

Calculating Earned Business Value for an Agile Project

By Dan Rawsthorne, PhD, CST

Given the widespread adoption of Agile methodologies that has taken place over the past few years, it's clear that these practices "work."

So why have so many organizations clung to software development that is artifact-driven and waterfallish? The most common arguments against Agile practices contend that they are too developer-centric, their processes too lightweight, and that business feedback is difficult to interpret. In fact, that last claim is critical: Many managers in larger organizations are reluctant to adopt Agile practices because they depend on the metrics of traditional project management.

In this paper, this problem is addressed by defining Earned Business Value (EBV), a new metric for Agile projects that replaces standard Earned Value Analysis (EVA) metrics. With EBV, Agile teams receive a clearer picture of a project's progress, which can be used to strategically adapt to evolving business conditions.

EVA is a powerful measurement method, which is common to "standard" project management. It comprises three basic metrics :

- Budgeted Cost of Work Scheduled (BCWS): The baseline cost up to the determined status date.
- Actual Cost of Work Performed (ACWP): The actual cost required to complete all or some portion of the tasks, up to the status date.
- Budgeted Cost of Work Performed (BCWP): The value of the work performed by the status date, measured in currency.

An EVA requires the combination and manipulation of these three metrics. Unfortunately, neither BCWP nor BCWS - which require a detailed, up-front plan to calculate - are applicable in an Agile environment. Certainly, it makes sense that a new development paradigm would necessitate new ways of charting performance. But the question persists: What metrics are available for Agile teams to quantify project progress and respond accordingly?

THE GOALS

Answering the above question requires first examining EVA. Why does it exist? What does it measure? What corresponding metric could replace it? Typically, what management wants to find in an EVA is the progress made on a particular project, i.e. how it's going and when it will be done. Put another way, management is looking for information about the "value" of the product. What they really want to know is how much value the product is currently providing or what percentage of the product is "done." In an Agile environment, the metric that measures the extent to which a product is complete, from a business perspective, is called Earned Business Value (EBV). But before turning to EBV, two concepts unique to Agile must first be discussed: the feature and the story.

A feature is a characteristic or attribute of a product for which work must be done to develop and deliver it. It is what allows a product to be advertised, marketed, and sold.

FEATURES AND STORIES

A feature is a characteristic or attribute of a product for which work must be done to develop and deliver it. A feature is what gives a product its purchase in the marketplace; that is, it is what allows a product to be advertised, marketed, and sold. Typically, a product (or release) is described as a list of features, examples of which might include:

- Advanced Customer Information Updating Capabilities
- XYZ Claims Form Compatibility
- Additional Reporting Options
- Improved User Interface
- Faster Report Rendering
- Securer Protection Against Hackers

Features are implemented in the product by working on stories, which are said to “belong” to the features. A story is the fundamental unit of value, requirements, and work. The story is both internally and externally visible and is therefore used as the interface between the business and development sides of a project. In Agile, stories are particularly significant because they represent the smallest unit of value that can be managed. Ron Jeffries, an authority on Extreme Programming, has relayed Alistair Cockburn’s description of stories as “promises for conversation.” This is because a story is a simple request for something of value, the details of which are fleshed out in an ongoing dialogue, or negotiation, between the product owner and the team. It is not a micromanagement tool, nor does it insist that a team complete the story in a particular way. More specifically, a story consists of the following:

- Description: A short description of the goal to be achieved.
- Size: A rough estimation of the effort required to achieve the goal. For estimating, Agile teams have used an infinite number of scales, including Story Points (SPs); T-shirt sizes (S,M,L,XL); the time required for completion; even gummy bears.
- Validation Criteria: Also known as “acceptance criteria,” this is what determines when a story is “done.” This is probably the most important attribute because being “done” is what allows the team to claim the value of a story.
- Business Value: Not all stories contain business value. However, those stories that drive development will always include a business value. This is the metric we are trying to calculate.

There are various types of stories. By way of illustration, what follows are hypothetical stories for a “Customer Information Update” feature:

- Determine what stakeholders want developed (analysis story, which is “done” when there is a validated user scenario, screen mockups, and success post-conditions)
- Develop the main execution flow (development story, which is “done” when there are verified tests that pass)
- Determine alternative flows and how to execute them (analysis story, which is usually time-boxed)
- Develop alternative flows X, Y, & Z (development story, one for each alternative flow)
- Stress-test the feature for 500 simultaneous users (test story, which is “done” when there is a validated testing regiment that passes)

The most valuable metrics provide management with a “big picture” perspective of the team’s work. Although many Agile users cringe at its “high ceremony” connotations, the Work Breakdown Structure, commonly used in standard project management, gives such a view.

WORK BREAKDOWN STRUCTURE

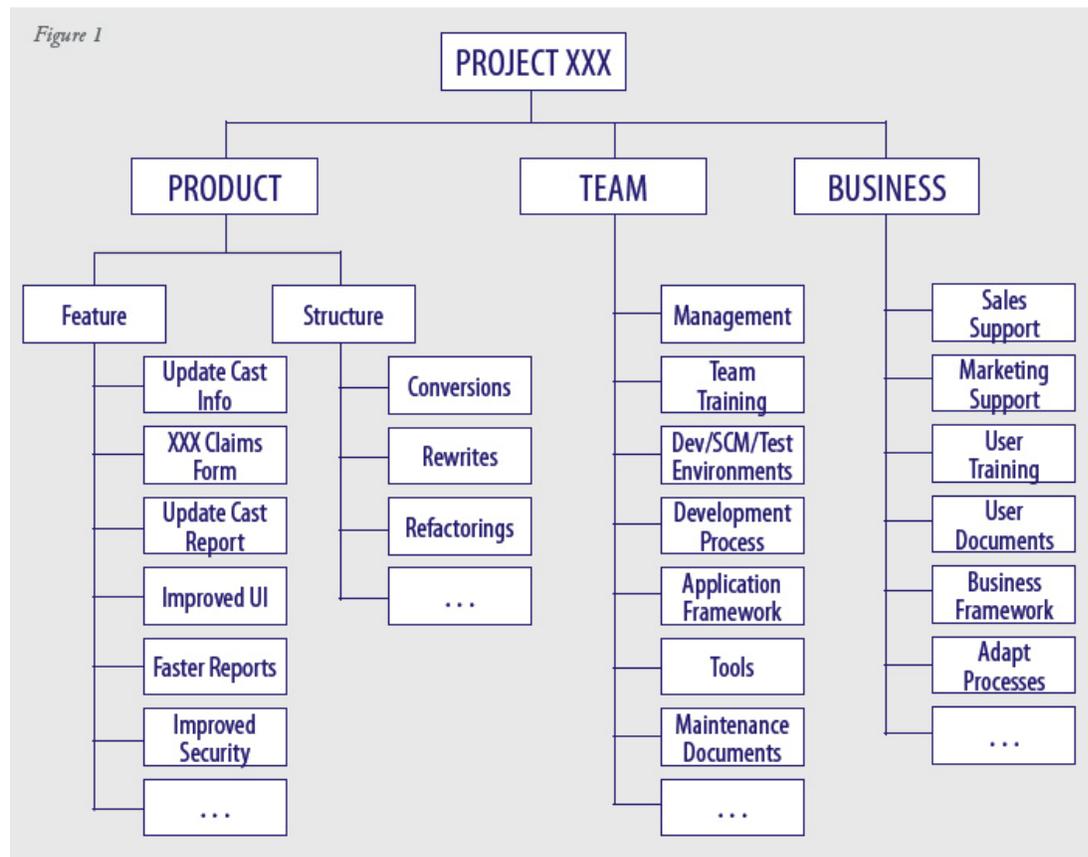
The most valuable metrics provide management with a “big picture” perspective of the team’s work. Although many Agile users cringe at its “high ceremony” connotations, the Work Breakdown Structure (WBS), commonly used in standard project management, gives such a view. Defined as a “deliverable-oriented grouping of project elements which organizes and defines the total scope of the project,” the WBS is a convenient way to categorize stories. The WBS I use organizes work into three areas:

- Product: Work that is directed to actually produce the software product.
- Team: Work that allows/enables the team to produce the product.
- Business: Work that allows the business to market, sell, install, or deliver the product to users.

Because every project is different, the actual structure of the WBS (what “groups” there are) is malleable and can be modified to suit an individual organization or project. The stories that live at the bottom of these groups constitute the work being done. A WBS for a fictional development project appears in Figure 1. We will use it to calculate business value.

CALCULATING BUSINESS VALUE

For illustrative purposes, Figure 1 provides the business weight of each WBS group, including stories (the ones at the bottom). Also, we note that the Business Value metric we are calculating is actually a percentage, not a dollar value.

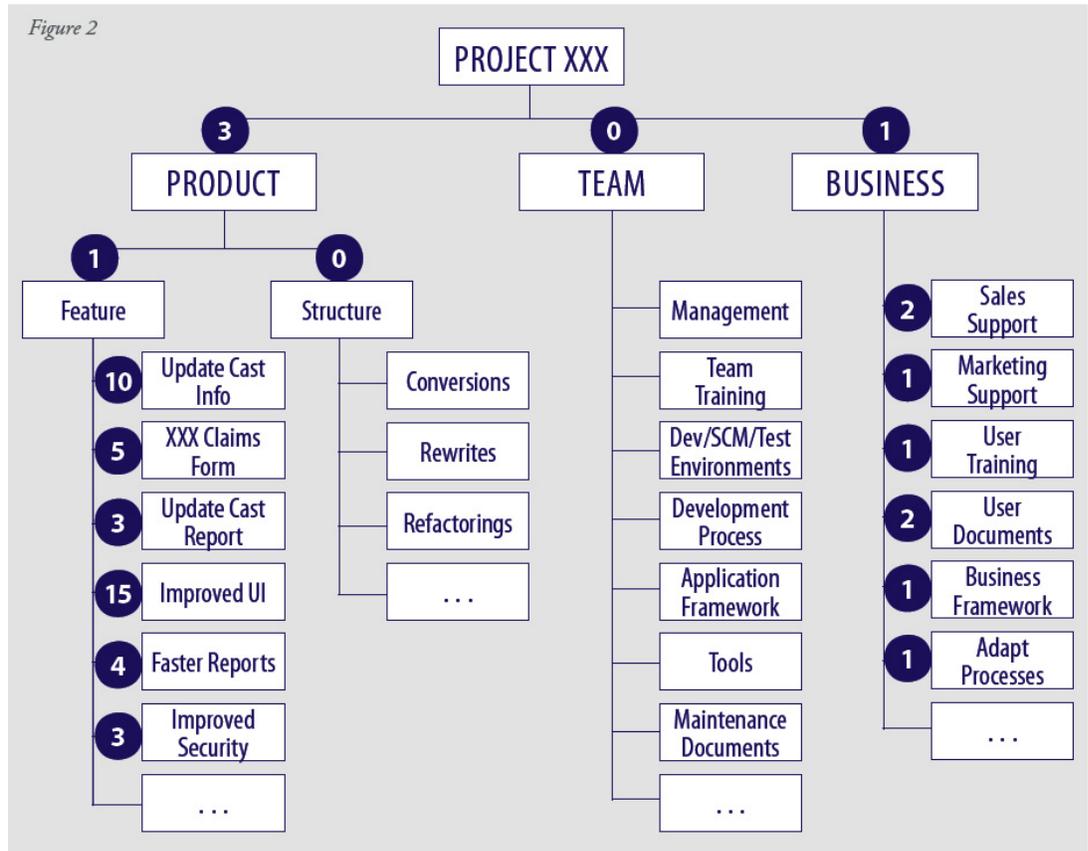


To begin, we will first assign additive weights to the groups of the WBS, which represent features and other organizations of stories. In the simplified view of this article, business value can be generated in the WBS by either adding features to the code or by any activity that enables the business to market or sell the product - there is no other opportunity for business value elsewhere in the WBS.

The assigned weights on the WBS are all relative; each group is compared to its siblings. As shown in Figure 2, the WBS with its assigned weights gives us the following information:

- The Product Leg is worth three (3) times as much as the Business Leg, while the Team Leg provides no business value.
- Adding features to the product supplies business value, but modifying or cleaning up code that already works does not.

- Improving the User Interface is the most important feature, worth one-and-a-half (1.5) times as much as the ability to Update Customer Information.
- User Documentation is the most important business bucket, but is still not worth much in comparison to any of the features.



Assigning weights to the WBS legs and stories is not a technical matter and should be handled by either the business or the customer with the single stipulation that the weights be additive. Beyond that, there is no prescriptive formula or requirements. For example, does “Team Training” possess business value or not? The answer is that it simply depends upon the project. Typically, it doesn’t provide value if paid for out of project funds, but might if paid for out of overhead (training) funds.

Once weights have been assigned to the WBS legs and buckets, we assign additional additive weights to the stories within each WBS bucket. For the stories we find in the “Update Customer Information” feature, their weights might be similar to those shown in Table 1.

Table 1

WT Story	
0	Determine what the stakeholders want
10	Develop the happy path
0	Determine alternative paths
3	Develop alternative path 1
2	Develop alternative path 2
5	Stress-test the feature for 500 simultaneous users

Note that the analysis stories have no weight; they only exist because they provide the development stories that do have business value.

Given this assignment of weights, we are now ready to calculate the Business Value (BV) of any WBS bucket, including stories. The idea is simple and contains two parts. First, the value of the

whole “tree” is 100 percent (or 1), which means that all the work gives all the value. This value of 1 is assigned to the top of the tree. Second, as we move down the tree, each bucket’s value is the appropriate percentage of its parent’s value, as compared to its “siblings” (the other children of its parents). This is a multiplicative formula because it’s based on taking percentages of percentages as we move down the tree. The calculation is recursive and can be denoted by the following formula:

$$BV(bucket) = BV(parent) \times \left[\frac{wt(bucket)}{wt(bucket) + \sum_{siblings} wt(sibling)} \right]$$

Let’s try an example and calculate the value of the Main Execution Flow of the “Update Customer Information” feature. According to Table 1 and the WBS with assigned weights, the feature leg is worth three-quarters (3/4) of the project’s total value; the “Update Customer Information” feature is worth ten-fortieths (10/40) of the feature leg; and the Main Execution Flow is worth ten-twentieths (10/20) of feature’s total value. Therefore, the Business Value of the Main Execution Flow of the “Update Customer Information” feature is (1)x(3/4)x(1/1)x(10/40)x(10/20)=9.4% of the project. Remember: This is a multiplicative formula, taking percentages of percentages as we move down the tree. If we were to add another feature with a weight of ten (10) (say “Account Report”), this would drop the percentage to seven-and-a-half (7.5), as the “(10/40)” piece of the formula becomes “(10/50).”

EARNED BUSINESS VALUE

Once I have calculated Business Value (BV), I can then generate an Earned Business Value (EBV), which I define as “the percentage of the known business value that is coded up and running.” In other words, EBV is the sum of all the business values for those stories that are complete - that have earned their business value.

$$EBV(Project) = \sum_{completed} BV(Story)$$

This calculation consists of two parts:

- A subjective part that determines the weights allowing us to calculate the BV; and
- An objective part that determines whether or not a story is “complete,” which is merely a “yes” or “no” (1 or 0). Because stories are small, it makes less sense to measure a percentage complete than to simply denote a story as “done or not.”

While the subjective first part is determined by the additive weights, the second is similarly straightforward. Once a story is defined, its validation criteria decide whether it is complete (and has earned its BV).

SUMMARY

For managers who are fearful that Agile practices cannot support adequate metrics, a functional Work Breakdown Structure (WBS) provides a structural framework for calculating Agile-specific business metrics. Both Business Value and Earned Business Value effectively replace the Earned Value Analysis (EVA), delivering numerous benefits including:

- The combination of standard burndown charts and EBV charts provides a clear, composite understanding of the progress of the project.
- The business can observe when a project is yielding diminishing returns, i.e. when the business value produced is less than the cost of the work being done. This could indicate that the project needs to move into a stabilization phase and release.
- If the value of a project is known in relation to the overall help of the business, these metrics provide a way to compare projects to determine whether resources should be allocated.

ABOUT THE AUTHOR



Daniel Rawsthorne

Dan Rawsthorne is a certified Scrum Trainer and one of CollabNet's transformation coaches, helping organizations transition to Agile practices. Rawsthorne is a 27-year veteran of the software industry, having worked in a wide range of capacities, from coder to project manager, for companies large and small. In addition to possessing deep knowledge of many software processes, procedures, and techniques, he holds a Ph.D. in mathematics. More of Rawsthorne's thoughts on Agile development and implementation can be found at his blog: <http://www.collab.net/blog/danrawsthorne>.

ABOUT COLLABNET

CollabNet leads the industry in Agile application lifecycle management (Agile ALM) in the Cloud. The CollabNet TeamForge™ ALM platform, CollabNet Subversion software configuration management (SCM) solution, and ScrumWorks® project and program management software enable teams using any environment, methodology, and technology to increase productivity by up to 50% and reduce the cost of software development by up to 80%. The company also offers training, including Certified ScrumMaster training, software development process improvement services, and an innovative community management approach to driving enterprise development success. As the founder of the open source Subversion project, CollabNet has collaborative development for distributed teams in its DNA. Millions of users at more than 2,500 organizations, including Applied Biosystems, Capgemini, Deutsche Bank, Reuters, and the U.S. Department of Defense, have transformed the way they develop software with CollabNet. For more information, visit www.collab.net.

REFERENCES

1. See <http://office.microsoft.com/en-gb/assistance/HP010342581033.aspx> for a discussion of these terms in the context of Microsoft Project, which is one of the most popular project management tools on the market.
2. Since this paper was originally written, a version of EVA metrics applicable to agility has been introduced. See Sulaiman, Barton, and Blackburn, AgileEVM - Earned Value Management in Scrum Projects.
3. Jeffries, Anderson, and Hendrickson. Extreme Programming Installed. Addison-Wesley. 2001. P. 29.
4. Project Management Body of Knowledge (PMBOK). See <http://www.hmaster.com/glossary.html>.