

# Metrics for Agile Projects:

Finding the Right Tools for the Job

An ESI International  
White Paper



+44 (0)20 7017 7100

[www.esi-emea.com](http://www.esi-emea.com)

# Contents

Abstract.....	3
Introduction .....	4
Tools for Tracking Progress.....	5
Exhibit 1: Iteration Status Chart.....	5
Exhibit 2: Burn-Down Chart.....	7
Team Communications.....	8
Applying EVM to Agile Project Management.....	12
Exhibit 3: Agile Earned Value Management .....	13
References .....	15
The ESI Solution .....	16



# Abstract

Agile project managers (PMs) have a wide range of metrics from which to choose that show different types of progress at the iteration and release levels of agile projects. A deeper understanding of the variety of metrics for agile will enable PMs to determine how and when to use them most effectively to communicate progress to project stakeholders. This paper will begin by reviewing metrics PMs currently use to track progress and report progress to stakeholders in a project, an iteration and a release. Details for iteration planning and estimating will build the framework for selecting progress reporting metrics. Best practices for progress reporting will include a detailed examination of how and when to use the methods available, including daily meetings, iteration delta tables, release and iteration burn-down charts, progress reports, running tested features (RTFs) and earned value management.

This informative paper will provide best practices project managers should use to expand the use of the metrics at their disposal to fine-tune the progress they are showing on their agile projects.



## Introduction

One of the key strengths of traditional project management methods is that they provide a robust, tested toolkit of metrics and reporting techniques that grant stakeholders visibility into the progress and challenges of the development effort. By measuring progress against the Gantt charts developed during project planning, by ticking off tasks on the work breakdown structure (WBS) as they are accomplished and by monitoring issues logs and defect, or “bug” lists, stakeholders can keep a finger on the pulse of the project and can set their expectations appropriately. Modern project managers (PMs) add indicators, like project dashboards, to further simplify the tracking of projects by refining the main points of progress and challenges into simple, easily parsed graphics.

Due to the nature of agile projects, many of these familiar metrics are not available. Since agile PMs are not developing long-term, multi-month project plans with predictions that reach far into the future, the familiar Gantt chart and WBS are rare in agile programmes. Agile PMs utilise other metrics, such as burn-down charts, to understand where their projects stand. The PM is responsible for the overall delivery of the project’s objectives and performs this work by keeping the team focused, encouraging teamwork and individual productivity, ensuring that the team has the resources required to be successful and managing the customer relationship as the product is built. Agile PMs are not focused on updating Gantt charts or keeping the project dashboard current; their focus instead is on keeping the performance of the team at the highest level of productivity and collaboration and in isolating the team from distractions, barriers and digressions that could harm their concentration. Since, in agile environments, the team members are expected to manage their own workloads, PMs serve to ensure that their teammates have the information, resources and technical coaching they need to be successful.



# Tools for Tracking Progress

During the building of an iteration, while the developers are focused on creating the features they have committed to deliver, the PM is responsible for understanding the progress that the team is making and keeping the stakeholders informed of the project's progress. Exhibit 1 is an example of a chart that PMs can use to communicate iteration status to the customer and stakeholder community. Designed to communicate the changes from one iteration to the next, this chart helps stakeholders see what the original plan for the current and next iterations looked like a couple of weeks ago and how they look today. Features that are new, have been deleted and/or moved from one iteration to another are all clearly indicated.

Two Weeks Ago		Today		Prev.	Curr. Status
Iteration 1	This is item 1...	This is item 1...	1	1	✓
	This is item 2...	This is item 2...	1	1	✓
	This is item 3...	This is item 3...	1	1	✓
	This is item 4...	This is item 4...	1		
Iteration 2	This is item 1...	This is item 5...	2		
	This is item 2...	This is item 6...	2		
	This is item 3...	This is item 15...	New		
	This is item 4...	This is item 8...	2		
Iteration 3	This is item 9...	This is item 13...	3		
	This is item 10...	This is item 9...	2		
	This is item 11...	This is item 7...	2		
	This is item 12...	This is item 11...	3		
	This is item 13...	This is item 10...	3		
	This is item 14...	This is item 12...	3		
		This is item 14...	X		

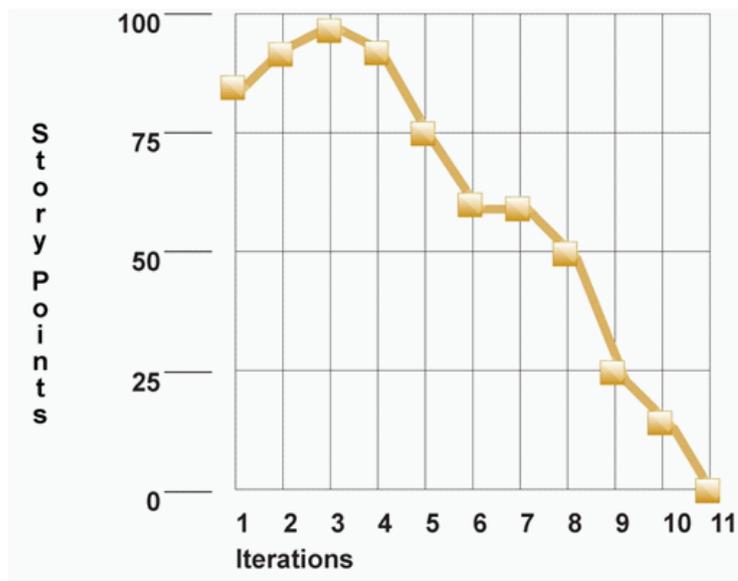
**Exhibit 1: Iteration Status Chart**

Another tool that many agile PMs use to track progress is the “burn-down chart,” illustrated in Exhibit 2. The PM starts at the upper left of the chart, before any development work is completed, and as features are completed or “burned down,” the PM tracks that progress with a simple line chart. In this example, the PM is tracking a release, which contains a total of 200 features, or story points, that the team must “burn through.” A burn-down chart also can track the progress of an iteration by substituting “days” for “iterations” on the y-axis. As features are built and tested, the number of story points remaining is plotted and a line connecting each point is drawn. By keeping the burn-down chart current, PMs can offer their customers a quick, visual representation of the functionality that has been delivered and is left to be delivered, and provide clues regarding the team’s productivity.

If the chart is a flat line and not moving down toward the zero point at the lower right, it shows that the team is stuck, not able to make progress as expected. It sends a signal to the PM that a technical issue, an organisational barrier or a teammate’s productivity requires attention. If the plot moves upward, it indicates that new features have been added to the backlog or features previously considered complete have moved back to development. In agile development, a new feature added to an iteration can mean many things. It can be a positive sign of fruitful interaction with the customer, or it can be the first indication of scope creep or project gold-plating (in which team members add features because they are “cool” or “elegant”).

If an upward tick in the burn-down chart signifies that a feature previously considered complete has moved back into the unfinished column, this is a negative indicator. Features should only be burned down on the chart if they are developed and tested and ready to be integrated into the iteration or release. Counting unfinished features creates all sorts of complications, from damaging internal team trust to corrupting the entire burn-down process. PMs should set clear rules regarding the state of completion required before a feature, or story point, is counted as burned down.

The burn-down chart is also very revealing in regards to the team's estimating success. If the plot is proceeding as expected, and the trend line is an even, steady descent from the top-left corner to the bottom right, this signifies that the team has been extraordinarily successful at predicting its velocity and selecting the right features and right number of features to include in the iterations. If, on the other hand, the line is choppy, sloping up and down and stalling frequently, it can indicate that the team is not cohering efficiently, or that unforeseen complexities are arising that must be addressed. It should be clear that, in all of these circumstances, the PM and the team can use the burn-down chart as a key indicator, not just of progress but of the internal productivity of the team.



**Exhibit 2: Burn-Down Chart**

It should be clear that, as developers go off to do the technical work of implementing the features for which they have signed up, there is the risk of creating little islands of technology, which do not integrate well into a coherent product. This risk, of course, exists in traditional as well as in agile projects, and the collaborative techniques of agile PMs should assist in minimising integration problems. Still, agile PMs must guard against this contingency, and the mechanism typically used for this often is referred to in the agile community as “frequent integration.” Most experienced PMs can cite an initiative in which all of the separate elements were tested and performing appropriately, only to find that, when put together, the interaction of the parts created unexpected issues, consequences and even failures. In fact, many projects are thrown off track by late integration challenges, which can be hellish to uncover and repair. Although integration is not a feature and will not show up in a task allocation session or on a burn-down chart, prudent PMs ensure that they have allocated time in their iteration cycle to pull the separate components of their design together as they are created and ensure that they are working as a unit. Some agile teams assign a single teammate to be the integration manager, with responsibility for reviewing components as they are completed and ensuring that they fit together.

## Team Communications

While the burn-down chart can provide an at-a-glance overview of the team’s progress, this is not enough to help the team stay on top of the project effort. In most agile approaches, teams use a variant of the daily meeting to track the daily issues, risks and challenges that arise. These sessions often are called “stand-up” meetings because agile proponents believe that, by standing rather than sitting around a conference table, teams are encouraged to keep these meetings short, concentrated and effective, which helps to avoid the plague of irrelevant and endless meetings that can distract from forward progress. In this “stand-up” method, the team stands together and, in round-robin fashion, answers three simple questions:

- What have you done since the last stand-up meeting?
- What will you do until the next stand-up meeting?
- What issues, risks or barriers have arisen that could distract you from accomplishing the iteration goals?

Other agile methods have their variants of the daily meeting, but all are focused on planning and communication at the granular, daily level. Some standard characteristics of a daily meeting are: each meeting is 15–20 minutes; everyone stands in a circle; each meeting occurs at the same location; the order of presentation is defined; team members share status/obstacles; and all team members are invited, but only people involved in the iteration speak at the meeting. This meeting is the forum for discussing the tasks that slipped or were completed early and the unexpected barriers or challenges that arose in the previous day of development, and for refining plans on a daily basis depending on the realities of this particular effort and team. Features and tasks can be swapped by team members as they discover their own suitability for the tasks they have accepted. Schedules can be refined as the team's velocity becomes clearer. This daily team meeting often is referred to as the heartbeat of the agile methodology.

Agile PMs have an important responsibility to the customer; they must coach the customer on the responsibilities of acting as the organisation's representative on the development team. This is often a new and unfamiliar role to customers, who are accustomed to developing a set of requirements and then waiting for the product to appear from the development lab. The intimate, continuous participation of the customer in the development process and in the decisions made is a key precept of agile development, but that does not mean that the customer is prepared to play that role effectively. Strong agile PMs help to mentor customers through the process, bringing them into the iteration planning sessions, inviting them to the daily meetings, and helping them to understand project status by teaching them to interpret the burn-down charts that the team uses to track its progress. The role of the customer in an agile project is demanding, and the PM must ensure that customers understand and fulfil their role. Without the constant input, reaction and support of the customer, the agile effort will flounder. The PM must ensure that this disconnection is avoided.

With iteration status charts, burn-down charts and daily stand-up meetings being effective progress reporting tools, sometimes these can be obscure to the uninitiated and do not answer stakeholder questions about value delivered, costs and resources consumed or future expectations. While agile theory believes that customers should be so intimately involved in agile development efforts that they know as well as the actual developers where the project stands at all times, in practical terms this is rarely the case. Even if it were, customers may not have the skills or language to pass that understanding to the entire stakeholder community. Clearly, it is critical that agile PMs devise strategies for tracking and measuring the team's performance on the project and for reporting that progress to the customer and stakeholders.



One of the key philosophies of agile development is that the elements that are not directly tied to developing the features of the product, such as documentation, reporting, administration and measurement, are to be kept “barely sufficient” — enough to deliver the value that the customer is expecting, but with a minimum of overhead, complexity and ceremony associated with them. This philosophy is applied in metrics as well. In many agile teams, the burn-down chart, or some variant, is the sole visible metric of progress. Other teams go to the opposite extreme and attempt to apply all of the artefacts of a traditional methodology — such as a WBS, Gantt chart, earned value management (EVM) and dashboard — to agile projects, even if they only apply to one iteration at a time. While most agile adherents would resist the adoption of the entire suite of traditional project management office (PMO)-style metrics, most would also agree that something more than the burn-down chart is required.

For those teams coming from a traditional project management environment, familiar techniques like EVM, with an agile twist, can bridge the gap between PMO-style practices and more agile methods. EVM is a project measurement technique used to evaluate and predict project performance against the plan. PMs training in the Project Management Institute (PMI®) process will recognise this technique, which measures three elements:

- Planned expenditures
- Actual expenditures
- Actual work performed against planned expenditures

This last element, of course, is the crux of the impact of EVM. It measures the value of what has been delivered for the amount of budget spent. Its underlying concept is obvious: If you have spent 80 percent of the budget and only delivered 20 percent of the value, you have a productivity problem. If you have spent 20 percent of the budget and achieved 80 percent of the value, you have got plenty of slack in case of contingencies, and you also probably have a serious problem in your estimating process.

Formal EVM is a rigorous process, measuring components such as:

- Planned value (PV), which represents the budgeted cost of a defined amount of work; for instance, how much was budgeted for the work expected to be done in the next three weeks
- Earned value (EV), representing the budget for the amount of work expected to be done during a defined time period
- Actual cost (AC), the real cost of delivering the actual work accomplished in a specified period

These metrics are then used as the basis for calculations that give PMs guidance as to the performance of their projects. Calculations like cost variance (CV), schedule variance (SV) and estimate to complete (ETC) can be derived from various formulas (*PMBOK® Guide*, pp. 182, 184-185).

While the method of calculation seems rigorous and disciplined, there are sceptics regarding the application of EVM. Since it claims to measure delivered value, some doubters question whether an incomplete project delivers any value at all: Of what value is half a runway, or three-quarters of a software application? This explicit connection of tasks completed with value would meet with tremendous resistance among agile proponents. Notwithstanding cynics, EVM is a widely accepted project measurement technique. Can EVM be used with agile techniques, and, if not, how can a rigorous measurement be applied to agile project processes?

In traditional project approaches, it is assumed that the scope is fixed, any changes will be managed through rigorous change control and the project will proceed in a predictable manner in which past performance is indicative of future progress. EVM calculations are made based on a detailed WBS, and PMs focus on areas out of compliance to keep the project on track. Managers and customers rely on EVM to understand project performance and to manage teams, contractors and vendors.

In agile projects, many of the underlying assumptions are not applicable. Project scope is assumed to be defined broadly, and only the current iteration is planned and estimated in detail. Changes are expected as the product is refined and optimised, making it difficult to baseline a project.



## Applying EVM to Agile Project Management

To apply measurement techniques like EVM to agile project management, PMs should focus on the expected outcome rather than the method. The use of burn-down charts as a measurement and reporting tool provides many of the benefits of EVM, but in a different form. If the burn-down charts are kept scrupulously current and analysed prudently, PMs can use them as a key indicator of project progress, problems and risks.

Some agile proponents have taken the application of EVM to agile programmes to a further level of rigor in a method they call, appropriately enough, AgileEVM. In a paper presented to the IEEE, a team of agile developers had proposed a disciplined, quantitative technique for applying EVM to agile methods (Sulaiman, Barton & Blackburn). AgileEVM uses story points, rather than tasks performed, as the basic unit of measurement and measures iterations planned against iterations completed to derive the value delivered.

As in EVM, Agile Earned Value Management (Exhibit 3) requires initial baselines such as number of planned iterations, number of planned story points in a release and planned budget for the release. Also needed are the total number of story points completed, number of iterations completed, actual cost and the number of story points added or deleted from the release plan.

AgileEVM Metric	Definition
PSP (planned story points)	Total number of story points planned for this release
PW (planned weeks)	Total number of planned development weeks
BAC (budget at completion)	Total budget for the release
AW (actual weeks)	Number of development weeks elapsed to date
CSP (completed story points)	Number of story points completed to date
PPC (planned percent complete)	$AW/PW$
APC (actual percent complete)	$CSP/PSP$
AC (actual cost)	Total budget spent to date
PV (planned value)	Budgeted cost of the story points that were scheduled to be completed as of today; $PPC * BAC$
EV (earned value)	Budgeted cost for the story points actually completed as of today; $APC * BAC$
CV (cost variance)	Difference between planned budget and actual spend; $EV - AC$
SV (schedule variance)	Difference between planned schedule and actual time spent; $EV - PV$
CPI (cost performance index)	$EV/AC$
SPI (schedule performance index)	$EV/PV$
ETC (estimate to complete)	Based on current state, how much additional budget is needed to complete the release; $1/CPI * (BAC - EV)$
EAC (estimate at complete)	Based on current state, what the total estimated cost of the release will be at completion; $EAC = AC + ETC$
Estimated time to complete	Based on the current state, the total estimated time needed to complete the release; $1/SPI * PW$

### Exhibit 3: Agile Earned Value Management



For example, the budget for the release is \$100,000 for a completion of 100 storyboards. At this time, you have completed 25 of the storyboards at a cost of \$20,000.

Actual percent complete = 25 completed storyboards/100 storyboards = 25 percent complete.

EV = actual percent complete X total budget = \$25,000.

These modifications make the version of EVM more compatible with agile techniques, while providing a familiar terminology and reporting mechanism.

Agile projects do not necessarily need to reinvent every artefact; traditional status reports can be used as long as it is clear that they refer to the current iteration rather than the entire effort. A typical status report, with an executive summary, accomplishments report, listing of upcoming activities, and issues/barriers list, can be quite effective and has the added advantage of being familiar to the audience. As long as the project team stays focused on completing the development work and delivering a working product or prototype rather than being distracted by status reporting, traditional methods of communicating progress can be effective.

It is important to remember that many executives, stakeholders and customers are steeped in traditional project reporting methods and can feel unconnected or uninformed during the migration to agile techniques. Customer-focused agile PMs work closely with their clients and sponsors to develop reporting mechanisms that tell clients what they need to know and inspire confidence in the team through transparency and communication.

## References

Project Management Institute. (2008) *A guide to the project management body of knowledge (PMBOK® Guide)* (4th ed.). Newtown Square, PA: Project Management Institute.

Sulaiman, T., Barton, B. & Blackburn, T. (2007). *AgileEVM – Earned value management in scrum projects*.

PMBOK® is a trademark of the Project Management Institute, Inc., and is registered in the United States and other nations.



## The ESI Solution

Agile PMs need to think and act differently than traditional PMs. They need to consider the specific needs of their agile project and utilise the variety of metrics to effectively communicate their progress to their project stakeholders. ESI can equip your agile project managers and team with the tools they need to implement successful agile projects. Whether it is knowledge assessments or training on the competencies required for agile project management, coaching and mentoring support, one-on-one consulting or reinforcement workshops, ESI will partner with your organisation to ensure project success.

ESI International, the leading provider of project management, programme management, business analysis and contract learning programmes has helped some of the world's most successful organisations drive results. Our top-quality performance improvement programmes and consulting services have enabled domestic and international corporations and public agencies to enhance their employees' skills and talents and improve their internal systems and processes. With the support of our educational partner, The George Washington University in Washington, DC, ESI has helped more than one million professionals and 1,000 clients worldwide achieve individual and organisational objectives.

Nancy Y. Nee, PMP, CSM, CBAP, Executive Director, ESI International was instrumental in the development of this paper.

+44 (0)20 7017 7100



For more information about how ESI can help your organisation, call us at +44 (0)20 7017 7100 or visit our website at [www.esi-emea.com](http://www.esi-emea.com).

[www.esi-emea.com](http://www.esi-emea.com)