

Lessons Learned Applying EVMS on Agile Programs

Alexandria, VA
February 19, 2015



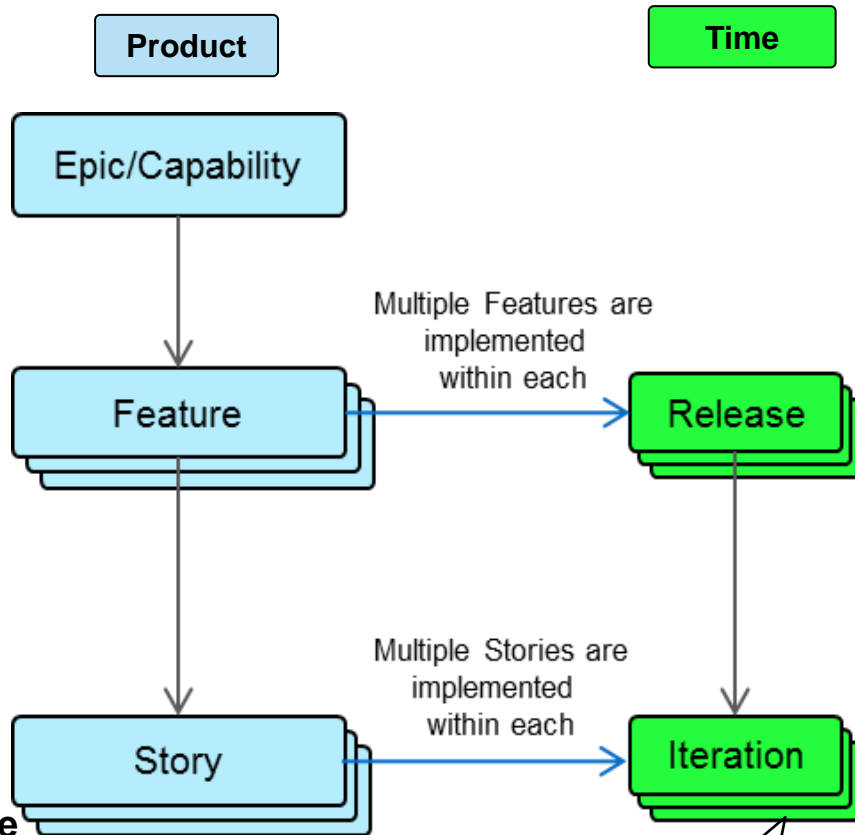
Rob Eisenberg LM Fellow
robert.j.eisenberg@lmco.com
301-240-5510

Ron Terbush LM EVM SME
ronald.j.terbush@lmco.com
301-240-6573

Agile Terminology



- **Epic/Capability:** A high level major system function.
- **Feature:** A well defined system function to be completed within a release.
- **Story:** A small but well defined system function that can be developed within one iteration.



- **Release:** Release content has clear goals and objectives and occurs on a regular cadence (not to be confused with a program milestone).
- **Iterations:** Recurring, non overlapping, cadence for development (nominally 2 – 3 weeks).

“SPRINT” is overtaking “iteration” at LM

Work Break Down Structure (WBS)



The WBS organizes the project deliverables into product based manageable units of work

The Agile WBS will nominally follow one of two basic structures, referred to here as “release centric” or “capability centric”. Which variation is employed is primarily driven by how the customer views the product to be delivered.

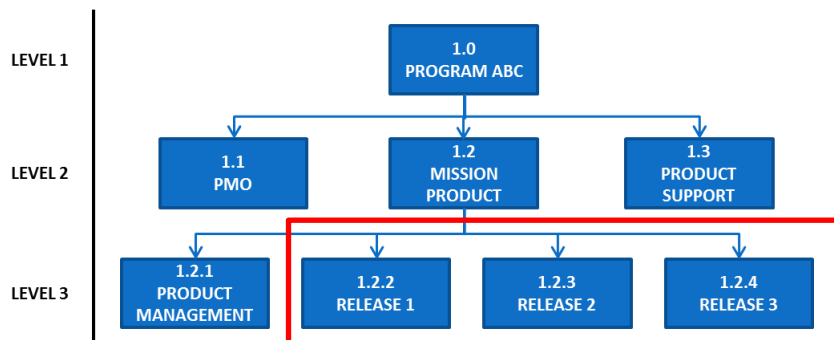
RELEASE CENTRIC

The customer views the product in terms of release. An example of this might be a large satellite ground system where the releases are based around major system events such as launch support, initial calibration, initial operations, and full system operations.

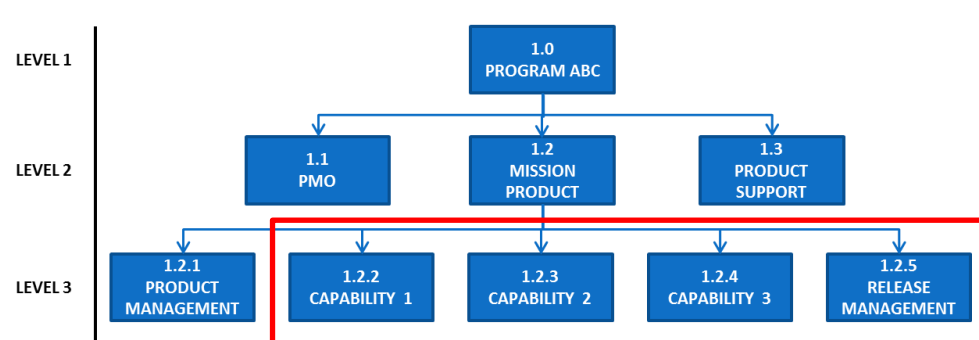
CAPABILITY CENTRIC

The customer views the product in terms of a set of discrete capabilities, where the releases are primarily viewed as time boxes for the ongoing and sustained delivery of Features. The release content may change greatly over time based upon changing priorities

WBS – RELEASE CENTRIC



WBS – CAPABILITY CENTRIC



Work Breakdown Structure



- **Challenges**

- **Customer requires MIL-STD-881C WBS**
- **Definition of “Release” was ambiguous (cadence vs. customer milestone/event)**
 - **WBS based on release cadence drives Control Account proliferation & administration**
 - **Defining the WBS based on customer milestone/event is sub optimal**
- **Transitioning to agile from a waterfall WBS**
- **Segregation by CLIN/Funding source**

- **Lessons Learned**

- **Utilize capability based WBS combined with customer milestone/event based IMP**
- **Work with customers to change traditional WBS practices**

Agile Program Planning

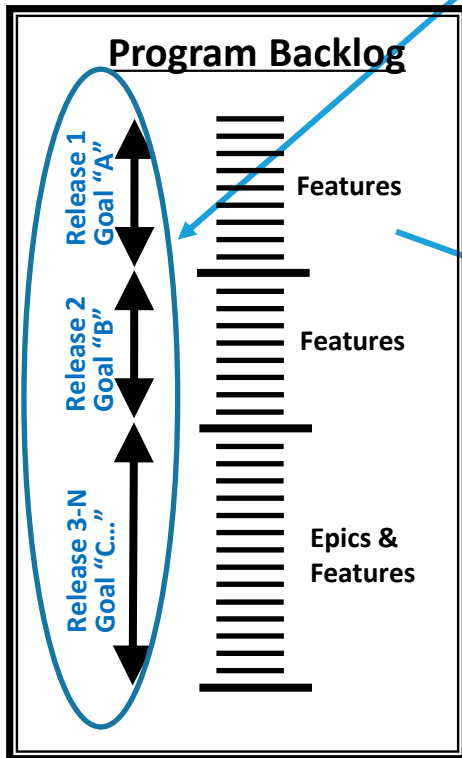


DEFINE THE WORK

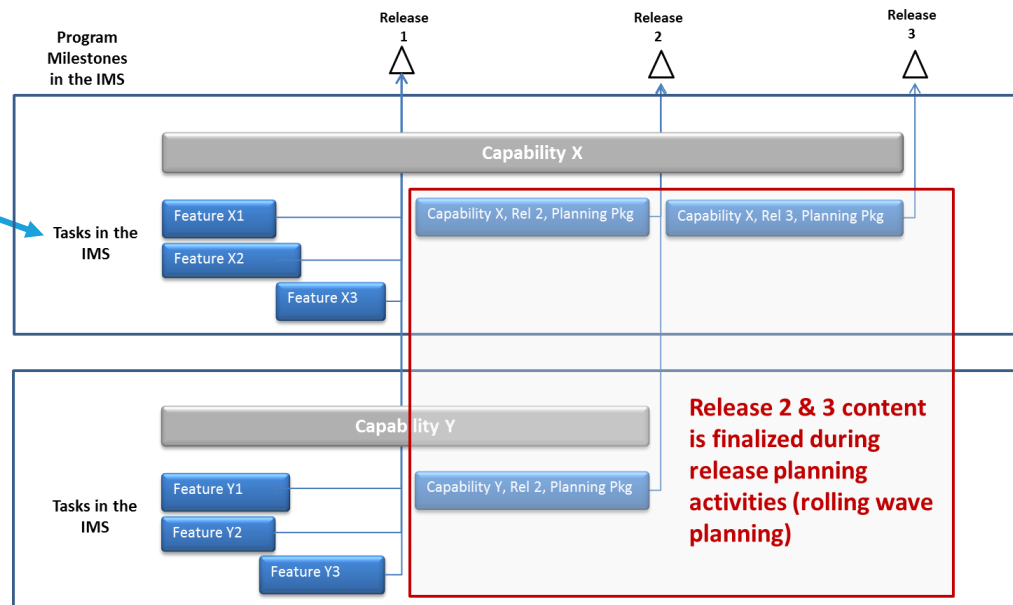
PLAN THE WORK

SOW Requirements are mapped to Epics and Features in the Program Backlog

The Program Plan is reflected in the Release Roadmap, which is an initial allocation of Features and Epics from the Program Backlog to releases based on the objectives and goals of each release.



SCHEDULE THE WORK



Features are prioritized using the Release Roadmap and planned in the IMS.

Cross release planning occurs before the first release begins, later releases will be less well-defined

Agile Program Planning



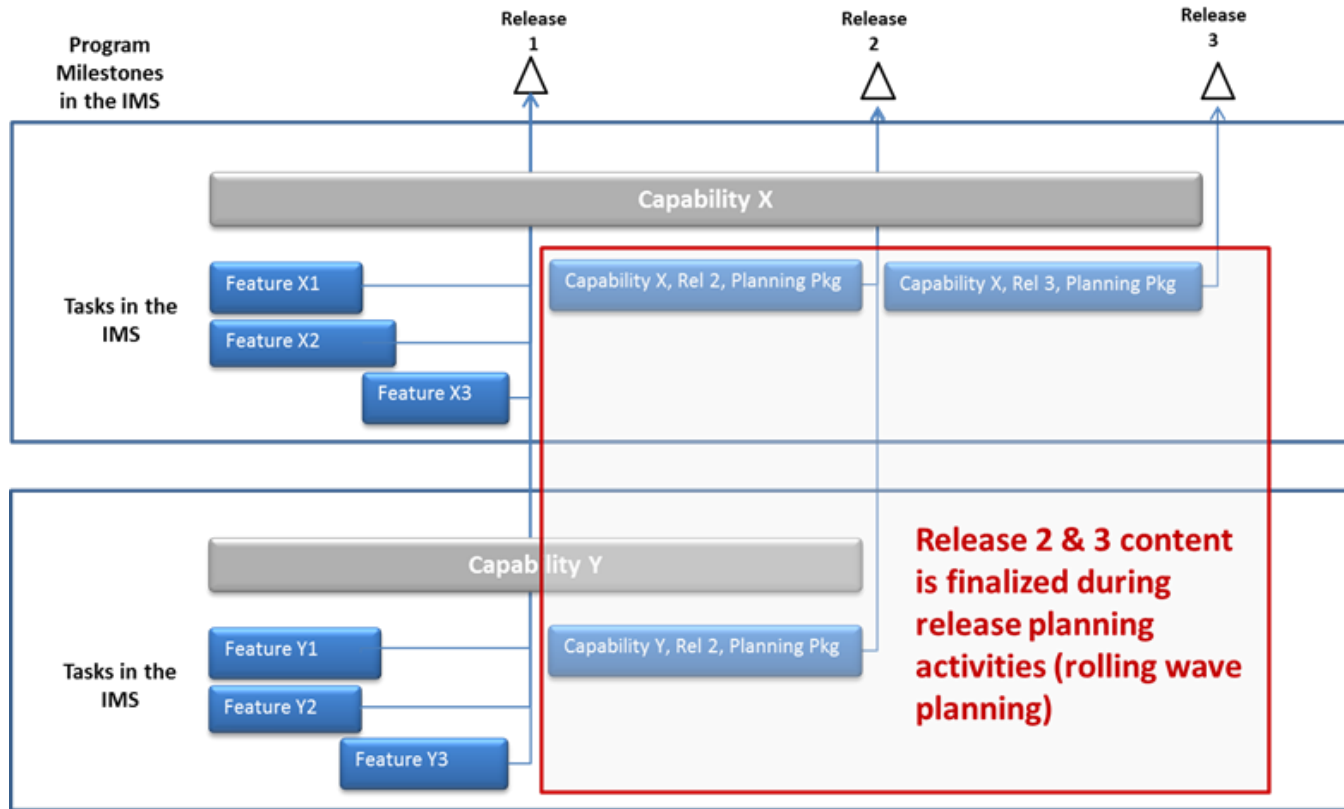
- **Challenges**

- **Mapping requirements (scope and budget) to Epics and Capabilities**
 - Bid waterfall ... executing agile
 - Transitioning from functional BOEs to Epics & Capabilities
- **Agile programs with undefined scope (bid as capacity)**
- **Culture including roles and responsibilities**

- **Lessons Learned**

- **Transitioning from bidding work in a waterfall fashion to Agile took some time**
- **Overall agile approach to planning is working well**
- **Agile programs with undefined scope do not accommodate EVM easily (and the same is true for waterfall)**
- **Cultural changes are harder than technical changes**

IMS and Critical Path



- The IMS should only go down to the level of Features (not story level)
- Utilize Rolling Wave Planning at Release Points
- Feature completion criteria and interdependencies are clearly defined

IMS and Critical Path



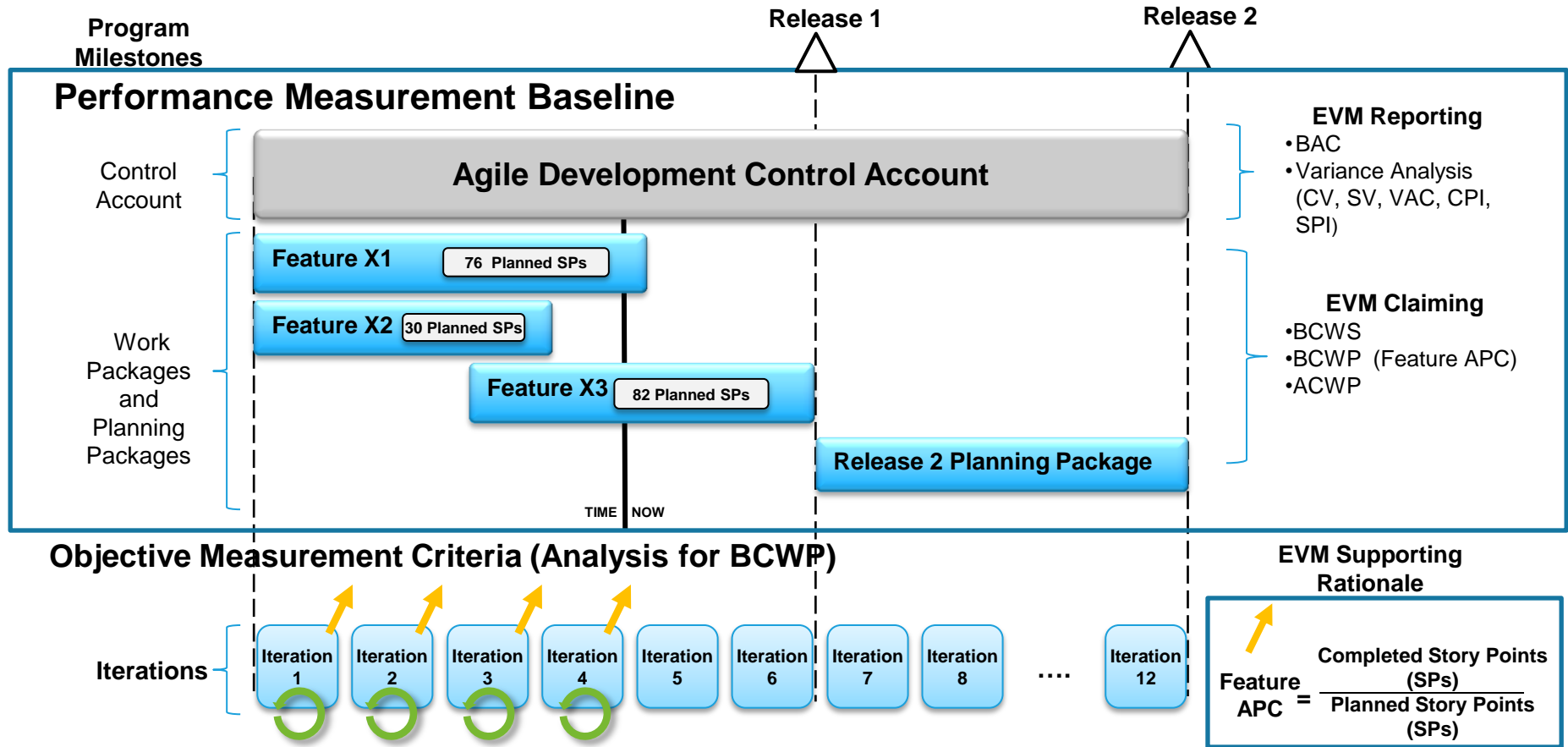
- **Challenges**

- **Just-in-time rolling wave planning (Change control period)**
- **Release cadence span too long**
- **Feature Acceptance Criteria drives completion (not completion of planned story points)**
- **Feature duration greater than 40 days**
- **Traditional Schedule Risk Analysis (Monte Carlo)**

- **Lessons Learned**

- **Leverage customer direction to bypass change control period**
- **IMS should provide critical path at high level (e.g. Features)**
 - **Story interdependencies can be modeled in agile tool**
- **IMS tasks at Feature level allows freedom to prioritize/update stories within the feature without impacting the IMS. Stories provide QBD.**
- **Use capacity, backlog and velocity for Schedule Risk Analysis**
- **Incorporate agile metrics into customer reviews and status meetings (replaces detailed IMS metrics – LS/LF)**

Control Account Hierarchy & EV



**Features are comprised of stories.
Each story is assigned a weighted story point (SP) value.
SP's are claimed at the completion of a story!**

Control Account Hierarchy & EV



- **Challenges**

- How to compute APC when stories change (added or deleted)?
 - Is scope the number of planned story points or feature acceptance?
 - Originally defined scope as number of SPs in order to manage change (prevent scope creep). Solved one problem but created another.
- Story credit (0/100) is not given until story acceptance at iteration demo. Iteration that spans accounting month causes roller-coaster SV/CV spikes.
- What happens to unfinished work at iteration and release points?

- **Lessons Learned**

- Objective Criteria (completion of stories at weighted SP value) is easy and objective.
- Agile team discipline (daily & iteration assessments) supports EV status & forecasting extremely well – better than non-agile programs.
- Clearly defined completion criteria allows the stories within a Feature to evolve without a change to budget.
- Consider taking 100% credit when Product Owner approves story (prior to demo). If other stakeholder involvement in approval is deemed critical take partial credit for stories when Product Owner approves, but pending demo (e.g., 80% at PO approval, 100% at demo acceptance).
- Iteration and release boundaries have no impact on unfinished work.
- Customer partnership and two way trust is critical for change management.

Estimate To Complete (forecasting)



	Date	Iteration #	Planned SPs per Iteration	Actual SPs per Iteration	Remaining SPs (Best Case)	Remaining SPs (Most Likely Case)	Remaining SPs (Worst Case)
Average SPs per iteration			AVG 32	Act AVG 30	AVG 36	AVG 30	AVG 24
Release A	10-Feb	1	29	16			
	24-Feb	2	32	31			
	9-Mar	3	33	37			
	23-Mar	4	33	25			
	6-Apr	5	32	39			
	TIME NOW				174	174	174
	13-Apr	6	33		138	144	150
	27-Apr	7	33		103	115	126
	11-May	8	33		67	85	102
	25-May	9	33		31	56	78
8-Jun	10	31		0	26	54	
TOTAL			322	148			
Release B	15-Jun	11				0	30
	29-Jun	12					6
	13-Jul	13					0
	27-Jul	14					
	10-Aug	15					
	24-Aug	16					
	7-Sep	17					
	21-Sep	18					

ASSUMPTIONS

BEST CASE: Avg SPs of highest 3 iterations
 MOST LIKELY: Avg SPs of all 5 iterations
 WORST CASE: Avg SPs of lowest 3 iterations

SCHEDULE FORECAST (TO GO)

BEST CASE: 5 Iterations
 MOST LIKELY: 6 Iterations
 WORST CASE: 8 Iterations

Agile team performance to date (velocity) provides a basis for forecasting estimate to complete (ETC) for the remaining work

Estimate To Complete (Forecasting)



- **Challenges**

- Determining ETC beyond current release.
- New and immature agile teams may have inconsistent velocity.

- **Lessons Learned**

- Program Backlog should be “coarse sized” to allow forecasting across releases.
- New teams will need a few iterations before accurate forecasting using velocity can be performed.
- Burn Down Charts (agile metrics) expose unfinished work. Gives insight into schedule and cost growth.
 - PM feedback *“Objective status of completed stories provided real progress and translated into early & fairly accurate ETC projections. ETC growth was quickly identified.”*

Do's and Don'ts



– DO

- **Leverage agile metrics and planning practices to support EVM planning, status, forecasting and analysis**
- **Have a product centric WBS**
- **Have a feature based IMS**
- **Use Feature completion criteria to define scope**
- **Use Rolling Wave Planning**
- **Size all Epics and Features in the program backlog**

– DON'T

- **Establish a release based WBS**
- **Put stories or iterations in the IMS**
- **Follow EVM or agile rules blindly**

Remaining Challenges

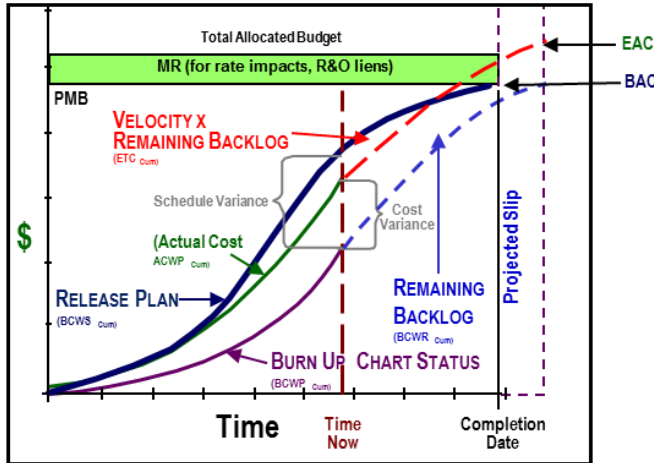


- **Change Management**
 - Managing change that may involve a change in scope

- **Culture**
 - Agile is undisciplined and other myths
 - Changing roles and responsibilities

- **Contracts & RFPs**
 - Require traditional milestones (PDR, CDR)
 - Require traditional documentation (artifacts)
 - Require WBS that is not accommodating to agile

PLATINUM CARD EVM FOR AGILE DEVELOPMENT



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)]

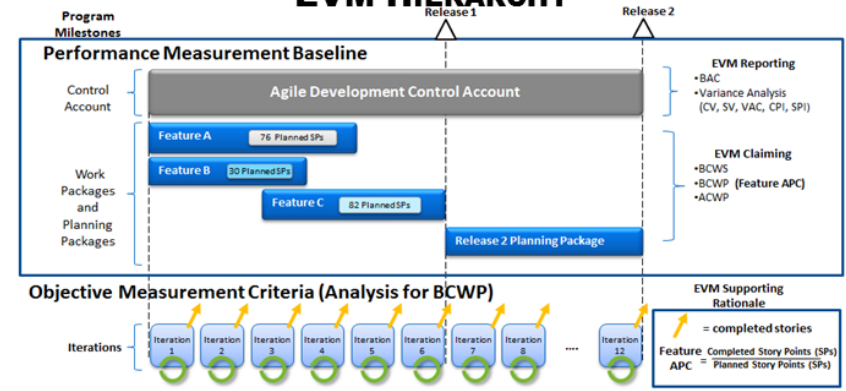
$$EAC_{CPI} = ACWP_{CUM} + [(Release Plan - Remaining Backlog) / CPI_{CUM}]$$

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

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

$$TCPI_{EAC} = \frac{Remaining Backlog (Velocity * Remaining Backlog)}{(BAC - BCWP_{CUM}) / (EAC - ACWP_{CUM})}$$

AGILE DEVELOPMENT EVM HIERARCHY



Agile Terminology

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

Acronyms

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

