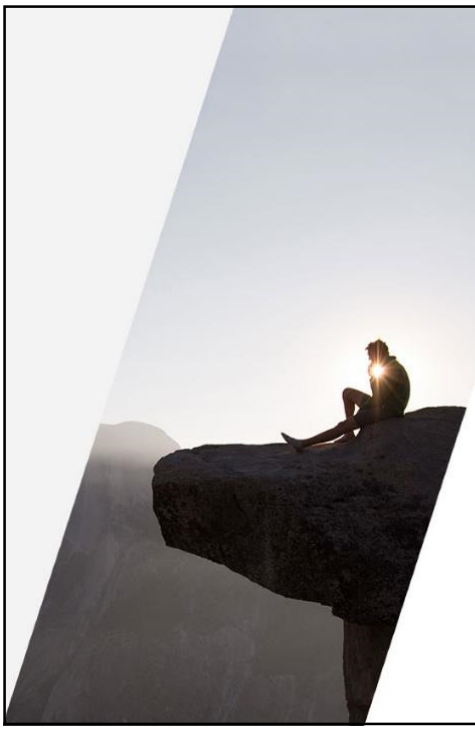
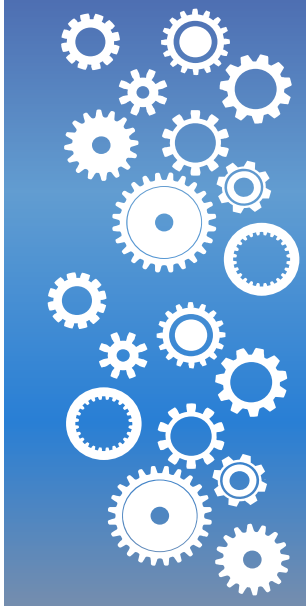


Let's re-examine….. How Agile is EVM?

Meri Duncanson
2021



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.

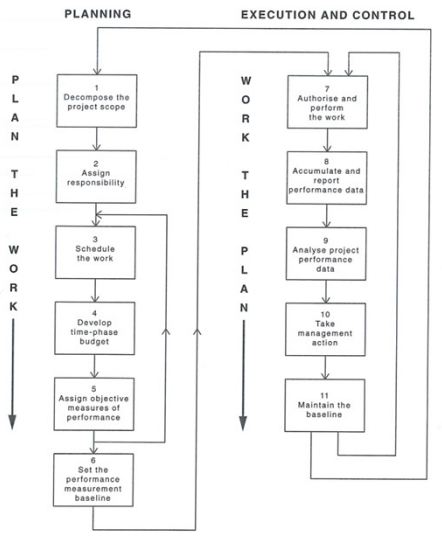
EVM helps provide the answers to the following questions

- 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?

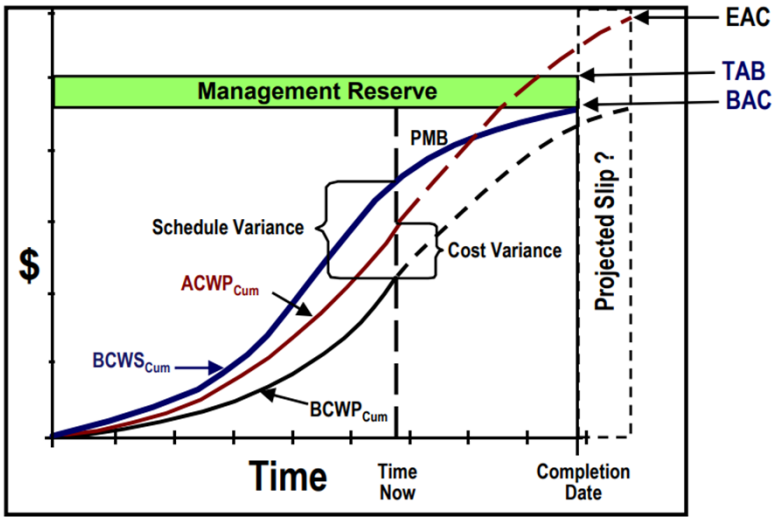
- What has happened - Why are we not on schedule/budget?
- What is happening - What are the impacts?
- What is going to happen - What are we doing to fix this?
- How much more will we have to spend to complete the entire scope of work?

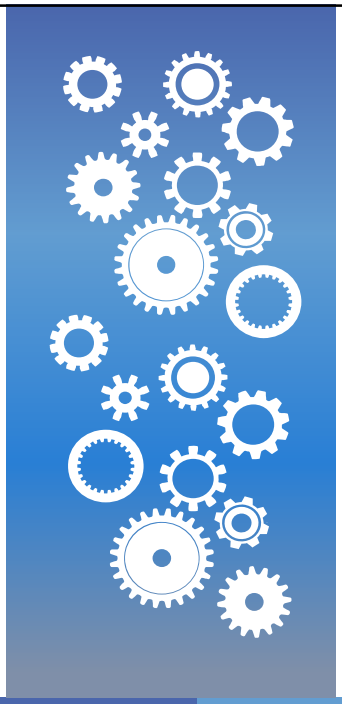
Earned Value Principles

Step 1: Decompose the project scope
Step 2: Assign responsibility
Step 3: Schedule the work
Step 4: Develop time-phased budget
Step 5: Assign objective measures of performance
Step 6: Set the performance measurement baseline
Step 7: Authorise and perform the work
Step 8: Accumulate and report performance data
Step 9: Analyse performance data
Step 10: Take management action
Step 11: Maintain the baseline



EVM 'Gold Card'





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, established through an unlimited number of practices, tools and processes.

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

Agile Manifesto

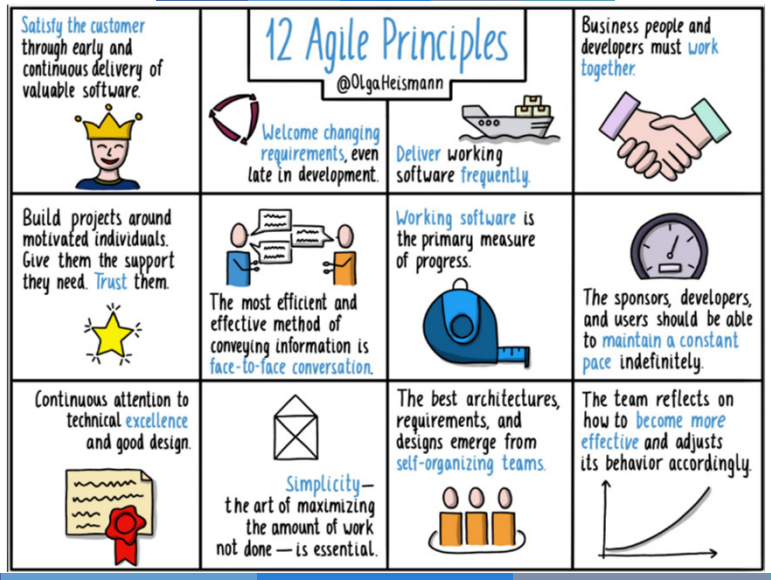
Individuals and interactions over processes and tools
Processes and tools by their nature are less responsible to change and customer needs

Working software over comprehensive documentation
Agile is about streamlining, not eliminating documentation

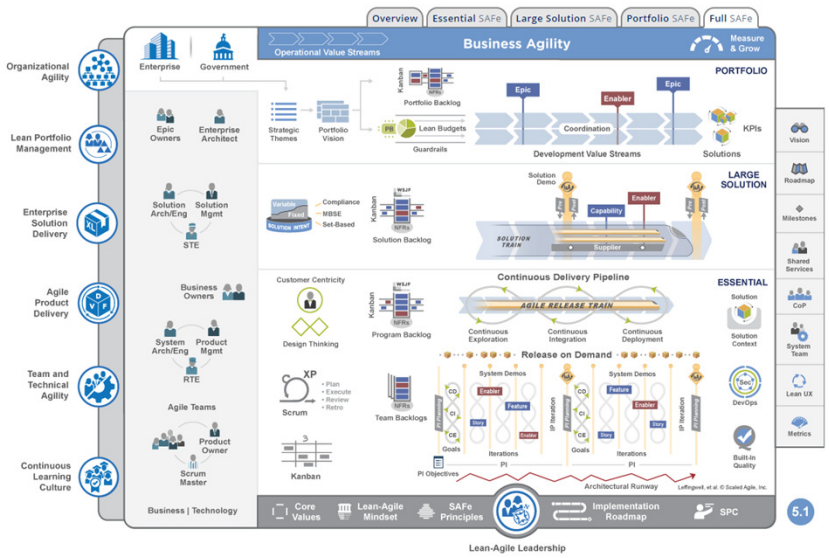
Customer collaboration over contract negotiation
Agile includes the customer throughout the lifecycle of the project and values their feedback

Responding to change over following a plan
Agile works in short iterations [sprints] to embrace change

Twelve Principles of Agile



Scaled Agile Framework - SAFe



Agile and EVM

The capabilities of Agile and Earned Value Management can be symbiotic, as long as there are accommodations made in the application of both methodologies.

Agile provides a high degree of flexibility and rapid development, but earned value management techniques can be applied to the Agile framework in order to mitigate scope creep, ensure quality requirements are met, and provide project status and progress to key stakeholders.

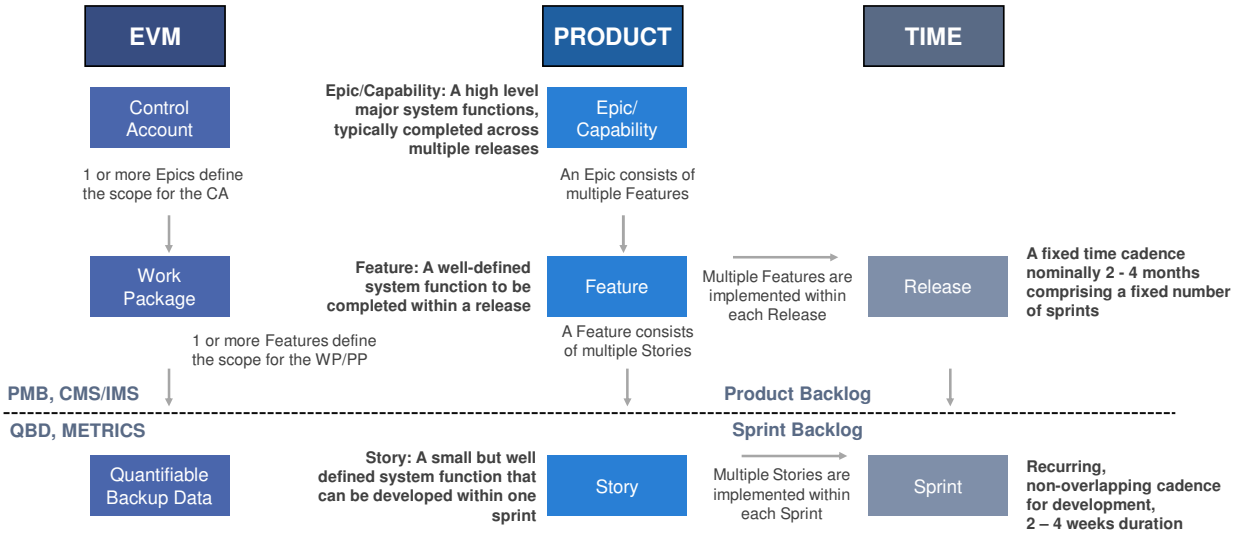
Implementing Earned Value Management on Agile Projects – Kyle Manship - University of Oregon - 2018

Agile EVM

An interesting arranged marriage

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

Agile – EVM Relationships



Agile EVM - Definitions

Epic/ Capability

A high level system functionality defined by the customer to meet a specific requirement/need.
All Epics/Capabilities 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 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 a small piece of desired functionality, written in the user's language, that contribute to the completion of a Feature and can be completed within a single Sprint. It should be clear, feasible and testable.

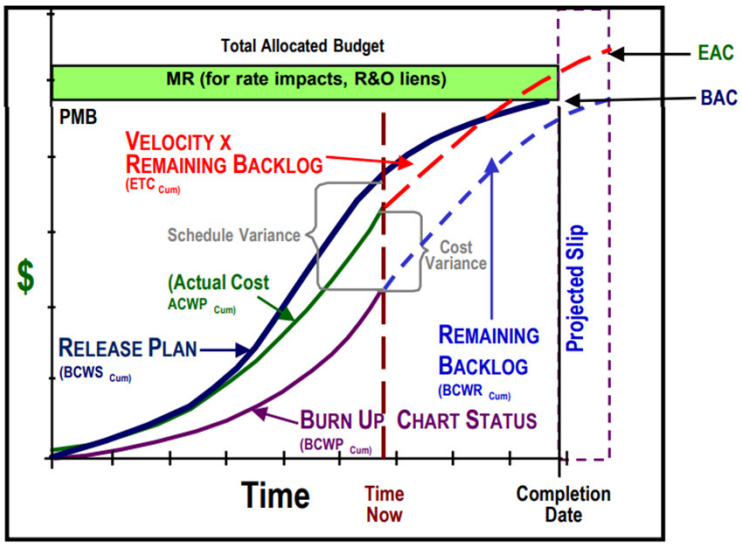
Sprint/Iteration

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.

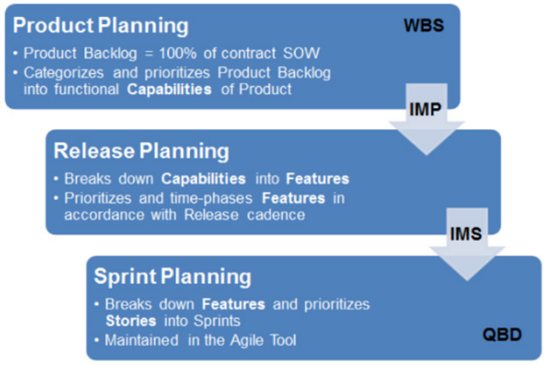
Agile EVM 'Platinum Card'



Agile EVM - Planning

Agile EVM program planning is driven by the prioritisation of functional capabilities defined by the customer or product owner, 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.



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 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

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 - WBS

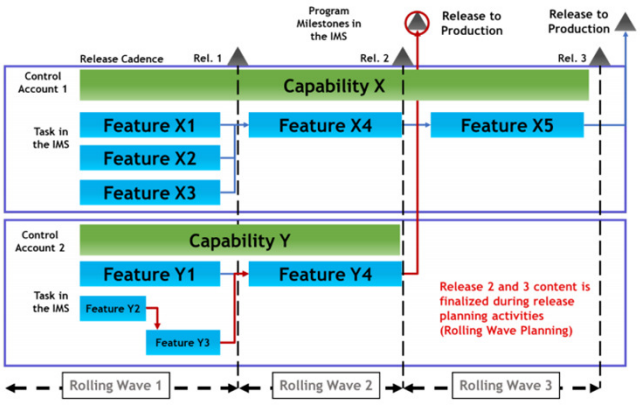
1.1	Prime Mission Subsystem
1.1.1	Computer Software Configuration Item A
1.1.1.1	CSCI Requirements Analysis
1.1.1.2	CSCI Design
1.1.1.3	CSCI Code and Unit Test
1.1.1.4	CSCI Integration and Test
1.1.2	Computer Software Configuration Item B
1.1.3	High level Integration, Assembly, Test, and Checkout
1.1.4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

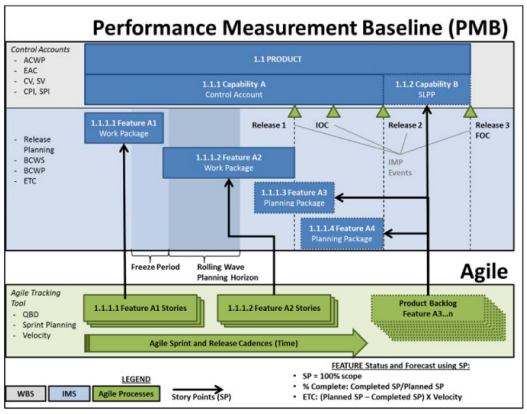
1.1	Prime Mission Subsystem
1.1.1	Capability A
1.1.1.1	Feature A1
1.1.1.2	Feature A2
1.1.1.3	Feature A3
1.1.1.4	Feature A4
1.1.2	Capability B
1.1.3	High level Integration, Assembly, Test, and Checkout
1.1.4	...

Agile EVM - Scheduling

The schedule (CMS/IMS) is established as part of the Agile release planning process.

It involves the identification and prioritisation of stories within features within a capability for each release.

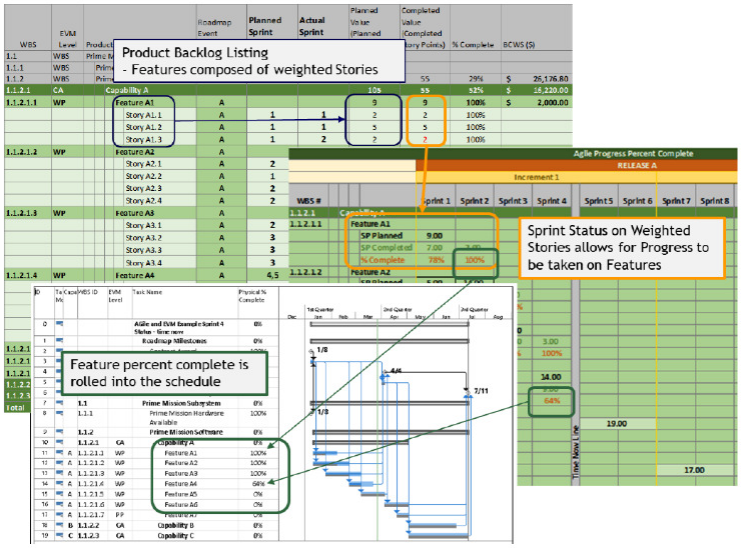




Agile EVM - Scheduling

- 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.
- The scope within the 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



Agile EVM - EV 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.

Agile EVM - Variance Analysis

- Within the EVMS, variance analysis is conducted at work package and control account level.
- Variance analysis may utilise 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 standard variance analysis process.

Agile EVM - 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

ETCs at the Feature level are summarised to the Capability level and compared to Actual Costs (ACWP) to determine Control Account Estimate at Completion (EAC).

Agile and EVMS Interfaces

Agile	EVM	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

Prepared Questions

Should I have sprints in my CMS/IMS?

- Sprints 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.

Prepared 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, without the Feature included.
  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.

Prepared Questions

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.

Prepared Questions

What Earned Value Technique is most appropriate for Agile?

- Agile development is a discrete effort and the most objective EVT is the 0/100 technique, or % complete with Quantifiable Backup Data (QBD).
- EV should reflect the technical accomplishment.

Prepared Questions

The team will need to complete additional stories in order to meet the completion criteria of a particular feature, is this a scope change?

- Not if the additional stories are still consistent with the acceptance criteria of the feature, and simply provide grater granularity to how work will be performed, then this is not a scope change.

Prepared Questions

The team will be able to meet all the objective technical completion criteria for a particular feature without having to complete all the planned stories, its this a scope change?

- No – this is a change in efficiency, and would likely show as a positive schedule variance, and if less hours expended – a positive cost variance may also occur.

Prepared Questions

An Agile EVM project does not have the ability to support or undertake an Integrated Baseline Review (IBR).

- Not necessarily true…. The IBR confirms that "performance measurement baseline covers the entire scope of work, the work is realistically and accurately scheduled, the proper amount and mix of resources have been assigned to tasks, and proper objective indicators have been selected for measurement of task accomplishment."
- This can be achieved using a mixture of EVM and Agile artefacts and processes. For example:
  - *Ensure the program has captured all the customer requirements, including an understanding of the operational concept* – the following artefacts would evidenced; SOW, WBS/Dictionary, IMP, Product Backlog
  - *Ensure the organisation structure is appropriate for the program requirements and the staffing plan is credible* – Control Account Plans, RAM, RACI, Staffing Plan, Agile team definition.

Prepared Questions

Is the (Agile) Roadmap the same as the (EVM) Integrated Master Plan (IMP)?

- The IMP is an event-based, top-level plan consisting of a hierarchy of Program Events. Each event is decomposed into specific accomplishments and each specific accomplishment is decomposed into specific Criteria.
- Agile roadmap's main function is to establish a long-term, high-level overview of the product's expected evolution.
- The Agile Roadmap is an element of the IMP, however the IMP includes ALL contracted scope, from contract award to contract completion.

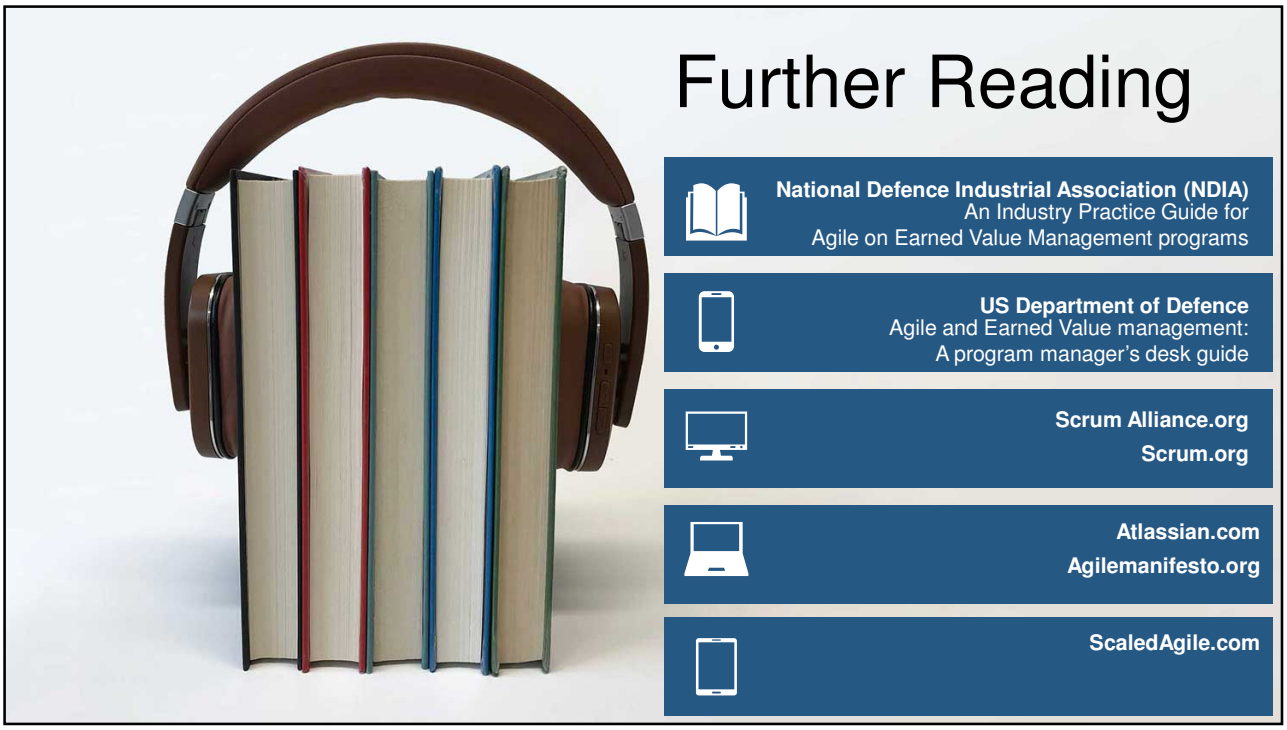
Prepared Questions

How does Agile and EVM work together – Agile is all about changing scope to keep the customer happy ?

- Agile EVM necessitates active management, diligence, and oversight as required.
- It involves using controlled change processes for removing or adding Features, and involving the customer in the process. This may cause a contract change or can be used to carefully 'trade' Features
- Adding or removing stories 'just' causes schedule variance – not a baseline change

Prepared Questions

- How does a company implement two competing 'cultures'
  - EVM gives Agile the traditional governance and assurance that senior management and executives are looking for.
  - Agile gives EVM 'street cred' and customer centric focus.


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
ScaledAgile.com


Additional
Information

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 - Definitions

BCWS / PV

How much of the work <u>should</u> 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 <u>is</u> complete?

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

ACWP / AC

How much did it <u>cost</u> to do the work?

The actual amount spent to accomplish the work.

Earned Value - Definitions

BAC

What is the total job supposed to cost?

The total approved budget allocated to complete the project

ETC

How much of the work remaining

The cost required to complete the remaining work.

EAC = ACWP + ETC

EAC

What do we now expect the total job to cost?

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

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)

AGILE

Agile is is an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organising 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 - Definitions

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 prioritized list of all the Features that need to be built, determining the total scope of the work for the project/release.

Agile - Definitions

Sprint Backlog

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

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

Release

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

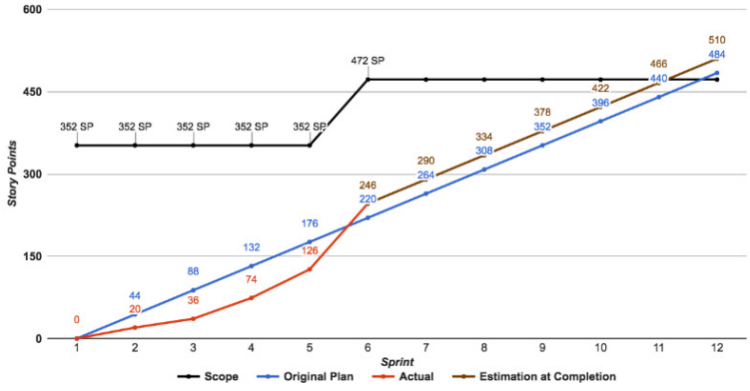
Velocity

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 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.



4 AGILE VALUES

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

INDIVIDUALS & INTERACTIONS Over Processes and Tools.

WORKING SOFTWARE Over Comprehensive Documentation.

CUSTOMER COLLABORATION Over Contract Negotiation.

RESPONDING TO CHANGE Over Following a Plan.

That is, while there is value in the items on the right, we value the items on the left more.

12 AGILE PRINCIPLES

1. Our highest priority is to **SATISFY THE CUSTOMER** through early and continuous delivery of valuable solutions.

2. **WELCOME CHANGING REQUIREMENTS**, even late in development. Agile processes harness change for the customer's competitive advantage.

3. **DELIVER WORKING SOLUTIONS FREQUENTLY**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must **WORK TOGETHER** daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and **SUPPORT** they need, **AND TRUST** them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is **FACE-TO-FACE CONVERSATION.**

7. **WORKING SOLUTIONS** are the primary measure of progress.

8. Agile processes promote **SUSTAINABLE DEVELOPMENT.** The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to **TECHNICAL EXCELLENCE** and good design enhances agility.

10. **SIMPLICITY**--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from **SELF-ORGANIZING TEAMS.**

12. At regular intervals, the team **REFLECTS** on how to become more effective, then tunes **AND ADJUSTS** its behavior accordingly.

4 SCRUM ROLES

SCRUM TEAM
The Scrum Team is 10 or fewer people including one Product Owner, one Scrum Master and Developers.

PRODUCT OWNER
A single decision-maker who is responsible for prioritizing the backlog and maximizing the value delivered by the Scrum Team.

DEVELOPERS
Cross-functional team of 3-9 people who plan, adapt and hold each other accountable to deliver a usable increment each sprint.

SCRUM MASTER
Servant Leader, coach & trainer who supports the Scrum Team, Product Owner and Org to adopt Scrum as defined in the Scrum Guide.

5 SCRUM EVENTS

THE SPRINT

PURPOSE – The sprint is a fixed length event up to a month in length where all the work is completed to turn ideas into value. Considered the heartbeat of Scrum.

1. Every Sprint should be the **SAME LENGTH.**
2. As soon as one Sprint ends, the **NEXT SPRINT BEGINS.**
3. The Sprint is a container for all the other **SCRUM EVENTS.**

SPRINT PLANNING

PURPOSE - Understand WHY the Sprint is valuable, WHAT prioritized items the team will work on and HOW the team will complete the work.

1. The **SCRUM TEAM IS RESPONSIBLE** for planning each backlog item and taking on a realistic amount of work based on their capacity and past performance.
2. The Scrum Team **PLANS THE WORK TOGETHER** with the goal of **COMPLETING the work TOGETHER.**
3. The Sprint Goal, selected Backlog Items and plan for delivering them is called the **SPRINT BACKLOG.**

DAILY SCRUM

PURPOSE – Inspect progress toward the sprint goal, coordinate efforts and adapt plans.

1. The Daily Scrum is for the **DEVELOPERS** to improve communication and decision-making.
2. The format can vary but the **FOCUS** is on hitting the **SPRINT GOALS.**
3. The meeting should last **LESS THAN 15 MINUTES** and be held at the same time and place every working day of the sprint.

SPRINT REVIEW

PURPOSE - Demonstrate progress, inspect the team results and get feedback for future adaptations.

1. The Scrum Team should show actual working **RESULTS FROM THE USER'S PERSPECTIVE.** Don't show lines of code or PowerPoint.
2. **GET ORGANIZED**, start on time and be succinct. Plan for 4-5 minutes per completed backlog item and leave time for engagement with stakeholders.
3. **EXPECT FEEDBACK** including new requests.

RETROSPECTIVE

PURPOSE - Allow the Scrum team to pause, reflect and play ways to improve team quality and effectiveness.

1. Retrospectives are **FOR THE SCRUM TEAM** and the team decides who should attend.
2. **NO BLAMING OR COMPLAINING.** Assume that everyone did the best they could under the circumstances.
3. Use **ROOT CAUSE ANALYSIS** to go beneath the surface. Select just one or two improvement actions each sprint.