

# GAO and DoD Say Earned Value Management (EVM) Makes Sense on Agile Programs. But Is It a Good Fit?

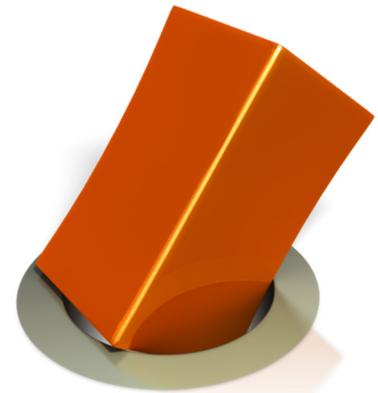
*“However, EVM can be a valuable performance management tool that decision makers can use to see how the [Agile] program is progressing compared to its initial plan.”*

*GAO Agile Assessment Guide, GAO-20-590G, September 2020*

## Bottom Line Up Front (BLUF)

Yes, you can find a way to apply Earned Value Management (EVM) to programs using agile methods. But should you?

Despite the GAO’s claim that EVM can be a “valuable performance management tool” on agile programs, EVM is at odds with several core agile values and principles and is unlikely to be a good fit. Instead, programs can use agile-friendly methods designed to more effectively achieve EVM’s intent of measuring progress.



*Figure 1. Square Peg, Round*

## What is EVM?

The GAO describes EVM as a way to measure the value of work accomplished in a given period compared against: (a) the planned value of work scheduled for that period; and (b) the actual cost of work accomplished<sup>1</sup>.

The US Air Force used early forms of EVM starting in 1966 and described the concepts in the Cost/Schedule Planning Control Specification (C/SPCS)<sup>2</sup>. While there have been minor updates and additions, after 50 years the core EVM concepts are the same.

<sup>1</sup> Summarized from the GAO Cost Estimating and Assessment Guide, GAO-20-195G, March 2020. Please refer to this document for a deeper introduction to EVM and examples.

<sup>2</sup> It is worth noting the practice did not originally include the word “value”. The focus is on controlling cost and schedule.

In 1966 when the Air Force described the concepts, almost all IT projects followed a waterfall approach. Using waterfall, customers did not have the ability to effectively evaluate an IT product until it was ready for testing<sup>3</sup>. And getting the product ready to test often took months or years because of waterfall's sequential, "build it all" nature.

The Government needed a way to make sure large programs were on-track, and EVM supplied a means to do that.

## Is EVM a Fit for Agile Programs?

To decide whether earned value is a fit for agile programs, we need to understand its makeup – in other words, **what problem is EVM designed to solve?**

*To decide whether earned value is a fit for agile programs, we need to understand its makeup – in other words, what problem is EVM designed to solve?*

EVM's designers structured it to show differences between a cost, schedule, and scope estimate (a.k.a., a performance management baseline) and actual work delivered<sup>4</sup>. EVM focuses on two aggregate measures: the Cost Performance Index (CPI) and the Schedule Performance Index (SPI). We highlight these measures because we continue to hear clients say that EVM is good at measuring "value". This is due to differing definitions of the term.

In a waterfall world, a project that delivers a requirements document on-time may be viewed as supplying value. **Agile programs embrace a different view: only working products can give value to consumers. Documents about what the software should do or how it might work do not.** This is a foundational concept of real agility.

Here are examples from an agile standpoint to clarify. Would a farmer waiting for a financial grant say that the U.S. Department of Agriculture (USDA) finishing a requirements document for a new grants management system provided them with any value? Would a veteran wanting to make a medical appointment say that when the U.S. Department of Veterans Affairs finished a design document on time, it helped them make their appointment faster? The answer is obviously no. As a result, EVM cannot measure value from an agilist's perspective.

<sup>3</sup> Our discussion is focused on IT products like software. EVM can be of more value on certain projects; GAO provides the example of laying railroad tracks.

<sup>4</sup> This is a simplification, but we stand behind the core focus.

To drive home the point, consider this: **you could have perfect EVM metrics for years – literally from program inception through implementation – and still deliver an IT product that the customer says provided little to no value.**

How is that possible? CPI and SPI will be perfect if you adhere to your plan – that is what they measure. If your plan is to build a requirements document in 12 months (your schedule) for \$10M (your cost), and you execute according to that plan, your EVM metrics will be perfect. If the Government cancels the program at that point, you still have perfect EVM metrics. Even if a customer signs off on the requirements, the program has not delivered what agile practitioners view as value<sup>5</sup>.

*You could have near perfect EVM metrics for years – literally from program inception through implementation – and still deliver an IT product that the customer says provided little to no value.*

Is that just a theoretical concern? Unfortunately not. EVM metrics often show IT programs are on-track until they fall off a cliff<sup>6</sup>. There are a variety of reasons why.

One reason is because **requirements documents are often ineffective at creating a shared understanding of business needs with technical staff. So even when we “build what customers asked for” there are often problems.** Agile programs take a different approach to requirements that reduce these challenges by focusing on interactive discussions and addressing needs in small batches.

Another reason is human nature. If we choose to measure adherence to cost and schedule (i.e., EVM) and try to ‘correct problems’ when those two metrics go yellow or red, Goodhart’s Law applies. Simply put, Goodhart said that when a measure becomes a target, it ceases to be a good measure. In other words, people may unconsciously work to make metrics show a positive result even when there are difficulties. If you have ever heard someone report, “that task is 50% complete” when you know the team has done almost no work on it, you have seen Goodhart’s Law in action.

Some of our clients view EVM simply as a way to track a program cost and schedule. That’s fine. We are not suggesting that tracking cost and schedule is wrong. But tracking those two things

<sup>5</sup> Agile programs believe that customer signoffs on large documents are ineffective in preventing many misunderstandings. This belief is one of many reasons agile programs take a different approach.

<sup>6</sup> It is often at this point that programs re-baseline their schedule to show that work is now going exactly according to the [newly revised] plan.



does not require a special system that is “consistent with the 32 Guidelines contained in the Electronic Industries Alliance Standard-748 EVMS (EIA-748)”. Any spreadsheet can track cost and schedule without an investment in rigorous EVMS processes and tools. The GAO’s critiques of how agencies implement EVM highlights the time, cost, and complexity of doing EVM properly<sup>8</sup>. You do not need a stable of master schedulers to track cost and schedule in a spreadsheet.

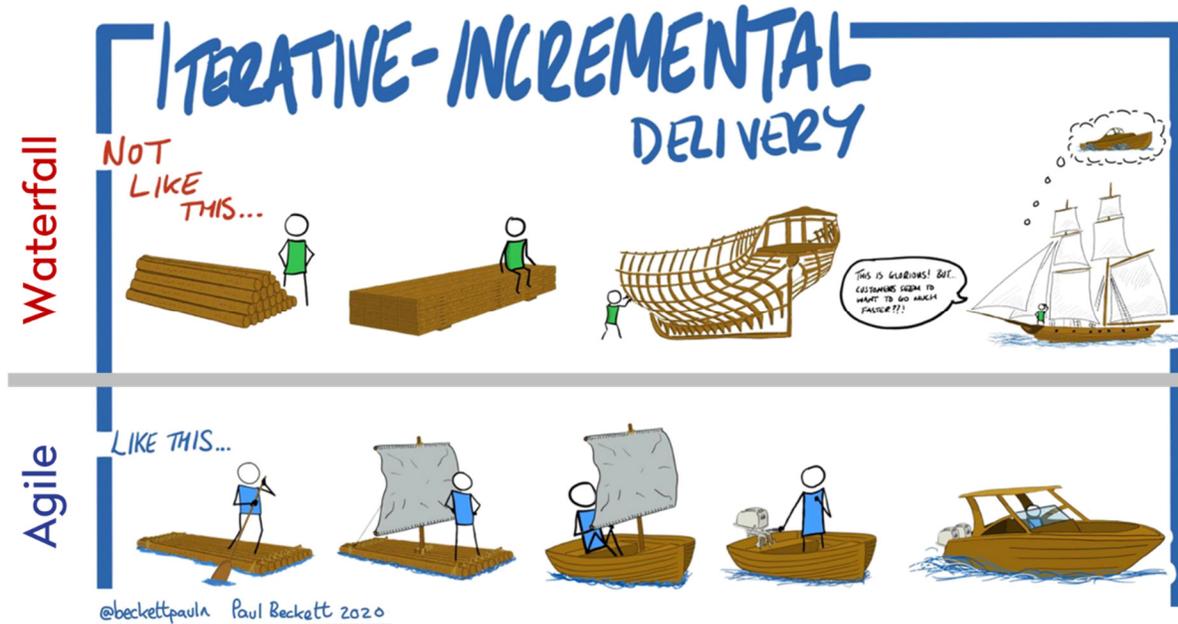


Figure 2. Pivoting Scope: Comparing Waterfall and Agile<sup>9</sup>

## EVM on Agile Programs Is Messy

A foundational concept in agile is that only working products can provide value, and EVM is not designed to measure that type of value. But even if EVM could measure value, would it work on an agile program? We could make it work, but it is not a natural fit. Here’s why.

On a waterfall program, project staff decide the cost, schedule, and scope near the start of the program. These decisions form what is called the EVM performance management baseline. EVM needs this baseline to measure against and discourages the practice of “rebaselining” regularly because it makes the SPI and CPI metrics meaningless.

<sup>7</sup> From the DoD EVM Implementation Guide (EVMIG), January 2019.

<sup>8</sup> For examples, see GAO-10-2 published Oct 8, 2009 and GAO-13-22 published Nov 19, 2012.

<sup>9</sup> We thank Paul Beckett of Edinburg Agile for allowing us to share his beautiful graphics.

*Effective agile programs do not commit to an exact scope and timeline at the start.  
Without a relatively static baseline, EVM is ineffective.*

In contrast, effective<sup>10</sup> agile programs do not commit to an exact scope and timeline at the start.  
**Without a relatively static baseline, EVM is ineffective.**

Instead of building static long-term schedules and requirements, agile programs create hypotheses about what software consumers (e.g., citizens, veterans, war fighters) need, evaluate those beliefs through small experiments, and calculate how to satisfy those needs effectively and efficiently.

**Because of this focus, agile programs fully expect to pivot their scope and timeline as they learn.** These pivots mean key activities and scope can change from month to month. Agile programs encourage these pivots. EVM requires procedures and approvals to change the initial plan, which opposes agile's focus on flexibility.

One reason agile programs use this approach is because of studies showing that people are inherently bad at estimating. For example, a McKinsey and University of Oxford study of 5,400 projects found that the average cost overrun on large software projects was 66% and average schedule overrun was 33%. But CPI and SPI are measures of adherence to estimates for cost and schedule, again suggesting EVM is not the best fit for agile programs.

*The intent of EVM is to answer this question: are customers on track to receive the value they expect from an IT product? Luckily, there are agile-friendly methods designed to address it.*

## Agile Friendly Alternatives to EVM

While EVM only measures cost (CPI) and schedule (SPI), we agree that people need a way to measure value on agile programs. We need to be able to answer the question, "are customers on track to receive the value they expect from an IT product while it is being built?" Luckily, there are agile-friendly methods designed to do that.

---

<sup>10</sup> We highlight effective here because "bad agile" or what the Department of Defense (DoD) refers to as "Agile BS" is rampant. We and the DoD offer "Agile BS Detection" tools to help.

## Use a working product [/software] as the primary measure of progress

In waterfall programs, we commonly use requirements, designs, and other documents as proxies to measure progress. But as discussed above, documents are bad proxies and do not constitute value on agile programs.

In contrast, agile programs believe the best way for customers to gauge progress and value is through hands-on experimentation with a working IT product. So agile programs build something customers can interact with and evaluate often – most commonly, every two weeks<sup>11</sup>.

*Agile programs believe the best way for customers to gauge progress and value is through hands-on experimentation with a working IT product.*

## Supplement working software with value metrics

Because agile programs create working software frequently, customers can evaluate the business value on a regular cadence.

For example, at the beginning of a quarter business owners<sup>12</sup> can assign a planned value to three major features a program expects to deliver over the next 90 days. At the end of the period, those same business owners can assign an actual value score to those same features.

While subjective, this process allows fast feedback and frequent improvements to avoid major disconnects that occur when projects take years to deliver something users can interact with.

Using these methods helps us adhere to several foundational agile concepts and principles.

### Concept 1: Maximize the amount of work not done

Part of why effective agile programs save money is because they try to do less. In this case, that means since agile programs can measure value through working software, we do not need to create and monitor EVM metrics focused on a “value proxy” (e.g., progress on a design document). No proxy is needed. No adherence to the 32 ANSI-748/A EVM System requirements or special software is required. The results: lower cost, lower administrative overhead, and a more effective way to gauge progress and value.

<sup>11</sup> Scrum, the most popular agile method in the world, most commonly shows working software every 2 weeks.

<sup>12</sup> You can accomplish the same thing with a representative subset of system users.

## Concept 2: What we think we need often changes as we interact with working products

Waterfall approaches imagine people can know, capture, and use a requirements document to clearly communicate their needs at the start of a project. Many of us who have been part of these processes are painfully aware that these beliefs are flawed. And lessons learned literature routinely cites poor requirements as a key reason for failure.

Agile programs assume the opposite – we do not know what we want, and even if we did, documents are an ineffective way to communicate it. Interacting with working products helps us define and prioritize what we need. Agile programs assume we will learn as we go, and support frequent changes.

*EVM's concept of measuring performance against a baseline suggests that the initial plan is vital. Agile programs assume the original plan is going to change – often many times – so the baseline is much less important than the pivots.*

Put another way, agile values and principles embrace change. In contrast, EVM's concept of measuring performance against a baseline suggests that the initial plan is vital. **Agile programs assume the original plan is going to change – often many times – so the baseline is much less important than the pivots.**

## Conclusions

Sure, you can find a way to apply EVM to agile programs, but it will be messy. And why would you want to?

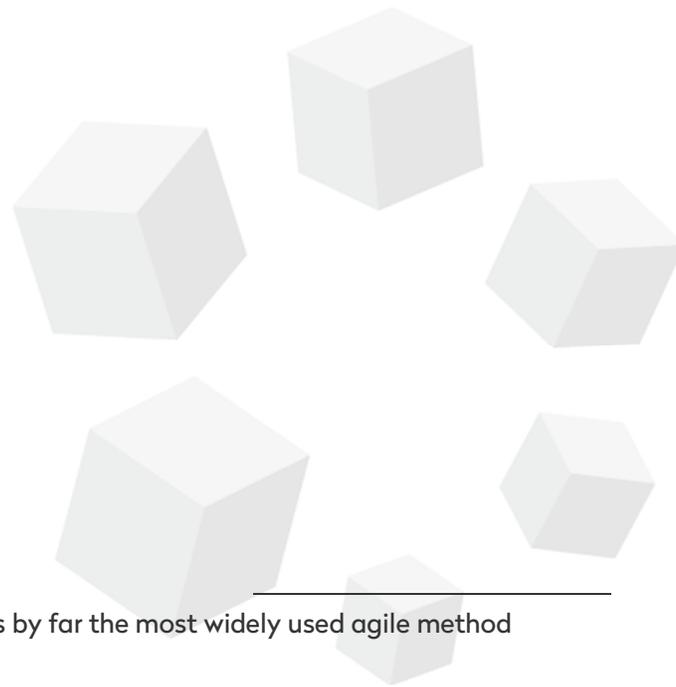
EVM measures cost (CPI) and schedule (SPI), not what agilists consider value. And agile values and principles are at-odds with several EVM practices, like measuring cost, schedule, and scope against a baseline, and requiring a change control process to alter the baseline.

We understand the drive to apply a method many are familiar with to agile programs. But how we perform agile programs is hugely different from waterfall-based EVM concepts devised in the 1960's.



Instead of EVM, we propose using methods built-in to agile projects like showing a working product every two weeks<sup>13</sup>, measuring the difference between planned and delivered business value at regular intervals, and adapting our plan and scope over short cycles.

Agile's creators recognized the problem EVM aimed to solve over twenty years ago. By design, techniques that align with its values and principles do a better job of validating value delivery than EVM ever could.



---

<sup>13</sup> Using short sprints (e.g., two-weeks) is a Scrum concept. Scrum is by far the most widely used agile method in the world.