How Agile is Earned Value Management?

Meri Duncanson
September 2019

01 What is EVM

02 What is Agile

03 Agile and EVM

04 Questions

05 Further Reading

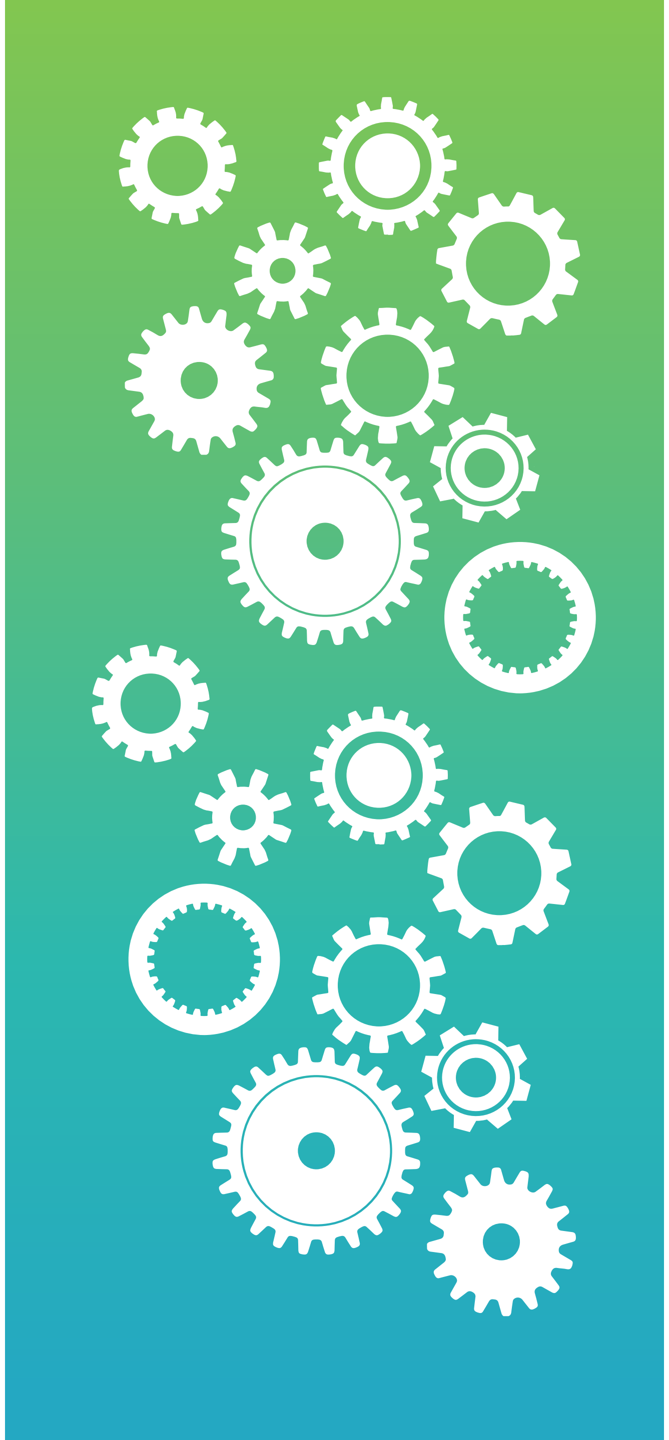
Earned Value Management

- EVM is a technique to measure performance of a project as it moves from project initiation to project closure. It provides a means to forecast future performance based on past performance (PMI PMBoK)

- EVM is a project control process based on a structured approach to planning, cost collection and performance measurement (APM)

- EVM is a project management technique for measuring project performance and progress in an objective manner (Wikipedia)

- EVM facilitates the integration of project scope, time and cost objectives and the establishment of a baseline plan for performance measurement (APMG)



Earned Value Management

EVM is a disciplined systematic project management process that integrates the project’s technical scope of work with schedule and cost for project planning and control.

The goal of EVM is to develop a realistic project performance measurement baseline which is then used to measure performance.

The resulting performance metrics provide early visibility into technical, cost, and schedule variances and issues which can then be used for decision making and corrective actions.

EVM provides quantitative data for analysis and project decision making using trending information to determine future project performance.

EVM provides answers to the following questions

How much have we spent?

How much did we get for our money?

How much longer will it take to finish?

How efficiently is the work being done?

What has happened - Why are we not on schedule/budget?

What is happening - What are the impacts?

What is going to happen - What are we doing to fix this?

How much more will we have to spend to complete the entire scope of work?

Earned Value Principles

1. Organize the entire scope of the project using a Work Breakdown Structure (WBS)
2. Organize the project team using an Organization Breakdown Structure (OBS)
3. Integrate the project work with the project team to create management control points (Control Accounts)
4. Schedule the project work in the Control Accounts across the entire project duration at the appropriate level of detail
5. Establish time-phased budgets for the scheduled work in the Control Accounts
6. Establish the scope/schedule/budget baseline as the Performance Measurement Baseline (PMB)
7. Authorize the scope/schedule/budget and control the start/stop of work
8. Periodically measure the schedule and the value of completed work and determine the Earned Value
9. Record direct costs (actual costs) and summarize into the Control Accounts
10. Compare planned, accomplished, and spent to analyze the performance and associated variances
11. Develop realistic time and cost estimates for the remaining effort in the Control Accounts
12. Rigorously control changes to the Performance Measurement Baseline

Earned Value - Definitions

BCWS / PV

How much of the work should be complete?

The portion of the project budget planned to be done at any given point in time.

BCWP / EV

How much of the work is complete?

The percent of the total budget actually completed at a point in time.

ACWP / AC

How much did it cost to do the work?

The amount spent to accomplish the work that has been completed.

Earned Value - Definitions

BAC

What was the total job supposed to cost?

The total approved budget allocated to complete the project

EAC

What do we now expect the total job to cost?

The estimated total cost of the project when the project is complete.

EAC = ACWP + ETC

Earned Value - Definitions

Schedule Variance (SV)

The difference between the work you achieved and the work you had planned to achieve

SV = EV-PV
SV = BCWP=BCWS

Cost Variance (CV)

The difference between the work you achieved and the amount actually spent for the work achieved

CV = EV-AC
CV = BCWP-ACWP

Earned Value - Definitions

Schedule Performance Index (SPI)

The ratio of the approved budget for the work performed to the approved budget for the work planned

SPI = EV/PV
SPI = BCWP/BCWS

Cost Performance Index (CPI)

The ratio of the approved budget for the work performed to what was actually spent to perform the work

CPI = PV/AC
CPI = BCWP/ACWP

Earned Value - Definitions

Variance at Completion (VAC)

The difference between the starting project budget (BAC) and the forecasted project budget when the project is complete (EAC)

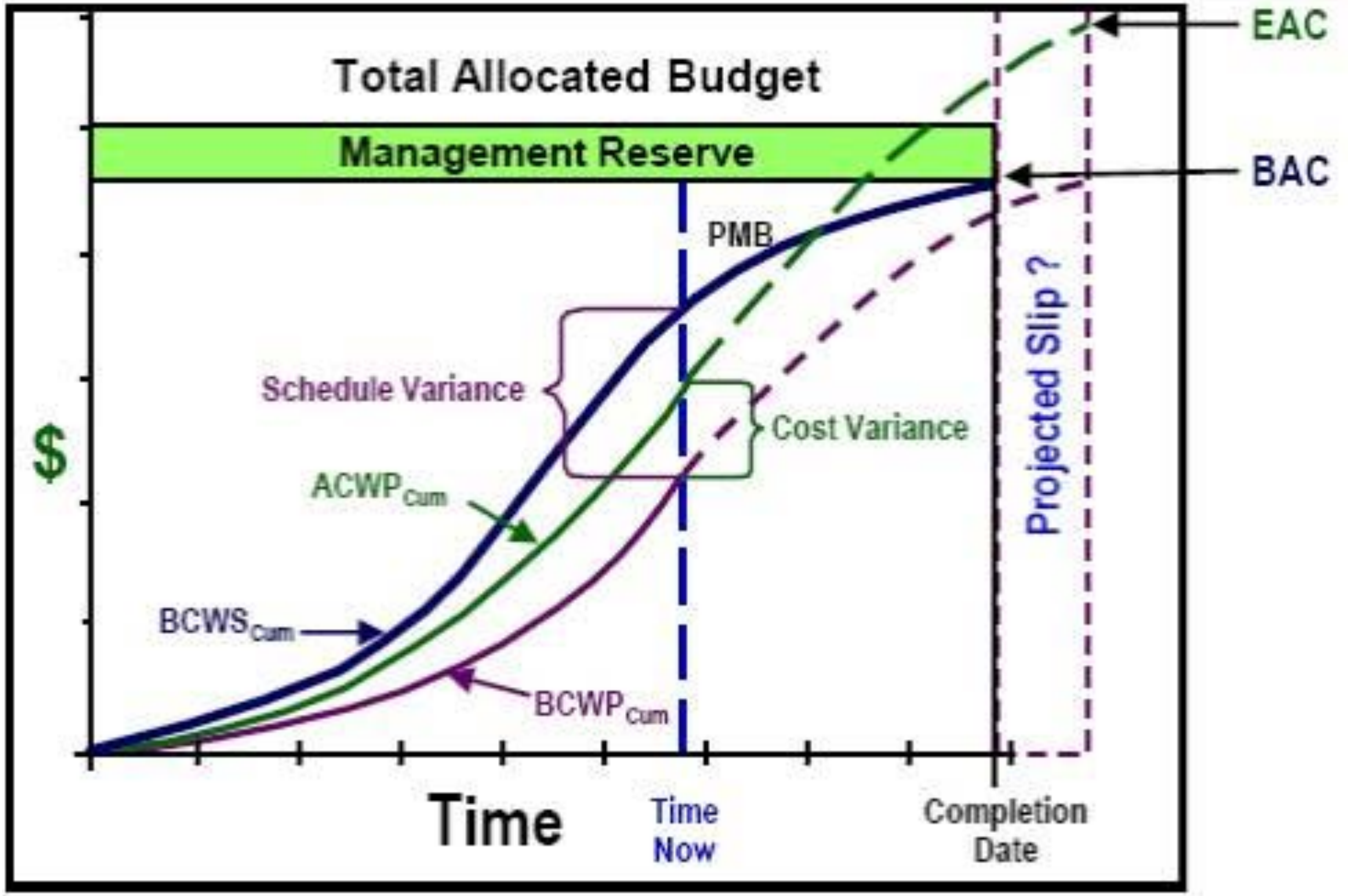
VAC = BAC - EAC

To Complete Performance Index (TCPI)

The estimate of the future cost performance that is needed to complete the project within the approved budget

TCPI = (BAC-EV)/(EAC/AC)
TCPI = (BAC-BCWP)/(EAC/ACWP)

Earned Value - Definitions



Build a wall

Scenario:

Building a wall – by the end of August we have budgeted to have completed $5,500 worth of work, however we have spent $6,100 and only achieved $5,000 worth of work.

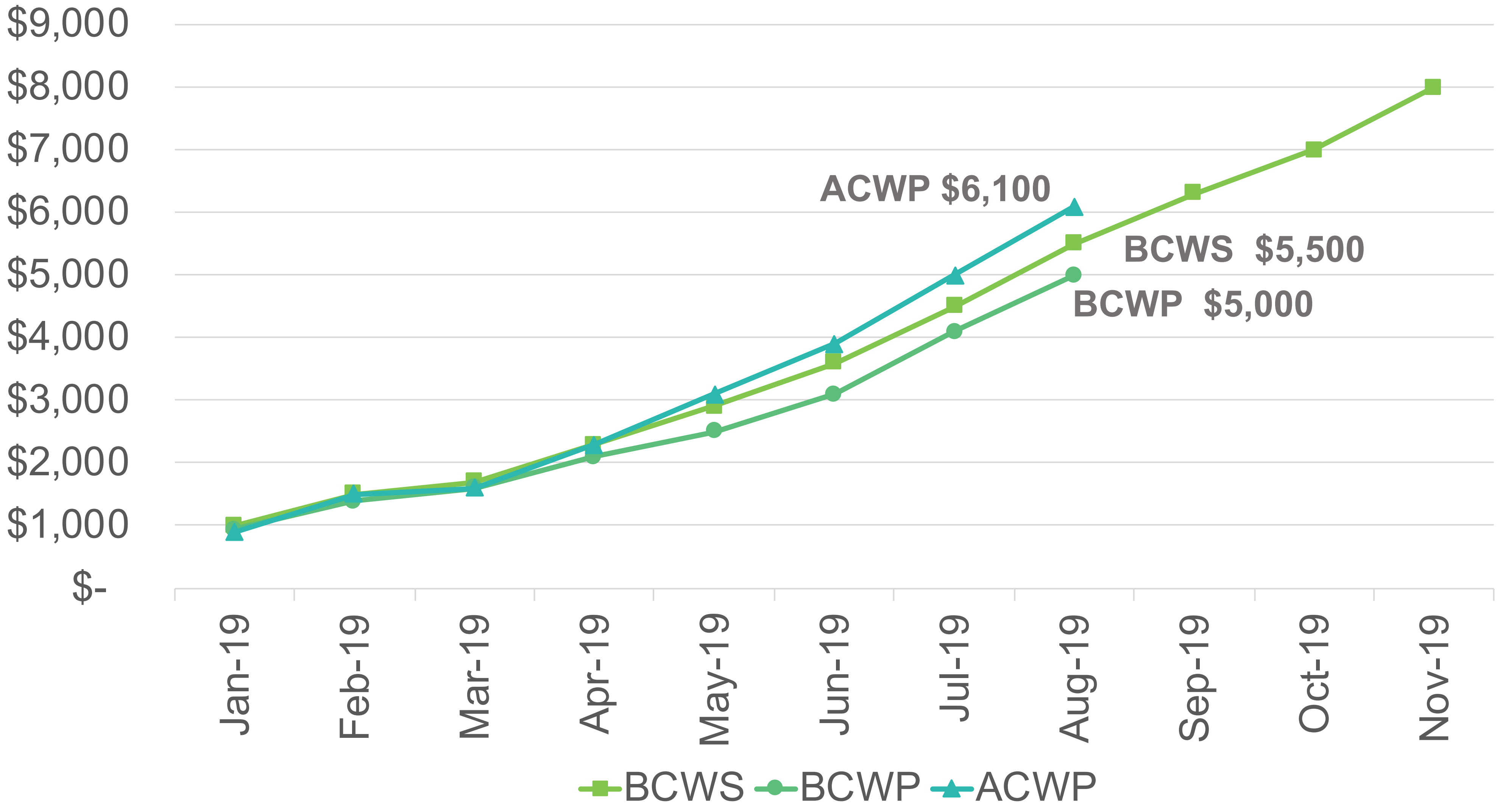
EV calculation:

- BCWS = $5,500,
- BCWP = $5,000,
- ACWP = $6,100;
- Cost Variance = −$1,100;
- Schedule Variance = −$500

EV analysis:

In August, we spent $6,100 but only accomplished $5,000 worth of work; therefore, there is a cost overrun of $1,100 as well as a delay in schedule of $500.

Build a wall



$9,000
$8,000
$7,000
$6,000
$5,000
$4,000
$3,000
$2,000
$1,000
$-
Jan-19
Feb-19
Mar-19
Apr-19
May-19
Jun-19
Jul-19
Aug-19
Sep-19
Oct-19
Nov-19
ACWP $6,100
BCWS $5,500
BCWP $5,000
BCWS
BCWP
ACWP

Build a wall

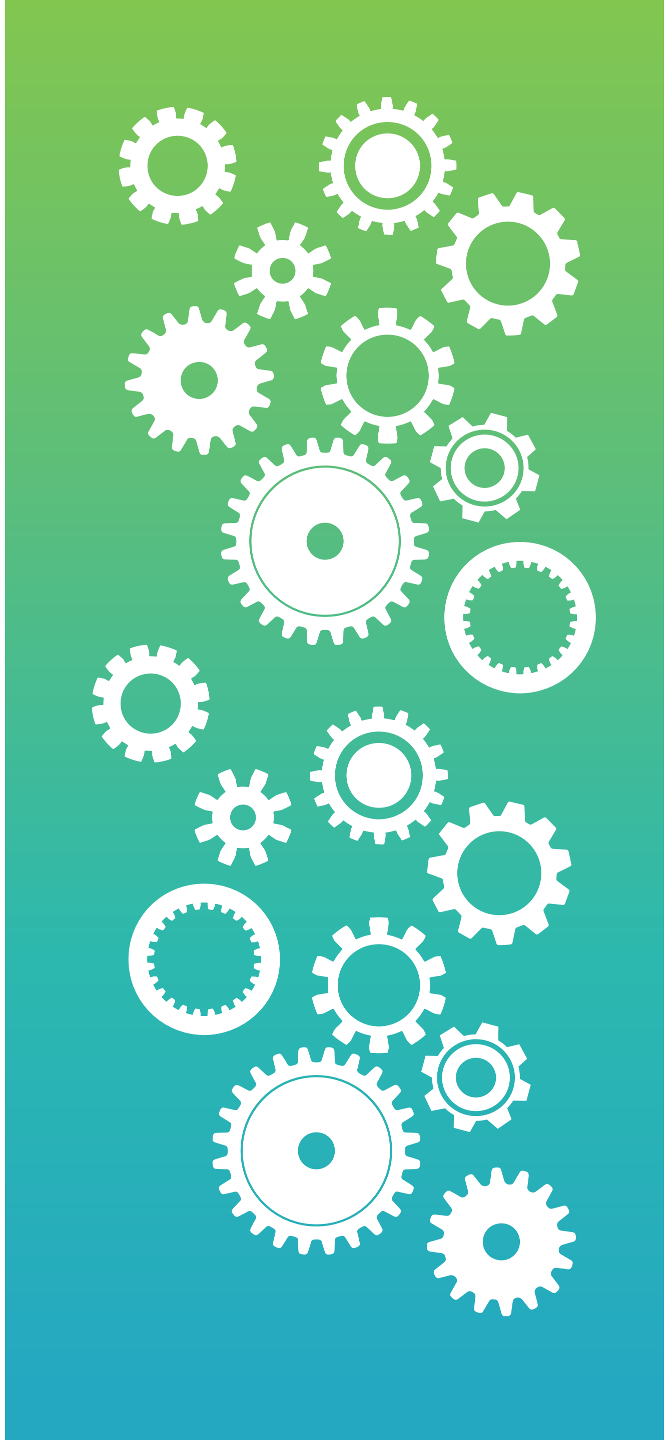
The next step would be to review these results and determine whether action is needed.

- Accept the schedule delay and find the additional funding to cover the cost overrun and schedule delay.
- Recommend changing an underlying condition that is causing the unfavorable results. These include:
  - **Scope:** Have the requirements have increased beyond what the allocated schedule and cost can manage? Suggest returning to the original scope or implementing a baseline change or CCP.
  - **Schedule:** Consider how the schedule was determined. Was it by an engineering analysis and a work breakdown structure, or, was it determined by an imposed date? Suggest structuring a realistic schedule.
  - **Cost:** Consider team productivity and whether the original plan made unrealistic assumptions. Consider adding additional, exceptionally capable staff. However, keep in mind that, in general, increasing staff significantly usually will not erase lost productivity but will incur additional costs.

AGILE

- Agile is is an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers/end users. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change. (Wikipedia)

- Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly. (Atlassian)



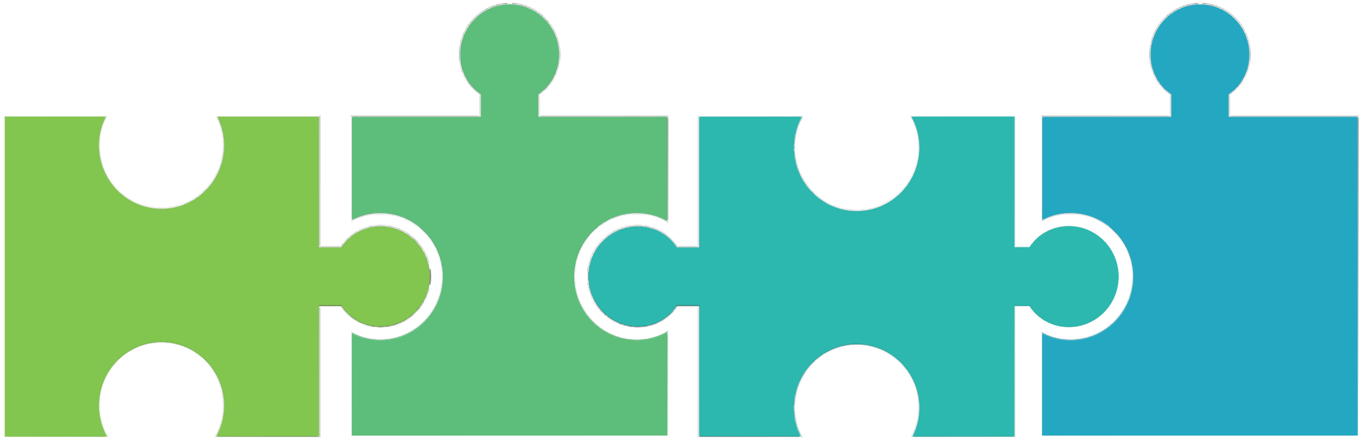
AGILE

Agile development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development.

Agile is a mindset, described by four values and twelve principles, manifested through and unlimited number of practices, tools and processes.

Agile relies on adaptability and the ability to reprioritize based on stakeholder requirements as the product/project develops.

Four Values of Agile



Individuals and interactions
over processes and tools

Working software
over comprehensive documentation

Customer collaboration
over contract negotiation

Responding to change
over following a plan

Twelve Principles of Agile

1. Deliver customer satisfaction by delivering valuable software continuously
2. Always accept change of requirements no matter how early or late in the project
3. Deliver software that works within a shorter timescale
4. Both developers and business professionals must work closely together daily throughout the project
5. Information is best transferred between parties in face-to-face conversations
6. Motivate people to build a project by creating an environment of appreciation, trust, and empowerment
7. Working software is the key measure of progress
8. The agile process promotes sustainable development
9. Continuous attention to excellence and quality in technical development and design boosts the agility
10. Simplicity is a vital part of effective agile management
11. Self-organized teams produce the best architecture, requirements, and design
12. Teams should reflect through inspection and adaption to be more effective

Agile- Definitions

Story

Feature

Epic/Capability

Product Backlog

Story

The smallest unit of work in an agile framework.
It should be clear, feasible and testable.

“As a <user role> I want to achieve <goal> so that I get <benefit>.”

Feature

Discrete functionality within an Epic/Capability, scheduled to be completed within a release.
Comprised of a collection of stories, features should have clearly defined objective technical completion criteria.

Epic/Capability

High level system functionality defined by the customer or stakeholder, split into Features and eventually Stories
Epics are parts of the product backlog and have a form of relative sizing estimate.

Product Backlog

The list of all the items/features that need to be built, determining the total scope of the work for the project/release.
Each item is sized and quantified into a number of Stories.

AGILE- Definitions

Sprint Backlog

Sprint

Release

Velocity

Sprint Backlog

The subset of product backlog that a team targets to deliver during a sprint in order to accomplish the sprint goal and make progress toward a desired outcome.

Sprint/Iteration

A fixed time period to complete a set amount of work.
Sprint planning is a collaborative event where the team answers two basic questions: What work can get done in this sprint, and how will the chosen work get done.

Release

A combination of features that when packaged together make for a coherent deliverable to customers or users.

Velocity

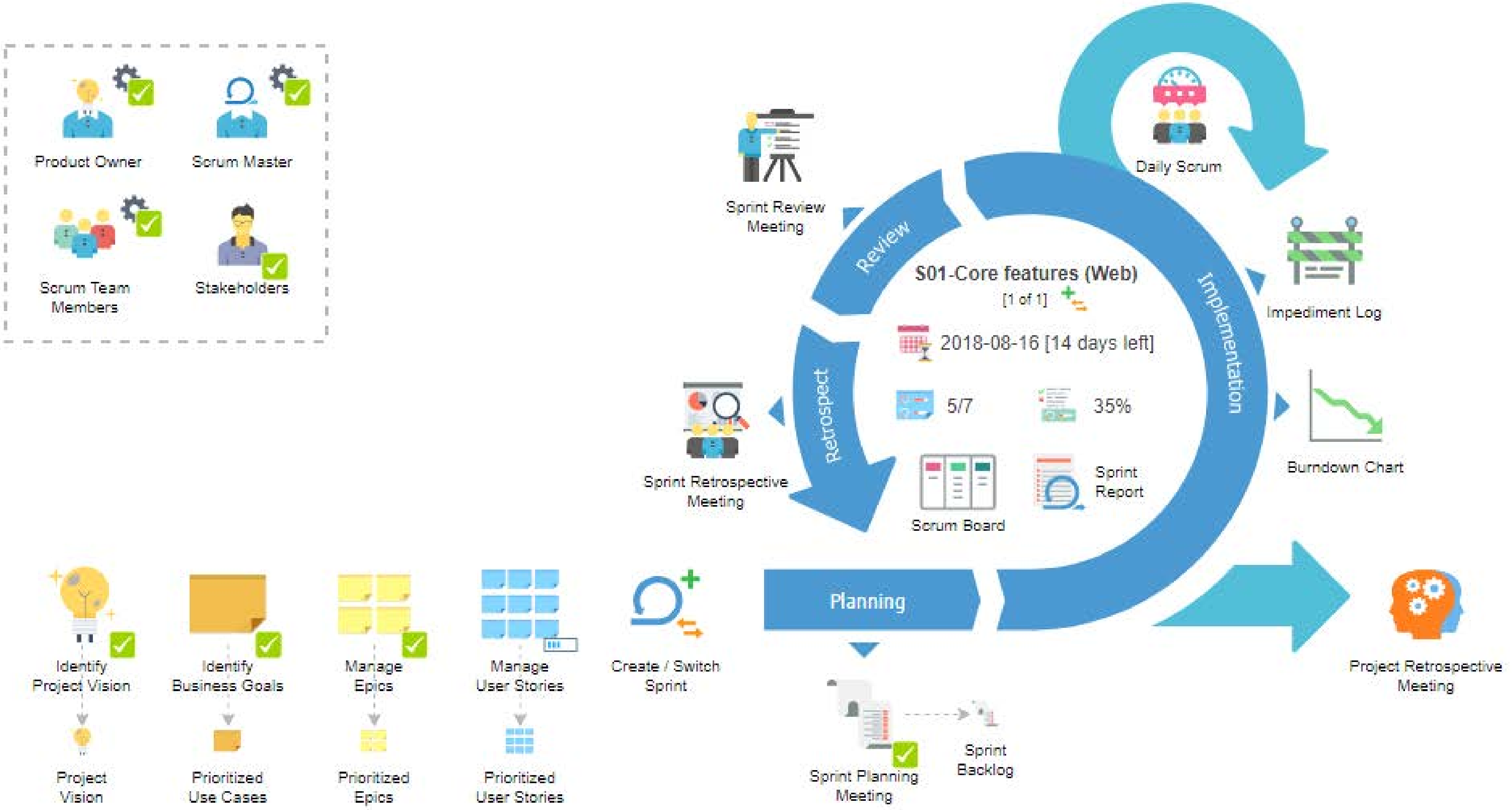
The ‘amount’ of functionality a team can deliver within a single iteration, the Velocity is measured by the number of points a development team can complete, based on its actual past performance.

Agile and Scrum

Agile is a structured and iterative approach to project and product development. It recognizes the volatility of product development, and provides a methodology for self-organizing teams to rapidly respond to change.
Agile is a framework and there are various approaches to implement Agile - one of the most popular is Scrum.

Scrum is a framework that assists teams to work together. Scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on the wins and losses to continuously improve.
Scrum describes a set of meetings, tools, and roles that work in concert to help the teams structure and manage their work.

Scrum



Product Owner
Scrum Master
Scrum Team Members
Stakeholders
Sprint Review Meeting
Daily Scrum
Review
S01-Core features (Web)
[1 of 1]
2018-08-16 [14 days left]
5/7
35%
Implementation
Impediment Log
Retrospect
Sprint Retrospective Meeting
Scrum Board
Sprint Report
Burndown Chart
Planning
Identify Project Vision
Identify Business Goals
Manage Epics
Manage User Stories
Create / Switch Sprint
Project Retrospective Meeting
Project Vision
Prioritized Use Cases
Prioritized Epics
Prioritized User Stories
Sprint Planning Meeting
Sprint Backlog

Scrum process

The Scrum Alliance defines the process through the following steps:

- A product owner creates a prioritized wish list, known as a product backlog.
- During sprint planning, the team pulls a portion of scope from the top of the product backlog, thereby creating a sprint backlog, and decides how to implement those pieces.
- The team has a certain amount of time - a sprint (usually two to four weeks) - to complete its work, but it meets each day in a daily Scrum to assess its progress.
- Along the way, the Scrum Master keeps the team focused on its goal.
- At the end of the sprint, the work should be potentially shippable: ready to hand to a customer, put on a store shelf, or show to a stakeholder.
- The sprint ends with a sprint review and retrospective.
- As the next sprint begins, the team chooses another chunk of the product backlog and begins working again.

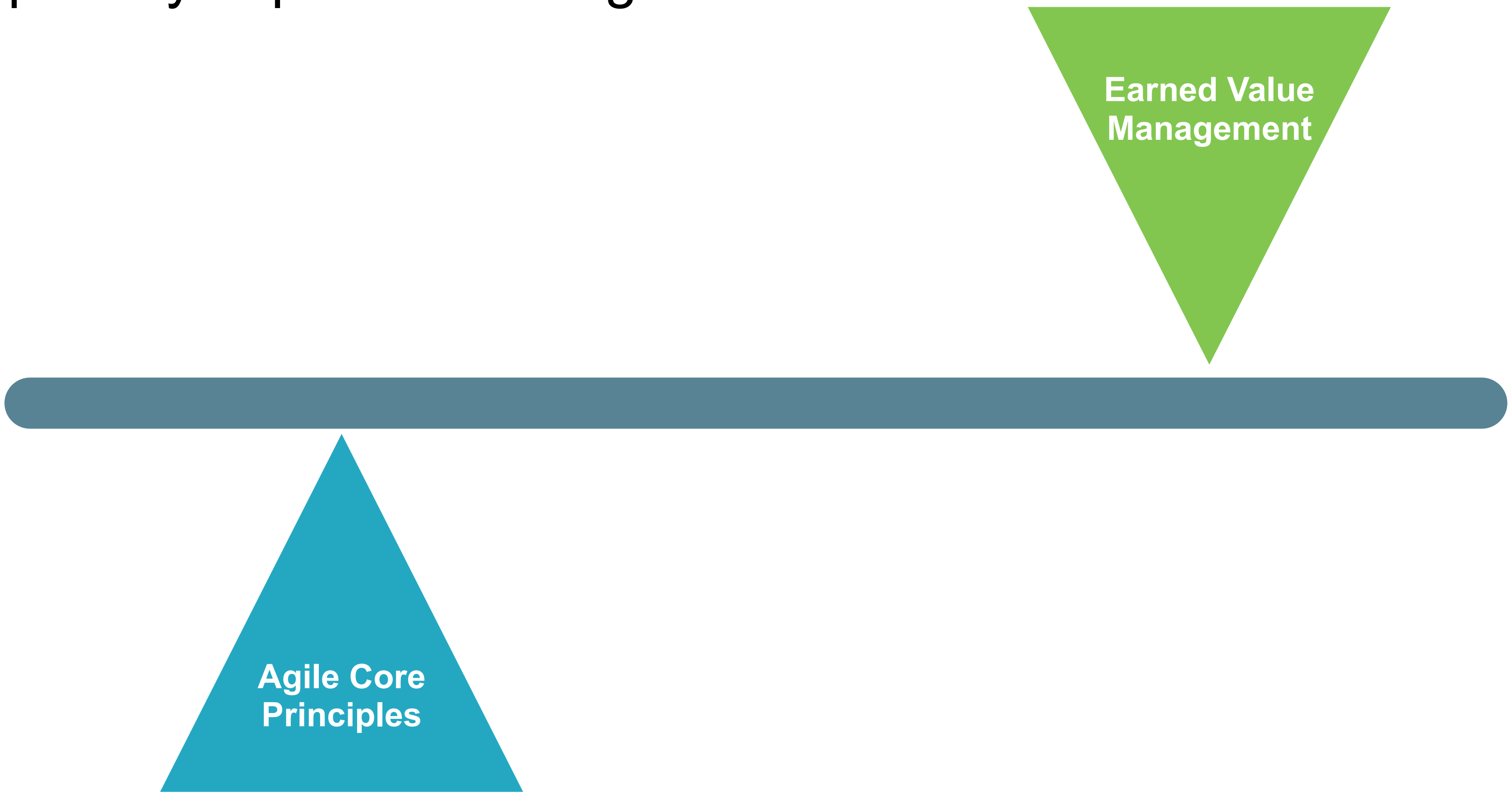
EVM and Agile

An interesting arranged marriage

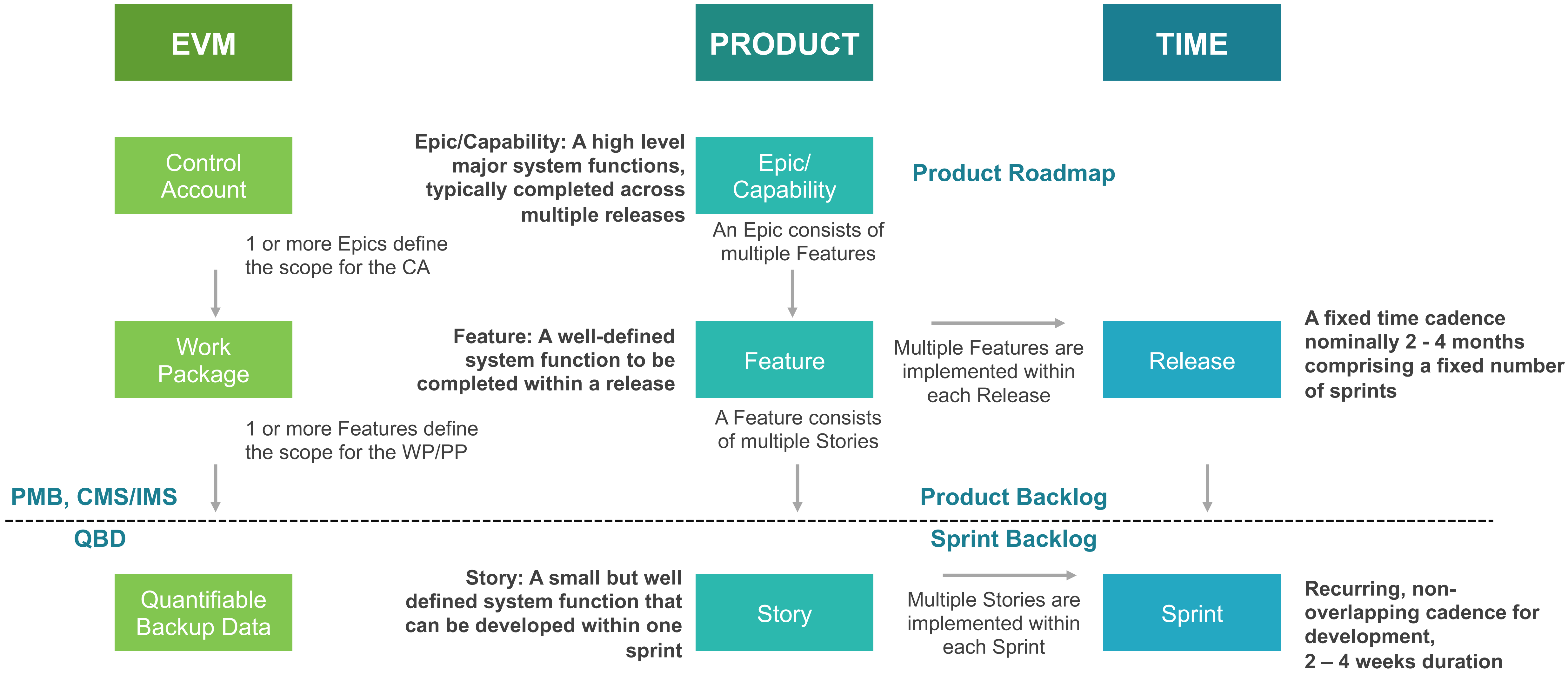
Agile EVM is an adapted implementation of EVM that uses the Scrum framework artifacts as inputs, uses traditional EVM calculations, and is expressed in traditional EVM metrics.

Balancing EVM and Agile

The use of Agile on a program or parts of a program does not need to be in conflict with Earned Value Management, they can be complementary when carefully aligned and appropriately implemented together.



AGILE – EVM Relationships



EVM

PRODUCT

TIME

Control Account

Epic/Capability: A high level major system functions, typically completed across multiple releases

Epic/ Capability

Product Roadmap

1 or more Epics define the scope for the CA

An Epic consists of multiple Features

Work Package

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

Feature

Multiple Features are implemented within each Release

Release

A fixed time cadence nominally 2 - 4 months comprising a fixed number of sprints

1 or more Features define the scope for the WP/PP

A Feature consists of multiple Stories

PMB, CMS/IMS

Product Backlog

QBD

Sprint Backlog

Quantifiable Backup Data

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

Story

Multiple Stories are implemented within each Sprint

Sprint

Recurring, non-overlapping cadence for development,
2 – 4 weeks duration

AGILE EVM - Definitions

Epic/ Capability

A high level system functionality defined by the customer to meet a specific requirement/need.
All Epics/Capabilities should have clearly defined technical completion criteria.
Epics/Capabilities are found at the Control Account level of the WBS and are usually composed of multiple Features

Feature

Discrete part of system functionality to meet the delivery a Capability.
All Features should have clearly defined objective technical completion criteria.
Features are typically found at the Work Package level of the WBS and can typically be completed in a single Release

Release

Term used to describe a concrete time box or cadence used to complete Features. Release duration can vary, but is typically two to four months.
It's useful to synchronize the release cadence with the detailed rolling wave planning horizon.

Story

Term used to describe activities that contribute to the completion of a Feature and can be completed within a single Sprint

Sprint/Iteration

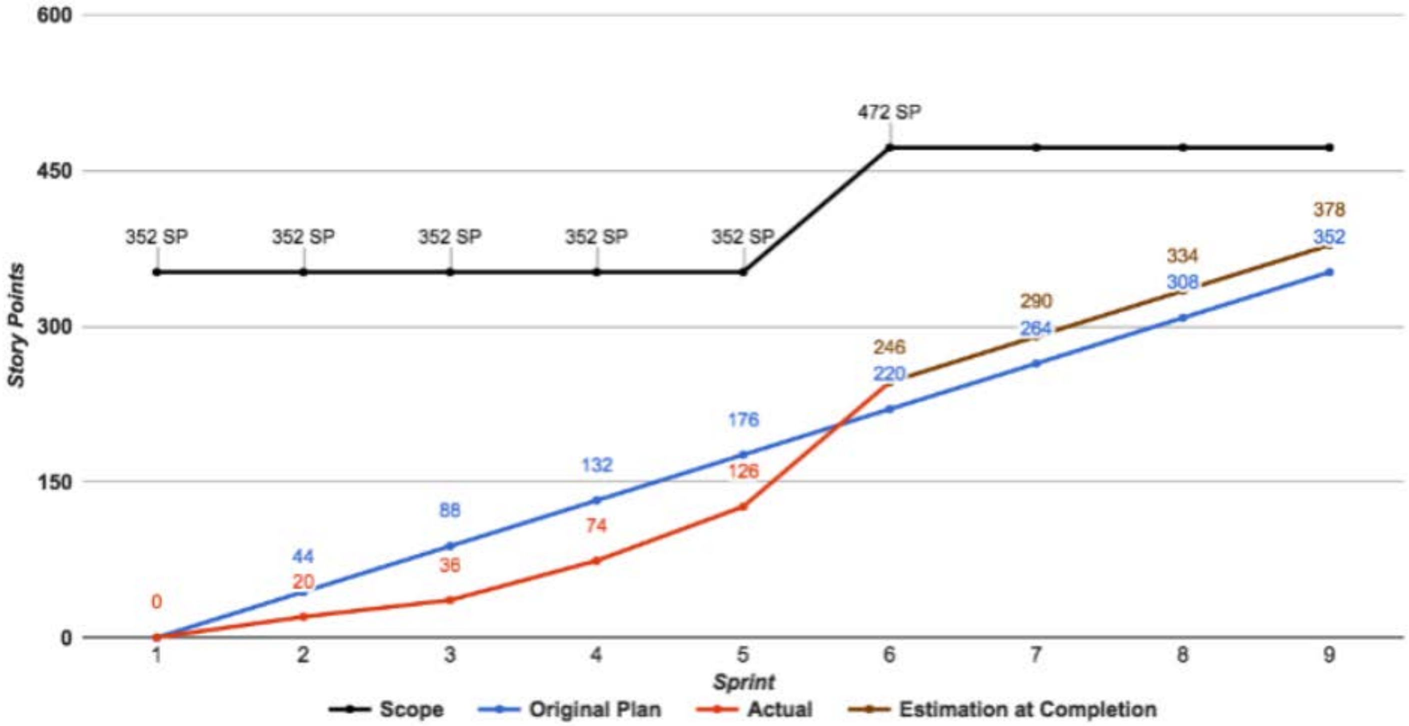
A concrete time box or cadence used to complete Stories. Sprint duration can vary, but is typically two to four weeks

Velocity

A measure of the amount of work a team can complete within a Sprint.

Agile Burn up chart

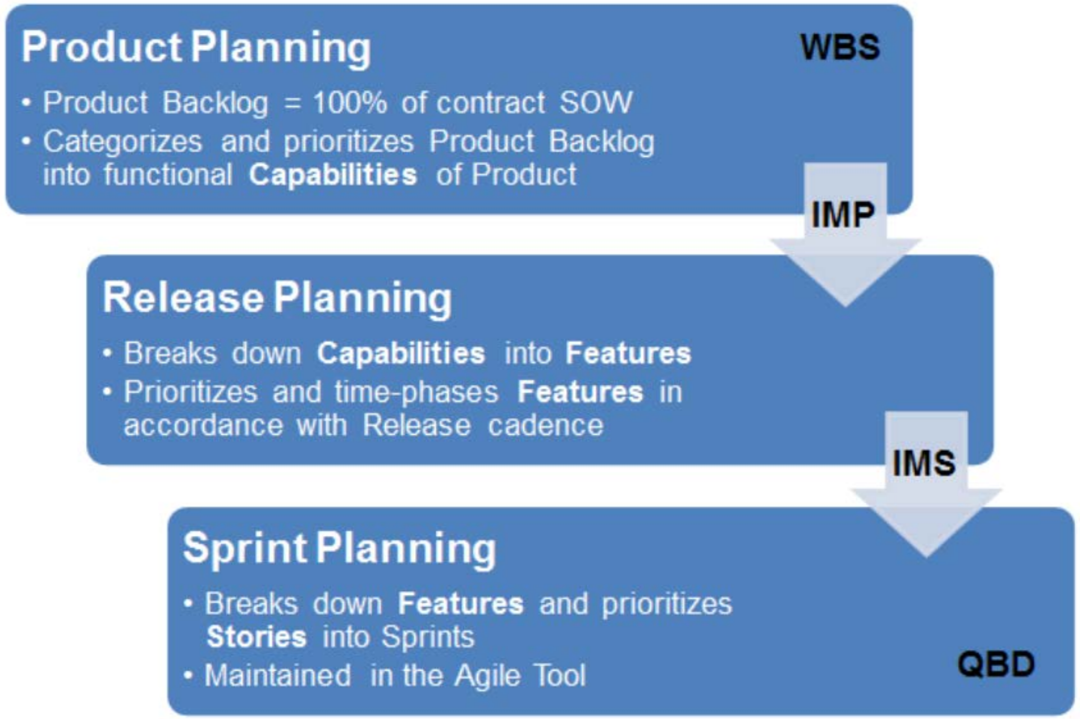
The Agile burn-up chart closely resembles the BCWS and BCWP plots of an EVM graph.
Original planned story point completion is analogous to BCWS, and the actual story point completions are analogous to BCWP.



Agile EVM - Planning

Agile EVM program planning is driven by the prioritization of functional capabilities defined by the customer, and is performed by the development team with customer involvement and approval.

Planning is broken down into product, release, and sprint planning, and is aligned with the Performance Measurement Baseline (PMB) establishment and maintenance processes within the EVMS.



Product Planning WBS

- Product Backlog = 100% of contract SOW
- Categorizes and prioritizes Product Backlog into functional **Capabilities** of Product

IMP

Release Planning

- Breaks down **Capabilities** into **Features**
- Prioritizes and time-phases **Features** in accordance with Release cadence

IMS

Sprint Planning

- Breaks down **Features** and prioritizes **Stories** into Sprints
- Maintained in the Agile Tool

QBD

Agile EVM - Planning

Product Planning

Begins with contract award and is the basis of the contract work breakdown structure (WBS) and initial integrated master plan (IMP).

Product planning requires the product owner and customer representatives to specify and prioritize the initial capabilities needed to deliver the contractually required system, resulting in the product backlog (100% of scope).

These capabilities are grouped into builds or capability releases to develop the product roadmap.

Agile EVM - Planning

Release Planning

Maps the product backlog with its capabilities to features and candidate stories to be delivered with each sprint, based on customer priority.

This planning defines what the product does, and when the functionality is delivered through the development of its time phased features.

This process results in completion of the initial schedule (CMS/IMS).
Release planning continues throughout the life of the program and is executed in accordance with program freeze period, and through the rolling wave process in the schedule.

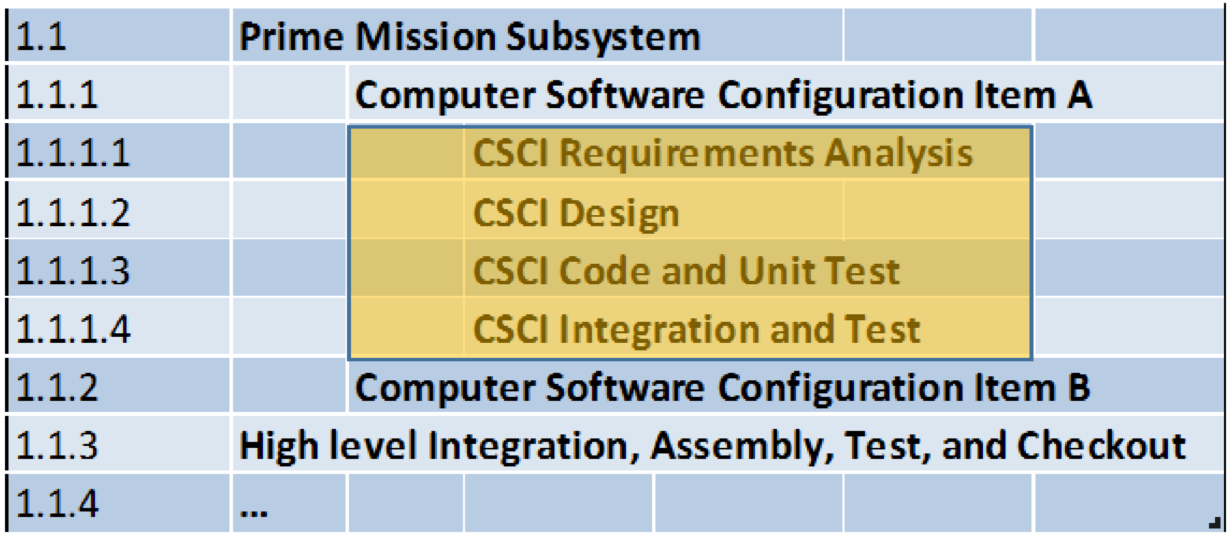
Agile EVM - Planning

Sprint Planning

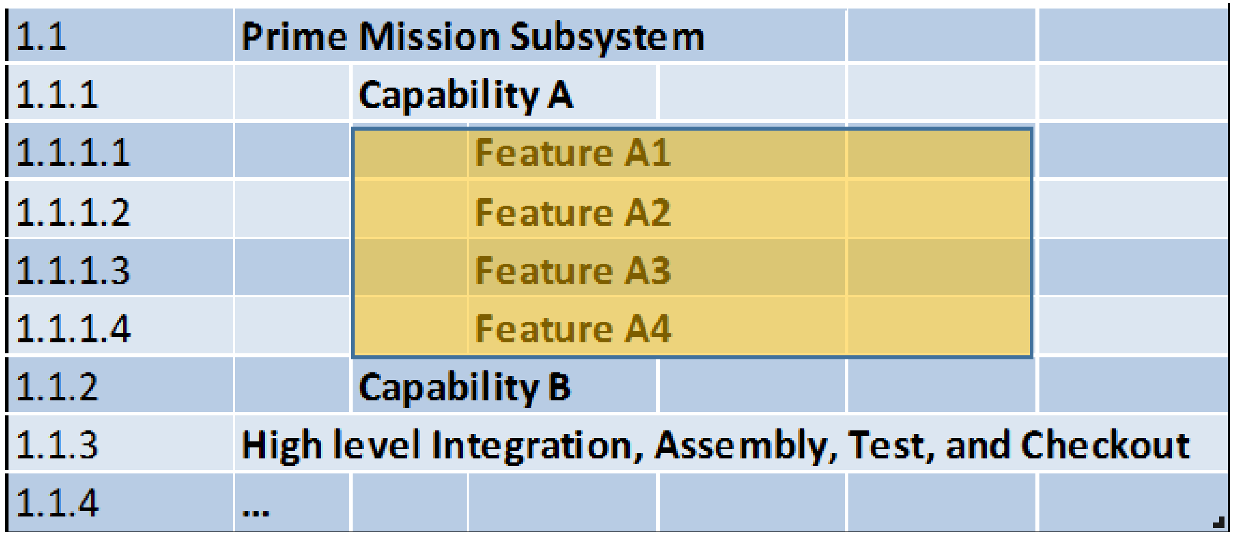
Is the collaboration between the team and the product owner to determine which features and stories will be developed in the coming sprint.

Sprints are a fixed time box, based on the program’s cadence, and are independent of scope.

Agile EVM - Work Breakdown Structure



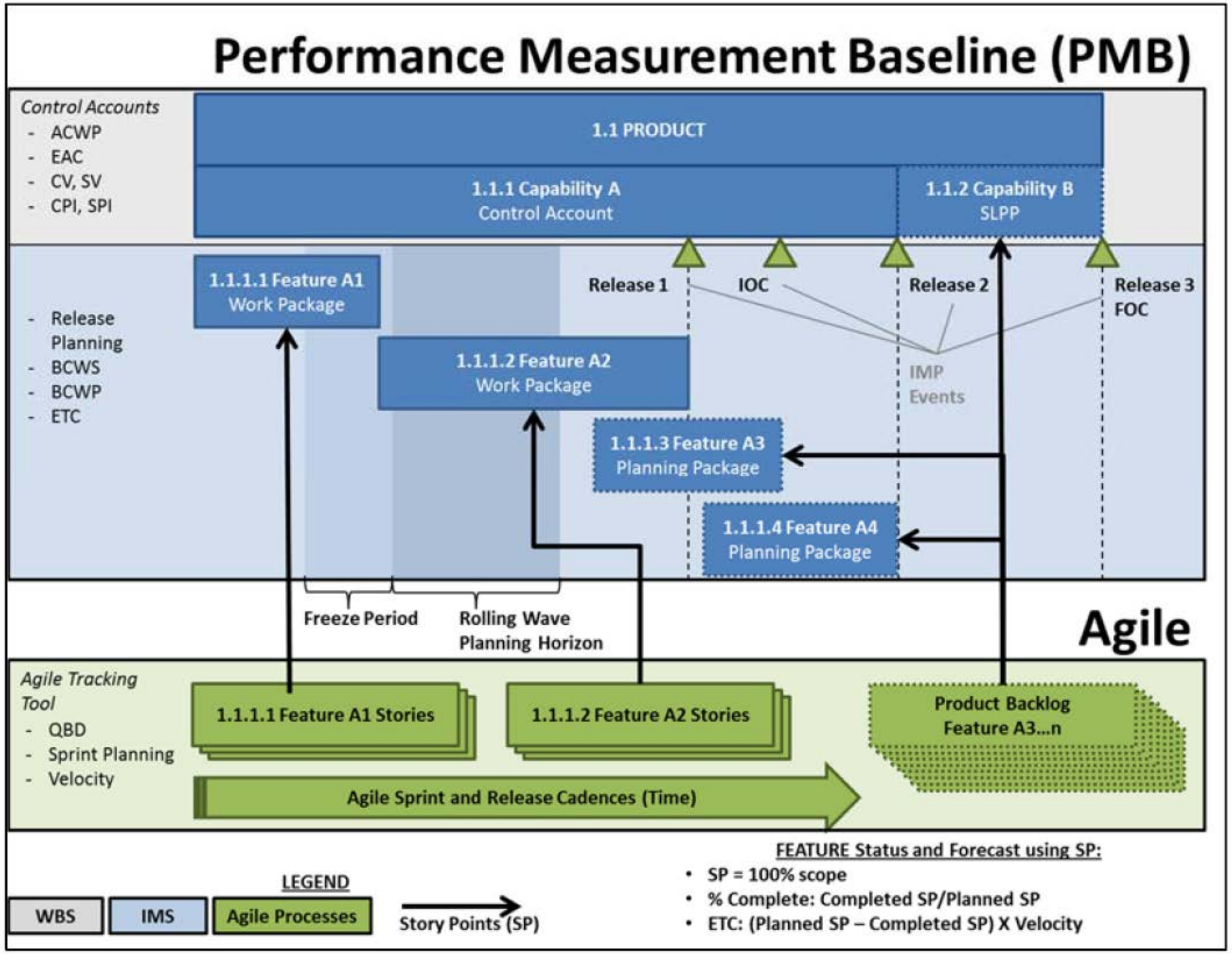
1.1	Prime Mission Subsystem		
1.1.1		Computer Software Configuration Item A	
1.1.1.1			CSCI Requirements Analysis
1.1.1.2			CSCI Design
1.1.1.3			CSCI Code and Unit Test
1.1.1.4			CSCI Integration and Test
1.1.2		Computer Software Configuration Item B	
1.1.3	High level Integration, Assembly, Test, and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



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

Agile EVM - Scheduling

The Schedule (CMS/IMS) is established as part of the Agile release planning and involves the identification and prioritization of stories within features within a capability release.



FEATURE Status and Forecast using SP:

- SP = 100% scope
- % Complete: Completed SP/Planned SP
- ETC: (Planned SP – Completed SP) X Velocity

Agile EVM - Scheduling

Features are detail planned for the current release.

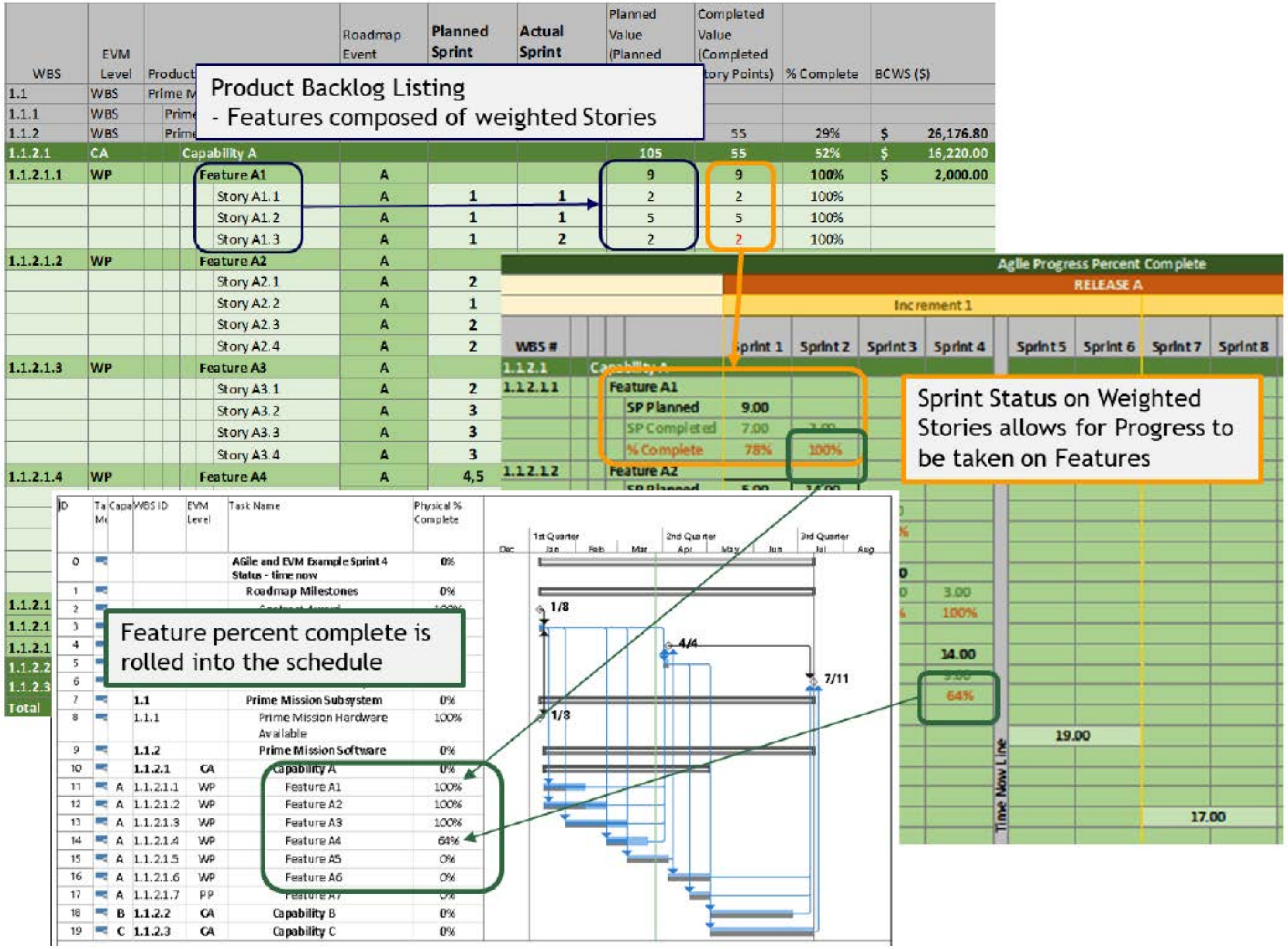
Subsequent releases may be developed using planning packages within the rolling wave planning process, by feature or by capability.

Features are sized and estimated to fit within release plans and represent significant pieces of capabilities with a clear acceptance criteria.

Stories account for 100% of the work for each associated feature. Stories are maintained by development teams outside of the schedule, within the Agile tool, providing a source of quantifiable backup data (QBD).

Features are the logical lowest level element included in the schedule necessary for visibility into program performance.

Agile EVM - Scheduling



Product Backlog Listing
- Features composed of weighted Stories

WBS	EVM Level	Product	Roadmap Event	Planned Sprint	Actual Sprint	Planned Value (Planned	Completed Value (Completed Story Points)	% Complete	BCWS ($)
1.1	WBS	Prime M							
1.1.1	WBS	Prime							
1.1.2	WBS	Prime					55	29%	$ 26,176.80
1.1.2.1	CA	Capability A				105	55	52%	$ 16,220.00
1.1.2.1.1	WP	Feature A1	A			9	9	100%	$ 2,000.00
		Story A1.1	A	1	1	2	2	100%	
		Story A1.2	A	1	1	5	5	100%	
		Story A1.3	A	1	2	2	2	100%	
1.1.2.1.2	WP	Feature A2	A						
		Story A2.1	A	2					
		Story A2.2	A	1					
		Story A2.3	A	2					
		Story A2.4	A	2					
1.1.2.1.3	WP	Feature A3	A						
		Story A3.1	A	2					
		Story A3.2	A	3					
		Story A3.3	A	3					
		Story A3.4	A	3					
1.1.2.1.4	WP	Feature A4	A	4,5					
1.1.2.1									
1.1.2.1									
1.1.2.1									
1.1.2.2									
1.1.2.3									
Total									

Agile Progress Percent Complete

RELEASE A

Increment 1

WBS #		Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8
1.1.2.1	Capability A								
1.1.2.1.1	Feature A1								
	SP Planned	9.00							
	SP Completed	7.00	2.00						
	% Complete	78%	100%						
1.1.2.1.2	Feature A2								
	SP Planned	5.00	14.00						

Sprint Status on Weighted Stories allows for Progress to be taken on Features

ID	Task Mode	Capability	WBS ID	EVM Level	Task Name	Physical % Complete
0					AGile and EVM Example Sprint 4 Status - time now	0%
1					Roadmap Milestones	0%
2					Contract Award	100%
3						
4						
5						
6						
7			1.1		Prime Mission Subsystem	0%
8			1.1.1		Prime Mission Hardware Available	100%
9			1.1.2		Prime Mission Software	0%
10			1.1.2.1	CA	Capability A	0%
11		A	1.1.2.1.1	WP	Feature A1	100%
12		A	1.1.2.1.2	WP	Feature A2	100%
13		A	1.1.2.1.3	WP	Feature A3	100%
14		A	1.1.2.1.4	WP	Feature A4	64%
15		A	1.1.2.1.5	WP	Feature A5	0%
16		A	1.1.2.1.6	WP	Feature A6	0%
17		A	1.1.2.1.7	PP	Feature A7	0%
18		B	1.1.2.2	CA	Capability B	0%
19		C	1.1.2.3	CA	Capability C	0%

Feature percent complete is rolled into the schedule

EV methods and techniques

It is imperative that progress/performance is tied to the completion of scope (technical progress) and not the completion of time boxed events such as releases or sprints.

Stories are used to measure progress and calculate Earned Value performance as quantifiable backup data (QBD) representing technical accomplishment towards a Feature.

The criteria is defined and documented in the Agile Tool, similar to the utilization of an MRP system to provide backup data.

The relationship between the QBD in the Agile tool and the schedule is documented and evidenced through unique Story/Feature/Capability coding.

Variance Analysis

Within the EVMS, variance analysis is conducted at work package and control account level.

Variance analysis may utilize Agile metrics and reports such as burn up, burn down, and velocity reports, as necessary to provide additional insight into program variances.

The sprint retrospective conducted at the completion of each sprint will be the richest source of supporting information for the root cause analysis and corrective actions within the variance analysis process.

Estimate at Completion

Velocity and other performance data generated from the Agile Tool are used to establish a forecast for Estimate to Complete (ETC).
The formula for determining Feature Remaining Effort in hours is based on Story Points (SP):

- Feature ETC = (Total Currently Forecasted SP – Total Completed SP) x (Total Hours to Date / Total Completed SP)

  Or:

- Feature ETC = (Total Currently Forecasted SP – Total Completed SP) x Velocity

Feature ETCs are summarized to the Capability level and compared to ACWP to determine Control Account EAC.

Agile and EVMS interfaces

Agile Tool	EVMS	Mapping
Capability	Control Account	Maintained in the Agile tool and the Contract WBS
Feature	Work Package/Planning Package	Maintained in the Agile tool, in QBD documents, and in the program schedule, coded to Capability
Feature Story Points	QBD	Summary of Story Points maintained in Agile tool and in QBD file, coded to Features

Questions

Questions

- Can I have sprints and releases in my CMS/IMS?
    - Sprints and releases are discouraged. They are simply time boxes – there is no value in scheduling them as they will occur independent as to whether or not the scope is completed. Project schedules plan and execute the scope, not time. If sprints and releases are shown in the schedule, they should be at the top for informational purposes and must not have logic that feeds into the capabilities, features or critical path.
- What Earned Value Technique is most appropriate for Agile?
    - Agile development is a discrete effort and the most objective EVT is % complete with quantifiable backup data (QBD) or 0/100.
    - EV should reflect the technical accomplishment. Using complete stories within a feature are the ideal QBD.

Questions

- If you delay completion of a feature – will that delay the start of the next release?
  - No. If a feature is delayed or does not complete on time, it will show a late finish in the schedule and will probably then occur during the next release. Since the releases are fixed time boxes, the release would still end on the predetermined date. There should be no logic on the release, features may be tied to delivery milestones or other scheduled tasks, which would show a schedule slip, however, the release start and finish date will remain constant.
- Do I need to plan all of my stories and their associated tasks in my CMS/IMS?
  - No. The schedule should extend down to the feature level and have any necessary relationships to milestones, deliverables or interdependencies. The stories and their associated tasks would exist within the Agile tool and be traceable through unique Story/Feature/Capability coding.

Further Reading

National Defence Industrial Association (NDIA)
An Industry Practice Guide for Agile on Earned Value Management programs

US Department of Defence
Agile and Earned Value management: A program manager's desk guide

Scrum Alliance.org

Scrum.org

Atlassian.com

Agilemanifesto.org

Thank You