

# **How to Implement Earned Value Management and Agile in Large Projects**

Bonus Chapter to the Book:

**Earned Value Management – Fast Start Guide**



**ROLAND WANNER**

**Author Contact:**

Roland Wanner

E-Mail: [info@rolandwanner.com](mailto:info@rolandwanner.com)

Internet: [www.rolandwanner.com](http://www.rolandwanner.com)



**Disclaimer**

This publication is designed to provide competent and reliable information regarding the subject matter covered. However, it is not intended as a substitute for legal or other professional services. Readers are urged to consult a variety of sources and professional expert assistance. While every effort has been made to make this book accurate, it may contain typographical and content errors.

For any questions or suggestions, please contact: [info@rolandwanner.com](mailto:info@rolandwanner.com)

The information expressed herein is the opinion of the author and is not intended to reflect upon any particular person or company. The author and publisher shall have no responsibility or liability with respect to any losses or damage caused, or alleged to be caused, by the information or application of the information contained in this book.

Please note that software and hardware names used in the book as well as brand names and product names, e.g. PMBOK® Guide, of the respective companies are generally subject to trademark, brand or patent protection.

**Copyright © 2020 Roland Wanner**

All rights reserved, including those of partial reproduction as well as photomechanical and electronic reproduction.



# **Content**

Introduction .....	4
Why EVM for Agile Software Projects? .....	4
Agile Projects for the U.S. Government .....	5
Agile EVM Delivers the Required Transparency .....	6
Agile Program Planning.....	6
Product Planning.....	7
Release Planning.....	8
Sprint Planning .....	8
Daily Planning – The Daily Scrum .....	9
The Agile EVM Performance Measurement Baseline (PMB).....	11
Agile Product and Time Hierarchy Aligned with EVM .....	12
The Product Backlog.....	14
The Integrated Master Plan (IMP) .....	15
Integrated Master Schedule (IMS) .....	16
Freeze Period Considerations.....	17
Control Account Plan.....	17
Aligning Agile Progress Metrics with Earned Value Reporting Levels .....	18
How to Analyze & Report Earned Value Performance .....	20
Consider Feature Cost/Risk When Establishing the Baseline .....	21
How to Use Agile Metrics to Forecast ETC/EAC.....	21
Further Details .....	23
References and Further Reading.....	23
Earned Value Management – Fast Start Guide.....	24

# Introduction

---

With agile projects not only mobile apps or application software are developed—agile projects are more and more embedded into larger and complex development projects and programs in the industry where software is needed.

To deliver software intensive systems more quickly and affordably is constantly growing—not only in the private industry but also in the defense industry. This led to an increased focus on capability-based planning and iterative product development with the goal to deliver the highest priority system features to the stakeholders as quickly as possible.

Agile has emerged as the leading industry software development methodology and has also seen growing adoption across the DoD and other federal agencies. Agile implements the needed method by focusing on small, frequent capability releases. It delivers working software while responding rapidly to changes in operations, technology, and budgets, and actively involving users throughout the development to ensure high operational value.

This document gives you an overview of how EVM can be applied to complex projects with agile developed software components. It is based on the NDIA "*An Industry Practice Guide for Agile on Earned Value Management Programs*".

## Why EVM for Agile Software Projects?

---

Many prominent representatives of agile software development believe that EVM has no place in software projects. Useless key figures that generate no value for the customer and only bureaucratic effort that Agile actually wants to avoid! Agile proponents assume these prerequisites:

*"In Agile, we start from the assumption that we cannot predict the future. While we know generally how much time scope and budget we have, we know*

*that it is waste to spend large amounts of time and resources building predictive plans. We always flex on scope, and we inspect and adapt to the learning of the team, customer, and business environment."*

"We find out what the user needs while we develop the software." That is a nice concept, but what do you say to your customer when they ask in a serious voice?

- How much will it cost?
- When will the project be ready?
- What features will I get until when?
- How confident are you with your answers?

Whether agile project or waterfall project, the customer does not have unlimited budget, time and patience. Therefore, you need an instrument with which you can give reliable answers.

Earned Value Management is already a critical success factor in many IT areas, e.g. ERP or system integration projects. By this I mean that you will have problems if you as a contractor cannot give a forecast with 80% certainty for projects over 1 million dollars. SOX or CobiT 4.0 also define "Program Performance Measurements" as necessary for IT areas.

## Agile Projects for the U.S. Government

---

IT projects of the American government often take several years and consume billions of dollars. In December 2010, the Office Management and Budget (OMB) published a 25-point plan to reform the government's information technology management.

Regarding IT programs, the following point is relevant: Government IT programs must be structured to deliver functionality in release cycles not longer than 12 months, ideally less than 6 months. The first usable functionality should be delivered no later than 18 months after program start. This means delivering results faster and reducing costs and technology risks.

# **Agile EVM Delivers the Required Transparency**

---

You often hear that it is too difficult and complex to apply EVM methods in agile projects or large and complex IT programs, and that EVM cannot deal so easily with changing requirements in agile projects. Many agile projects that have implemented a simple, customized EVM have already confirmed that this is not the case.

The central question in project monitoring is: "How do we measure our project progress in relation to the plan and what have we spent on the work done? When monitoring agile projects, the burn down or burn up chart plays a central role. However, these only monitor created features and/or user stories, i.e. technical progress. Agile metrics do not provide information about "Estimate at Completion" of a release or cost metrics that support the business when it comes to making decisions, such as when requirements in a release change. EVM, on the other hand, monitors costs, schedule, and technical performance—normally not on the agile level but on control account level.

Agile EVM requires only a few input parameters:

- The current costs of the project
- A release plan with the number of sprints
- An estimated product backlog
- The estimated development velocity

For all estimates at sprint level we use user story points as a measure of story size and velocity.

# **Agile Program Planning**

---

Work planning in Agile development is driven by prioritization of business value defined by the customer. This planning process focuses on the functionality or capabilities needed by the customer. Epics or

Capabilities contain product Features decomposed into User Stories. This decomposition allows for delivery of value to be tracked. The functionality or capabilities needed are listed on the Product Backlog. The Product Roadmap prioritizes the items on the Product Backlog based on business value and dependencies.

## Product Planning

Product Planning is a continuous control activity that encompasses the entire product goals of the program and establishes the Product Backlog and Product Roadmap. The Product Backlog is the master list of all functionality at the Epic and Feature level that is desired in the product and any other elements needed to produce the product, even if not in the final product.

The Product Roadmap prioritizes the items on the Product Backlog based on business value and dependencies by the Product Owner (PO) and Stakeholders. The Product Roadmap may precede, inform, or supplant the development of an IMP, and informs the top-level plan of the IMS. Due to its architectural significance, product planning may inform structural elements of the program such as the program Work Breakdown Structure (WBS).

Product planning is performed throughout the life of the program to refine and update the Product Backlog, refining Epics and Capabilities into Features and User Stories comprising those Epics and is informed through change control practices of the program. Typically, the Product Owner (PO), with Customer representatives, is responsible for managing the product planning activities. Program leadership assigns the PO who may also fill the role of a Control Account Manager (CAM)

Every Scrum team has a Product Owner (PO), representing the customer's value. In programs with multiple teams, there may also be a Chief Product Owner that coordinates across program POs and their teams. Program leadership assigns the PO who may also fill the role of a Control Account Manager (CAM).

## Release Planning

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

The Product Backlog and Product Roadmap identifying required product functionality are inputs to the Release Plan. Selected Features and their Stories define what the product must do and when the functionality will be delivered within the Release. It is within Release Planning and as part of the Integrated Master Schedule planning that Features are assigned to Work Packages and Planning Packages.

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

The CAM(s) use the output of Release Planning to detail plan the next increment of work or rolling wave within the PMB. Work Packages may align with individual Features or with logical groups of related Features, or—at times—Epics. The budget for each Work Package is allocated from the authorized budget for the planning package/control account in terms of hours and resources.

## Sprint Planning

The Release duration is expressed as a number of Sprints of equal length. Sprint Planning encompasses the product goals for the next Sprint, typically 2 to 4 weeks in duration.

Features selected from the release planning process are decomposed into Stories in preparation for the Sprint Planning. These Stories are prioritized by the Product Owner according to customer value, risk and other criteria.

In Sprint Planning the Product Owner and the Development Team determine what Features and Stories from the Product Backlog will be developed in the coming Sprint. The Development Team commits to the completion of specific Stories and confirm the criteria for work completion. The Scrum Master is responsible for facilitating Sprint Planning.

The Sprint Planning process is completed before any work starts on the Sprint. The Development team's list of Stories for the upcoming Sprint are placed in the Sprint Backlog.

During each Sprint, as Stories are completed, progress is determined by the completion status of the planned Stories for the Feature assigned to that Work Package.

## Daily Planning – The Daily Scrum

During the Daily Scrum standup meeting, the Development Team reviews development activities in the Sprint for alignment with the Sprint Plan to confirm progress of the Team's commitment to the Sprint goal and what impediments exist.

Product completion status is reviewed daily and updated to provide program performance measurement, based on the number of Story Points associated with completed Stories compared to the planned Story Points for the Sprint.

The tiered Agile planning levels are shown in the next figure. The hierarchy of the Planning Artifacts is described in more detail on page 15



Planning Level	Planning Frequency	Planning Horizon	Planning Precision	Planning Artifact	EVM Processes
<b>Product Planning</b>	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
<b>Release Planning</b>	Each Release	Release	Feature/Stories	Product Backlog Updates, Release Plan	IMS planning of Features to Work & Planning Packages. Networking them to Capabilities and Releases
<b>Sprint Planning</b>	Each Sprint	Weeks	Stories/Tasks	Sprint Backlog	Defining measures of effort and duration for Work & Planning Packages based on Release, Sprint, Story alignment to Features
<b>Daily Planning</b>	Daily	Day	Tasks	Updated Sprint Backlog	Update story status to determine EV for each Work Package

Figure 1: Agile planning levels and their relationship to EVM processes

# The Agile EVM Performance Measurement Baseline (PMB)

The Work Breakdown Structure (WBS) defines the program in terms of hierarchically-related, product-oriented elements. The WBS is a product-oriented family tree (composed of hardware, software, services, data, and facilities) that displays and defines the product to be developed during the acquisition. The WBS represents all scope and work being performed on a program, both level of effort (such as program management) and discrete deliverables.

For programs using Agile methodologies, the WBS should be the basis for the Product Backlog. This structure focuses on completed products that provide measurable customer value implemented in Epics and Capabilities. Agile development Sprints and Releases are just time-boxes established for executing work and do not represent product and should not appear in the contract WBS. While there is no single standard template for a WBS, MIL-STD-881D is a common reference used in DoD systems and automated information systems. Below you find an example WBS, indicating WBS Number, Task Name, and comments on how best to apply in an Agile EVM program.

WBS	Task Name	Notes
1	<b>Automated Information System (AIS)</b>	
1.1	<b>Automated Information System Prime Mission Product Release/Increment X</b>	Multiple elements at this level would be appropriate if the customer views major deliveries as independent products and desires a WBS organized around them. The key point is that elements at this level have no relationship with the Agile “release”
1.1.1	<b>Custom Applications SW 1..n</b>	Appropriate if Computer Software Configuration Items (CSCIs) are viewed as key products (with Epics/Capabilities contained within them); may be at L4 or not present at all (as explained below)

1.1.1.2	<b>Subsystem SW CSCI</b> 1..n	Appropriate if Computer Software Configuration Items (CSCIs) are viewed as key products (with Epics/Capabilities contained within them); may be at L4 or not present at all (as explained below)
1.1.1.2 or 1.1.1.2.1	<b>Agile Epic/Capability</b> 1..n	Would occur at Level 4 or 5. When Epics/Capabilities are the primary organizing method for products then these could be at L4 (preferred). Alternatively, Epics/Capabilities could be viewed as products within CSCIs.

*Figure 2: Example WBS for an Agile EVM program*

## Agile Product and Time Hierarchy Aligned with EVM

---

The next figure shows you a typical EVMS to Agile Hierarchy alignment. It illustrates that traceability between the EVMS and Agile hierarchies is defined and maintained throughout the program, aligning Scope and Budget via assigning sized Agile Products to Control Accounts, Work Packages and Planning Packages within the EVMS. Sizing of Agile Products is based on complexity of effort and is calibrated to equate to resources planned for each product. Depending on program size and system description, other alignments have been observed in industry also.

Note that traceability both within and between each hierarchy has been defined at program start at the CA/Epic/Capability and WP/PP levels, and for more detailed levels, at successive Release Planning/Rolling Wave Planning and Sprint Planning activities. 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 progress assessment.

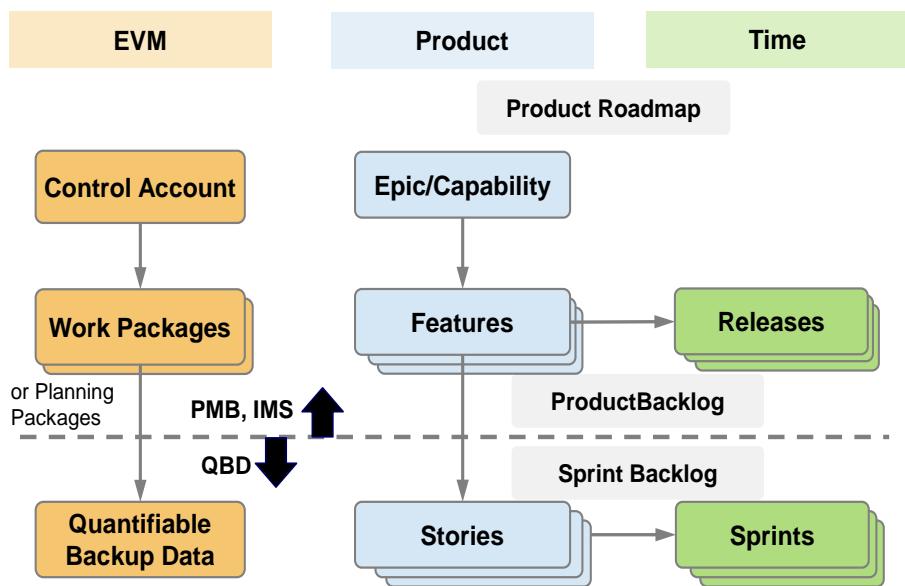


Figure 2: Typical alignment of EVMS to the Agile Hierarchy

# The Product Backlog

In the following Figure you can see visualized the product backlog which is almost identical to that in an “normal” agile Project. The Product Backlog includes a coding structure that traces to the Control Account Plan (CAP) with WP’s and PP’s. Budgets for Features are allocated based on the complexity of the effort. The Control Account BAC represents the planned cost for completing the product (EPIC).

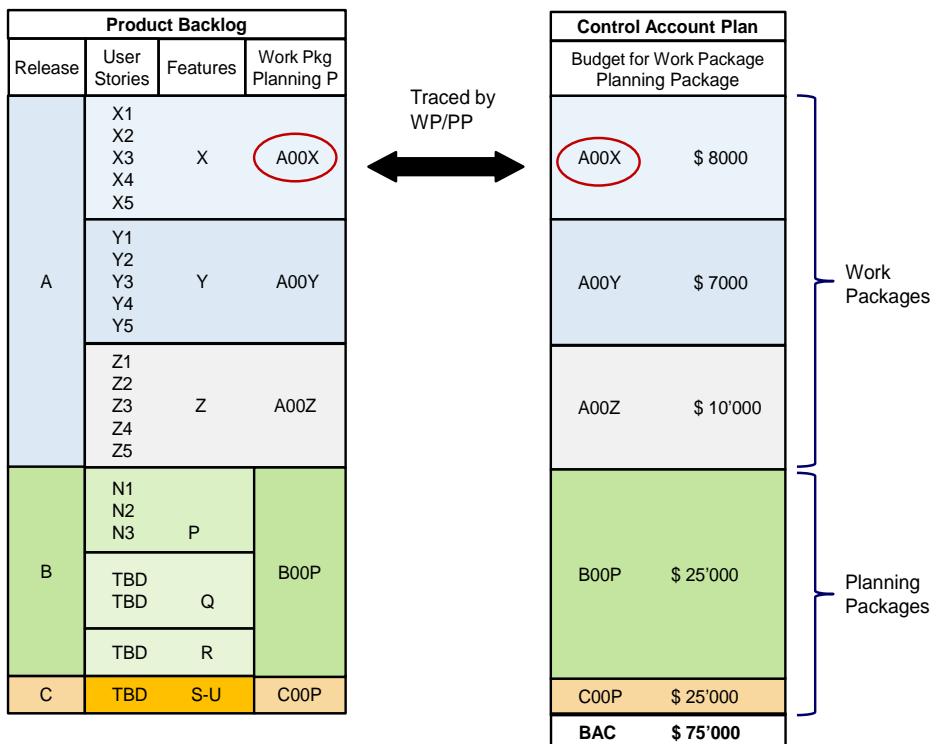


Figure 3: The Product Backlog traces to the Control Account Plan

# The Integrated Master Plan (IMP)

The Integrated Master Plan (IMP) precedes the Integrated Master Schedule (IMS) and draws from the product-oriented WBS, Statement of Work, Statement of Objectives, and Concept of Operations. In Agile the IMP Program Events can describe Epics/Capabilities of the product defined in these documents. Program Events can include major customer milestones, Capability Releases, and other customer deliveries.

The IMP may be initially developed in conjunction with an Agile Roadmap or the Product Backlog planning activity. The IMP reflects all the major customer milestones and deliveries, showing the order of the capabilities produced by the program. The Product Roadmap and Release Planning is the basis of the Definition of Done (DOD). The DOD equates to IMP's Significant Accomplishments and Accomplish-

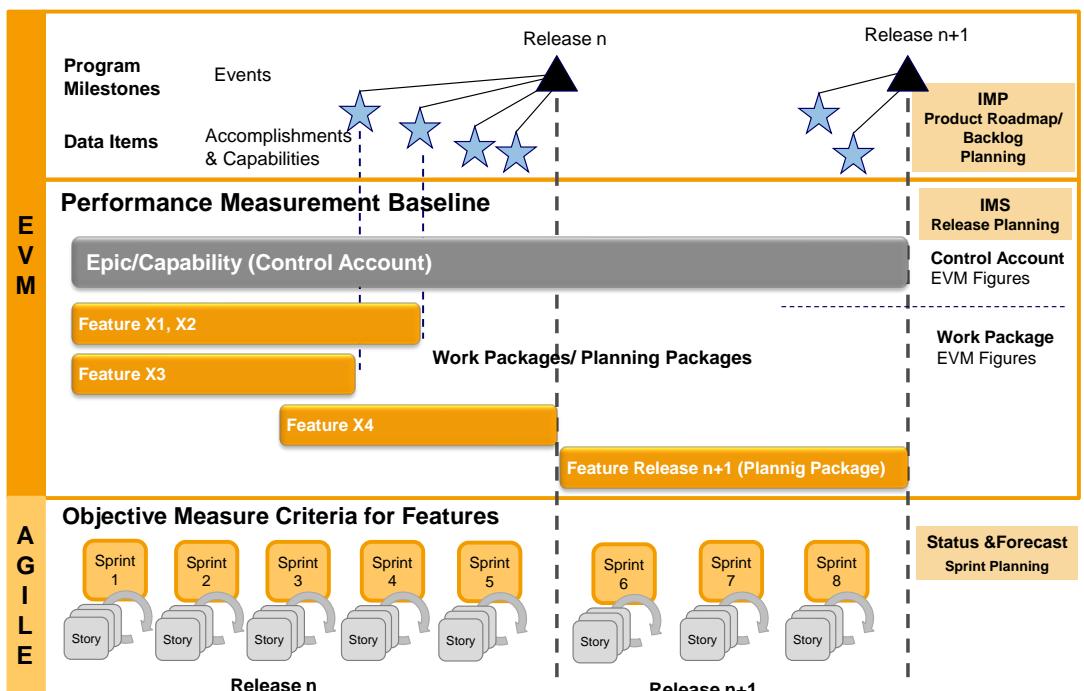


Figure 4: Agile IMP Event to EVMS Hierarchies

ment Criteria, for each Epic/Capability and their Features. The next figure shows Agile product and time hierarchies, and illustrates how the IMP, EVMS, and Agile elements are vertically and horizontally traceable in a single framework integrating Agile and Earned Value Management.

In this example, IMP events, that are compatible with Agile programs, include planned customer deliveries aligned to customer milestones. Initial delivery of completed work products, and later deliveries, are aligned with key mission milestones.

## **Integrated Master Schedule (IMS)**

---

Within the IMS Epics/Capabilities are decomposed into Features and Stories. An Epic/Capability delivers one or more Features and a Feature delivers one or more Stories. On larger programs, one or more “Sub-Epics” may exist between Epics and Features to manage the product decomposition to usable sizes.

Features are sized to fit within Agile Releases and represent significant pieces of the delivered product. Features should be the lowest level of an IMS, if there are no logic dependencies necessary for management control at a lower level. Stories serve as the implementation details of the Feature and are more efficiently maintained by Scrum teams outside the IMS in an Agile development tool.

Stories implement the Features in the IMS. Stories are assigned to Sprints in the Agile management tool. Features are traceable to work packages in the IMS by including an IMS reference (e.g. work package ID) as an attribute of the Feature. This traceability provides the needed visibility to Program Management from the Planned Value (PV) to objectively assess accomplishments at the work performance level in accordance with EIA-748.

## Freeze Period Considerations

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

## Control Account Plan

**Control Account** (CA) scope corresponds to Epics/Capabilities and their Features of the system. It is recommended that CA scope correspond to a single Epic/Capability. Work Packages are an element of control within CAs.

**Work Package** are an element of control within CAs. The number, content, size, and duration of Work Packages needed in a CA will vary subject to internal management needs and organizational policies along with the size and complexity of the program. A work package is the point where work is planned, progress is measured, and earned value is assessed. It is recommended to align one Feature or at most a small set of logically related Features with a Work Package.

After initial planning, Work Packages are defined during program execution through a series of Rolling Wave or Release planning cycles. For Features beyond the current Agile Release, the scope may be in Planning Packages, which will be refined during future release planning cycles.

# Aligning Agile Progress Metrics with Earned Value Reporting Levels

The next two figures are examples of Agile progress reporting used to status the PMB in the Earned Value Management System. In the next figure, the completion of Agile Stories, with attributed Story Points proportional to the effort, determines the completion status (Percent Complete (PC)) for a Feature, which is the lowest reporting level. The Story Points assigned therefore create a weighted Story Value for product completion status calculations.

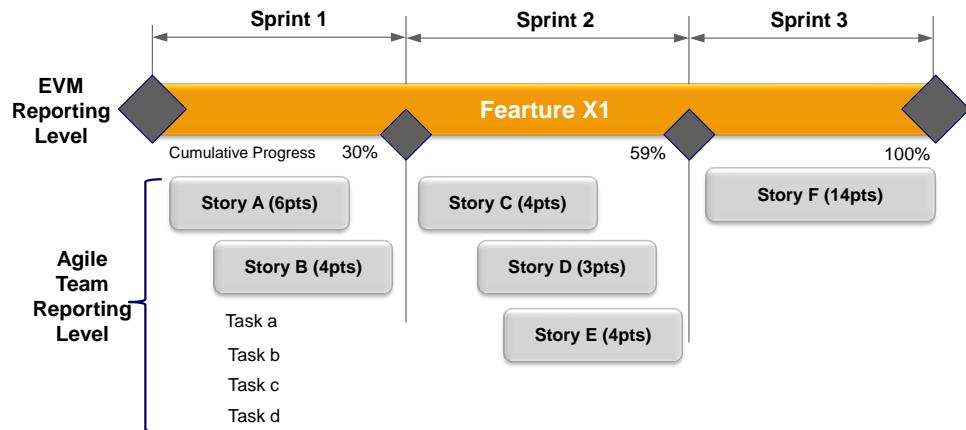


Figure 5: Agile product completion status rolling up into EVM reporting at the Feature level

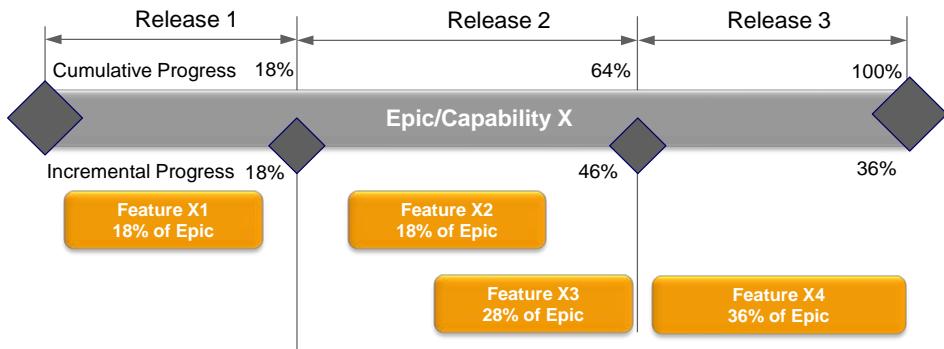
Percent Complete (PC) for the feature at the end of each sprint assuming stories are completed:

At the end of sprint 1, feature PC= 30%

At the end of sprint 2, feature PC= 59%

At the end of sprint 3, feature PC= 100%

Figure 6 shows measures of Percent Complete (PC) at the Capability level which are derived from Percent Complete at the Feature level using PC from the Feature level in Figure 5. Individual Feature Completion (PC) will still be determined based on completed Stories (similar to Figure 5.); now the Feature PC is used in roll-up reporting to the higher level item. One or more features are contained in a Work Package; therefore, the Epics/Capabilities, comprising Features, would logically align to CAs.



*Figure 6: Rollup of Agile product completion status to a higher level EVM reporting*

Percent Complete (PC) for the Epic/Capability at the end of each release:

At the end of release 1, Epic PC= 18%

At the end of release 2, Epic PC= 64%

At the end of release 3, Epic PC= 100%

# How to Analyze & Report Earned Value Performance

Progress can be calculated for a work package made up of a Feature or Features by tracking the completion of User Stories that are assigned to the feature(s). Agile progress reports showing weighted Stories (using story points) completed divided by total weighted Stories planned for the Feature Work Package can be used for the earned value technique of Percent Complete (PC). See an example of this calculation of Feature level status by giving 100% weighted Story value credit when the Story is completed (shown below in Figure 3-3).

Feature 1					
Agile Tool ID	Task Description	Story Points	Story Weighting	Story Complete %	NWA %Claim
PMG-245	Story 1	4	8.7%	100%	8.7%
PMG-245	Story 2	10	21.7%	0%	0%
PMG-245	Story 3	16	34.8%	100%	34.8%
PMG-245	Story 4	10	21.7%	0%	0%
PMG-245	Story 5	6	13%	0%	0%
<b>Feature</b>	Total Story Points	<b>46</b>	<b>100%</b>		<b>43.5</b>

*Table 1: Example of how planned Story Points may be applied to create QBD to calculate earned value*

## **Consider Feature Cost/Risk When Establishing the Baseline**

---

In Agile development, as well as in any product development, there are always uncertainties. It is recommended that these complexity factors be included in the relative sizing of Epics/Features used when establishing a Work Package cost/schedule baseline for that Feature.

Planning should also include the isolation of any reserve capacity or assumptions for defect time. Staff utilization, specifically the assumptions made during original complexity estimation for development focus factor, must also be considered when establishing baseline values.

As usual in any earned value managed program, unknown risks may be held at a higher level against Management Reserve for use when in-scope unanticipated work is discovered, and new functionality must be added to complete a product.

## **How to Use Agile Metrics to Forecast ETC/EAC**

---

The Agile methodology promotes incremental, iterative planning. When establishing the PMB, Planning Packages are typically employed which support this incremental approach. Agile does not advocate detailed planning all the way through to program end, which traditionally enables ETC/EAC forecasting. This avoidance of detailed longer-term planning is based on the principle that it is not possible to do accurately early on given limited data and the likelihood of customer-desired outcomes changing. Yet EAC forecasting is essential in EVM-managed, or any managed program.

For an Agile EVM managed program, a program's entire budget can be plotted out at a summary level via roadmap planning and top-level IMP/IMS. At the roadmap level, Epics/Capabilities and Features are estimated and allocated to Releases, and a baseline is established. At each

Release Planning event, the Planning Package for the next Release will be detail planned by finalizing the assignment of Features to the Release and Work Packages that have been initially allocated to the roadmap Epic/Capability plan in the form of Work Packages in the first Release and Planning Packages in subsequent Releases. The CAM should assess the complexity of remaining work in the Product Backlog that is aligned to those planning packages and compare it to the budget allocated to support EAC analysis.

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

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

The formulas below express how to calculate progress via PC on a single Feature as weighted Story Values expressed in Story Points (SP) completed versus the total weighted Stories planned. The second formula calculates the remaining hours of effort for a Feature using planned and completed weighted Stories in SP and hours used per completed Story Point.

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

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

## Further Details

This was a “short” summary with the most important content of the NDIA “An Industry Practice Guide for Agile on Earned Value Management Programs”. If you need more detailed information, check this Practice Guide. Among others, the following additional content from the very comprehensive guide is not included in this summary:

- Variance from the Baseline: Examples with Agile EVM
- Managing Baseline Change on Agile Programs
- Forecast Change Scenarios
- Contracting for Agile and EVM

## References and Further Reading

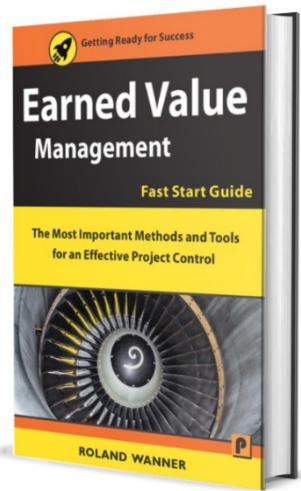
---

- Earned Value Management – Fast Start Guide: The Most Important Methods and Tools for an Effective Project Control, Feb 2020
- NDIA: “An Industry Practice Guide for Agile on Earned Value Management Programs”, Version 1.3, 26 May 2019
- PARCA: Agile and Earned Value Management: A Program Managers’s Desk Guide, 16 April 2018

# Earned Value Management – Fast Start Guide

**Discover How to Make Your Project Control Even More Effective and Bring More Transparency and Security to Your Projects**

This book covers the basic concepts of EARNED VALUE MANAGEMENT in an easy understandable way. You will find on more than 200 pages comprehensive knowledge about Earned Value Management, simply explained with more than 50 illustrations—and aligned with the PMBOK® Guide 6th Edition 2018.



**Buy the book direct from the Author (PDF) on [Gumroad](#)**

Buy the book as print copy or Kindle e-book on [Amazon.com](#)

**Bulk copies** with great discount are available direct from the author. Ask for a quote: [info@rolandwanner.com](mailto:info@rolandwanner.com)

