



National Defense Industrial Association
Integrated Program Management Division

An Industry Practice Guide for Agile on Earned Value Management Programs

March 26, 2018
Version 1.2

1

11/26/2018

The NDIA's *Industry Practice Guide for Agile on EVM Programs* provides practices drawn from lessons learned by multiple aerospace and defense firms in their implementation of using Agile methods to design and develop software in conjunction with using earned value management to organize, plan & budget, analyze, and control Product Backlog and Baseline changes .

This Guide was originally released in 2016. Version 1.1 was released in 2017, and Version 1.2 was released in March of 2018. Version 1.2 added a new section on Contracting for Agile and EVM and Agile IBR Considerations.

This course was developed to present the information contained within the Guide.

The Guide assumes the reader has a basic understanding of Earned Value and Agile development techniques, the training follows suit.

Different organizations use different terminology for similar things. For example, some use Epics for large system capabilities, while other organizations use the term Capability. Some organizations use Sprint and Iteration interchangeably, while others use Sprints only to refer to Scrum time-boxes, and Iteration as a more generic term for any time-box (i.e. all Sprints are Iterations, but not all Iterations are Sprints). This training does not try to recommend any specific terminology, and in general uses Epic/Capability and Sprint/Iteration interchangeably. Instructors should use the terminology that best suits the needs of the attendees.

General Instructions for Teacher:

Describe key points you would emphasize when presenting, especially if those words aren't on the slide

For busy slides and those with graphics/tables, describe the order in which you would present the content

Consider notes regarding anticipatable/common student questions

Consider notes on dealing with "exceptions", i.e., cases where the ideal isn't possible, such as when you can't perform just-in-time release planning prior to performing rolling wave.

Course Content & Learning Objective

NDIA

Chapters 1 thru 6 of the guide present the recommended approach for:

1. Establishing a product planning cycle that is driven by prioritization of the business cycle defined by the customer
2. Developing of the program WBS, IMP and IMS on an Agile development program
3. Planning program scope and assess progress within an Agile development framework
4. Analyzing program performance using Agile and EVM metrics
5. Implementing changes in a controlled manner
6. Contracting / Acquisition Considerations for Agile and EVM

Appendix A thru E provide:

- A. Agile/EVM Data Dictionary
- B. Examples of EVM Agile progress report charts
- C. List of Agile and EVM reference materials
- D. Details on the Product Roadmap, Release Planning, and Rolling Wave Planning
- E. IBR Considerations

2

11/26/2018

To summarize chapter 1 thru chapter 6

Planning is product based and done through iterative and incremental decomposition of Capabilities/Epics and subsequently Features into smaller components of product functionality.

The Guide provides the recommended approach for the WBS thru the CA and WP/PP level, the IMP and the IMS to be used on an Agile development program. It also discusses program performance measurement in an Agile iterative development framework long with the rules and best practices to maintain control of the PMB when implementing baseline changes. Chapter 6 begins with a definition of the agile process, intended for a contracting / acquisition professional. The intent of considerations discussed is identify which elements a contract could be approached for a contract requiring EVM to measure progress when Agile is the preferred method to deliver products in an iterative manner.

Appendix A is an EVM Agile Data Dictionary, with both Agile and EVM terms and definitions.

Appendix B provides examples of EVM Agile progress report charts.

Appendix C lists reference material pertaining to Agile Development and EVM

Appendix D describes in detail the method of building a product roadmap and conducting Release Planning and Rolling Wave Planning Products

Appendix E provides 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.

2

AGILE PROGRAM PLANNING

3

11/26/2018

Agile Planning Levels Related to EVM Processes

	Planning Level	Planning Frequency	Planning Horizon	Planning Precision	Planning Artifact	EVM Processes
Increasing Precision	Product Planning	Project startup; updates throughout the project	Project Duration	Capabilities Releases	Product Backlog; Prod Roadmap; Minimal Viable Product (MVP)	IMP planning of Epics/ Capabilities to Releases (Cadency and Capability).
	Release Planning	Each Cadence Release	Cadence Release	Feature /Stories	Product Backlog Updates Release Plan	IMS planning of Features to Work & Planning Packages. Networking them to Capabilities and Releases.
	Sprint Planning	Each sprint	Weeks	Stories/Tasks	Sprint Backlog	Defining measure of effort and duration for Work and Planning Packages based on Release Sprint Story alignment to Features.
	Daily Planning	Daily	Day	Tasks	Updated Sprint Backlog	Update story status in order to determine EV for each Work Package

4

11/26/2018

Teaching Notes:

Convey the iterative, recursive nature of Agile planning. There are multiple levels of planning that refine the understanding/definition of work to be done over the life of the project.

Focus is on the target (meeting business objectives), not the plan. Changing plans to adapt to changing customer needs or other program circumstances is normal and expected.

Product Planning:

performed at the beginning of the program;

defines all contract scope at the Capability level in the product backlog;

Creates the product roadmap by time-phasing the product-backlog in accordance with contract milestones and deliverables

Provides the technical scope definition of the initial performance measurement baseline

Repeated as needed throughout the life of the program based on program progress and customer direction

Cadence Release Planning:

Defines features for the upcoming rolling wave and maps those features to specific work packages to establish the updated earned value baseline

Features are decomposed from their parent Capabilities

May also be referred to as Increment Planning

There are two types of Releases:

- 1) Cadence Release – which is a time-based release and occurs on a regular schedule, typically quarterly, and is released either internally for baseline management or externally to the client/production environment. Cadence Releases most closely align to EVM Rolling Wave Planning and may result Schedule Variance if planned scope in the release are delayed to a future release.
- 2) Capability Release – is a scope based release and is not held to a regular delivery schedule – the release will be issued when it is ready, and therefore will not likely show a schedule variance, but would likely show a cost variance if it is late.

Sprint and Daily Planning

From an EVM perspective, Provides updated outlooks for in-progress features and work packages.

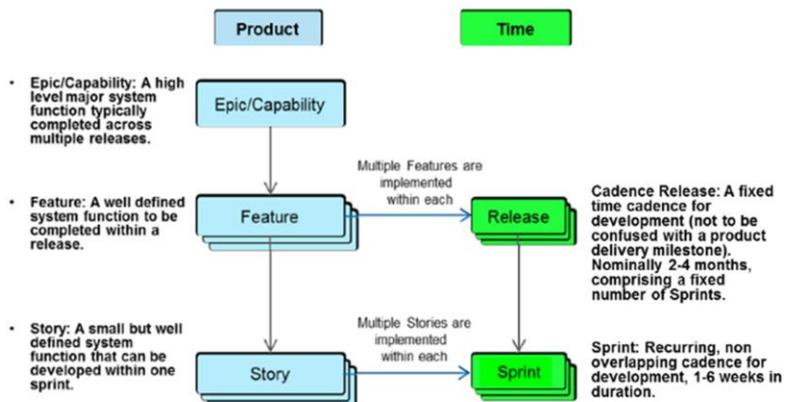
Sprint Planning includes breaking Features down into User Stories, assigning Story Points to the Stories, and Prioritizing the Stories in the Sprint Backlog

Key Points:

- Agile planning starts at the beginning of the project with a definition of the scope of the entire project at a high level.
- The scope definition is refined throughout the life of the project through a series of regular Release Planning events.
- The Feature level scope definition coming out of Release Planning is aligned to/reflected in the IMS as part of Rolling Wave planning.
- Release Planning is also referred to as Increment Planning in the Scaled Agile Framework (SAFe).

Agile EVM Terminology

NDIA



5

11/26/2018

Teaching Notes:

- Illustrates the two separate hierarchies used in Agile, for Product, applicable to WBS and measuring performance, and for Time, the cadence for planning and work execution
- Separate Product and Time hierarchies allow work to be planned by periodically assigning appropriately-sized products into selected Releases or Sprints.

Epic/Capability – customer required ability of the system that provides value

Feature – Part of an Epic which can be completed within an incremental release

Story – Part of a feature which can be completed within one sprint (also referred to as iteration)

Key Points:

- Progress is measured by completion of Product, not passage of Time
- A Feature defines scope and is baselined; Stories are lower level work items created by the team to implement the scope defined by the Feature

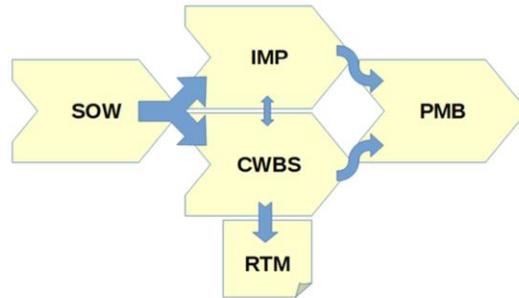
3

**AGILE EVM PERFORMANCE
MEASUREMENT BASELINE**

Traditional EVM Decomposition

- Traditionally, a statement of work (SOW) is broken down by a team into manageable chunks through WBS and OBS decomposition into a product-oriented view of a program.
- Engineering manages the requirements traceability matrix (RTM) and works to ensure traceability is maintained to the original SOW as the solution matures.

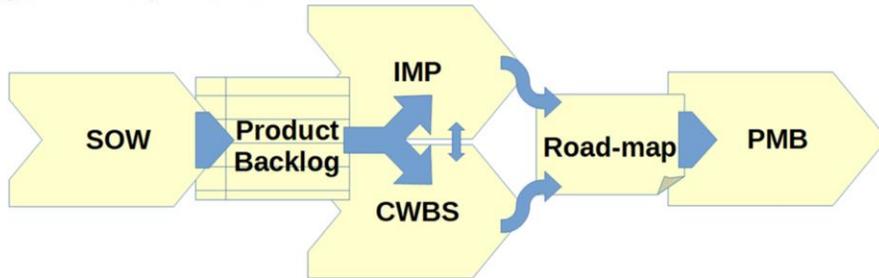
Traditional EVM Decomposition



Teaching notes:

review the traditional components of EVM decomposition

Agile Decomposition



Agile EVM Decomposition

- Agile software and management methods do not introduce significant changes to the typical approach. Instead, it provides the same type of trace to the original SOW by emphasizing the product backlog development.
- Product backlog prioritization determines the overall vision for the program from a business and project standpoint which helps to establish the IMP, CWBS and road-map used for execution.
- The Roadmap then helps with establishing the IMS and PMB.

Teaching notes:

Agile Product Backlog and Roadmap *augment* traditional planning artifacts and tools with useful product-based information.

If the entire contract is for software development and related functions only, it may be possible to remove the IMP as a CDRL and replace it with the Product Increment Roadmap, assuming that the roadmap represents the comprehensive technical approach.

The Product Increment Roadmap is part of the IMP, but, not necessarily the entire IMP, as the entire IMP / IMS represents all scope, even non-development scope, from contract award to contract completion. If you are attempting to elevate or substitute a project IMP with a Product Increment 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.

The Product Backlog is derived from the SOW and is used to construct the IMP and CWBS the Product Roadmap introduces the time-phasing of the work that is used to lay out the IMS/PMB

SOW = Statement of Work (scope provided by customer)

CWBS = Contract Work Breakdown Structure (MIL-STD-881)

IMP = Integrated Master Plan (includes high level program milestones (and ideally the Release Plan)

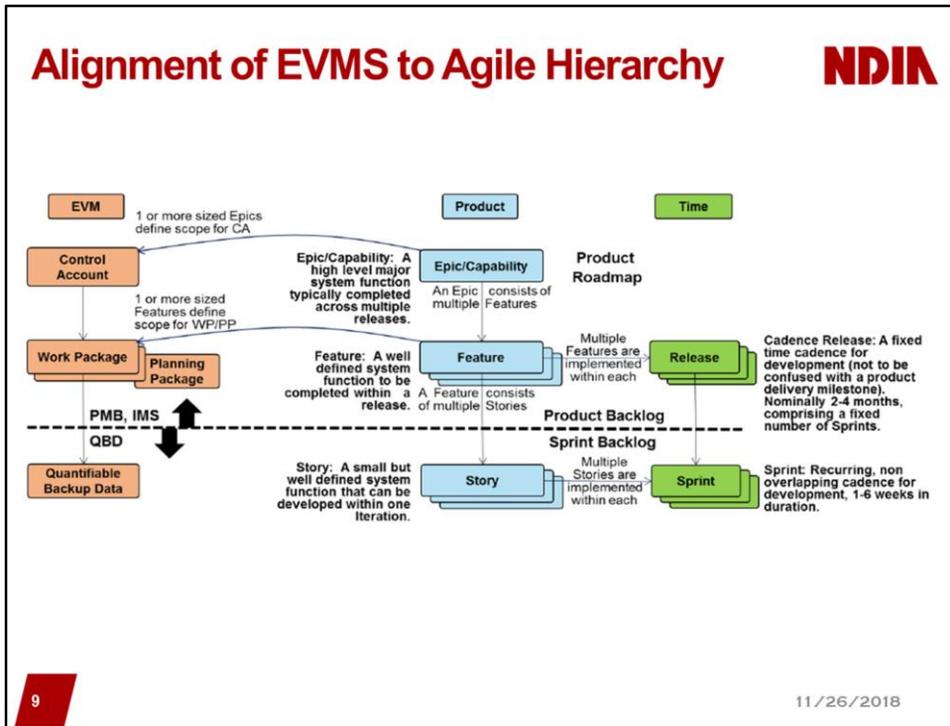
RTM = Requirements Traceability Matrix

IMS = Integrated Master Schedule (at least to the Work Package level – typically the Agile Feature Level)

PMB = Performance Management Baseline (cost estimates at Work Package/Feature Level)

Alignment of EVMS to Agile Hierarchy

NDIA



Teaching notes:

emphasize the relationship between EVM planning levels and Agile Product and Time dimensions.

Control accounts relate to Epics/Capabilities

Work packages relate to Features defined at Release Planning

QBD relates to the stories that implement the Features

Control Accounts and Work Packages comprise the PMB and are managed with baseline control rules

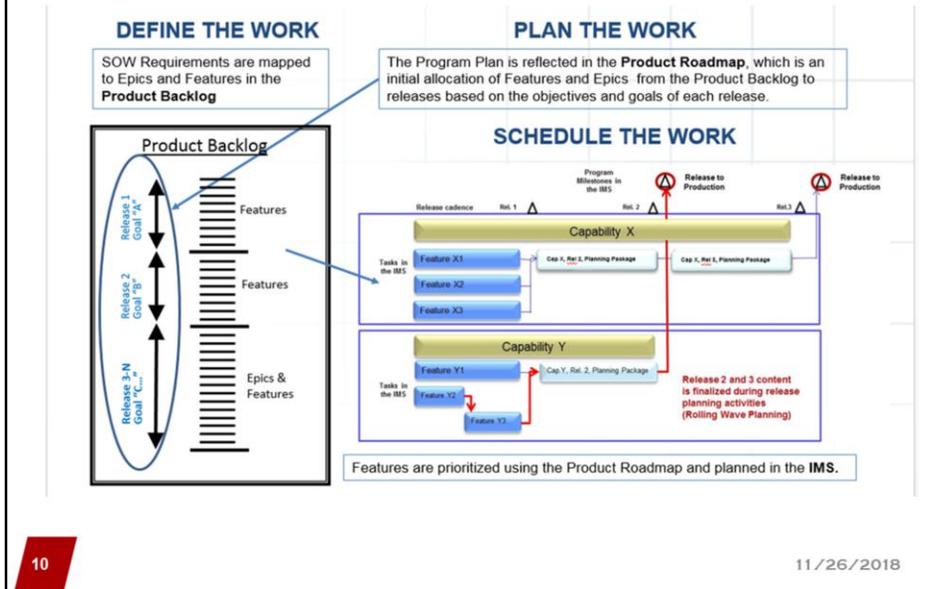
QBD provide backup detail and are managed with backup data rules (not BCRs).

From the NDIA Guide: “What is most important, as illustrated by the black dashed line, is

that there is a clear line established above which earned value is maintained, and below which Agile

methods are preserved that provide Quantifiable Backup Data to support appropriate baseline change management.”

Product Backlog and the IMS



Teaching notes:

This slide brings it all together; shows the work defined in the Product Backlog in a series of releases, mapping to the work schedule as a series of IMS tasks. IMS plan supports critical path analysis.

Presentation order:

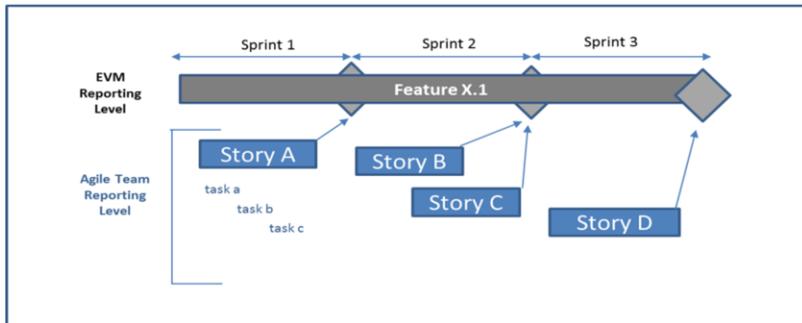
- Define the Work: establish the Epics and Features that derive from the SOW
- Plan the Work: determine the priority of the work in the product roadmap
- Schedule the Work: map Capabilities and Features to your PMB, using the Roadmap and program events/milestones to develop your time-phasing and critical path

4

STRUCTURES FOR PERFORMANCE METRICS

Measuring Earned Value in Agile

NDIA



Earned Value is Measured at the Feature Level

- Agile Teams report status of Tasks and Stories within a Sprint
- Feature % Complete is determined by the completed Story Points as stories are identified as complete

12

11/26/2018

- Need to emphasize that Stories are assigned to Features for performance calculation, but also assigned to Sprints for execution
- Sprints are just time boxes (like a month or a week) and has no explicit scope to claim performance against.
- Performance is always correlated from the Story (task, QBD) to the Feature (Work Package) it is a part of, and Features eventually roll up to Capabilities (Control Accounts)
- QBD details are in the Agile tool, not in the IMS but is traceable back up to the highest level to the CA.
- An example is provided on the next slide

Computing Percent Complete & BCWP

NDIA

Agile Tool ID	Task Description	Story Points	Story Weighting	Story Complete %	NWA %Claim
PMG-245	Story #1 Title	2	8.7%	100%	8.7%
PMG-246	Story #2 Title	5	21.7%	0%	0%
PGM-247	Story #3 Title	8	34.8%	100%	34.8%
PGM-248	Story #4 Title	5	21.7%	0%	0%
PGM-249	Story #5 Title	3	13.0%	0%	0%
	Total Story Points	23	100%		43.5%

Figure 8: Another example of how planned Story Points, or weighted Story Values, may be applied to create QBD to calculate earned value as a PC.

13

11/26/2018

- Feature %C is calculated as story point weight of completed stories divided by total story points of all the features associated with the work package
- Work package BCWP is determined by multiplying Feature %C by work package BAC
- Story point weight is determined by the development team, is based on each story's complexity relative to a team-designated reference story.
- Earlier versions of the guide proposed alternative approaches for claiming earned value based on Story completion status. Currently the recommended approach is to include the full story point weight in the earned value calculation when the story is completed (known as the 0/100 approach).
- Because the story points of in-progress stories are not included in the EV %C calculation, you may see cost and schedule variances due to in-progress work. However, this is normal and true for all projects, Agile and non-Agile, when dealing with in-progress work. On Agile projects, the desire to avoid unfavorable variances provides a good incentive for teams to ensure their stories are small enough to be achievable within the planned iteration, a key Agile principle.
- As the product backlog is refined, stories may be added or removed, impacting the earned value %C calculation. This is expected and a normal part of the development process.
- The guide offers alternative methods for claiming EV based on work completion that align with the different Agile methods. One example is using the completion of Kanban process steps, rather than completed stories, to calculate EV %C. This training focuses on the story-based earned value approach.

Example Forecast Formulas

NDIA "An Industry Practice Guide for Agile on Earned Value Management Programs"
v1.1 March 31, 2017

NDIA

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

$$\begin{aligned} \text{Feature Remaining Effort Hours} \\ = (\text{Total Planned SP} - \text{Total Completed SP}) \times \frac{\text{Total Hours for Sprints to Date}}{\text{Total Completed SP}} \end{aligned}$$

14

11/26/2018

These are the formulas from the Practice Guide.

The Total Stories are the number of Stories at the time of the calculation. This will account for added or deleted stories.

The first is Feature % Complete. Basically it is the Sum of the Completed Story Points divided by the Total Planned Story Points in the Feature.

As in the previous example it can be used to determine BCWP.

The Second Equation to calculate remaining effort can be confusing. It is important to understand that the Total hours for Sprints to Date relate strictly to the Feature you are evaluating, not the entire effort.

The Feature Remaining Effort Hours (essentially the ETC) can be calculated by:

- Step 1: First determining the remaining number of Story Points by subtracting the Sum of Completed Story Points [at the time of the calculation] from the Total Planned Story Points [number of story points in total you have for the Feature at the time the calculation is made]
- Step 2: Next, divide the total number of hours spent on the Feature to date (the ACWP) by the Total Completed Story Point for the Feature
- Finally, multiply the Remaining Story Points (step 1) by the Average number of hours per Story Point (step 2)
- This can also be calculated in dollars if that information is available

Example Velocity Chart

NDIA



15

11/26/2018

This slide shows two views of a sample backlog chart showing the number of story points completed for each sprint. The top chart shows "Release Velocity by Team" and the bottom shows the "Total Release Velocity" at the program level. Velocity (story points completed per sprint) is represented by the bar, and average velocity is represented by the moving average line from sprint to sprint.

Velocity is simply the average amount of work completed over a period of time.

It is most commonly understood to be the Number of Story Points Completed per Sprint, but could also be calculated as the Number of Features in a Release, or number of stories per Sprint. For this example, we are referring to Story Points/Sprint.

The first chart shows the Velocity of 3 Scrum Teams during a release.

- Teams 1 & 2 show the "ideal" increasing velocity trend, while their specific velocities differ from one another.
- Team 3, however, shows a more erratic velocity (demonstrated with the 2-period moving average)

Velocity can be thought of similar to a learning curve. When a team first forms, they are getting acclimated to the technical work, as well as team forming into the Agile process, so Velocity is expected to be lower at first and increase over time; until a point where the team hits a sustainable velocity and becomes predictable and can level off or continue to improve with each cycle.

The second chart shows the Total Release Velocity – with all 3 teams velocity added together. While you shouldn't compare one team's Velocity to another (unless the story points are normalized across teams), it is acceptable to add the velocities together to understand the overall capacity of total project team.

Velocity can be useful as a measure of a Scrum Team's CAPACITY to Complete Work in estimating future efforts, which can be used in forecasting (as shown on the next slide).

Since Velocity is based on Story points, it is important to remember that it is a metric that is specific to a Scrum Team and shouldn't be compared across the program, as Story estimation is unique to a Scrum team.

There is a difference in *comparing* of team velocities, versus the *addition* of the velocities. For example, it would be incorrect to assume since team A has a lower velocity than team B, that team A is producing less product. However it is legitimate to use their cumulative velocity of 25 pts/sprint (=10 + 15) to predict future cumulative story points completed. For example, if team A averages 10 pts/sprint, team B averages 15 pts/sprint and team C averages 20 pts/sprint, then the project as a whole averages the sum of those, or 45 pts/sprint. As long as the makeup of the teams do not change, the project should average 45 pts/sprint. Normalization is important when we correlate story points to another work attribute, for example points *per hour* or points *per line of code*; these will differ by team and if these measures are desired, normalization is required.

What questions can the room think to ask from the data on this chart?

Example Release Burn-Down Chart

NDIA

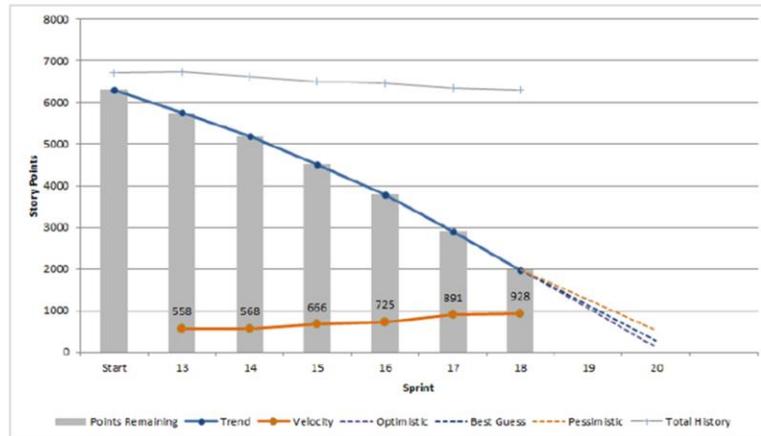


Figure 12: Example of a program level burndown chart across multiple teams, indicating overall status and predicted completion Sprint.

16

11/26/2018

This chart is taken directly from the NDIA Agile Guide. The Start Bar shows the number of Story Points at the Start of the second release (assume end of Release 12), and then shows the remaining story points at the end of each successive sprint (13-18) with projections to sprint 20.

Similar to a Release BurnUP chart, there is also a Release BurnDOWN chart. Both charts show the same data, but from a different view.

Where a BurnUP chart can show “will we complete all of the assigned tasks on time?” The BurnDOWN chart is useful to predict “When are we expected to complete all of the assigned tasks?”

The gray line across the top is the total cumulative story points in the backlog and is showing the change in total story points over time, adjusting for additions and deletions.

The gray bars indicate the number of incomplete story points at the beginning of each sprint, with the Start bar always equaling the current total story points in the release backlog and the remaining bars equaling the height of the previous bar minus the sprint velocity.

The solid orange line is Velocity (as previously discussed) and is represents the capacity of the team to complete work (as recent past performance could indicate future performance (similar to CPI or SPI).

The solid blue line shows the current completion trend.

To determine “when we are expected to complete the remaining backlog, the blue line is extrapolated to the point where it crosses the x-axis (remaining points = 0). In the example shown, there were

- Optimistic (purple dotted line) which looks like it assumes future work to complete at the same rate as presently executing (best of the last n iterations) and should complete in Sprint 20
- Most Likely (blue dotted line), average velocity of the last n iterations, still has a chance to complete in Sprint 20
- Pessimistic (orange dotted line), worst of the last n iterations, may actually push overall completion to a 21st sprint

Work that is not completed as planned within a release is re-prioritized in the backlog and moved to the next release

There are various ways to calculate completion dates, the important thing is that, similar in EVM, it is documented, justifiable, and consistent.

What questions could you ask given the data in this chart?
What concerns should the students have?

5

MANAGING BASELINE CHANGE ON AGILE PROGRAMS

Controlling Baseline Changes

NDIA

- Baseline change rules apply as prescribed by your company's EVM System Description
- May not revise the scope, schedule or budget of a Epic/Capability or a Feature without following your company's documented change control procedures.
- May add, modify or remove Stories during Backlog refinement as needed to implement the feature scope.
 - User stories describe the team's intent of how the exit criteria of the Feature Work Package will be satisfied and are developed and maintained below the level of the EVMS PMB.
 - Configuration control of the stories is maintained within your Agile tool so all changes to a Story will be documented.
- The program does Rolling Wave planning at Cadence Release points to allow for flexibility and discovery. **Rolling wave planning occurs after the Release Planning Event.**
- The Contractor should establish a Freeze Period that supports the flexible nature of Agile development.

18

11/26/2018

- 1) Requirements are understood at a high level, but we also expect some level of change to emerge as Stories/Features are built and new knowledge is gained. It comes down to having the appropriate level of granularity of the Feature exit criteria; detailed enough to have a good foundation of "what", but not so prescriptive as to inhibit beneficial change. *"Stories are not stand-alone requirements, in that they represent a statement of intent rather than a contractually required behavior" (Leffingwell)*
- 1) The program does Rolling Wave planning at Cadence Release points to allow for flexibility and discovery. Rolling wave planning occurs after the Release Planning Event. The current release is detail planned and decomposed into "Feature Work Packages". Budget for future releases remains in Planning Packages.
- 3) The Contractor should establish a Freeze Period that supports the flexible nature of Agile development. Discovery and change are a normal part of Agile development, and change assessments occur frequently
 - at 3 month Cadence 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 such as "current month plus 1" 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 EV.
 - ALSO MENTION HERE HOW GREAT FOR THE CUSTOMER TO BE INVOLVED AT CADENCE RELEASE PLANNING TO INCORPORATE HIS LATEST KNOWLEDGE/NEEDS of THE WAR FIGHTER.
 - Customer involvement/role should be identified in the program kickoff. Will the customer be the Product Owner or a Stakeholder when it comes to prioritizing the features in a release?

Cadence Release Planning/EVM Rolling Wave Planning



- **The Cadence Release Plan defines the set of Features that have been refined from Epics/Capabilities on the Product Roadmap that will be implemented within the Agile Cadence Release/EVM Rolling Wave period.**
 - The CAM uses the output of release planning to implement the EVM Rolling Wave Plan
 - PPs are converted to WPs (Features), Features are decomposed into Stories which are sized to facilitate the measurement of progress
 - The feature(s) are input into the IMS along with predecessor/successor task relationships, and then fed into the EV engine. The CAM validates the Product Backlog and Product Roadmap are consistent and traceable to the EVM data prior to approval
- **Work cannot start on the scope planned at the Release Planning event until it is incorporated and approved in the EVM PMB.**

19

11/26/2018

The objective of Agile Cadence Release Planning/EVM Rolling Wave Planning is to establish and detail plan the functionality to be implemented within the program's next Cadence Release/Rolling Wave period.

In a large program where both Agile and EVM is in practice, Rolling Wave planning that typically occurred on a traditional EVM program. Rolling wave planning on a traditional EVMS program is done every 6 or 12 months or at major milestones can be replaced by a Rolling Wave Planning at Agile Cadence Release events. Cadence Release events are nominally held every two to four months. 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.

Allowing programs to wait until after the planning meeting is held for the next increment of work before finalizing the detailed plan should minimize changes once the work has started.

The Cadence Release Plan defines the set of Features that have been refined from Epics/Capabilities on the Product Roadmap that will be implemented within the Agile Cadence Release/EVM Rolling Wave period.

The CAM uses the output of release planning (updated Product Backlog and Product Roadmap) to implement the EVM Rolling Wave Plan:

Planning packages are converted to work packages (remember the work package contains one or more Features. Features are decomposed into Stories which are sized to facilitate the measurement of progress. The Feature(s) are input into the IMS along with predecessor/successor task relationships, and then fed into the earned value engine. **Then compared to the Product Backlog and Product Roadmap to insure consistency and traceability**

Care must be taken to promptly recognize and capture impacts from the release planning events into the EVMS PMB as needed **before the work starts. This time sensitive flow needs to be addressed in the contractor's EVM System Description to ensure the freeze period for Agile scope does not conflict with the defined baseline change control rules.**

Agile Freeze Period Considerations

NDIA

- To avoid any misinterpretation of the DoD EVMSIG with respect to the freeze period, the freeze period should be adjusted, through formal changes to a company's System Description or other supplementary guidance, to be short enough that it accommodates the Agile planning cycle.
- 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 Cadence Release.
- It is acceptable to decompose planning packages and create work packages inside the current period for work that has not yet started. The detail plan must be approved prior to the start of the work, and such an approach must be compatible with the contractor EVM System Description.

20

11/26/2018

Freeze period considerations: The Contractor should establish a freeze period that supports the flexible nature of Agile development. Discovery and change are a normal part of Agile development, and change assessments occur frequently. A Contractor's freeze period should be defined in a way to support Agile and EV.

Explain how the freeze period relates to Rolling Wave Cycle

A typical Agile business rhythm holds the release planning meeting for the next increment of work at the end of the current increment, just prior to the start of the next increment or Cadence Release cycle. The Cadence Release event is the opportunity to get customer involvement. Rolling wave planning tied to Cadence Release event is not only for the latest information set into the plan but for customer collaboration to have their input on that which is most important to them and the war fighter.

It is acceptable to decompose planning packages and create work packages inside the current period for work that has not yet started. The detail plan must be approved prior to the start of the work, and such an approach must be compatible with the contractor EVM System Description.

First bullet: However, the approval cycle of a change(s) to a contractor's EVM System Description can be up to and longer than one year. The Contractor should document the Agile process used in the interim along with the plan for updating the system description

Second bullet: The customer should be highly integrated into the release planning process, with ample opportunity to provide input on the plan if there are concerns.

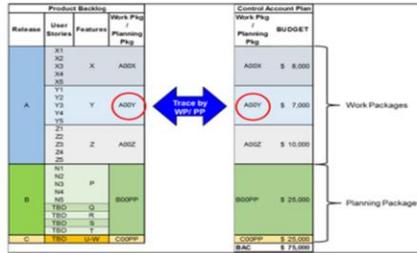
Third bullet: The detail plan must be approved prior to the start of the work, and such an approach must be compatible with the contractor EVM System Description.

Traceability of Baseline Changes across Agile and EVM artifacts



- A Release Roadmap is maintained that documents the prioritized product Backlog. Epics and Features on the Product Backlog are mapped to specific releases as part of the Product planning process.
- The Product Backlog documents the technical scope of each CA. Each CA maps to a subset of backlog items.
- All Items listed on the Product Backlog and Product Roadmap are consistent and traceable to a Work Package or Planning Package in the PMB
- If a baseline change has been made to a WP, the change must be traceable thru the Agile artifacts

- ▶ 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)



The Product Backlog traces to the Control Account Plan

21

11/26/2018

The intent of GL 29 is to reconcile current budgets to prior budgets in terms of changes to the authorized work and internal replanning in the detail needed by management for effective control.

To do this you need to establish a relationship / traceability path between the Agile artifacts and the EVM artifacts.

A Release Roadmap is maintained that documents the prioritized product Backlog. Epics and Features on the Product Backlog are mapped to specific releases as part of the Product planning process

The Product Backlog includes a coding structure that traces to the CAs and to the EV engine.

The intent of GL 29 is to reconcile current budgets to prior budgets in terms of changes to the authorized work and internal replanning in the detail needed by management for effective control.

To do this you need to insure that consistency and traceability can be demonstrated between the Agile artifacts and the EVM artifacts.

Baseline Change Scenarios



	Scenario	PMB	Backlog
1	The Work Package/Feature is 30% complete, but did not complete by the formal delivery date. The delivery date is held as planned and the customer accepts the delivery without the Feature functionality.	Although the customer accepted the delivery without the functionality of this Work Package's Feature, the scope will continue on, showing a schedule variance (SV) until the work is complete.	The unfinished Feature's stories are moved to the next release and planned in a Sprint.
2	The Contracting Officer (CO) issues a contract letter 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 entire unclaimed budget associated with the Epic/Capability is returned to Undistributed Budget (UB) until dispositioned by contract mod (de-scope).	The unfinished Stories, Features, and Epic/Capabilities are removed from the Product Backlog.
3	The exit criteria for Feature 1 Work Package is 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 has 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 to the contract) the budget will come from UB. If this is an unplanned in-scope increase the budget will come from Management Reserve (MR).	The exit criteria for Feature 1 is updated. Stories are created and added to the Product Backlog and mapped to Feature 1.

Baseline change scenarios

Page 23 – 24 of NDIA Guide, Lets walk through them

Scenario 1 – **Review Scenario.** The scope of the WP did not change so no change will be made to the work package. A negative schedule variance result in the earned value data. The stories are moved to the next release in the Product Backlog

In Scenario 2 - This is a contractually directed change. **Explain Scenario, PMB and Backlog on chart.**

1) the WP is closed, BCWS is set = to BCWP. The remaining BCWS is transferred to Undistributed Budget until the Mod is dispositioned/definitized. The subject unfinished Stories, Features, Epic/Capability(s) are removed from the Product Backlog and Product Roadmap.

Scenario 3 – **Explain Scenario, PMB and Backlog.** The Contractor needs to establish the definition of product and the Definition of Done at the Feature and Epic level to allow for the creation, removal, and modification of Stories that are developed to satisfy the Feature and Epic functionality without an impact to budget or scope. Requirements are understood at a high level, but we also expect some level of change to emerge as Stories/Features are built and new knowledge is gained. It comes down to having the appropriate level of granularity of the Feature exit criteria. Detailed enough to have a good foundation of “what”, but not so prescriptive as to inhibit beneficial change

Forecast Change Scenarios



	Scenario	PMB	Backlog
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 move those Stories to Sprint 4, which is beyond the baseline finish date of the Feature.	No change to the Feature Work Package baseline budget or baseline schedule. The in-progress Feature IMS task shows a slip to the baseline finish date. BCWP is only claimed for the Stories actually completed. BCWP compared to BCWS identifies a schedule variance. Reflect changes in IMS Forecast and EAC.	The Product Backlog is updated to move the Stories not completed in the first Sprint into the fourth Sprint.
2	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). <i>The Exit Criteria for the Feature has not changed.</i> The Feature WP is in progress and the subject Story is a QBD.	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 and EAC.	The Story is removed from the Product Backlog.
3	Features mapped to future releases are reprioritized based on discovery/user feedback and mapped to other future releases 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 roadmap should be analyzed for potential bow-wave and 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 the IMS forecast and EAC.	The product Backlog is updated and the Features are mapped to the resulting releases on the release roadmap.

Forecast change scenarios

Page 26 – 27 of NDIA Guide, Lets Walk Through them

Scenario 1 -- **1) Discuss Scenario, PMB and Backlog** In this case it’s simply a matter of the work not being able to be completed in the original time span. This is an unlikely scenario, moving sprint 1 stories to sprint 4. More realistic would be sprint 1 stories push into sprint 2, which pushes other stories in Feature to sprint 3, consequently moving stories in Sprint 4 beyond the baseline finish date..

Scenario 2 – **1) Discuss Scenario** as written on chart, **2) User Stories** are developed and maintained below the level of the EVM PMB. The Work Package/Feature level defines the “definition of done” (scope needed to achieve to complete the WP). The Stories/QBDs describe how the intent of the Feature/WP will be satisfied. **3) Discuss PMB and Backlog - (Give example of how change removal of a QBD results of decreasing performance)**

Scenario 3 - **1) Discuss Scenario, PMB and Backlog** Feature movement like this is not unusual, and should be done is close collaboration with the customer.

CONTRACTING FOR AGILE AND EVM

This section discusses contracting best practices for including Agile and EVM disciplines in government contract solicitations.

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. (A-11 Capital Programming Guide (July 2017)).

Defining the Agile Process

NDIA

- An “Agile” product should not be defined by a prescriptive set of requirements as typically seen in government contracting.
- 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 against the final product.
- The Definition of Done is a key component of defining the Agile product and is critical for both the incremental progress and the final product.

25

11/26/2018

Intent of this definition it to spread awareness and update thinking that each and every change in a requirement is a put or a take within the contract. Yes, requirements will be managed and the authority to implement change at the contract level remains the same, however, each contracts representative on the buying and selling side should be connected with each of the PMOs, product owners and stakeholders to assist in managing change and product business value.

First bullet: Agile product requirements expressed as desired outcomes rather than specific details about how the work is to be performed.

Second bullet: different layers of change management; for example contract change (scope), baseline change (rolling wave), engineering change (QBD management)

Third bullet: Agile Manifesto: customer collaboration over contract negotiation; Agile Principle: Business people and developers must work together daily throughout the project

Fourth bullet: The final product should be thought of as a set of features or capabilities (a truck with towing ability). When choosing to utilize the agile process, the collaboration occurring during each iteration, will focus on refining the individual pieces of each feature (the type of hitch). That flexibility should occur. However, if a change is encountered that will change the make-up or the vision of the final product (no hitch or a car), that change will drive contractual change, such as an ECP.

Fifth bullet: Consider including a definition of done as part of the contract objectives. If not as part of the contract, as part of the exit / completion criteria for reporting EV on work packages / features. Also related to the acceptance criteria of the feature, that is also conveyed in the user stories. This is another way that collaboration and expectations of the produce resonate throughout the agile process execution.

SOO versus SOW – Incorporating Scope NDIA

Factor	SOO	SOW
Government understanding	The government understands the objectives but expects the end state to evolve.	The government has a high level of confidence in the end state.
Change	Change is expected to be a significant factor in achieving the end state.	Change is not anticipated, or if encountered will not be disruptive.
Constraint	This approach provides the offerors trade space and flexibility in developing their proposals. It should probably be used unless the totality of the work effort required is very well understood by the government.	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.

26

11/26/2018

For Agile acquisition, a SOO with stated objectives is recommended. 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.

First table above highlights the differences between a SOO and a SOW.

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 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.

Second table above provides a comparison between a SOO, PWS and SOW.

SOO versus SOW – Incorporating Scope **NDIA**

	SOO	PWS	SOW
Buyer	<ol style="list-style-type: none"> 1. Describes requirement stated in outcomes only. 2. Does not identify a technical solution to the requirement. 3. Saves time in developing the solicitation. 	<ol style="list-style-type: none"> 1. Buyer defines work outcomes and results and 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 requirement to agency mission and/ or objectives 	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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. 	<ol style="list-style-type: none"> 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 	<ol style="list-style-type: none"> 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.

For Agile acquisition, a SOO with stated objectives is recommended. 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 above provides a comparison between a SOO, PWS and SOW.

Agile Solicitation Considerations

NDIA

For each solicitation consider the specific agile goals and do not use these items to constrain the agile process. Not all of the items are appropriate for all types of agile execution. Suggestion for consideration include:

- Definition of Done (DOD)
- Product Owner Responsibilities (Customer Interaction)
- Development Team Responsibilities
- Iterations
- Planning
- Reporting
- Testing
- Fixed Price versus Cost Plus
- Payment Milestones

28

11/26/2018

Section 6.3, starting on page 32 of the guide discusses 9 considerations for incorporation into the contract. The first 5 are new concepts and the buyer and seller should agree on this items during negotiations. The last 4 are not new, but, should be approached in a new way.

Definition of Done (DOD) - (AKA acceptance criteria), Does the produced working software matches the product vision? Recommended to develop this in parallel with negotiations and include as an appendix. The acceptance criteria (Agile) 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 of requirements to verify the software produced matches the product vision. Elements to consider for the Definition of Done include, and are not limited to: scope of tests to be conducted and passed, code reviews, coding standards, and code has been re-factored where necessary. 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 (the person directing the business value), 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 - An iteration (a fixed time box) can be an increment, a release, a capability drop - define for the solicitation that definition. Make a distinction of timing of the purpose of the iteration or a grouping of iterations - whether it is an internal release for developers or a push to production. How can the solicitations be approached in more of an iterative way through the use of 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 and release burndown and velocity. Utilize sprint reviews and Technical Interchange Meetings;

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 vs 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 knows exactly how much the effort will cost, with scope, and schedule firmly established, 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 specific 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.

Clauses and Agency Policy Citations



- This section is a cross reference for a list of potential clauses to be considered for inclusion when contracting for Agile and EVM
- EVMS Requirements References
 - Office of Management and Budget (OMB) Circular A-11 is the primary source of acquisition requirements.
 - DOD, DOE and NASA have agency-specific references
- Performance Based Contracting References
- Contractual Reporting and Data Deliverables
 - Working software is the primary deliverable
 - Allow for iterative delivery of “as-built” CDRLs
 - Provides list of CDRLs impacted by Agile process

29

11/26/2018

Section 6.4, starting on page 34 existing clauses and agency policy citations are referenced. It provides a cross reference for a list of potential clauses to be considered when contracting for Agile and EVM.

Key take away: The notification of EVM on a solicitation or contract does not change with the addition of the Agile methodology. There is not clause or provision for Agile that is being added to the solicitation. Consider minimizing CDRL duplication and overlap, specifically for System Engineering and Design, by finding lightweight ways to sustain your required Plans and approach documents to capture approaches, considerations and nuances associated with your Agile implementation (such as rhythms, ownership, etc.).

Despite any policy references to dollar thresholds, any of the clauses referenced in the guide 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.

EVM References - The Federal Acquisition Regulation (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 A-11. Agencies without supplemental guidance reference FAR Subpart 34.2 and the related FAR solicitation or contract clauses.

Performance Based Contracting- When contracting for an Agile methodology, it is recommended to include provisions for performance based contracting and use of a SOO. 2 references: Seven-Steps to Performance-Based Acquisition (guide/instructions for SOO, PWS and QASP etc.) and DAU Service Acquisition Mall provides tools and templates to create a performance-based service acquisition requirements

Contractual Reporting and Data Deliverables

- Contract reporting is directed by contract clauses and data item requirements.

- In an Agile software development contract, the working software 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” CDRLs or elimination of CDRLs no longer needed.

- In EVM, the Integrated Program Management Report (IPMR) is the primary CDRL. The Agile details 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)
Agile Reporting Metrics
IMP (see Section 3.2)

Contractual Change in an Agile and EVM Environment

NDIA

- Agile processes leverage changing requirements for customer advantage
- Express contractual requirements as outcomes to simplify change management
- Assess the nature of the change
 - Contractual scope change affecting high-level product requirement (SF-30)
 - EVM baseline change
 - Engineering trade-offs that affect neither contract scope nor EVM baseline
- Contracting Authority
 - Contractual authority does not change when utilizing EVM with Agile
 - Ultimate signing authority is between the Buying Contracting Officer and the Selling Contracts Manager
- Program management processes for managing change
 - Agile Ceremonies – may supplement or replace traditional reviews
 - Board Reviews (ERB, CCB, Risk)
 - Program Reviews
 - Contractual Documentation

30

11/26/2018

Agile Principle #2: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Section 6.5, starting on page 36 discusses the different types of change – changes in requirements, EVM baseline change and contractual change. Intent of this section is to continue to spread awareness and update thinking that each and every change in a requirement is a put or a take within the contract.

Bullet 3: 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?

Bullet 4: The section continues with a discussion of the Contracting Authority roles, which has not changed. Note: the product owner does not contracting authority.

Bullet 5: The section end with suggested Program Management Process adaptations for Agile and EVM. How can a PM adapt the existing mechanisms to communicate and manage change?

Key take away: 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 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

Closing Thoughts:

What do you think? Will you include both Agile and EVM on a solicitation? What is the value of doing EVM on Agile projects?

Future Thought Item:

Agile acknowledges that the future product to be delivered requires further definition (within the scope of the contract) and is an excellent mechanism to focus on what was done today and what will be done tomorrow. The Agile methodology alone may not provide comprehensive insight for cost and EAC management at the contract level. The contract identifies all scope to be completed, for all WBS items. EVM provides a mechanism to track how the product is evolving within the scope of the product WBS defined on the contract. In addition, EVM provides insight for the non-product WBS scope defined on the contract. Using Agile and EVM together complement the technical progression, scheduling, and total cost assessment of all scope throughout execution, providing a comprehensive strategic view across the entire WBS of the contract. The purpose of including both an Agile development methodology and EVM on a contract is to drive collaboration and insight 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 (A-11 Capital Programming Guide (July 2017)).

Appendix A

EVM/AGILE DATA DICTIONARY

Appendix A - EVM/Agile Data Dictionary

This appendix is organized into three sections.

1. **EVM Agile Data Dictionary.** This section provides an EVM Agile Data Dictionary and thesaurus of Agile terms and rationale. The intent is to create a common vernacular and a method to harmoniously blend EVM and Agile program management practices.
2. **Agile Data Dictionary.** The section is a combination of Agile terms from Scrum Alliance and Agile Alliance.
3. **EVM Data Dictionary.** This section includes common EV Systems Management (EVMS) terminology consistent with the EIA-748 Standard for EV Systems.

Principles mapped to Agile Concepts



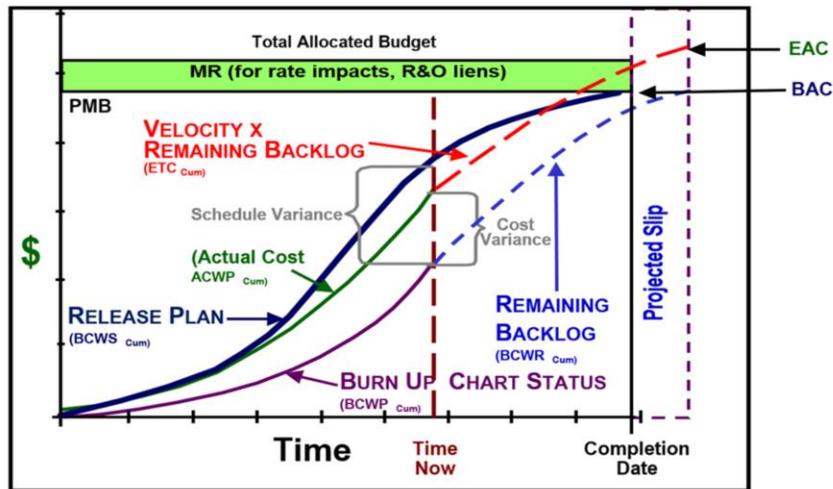
EVM Principle	SW Development Programs (Agile)
Decomposition of work into manageable pieces.	EPIC and Feature Based WBS for SW (Product Backlog)
Assignment of resources against that work.	SW Development Teams (Sprint Teams)
Assigning value to work to be accomplished.	Business Value assigned at Feature level and above; story point values used to plan and execute the detailed work
Time phasing of the work	Roadmap->Release Planning->Sprint Planning. Priority based execution to deliver incremental capability.
Tracking performance against technical objective criteria to claim value.	Agile metrics: Velocity, burndown and burn up charts, etc. EVM Metrics: CPI, SPI, TCPI, Variance Analysis, done at feature level of above.
Compare claimed value, actual costs, and planned value to support daily decision making.	Sprint Retrospective, Story point claims, EVM % complete taken at feature level of above.
Updating forecasts and technical plan as the team learns from history.	Agile is in a constant state of planning and executing, allows for creating a forecast as often as daily.

Appendix B

**EXAMPLES OF AGILE EVM
PROGRESS TRACKING CHARTS**

Platinum Card EVM for Agile Development

NDIA



35

11/26/2018

Using Agile you can track your EV metrics using the Agile Release Plan, Burn Up chart and underlining data and QBD of Story completed against features, Backlog, and Velocity.

Here's how Agile relates to these

- Timeline: Center dashed line marked Time Now shows left as the past and right as the future.
- Blue line - Agile Release Plan is the schedule at the feature level and represents the BCWS in EV. This plan will be revised with each rolling wave. First, starting with the backlog and then the plan is adjusted over time. This gives the BAC.
- Purple line is the Agile Burn Up which is the completed work or BCWP in EV at Feature or WP level. It also represents the percent complete. In the future the dotted line represents the remaining backlog to complete the Feature or WP based on past velocity.
- Green line is Agile actual cost to do work performed or ACWP in EV. Projecting this into the future yields the EAC and future is estimated by Velocity times Remaining Backlog (where velocity has been dollarized).
- Historical velocity and remaining backlog are used to do forward estimation.
- Cadence releases should align with rolling wave planning.

Platinum Card Formulas

NDIA

Variations 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})

For reference only

Not expected to be briefed, only included for reference

Example: Progress Tracking Report

NDIA

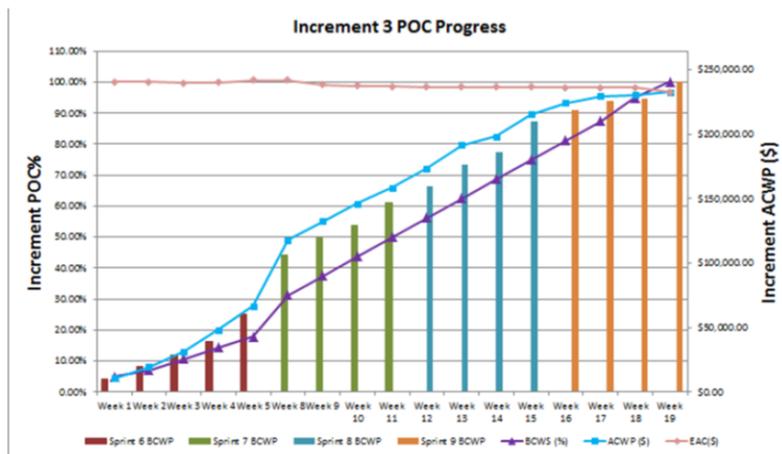


Figure 11: Example of a progress tracking report indicating both Agile and EVM progress data on graph.

37

11/26/2018

Brief description of this chart

This chart displays the Agile “burn-up” metrics via the bar graph (using the axis on the left), along with the Earned Value metrics via the lines (using the axis on the right). Gives the ability to visually see any disconnects or trends.

How is this chart used?

This chart is used to compare the Agile Metric data of work completed over time, to the costs expended and the EV plans that were established. Based upon the chart, the reviewer can determine if the program is trending to meet both cost and schedule per their plan.

Key for chart:

- EAC for the increment/release (salmon line)
- Budgeted Cost of Work Scheduled per week (purple line)
- Actual Cost of Work Performed per week (blue line)
- % of work completed in Agile Tool (bars)
- Cumulative - ACWP in blue, BCWS in purple, sprint bars - shows how as each sprint completes incremental progress towards the milestone

What is this chart telling us?

- In this example you can see that around Week 4 the trend showed that more work was completed than planned, however, the costs are relative to the work completed, representing no cost variance even if there was a slight positive schedule variance. For example, at week 4 the BCWS is about \$50,000 but the ACWP is about \$60,000, so this is showing a favorable schedule variance.
- Starting around Week 5 and 9 the trending data started to show that more costs were expended than the work completed, hence trending to a cost variance.
- EAC and ACWP are going down over time – puts in question how realistic this example may be.

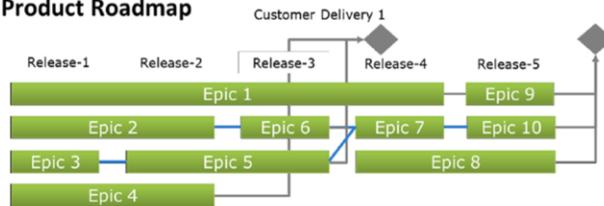
Appendix D

**PRODUCT ROADMAP, RELEASE
PLANNING, AND ROLLING WAVE
PLANNING PRODUCTS**

Roadmap Creation

NDIA

Product Roadmap



Product backlog

- Full scope of work
- Backlog items *include* sizing information

Product Roadmap provides

- High level
- Time-phased
- Priority based
- Aligned to customer milestones
- Framework for the PMB and IMS

Roadmap is developed

- Program start
- Refined through time

39

11/26/2018

The Product backlog

- Comprises the full scope of work required to satisfy the contract
- Scope is defined by Epics/Capabilities and Features. Note: Epics/Capabilities may optionally be decomposed into one or more levels between the Epics/Capabilities and the Features to help understand the product decomposition and facilitate planning, but any additional decomposition does not fundamentally alter the road mapping or planning processes.
- Backlog items include sizing information

The Program Product Roadmap provides a high level time-phasing of the backlog

- Ordering of work is geared toward providing a logical build-up of system capabilities, with higher priority capabilities being planned for earlier completion
- Roadmap aligns major work items (e.g., Epics) to customer milestones
- Roadmap is shared with the customer to obtain concurrence on high level program plan
- Roadmap includes key product dependencies which will form the basis for critical path
- Roadmap supports establishment of the PMB and IMS (roadmap creation should occur before IMS creation)

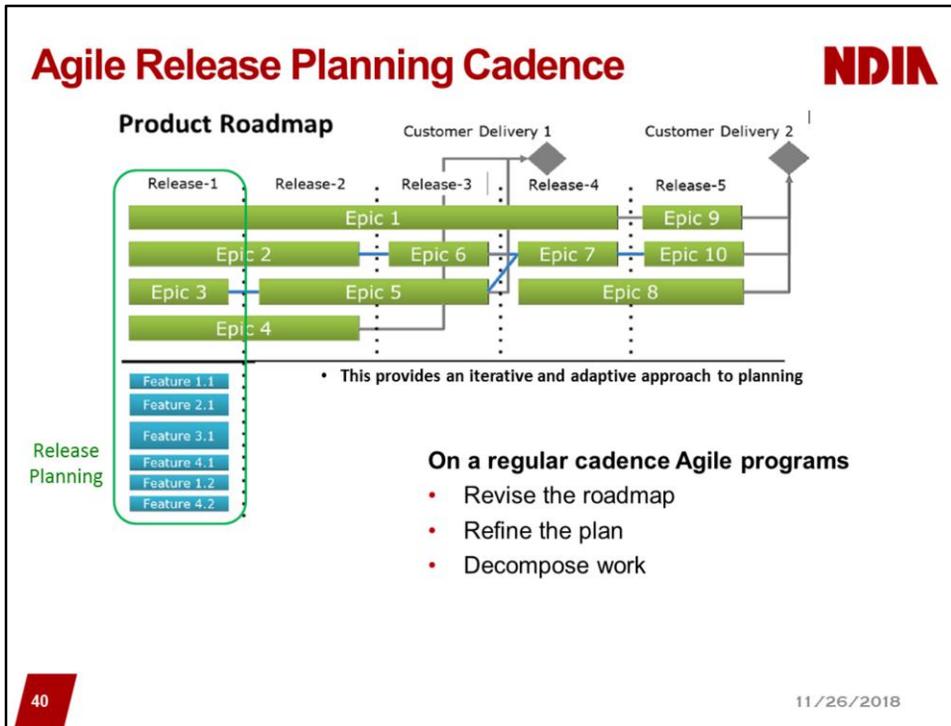
Roadmap is initially developed at program start

- The level of detail is coarser farther out in the future (near term Releases may show Features)
- Updates will occur every few months, nominally coincident with cadence based Release Planning

Key points in the figure

- Epics can span Agile cadence releases
- Key Epic-Epic dependencies are shown
- Dependencies necessary to support customer deliveries are shown
- Customer deliveries need not align with completion of Agile releases, as shown in Delivery 1

Agile Release Planning Cadence



On a regular cadence (nominally 2-4 months) Agile programs will conduct release planning

- Revise the overall roadmap based on new knowledge (e.g., changing priorities) and actual status of completed work
- Refine the plan for the near term period (i.e., the next cadence based Release)
- Decompose work into Features that can be completed within the next Release
- This provides an iterative and adaptive approach to planning that
 - Acknowledges uncertainty
 - Expects change
 - Keeps the focus on the outcomes, not the plan
 - Always uses the latest “truth” data to plan future work
 - Leverages rolling wave planning rhythms
 - Is entirely compatible with best practices for performance measurement

Although Agile is designed to accommodate change, caution must be exercised to manage that change, especially if it involves a change in scope. Scope changes cannot occur without a formal baseline change and should be done in a highly controlled manner. Even in-scope changes must be carefully managed to avoid excessive churn. Establishing well-defined ground rules with the customer up-front can help avoid problems in execution.

Key points in the figure

- The area in the green box shows what happens at release planning
- Features to be developed in the current release are defined/refined and planned for the current release
- Feature numbering in figure shows trace to Epics, e.g., Features 4.1 and 4.2 are part of Epic 4.
- Even within a planned release the features should be prioritized so the team understands which are most (and least) important should issues in execution occur.
- Feature level dependencies (not shown in the figure) are modeled; for Features that have no dependencies it is often convenient to have the Feature span the entire period of the release to enable maximum flexibility during execution to adjust the order of story and feature completion.
- Features are decomposed into Stories (not shown in the figure) to support planning and to support EVM measures during release execution. Using story completion is the preferred method for determining percent complete for a Feature.

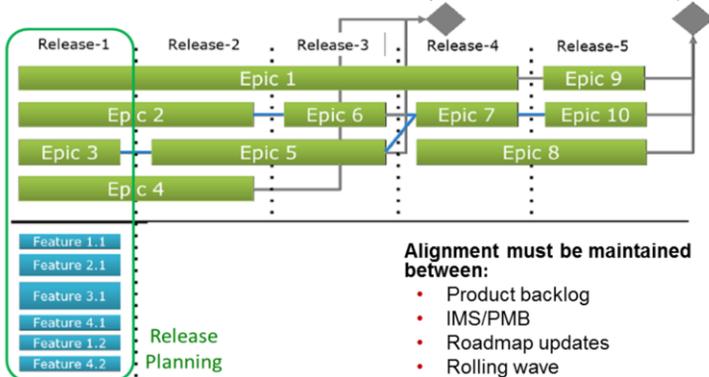
Rolling Wave Planning

NDIA

Rolling Wave Planning aligns with Agile Release Planning cadence

- Update PMB
- Maintain IMS in accordance to change management plan

Product Roadmap



41

11/26/2018

Rolling Wave Planning aligns with Agile Release Planning cadence (some programs consider release planning and rolling wave planning as coincident events)

- Decomposes Planning Packages into Work Packages
- Plans IMS for the next Release and may update future releases at a coarser (e.g. Epic) level
- Update PMB as needed based on plan changes

Alignment must be maintained between the backlog, IMS, and PMB as release planning, roadmap updates, and rolling waves occur by

- Maintaining mapping of Feature to Work Packages and Planning Packages
- Updating IMS with new work packages for the rolling wave
- Approving Baseline changes and signing Work Authorizations
- Decomposing Features to support percent complete calculations

Maintaining alignment between the PMB (including the IMS) and Agile products is critical!

Reminder: The relationship of Feature to work package (and IMS tasks) does not have to be (and frequently won't be) 1-1. For practical reasons a program may choose to model all or many features within an Epic (Control account) within the release to a single IMS task (work package).

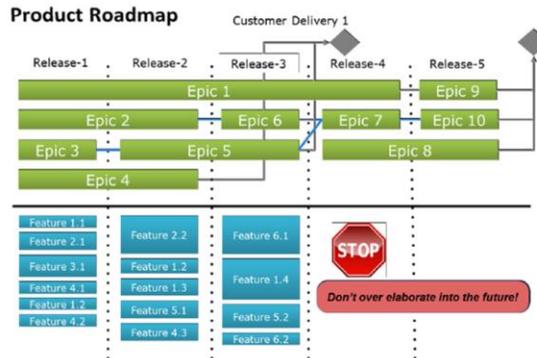
Note: Freeze period considerations are covered on a future slide

Roadmap Depth of Detail

It is often appropriate, at both roadmap creation and release planning to decompose *some* Epics into Sub-Epics and Features in future releases

Key points

- Only decompose where appropriate and needed
- Not all Epics will be decomposed to the same degree
- Level of decomposition may vary
- Features planned for future releases may require further refinement or decomposition before implementation



It is often appropriate, both at roadmap creation and release planning to decompose some Epics into Sub-Epics and Features in future releases to

- Understand scope
- Understand desired work sequencing for product build-up
- Understand critical path & dependencies to meet customer deliveries

Key points

- Only decompose where appropriate and to the depth needed (e.g., to model critical dependencies); avoid decomposing just because you can, especially farther out when there are more unknowns and increased detail only creates an illusion of increased accuracy.
- Other factors that will impact ordering besides dependencies are priority (customer value) and risk reduction (gaining early knowledge)
- Not all Epics will be decomposed to the same degree
- Level of decomposition may vary, even within a release time box
- Features planned for future releases may require further refinement or decomposition before implementation
- You might think that larger program should have a more detailed roadmap and IMS. Often the opposite is true. Excessive decomposition on a large program can actually create excess clutter and noise, making it harder to see the forest for the trees

Key points in the figure

- The physically bigger Feature boxes in Releases 2 & 3 imply larger “chunks” (Sub-epics) that will be further decomposed later (e.g., at release planning events)
- The roadmap (and IMS) will have a mix of Epics, Sub-Epics (i.e., oversized Features) and Features

Potential Planning Challenges



Roadmap Maintenance

- Ongoing process of updates, at least in alignment with release cadence
- Always reflect current truth

Mapping / movement of scope (Epics & Features) and budget.

- Product Backlog must reflect full program scope
- Mapped to work packages and planning packages
- If product moves budget must go with it

IMS Maintenance (Freeze Period baseline change control).

- Rolling wave and contractor defined freeze period
- Detail plan must be approved prior to the start of work
- Timing of planning and rolling wave can create challenges with “just in time” planning

Strike a balance

The flexibility of Agile, while maintaining integrity of cost, schedule, budget baseline.

43

11/26/2018

Roadmap Maintenance

- Always reflect **current truth- not a ‘what I wish’** – better to identify and acknowledge that a problem exists early than to allow wishful thinking to permeate the plan. No one likes bad news, but early identification provides more time to adjust and determine the best options
- This (current truth and the impact to overall schedule) can be a sensitive topic with both company management and customers
- As discussed on the prior slide, avoid over (or under) decomposition beyond the next release

Mapping / movement of scope

- This is not optional! Whatever specific practices and rhythms are employed it must support maintaining the integrity of the baseline and mapped between the Agile products and the PMB.

IMS Maintenance

Per DoD EVMSIG with respect to the freeze period, the following should occur:

- 1) The freeze period may be adjusted, through formal changes to a company’s system description or other supplementary guidance, to be short enough that it accommodates the Agile planning cycle, or
- 2) Customer direction to allow +changes in the freeze period may be obtained given support and participation of the Customer in release planning activities.

Optional discussion topic or if questions come up on this: Ideally release planning occurs prior to rolling wave planning. However, when “just in time” release planning cannot occur (e.g. because of internal company freeze periods or turn-around time for completing rolling wave updates) several options have been observed.

- Release planning is performed a few weeks earlier than desired, with the acknowledgement that variances will occur during execution as a result.
- Some degree of pre-release planning feature definition and sizing are done by the technical leads. This pre-planning is done to support rolling wave with the acknowledgement that variances will be identified at actual release planning and thus have to be managed during the release. In essence, an early “best guess” is done.
- Similar to the above, but the pre-planning (and IMS) are performed at a higher level of aggregation than Features. Using the example above, this early planning may include only 4 items, once each for Epics 1-4, with final decomposition of these still occurring at release planning. The IMS will only show one task for each epic, with the lower level Feature trace occurring below the IMS (e.g., in an Agile management tool).

Final point (not just for this slide, but more broadly): We have to acknowledge that specific company and program circumstances can make exact following of this NDIA or other guidance difficult. When that occurs focus on the core principles of EVM and Agile and find ways to adjust the details and practices in a way that will maintain the integrity of those core principles.

Appendix E

IBR CONSIDERATIONS

IBR Considerations - Introduction



- The purpose of the IBR Considerations Appendix within the NDIA Agile EVM Guide 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.
- This Appendix is not a comprehensive IBR checklist, but is limited to items that support the portions of the plan related to Agile methods
- The information provided prompts the IBR reviewer on areas to explore and questions to ask when looking at Agile artifacts in relation to evaluating the soundness of the program plan

IBR Considerations – Header Descriptions



IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
------------------------------------	----------------	-----------------------	---------------------------------------	--

Header Descriptions:

- **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 Considerations - Scope



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	<ul style="list-style-type: none"> SOW WBS/Dictionary IMP WADs MOD 	<ul style="list-style-type: none"> Product Backlog 	<p>Product Backlog:</p> <ul style="list-style-type: none"> At a minimum, contains a set of work items (typically called Capabilities or Epics) that cover the full breadth of the contract's technical scope. Backlog items map to the WBS Backlog items have size estimates* and acceptance criteria ** Requirements (top level specs, SOW) are mapped to Backlog items to demonstrate the Backlog encompasses the full scope of work

* Size Estimate: Backlog Items include an estimation of the "size" of each item, compared to other items in the backlog in order 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, Feature, and Story Level. Acceptance criteria constitute Definition of Done.

IBR Considerations - Time



IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
Time	Ensure the program has a viable IMS that supports the IMP, meets required integrity standards and demonstrates execution realism	<ul style="list-style-type: none"> • Contract Milestones • Program Summary Master Schedule • IMS • Schedule Risk Analysis 	<ul style="list-style-type: none"> • Roadmap 	<ul style="list-style-type: none"> • Roadmap: <ul style="list-style-type: none"> ○ Scope is included at a reasonable level of fidelity (capability/EPIC) and that there is a reasonable ordering of that scope over time. ○ Roadmap shows sequencing of scope and alignment to program milestones. Detail should be sufficient to facilitate critical path in the IMS. ○ Roadmap includes scope item size estimates ○ Roadmap consistent with staffing plan based on Roadmap item size estimates • IMS baseline is informed by the roadmap at an adequate level to insure 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, cadence release cycles)

IBR Considerations – Budget and Resources



IBR Project Management Constrains	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
Budget	Ensure the entire scope of work is included in a budget baseline and that adequate management reserve exists	<ul style="list-style-type: none"> Budget Logs (CBB) CAPs BOEs 	<ul style="list-style-type: none"> Product Backlog 	<p>Product Backlog:</p> <ul style="list-style-type: none"> Capabilities include a size estimate* based on assessment of technical size and complexity. The size estimate should be reliable to the budget value of the corresponding control account. Mapping of capabilities/EPICs/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	<ul style="list-style-type: none"> CAPs by EOC Org Chart/OBS RAM Roles & Responsibilities (RACI) Staffing Plan 	<ul style="list-style-type: none"> Agile teams defined Infrastructure for agile development defined (tools, environments, configurations, etc.) including the Agile management tool 	<ul style="list-style-type: none"> 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. Capabilities/Epics map to Control Accounts).

IBR Considerations – Quality and Risk



IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
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	<ul style="list-style-type: none"> Quality Management Plan Quality Assurance Plan Quality Metrics 	<ul style="list-style-type: none"> Product Backlog 	Product Backlog: <ul style="list-style-type: none"> Capabilities/Epics 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 Management (ROM) Plan	<ul style="list-style-type: none"> ROM Plan Risk and Ops Register Risk Mitigation Plans 	<ul style="list-style-type: none"> Backlog 	Backlog identifies significant risks and risk mitigation tasks as appropriate

** 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, Feature, and Story Level. Acceptance criteria constitute Definition of Done.

IBR Considerations – Project Integration



IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
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	<ul style="list-style-type: none"> • EVMS documentation • Program Procedures for baseline planning and baseline control • CBB Log • Technical execution documents and processes: Examples (PMP, SW Dev. Plan, SEMP) 	<ul style="list-style-type: none"> • Agile Framework • Backlog to IMS/EVMS mapping • Roadmap 	<ul style="list-style-type: none"> • Agile Framework: Appropriate to the type of program and deliverables desired, that indicates a well thought out plan. <ul style="list-style-type: none"> ◦ 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., SAFe) Key roles and 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

IBR Considerations – Customer Relations **NDIA**

IBR Project Management Constraints	Areas of Focus	Typical IBR Artifacts	Agile Specific Artifacts or Processes	Attributes of Agile Artifacts or Processes
Customer Relations	Ensure the programs priorities are aligned with the customers priorities	<ul style="list-style-type: none"> • Communication Plan • Joint Management Program and Business Management Review documentation, including agendas & participants • Org Chart that includes customer roles/mapping • Feedback (surveys, CPARs) • Business Rhythm Calendar • Program Management Chart Decks • Program Action Item Database 	<ul style="list-style-type: none"> • Increment or Release Review agenda and participants 	<p>Communication plan includes:</p> <ul style="list-style-type: none"> • 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 • Content, format, analysis method and frequency of Agile measures agreed to with the customer as part of the program business rhythm and customer reviews.