all functional disciplines. Between managing change through the Agile process and the engineering process, there is transparency and documentation of the product and corresponding requirements.

1.4 Product and Time Hierarchy

Figure 1-1 illustrates the two separate hierarchies used in Agile, for Product and for Time. Separate Product and Time hierarchies allow work to be planned by periodically assigning appropriately sized products into selected Releases or Sprints.


Product
Time
Epic A
Feature(s)
Stories
Multiple Features are implemented within each
Multiple Stories are implemented within each
Release(s)
Sprint(s)

Epic/Capability: A high-level major system function typically completed across multiple releases.

Feature: A well-defined system function to be completed within a release.

Story: A small but well-defined system function that can be developed within one sprint.

Cadence Release: A fixed time cadence for development (not to be confused with a product delivery milestone); nominally 2-4 months, comprising a fixed number of complete iterations.

Sprint: Recurring, non overlapping, cadence for development, 2-4 weeks in duration.

Figure 1-1: Hierarchy of Agile Products and Agile Timeboxed Elements and Relationships illustrates the two hierarchies in Agile: Product, based on WBS, and Time, the rhythm for executing work.

2 The Performance Measurement Baseline (PMB) and Agile Methods

2.1 The Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) defines the program in terms of hierarchically related, product-oriented elements. The WBS is a product-oriented family tree (composed of hardware, software, services, data, and facilities) that displays and defines the product to be developed during the acquisition. [5] The WBS represents all the scope being worked and work being performed on a program, both level of effort (LOE) (such as program management) and discrete deliverables. For programs using Agile methodologies, the WBS should align with the Product Backlog. The Product Backlog focuses on completed products that provide measurable customer value implemented in Epics and Capabilities.

The WBS is integrated with the product hierarchy on the program, as the product oriented WBS is extended. Control Accounts are developed to ensure effective planning and decomposition of epics, capabilities, features, etc. Agile development Releases are fixed-length blocks of time, which are used for Product Roadmap time phasing—they do not capture work or represent product, and therefore should not appear in the WBS.

While there is no single standard template for a WBS, MIL-STD-881-Current Version is a common reference used in DoD systems and automated information systems. The WBS outlined in MIL-STD-881-Current Version Appendix J is selected to create a template that illustrates the application of Agile development techniques. MIL-STD-881-Current Version allows considerable tailoring for specific programs.

There are options for what defines the Agile product beyond the necessary Epics/Capabilities, as described in Table 2-1 for a software product (e.g., Information Systems (IS) / Defense Business Systems (DBS)). Table 2-1 does not attempt to provide a comprehensive picture of the WBS, but instead focuses on the core Agile software products.

[5] MIL-STD-881 (F is the most recent version as of the publication of this Guide)

WBS	Task Name	Notes
1	**Information Systems (IS)**	
1.1	**IS Prime Mission Product Release X**	Multiple elements at this level would be appropriate if the customer views major deliveries as independent products and desires a WBS organized around them (e.g., the deliveries are viewed as separate projects). The key point is that elements at this level have no relationship with the Agile cadence "release".
1.1.1	**Custom Applications SW 1..n**	
1.1.1.2	**Subsystem SW CSCI 1..n**	Appropriate if Computer Software Configuration Items (CSCIs) are viewed as key products (with Epics/Capabilities contained within them); may be at L4 or not present at all (as explained below)
1.1.1.2 or 1.1.1.2.1	**Agile Epic/Capability 1..n**	Would occur at Level 4 or 5. When Epics/Capabilities are the primary organizing method for products then these could be at L4 (preferred). Alternatively, Epics/Capabilities could be viewed as products within CSCIs. Epics/Capabilities are often preferred over CSCIs in the WBS, as Epics/Capabilities are organized around system functionality (value add, end user products) while CSCIs are organized around the internal architectural structure of the system, which doesn't necessarily align directly with usable functionality and customer value.

Table 2-1: Example WBS, indicating WBS Number, Task Name, and comments on how best to apply in a program with both Agile and EVMS.

Another example WBS shown in Table 2-2, derived from MIL-STD-881-Current Version Appendix B on Electronic Systems/Generic Systems, indicates how Agile is incorporated into a program involving both software and hardware development. Again, Table 2-2 does not attempt to provide a comprehensive picture of the WBS; instead, it focuses on the core Agile developed products. The example below is meant to be adapted based on the contract awarded and not all the exact line items depicted will necessarily convey one to one to an actual execution WBS.

There are programs with a hybrid WBS, where sections of the WBS will be Agile and other sections will not be Agile. Take, for example, in the Program Management section of the WBS, the Program Management team may not be utilizing an Agile methodology to plan. In this case, rely on non-Agile approaches to create the WBS.

The key for building the WBS and establishing the performance measurement plan, is to be consistent within each individual section of the WBS. Meaning if the Prime Mission Product (1.1 WBS consistent with MIL STD 881) is to be capability and feature based, ensure that each lower level of all WBS items within 1.1 are consistent at the feature level. It is recommended to not co-mingle Agile (feature) and non-Agile (not features) WBS items within the same WBS section, in this example within WBS 1.1.

Again, the WBS samples provided are not meant to present a comprehensive picture. Use the samples and in practice apply Agile thinking based on the Agile Implementation Plan and Program Management Plan of the specific contract.

WBS	Task Name	Notes
1.0	**Electronics System**	
1.1	**Prime Mission Product**	
1.1.1 (L3) and/or	**Product 1 . . n**	For products that are hardware only or hardware and software combined as the key deliverables.
1.1.1 (L3)	**Software Product 1 . . n**	For software applications that are viewed as key products/deliverables. Choose the appropriate Level 3 (L3) for the project.
1.1.1.X (L4) and	**Agile Epic/Capability 1 . . n**	When Epics/Capabilities are the primary organizing method for products then these could be at L3. Epics/Capabilities are often preferred over CSCIs/Subsystems in the WBS, as Epics/Capabilities are organized around system functionality (value add, end user products) while CSCIs/Subsystems are organized around the internal architectural structure of the system, which doesn't necessarily align directly with usable functionality and customer value. Each Capability L4 WBS Includes all systems, and development and integration of each Capability on its own.
1.1.1.Y (L4)	**Agile Epic/Capability Systems, Integration and Test**	Includes all systems, integration and test activities (in a host environment) associated with PMP Software product (L4). Also includes DO-178/CSCI requirements-based testing activities not completed within each Capability defined in 1.1.1.x. (Note: WBS not needed if all effort covered within each 1.1.1.x, or in 1.1.Z (PMP Integration Assembly, Test and Checkout)).
1.1.Z (L3)		PMP integration assembly, test and checkout (e.g., includes system/ARP-4754 verification) of all Products.

Table 2-2: Example WBS, indicating WBS Number, Task Name, and comments on how best to apply in a program with both Agile Methods and EVMS Integration.

2.2 Integrated Master Plan (IMP)

The IMP and IMS are fundamental management tools that are critical to performing effective planning, scheduling and execution of work efforts. When executing a project with both Agile and EVM practices, both the IMP and the IMS may require special attention and should be considered for tailoring to the project scope document identified in the solicitation. The IMP precedes the IMS and draws from the Statement of Work (SOW), Statement of Objectives (SOO), Concept of Operations (CONOP) and the product oriented WBS. In Agile, the IMP Program Events may consist of Epics/Capabilities which align with major customer milestones.

The IMP may be developed in conjunction with Product Planning and align with the Product Roadmap. The IMP defines all major customer milestones and deliveries. It is time phased, showing the initial order of the Capabilities produced by the program. Capability completion corresponds to the IMPs Significant Accomplishments and the Capability acceptance criteria constitutes the Accomplishment Criteria. Figure 2-1, Agile IMP Event to EVMS Hierarchies, is an example graphic illustrating how the IMP, EVM, and Agile elements are vertically and horizontally traceable in a single framework integrating Agile and Earned Value Management.

IMP events that are compatible with Agile programs include planned customer deliveries aligned to customer milestones. Initial delivery of completed work products, and later deliveries, are aligned with key mission milestones. For example, if building a space vehicle system, the control system events include deliveries to support launch, Initial Operational Capability (IOC), and Full Operational Capability (FOC). The IMP events may also include customer demonstration events, e.g., formal demos of an initial flight demonstration.

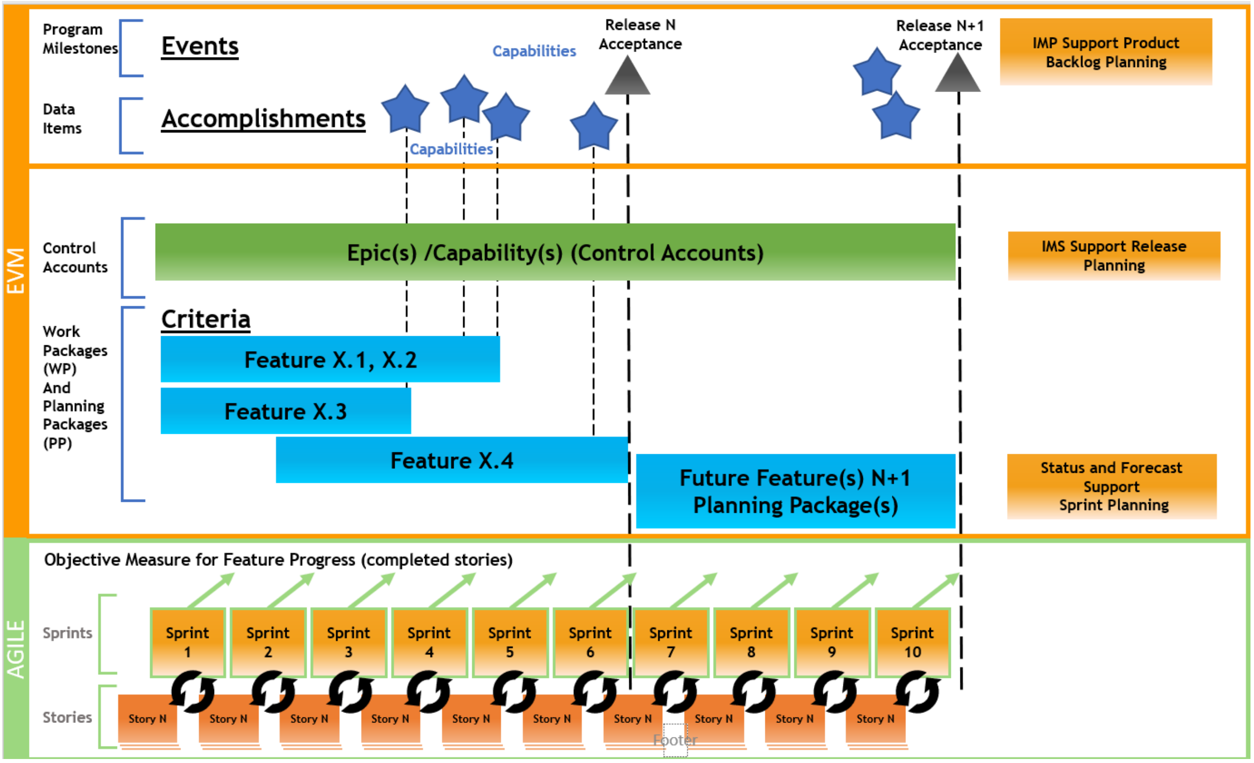


Figure 2-1: Agile IMP Event to EVMS Hierarchies. In this example, IMP events are equivalent to Customer Releases, with Significant Accomplishments and Accomplishment Criteria representing delivered capabilities delivered in Work Packages where Features are implemented.

2.2.1 Agile Events Identified within an IMP

While the Federal Acquisition Regulation (FAR) may not require something specifically known as a "Critical Design Review" (CDR) there are other governing DoD Systems Engineering policy instructions that may still require specific reviews. Some Department of Defense instructions DoDI 5000.02 and DoDI 5000.88 require acquisition program managers to establish events and associated engineering review activities to assess the maturity of a system. These references define a Software Development Life Cycle (SDLC) that suggests milestones and each of those milestones has a purpose. As Agile implementations mature and Agile training is received across the DoD, it is becoming more common to for engineering and delivery approaches to align with the cadence-based solutions for the benefit of the mission.

An Agile development contract must consider the purpose of the activities and milestones and in coordination with your customer, adjust them according to the Agile activities and milestones relative to the product being developed. For example, if the program or contract will do Release Planning, identify if it represents an IMP event or accomplishment that needs to be tracked. Focusing on the Agile approach with appropriate insight into an accomplishment will enable teams to identify relevant milestones for elevation. Ensure the comprehensive technical approach is reflected in the IMP.

The IMP hierarchy outlines what will be done to demonstrate the completion of the program including:

- Define Event: Logical or product maturity points, consider representing a historical single event by a block of time in the schedule to iterate on maturing the system for a particular demonstration of the evolving architecture (“a CDR season”).
- Define Accomplishment: Logical component of the event or product, which demonstrates what specific items will comprise the specific “event”.
- Define Criteria: Logical smaller segments of effort demonstrating how specific accomplishments will be completed.

If the entire contract is for Agile development and related functions only, and the Product Roadmap represents the comprehensive technical approach, it may be possible to remove the IMP as a contractual requirement and replace it with the Product Roadmap.

2.2.2 Agile Project Nuances for IMP Application

Specifics of the project scope relative to the government’s broader programmatic effort may need to be considered in the IMP. Items for consideration include:

- Is there a formal IMP requirement that the government is tracking? Which piece of the larger effort is your contract supporting? Review the SOO for IMP structure and content requirements. Coordinate with government counterparts to understand how each piece fits. The recommendation is to utilize the IMP concepts in a logical way to support reporting insight.
- For systems to be deployed, DoDI 5000.02 requires multiple gates leading to the final Full Deployment Decision (FDD). Understand where your program is within the system development lifecycle. Negotiate with your customer the appropriate events and corresponding accomplishments needed for the deployment decision to be made. Refer to section 5.6 of this Guide for information on adapting milestone reviews on Agile programs.
- The Product Roadmap is part of the IMP, but not necessarily the entire IMP. The IMP / IMS represents the entire scope, even non-development scope, from contract award to contract completion. If you are attempting replace a project IMP with a Product Roadmap, you will need to review and ensure that appropriate scope coverage, across all areas, exists and allows for effective visibility into the required events and accomplishments.

2.3 Integrated Master Schedule (IMS)

As stated in Section 1, Epics/Capabilities are decomposed into Feature and Story (User Story) entities. An Epic/Capability delivers one or more Features, and a Feature is implemented by one or more Stories. On larger programs, one or more “sub-Epics/Capabilities” may exist between Epics/Capabilities and Features to manage the product decomposition to usable sizes, hence the chosen term in this Guide of Epic/Capability. Features are sized to fit within Agile Releases and represent significant pieces of the delivered product.

The product hierarchy at the feature level should be the lowest level of an IMS. Considerations for scaling the feature-driven work below the feature level (and above the story level) in the IMS is a planning factor to ensure complete performance management and control is achieved. If there is a need to track a subordinate level of detail, then the Feature scope must be defined at a lower level of detail so that the desired level of tracking and IMS logic is supported. Stories serve as the implementation details of the Feature and are more efficiently maintained by Scrum teams outside the IMS in an Agile development tool.

At program start, an initial Product Roadmap with work product functionality will be created showing a plan for Epic/Capability and Feature development across the Releases. Product Roadmaps must consider architectural and product dependencies as well as customer milestones. The IMS content, Features and their associated start/end dates and dependencies will be finalized through Rolling Wave planning prior to the start of the execution of the associated Release. Figure 2-2 shows a Rolling Wave Planning process in the IMS; Release 1 planned, while the content for the next Releases still contained in Planning Packages. The content of these remaining Planning Packages will be refined in subsequent Rolling Waves.

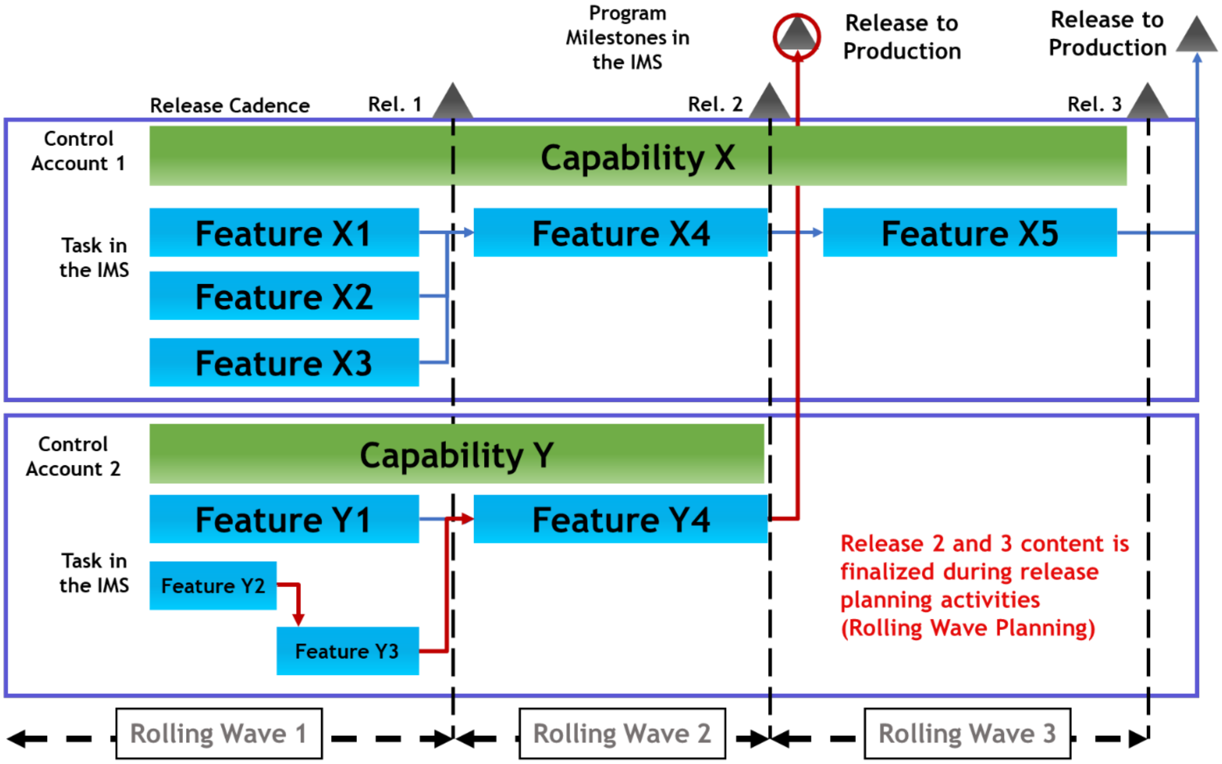
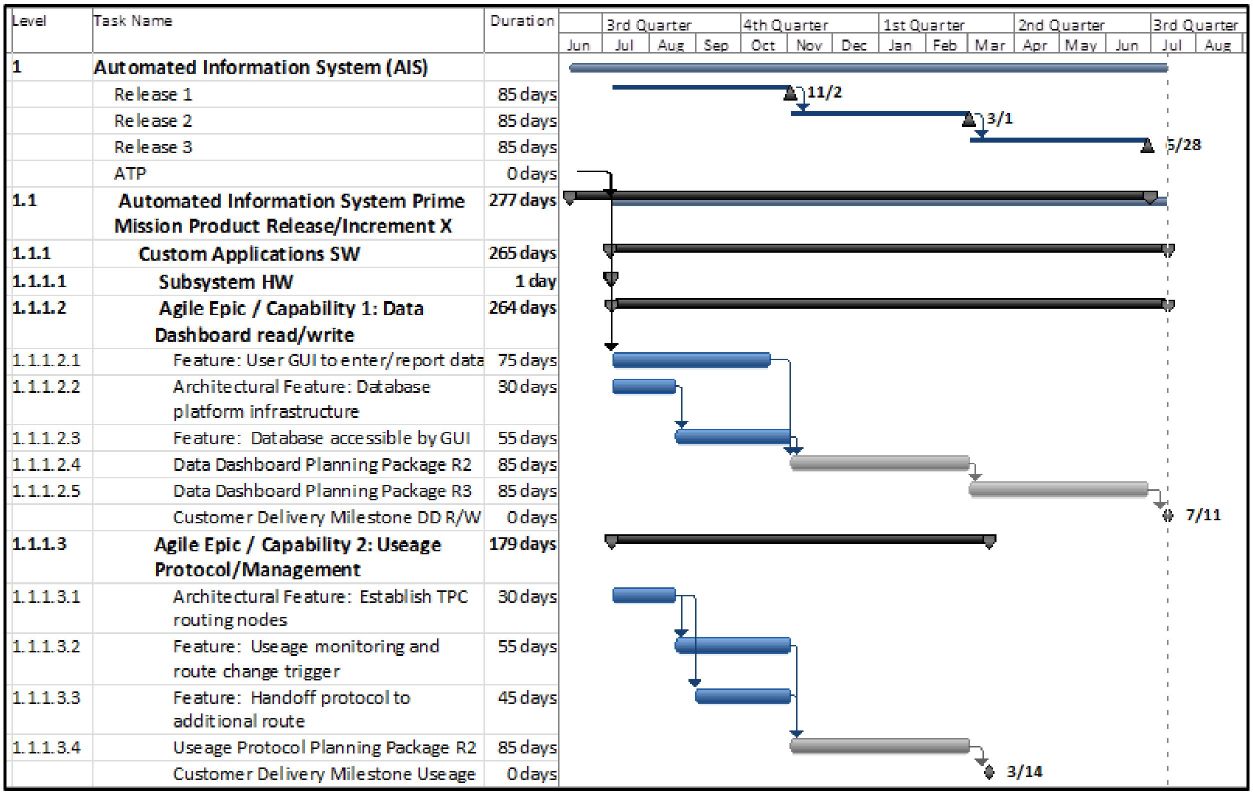


Figure 2-2: Illustration of Rolling Wave Planning in an IMS

Stories implement the Features in the IMS and are linked to Features in the Agile management tool. Work package scope is comprised of feature(s) as prioritized in Product backlog and planned in the Roadmap and the IMS. The IMS work package and/or activity/task, as applicable, may include an IMS reference (e.g., work package ID) that links relationship to the feature(s). This traceability provides the needed visibility to Program Management for the BCWS to *objectively assess accomplishments at the work performance level* in accordance with EIA-748-Current Version, Page 1.

Features may be longer in duration compared to programs not using the Agile methodology. This is suitable if the task reflects the work, possesses accurate network logic, and is backed up by Agile-based QBD as required based on EVT chosen. Feature Duration should consider the expected time to complete the effort and not automatically be planned to span a specific time-box. Calculating the fraction of completion of stories created to implement a Feature's scope of effort provides a recommended method for assessing credit, by dividing total completed Stories by total planned Stories for that Feature. Specifically, full credit is taken upon Story completion (100%) to mark progress towards Feature completion. Other methods for claiming progress of completed scope of effort are outlined in Section 3.3.

Example IMS tasks and subtasks are shown in Figure 2-3 below. These correspond to Control Accounts (CAs) and Work Packages. Work Packages align with a single Feature or group of related Features. Figure 2-3, an example of an IMS subset, is based on the example WBS in Table 2-1. It shows part of a program with Releases of 85 working days. Two Epic/Capabilities are developed, each requiring three Features that would each trace to a Work Package, plus Planning Packages assigned to future Releases. The Release Milestones are fixed dates, constrained by date or "timeboxed", and are not dependent on other IMS tasks..



Level	Task Name	Duration
1	**Automated Information System (AIS)**	
	Release 1	85 days
	Release 2	85 days
	Release 3	85 days
	ATP	0 days
1.1	**Automated Information System Prime Mission Product Release/Increment X**	**277 days**
1.1.1	**Custom Applications SW**	**265 days**
1.1.1.1	**Subsystem HW**	**1 day**
1.1.1.2	**Agile Epic / Capability 1: Data Dashboard read/write**	**264 days**
1.1.1.2.1	Feature: User GUI to enter/report data	75 days
1.1.1.2.2	Architectural Feature: Database platform infrastructure	30 days
1.1.1.2.3	Feature: Database accessible by GUI	55 days
1.1.1.2.4	Data Dashboard Planning Package R2	85 days
1.1.1.2.5	Data Dashboard Planning Package R3	85 days
	Customer Delivery Milestone DD R/W	0 days
1.1.1.3	**Agile Epic / Capability 2: Useage Protocol/Management**	**179 days**
1.1.1.3.1	Architectural Feature: Establish TPC routing nodes	30 days
1.1.1.3.2	Feature: Useage monitoring and route change trigger	55 days
1.1.1.3.3	Feature: Handoff protocol to additional route	45 days
1.1.1.3.4	Useage Protocol Planning Package R2	85 days
	Customer Delivery Milestone Useage	0 days

Figure 2-3: Example of an IMS subset, based on the WBS example in Table 2-1.[6]

IMS considerations drawing from the IMS example in Figure 2-3:

- Networking between Work Packages shows dependencies across product Features. In Figure 2-3, the Architectural Feature of level 1.1.1.2.2 for a *Database platform infrastructure* must complete before the Feature of *Database accessible by GUI,* level 1.1.1.2.3, can be started. Other dependencies include test equipment, power supplies, hardware, or simulation software, as well as dependencies between the to-be-developed products. The cross-functional Agile teams should minimize dependencies/handoffs

[6] Note that Figure 2-3 utilizes MIL-STD-881 Rev. D, whereas Table 2-1 reflects MIL-STD-881 Rev. C. Adapt this guidance based on later MIL-STD-881 releases.

between teams based on disciplines (e.g., systems engineering, development, and test). To the extent that product level dependencies still exist, they must be modeled in the IMS to establish a critical path.

- The IMS is baselined prior to any work for the Release content being started. Release Planning in the IMS defines where the IMS is synchronized with the Agile plan, prior to execution of the work.
- IMS progress is informed by Agile progress tracking reports through burn-up or burn-down reports. See Appendix G, Using Agile Metrics, for more details.
- In the IMS, work or planning package tasks can span the duration of a Release given no significant inter-CAM handoffs or major Feature-to-Feature dependencies will be modelled.
- Releases and/or Sprints are timeboxes that start and end at specified times. They may be included in the IMS, for reference purposes to inform the customer of potential demonstrations and the Agile cadence. There should not be dependencies between these time-boxes and work packages. Time boxes are not part of the critical path and do not represent scope, budget or forecast.

2.4 PMB to Product Hierarchy Alignment

Figure 2-4 illustrates a typical, not mandatory, EVMS to Product Hierarchy alignment. The figure illustrates that traceability between the EVMS, and Agile hierarchies is defined and maintained throughout the program, aligning Scope and Budget via assigning sized Agile Products to CA, WP and PP within the EVMS. Sizing of Agile Products is based on complexity of effort and is calibrated to equate to resources planned for each product. See Section 5 for more detail and an illustration of how scope/budget alignment is maintained within both hierarchies.

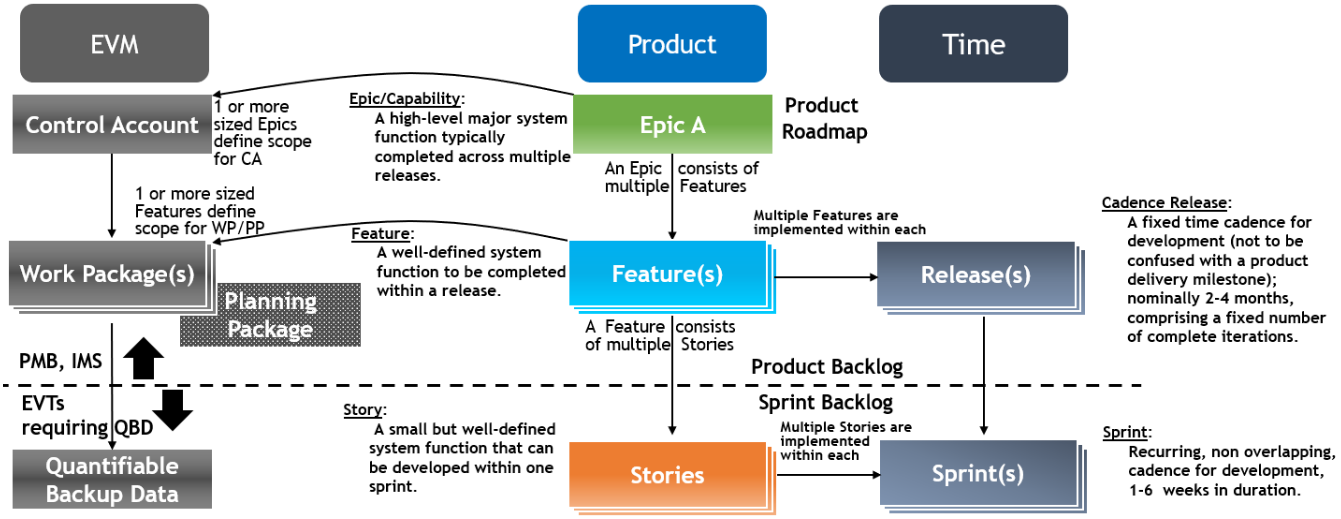

EVM
Product
Time
Control Account
1 or more sized Epics define scope for CA
Epic/Capability:
A high-level major system function typically completed across multiple releases.
Epic A
Product Roadmap
An Epic multiple
consists of Features
1 or more sized Features define scope for WP/PP
Feature:
A well-defined system function to be completed within a release.
Multiple Features are implemented within each
Cadence Release:
A fixed time cadence for development (not to be confused with a product delivery milestone); nominally 2-4 months, comprising a fixed number of complete iterations.
Work Package(s)
Planning Package
Feature(s)
Release(s)
A Feature of multiple
consists Stories
PMB, IMS
Product Backlog
EVTs requiring QBD
Sprint Backlog
Story:
A small but well-defined system function that can be developed within one sprint.
Multiple Stories are implemented within each
Sprint:
Recurring, non overlapping, cadence for development, 1-6 weeks in duration.
Quantifiable Backup Data
Stories
Sprint(s)

Figure 2-4: Typical alignment of EVMS to the Product Hierarchy, however, depending on program size and system description, other alignments have been observed in industry also. Note that traceability both within and between each hierarchy has been defined at program start at the CA/Epic/Capability and WP/PP levels, and for more detailed levels, at successive Release Planning/Rolling Wave Planning and Sprint Planning activities. The most important concept, as illustrated by the black dashed line, is establishing a clear line above which earned value is maintained (the feature is the lowest IMS level), and below which Agile methods are preserved that underpin and support appropriate progress assessment.

3 Structures for Performance Metrics

This section describes current best practices in industry for how to plan and measure program earned value performance in Work Packages and CAs, using Agile progress measures.

3.1 Work Authorization and Control Account Plan

For purposes of this process illustration, Control Account (CA) scope corresponds to Epics/Capabilities and their Features of the system. Product planning (agile) and planning, scheduling, and budgeting (EVMS) starts with an integrated product hierarchy that extends from a product oriented WBS to the Epics/capabilities in product backlog as prioritized in the product roadmap. If the CWBS is not extended down to the Epic level, then the Control Account work authorization scope description (and/or with traceability to the applicable agile tool artifact) is the bridge to define how the WBS and the Epics/Capabilities comprise the authorized work in a control account that is under configuration management control and baseline change control. The schedule for delivery of system functions results from the planned Release of working products, the span of control desired by program leadership, and other similar considerations. Thus, CA durations may vary from one-to-many Releases. However, it is recommended that CA scope correspond to a single Epic/Capability.

WPs are an element of control within CAs. Work package scope is comprised of one feature or a set of logically related grouping of features as prioritized in product backlog and planned in the IMS and Agile roadmap. A work package is the point where scope is planned, progress is measured, and earned value is assessed. It is recommended to align one Feature or at most a small set of logically related Features with a Work Package.

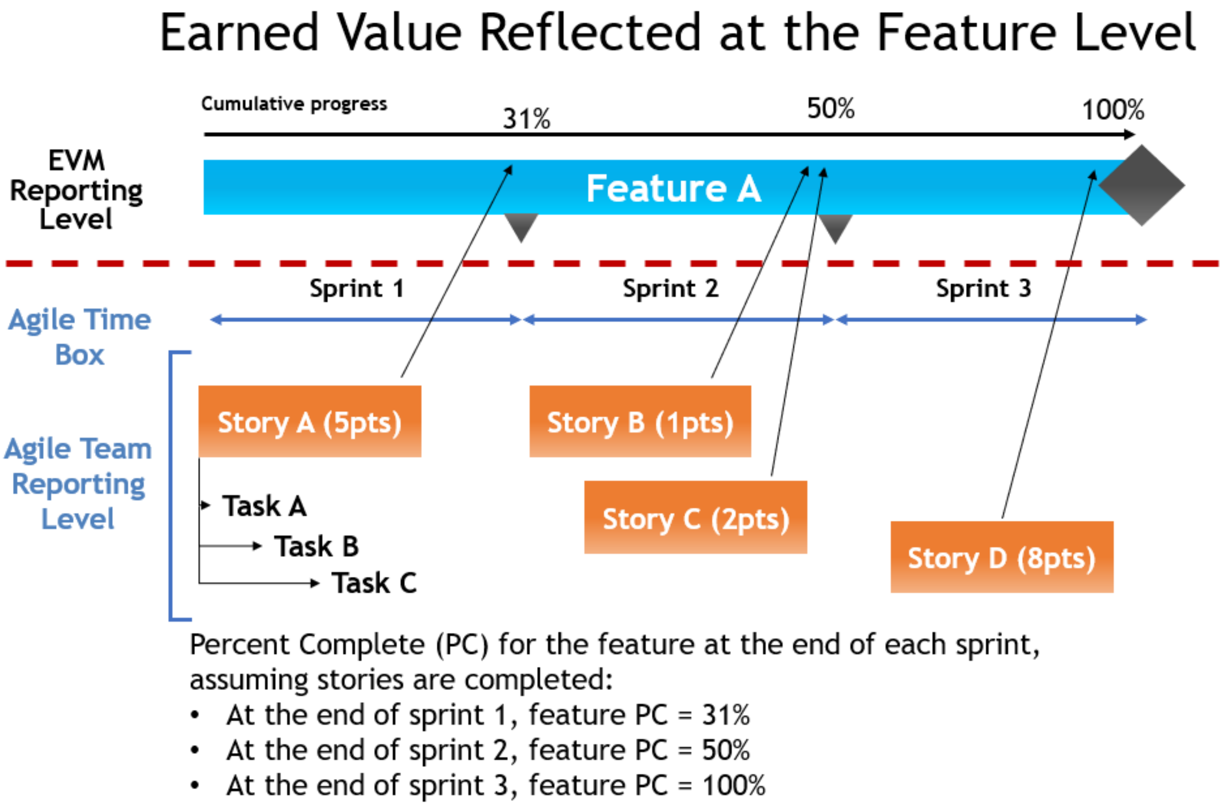
Consistent with existing EVM policy and practices, the Work Package contains the contractually authorized scope, schedule, and budget to be measured. On Agile programs, the feature in the Product Hierarchy (Figure 2-4) is typically aligned with the Work Package in the EVM hierarchy. Accordingly, it is recommended that the Features in the system contain the contractually authorized scope, schedule, and budget to be measured. The budget for the Work Package is determined by the estimated effort to complete the work scope in terms of hours and resources in relation to and within the parameters of the budget authorized to the control account. A single Work Package corresponds to one or more Features and the Work Package Period of Performance (PoP) may span the Release (a group of sprints) duration or only a part of it. While a WP may contain multiple Features, each Feature should be entirely contained within a single WP.

There should be a logical relationship between Features and Epics/Capabilities within the program's WBS, Control Account and Work Package structure. The Features are scheduled to be completed by a specific Release as represented in the IMS. As an example, the IMS in Figure 2-3 shows two Agile CAs: 1.1.1.2 and 1.1.1.3. The 1.1.1.2 CA, *Data Dashboard read/write*, contains Work Packages, such as 1.1.1.2.1, *User Graphical User Interface (GUI) to Enter/Report Data,* that each align to a Feature. Epic/Capability milestones align to CAs as well, an example of which is CA 1.1.1.3, *Usage Protocol/Management*, which aligns with a Customer Delivery Milestone on March 14th.

After initial planning, Work Packages are defined during program execution through a series of Rolling Wave or Release Planning cycles in concert with Release Planning Cycles. The Release Planning period is a fixed duration determined in Product Planning at the start of the program, and each WP should be scheduled to fit within one Agile Release. For Features beyond the current Agile Release, the scope may be in Planning Packages, which will be refined during future Release Planning cycles.

3.2 Aligning Agile Progress Metrics with Earned Value Reporting Levels

Figure 3-1 and Figure 3-2 are samples of Agile progress reporting used to status the PMB in the Earned Value Management System, based on the Earned Value Technique (EVT) chosen by the CAM for the Work Package, consistent with the Company's EVMSD. In Figure 3-1, the completion of Agile Stories (consistent with the acceptance criteria) determines the completion status for a Work Package (assumption, EVT is % Complete with QBD). There are alternate options to measure performance, which will be compliant with the Agile Methodology implemented and the EVMSD. In this sample, the completed Story Points of the Story work items of the Feature associated with the Work Package are used to calculate the Work Package EV percent complete.



Percent Complete (PC) for the feature at the end of each sprint, assuming stories are completed:

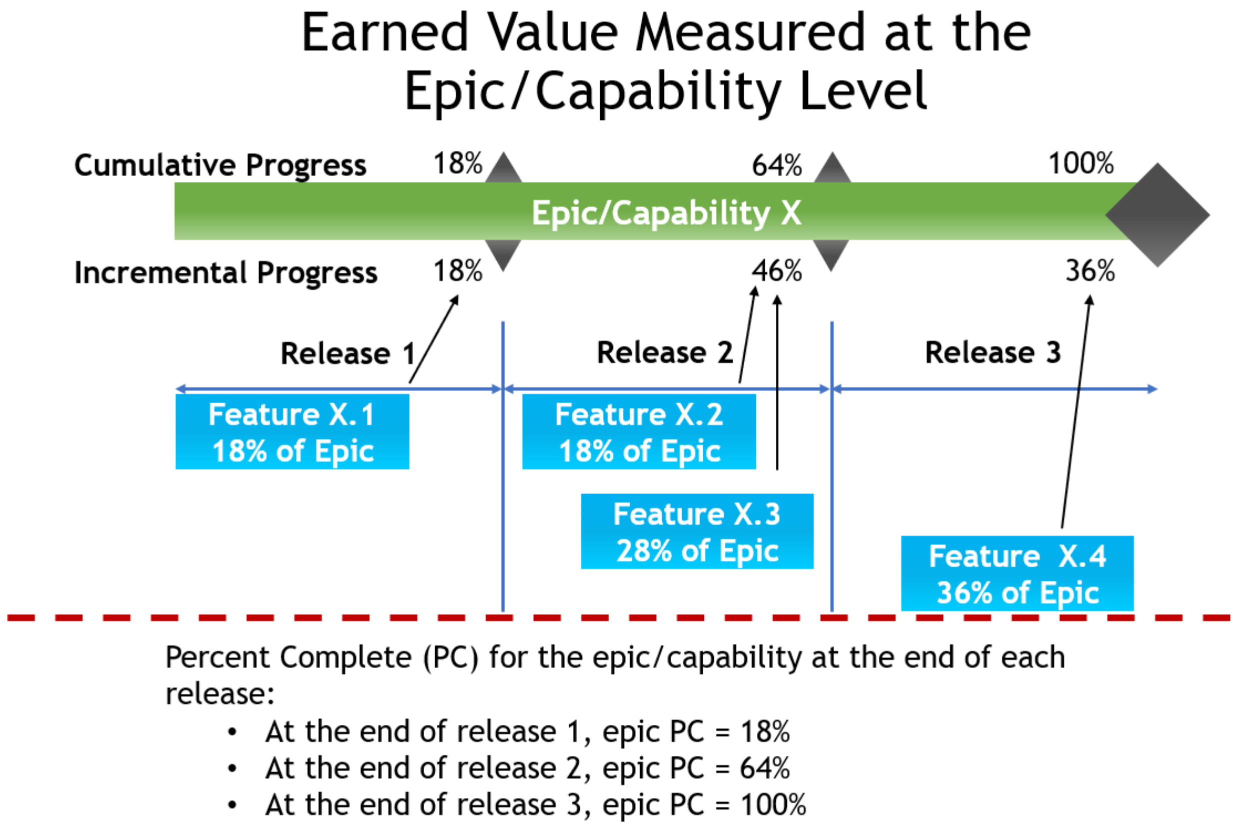
- At the end of sprint 1, feature PC = 31%
- At the end of sprint 2, feature PC = 50%
- At the end of sprint 3, feature PC = 100%

Figure 3-1: Example of Agile product completion status rolling up into EVM reporting at the Feature level. The Feature is planned to be developed over 3 Sprints, with EV percent complete calculated using the Feature's weighted Story Point values completed. The Agile Team may be working on other Features not shown in this example; Story Points indicate only part of their total workload.

Figure 3-2 shows an example of rollup measures of EV Percent Complete (PC) at the Capability level which are derived from Percent Complete at the Feature level using PC from the Feature level as depicted in Figure 3-1 and the completion of Agile Stories determines the completion status for a Work Package (assumption, EVT is % Complete with QBD).

Progress and completion of individual Features is still determined based on completed Stories (as shown in Figure 3-1); but at the Epic/Capability level, the EV PC calculated at the Feature level is rolled-up into the higher-level Epic/Capability.

The Epics/Capabilities and Features shown in Figures 3-1 and 3-2 align well with EV reporting levels. More specifically, the WP scope is comprised of one or more features, and therefore, the shared product hierarchy in the agile tool is integrated with the EVM hierarchy and authorized work.



Earned Value Measured at the Epic/Capability Level

Percent Complete (PC) for the epic/capability at the end of each release:

- At the end of release 1, epic PC = 18%
- At the end of release 2, epic PC = 64%
- At the end of release 3, epic PC = 100%

Figure 3-2: Example of a higher level of rollup of Agile product completion status to EVM reporting. Features with Work Package level earned value roll up at each Release to weighted milestones based on Features completed within each Release, assuming an EVT of % Complete with QBD is chosen by the CAM. It is not a best practice to measure completion of the Agile Time Box.

3.3 Computing & Reporting Earned Value Performance

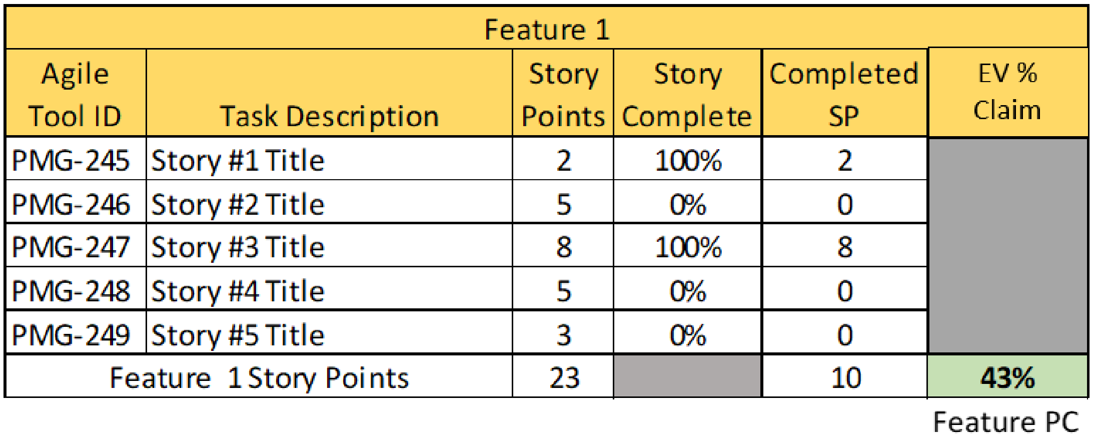
Progress can be calculated for a Work Package, which can be composed of a single Feature or multiple Features, by tracking the completion of Stories that are assigned to the Feature(s), consistent with the EVT of that Work Package. Figure 3-3 shows an example of how to calculate EV Percent Complete (PC) of the work package using this approach to report status for a single Feature. As shown in this example, total EV Percent Complete for the Feature Work Package is calculated by summing the total number of Story Points *completed* for the Stories in the Feature, and dividing it by the total number of Story Points *estimated* for that Feature[7]:

$$Feature\ Percent\ Complete = \frac{Total\ Completed\ Weighted\ Stories\ (in\ SP)}{Total\ Estimated\ Weighted\ Stories\ (in\ SP)}$$

[7] See Appendix C, Reference 6 for additional information on normalizing story points estimated across a program. Care must be taken when attempting to use story point information if not properly normalized and assessed during Release Planning.

Agile progress reports underpin the details captured in the sample QBD associated with the Feature Work Package and assumed EVT of % Complete with QBD (as defined by the EVMSD).

The QBD Tasks are the completion criteria used to calculate EV of the Feature Work Package. The criteria itself does not directly equate to hours or budget within the Work Package.



Feature 1					
Agile Tool ID	Task Description	Story Points	Story Complete	Completed SP	EV % Claim
PMG-245	Story #1 Title	2	100%	2	
PMG-246	Story #2 Title	5	0%	0	
PMG-247	Story #3 Title	8	100%	8	
PMG-248	Story #4 Title	5	0%	0	
PMG-249	Story #5 Title	3	0%	0	
Feature 1 Story Points		23		10	**43%**

Feature PC

Figure 3-3: Example of how planned stories defined to implement Feature 1 Work Package may be applied to create QBD to calculate earned value as a PC. In this example, each story is "weighted" using Story Points (relative size estimates). PC is claimed for each story completed.

This approach is consistent with the AAP Agile and EVM PM Desk Guide Update Approved for Nov 2020_FINAL Measuring Progress Section: "Item b. *Claiming performance* identifies four (4) guidelines when claiming performance."[8] As documented in Section 1.3 of this Guide, the stories and corresponding story points may change at each sprint boundary. There is business value in understanding and tracking the changes through configuration management of the QBD. Capture of these changes to ensure the future performance reported is consistent with the technical evolution of how the feature is being implemented.

3.4 When Do You Take Credit for a Story

During initial adoption of integrating Agile and EVM practices, industry adopted several options on when one could claim progress on a Feature Level WP when using Stories as QBD.[9] Due to the misalignment of accounting periods and Sprint cycles, methods included taking partial credit for a Story based on some lower-level objective measure of the story itself to "normalize" variances. Since the inception of this Guide, industry has moved to the standardized use of claiming progress only when the story is 100% complete as the most objective measurement of credit for the Feature WP. This aligns with the binary nature of the corresponding Agile principle "Working product is the primary measure of progress".

[8] https://www.acq.osd.mil/asda/ae/ada/ipm/docs/AAP%20Agile%20and%20EVM%20PM%20Desk%20Guide%20Update%20Approved%20for%20Nov%202020_FINAL.pdf

[9] As noted in Section 4.3, stories, while a common logical integration point for Agile and EVM are not required for claiming progress. This story-centric method has been provided as a best practice approach.

Another approach used to claim performance is to avoid underlying Stories and elevate performance assessment to the Feature level. Doing so enables the capture of progress against incremental steps within an Agile process, workflow, or Kanban to provide fidelity for capturing progress against work in progress. This aligns performance reporting against objective architectural elements and completion.

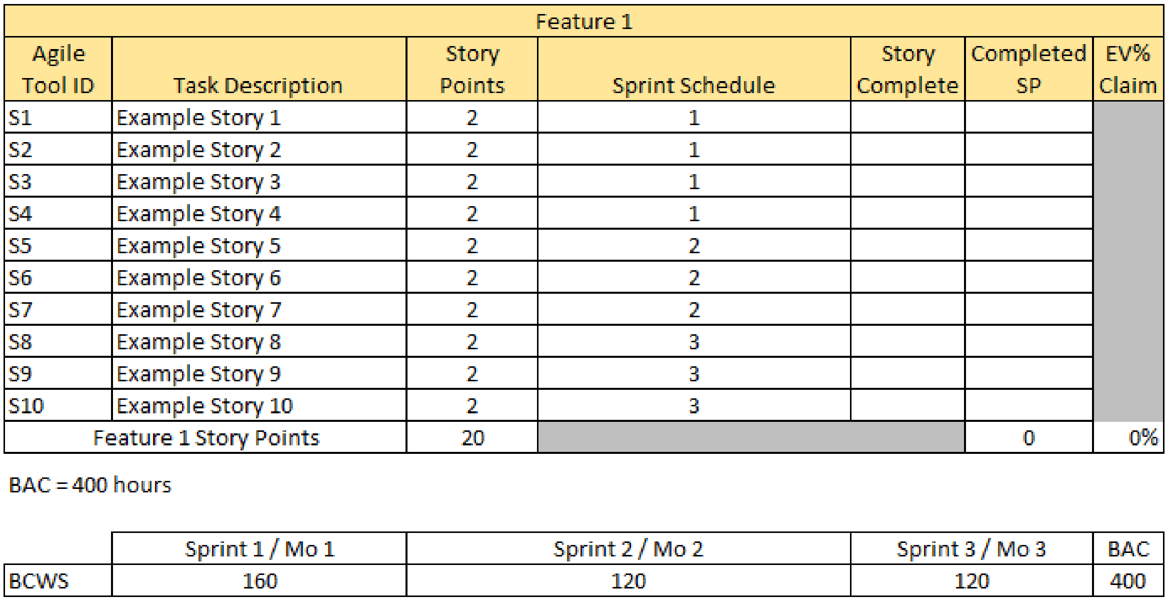
3.5 Feature Cost/Risk to be Considered When Establishing Baseline

In Agile development, as well as in any product development, there are always uncertainties. It is recommended that these complexity factors be included in the relative sizing of Epics/Capabilities and Features used when establishing a Work Package technical/schedule/cost baseline for that Feature. Planning should also include the isolation of any reserve capacity or assumptions for time needed to work off defects. Staff utilization, specifically the assumptions made during original complexity estimation for development focus factor, must also be considered when establishing baseline values. As usual in any Earned Value managed program, unknown risks may be held at a higher level against Management Reserve for use when in-scope unanticipated work is discovered, and new functionality must be added to complete a product.

3.6 Variance from the Baseline: Examples for Agile and EVM

Variance, the difference between an expected or desired value and an actual measured value, is a natural consequence of developing complex products. There are cases where the scope achieved took more or less time or effort than planned. The examples below show how cost and schedule variances could be observed on an Agile program. For this example, all EVM metrics (BCWS, BCWP, ACWP) are in hours (not dollars) for simplicity.

Assume that there is a plan to complete a Feature, with planned labor of 400 hours to complete the associated work. The Feature consists of 10 Stories of 2 points each of weighted Story Value, totaling 20.



Feature 1						
Agile Tool ID	Task Description	Story Points	Sprint Schedule	Story Complete	Completed SP	EV% Claim
S1	Example Story 1	2	1			
S2	Example Story 2	2	1			
S3	Example Story 3	2	1			
S4	Example Story 4	2	1			
S5	Example Story 5	2	2			
S6	Example Story 6	2	2			
S7	Example Story 7	2	2			
S8	Example Story 8	2	3			
S9	Example Story 9	2	3			
S10	Example Story 10	2	3			
Feature 1 Story Points		20			0	0%

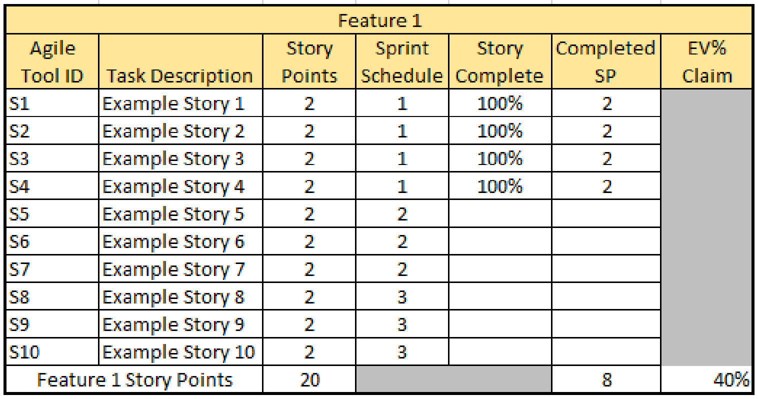
BAC = 400 hours

	Sprint 1 / Mo 1	Sprint 2 / Mo 2	Sprint 3 / Mo 3	BAC
BCWS	160	120	120	400

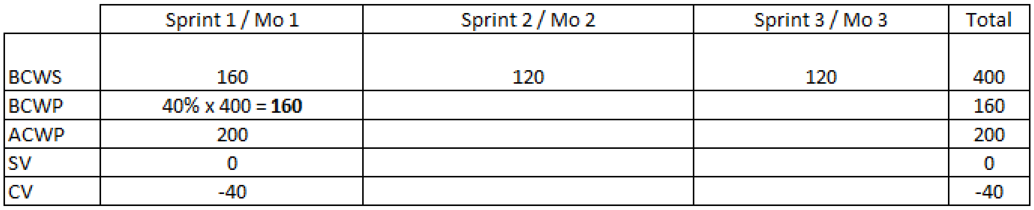
During the first Sprint, the team plans to complete 4 Stories for a total of 8 story points (SP). This equates to an estimate of 160 hours of labor (8/20 * 400 hours) for BCWS.

The following are examples of possible variances after a reporting month. For simplicity, these examples assume the Sprint duration is aligned with the reporting month. In practice, the Sprint duration may or may not align with the reporting month and the sprint duration will be consistent with the Agile Implementation Plan.

1. *On Schedule, Negative Cost Variance.* Consider the case where a team completes the planned amount of work in a Sprint for the Feature but incurred 200 hours of actuals rather than the expected 160 hours to complete it. This could result in a cost variance at the Work Package level if the remaining 6 Stories complete as planned.

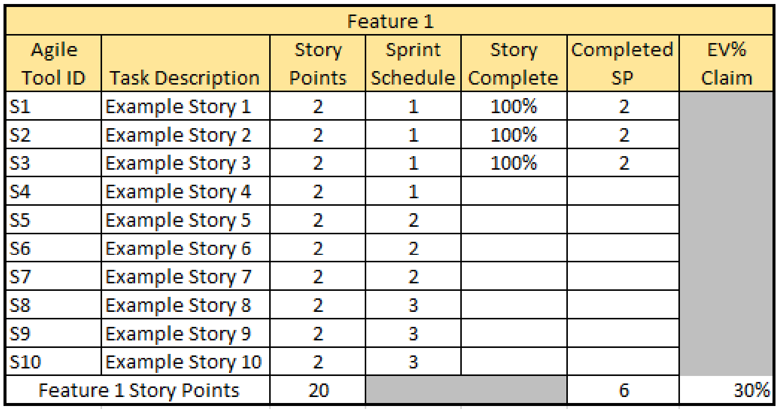


Feature 1						
Agile Tool ID	Task Description	Story Points	Sprint Schedule	Story Complete	Completed SP	EV% Claim
S1	Example Story 1	2	1	100%	2	
S2	Example Story 2	2	1	100%	2	
S3	Example Story 3	2	1	100%	2	
S4	Example Story 4	2	1	100%	2	
S5	Example Story 5	2	2			
S6	Example Story 6	2	2			
S7	Example Story 7	2	2			
S8	Example Story 8	2	3			
S9	Example Story 9	2	3			
S10	Example Story 10	2	3			
Feature 1 Story Points		20			8	40%

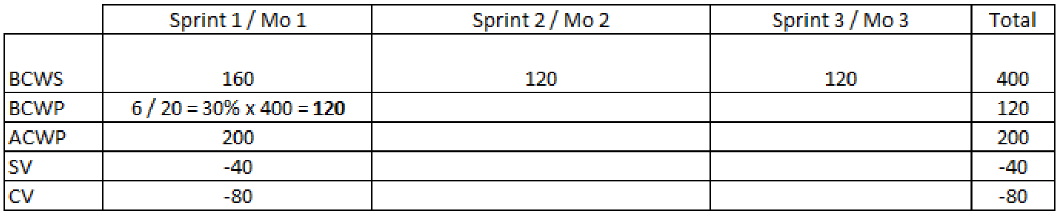


	Sprint 1 / Mo 1	Sprint 2 / Mo 2	Sprint 3 / Mo 3	Total
BCWS	160	120	120	400
BCWP	40% x 400 = **160**			160
ACWP	200			200
SV	0			0
CV	-40			-40

2. *Negative Schedule Variance, Negative Cost Variance.* A schedule variance could appear at the Work Package level if the team completed 3 of 4 planned Stories (6 of their planned 8 Story Points) using the hours associated with those Stories, with the remaining Story allocated to a later Sprint and no change to the Feature scope or exit criteria.



Feature 1						
Agile Tool ID	Task Description	Story Points	Sprint Schedule	Story Complete	Completed SP	EV% Claim
S1	Example Story 1	2	1	100%	2	
S2	Example Story 2	2	1	100%	2	
S3	Example Story 3	2	1	100%	2	
S4	Example Story 4	2	1			
S5	Example Story 5	2	2			
S6	Example Story 6	2	2			
S7	Example Story 7	2	2			
S8	Example Story 8	2	3			
S9	Example Story 9	2	3			
S10	Example Story 10	2	3			
Feature 1 Story Points		20			6	30%



	Sprint 1 / Mo 1	Sprint 2 / Mo 2	Sprint 3 / Mo 3	Total
BCWS	160	120	120	400
BCWP	6 / 20 = 30% x 400 = **120**			120
ACWP	200			200
SV	-40			-40
CV	-80			-80

3. *Adding a New Story:* During Sprint 1 Planning, a team discovers an additional implementation detail to satisfy the exit criteria of the feature. They document this detail as a new story in the Product Backlog (QBD). At the end of the sprint 1 / accounting month, status is taken, as reflected in hours in the table below.

Feature 1						
Agile		Story	Sprint	Story	Completed	EV%
Tool ID	Task Description	Points	Schedule	Complete	SP	Claim
S1	Example Story 1	2	1	100%	2	
S2	Example Story 2	2	1	100%	2	
S3	Example Story 3	2	1	100%	2	
S4	Example Story 4	2	1	100%	2	
S5	Example Story 5	2	2			
S6	Example Story 6	2	2			
S7	Example Story 7	2	2			
SNew	Example Story New	2	2			
S8	Example Story 8	2	3			
S9	Example Story 9	2	3			
S10	Example Story 10	2	3			
Feature 1 Story Points		22			8	36%

	Sprint 1 / Mo 1	Sprint 2 / Mo 2	Sprint 3 / Mo 3	Total
BCWS	160	120	120	400
BCWP	8 / 22 = 36% x 400 = 144			144
ACWP	160			160
SV	-16			-16
CV	-16			-16

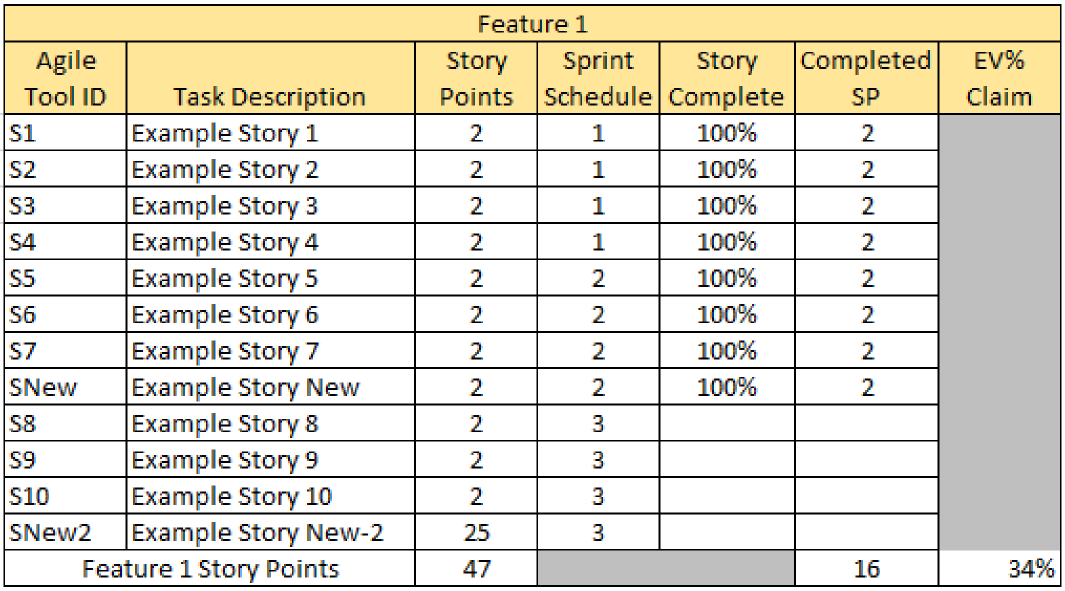
4. *Adding a Second new Story after Original Status:* During Sprint 2 planning, a team discovers an additional implementation detail to satisfy the exit criteria of the feature. The new Story goes into the Product Backlog (QBD) for a future Sprint assignment. At the end of the sprint 2 / accounting month, status is taken. The 66% complete considers the additional implementation detail added.

Feature 1						
Agile		Story	Sprint	Story	Completed	EV%
Tool ID	Task Description	Points	Schedule	Complete	SP	Claim
S1	Example Story 1	2	1	100%	2	
S2	Example Story 2	2	1	100%	2	
S3	Example Story 3	2	1	100%	2	
S4	Example Story 4	2	1	100%	2	
S5	Example Story 5	2	2	100%	2	
S6	Example Story 6	2	2	100%	2	
S7	Example Story 7	2	2	100%	2	
SNew	Example Story New	2	2	100%	2	
S8	Example Story 8	2	3			
S9	Example Story 9	2	3			
S10	Example Story 10	2	3			
SNew2	Example Story New-2	2	3			
Feature 1 Story Points		24			16	66%

Note – the new implementation detail (inefficiency) informs the negative cost variance.

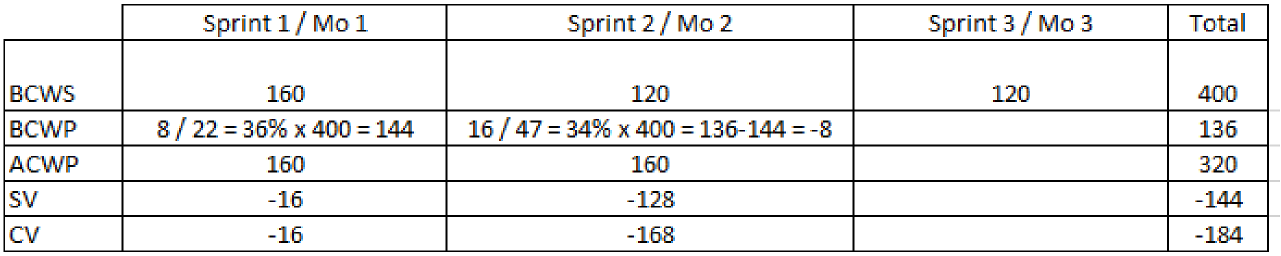
	Sprint 1 / Mo 1	Sprint 2 / Mo 2	Sprint 3 / Mo 3	Total
BCWS	160	120	120	400
BCWP	8 / 22 = 36% x 400 = 144	16 / 24 = 66% x 400 = 264-144 = 120		264
ACWP	160	160		320
SV	-16	0		-16
CV	-16	-40		-56

5. *Addition of Significant Implementation Detail Discovered during Development:* During Sprint 2 Planning, a team discovers an additional implementation detail worth 25 story points to satisfy the exit criteria of the feature. The new Story goes into the Product Backlog (QBD) for a future Sprint assignment. At the end of the sprint 2 / accounting month, status is taken. It is not typical for mature Agile teams to continually increase the number of stories or story points after the Release Plan is complete. Consistently changing story points (QBD) is an indicator that the Agile Implementation may have issues.



Feature 1						
Agile Tool ID	Task Description	Story Points	Sprint Schedule	Story Complete	Completed SP	EV% Claim
S1	Example Story 1	2	1	100%	2	
S2	Example Story 2	2	1	100%	2	
S3	Example Story 3	2	1	100%	2	
S4	Example Story 4	2	1	100%	2	
S5	Example Story 5	2	2	100%	2	
S6	Example Story 6	2	2	100%	2	
S7	Example Story 7	2	2	100%	2	
SNew	Example Story New	2	2	100%	2	
S8	Example Story 8	2	3			
S9	Example Story 9	2	3			
S10	Example Story 10	2	3			
SNew2	Example Story New-2	25	3			
Feature 1 Story Points		47			16	34%

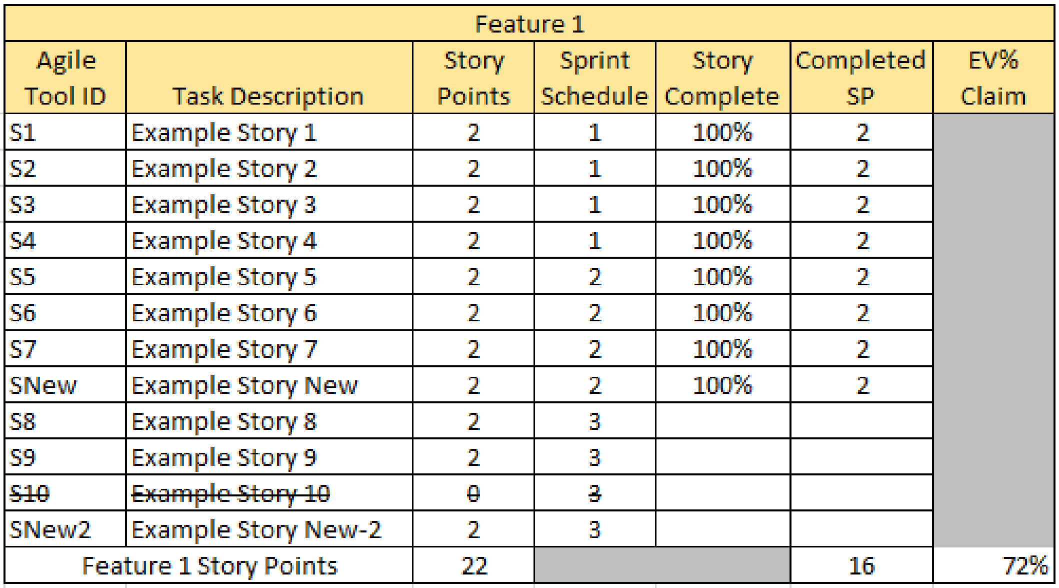
Note that the completed story points remain at 16, as example #4 demonstrates. Further note that since the new story is estimated at a significantly larger number of points than the other stories, the denominator has more than doubled since the original estimate. In this example, the cumulative % complete of the work package is now less than the Month 1 cumulative %, which causes the EV reported to regress, which is captured in current Month 2. In execution, teams typically will not add a story this large. This example is included to show the large amount the denominator must increase, relative to the current month's completion for EV to go backwards. Based on the cumulative EV % complete (in this example being 34%), the impact (reflected in hours) is realized in the current period reports, as all previously reported EV does not retroactively change.



	Sprint 1 / Mo 1	Sprint 2 / Mo 2	Sprint 3 / Mo 3	Total
BCWS	160	120	120	400
BCWP	8 / 22 = 36% x 400 = 144	16 / 47 = 34% x 400 = 136-144 = -8		136
ACWP	160	160		320
SV	-16	-128		-144
CV	-16	-168		-184

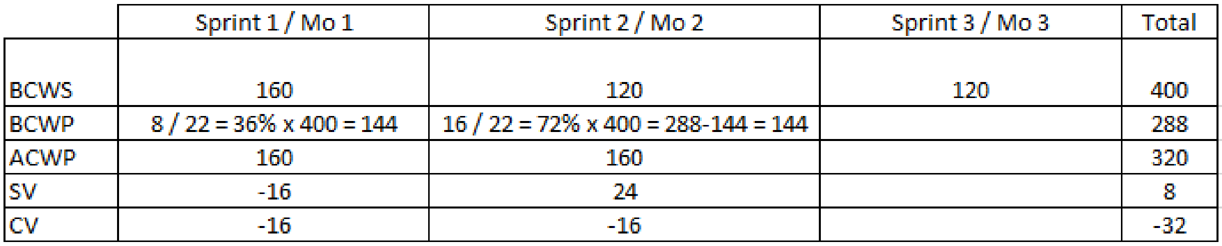
6. *Remove Implementation Detail:* During Sprint 2 planning a team discovers an implementation detail (S10) is no longer required to satisfy the exit criteria of the feature

This example shows the impact of zeroing out the story points associated with an implementation detail that was not needed to satisfy the exit criteria of the feature, which could be due to efficiencies gained. The story points are zeroed out in this example simply to show the impact of a reduced denominator. In practice, removing a story is done according to the documented Agile Plan, the corporate EVMSD and any specific program procedures.



Feature 1						
Agile Tool ID	Task Description	Story Points	Sprint Schedule	Story Complete	Completed SP	EV% Claim
S1	Example Story 1	2	1	100%	2	
S2	Example Story 2	2	1	100%	2	
S3	Example Story 3	2	1	100%	2	
S4	Example Story 4	2	1	100%	2	
S5	Example Story 5	2	2	100%	2	
S6	Example Story 6	2	2	100%	2	
S7	Example Story 7	2	2	100%	2	
SNew	Example Story New	2	2	100%	2	
S8	Example Story 8	2	3			
S9	Example Story 9	2	3			
~~S10~~	~~Example Story 10~~	~~0~~	~~3~~			
SNew2	Example Story New-2	2	3			
Feature 1 Story Points		22			16	72%

Note that the completed story points remain at 16, as example #4 demonstrates. With the change in the denominator from 24 to 22, the cumulative EV % complete goes to 72%. The impact of the new cumulative EV % complete is realized in the current period reports (reflected in hours below), as all previously reported EV does not retroactively change.



	Sprint 1 / Mo 1	Sprint 2 / Mo 2	Sprint 3 / Mo 3	Total
BCWS	160	120	120	400
BCWP	8 / 22 = 36% x 400 = 144	16 / 22 = 72% x 400 = 288-144 = 144		288
ACWP	160	160		320
SV	-16	24		8
CV	-16	-16		-32

The current period impact discussed in examples #5 and #6 is appropriate relative to efficiencies or inefficiencies being realized within the Agile teams as the work package completes. Updating the implementation details of the work package provides transparent insight and EV accuracy. Ignoring changes in the implementation details (i.e., not keeping the Product Backlog (QBD) consistent with the Agile tracking) will eventually violate EVMS criteria of accurate forecasting, accurate ETCs, and technical metrics diverging from EVM metrics. Ignoring changes in the implementation details may result in a work package being 100% complete (BCWS=BAC) with an ETC to complete the implementation details documented in the Product Backlog that cannot be incorporated into QBD / EV performance.

In each of these cases an EVM variance could appear at the Work Package level based on QBD calculations for that Feature; in any case Feature performance can be both projected and managed using the Agile workflow. Section 4.3 and 4.4 provide scenarios describing the effects of change and resulting cost and schedule impacts.

In some cases, rework may be identified for a closed Feature. The corporate Agile and EVM practices will document how to handle rework or additional effort discovered on a feature after that feature is closed and signed off by the product owner. In practice, companies may have different compliant solutions to address this scenario. Typically, the team will want to understand the reason for the rework. Confirm within the CWBS Dictionary the appropriate WBS for the rework to be completed. If the previously completed and closed Feature WP truly requires rework, one solution could be to consider opening a new work package in a new release, based on the placement of the rework in the Product Backlog and determine the source of the budget to complete the scope. Additional Baseline Change scenarios are documented in section 4.3 of this Guide.

This is consistent with the 17 November 2020 release of the OUSD(A&S) AAP IPMD Agile and Earned Value Management: A Program Managers Desk Guide (link contained in footnote 7). Section 1, Measuring Progress beginning on page 8 states:[10]

“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 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:

1. all stories reflect technical accomplishment towards a feature
2. once established, story point values do not change
3. stories can be added or removed from the QBD through the development process to support technical completion of a feature
4. The process by which stories are used 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.”

3.7 How to Use Agile Metrics to Support Forecasting ETC/EAC

Agile methodologies promote incremental, iterative planning. When establishing the PMB, Planning Packages are typically employed, which support this incremental planning approach. Agile does not advocate detailed planning all the way through to program end, which traditionally

[10] Note: This is an excerpt from an all-inclusive document. The citation will not change in this document. It is mean to reference work packages that may require QBD based on the Corporate EVMSD.

underpins ETC/EAC forecasting refinement. This avoidance of detailed longer-term planning is based on limited detailed data in out years of the program and the likelihood of customer-desired outcomes changing. Yet EAC forecasting is essential in EVM-managed, or any managed program.

For a program managed with both Agile and EVM, a program's entire budget can be plotted out at a summary level via Product Roadmap planning and a top-level IMP and/or IMS. At the Product Roadmap level, Epics/Capabilities, as well as a limited set of decomposed Features, are estimated and allocated to Releases, and when scope, schedule and budget are approved, an EVM baseline is established. As discussed in section 1.2, at each Release Planning event, the Planning Package for the next Release will be detail planned. The CAM assess the complexity of remaining work in the Product Backlog that is identified for the next rolling wave and compare it to the budget allocated to support EAC analysis. Since Stories and / or Story Point estimates vary by team, the CAM will analyze and understand the potential predictability of Stories and / or Story Points to the hours forecast for the remaining effort. The CAM estimates the number of hours each resource requires to complete the scope of work, so that the resource hours needed to complete a Feature can be rolled up and monetized to develop the Feature's ETC.

On a program employing traditional waterfall development, a Planning Package could be 6 to 12 months in duration or longer and span multiple program events. On an Agile program, the Planning Package is typically much shorter in duration, as it aligns with the Release duration. In this way, the strong planning rhythm offered by Agile enables Rolling Wave planning in traditional EVM to be taken to a new level of currency and accuracy, supported by Agile planning practices. [11]

Each Sprint within a Release includes work activities for product development. Work performance for deliverables completed in past Sprints and Releases can be used to generate a team efficiency factor that can support the Feature ETC and EAC. Predictions can be performed for future work base on the relative size of completed work, actual schedule performance and actual cost. Note that, consistent with EVM Policy, changes in estimated work made as the program progresses are not changes in work scope; scope remains the same as described in the program baseline.

The formulas in Appendix B include methods to calculate an estimate to complete (ETC) and are illustrated below.

The first formula, which was used in the QBD calculation example in Figure 3-3, shows the basis for calculating progress on a single Feature, which is done using Story Points assigned to the Stories that are derived from that Feature:

$$Feature\ Percent\ Complete = \frac{Total\ Completed\ Weighted\ Stories\ (in\ SP)}{Total\ Estimated\ Weighted\ Stories\ (in\ SP)}$$

A second formula shows an example of how to calculate remaining hours of effort for a Feature again using Story points (SP) assigned to the Stories that are derived from that Feature. This equation also leverages the total inception-to-date hours spent on the entire project in a ratio with

[11] See Agile and Earned Value Management: A Program Manager's Desk Guide, OUSD AT&L (PARCA), 16 April 2018 Section 2.d for guidance on developing Rolling Waves. As noted in Section 4.3, stories, while a common logical integration point for Agile and EVM are not required for claiming progress. This story-centric method

actual total number of Story Points completed within those spent hours to create a projection for the remaining work.

$$\begin{aligned}&\mathit{Feature\ Remaining\ Effort\ Hours}\\&\quad= (\mathit{Estimated\ SP\ for\ Feature} - \mathit{Completed\ SP\ for\ Feature})\\&\quad\times (\mathit{Total\ Hours\ spent\ to\ Date})/(\mathit{Total\ SP\ Completed\ to\ Date})\end{aligned}$$

4 Managing Baseline Change on Agile Programs

This section speaks to industry best practices for managing baseline changes on Agile development programs also using Earned Value Management. These best practices represent a knowledge network of Earned Value and Agile practitioners promoting a consolidated view. There are various policies, procedures, processes, and tools within industry and this Guide recognizes variability can exist. Below is a set of scenarios and associated guidance that are currently occurring within industry.

4.1 Baseline Change Parameters

The content in this section is documented based on fundamental parameters.

- Recognizing that Agile development methodology is in use across a wide variety of programs and companies, this discussion is limited to contracts that requires and would benefit from an EVMS, i.e., contracts that have some level of pre-defined goals or outcomes (requirements) tied to program events or milestones.
- Each organization will determine the Product Backlog Change Management process. Best practices suggest an alignment between the engineering / Product Backlog change process with the EVMS Change Management Process.
- Commercial programs developing product to take to market are not addressed.
- LOE or staff augmentation contracts awarded in support of a government led initiative are not addressed.

4.2 Baseline Assumptions

There are Agile terminology and assumptions made in the establishment of a program Performance Measurement Baseline (PMB) for an EVMS. These are used as the basis for the change scenarios in Section 4.3.

The program described here assumes an Agile implementation methodology that includes planning work within recurring timeboxed boundaries such as Sprints and Releases as described in Section 2.1.

- Product Hierarchy (best practice example):
    - The product hierarchy is made up of Epics/Capabilities that are decomposed into Features, which are sized to be scheduled to complete within a single Release. Each Feature is further decomposed into Stories, which are sized to complete within one Sprint, as depicted in Section 2.1. Stories are developed and maintained below the level of the EVMS PMB.
- EVM Hierarchy and definitions (best practice example):
    - CAs for this program are established at the Epic/Capability (product) level and may span many releases.
    - Work Packages for EVMS are created at the Feature Level. Feature Work Packages represent working product and have documented exit criteria (Acceptance Criteria).
    - Planning Packages represent working product associated with a future release.
    - The Product Backlog documents the technical scope of each CA.

  - All items listed on the Product Backlog include rough size complexity estimates (weighted Story Value in Story Points, ideal hours by resource, T-Shirt size (relative sizing method for typically smaller Agile efforts, in S, M, L, XL etc.), Fibonacci sequence, other) that are refined over time as knowledge is gained.
  - All Items listed on the Product Backlog are traceable to a Work Package or Planning Package in the PMB.
- A Product Roadmap is maintained that represents the prioritized Product Backlog. Epics/Capabilities and Features on the Product Backlog are mapped to specific releases as part of the Product Planning process.
  - Product Backlog grooming (refining) is a continuous and normal part of Agile management, and it is possible that Future Epics/Capabilities and Features may be reprioritized and mapped to different releases based on discovery or user feedback.
- The program does Rolling Wave planning at Release points. Rolling wave planning occurs after the Release Planning Event. The current release is detail planned and decomposed into “Feature Work Packages” (for Section 5 scenarios assume that there is only one Feature for each Work Package). Budget for future releases remains in Planning Packages.

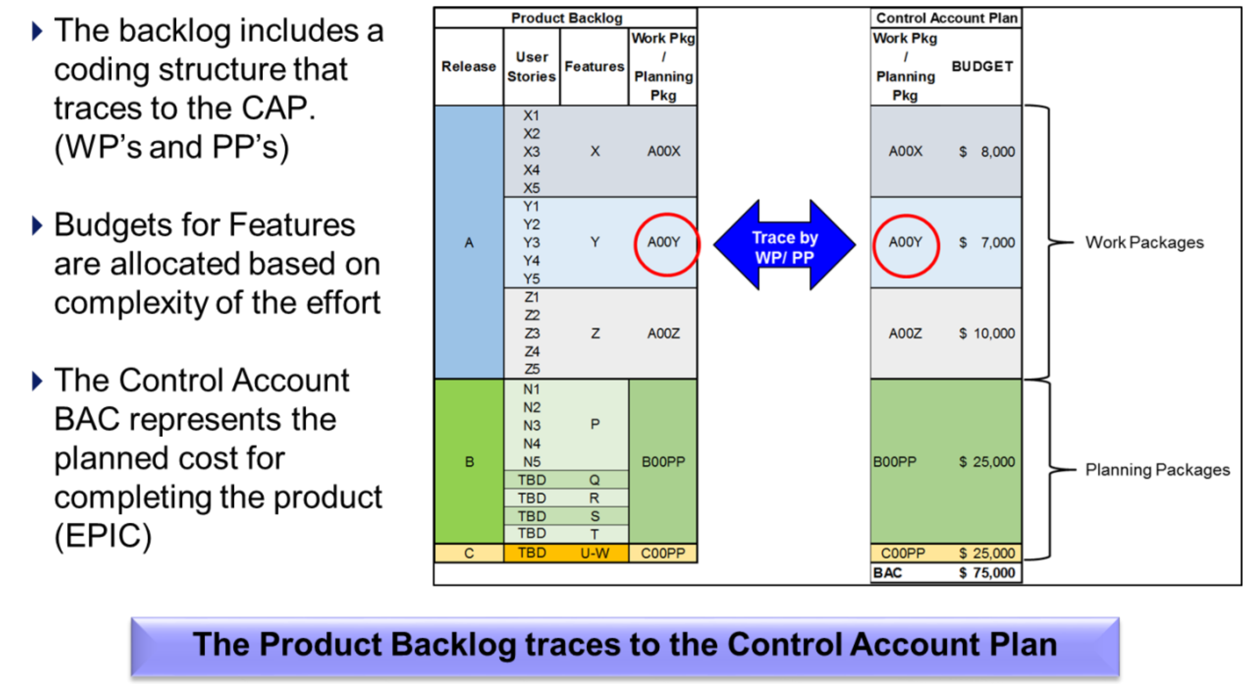
4.3 Baseline Change Scenarios (Scope and Budget)

Baseline Change Scenarios		
Scenario	PMB Action	Product Backlog Action
1. The Work Package/Feature is not open, and work has not started. It is determined the Feature is not needed for the current release. (Scenario 4-1 graphic included at end of Section 4.3.)	Baseline Change: Re-plan Work Package to future release. If the baseline start of the Feature is inside the program’s “freeze period”, appropriate control and notification mechanisms apply.	Feature and related stories are mapped to future releases within the Product Backlog.
2. The Work Package/Feature is 30% complete but did not complete by a formal delivery date. The delivery date is held as planned. The customer accepts the delivery without the Feature functionality.	In most cases, this is not a baseline change. Although the customer accepted the delivery, the original plan was not met. In this case the Feature remains open, showing a schedule variance until the work is completed. Depending on how long the Feature remains open, consider a change to the ETC, and analyze the cost impact.	The unfinished Feature's stories are assigned to a future sprint with the next release. The WP identifier remains unchanged.

Baseline Change Scenarios		
Scenario	PMB Action	Product Backlog Action
3. Features for the current Release are re-prioritized. A planned Feature is swapped with a different Feature from the Product Backlog of a similar size that was mapped to a future release. ***(This is unusual.)***	Baseline Change: The swap is documented, even if the overall budget and baseline schedule dates do not change. IMS task descriptions and Feature Work Package descriptions/exit criteria are updated as necessary. The IMS is checked to ensure interdependencies remain valid. If the baseline start of the Feature is inside the program's "freeze period", appropriate control and notification mechanisms apply. Each EVMSD will be referenced for changing an open work package if the re-prioritization occurs on an open work package. The PMB action will be consistent with the EVMSD.	Features and related Stories are re-mapped to applicable WP and release PP. WP and PP identifiers are updated. Feature release and Story sprint assignments are updated in the Product Backlog.
4. The Contracting Officer (CO) issues a contract modification which removes the scope of an Epic/Capability (requirement). The change affects a Feature which is currently baselined in an open Work Package.	Baseline Change: The in-progress WP is closed by setting BCWS equal to BCWP. The unclaimed budget associated with the Capability is returned to Undistributed Budget (UB) until dispositioned by contract modification (de-scope).	The unfinished Stories, Features and Epic/Capability are removed from the Product Backlog.
5. The exit criteria for Feature 1 Work Package are updated to add additional functionality (requirements) to that Feature. Stories are created to satisfy the additional requirements. ***The important consideration here is that the exit criteria of the Feature Work Package have changed.***	Baseline Change: The scope of Feature 1 has increased. Budget must be added for that new scope. If this is the result of a customer desired enhancement (new scope) the budget will come from UB. If this is an un-planned in-scope increase, the budget will come from Management Reserve (MR). If the baseline start of the Feature is inside the program's "freeze period", appropriate control and notification mechanisms apply, including provisions from an EVMSD on changes to in-progress work packages	The exit criteria for Feature 1 are updated. Stories are created and added to the Product Backlog and mapped to Feature 1.

Scenario 4-1 Graphics: Example of a Change Modeled in the PMB and Product Backlog

In figure 4-1, an un-started baselined Feature Work Package is not needed for the current release and is rebaselined to a future release. The two figures below depict this scenario. Figure 4-1 shows the current CA baseline, and how it is modeled in the Product Backlog and in the Control Account Plan (CAP). To maintain traceability from the Product Backlog to the CAP, a common field (the Work Package/PP ID number) is found in both.



- The backlog includes a coding structure that traces to the CAP. (WP's and PP's)
- Budgets for Features are allocated based on complexity of the effort
- The Control Account BAC represents the planned cost for completing the product (EPIC)

Product Backlog

Release	User Stories	Features	Work Pkg / Planning Pkg
A	X1 X2 X3 X4 X5	X	A00X
	Y1 Y2 Y3 Y4 Y5	Y	A00Y
	Z1 Z2 Z3 Z4 Z5	Z	A00Z
B	N1 N2 N3 N4 N5	P	B00PP
	TBD	Q	
	TBD	R	
	TBD	S	
	TBD	T	
C	TBD	U-W	C00PP

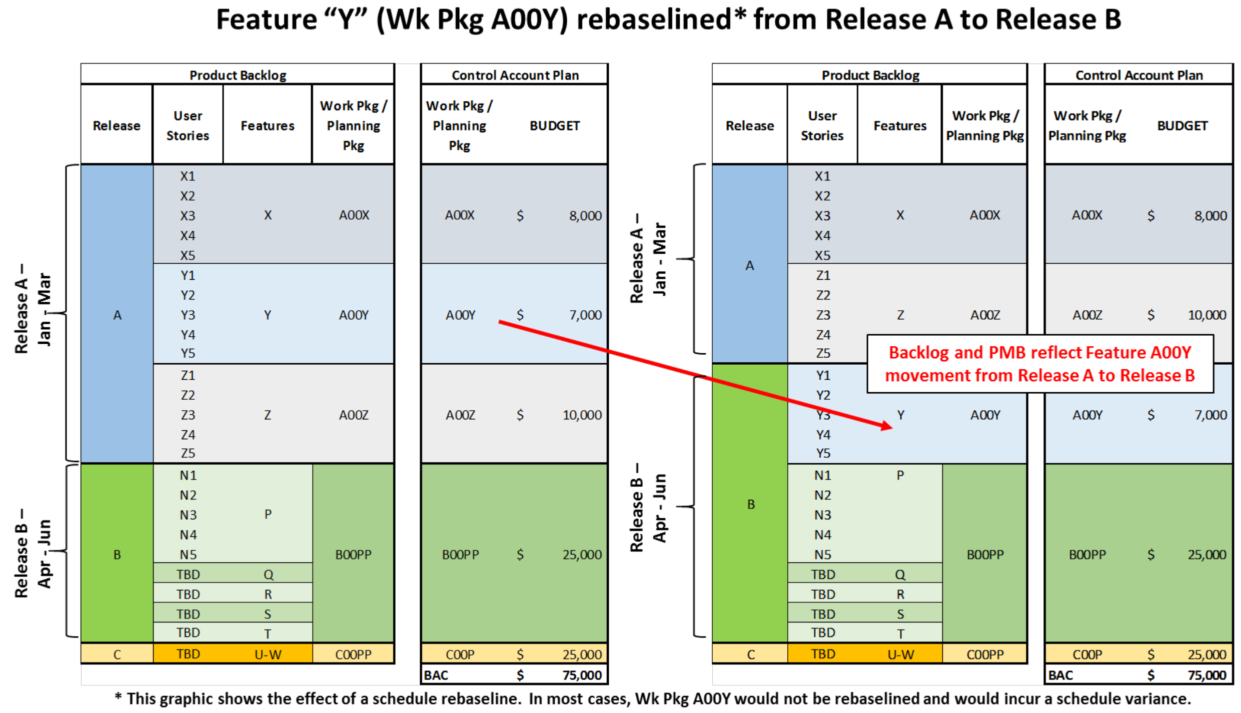
Control Account Plan

Work Pkg / Planning Pkg	BUDGET
A00X	$ 8,000
A00Y	$ 7,000
A00Z	$ 10,000
B00PP	$ 25,000
C00PP	$ 25,000
BAC	$ 75,000

The Product Backlog traces to the Control Account Plan

Figure 4-1: Product Backlog to Control Account Plan Traceability Example

In the Figure 4-2 below, Feature Y is rebaselined to a future release and the graphic shows how the change is modeled in the Product Backlog and the CAP. The Feature and associated Stories are moved to the next release in the Product Backlog, and the Feature Work Package in the CAP is rebaselined, moving the budget for Feature Y into the Release B time frame. This demonstrates the movement of scope and budget together



Feature "Y" (Wk Pkg A00Y) rebaselined* from Release A to Release B

Product Backlog

Release	User Stories	Features	Work Pkg / Planning Pkg
A	X1 X2 X3 X4 X5	X	A00X
	Y1 Y2 Y3 Y4 Y5	Y	A00Y
	Z1 Z2 Z3 Z4 Z5	Z	A00Z
B	N1 N2 N3 N4 N5	P	B00PP
	TBD	Q	
	TBD	R	
	TBD	S	
	TBD	T	
C	TBD	U-W	C00PP

Control Account Plan

Work Pkg / Planning Pkg	BUDGET
A00X	$ 8,000
A00Y	$ 7,000
A00Z	$ 10,000
B00PP	$ 25,000
C00P	$ 25,000
BAC	$ 75,000

Product Backlog

Release	User Stories	Features	Work Pkg / Planning Pkg
A	X1 X2 X3 X4 X5	X	A00X
	Z1 Z2 Z3 Z4 Z5	Z	A00Z
B	Y1 Y2 Y3 Y4 Y5	Y	A00Y
	N1 N2 N3 N4 N5	P	B00PP
	TBD	Q	
	TBD	R	
	TBD	S	
	TBD	T	
C	TBD	U-W	C00PP

Control Account Plan

Work Pkg / Planning Pkg	BUDGET
A00X	$ 8,000
A00Z	$ 10,000
A00Y	$ 7,000
B00PP	$ 25,000
C00P	$ 25,000
BAC	$ 75,000

Backlog and PMB reflect Feature A00Y movement from Release A to Release B

* This graphic shows the effect of a schedule rebaseline. In most cases, Wk Pkg A00Y would not be rebaselined and would incur a schedule variance.

Figure 4-2: Product Baseline to Control Account Plan, Changes Traced Example

4.4 Forecast Change Scenarios (No Scope Change)

Forecast Change Scenarios		
Scenario	PMB Action	Product Backlog Action
1. A Feature Work Package that spans 3 Sprints has started. The team determines that some of the Stories mapped to the Feature planned in the first Sprint will not be completed and moves those Stories to the second Sprint which still falls inside the baseline finish date of the Feature.	No change to Feature Work Package baseline budget or baseline schedule. Stories can be moved from Sprint to Sprint within the planned duration of the Feature Work Package without impacting the baseline.	The product Backlog is updated to move the Stories not completed in the first Sprint into the second Sprint.
2. A Feature Work Package that spans 3 Sprints has started. The team determines that some of the Stories mapped to the Feature planned in the first Sprint will not be completed and move those Stories to Sprint 4, which is beyond the baseline finish date of the Feature.	No change to Feature Work Package baseline budget or baseline schedule. The in-progress Feature IMS task shows a slip to the forecasted finish date. BCWP is only claimed for the Stories actually completed. BCWP compared to BCWS identifies a schedule variance. Reflect changes in IMS Forecast dates and EV Cost Tool EAC.	The Product Backlog is updated to move the Stories not completed in the first Sprint into the fourth Sprint.
3. A Feature Work Package has started but will not be completed by a formal delivery date. Customer states that the functionality is needed for the formal delivery.	No change to Feature Work Package baseline budget or baseline schedule. The Feature is forecasted to slip beyond the delivery date. The IMS shows a late delivery. Critical Path (float) is impacted. Reflect changes in IMS Forecast dates and EV Cost Tool EAC.	The unfinished Stories are moved into the Sprint in the next release cycle where they are forecasted to be completed.
4. The PO and team determine a Story is deemed unnecessary for the accomplishment of the Feature due to an increased understanding of Feature exit criteria (requirements). ***The Exit Criteria for the Feature has not changed.*** The Feature WP is in progress. Feature QBD is the Stories mapped to the Feature.	No change to Feature Work Package baseline budget or baseline schedule. Feature QBD is updated to remove the Story. Removal of the Story from QBD may result in an increase in Feature WP percent complete since the percentage of unfinished effort has decreased. Reflect changes in IMS Forecast dates and EV Cost Tool EAC.	The Story is removed from the Product Backlog.
5. The PO and team determine a Story needs to be added for the accomplishment of the Feature due to an increased understanding of Feature exit	No change to Feature Work Package baseline budget or baseline schedule. Feature QBD is updated to add the Story. Adding the Story to the QBD may result in a	The Story is added to the Product Backlog and mapped to the Feature. A Feature

Forecast Change Scenarios		
Scenario	PMB Action	Product Backlog Action
criteria (requirements). ***The Exit Criteria for the Feature has not changed.*** The Feature WP is in progress. Feature QBD is the Stories mapped to the Feature.	decrease in Feature WP percent complete since the percentage of unfinished effort has increased (effectively de-earning reported EV). Reflect changes in IMS Forecast dates if required dates and EV Cost Tool EAC.	Work Package identifier is added.
6. After a Feature Work Package and the associated Stories are accepted and claimed 100% complete, a problem is found. The defect is defined as critical and accordingly must be corrected before the functionality can be released. A Defect Report (DR) is written.	a. If a stand-alone Work Package has already been established for critical DRs in the current release, the new DR is added to the QBD for that Work Package. b. If a separate work package for critical DRs has not been established, it may be appropriate in some cases to reduce BCWP on the Feature Work Package if the work is not truly completed. The Feature QBD percent complete and forecast finish date are adjusted accordingly. Reflect changes in IMS Forecast dates and EV Cost Tool EAC. c. If the DR is truly unplanned – in scope effort, Management Reserve may be applied to the WP.	a. The new DR Story is added to the product Backlog and mapped to the established DR Work Package. b. The DR Story is added to product Backlog and mapped to the Feature Work Package. c. The DR Story is added to product Backlog and mapped to the Feature Work Package.
7. Features mapped to future releases are reprioritized based on discovery and user feedback and mapped to other future releases. Budget for future releases is in a Planning Package.	No change to budget or baseline schedule. This is not a baseline change because this work has not been detail planned. This kind of re-prioritization is expected; however, the Product Roadmap should be analyzed for potential bow-wave (work consistently moving 'to the right' without corresponding work moving forward) and related critical path impacts. If a bow-wave is apparent, a baseline change may be required to adjust the PP monthly budget spread. Reflect changes in IMS Forecast dates and EV Cost Tool EAC.	The product Backlog is updated, and the Features are mapped to the resulting releases on the Product Roadmap.

Scenario 4-2 Forecast Change Scenarios (No Scope Change)

4.5 Agile/EV Recommendations

- The Contractor should establish a freeze period that supports the flexible nature and shorter planning cycles of Agile development (Appendix C, Reference 8, Guideline 29).

Discovery and change are a normal part of Agile development, and change assessments occur frequently, often at the end of each Sprint. Assuming a Sprint cadence of every 2 weeks and Rolling Wave planning at 3-month Release points, the Contractor may want to establish a short freeze period, perhaps a 2-week forward window, or the current Sprint Period of Performance (POP). A traditional freeze period will greatly limit the program's ability to respond to change quickly. A Contractor's freeze period should be defined in a way to support Agile and EVM. The freeze period should be adjusted, through formal changes to a company's EVMSD or other supplementary guidance, to be short enough that it accommodates the Agile planning cycle. However, while a Contractor's EVMSD is in the process of being updated to incorporate Agile adaptation for freeze period, the contractor program should document the Agile process used in the interim along with the plan for updating the EVMSD.

  - A key point is that planning, including detail planning of planning packages, completes prior to the start of work for any of the products in the upcoming Release. The customer should be highly integrated into the Release Planning process, with ample opportunity to provide input on the plan if there are concerns.
  - For Performance Assessments and Root Cause Analyses (PARCA) guidance on this topic, please see Section 2.e of the Agile and Earned Value Management: A Program Manager's Desk Guide, OUSD AT&L (PARCA), 16 April 2018.

- The Contractor should establish budgets, or MR reserves, that are inclusive of estimated Defect Report (DR) corrections related to the development effort. When establishing the PMB, some portion of the development effort's budget is retained for eventual DR work off. This allows for risk reduction and addresses the reality of defect identification during later program phases. Proactive identification of DR budgets or reserved capacity can also be accommodated by including this in assumptions for an Epic's/Capability's Features.

5 Contracting for Agile and EVM

This section provides supplemental information for contracting guidance to address best practices for instances where government solicitations require integrating both an EVMS and an Agile development process. It provides the foundation and background to evolve the approach to contracting for Agile and EVM. The working group recognizes there are various policies, procedures, processes, and tools within industry and developed this section understanding that variability exists. We encourage continuous feedback, comments, ideas, and suggestions to the working group to continue to promote best practices on this topic.

There are several considerations to be made when entering into a solicitation or contract requiring both an Agile methodology and EVM practices. In some cases, specific clauses are required as well as the recommended use of performance based contracting principles. Traditional artifacts, such as the Integrated Master Plan and System Engineering CDRLs should be approached differently. Managing change with both Agile and EVM requires a mutual understanding of the definition of "change" as applied to the contract scope.

The purpose of including both an Agile development methodology and EVM on a contract is to drive collaboration on the product with a heightened awareness of schedule and cost. EVM is not tied to any specific development methodology and does not prevent the use of other risk management techniques. EVM and agile development are complementary and can be used on the same project. Agile development can be used to incrementally deliver functionality to the customer while EVM provides a standard method for measuring progress. Reference Office of

Management and Budget (OMB) Circular A-11, Supplement to the Capital Programming Guide July 2017 OMB Circular A-11)).

5.1 Defining the Agile Product

An "Agile" product should not be defined by a prescriptive set of requirements as typically seen in government contracting, but rather should be defined by the agency mission critical capabilities which are to be enabled by the program. When Agile is used to create products, not every change equates directly to an Engineering Change Proposal (ECP) or an EVM baseline change. It is critical that all stakeholders of both the buying and the selling entities work together to evolve the final product. Change management at the contract level should be assessed as compared to the program capabilities which have been defined.

5.2 Successful Agile Contracting

The Software Engineering Institute and Carnegie Mellon University published the RFP Patterns and Techniques for Successful Agile Contracting in November 2016, which introduces recommendations for the appropriate incorporation of a scope document in an RFP. Section C of an RFP usually provides the government's (buyer) requirements and expectations of the contractor's (seller) performance in the form of a Statement of Objectives (SOO) or Statement of Work (SOW). The SOO reflects a Performance-Based Acquisition (PBA) and is best suited for an Agile acquisition.[12] If a SOO is provided, the government will normally expect the contractor to provide a SOW or a Performance Work Statement (PWS) as part of its proposal.

A government-provided SOW is best suited for a traditional acquisition in which the government has a high degree of confidence in the ability to specify (both qualitatively and quantitatively) the expected approach and product end state. Table 5-1 highlights the differences between a SOO and a SOW.

SOO	Factor	SOW
The government understands the objectives but expects the end state to evolve.	Government understanding	The government has a high level of confidence in the end state.
Change is expected to be a significant factor in achieving the end state.	Change	Change is not anticipated, or if encountered will not be disruptive.
This approach provides the offeror trade space and flexibility in developing their proposal based on their experience regarding the most efficient process to develop the defined capabilities.	Constraint	Constrains offerors to the specific tasks identified, so must be unambiguous and comprehensive. The government needs to apply specific constraints on the tradeoff space of lifecycle cost, performance, interoperability, logistics/training, etc.

Table 5-1: SOO and SOW Differences

The scope defining document (SOO, SOW, or PWS) should communicate the product required, the quality to standards to be achieved, the required date and any schedule or intermediate deliverable items required. An Agile product is not a pre-defined, prescriptive set of requirements. For the Agile methodology to be effective, the seller, buyer and product owner must work together, and such collaboration and flexibility must be documented in the contract and scope control

[12] A SOO has been provided as a best practice approach. http://acqnotes.com/acqnote/acquisitions/performance-based-acquisitions

document. It is recommended that the documented requirements are flexible enough to not establish impediments that inhibit the contracting officer to use the right clauses to bound the contract and manage change in execution. A comparison between a SOO, PWS and SOW is contained in Table 5-2.

	SOO	PWS	SOW
Buyer	1. Describes requirements defined as the capability's outcome. 2. Does not identify a technical solution to the requirement 3. Saves time in developing the solicitation	1. Buyer defines the capabilities outcome and a minimum viable product which results in a detailed PWS. 2. Buyer has more control over what the bidders may propose. 3. May describe performance measures and Quality Assurance objectives or request information from bidders. 4. Links the capabilities to the agency mission and objectives rather than prescribing how the work shall be accomplished.	1. Buyer provides a detailed description of the specific services or tasks the contractor is expected to accomplish the work. 2. Buyer has more control over what the bidder may propose. 3. Used when requirements are well known and provides significant details regarding exactly "how" the work is to be performed.
Seller	1. Prepares a detailed work plan that serves as the PWS. 2. Includes performance measures, and quality assurance objectives & incentives. 3. Is free to propose what they believe is the best manner in which to achieve the required outcomes. 4. Encourages seller innovation	1. Prepares a proposal that corresponds closely to work approach as described by the Buyer, but still with a goal of achieving desired outcomes. 2. Proposes to meet required quality assurance objectives and/or performance metrics 3. Enables assessment of work performance against measurable performance standards	1. Prepares a detailed proposal that complies as much as possible with the stated requirements. 2. Is usually not free to propose a different solution except as an alternative proposal 3. Does not encourage seller innovation.

Table 5-2: SOO/PWS/SOW Comparison

Product Roadmap

In addition to a SOO with stated objectives, an Agile product can also be described in the performance-based contract by using a goal-oriented Product Roadmap that identifies the product functionality or Epics/Capabilities. Recommend that the corresponding metrics, names, dates, and goals be associated with the acceptance criteria. There should be enough detail in the stated objectives or road map to describe the complete end product, but not so much detail that it prevents execution of a collaborative Agile method. Collaboration between the buyer, the seller and product owner(s) will ensure the business value described is achieved during contract execution for the end product. As progress is made on accomplishing the objectives, each incremental step of feature development should build on the previous one completed and focus on the end goal or a vision of the project.

If a Product Roadmap is utilized, consider its level of specificity. Target the general product needs and objectives to be described in terms of Epics /Capabilities, and not stories. The recommended level is where the buyer – seller team has flexibility to define the stories within the appropriate planning horizon, update the plan, and revise the final technical implementation without needing to make modifications to the contract or EVM baseline. How the work is to be executed is not defined in the Product Roadmap but is captured in the Agile process and implemented within the project's Agile management tool. The Product Roadmap is typically characterized by the theory of preservation of alternatives until the latest possible time.

Integrated Master Plan

If an Integrated Master Plan (IMP) is required, the Product Roadmap should be included in the appropriate section of the IMP. Reflecting the "accomplishment" of the target goals is a natural convergence of the product roadmap and the events, accomplishments, and criteria documented in the contract IMP. See Sections 2.2 for further discussions on an Agile IMP.

Contract Scope Control Document

The contract scope control document should be specific as to the capabilities required, cite the objectives leading to a releasable complete solution and include Acceptance Criteria. However, it should also allow the team the necessary flexibility to be 'Agile' and determine throughout the development exactly how those broad capabilities will be achieved. The contract narrative should focus on small, frequent Epic/Capability/Feature Releases, rapid response to changes in technology, and facilitating an open dialog between the developers and end users to ensure high operational value.

Documentation should be kept to a minimum and used for reporting purposes to demonstrate frequent iterations and measure progress of the project. Table 5-3 outlines these factors, with the Agile Manifesto and appropriate contracting discussion.

Factor	Agile Manifesto	Contracting Discussion
Documentation	Working software over comprehensive documentation	Review the CDRLs and update the contract accordingly. Use an "as-built" approach to create the most absolutely necessary documentation required.
Planning	Responding to change over following a plan	Use the product roadmap and rolling wave planning together for just in time planning, based on top priorities.
Change Management	Customer collaboration over contract negotiation	In Government Contracting there will be requirements that support a target end product. Use the SOO and constant customer collaboration to manage the requirements matrix.

Table 5-3: Factors, Agile Manifesto and Contracting Discussion.

The project artifacts above will reduce program technical risk, schedule, cost and will remove waterfall project management method constraints. Once the product baseline and the conceptual design emerge, the shift to small iterations and teams will enable and streamline agility and bring the necessary stakeholders together.

5.3 Agile and EVM Solicitation Considerations

There are specific items that may be cited in an Agile and EVM Solicitation. For each solicitation consider the specific Agile goals and do not use these items to constrain the Agile process. Not all the items are appropriate for all types of Agile execution. Suggestion for consideration include:

Definition of Done (DOD)

Include a provision to agree to a Definition of Done (typically a check list for a team to verify the quality standards to denote story completion). Include a provision to agree to the Acceptance Criteria (the validation of system performance against the scope criteria and requirements). Recommended to develop this in parallel with negotiations and include as an appendix. The acceptance criteria (Agile) include the definition of done and should be consistent with the exit criteria (EVM) of the work packages.

Include a mechanism in the contract to verify this, such as a demo. If not a demo, a documented provision to account for the selling off requirements to verify the functionality produced matches the product vision. The demo or sell-off should occur within a reasonable amount of time after the progress is completed, not at the end of the contract.

Elements to consider for the Definition of Done include and are not limited to an agreed to checklist for example, coded per coding standards, peer reviewed, unit tested, etc. Elements to consider for the Acceptance Criteria are the scope of tests to be conducted and passed, to demonstrate the scope and requirement of the contract are satisfied and the system is performing. The Definition of Done can be defined at various levels, for a story, a feature, a sprint and / or a release.

Product Owner Responsibilities (Customer Interaction)

Include a provision to address the key responsibilities of the Product Owner, defining customer interaction. Examples include and are not limited to the initial development and prioritization of the product backlog, potential co-location with team, ongoing revisions and re-prioritization of the product backlog and participation in relevant Agile ceremonies (planning, review, demo, sell-off). It is recommended that the Product Owner / Customer “Proxy” be included on the Buyer IBR team.

Development Team Responsibilities

Include a provision to address the key responsibilities of the development team. Examples include and are not limited to the team composition and skill set, time commitment (dedicated or not), a specific number of teams for the contract, potential team co-location and the potential for reassignment without buyer permission.

Iterations

How can the solicitations be approached in more of an iterative way using definitizing options associated with certain incremental objectives established? Modular contracting? Task Orders? Examples include and are not limited to agreements to run a series of iterations, plan and implement each iteration according to a preselected methodology, require written minutes as output from planning sessions, and synchronize Agile Release Planning with EVM Rolling Wave Planning.

Planning

Does the contract need to include a provision for formal planning? Examples include, and are not limited to key roles defined, SOW includes product vision and outcomes, high priority items identified in the contraction, process for prioritization / re-prioritization / equivalency swaps, expectations for meeting attendance, and synchronize Agile Release Planning with EVM Rolling Wave Planning.

Reporting

Include a provision for how reporting, including metrics and performance measures will be different. The Agile metrics and EVM data should report a consistent story. Examples include, and are not limited to working software, modified Software Development / Enterprise Performance Life Cycles, test plans per sprint, sprint burn down charts, product backlogs, Epic/Capability and release burndown and velocity.

Testing

Does the contract need to include specific testing provisions? Examples include, and are not limited to multiple testing subcontractors, outsourcing impact to quality, outsourcing impact to team, success metrics defined, integration of outsourced effort, and accounting for the cost of technical subcontract management.

Fixed Price versus Cost Plus

Agile and EVM can be implemented under both Cost type and Fixed price type contracts. While a cost type contract can allow more flexibility, Buyers often feel that they are not able to control program costs given an open-ended contract with only desired outcomes. Using a modular or incremental approach can be an effective scope and cost control mechanism. Under an Agile and EVM Fixed Price contract, the Buyer can articulate the objectives of the contract enough for a Seller to bid an approach to deliver scope, within a certain schedule and price. The Buyer and Seller must adopt a cooperative program management process that allows the development team the flexibility to make equivalency trade-offs to achieve a workable product within the constraints of the contract.

Payment Milestones

Performance based Milestone payments may be appropriate (See FAR 32.10) for agile development contracts. Consideration should be given to establishing payment milestones during contract negotiations, allowing for the payment of costs, award, or incentive fees. The IMP / IMS may be used to provide insight into schedule critical path(s), performance risks, and milestones at which risk is retired that should be considered in the selection of payment milestones.

It is recommended to not be overly prescriptive. The payment milestones should be based on significant events or accomplishments and not a finite list of features or number of sprints or releases to be completed. Let the Agile process deliver the product and the payment milestones be based on significant events or accomplishments. The engineering should not be constrained by business and a rigid payment milestone schedule.

5.4 Clauses and Agency Policy Citations

The guidance in this section is intended for the Executive Branch of the Federal Government for large infrastructure type contracts. These contracts present sufficient risk to warrant including provisions in the solicitation for supporting the appropriate program management processes and disciplines to bind the contract in execution. The notification of EVM on a solicitation or contract does not change with the addition of the Agile methodology. This section is a cross reference for a list of potential clauses to be considered for inclusion when contracting for Agile and EVM. It does not supersede any other guidance for contracting for EVM. Depending on the agency conducting the solicitation, a combination of these may apply.

Despite any policy references to dollar thresholds, any of the clauses below can be included on a contract should the risk warrant its inclusion. Despite the summary of policy included in this section, the clauses included in the contract awarded will drive contract execution. The list is provided for reference for applicability and is not intended to be a comprehensive set of instructions or exhaustive instructions for contracting for EVM and Agile and will vary by the issuing agency.

5.4.1 EVMS Requirement References

The source of acquisition requirements for an EVMS is the OMB Circular A-11, Supplement to the Capital Programming Guide. All subsequent federal and agency specific acquisition requirements reference OMB Circular A-11. The NDIA IPMD Earned Value Management Systems Application Guide summarizes the federal and agency specific acquisition documents that reference the OMB Circular A-11 EVMS requirements.

The (FAR) Subpart 34.2 (34.201, Policy) states: “An Earned Value Management System (EVMS) is required for major acquisitions for development, in accordance with OMB Circular A-11. The Government may also require an EVMS for other acquisitions, in accordance with agency procedures.” Agencies may define their EVMS requirements in agency supplements to the FAR with specific instructions, orders, and guides in accordance with the OMB Circular A-11. Agencies without supplemental guidance reference FAR Subpart 34.2 and the related FAR solicitation or contract clauses.

Agency specific notes:

- DoD applies the A-11 and FAR with the Adaptive Acquisitions Framework Document Identification (AAFDID), found at URL: https://www.dau.edu/aafdid/Pages/about.aspx. On the Major Capability Acquisition tab, towards the bottom, reference the EVMS Application Requirements (https://www.dau.edu/aafdid/Pages/EVMS-Application-Requirements.aspx) and the EVMS Reporting Requirements (https://www.dau.edu/aafdid/Pages/EVMS-Reporting-Requirements.aspx).
- DOE applies the A-11 with DOE Order 413.3B. This order sets the thresholds for EVMS certification and surveillance reviews. DOE Guide 413.3-10B (April 2022) describes how DOE implements EVM on DOE programs.
- NASA applies the A-11 with FAR supplements to fit NASA’s mission objectives. NASA’s supplement Part 1834 Major System Acquisition, Subpart 1834.2 EVMS, sets the thresholds for EVM requirements.

5.4.2 Performance Based Contracting References

When contracting for an Agile methodology, it is recommended to include provisions for performance-based contracting and use of a SOO. Policy documents such as OMB Circular A-11 and from the Office of Federal Procurement Policy (OFPP) encourage the use of Performance Based Contracting Acquisition (PBSA) and more specifically, FAR Subpart 37.6 describes “Performance-Based Acquisition”. FAR Part 37 Service requires the use of performance-based acquisition for services to the maximum extent practical and prescribes policies and procedures for use of performance-based acquisition methods. Two additional sources for reference are:

- OFPP Seven-Steps to Performance-Based Acquisition (guide/instructions for SOO, PWS and QASP etc.).

- DAU Service Acquisition Mall provides tools and templates to create a performance-based service acquisition requirements.

5.4.3 Contractual Reporting and Data Deliverables

Contract reporting is directed by contract clauses and data item requirements. Contract performance status and funds reporting is essentially unchanged from other contract types and consists of:

- Contract funding including Limitation of Funds, and Limitation of Cost as appropriate.
- EVM reporting and the Cost and Software Data Report (CSDR) with minor modifications discussed elsewhere in this document account for the inherent differences between waterfall and agile programs.

Other reporting requirements, especially in defense contracts, are established by various Data Item Description (DID) requirements which are typically assigned and cataloged on a DD Form 1423 – Contract Data Requirements List (CDRLs). Prescriptions for these DIDs are contained in agency-specific clauses or policy guidance.

In an Agile development contract, the working products being developed as a component of the final product is the primary deliverable. Consider modifications to the CDRL expectations given the iterative development fashion and the customer involvement in various activities, such as allowing for “as-built” CDRL’s or elimination of CDRLs no longer needed.

EVM reporting for most Agencies is based on the Integrated Program Management Report (IPMR) formats. For DoD contracts the primary DID specified in CDRLs is the Integrated Program Management Data Analysis and Report (IPMRDAR).

The Agile management tool performance details should fully support and underpin the EVM data, and the entire set of reporting and management data should work together to tell a consistent story and provide more accurate, timely and reliable data.

The following types of CDRLs are identified as being impacted by the Agile process and future guidance is forth coming to expand information:

- System Engineering CDRLs
- Design CDRLs (depending on contract)
- SW CDRLs
- Test CDRLs
- Training CDRLs
- Program Management CDRLs (including EVM IPMR/IPMDAR)
- Agile Reporting Metrics
- IMP (see Section 2.2)

5.5 Contractual Change in an Agile and EVM Environment

All Federal contracts are required to include one of the FAR (or Agency specific supplements) cited “changes” clause which asserts that the Buyer (Government) has a unilateral right to change specific aspects of the contract at any time for its sole convenience. This right is counterbalanced by the Seller’s (Contractor’s) right to request an “equitable adjustment” of the contract value and or to avail themselves of the claims process (via the Disputes Act).

The legacy in government contracting is managing requirements. When requirements are removed, there is an expectation that, consideration will be given to the buyer and the buyer often expects value returned. Conversely, when requirements are added to the contract, the seller expects to receive additional contract value, budget and funding corresponding to the increase in the requirement(s). Typically, when using an EVMS, these changes would also impact the PMB.

Not all changes to the project are "changes" from a contractual point of view or an EVM baseline change. Contractual changes are communicated in writing (usually via a SF30) signed by one or both parties to form a 'supplemental agreement' to the contract.

Some changes can be made outside the mechanism of the "changes" clause. These so called "Constructive Changes" are to be avoided. Examples of these include but are not limited to improper or excessive inspection / application of technical standards, failure to cooperate with the contractor, defective specifications or improper or inappropriate direction of government. Many "changes" do not rise to the level of a contractual change. Example of these include but are not limited to are simple performance trade-offs that do not materially change the terms or conditions of the contract, are resequencing of tasks or events that mutually benefit the parties and do not impact the contract schedule or cost, or are definitions of work that is to be done under the contract (e.g., "technical guidance") that do not change the contract schedule or cost.

Agile, due to its very nature allows (or often encourages) pivots in various directions as the work progresses and more is known. This characteristic can present contractual issues unless:

- Contractual requirements are stated in terms of desired capabilities or functional outcomes
- The work and/or cost are constrained through an appropriate contractual mechanism
- The CAM and Product Owner along with the PM should consider the types of change and be aware of the types of change within the Agile process execution and consult on a regular basis with the contracts officer to confirm the type of change

When interpreting change on an Agile and EVM contract, the fundamental consideration of each change should focus on the scope of the contract: Consider the highest level "requirement" or product. Is the highest-level product changing? Are the boundaries of the requirements or product purchases changing?

For example:

- If the contract is for a Pickup Truck and the customer determines an SUV is required, is this a variant or are they two different vehicles?
- The buyer and the seller should discuss whether this a variant or a new vehicle and agree to modify the contract. The buyer and seller should also consider if the change can be accommodated within the original scope, schedule and budget negotiated during the solicitation.
- If the joint buying and selling team, as coordinated with the product owner's visions decides to accept the change the contract should be modified, and the price should be negotiated through the official contracting authority. They should also ensure the corresponding requirements represent the latest definitized changes within the product boundaries.

5.5.1 Contracting Authority:

As of the writing of this document, the contractual authority in Government Contracting does not change when utilizing Agile and EVM together. The Product Owner does have the authority to make business value decisions that should be coordinated with the Buying Government Program

Management Office. The list below are the contributors to manage contractual change, with the ultimate signing authority being between the Buying Contracting Officer and the Selling Contracts Manager:

- Buyer: Contracting Officer (CO) / Procurement Contracting Officer (PCO)
- Buyer: Administrative Contracting Officer (ACO)
- Buyer: Contracting Officer's Representative (COR) is the authority for technical guidance, refinement of a technical process or technical definition
- Buyer: Government Program Management Authority
- Seller: Contracts Manager
- Seller: Contracting Program Manager

5.5.2 Program Management Process

There are several recommended Program Management Processes for managing all types of change outlined in the following section. Considering defining the necessary processes applicable to the agile solicitation, including:

- Agile Ceremonies – various team reviews at multiple levels and time increments for planning and demonstration as a mechanism for all stakeholders to see and accept incremental progress of the completion of the product, as directed by the Product Owner (customer "proxy"). Examples include Release Planning, Sprint Demos, Scrum Meetings, and Release Demos. Agile Ceremonies may supplement or replace the typical reviews being conducted today. Consider documenting how the Agile Ceremonies can be applied in lieu of the traditional examples cited below.
- Engineering Review Board (ERB) / Defect Review Board (DRB) – used to manage and review the technical components of the product / requirements
- Configuration Control Board (CCB) – used to manage and review the impacts that a potential technical change will have on schedule and budget
- Risk and Opportunity Management Review Board (ROMB) – used as a forum to identify risks and opportunities when planning the project and track the potential that a risk or opportunity will materialize and be mitigated or captured
- Program Reviews – a comprehensive review of scope completion within the schedule and budget of the solicitation
- Contractual Documentation – based on the outcome of the ERB, CCB, ROMB, Agile Ceremonies and Program Reviews, determine appropriate items to be coordinated through contractual change channels, such as letters, ECPs or Requirements Lists (Equivalency Swaps)

5.6 Systems Engineering Technical Reviews (SETR)

If a Program Management Office intends to embrace Agile methods on a DoD program, it will need to determine how to meet the criteria for the major milestone reviews, particularly System Requirements Review (SRR), Preliminary Design Review (PDR), Critical Design Review (CDR) and Test Readiness Review (TRR). Each of these reviews is typically a one-time event with entrance and exit criteria based on completion of the corresponding development phase. Conversely, Agile development emphasizes incremental development of system functionality through iterative execution of development phases for the duration of the program. Despite this difference in emphasis and method, Agile programs can utilize a tailored milestone review approach in which the reviews focus on the incremental progress of the system rather than the completion of development phases. In this way, the Agile program adopts a progressive technical review scheme, where each successive wave of reviews builds on its predecessors.

Table 5.4 below provides recommendations for adapting technical reviews on programs with an EVM requirement that are using the Agile development framework. The emphasis here is on characterizing the relationship between the adapted iterative technical practice and the associated EVM practices.

Technical Review	Purpose	Adapted Iterative Technical Practice	EVM practice
Kick Off	• Post Award Conference or Post Award Orientation. A Post Award Orientation aids both Government and contractor personnel to (1) achieve a clear and mutual understanding of all contract requirements, and (2) identify and resolve potential problems. However, it is not a substitute for the contractor fully understanding the work requirements at the time offers are submitted, nor is it to be used to alter the final agreement arrived at in any negotiations leading to contract award. The Post Award Orientation is encouraged to assist small business concerns; small, disadvantaged business concerns; veteran-owned small business concerns; service-disabled veteran-owned small business concerns; HUBZone small business concerns; and women-owned small business concerns. While cognizant Government or contractor personnel may request the contracting officer to arrange for orientation, it is up to the contracting officer to decide whether a Post Award Orientation in any form is necessary. Maximum benefits will be realized when orientation is conducted promptly after award. (cf. FAR subpart 42.5, 42.501 General.)	• Use the Post Award Conference to review the process associated with the Agile methodology. Product Owners and Stakeholders should attend to foster collaboration and communication. Conduct review of initial System Capabilities and Product Roadmap.	• Overview of EVM policies. Initial review PMB; mapping of Capabilities to PMB. • Leverage kick-off activities in support of ongoing IBR preparation. This will lead up to conducting the IBR.
SRR	• Ensure the level of understanding of top-level system requirements is adequate to support further requirements analysis and design activities, and that the system can proceed into initial system design with acceptable risk. (IEEE 15288-2)	**Adapted SRR** • Review top-level requirements, Development Plans (systems and software development plans), System Capabilities	• Update and refine PMB consistent with EVM change management policies based on SRR results, if required.

Technical Review	Purpose	Adapted Iterative Technical Practice	EVM practice
		Baseline, and Product Roadmap.	
PDR/CDR	• PDR: ensure the preliminary design for the system under review is sufficiently mature and ready to proceed into detailed design and can meet the stated performance requirements within program budget, schedule, risk, and other program and system constraints. • CDR: ensure that the detailed design for the system under review is adequate to proceed into fabrication, system integration, demonstration and test and can meet stated performance requirements within budget, schedule, risk, and other system constraints. (IEEE 15288-2)	**Incremental Progress Reviews** • Demonstration of completed product including insight into completed features and other development artifacts, such as architecture, requirements, design, and software. • Release Planning: Selection of features to be developed in the next increment.	• Earned value reported (BCWP) and Variance Analysis based on product completed to date as presented at incremental progress reviews. • Rolling Wave Planning: update and refine PMB based on increment planning results, if required.
TRR	• Assess test objectives, test methods and procedures, test scope, safety, readiness for acquirer and supplier development test and evaluation (DT&E), and whether test resources have been properly identified and obtained. (IEEE 15288.2)	• **Internal Test Event reviews:** Integrated with the Incremental Progress review described above. Includes insight into incremental test artifacts and results. • Final internal test event review: demonstration of lower-level specification selloff; may be combined with incremental progress review. • **External Test Event reviews:** higher-level specification selloff; results part of traditional government-led DT test event reviews. For these formal test events, there may be multiple TRRs held to achieve the system stability and removal of system defects. These formal test events may be more waterfall in nature, with incremental test cycles / sprints to execute the required test procedures to ensure system safety and worthiness	• Internal: Earned value reported (BCWP) and Variance Analysis is based on product testing completed as presented at incremental progress reviews. • External: EVM and EVTs tied to higher-level specifications CA and WP. Earned value reported (BCWP) and Variance Analysis is based on progress made towards completion of formal test events

Table 5-4: Technical Reviews adapted for Agile development

Figure 5-1 below displays the timeline for both traditional and Agile SETRs as well as relationship between Agile SETRs and associated Program Management activities. This diagram is derived from figure 4 of SEI/CMU *RFP Patterns and Techniques for Successful Agile Contracting*.[i]

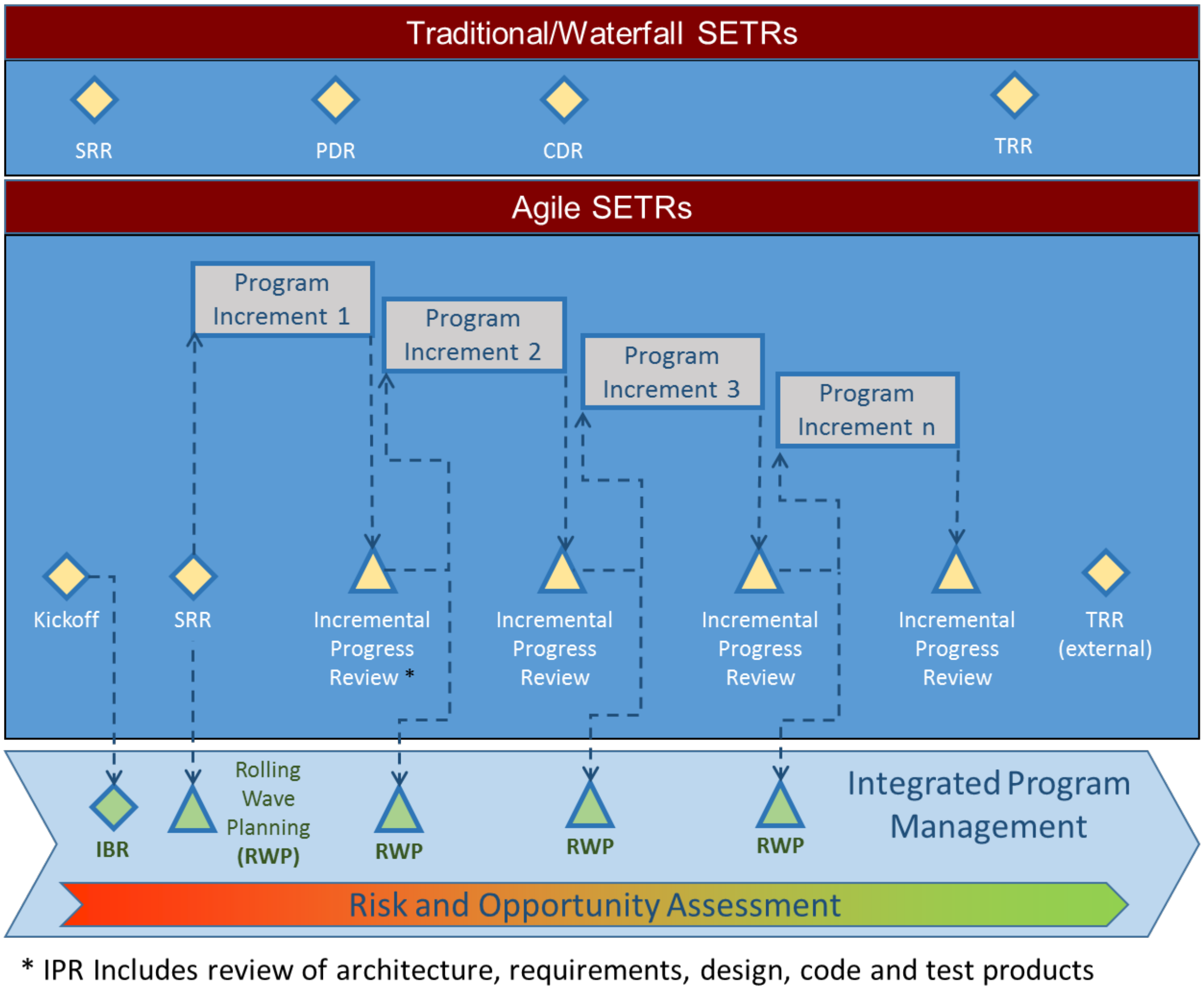

Traditional/Waterfall SETRs
SRR
PDR
CDR
TRR
Agile SETRs
Program Increment 1
Program Increment 2
Program Increment 3
Program Increment n
Kickoff
SRR
Incremental Progress Review *
Incremental Progress Review
Incremental Progress Review
Incremental Progress Review
TRR (external)
IBR
Rolling Wave Planning (RWP)
RWP
RWP
RWP
Integrated Program Management
Risk and Opportunity Assessment
* IPR Includes review of architecture, requirements, design, code and test products

Figure 5-1: Timeline for traditional and Agile SETRs

i. Agile Acquisition and Milestone Reviews, Copyright 2017 Carnegie Mellon University. All Rights Reserved.

ii. RFP Patterns and Techniques for Successful Agile Contracting, CMU/SEI-2016-SR-025

Appendix A – Agile Data Dictionary

AGILE TERM	AGILE DEFINITION
Agile	A holistic approach and mind set to working that encourages collaboration, self-organizing teams focused on outcomes to deliver value that meets customer needs with a predictable, steady flow.
Agile Tool	A tool that supports Agile ways of working is usually one that helps keep work visible. There isn't a single tool that is recommended today. Larger programs typically use a common tool for consistency.
Agile Release Train	The Agile Release Train (ART) is a long-lived team of Agile teams, which, along with other stakeholders, incrementally develops, delivers, and where applicable operates, one or more solutions in a value stream.
Burndown Chart	The trend of work remaining across time in a Sprint, a release or in a product. The burn down chart is a publicly displayed chart showing remaining work in the Sprint Backlog. Updated every day, it gives a simple view of the Sprint progress.
Backlog Grooming	The team (or part of the team including the PO) meet regularly to "groom the product Backlog", in a formal or informal meeting which can lead to any of the following: • removing Stories that no longer appear relevant • creating new Stories in response to newly discovered needs • re-assessing the relative priority of Stories • assigning estimates to Stories which have yet to receive one • correcting estimates in light of newly discovered information • splitting Stories which are high priority but too coarse grained to fit in an upcoming Sprint
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 program. See Product Backlog.
Buyer	Buyer should be considered as the Government Customer. The individual with the contracting authority represents the buyer for legal purposes but the "Buyer" is in fact the entire customer team
Cadence	Refer to definition for Release:
Capability	Capability and Epics are used interchangeably in this Guide. Both are recognized as customer required abilities of the system that provide value and is associated with specific Feature(s) and their Stories that must be satisfied for its completion.
Daily Scrum Meeting	A short status meeting held daily by each team. Team members synchronize their work and progress and report any impediments to the Scrum Master for removal.
Definition of Done	Complete as mutually agreed to by all parties and conforming to an organization's standards, conventions, and guidelines. Note that an outcome of product and Release Planning is to create a Definition of Done, which equates to IMP accomplishment criteria, for Epic/Capabilities and Features respectively. Typically, this is a checklist.

AGILE TERM	AGILE DEFINITION
Demo	A demo is a key part of Agile practices which involves an Agile team demonstrating work accomplished. There are team and organizational demonstrations that should provide an integrated view of the work accomplished. This typically includes a demonstration (not a conceptual discussion or presentation) of the product to the customer in order to receive feedback and ensure the outcome meets the customer needs. A demo may be informal or include formal "sell-off" and formal acceptance of the functionality by the customer.
Epic	Epics may represent core business capabilities which are defined by the customer or stakeholders. A large grained definition of a need that will likely take more than one release to complete. Can be split into Features and eventually Stories. Epics are part of the product Backlog and should have some form of relative sizing estimate. Capability and Epics are used interchangeably in this Guide.
Feature	A discrete or coherent functionality within an Epic/Capability, scheduled to be completed within a release (cadence or Capability), and comprised of a collection of logically cohesive Stories. All Features should have clearly defined objective technical completion criteria. This is the lowest level of earned value baseline scope definition.
Handoff	A significant interdependency where the owner of the predecessor task is different than the owner of the successor task. Handoff tasks are critical to ensuring schedule integration and on-time performance. As a result, it is important that these tasks are clearly identified and visible to Agile teams, CAMs, and program management. Internal Handoff is a significant interdependency between two CAMs, internal to the company. External Handoff of a significant interdependency between a company CAM and a supplier or customer. GFE/GFI deliveries are also considered External Handoffs. Handoff Task the predecessor or provider activity in the Handoff relationship.
Increment	Synonymous with **Release** Specific timing and purpose of Increment defined by each Agile Implementation.
Iteration	Synonymous with Sprint
Lean	As defined by Wikipedia, Lean Six Sigma is a methodology that relies on a collaborative team effort to improve performance by systematically removing waste and reducing variation. It combines lean manufacturing/lean enterprise and Six Sigma to eliminate the eight kinds of waste: Defects, Over-Production, Waiting, Non-Utilized Talent, Transportation, Inventory, Motion, and Extra Processing
Product Backlog	The master list of all functionality at the Epic/Capability and Feature level that is desired in the product and any other elements needed to produce the product, even if not in the final product. Product Backlog is prioritized from most to least important. The authoritative source that contributes to product completion and may be referred to as the "Backlog". The Product Backlog may have different views, for example a sprint or release backlog. • if an item on the Backlog does not contribute to the program's goal, it should be removed. • on the other hand, if at any time a task or Feature becomes known that is considered necessary to the program, it should be added to the Backlog. • this Guide may not be specific in every case, as the type of "Backlog" may be different based on the Agile implementation

AGILE TERM	AGILE DEFINITION
Product Backlog Planning	A process in which the team maps the product Epic/Capabilities to Features that are to be accomplished based on customer agreement that specifies what the product must do and when the functionality will be delivered within a timeboxed schedule. It is a continuous control activity that encompasses the entire product goals of the program.
Product Owner (PO)	The person responsible for maintaining the Product Backlog by representing the interests of the stakeholders. The product owner is a new role to be established when contracting for Agile. The role of the product owner may start to be defined and included as part of the solicitation of the contract. The definition should include identifying if the buyer or the seller is supplying the PO. Best practice recommends that the buyer is responsible to identify and provide a product owner. The full set of responsibilities may be finalized as part of kick-off. Consider implementing a service level agreement.
Release[13]	"Release" is a concept associated with incrementally maturing the implementation of the system. There are several types of releases and corresponding standard cadence time boxes. The authority to release is specific to each program and who has the authority to deploy to operations. • "Release" – working software, deliverable, solution, or product released on a regular or timeboxed schedule. Timebox length varies by program, based on the program's goals (can be referred to as build or increment). "A grouping of Epics/Capabilities or Features that can be used for demonstration, evaluation, or delivery. A release may be internal for integration, testing, or demonstration; or external, to system test or as user delivery. A release may be used on a time block or on product maturity" • Internal Release – "A release that is ready for internal use outside of the development team. It may be used for integration, testing or demonstration" • Candidate Release – "or External Release – a release that has been through the pipeline and systems test, and is ready for transition to the user" • Operational Release – "or Deployment Release – a release that has been approved for operational use" The content of the release is determined through Product Backlog refinement/Release Planning.
Release Planning	A process in which the team maps the product Backlog Epic/Capabilities to Features and Stories that are to be accomplished based on customer agreement that specifies what the product must do and when the functionality will be delivered within a timeboxed schedule. Specific timing and purpose of Release Planning defined by each Agile Implementation.
Scrum	An incremental product development methodology commonly used to manage the program when applying Agile practices. A Scrum team works in a highly collaborative and team centric manner to achieve the team objectives.

[13] The definition of Release is consistent with the Practical Software and Systems Measurement Continuous Iterative Development Measurement Framework, Part 1, Version 2.1 dated 15 April 2021 (https://www.psmsc.com/Downloads/CIDProducts/CID%20Measurement%20Framework%20Part%201%20-%20v2-1.pdf)

AGILE TERM	AGILE DEFINITION
Scrum Master	The person responsible for the Scrum process, making sure it is used correctly and maximizes its benefits. Scrum is facilitated by a Scrum Master, whose primary job is to remove impediments to the ability of the team to deliver the Sprint goal. The Scrum Master is not the leader of the team (as they are self-organizing) but acts as a buffer between the team and any distracting influences. The Scrum Master ensures that the Scrum process is used as intended.
Scrum Team	The Scrum Team is made up of the PO, Scrum Master and Team.
Seller	Refers to the contractor providing the solution and product requested by the buyer.
Statement of Objective (SOO)	Provides basic, top-level objectives of an acquisition and is provided in the request for proposal (RFP) in lieu of a government-written statement of work (SOW).
Sprint	A timebox of work for which the duration is defined by the team and related to their optimal work cadence. Sprint durations are typically fixed and are usually between 1 and 6 weeks in duration. During the Sprint, the team works to turn a portion of the Product Backlog it has selected into an increment of potentially shippable product functionality.
Sprint Backlog	A view of the Product Backlog. A list of tasks to be completed during the Sprint.
Stakeholder	Someone with an interest in the outcome of a program, either because he or she has funded it, will use it or will be affected by it.
Story (User Story)	Part of a Feature that can be estimated in Relative size and complexity and prioritized in Sprint Backlog. Stories are sized to fit within a Sprint. The completion of Stories can be used to calculate earned value.
Story Points (estimates in)	Agile teams may express estimates in units of “Story Points” (SP), providing for the use of Story Point Velocity for planning purposes. "Velocity", in the sense Agile teams use the term, has no preferred unit of measurement. Velocity allows teams to compute the expected remaining duration of the program, as a number of Sprints, each Sprint delivering some amount of Features. Another important reason has to do with the social and psychological aspects of estimation: using units such as Story Points to estimate a weighted Story Value, emphasizing relative difficulty over absolute duration, relieves some of the tensions that often arise between developers and managers around estimation: for instance, asking developers for an estimate then holding them accountable as if it had been a firm commitment.
Team	A cross-functional group of people that is responsible for managing itself to develop product for every Sprint. Team members’ work together consistently in a predefined pattern. In contrast to traditional methods that bring people in as needed.
Timebox or Timeboxed	A period of time that cannot be exceeded and within which an event or meeting occurs. An example is the Daily Scrum meeting which is typically timeboxed to 15 minutes and ends at that time regardless.
Velocity	At the end of each Sprint, the team adds up effort associated with Stories that were completed during that Sprint. This total is called velocity. (Completed weighted Story Value in Story Points / Sprint Length) Knowing velocity, the team can compute (or revise) an estimate of how long the program will take to complete, based on the estimates associated with remaining Stories and assuming that velocity over the remaining Sprints will remain approximately the same.

Appendix B - Examples of Progress Tracking Charts with Agile and EVM

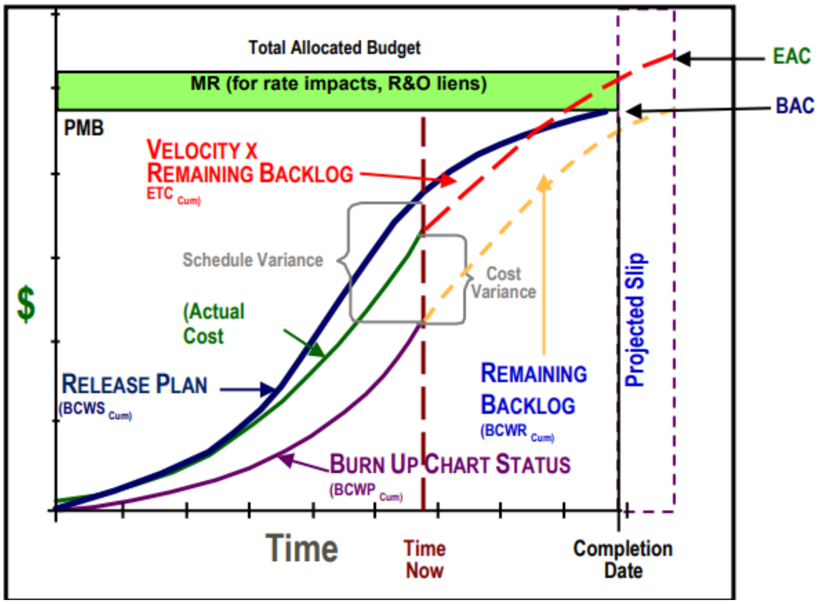
Graphs can be created that overlay Agile program data metrics on the EVM calculations and metrics to show how Agile may be used to perform EVM analysis for a program with the Agile-EVM model of application. For example, a graph connecting Story Points (associated with completed weighted Stories) burn-up status with Performance Management Baseline (PMB) type data as a plot. This is illustrated in an example below in Figures B-1, B-2 (Lockheed Martin Platinum Card) that illustrates both PMB and completed Story burn-up in Story Points. It is recommended that this example be applied according to each individual corporate or agency specific Agile and EVM implementation.

Figures B-1 and B-2 are the copyright of Lockheed Martin Corporation and are included in this Guide for sample reference. Some of the formulas documented in the Lockheed Martin Platinum Card may be worded slightly differently than other sections of this Guide, as the Guide contains updates. The intent of the formulas is the same, Guide updates may include clarified wording. Figure B-3 is a slightly different example from Rockwell Collins that shows explicitly the progress as measured via Story Points associated with completed Stories in the Agile Tool as “bars” on the graph as indexed by the left axis, along with the costs as indexed by the right axis to be able to visually see any disconnects or trends. Figure B-4 shows an example program-level remaining weighted Stories burndown chart in Story Points courtesy of Raytheon with both overall status and forecasted Sprint iteration number the program will complete.

All figures were used with permission during the original publication of the guide and are samples that will not be updated. All figures are for reference for other companies to implement and update according to each corporate environment. These figures are not provided to be copied.

PLATINUM CARD

EVM FOR AGILE DEVELOPMENT



Variances Positive is Favorable, Negative is Unfavorable

Cost Variance CV = Burn Up Status – Actual Cost (BCWP – ACWP)

CV % = (CV / BCWP) * 100

Schedule Variance SV = Burn Up Status – Release Plan (BCWP – BCWS)

SV % = (SV / BCWS) * 100

Variance at Completion VAC = BAC – EAC

VAC % = (VAC / BAC) * 100

DoD Metrics

Favorable is > 1.0, Unfavorable is < 1.0

Cost Efficiency CPI = Burn Up Status / Actual Cost (BCWP / ACWP)

Schedule Efficiency SPI = Burn Up Status / Release Plan (BCWP / BCWS)

Program Agile Team Estimate @ Completion

ETC = Velocity x Remaining Backlog

EAC = Actual Cost + (Velocity x Remaining Backlog)

Independent Estimate @ Completion #

= ACTUALS TO DATE + [(REMAINING WORK) / (PERFORMANCE FACTOR)]

EAC_{CPI} = $ACWP_{CUM}$ + [Remaining Backlog / CPI_{CUM}]

= $ACWP_{CUM}$ + [(BAC – $BCWP_{CUM}$) / CPI_{CUM}]

$EAC_{Composite}$ = $ACWP_{CUM}$ + [Remaining Backlog / (CPI_{CUM} * SPI_{CUM})]

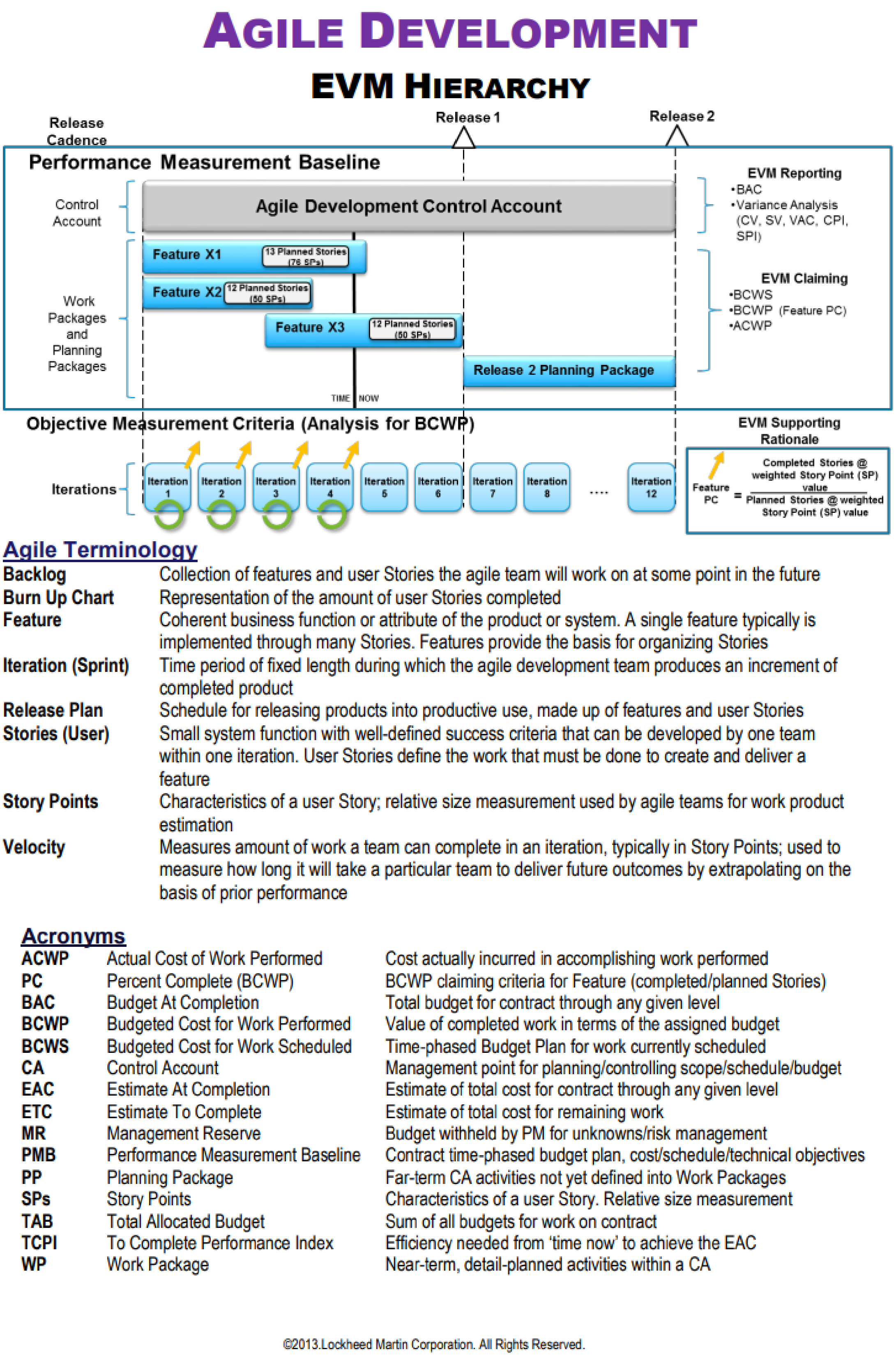
= $ACWP_{CUM}$ + [(BAC – $BCWP_{CUM}$) / (CPI_{CUM} * SPI_{CUM})]

To Complete Performance Index (TCPI) = Work Remaining / Cost Remaining

$TCPI_{EAC}$ = Remaining Backlog / (Velocity * Remaining Backlog)

= (BAC – $BCWP_{CUM}$) / (EAC – $ACWP_{CUM}$)

Figure B-1: Front side of example "Platinum Card" for integrating Agile and EVM, indicating both Agile (Burn-Up) and EVM (PMB) baseline plan and progress data.



AGILE DEVELOPMENT

EVM HIERARCHY

Release Cadence
Release 1
Release 2
Performance Measurement Baseline
Control Account
Agile Development Control Account
Work Packages and Planning Packages
Feature X1
13 Planned Stories (76 SPs)
Feature X2
12 Planned Stories (50 SPs)
Feature X3
12 Planned Stories (50 SPs)
Release 2 Planning Package
TIME
NOW
EVM Reporting
•BAC
•Variance Analysis (CV, SV, VAC, CPI, SPI)
EVM Claiming
•BCWS
•BCWP (Feature PC)
•ACWP
Objective Measurement Criteria (Analysis for BCWP)
Iterations
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Iteration 7
Iteration 8
....
Iteration 12
EVM Supporting Rationale
Feature PC = Completed Stories @ weighted Story Point (SP) value / Planned Stories @ weighted Story Point (SP) value

Agile Terminology

Backlog	Collection of features and user Stories the agile team will work on at some point in the future
Burn Up Chart	Representation of the amount of user Stories completed
Feature	Coherent business function or attribute of the product or system. A single feature typically is implemented through many Stories. Features provide the basis for organizing Stories
Iteration (Sprint)	Time period of fixed length during which the agile development team produces an increment of completed product
Release Plan	Schedule for releasing products into productive use, made up of features and user Stories
Stories (User)	Small system function with well-defined success criteria that can be developed by one team within one iteration. User Stories define the work that must be done to create and deliver a feature
Story Points	Characteristics of a user Story; relative size measurement used by agile teams for work product estimation
Velocity	Measures amount of work a team can complete in an iteration, typically in Story Points; used to measure how long it will take a particular team to deliver future outcomes by extrapolating on the basis of prior performance

Acronyms

ACWP	Actual Cost of Work Performed	Cost actually incurred in accomplishing work performed
PC	Percent Complete (BCWP)	BCWP claiming criteria for Feature (completed/planned Stories)
BAC	Budget At Completion	Total budget for contract through any given level
BCWP	Budgeted Cost for Work Performed	Value of completed work in terms of the assigned budget
BCWS	Budgeted Cost for Work Scheduled	Time-phased Budget Plan for work currently scheduled
CA	Control Account	Management point for planning/controlling scope/schedule/budget
EAC	Estimate At Completion	Estimate of total cost for contract through any given level
ETC	Estimate To Complete	Estimate of total cost for remaining work
MR	Management Reserve	Budget withheld by PM for unknowns/risk management
PMB	Performance Measurement Baseline	Contract time-phased budget plan, cost/schedule/technical objectives
PP	Planning Package	Far-term CA activities not yet defined into Work Packages
SPs	Story Points	Characteristics of a user Story. Relative size measurement
TAB	Total Allocated Budget	Sum of all budgets for work on contract
TCPI	To Complete Performance Index	Efficiency needed from 'time now' to achieve the EAC
WP	Work Package	Near-term, detail-planned activities within a CA

Figure B-2: Back side of example "Platinum Card" for integrating Agile and EVM.

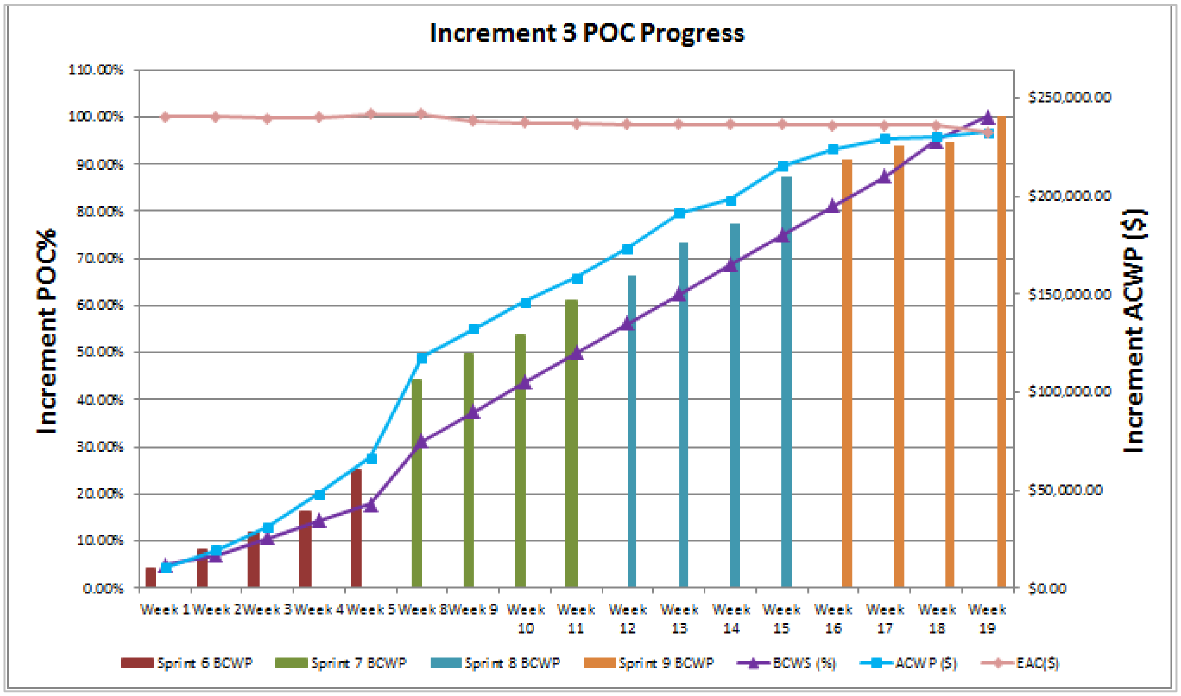


Figure B-3: Example of a progress tracking report indicating both Agile and EVM progress data on graph.

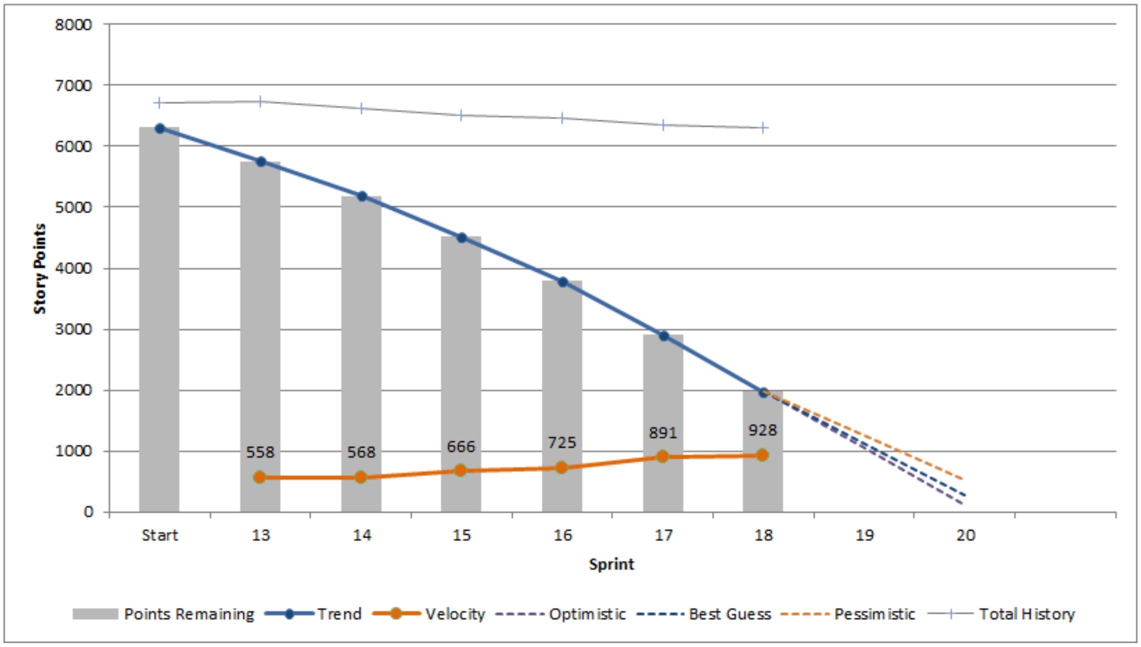


Figure B-4: Example of a program level burndown chart across multiple teams, indicating overall status and predicted completion Sprint.

Appendix C - References

1. *Agile Planning and Estimating*, Mike Cohn, 2006, Pearson Education, Inc., ISBN 0-13-147941-5
2. *Scaling Lean & Agile Development*, Craig Larman, Bas Vodde, 2009 Pearson Education Inc., ISBN-13: 978-0-321-48096-5
3. *The Software Project Manager's Bridge to Agility*, Michele Sliger and Stacia Broderick, 2008, Pearson Education, Inc., ISBN-13: 978-0-321-50275-9
4. *A Compendium on The Application of EVM to Agile Development and The Application of Agile Development to EVM*, Glen Alleman, Eric Christoph, Luis C Contreras, Andrea Nibert, Ray W Stratton, Ron Terbush, Howard Zillman, The Measurable News, 2015.01 Issue, pp 11-38.
5. *Digital Playbook – TechFAR Handbook*, draft
6. Dean Leffingwell et al, the Scaled Agile Framework® creator's website; specifically, Story Point normalization method under the section titled Normalizing Story Point Estimating:
7. Scaled Agile Framework® (Leffingwell, et. all). The Scaled Agile Framework® ("SAFe®") is a proven knowledge base for implementing Agile practices at enterprise scale.
8. *DoD EVMS Interpretation Guide*, EVMS GUIDELINES: PLANNING, SCHEDULING, AND BUDGETING and REVISIONS AND DATA MAINTENANCE CATEGORIES: Guidelines (10, 28 – 32)
9. The Software Engineering Institute and Carnegie Mellon University Published the *RFP Patterns and Techniques for Successful Agile Contracting*
10. The Department of Defense's (DoD) Integrated Master Plan and Integrated Master Schedule (IMP/IMS) Preparation and Use Guide Version 0.9
11. DoDI 5000.02
12. AAP Desktop Agile and EVM Guide for PMs
13. Seven-Steps to Performance-Based Acquisition (guide/instructions for SOO, PWS and QASP etc.)
14. DAU Service Acquisition Mall provides tools and templates to create a performance-based service acquisition requirements

General References on EV Systems, Program Management, and Work Breakdown Structure

1. *MIL-STD-881, Department of Defense Standard Practice: Work Breakdown Structures for Defense Materiel Items*, (Current Version)
2. *DoD IPMR Implementation Guide*, February 2016
3. *DoD IPMDAR Implementation Guide, August 2021*
4. *PMBOK, A Guide to the Project Management Body of Knowledge*, PMI
5. *Practice Standard for EV*, PMI
6. *Practice Standard for Work Breakdown Structures*, PMI
7. *DoD Instruction 5000.02, Operation of the Defense Acquisition System is now titled "Operation of the Adaptive Acquisition Framework"*
8. *SAE International EIA-748-D EV System (EVMS) Standard* (EIA-748)
9. *EV Systems EIA-748-D Intent Guide*, August 2018, NDIA
10. Acquisitions Analytics and Policy (AAP) EVM Division
11. Scrum Alliance
12. The Agile Alliance
13. *The Software Project Manager's Bridge to Agility*, Sliger, Michele; Broderick, Stacia, 2008

Appendix D - Product Roadmap, Release Planning, and Rolling Wave Planning Products

This appendix elaborates on the Agile project planning process and integrating it with the EVM planning process introduced in Sections 1.2, 2.1, 2.2, and 3.3.

Product Planning: Product Backlog and Product Roadmap

The Product Backlog is the prioritized list of system functionality required for the project or program. The Product Roadmap is the time-phased delivery plan for the functionality in the Product Backlog. The Product Roadmap is also referred to as the "Program Roadmap" or "Release Roadmap".

The Product Backlog and Product Roadmap are created during Product Planning, the initial program planning performed, usually during the proposal time frame or at program start, at the latest. During Product Planning, the Product Owner(s) and customer representatives specify and prioritize the initial set of system Epics/Capabilities needed to deliver the contractually required system, thus forming the initial Product Backlog. The System Epics/Capabilities are then prioritized into Releases and aligned with the customer deliveries, thus forming the Product Roadmap. The Epics/Capabilities shown in the Product Roadmap reflect the full program scope (as defined in the Statement of Work or Statement of Objectives). Note that some Epics/Capabilities flow into Customer Deliveries with defined dates that may not coincide with the completion of a particular Release. See Figure D-1 below for an example Product Roadmap.

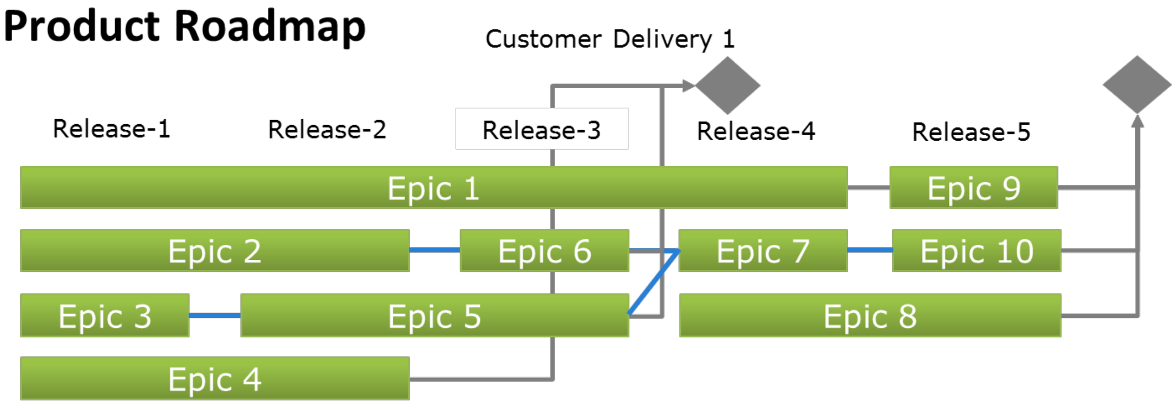


Figure D-1: The initial Product Roadmap completes the Epics/Capabilities planning and incorporates customer delivery milestones.

Release Planning: Release Plan

With the initial Product Backlog and Product Roadmap established, the program conducts Release Planning. The objective of Release Planning is to establish the functionality to be implemented within the program's next Release. In Release Planning, the Product Owner(s) decompose Epics/Capabilities from the Product Roadmap into a lower-level expression of system functionality called Features. A Feature is a piece of an Epic/Capability that can be completed within one Release. This sizing to one Release is what distinguishes the Feature from its associated Epic/Capability. The Release Plan then is the set of Features planned to be implemented in that Release. In Figure D-2 the Product Roadmap includes the Features for the first Release.

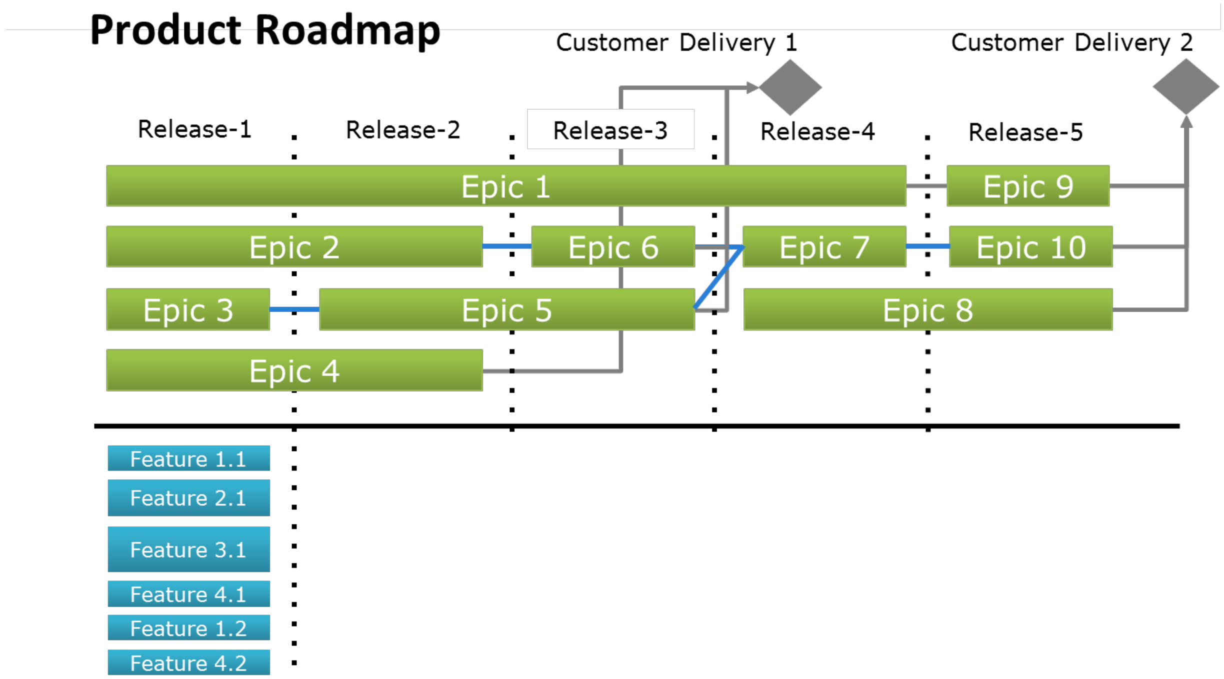


Figure D-2: The updated Product Roadmap completes the Features planning for Release-1.

It is often the case that programs desire to have a Feature-level view of the Product Roadmap beyond the current or just-planned Release. In this case, the program establishes broadly defined Features for future Releases. In Figure D-2, the Product Roadmap shows the Release Plan for the Release as well as initial Features for Releases 2 and 3. Programs are cautioned that planning Features beyond the next Release can add unnecessary and wasteful work to keep the detailed plan up to date because of emerging or changing Customer needs and other knowledge gained from the execution of the earlier Releases. Where a program has well-defined, predictable, and stable product definition and customer needs for the duration of the program, it may be appropriate to plan to the Feature level of detail for the whole program, and periodically review the Product Roadmap at Release Planning events for currency and needed updates.

The Product Roadmap must also be of appropriate detail to model key product dependencies (as shown with the Epic/Capability dependencies in Figure D-3) to demonstrate the critical path. Product Roadmap updates may impact the EVM Performance Measurement Baseline and should be dealt with per the company's EVM System Description for baseline change management. As needed, results from Release Planning events are fed into subsequent IMS rolling wave planning activities to update and synchronize the Agile and EVM planning products.

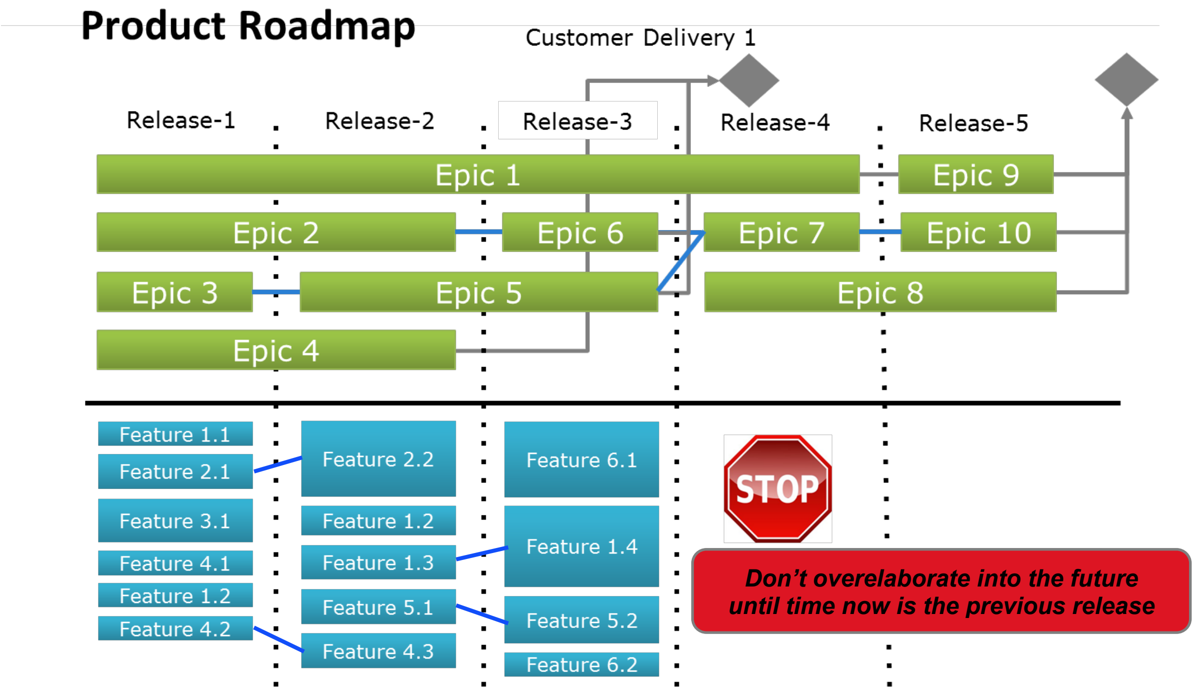


Figure D-3: An alternate updated Product Roadmap with planning to three Releases out.

Figure D-3 provides an alternative updated Product Roadmap that completes the Feature planning for Release-1 and provides initial Features for Release-2 and Release-3. Planning three releases out could be done given stable Epic/Capability plans.

The process to create and maintain a Product Roadmap includes the following steps:

1. Create, size, and prioritize Epics/Capabilities, which provide the highest level of product definition in the Product Backlog for the full scope of work.
2. Bin the Product Roadmap Epics/Capabilities into Releases based on factors such as priority (to maximize value delivery), product dependencies, and risk reduction. Include any fixed-date customer milestones and show product dependencies to support them.
3. Decompose, size, and prioritize near-term Epics/Capabilities into Features for the first 2-3 releases, or longer, as needed to understand key product dependencies.
4. Refine the Product Roadmap with those decomposed products.
5. Review the Product Roadmap with the customer and other key stakeholders to gain concurrence on this high-level program plan.
6. Periodically review and update the Product Roadmap, nominally in alignment with Release Planning events, filling in upcoming releases with Epics/Capabilities decomposed into Features from the updated Product Backlog. Some Features in future Releases may not be completely decomposed; each ensuing Release Planning event for that release completes the Feature decomposition, updating both the Product Roadmap and Product Backlog.

Note the granularity of a Product Roadmap depends on the size of the program – a small program with one or two Agile teams may only need a single page Product Roadmap while a 40-team SAFe® -based program with multiple major value streams requires something much more substantial.

Aligning the Release Planning Results with the EVMS PMB

The following activities are generally necessary to define and maintain traceability between the Agile management tool and the Performance Measurement Baseline to support EVM, and further validate the Product Backlog satisfies a program's contract Statement of Work or Statement of Objectives. Note the initial traceability and mapping of high-level Agile products (e.g., Epics/Capabilities) to control accounts should have been established at program start to define the Performance Measurement Baseline. The activities listed below should be accomplished or revisited to maintain the performance measurement baseline upon completion of each Release Planning event. They should occur before the start of the Release planned work.

Product Backlog Activities:

- Mapping or re-Mapping of Features to Work Packages or Planning Packages (e.g., each Feature has a WP attribute, with the value set to the specific WP for that feature). This mapping/re-mapping activity is primarily an exercise in successive and iterative refinement to the established baseline.
- Optional: Mapping of Features to the Release (e.g., each Feature has a Release attribute, with the value set to the specific Release for that feature). This is useful for determining Feature status on a Release basis. For example, you may want to know the Feature Percent Complete of all Features in Release 3.
- Update any tools used to determine EV percent complete with the new Features (e.g., Agile Management tool or Excel workbooks).

Integrated Master Schedule Activities:

- The IMS is updated with new work packages for the rolling wave; the rolling wave and IMS updates are aligned with Release Planning.
- Rolling wave Baseline Change Requests are approved and Work Authorizations signed off.
- Optional (but very helpful): Conduct a rolling wave out brief with Control Account Managers/Product Owners (CAMs/POs) that reviews the mapping of WP to CAMs/POs, mapping of Features to work packages, as well as work package budgets and periods of performance. This sets the expectations of EVM impacts as a result of rolling wave planning, which was informed by the completed Release Planning event.

Prior to Starting a Work Package:

- Verify stories have been created for all Features in the work package so that Percent Complete can be calculated.

Context and Role of the Product Roadmap, Relationship to the IMS

The Product Roadmap often forms the foundation for the IMS. The Product Roadmap shows the planned sequence of product development, includes key product dependencies and relationships to customer milestones, and provides a basis for subsequent rolling wave planning. The different and complementary roles of the Product Roadmap and IMS are summarized in this section.

The Product Roadmap can precede and inform IMP and IMS development, and even supplement the IMP when Definition of Done and assignment of Events, Accomplishments, and Criteria are completed. The Product Roadmap defines the sequence of work related to product elements or capabilities which require effort to complete along with their top-level timeframes. Thus, the initial Product Roadmap at the Epic/Capability level should be developed to define the required work at a summary level before the IMS is developed to define activities and logic. As the lower-level details in the Product Roadmap are generated, including Features for nearer-term Releases, the IMS can be generated shortly thereafter in an initial planning or rolling wave activity. The IMS is synchronized with the Product Roadmap in terms of major dependencies, sequences of work, and coordination of Release Planning events to rolling wave events.

The networking logic in the IMS, often at a work package detailed level in near term and at a planning package level in following rolling wave periods, allows critical path analysis. Equivalently the Product Roadmap captures dependencies and sequences at a top level throughout the program (Epic/Capability level). However, the Product Roadmap sequence can, where no dependency dictates otherwise, also reflect a product element's priority for value delivery as well as its predecessors and successors.

The IMS tasks have a defined duration, which in the Product Roadmap is initially only defined at the Epic/Capability level (Epic/Capability duration defined as an integer number of releases). Features are binned into a particular Release and no duration is assigned. This dissociation of work from duration and restriction of detail planning to only the nearest few Releases originated from the low predictability for more detailed work and for work planned to take place in the longer term. Similarly, rolling wave planning to flesh out IMS planning package summary tasks reflects the lack of predictability in longer term and more detailed tasks. The IMS tasks only reflect the planned Features with baselined durations at the completion of Release Planning and rolling wave planning for the upcoming release.

The process of reviewing and updating the Product Roadmap and the IMS should be designed to be synchronized and complementary. When a Release Planning event is completed, and the sequence and definition of work to build product elements/capabilities is documented, impacts to the IMS can be flowed into a subsequent rolling wave planning event or as a schedule change subject to approvals defined by the EVM System Description. Care must be taken to promptly recognize and capture impacts from the Release Planning events into the EVMS performance measurement baseline as needed before the pertinent work starts. This time-sensitive flow avoids timing conflicts with the freeze period (See Sections 3.4 and 5.5) and avoids significant lag between the work planned and the work contained in the performance measurement baseline. Reconciliation of planning and financial business rhythms, as well as review of the EVM System Description, is warranted to achieve a smooth and timely flow from work planning to execution.

Appendix E – IBR Considerations

Initial Baseline Review (IBR) considerations for a program implementing Agile Development Methodologies.

An important event for any program starting up a new scope of work is a comprehensive review of the program plan to confirm that the *"performance measurement baseline covers the entire scope of work, the work is realistically and accurately scheduled, the proper amount and mix of resources have been assigned to tasks, and proper objective indicators have been selected for measurement of task accomplishment." (NRO IBR Team Handbook)* The Initial Baseline Review (IBR) is focused on the achievability of the program plan. It is not a process review.

Purpose: The purpose of this section is to provide the program reviewer with a list of artifacts and processes that can be used to augment standard IBR artifacts when evaluating programs implementing Agile methods. Accordingly, the matrix below is not a comprehensive IBR checklist, but is limited to items that support the portions of the plan related to Agile methods.

Value Statement: The value in the information below is that it provides prompts for the reviewer on areas to explore and questions to ask when looking at Agile artifacts in relation to evaluating the soundness of the program plan.

Assumptions: Items in the matrix provided represent the artifacts and processes described elsewhere in this NDIA integrating Agile and EVM Guide. For programs whose Agile implementations differ from what is described in this Guide, some, or all of the items in the matrix below may not apply.

The columns in the table are set up as follows:

- IBR Project Management Constraints (adapted from: A Systems Approach to Planning, Scheduling, and Controlling, 6th edition; Project Management Institute, Project Management Body of Knowledge)
- Area of Focus: Topics to be explored in the focus area related to baseline achievability.
- Typical IBR Artifacts: Artifacts that support the Area of Focus discussion.
- Agile Specific Artifacts or Processes: Unique to "agile" tools, artifacts and processes that would provide the information that support the Area of Focus discussion.
- Attributes of Agile Artifacts or Processes: Content in the artifact or process would indicate a robust well-thought-out plan.

IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
Scope	Ensure the program has captured all the customer requirements, including an understanding of the operational concept	• SOW • WBS/Dictionary • IMP • WADs • MOD	• Product Backlog	Product Backlog: • At a minimum, contains a set of work items (typically called Epics / Capabilities) that cover the full breadth of the contract's technical scope. • Product Backlog items map to the WBS • Product Backlog items have size estimates* and acceptance criteria ** • Requirements (top level specs, SOW) are mapped to Backlog items to demonstrate the Product Backlog encompasses the full scope of work
Time	Ensure the program has a viable IMS that supports the IMP, meets required integrity standards and demonstrates execution realism	• Contract Milestones • Program Summary Master Schedule • IMS • Schedule Risk Analysis	• Product Roadmap	• Product Roadmap: ○ Scope is included at a reasonable level of fidelity (Epic/Capability) and that there is a reasonable ordering of that scope over time. ○ Product Roadmap shows sequencing of scope and alignment to program milestones. Detail should be sufficient to facilitate critical path in the IMS, ○ Product Roadmap includes scope item size estimates ○ Product Roadmap consistent with staffing plan based on Product Roadmap item size estimates • IMS baseline is informed by the Product Roadmap at an adequate level to ensure proper schedule controls based on the program's approach to execution (incremental, Flexible, Defined deliverables) • Dependencies in the IMS represent the sequence of activities needed to complete the product. • Discrete IMS tasks represent work scope, not agile cadence "time box events" that occur on a regular cycle (e.g. sprints, iterations, release cycles)
Budget	Ensure the entire scope of work is included in a budget baseline and	• Budget Logs (CBB) • CAPs	• Product Backlog	Product Backlog:

IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
	that adequate management reserve exists	• BOEs		• Capabilities include a size estimate* based on assessment of technical size and complexity. The size estimate should be relatable to the budget value of the corresponding control account. • Mapping of Epics/Capabilities/Features in the Backlog to control accounts in the EVMS must exist.
Resources	Ensure the organization structure is appropriate for the program requirements and the staffing plan is credible. Ensure the program has the appropriate facilities, tools and other infrastructure in place	• CAPs by EOC • Org Chart/OBS • RAM • Roles & Responsibilities (RACI) • Staffing Plan	• Agile teams defined • Infrastructure for agile development defined (tools, environments, configurations, etc.) including the Agile management tool	• The program can demonstrate that the organization has the skills necessary to execute the program using agile methods or has a plan for obtaining them. • The program provides an overview of the Agile team collaboration approach (e.g.co-location, facility/communication resources that support agile method efficiencies). • Environments are established to support agile continuous integration and test, if applicable. If not already established, the program can demonstrate it has a plan for establishment. • The OBS is structured to support the way the program intends to manage the work and supports the WBS / Control Account breakout (e.g., Epics/Capabilities map to Control Accounts).
Quality	Ensure the program has a clear acceptance strategy for customer “sell off” defined. Ensure schedule status is recorded accurately and schedule tasks have clear exit/acceptance criteria	• Quality Management Plan • Quality Assurance Plan • Quality Metrics	• Product Backlog	Product Backlog: • Epics/Capabilities have documented acceptance criteria ** based on intended functionality. • All work is documented in the backlog
Risk	Ensure the program has established a Risk & Opportunity board conducted in accordance with the Risk & Opportunity	• ROM Plan • Risk and Ops Register • Risk Mitigation Plans	• Backlog	Backlog identifies significant risks and risk mitigation tasks as appropriate

IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
	Management (ROM) Plan			
Project Integration	Ensure the program has implemented effective management processes and business rhythms, including PPM/EVM. Ensure the program approach, plans and processes are sufficient to meet program requirements	• EVMS documentation • Program Procedures for baseline planning and baseline control • CBB Log • Technical execution documents and processes: Examples (PMP, SW Dev. Plan, SEMP)	• Agile Framework • Backlog to IMS/EVMS mapping • Roadmap	• Agile Framework: Appropriate to the type of program and deliverables desired, that indicates a well thought out plan. ○ Framework includes Agile business rhythms, cadences etc. ○ Method for estimating “relative sizing” of work (e.g., hours, points) has been defined. ○ defines development process (iterative requirements development approach) ○ supports the type of scope under development (H/W, S/W) ○ describes how the process integrates with other management processes (R&O, PPM/EVM, TPMs) ○ If scaling (e.g., Scrum@Scale or Nexus or SAFe,) Key roles and organization structure (e.g., Agile Release Trains) defined • EVM documentation includes instructions and constructs related to traceability from the product backlog to the IMS & EVMS (schedule ID, WBS ID) and how lower-level status information in the agile tool (e.g., stories or features) translates into progress in the IMS and work packages (QBD). • Roadmap informs Rolling Wave Process & Change management
Customer Relations	Ensure the programs priorities and aligned with the customers priorities	• Communication Plan • Joint Management Program and Business Management Review documentation, including agendas & participants	• Increment or Release Review agenda and participants	Communication plan includes: • Roles and responsibilities for customer and contractor personnel involved in customer alignment. For example, does customer or contractor fulfill the product owner role? • Customer/Contractor approach for developing and maintaining Product Backlog • Customer participation in planning events such as increment planning and sprint planning

IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
		• Org Chart that includes customer roles/mapping • Feedback (surveys, CPARs) • Business Rhythm Calendar • Program Management Chart Decks • Program Action Item Database		• Content, format, analysis method and frequency of Agile measures agreed to with the customer as part of the program business rhythm and customer reviews.

* Size Estimate: Backlog Items include an estimation of the “size” of each item, compared to other items in the backlog to determine relative complexity or time required to allocate to each task. Size Estimates are often not hours or dollars based, but use other methods, like story points or T-Shirt sizing to determine relative sizing.

** Acceptance Criteria: Acceptance Criteria are a set of statements, each with a clear pass/fail result, that specify both functional and non-functional requirements, and are applicable at the Epic/Capability, Feature, and Story Level. Acceptance criteria is predefined to demonstrate scope and requirement (including the definition of done as a checklist) completion.

Appendix F – Request for Proposal (RFP) Content

This Appendix addresses Request for Proposal (RFP) content to support an iterative, adaptive, and incremental software development approach that "may" include Agile development but is open to other development approaches. Section 4 of SEI Carnegie Mellon's "RFP Patterns and Techniques for Successful Agile Contracting" dated November 2016 includes some good information on specific considerations for Agile contracting.

This Appendix suggests proposed RFP language and specifically addresses Section C (Statement of Work) and Section L (Evaluation Factors), with Section C addressing Software Development only. This information will assist organizations in developing RFPs for software development programs. This appendix will evolve over time. Future modifications may include sections for Systems Engineering and Test as well as a proposed Contract Data Requirements List (CDRL) relative to and Agile-like contracting process.

Proposed Language

Statement of Work (Section C)

1.0 Scope

This Statement of Work (SOW) addresses the [Design, Development, Deployment, Operations and Maintenance] for [Program Name] Program. Since all requirements will be evolving throughout the development process, the effort needs to support and maintain an iterative, adaptive, and incremental delivery of the software aspects of the system. It also includes, as required, modifying the software system or component after delivery to correct faults; improving performance or other attributes; adapting to a changed environment or maintenance activities focused on anticipated problems; and performing preventative maintenance to support a continuously operating and reliable, stable, and secure application.

Contractors shall form a cohesive team to include the Government and other contractors to foster transparency and information sharing for successful task execution.

3.0 Requirements

3.1 Software Development

The Contractor shall update, execute, and maintain a Software Development process utilizing best practices to perform software requirements analysis, design, implementation, integration, and testing. The contractor's software development process shall support a collaborative environment for implementing the software aspects of the system. Software deliveries shall be iterative, adaptive, and incremental, allowing for the adaption of the emerging implementation of the system for [Program Name] Program. The Contractor's process shall provide the ability to identify, contain, and remove defects as early as possible in the software development process by providing near real-time access to its Software Development Environment (SDE), development documentation, and any other relevant data. Where practicable, automation shall be utilized to gain development efficiencies in the software development process. The contractor's SW development process, procedures and tools shall be documented in a Software Development Plan (SDP) (DI-IPSC-81427B).

3.1.1 Software Deployment

The Contractor shall use the written procedures, standards, and methodology documented in the SDP, for software design practices to ensure the quality and maintainability of all systems. The Contractor shall obtain Government approval of proposed software implementation as part of the incremental planning of the software deployment. The Contractor shall define and deliver the approved software for each release and shall report functionality completed, software deficiencies, and update the definition of remaining work to be planned for the next planning horizon. Software documentation, including design and operations documentation, shall be updated according to the processes described in the SDP.

Evaluation Factors (Section L) – Proposed

Element 1: Software Development Approach

The Offeror shall describe its software development approach and illustrate its intended method for accomplishing the software development requirements defined in Section C. The Offeror shall specifically demonstrate its software development capabilities and resources that will be used to support the development and testing efforts necessary for the development of [PROGRAM] capabilities and interfaces. Specifically, the Offeror shall:

a. Cite the development technique(s) being employed and describe your approach.
b. Describe your approach for iterative planning.
c. Describe how the product will be demonstrated iteratively to the customer and key stakeholders.
d. Describe your process for Open Architecture (OA), Commonality of Hardware, Software/Firmware and Interfaces, Cybersecurity, and prospective Critical Program Information (CPI) with current protection rationale.
e. Describe your Configuration Management process.
f. Describe your approach to artifact delivery; when documents such as the SRS, SDD, Software Test Plan and System Integration Plan will be available.
g. Describe how the software development effort will be synchronized and coordinated with systems engineering activities and reviews.
h. List and describe the software metrics to be used.
i. Describe how software development activities will be coordinated with the Integration and Test (I&T) team, and how it will be assured that the I&T team can keep up with testing all the software releases.

Offerors shall submit an SDP rationale which describes why their specific approach is appropriate for the system to be procured, developed, or maintained and how their proposed processes are equivalent to those articulated by CMMI® capability [level 3]. The SDP rationale is subject to the technical proposal page limitation of the solicitation and shall not exceed [5 pages].

The Offeror shall describe its approach to providing the Government early insight into the development process by providing access to its Software Development Environment, development documentation, and any other relevant data throughout the development process. The Offeror shall describe its reuse philosophy and its approach to minimizing inter-component dependency. The Offeror shall describe why its software development approach is appropriate for [Program Name].

The Offeror shall provide a plan for long term software sustainment and maintenance and the reduction of software life-cycle maintenance costs. The Offeror shall provide historical metrics as evidence of software reliability improvements in terms of build stability prior to delivery on previous projects of similar scope.

The Offeror shall submit a description of previous relevant experience, within the past [36 months] in developing software of the similar size and complexity as that required under the statement of work. As a part of this description, the Offeror shall describe the extent to which personnel who contributed to these previous efforts will be supporting any resultant contract.

The Offeror shall describe any previous relevant Capability Maturity Model Integration (CMMI)® or equivalent model-based process maturity appraisals performed within the past [36 months]. As a part of this description, Offerors shall identify the organizational entity and location where the appraisal was performed, the type of evaluation, the organization performing the evaluation, and the level earned. This description shall not exceed two (2) pages.

Appendix G – Using Agile Metrics to Support Analysis and Forecasting

Agile metrics can be very powerful when used to supplement traditional communication channels between contractor and customer. Within industry, there are a myriad of metrics available to contractors for implementation and incorporation into their management toolkits.

The challenging part can be scaling down metrics used by a program to a small subset that are most beneficial to the specific circumstances and complexities of that program. The use of too many metrics can create a situation of "paralysis through analysis", where too many data points potentially provide too many conflicting points of view and become burdensome to maintain accurately and in a timely manner.

A suggestion for determining the most appropriate metrics is to view the metrics through higher level categories, e.g., quality metrics, velocity metrics, etc. and to select the most pertinent one or two metrics from each category for your program. It is also important that metrics be as direct and easy to understand as possible. Once the correct mix of metrics have been selected, the next step to aiding communication is to allow for easy access to the data. This may occur through providing customers direct access into Agile Management tools (VersionOne, Rally, Jira, etc.), reoccurring briefings, or a shared portal where Agile Metrics are maintained, e.g., dashboard setting.

In summary, the keys to using metrics to aid communication are:

1. Select a small subset of pertinent metrics covering categories most important to the customer.
2. Set up a clear path for the customer to view and utilize the metrics.

A core tenant of Agile is "transparency" and the use of agile metrics, whether reflecting a favorable or unfavorable message, is important to developing a trusting relationship between contractor and customer.

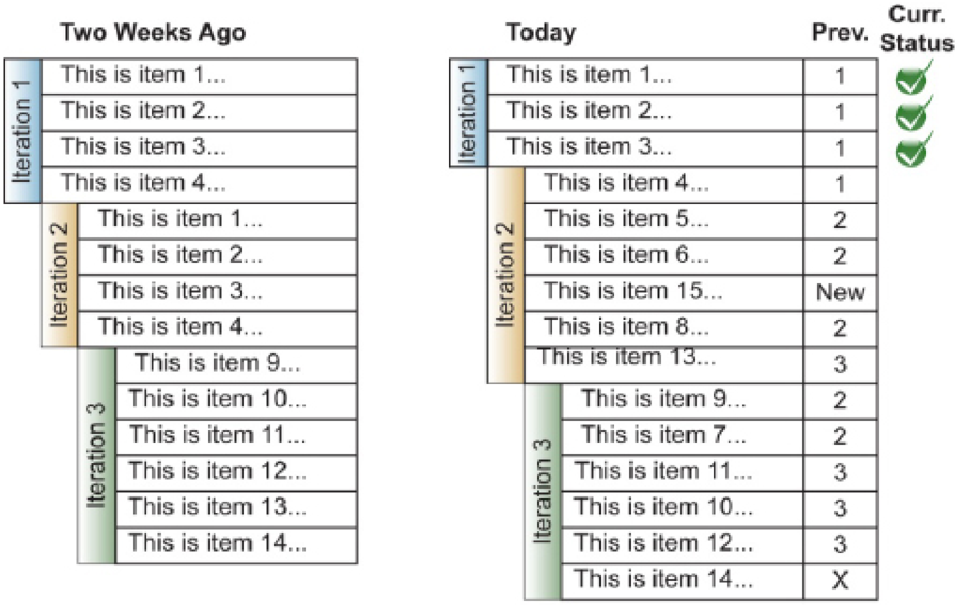
When implemented correctly, the use of agile metrics should provide management and the customer a real time view into near term performance, potential issues and/or opportunities. The goal of these metrics is to ensure that the tasks planned in the current sprint or release remain on track from a cost, schedule, and quality perspective. Over time, the maturity or optimization of agile implementations can be viewed through cost, quality and productivity improvements. They also allow insight into return on investment (ROI) for customers and trends for contractors to make course corrections to their optimization efforts.

The following section examines several high-level categories of metrics and methods for exploiting them.

1. Agile Metrics Usage in Determining Schedule Risk

There are numerous metrics that can be used to convey schedule risk. These include Iteration status charts, burn-down (or burn-up) charts or progress reports.

Iteration (or Sprint) status charts[14] are a simple way to communicate changes from one iteration to the next. They allow stakeholders to see which tasks (stories) have been completed, deleted, added, or moved from iteration to iteration. Continual changes to the iteration status chart from one reporting period to another could indicate volatility and therefore, may indicate schedule risk. They also could simply represent changes in the iteration due to business value decisions. Either way, the change could be identified and analyzed to determine if it represents risk to the program. These findings could be further used to document: the changes, the nature of the change (business values or determined by additional factors), and impact of change (e.g., schedule delays, additional risk, technical debt, etc.)



	Two Weeks Ago
Iteration 1	This is item 1...
	This is item 2...
	This is item 3...
	This is item 4...
Iteration 2	This is item 1...
	This is item 2...
	This is item 3...
	This is item 4...
Iteration 3	This is item 9...
	This is item 10...
	This is item 11...
	This is item 12...
	This is item 13...
	This is item 14...

	Today	Prev.	Curr. Status
Iteration 1	This is item 1...	1	✓
	This is item 2...	1	✓
	This is item 3...	1	✓
Iteration 2	This is item 4...	1	
	This is item 5...	2	
	This is item 6...	2	
	This is item 15...	New	
	This is item 8...	2	
	This is item 13...	3	
Iteration 3	This is item 9...	2	
	This is item 7...	2	
	This is item 11...	3	
	This is item 10...	3	
	This is item 12...	3	
	This is item 14...	X	

Figure G-1: Iteration Status Charts

Burn-down and burn-up charts are simple line charts that plot the work planned versus the work completed. These can be used at the Portfolio/Epic/Capabiilty, Program/Feature, or Team/Iteration level. A burn-down chart is a single line that displays how much work is remaining for the epic/capability, feature, or iteration. A burn-up chart is represented with two lines and displays how much work has been completed against that which was planned. As you can see in the charts below, the end result is the same, but the burn-up chart contains more detail. When a burn-down chart flat lines, there is no additional information provided and it is impossible to tell from the chart what is causing the lack of progress. Using the burn-up chart, you can see that work was added during that period, progress does not flat line and the team was still able to complete all the work.

[14] Nee, N. Y. (2010). Metrics for agile projects: finding the right tools for the job. Paper presented at PMI® Global Congress 2010—North America, Washington, DC. Newtown Square, PA: Project Management Institute.

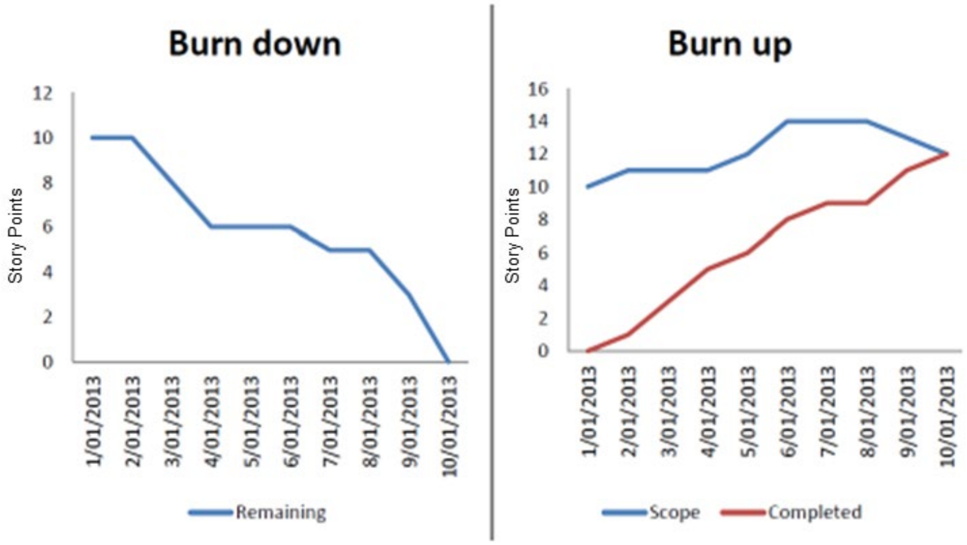


Figure G-2: Burn down and up charts

Progress Reports[15] can be used to provide a quick view of the status of all epics/capabilities and enablers in a portfolio or all features and enablers in a release. For Epics/Capabilities, the report might look like this:

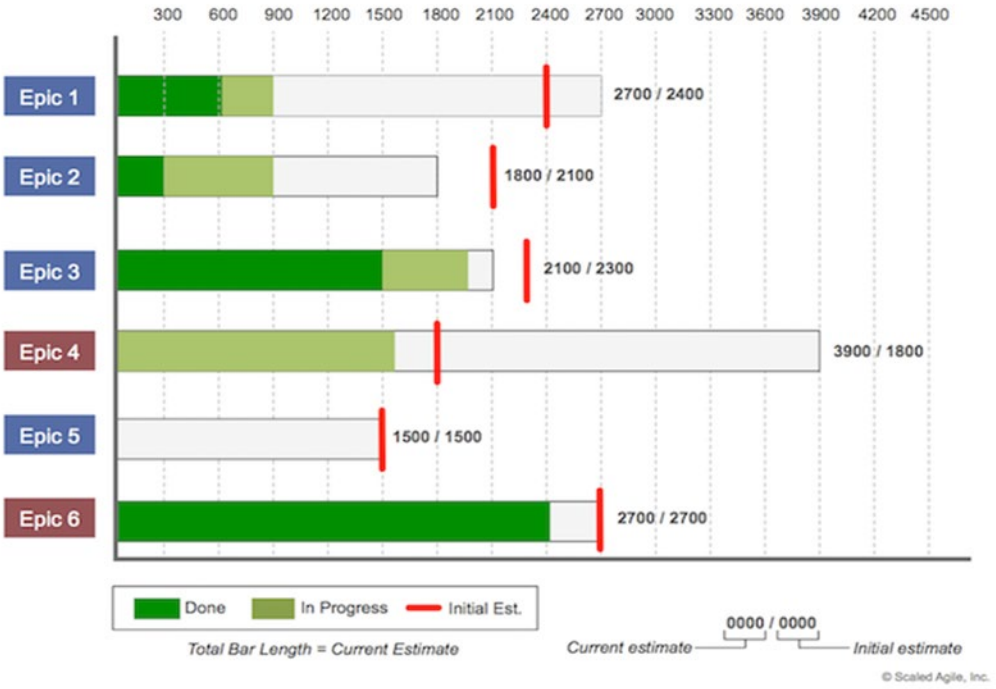


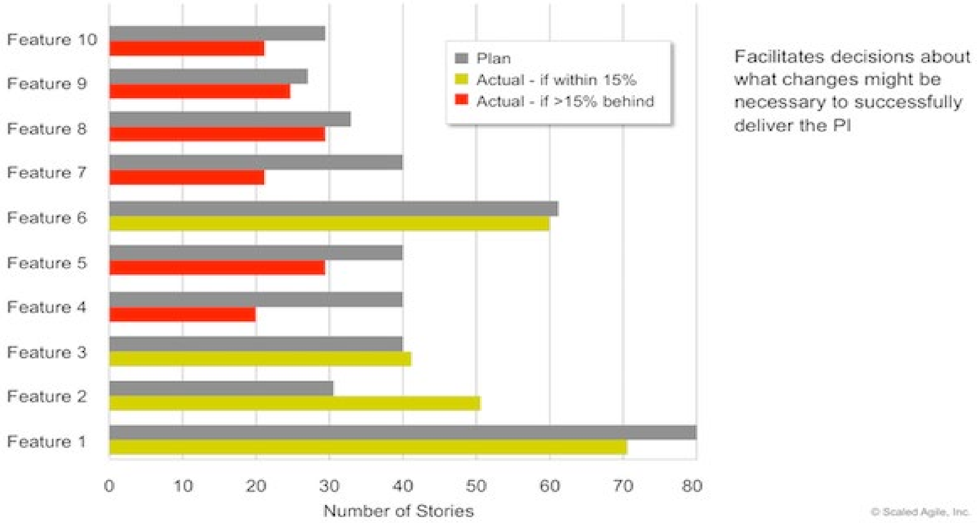
Figure G-3: Epic/Capability Progress Report

Epic/Capability names are indicated along the Y axis, blue for program planned and red for enabler epics/capabilities, while story points are indicated along the X axis. The bar length indicates the total number of story points for that epic/capability with dark green indicating

[15] Scaled Agile Framework (SAFe)– Epic Progress Measure. Accessed Oct 10, 2018, from: https://www.scaledagileframework.com/metrics/#PF4

completed and light green indicating "in progress". The red vertical line shows the initial epic/capability estimates with the numbers representing current estimate versus initial estimate. From this report, it is easy to see the progress on each epic/capability and, where there is growth in story points (e.g., Epic/Capability 1 and Epic/Capability 4). This information can be used to indicate progress and determine if all epics/capabilities will complete within the allotted schedule.

For Features, the report might look like this:



Facilitates decisions about what changes might be necessary to successfully deliver the PI

© Scaled Agile, Inc.

Figure G-4: Feature Progress Report

Feature names are indicated along the Y axis and the bars represent planned stories versus actual stories complete. Green represents that the feature is on track and red represents that it is behind schedule. This information can be used to indicate progress and determine if all features will be completed within the allotted schedule.

While the metrics outlined in the paragraphs above are valuable in determining progress and identifying schedule risks, they are typically collected on a weekly or monthly basis. This, however, might not be frequently enough to keep the program on track. Daily stand-up meetings are a reliable source of determining temporal risk within a program on a day-to-day basis. Each day, team members identify issues, risks, or roadblocks to complete the work planned in a sprint. These problems can then be brought to program management's attention and mitigated real time. Daily stand-up meetings can also be used to refine plans or even swap tasks between team members to create better workflow and speed execution.

2. Agile Metrics Usage in Determining Structural Risk

Several different metrics can be helpful in conveying structural (or technical / financial) risks, depending on the nature of the program and the nature of the technical challenge. It is recommended to consider several different metrics, and then choose the ones which help provide the best insight to the program. Additionally, the metrics chosen should be re-evaluated regularly to help ensure that they continue to provide the most effective and valuable insight. Some of the common metrics include:

3. Technical and Process Metrics

3.1 Technical Debt

Technical Debt is a concept that results from either deferring software defects or deferring development work by implementing short-term solutions (workarounds) which will eventually need to be re-worked into long-term solutions. This can be tracked by number of issues or defects; oftentimes an estimated dollar value is placed on the future work allowing technical debt to be tracked in terms of cost. However, it is measured, larger amounts of technical debt often correspond to structural program risks such as unexpected re-work, late-stage defect identification, and more difficulty in implementing new functionality. Technical debt often requires teams to plan for re-engineering and product enhancement as future backlog items, which may require deferment of other more user-requested functionality until the technical debt is overcome.

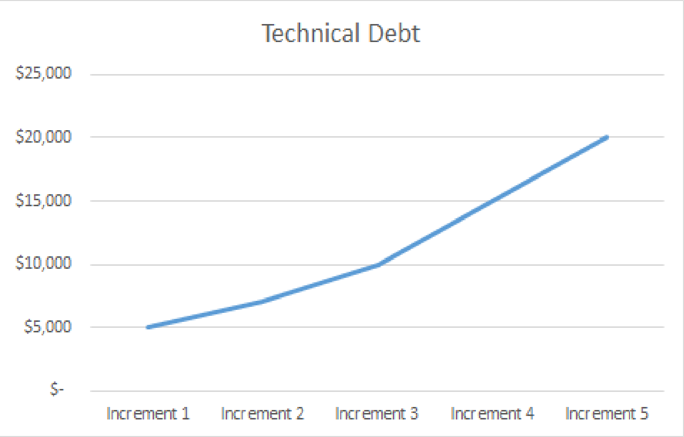


Figure G-5: Technical Debt

3.2 Test Coverage

Test coverage measures how much of the software code is exercised by test procedures during testing events. This is different than having full test coverage (traceability) for the system requirements, and often requires some form of specialized tools or instrumentation of the code to measure. Identifying how much code does not have coverage can be used to identify areas in which defects may be found late in the development process requiring unexpected re-work.

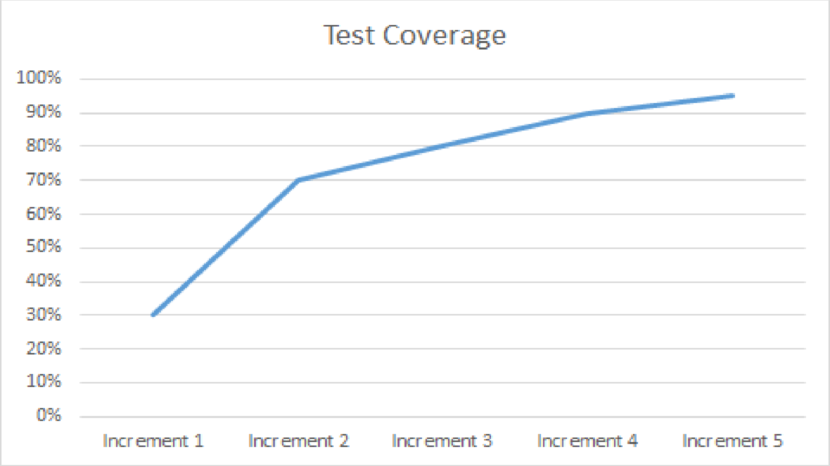


Figure G-6: Test Coverage

3.3 Code Churn

Code churn measures how often parts of the software code have needed to be re-worked by the team. This is often due to the initial implementation not meeting requirements, not performing as expected, having defects needing to be fixed, or not integrating with a larger system as expected. Identifying teams or parts of the code which have high amounts of churn is useful in identifying parts of the system which are more technically complex and may be more likely to result in issues being identified late in the development process.

3.4 Test Case Pass Rate

Test Case Pass Rate measures the outcome of test cases as they are executed as a part of each release. A pass rate which stays low could indicate challenges in progressing with technical development and a likely risk to total cost and schedule. Sudden drops in the pass rate can also be a leading indicator that the technical complexity has increased and there is a risk that unexpected problems or defects could be found late in development.

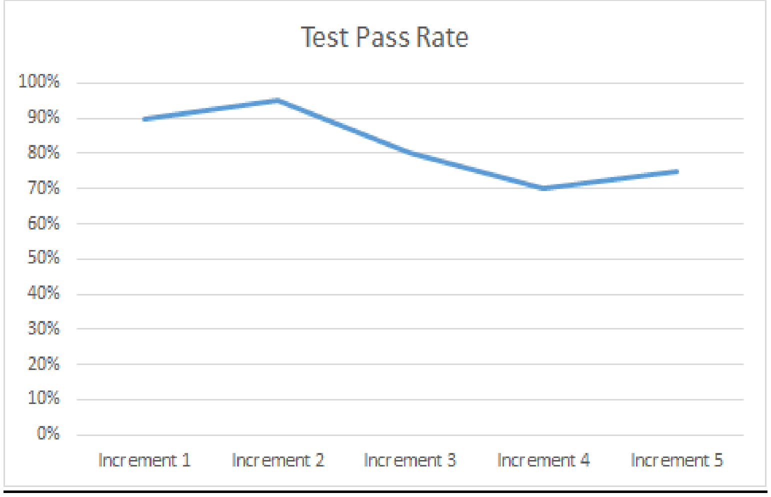


Figure G-7: Test Pass Rate

4. Estimate Accuracy (Variance)

Story points are usually only re-estimated when the team discovers that there is something significant in the size of effort (either bigger or smaller) that they didn't realize before. Having significant growth in story points across releases could be indicative of the team not fully understanding the work, and the risk that future work could be more complicated than planned.

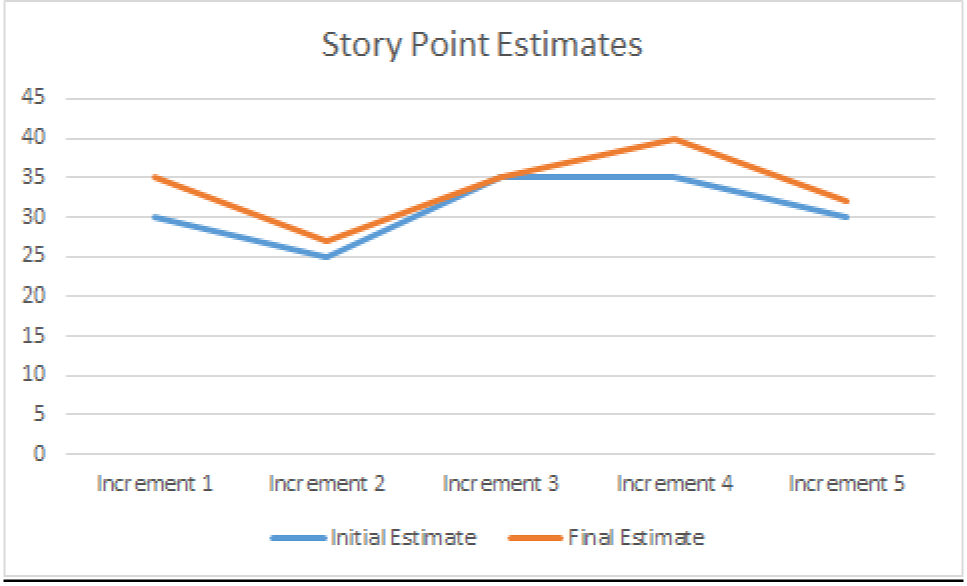


Figure G-8: Story Point Estimates

Appendix H – Agile/EV Guide Contributors

This Guide was compiled by the NDIA Integrated Program Management Division (IPMD) Agile/Earned Value Working Group. The NDIA IPMD thanks the authors and reviewers from across industry and Government who contributed to the generation and improvement of this publication. Their diverse perspectives, expertise, and insight defined proven practices of Agile on Earned Value managed programs.

Appendix I – Unique Acronyms Used in this Guide

The abbreviations and acronyms listed below are unique to this Guide and not found in other NDIA IPMD Guides. Please refer to the NDIA Master Definitions List linked below for common acronyms used across the IPMD industry guides.

NDIA Master Definitions List for IPMD Guides

ACO	Administrative Contracting Officer
AIS	Automated Information System
AKA	Also Known As
CCB	Configuration Control Board
CFA	Cognizant Federal Agency
COR	Contracting Officer Representative
CSCI	Computer Software Configuration Item
DoDI	Department of Defense Instruction
DRB	Defect Review Board
ERB	Engineering Review Board
GUI	Graphical User Interface
HW	Hardware
PBA	Performance Based Acquisition
PC	Percent Complete
PCO	Procurement Contracting Officer
PWS	Performance Work Statement
QBD	Quantifiable Backup Data
SAFe®	Scaled Agile Framework®
SP	Story Points
SW	Software