AGILE'S EARNED SCHEDULE BASELINE
By Robert Van De Velde, PhD

ABSTRACT

For managing schedule performance on an Agile project, canonical Agile techniques fall short. Estimates are essential for meeting project time constraints, but Agile's relative estimates are too "fuzzy" to meet the needs of Sprint planning. Typical Agile Burndown charts focus on Release Point counts that only suggest how well or poorly the schedule is performing. The addition of Earned Value helps, but EVM's traditional schedule metrics are inadequate. Earned Schedule fills the gaps. It provides a robust baseline to measure past schedule performance and to estimate future impact on delivery date and performance level.

For plan-driven projects, schedule baselines are a given. For Agile projects, baselines are suspect. The difference is evident in the charters of the two approaches.

On one hand, the Project Management Institute states that the schedule baseline is the time-phased plan against which project execution is measured and managed ("PMB", 2013, p. 549). Without the baseline, there is no basis for measuring and managing schedule performance.

On the other hand, both the Agile Manifesto (Beck et al., 2001a) and Principles (Beck et al., 2001b) omit any reference to baselines. The Manifesto takes the omission a step further by stating that change is valued over following a plan. As the schedule baseline is, by definition, a plan, it is clearly not a high priority for Agile.

The different perspectives are rooted in disparate objectives. Canonical Agile projects seek the early and continuous delivery of the Product Vision. Orthodox plan-driven projects seek the on-time and on-budget delivery of the project objectives.

Schedule and cost constraints make a crucial difference. To deliver on-time and on-budget, you need credible targets for timeline and funding. Baselines embody the targets.

Whether or not Agile projects are ever free of such constraints is debatable. That debate is not the focus here. Instead, the focus is on those projects that use the Agile framework but are also bound by constraints, specifically the time constraint.

DEFINING THE SCHEDULE BASELINE

In Agile projects, a schedule baseline can be derived from velocity. Velocity is the number of Release Points (aka, Story Points) for each Sprint.

How is the velocity set? Again, there are challenges from the canonical Agile view. As the Agile Alliance puts it: "phrases such as 'setting the velocity' reveal a basic misunderstanding" ("Velocity", n.d.). In that view, velocity is a retroactive measurement, something done after the fact. It is not a forward-looking estimate of the future completion rate.

The objection is another consequence of omitting time as a constraint. Without that constraint, you simply measure the number of Release Points completed in each Sprint. At most, that number can be kept in mind as Product Backlog Items are selected for the next Sprint.

In a time-constrained Agile project, the "next Sprint" is not enough. Velocity measures must reach beyond the next Sprint to cover the whole project timeline. Without that scope, you cannot reasonably commit to a delivery date. With that scope, you need an estimate, specifically an estimate that encompasses the whole timeline.

SCRUM ESTIMATES

There are some variants of Agile that advocate forward-looking velocity estimates. For instance, the Scrum framework uses techniques such as T-shirt sizing, Planning Poker, and Fibonacci bucketing to produce estimates (Sil, 2013).

Scrum estimates are for the relative size of Product Backlog Items. The size is determined by comparing Items in the Backlog: Item A is larger than Item B which is larger than Item C. But, what dimension is being sized? That is not so clear. Some say it is the amount of complexity; others say it is cost; still others say it is uncertainty.

In the end, Scrum's relative estimates are viewed as replacements for absolute Work Hour estimates ("Relative Estimation", n.d.). So, complexity, cost, and uncertainty are just considerations used to size work effort.

Relative estimates often differentiate sizes by assigning numbers from a geometric series or from the Fibonacci sequence. That gives the appearance of quantifying the differences. For instance, an Item assigned an 8 is larger than one assigned a 2.

Many proponents of Agile do not stop there. They say that relative estimates also indicate how much larger one Item is than another (Singh, 2016). For instance, an Item assigned an 8 is four-times larger than one assigned a 2.

We have found that Agile teams often agree on the order between Items. But, we have observed frequent disagreement over how much difference there is between Items. Is an Item labeled with a Fibonacci number of "21" really 7 times larger than one labeled as "3"? Is an Item labeled with a geometric series number "3" really one-third the effort of a "9"?

LIMITS OF RELATIVE ESTIMATES

The disagreements reflect divergent beliefs about size. The situation is similar to one that occurs in social science and marketing research. There, Likert scales measure psychological states such as levels of satisfaction (e.g., with a product or service). The comparative levels are often associated with numbers from 1 to 5, as depicted in Figure 1.[1]

Figure 1

Likert scales have been studied extensively (Stevens, 1946; Michell, 1986; Sauro, 2011). The studies have raised questions such as: do the numbers have an objective numerical basis, and are the intervals between levels equal? The upshot is that the scale represents subjective states and that such states cannot be objectively measured. So, we cannot be sure that the difference in assigned numbers is the same as the attribute they represent.

Similarly, Agile team members use numbers to express beliefs about the relative size of work effort. But, what one person believes is twice as much effort may differ from what another person believes is twice the effort. So, although 4 is twice the size of 2, we cannot be sure that everyone on the team means the same thing when they assign "4" to an Item. The most that we can be sure of is that the Item is larger and more-or-less twice the size of a "2".

WHY "FUZZINESS" MATTERS

Does the "fuzziness" matter? Yes, it matters. The estimates are used in Sprint planning. Sprint planning, in turn, is important for meeting time constraints. Backlog Items are selected to fit into a Sprint based in part on the estimated velocity of the team. Without that guide, the Sprint goal might be set too high or too low. Either way, project time commitments would be undermined.

For selecting Items of the right size, relative estimates fall short. Using fuzzy estimates is like driving through a town that has posted its speed limit as "35-ish". If you are not worried about time—just drive at 15, and you should be OK. Or, if you do not care about safety, drive at 55 but recognize that you might crash.

If time matters, you need to know how fast you can go without breaking the limit but still getting through as quickly and safely as possible. That is what a cardinal estimate tells you. It

1) Technically, there is a difference between a Likert scale and a Likert item (Vanek, 2012; Uebersax, 2006). The item is what a survey respondent is asked to evaluate, e.g., level of satisfaction with a recently purchased product. In well-formed surveys, there are multiple items intended to reveal the respondent's underlying psychological state. The Likert scale is the sum of all the items. So, in a survey with 5 items and responses ranging from 1=Very Dissatisfied to 5=Very Satisfied, the Likert scale is 5 to 25. Speaking precisely, therefore, Figure 1 represents the format of a Likert-type item.

goes beyond the subjective state to something that can be objectively measured. It provides a clear baseline for assessing performance on past Sprints and planning future ones. That is why we cannot stop the estimating process with relative estimates. Instead, we need to develop cardinal estimates.[2]

LIMITS OF RELEASE POINT BASELINES

Armed with cardinal velocity estimates, the Release Point baseline can be set. It is runs from the Release Point total to zero. The end of each Sprint marks completion of an increment of estimated velocity. For comparison, the number of Release Points actually completed at the end of each Sprint is used to decrement the Release Point total. A burn chart is generally used to illustrate the estimated and actual velocity. Figure 2 illustrates the chart, specifically the Burndown Chart, from a recent project.
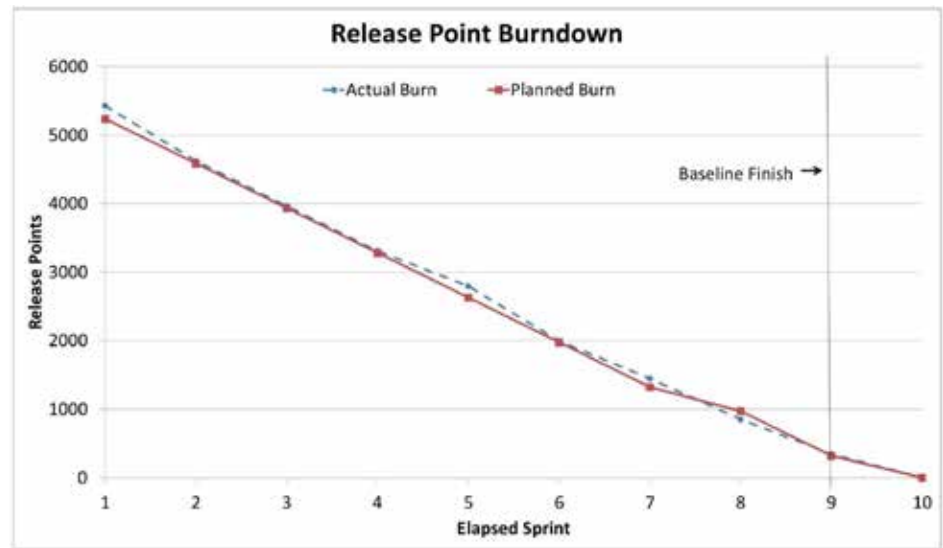

Figure 2

Schedule performance is inferred from the relationship between the Actual Burn line and the Planned Burn line. When the Actual Burn line is above the Planned Burn line, it suggests that schedule performance is lagging. When the Actual Burn line is below the Planned Burn line, it suggests that schedule performance is better than expected. Finally, the size of the gap between the two lines suggests the level of schedule performance efficiency.

It is important to note that the chart only suggests how well the schedule is performing. The data points in the chart are not units of time but are, instead, Release Point counts. There is no quantification of schedule performance efficiency. Finally, there is no estimate of the performance impact on delivery date. The Release Point Baseline and its pictorialization, the burn chart, are good tools for a quick reading on the situation, but they omit key pieces of information.

One reason they fall short is that Release Points do not measure value. On time-constrained projects, we need to know not only the number of Points but also the value of those points. That knowledge enables us to see beyond the completion of low value items that make burn charts look good but are not really moving the project forward.

Note that the term "value" here reflects the efficiency of execution rather than its effectiveness. That is, it is Earned Value rather than Business Value (Alleman, 2011; Alleman, 2009). Earned Value Management (EVM) tells us the budgeted cost of the planned velocity and the amount of that budget earned by the actual velocity. The traditional EVM measure of schedule performance, the Schedule Performance Index (SPI), is the ratio between the cumulative earned value and total planned value.

Unfortunately, traditional EVM measures of schedule performance are inadequate. After a project is about two-thirds complete, the SPI rises inexorably to a perfect 1.0. Even if the project finishes late, the SPI ends at 1.0—a counterintuitive indication of performance. Furthermore, traditional EVM measures do not include an estimate at completion for time (EACt) or an estimate of the schedule performance level required to complete on time (TSPI).[3] Finally, traditional EVM schedule performance measures are framed in terms of dollars, rather than units of time.

Fortunately, Earned Schedule has closed the gaps.

2) Discussion of techniques for producing cardinal estimates in Agile projects is outside the scope of this paper. Glen Alleman has many instructive posts on the topic in his blog (see Alleman, 2015a; Alleman, 2015b; and Alleman, 2017).

3) There are extensions to traditional EVM that include formulas for EACt (e.g., Anbari, 2001). The extensions are undermined by reliance on SPI.

EARNED SCHEDULE

The amount of time earned on a project is defined as "the time at which the value currently earned should have been earned" (Lipke, 2009, p. 14). The definition neatly ties planned value and earned value into time, framing ES metrics in units of time rather than cost.

The calculation of Earned Schedule is simple. Count the number of Sprints in which the current total of Earned Value is greater than or equal to the cumulative Planned Value.[4] Usually, some Earned Value remains after the last full Sprint is counted. The fractional time earned equals the ratio between the left-over Earned Value and the Planned Value for the next Sprint beyond the last full one.

Schedule performance efficiency for time (SPIt) is calculated as the ratio between the amount of schedule earned and the actual time. Based on this efficiency, the EACt is estimated as the ratio between the Planned Duration and SPIt. Finally, the TSPI is the ratio between the Planned Duration less the Earned Schedule and the Planned Duration less the Actual Time.

Studies have repeatedly demonstrated that Earned Schedule metrics are superior to other EVM schedule performance measures (Vanhoucke and Vandevoorde, 2007; Lipke, 2008; Crumrine and Ritschel, 2013). In our own practice, we have used the metrics successfully on many projects—some of which used the Agile framework. We have found ES metrics to be especially useful in managing hybrid project portfolios comprising both plan-driven and Agile projects.

EARNED SCHEDULE BASELINE

The Earned Schedule baseline is framed by the pace of Release Point delivery. Cardinal estimates for the number of planned Release Points and the mean velocity determine the number of Sprints. Given the project start date and number of Sprints, the planned finish date is set. The Earned Schedule baseline fits into that frame as follows: for each elapsed Sprint, one Sprint should be earned.

If all Release Points carry the same value, the mean velocity ensures that the periodic Planned Value is the same for each planned Sprint. It follows that the periodic amount of planned Earned Schedule is the same for each planned Sprint. Expressed as a burn chart, the baseline runs straight from the end of the first Sprint to the end of the last planned Sprint.

It is possible for Release Points to carry different values. For instance, if Release Points express Work Hours, their value might vary depending on the resource responsible for delivering them. In such cases, the baseline is still one earned Sprint for each elapsed Sprint, but there's a catch.

If Release Points carry different values, a mean Planned Value should be set for all Sprints and used to guide the selection of Items for the next Sprint. A constant Planned Value ensures that the periodic Earned Schedule is the same across all planned Sprints. The ES baseline, therefore, will be isomorphic with the Release Point baseline, providing common ground for comparison.

By contrast, the actual ES burn will usually be non-linear. The amount of schedule earned generally varies from Sprint to Sprint. At the end of each Sprint, the total number of Sprints is decremented by the total amount of schedule earned. So, in a burn chart, the ES burn line typically weaves around the ES baseline.

Figure 3 illustrates the ES Burndown from the same project used for Figure 2. The ES baseline runs straight from the beginning to the planned finish. The ES Burn line runs on or above the baseline. The chart is enhanced with a table displaying additional ES metrics.

---

4) A Sprint is defined as a time-boxed unit of delivery. All Sprints in a project have the same duration, normally 1 week to 1 month. Calculating ES in terms of Sprints is, therefore, equivalent to calculating ES in units of time.
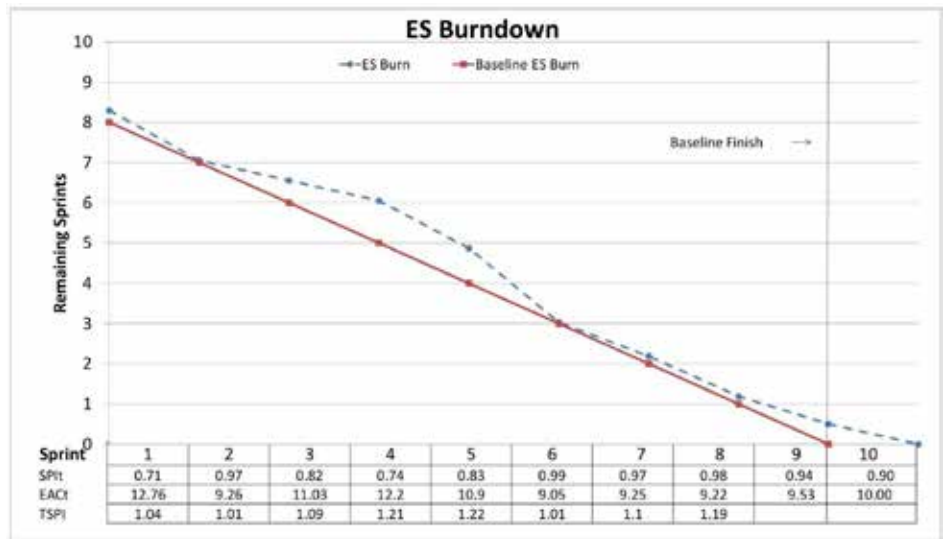
ES Burndown

Sprint	1	2	3	4	5	6	7	8	9	10
SPIt	0.71	0.97	0.82	0.74	0.83	0.99	0.97	0.98	0.94	0.90
EACt	12.76	9.26	11.03	12.2	10.9	9.05	9.25	9.22	9.53	10.00
TSPI	1.04	1.01	1.09	1.21	1.22	1.01	1.1	1.19		

Figure 3

INTERPRETING THE ES BURNDOWN CHART

The ES Burndown Chart is easy to interpret. If the ES Burn line is above the ES baseline, schedule performance lags the estimate. If the ES Burn line is below the baseline, performance is ahead of the estimate. If it is on the line, schedule performance is exactly on the estimate. Because the data points represent units of time, the chart explicitly shows how time is being used on the project.

In Figure 3, the chart makes it clear that schedule performance is lagging in most Sprints. The performance deficit is clearest when a large gap opens at Sprints 2 through 5. By contrast, Figure 2 suggests that the project is generally running on or slightly behind schedule. (In Sprint 8, it even appears that the project is slightly ahead of schedule.) Given that the project actually finished after the Baseline Finish, the ES Burndown is a more accurate representation of schedule performance.

The project's backstory helps explain the chart. To build momentum, the project team decided to spend a couple of Sprints working on quick deliverables, even though they were low value and did not align with the mean Planned Value. The quick deliverables preserved Release Point production at the estimated rate but delivered less Earned Value than planned. Hence, the Release Point chart tracked close to plan, and the ES chart did not.

The table included with the chart quantified how time was being used on the project. When the SPIt was below 1.0, schedule performance was lagging. If the SPIt had been over 1.0, schedule performance would have been better than expected. As commonly happens, the SPIt was never exactly equal to 1.0.

Early in the project, as the team built momentum, the SPIt suffered. Eventually, the team returned to the original plan, and the SPIt improved. The improvement did not, however, recover all of the lost time. The SPIt accurately reflected performance throughout the project lifecycle. When the project exceeded the Baseline Finish, the SPIt ended below 1.0.

The EACt and TSPI threw additional light on schedule performance. The Baseline Finish had been set at Sprint 9 with a Committed Finish at Sprint 10.[5] The EACt consistently showed that, given actual performance levels, the Baseline Finish would not be met.[6]

The TSPI re-enforced that view. In several Sprints, the TSPI exceeded the commonly accepted threshold for recoverability (i.e., a value of 1.1 as per Lipke, 2016; Lipke, 2009, pp. 90-91). Unsurprisingly, it was beyond the threshold at Sprint 8, just before the TSPI became undefined.[7]

The most positive signal in the metrics surfaced after Sprint 5. The EACt began to show that the Committed Finish would be met.

In the example, although schedule performance was not uniformly bad, it was never good enough to meet the Baseline Finish. The ES Burndown Chart signalled the potential for delay, and the associated metrics quantified the implications.

5) The Baseline Finish includes Contingency but excludes Reserve. The Committed Finish includes both (see Van De Velde, 2017). Often, Contingency and Reserve are limited to allocations of money. They should also cover allocations of time.
6) It is also possible to subject the EACt to statistical analysis. The likely range of a project's EACt values can be calculated based on its historical SPIt metrics (see Van De Velde, 2015a-2015c).
7) By definition, the TSPI becomes undefined once the Actual Duration reaches the Planned Duration.

CONCLUSION

On time-constrained Agile projects, the Earned Schedule Baseline provides a robust yardstick for measuring schedule performance. Using the ES Baseline entails divergence from both canonical Agile practice and plan-driven methodology. The discrepancies must be acknowledged, but in the end, the value that Earned Schedule brings to Agile projects makes the departures worthwhile.

REFERENCES

Alleman, G.B. (2009, January 10). Some Serious Misunderstandings About Earned Value [Blog post]. Retrieved from http://herdingcats.typepad.com/my_weblog/2009/01/some-serious-misunderstandings-about-earned-value.html

Alleman, G.B. (2011, August 02). Measuring Business Value [Blog post]. Retrieved from http://herdingcats.typepad.com/my_weblog/2011/08/measuring-business-value.html

Alleman, G.B. (2015a, March 10). Criteria for a "Good" Estimate [Blog post]. Retrieved from http://herdingcats.typepad.com/my_weblog/2015/03/criteria-for-a-good-estimate.html

Alleman, G.B. (2015b, December 26). The Story Point Problem [Blog post]. Retrieved from http://herdingcats.typepad.com/my_weblog/2015/12/the-story-point-problem.html/

Alleman, G.B. (2017, April 6). Velocity versus Speed (Update) [Blog post]. Retrieved from http://herdingcats.typepad.com/my_weblog/2017/04/velocity-versus-speed.html

Anbari, F.T. (2001, November). Applications and Extensions of the Earned Value Analysis Method. In *Proceedings of the Project Management Institute 2001 Seminars & Symposium*, 1-10. Nashville, TN, USA. Newtown Square, PA: Project Management Institute. Retrieved from http://www.pmi.org/learning/library/earned-value-analysis-forecast-outcomes-7898?id=7898

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber K., Sutherland, J., and Thomas, D.. (2001a). Agile Manifesto. Retrieved from http://agilemanifesto.org/

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber K., Sutherland, J., and Thomas, D.. (2001b). Principles Behind the Agile Manifesto. Retrieved from http://agilemanifesto.org/principles.html

Crumrine, K. and Ritschel, J.D. (2013). A Comparison of Earned Value Management and Earned Schedule as Schedule Predictors on DoD ACAT I Programs. *The Measurable News*, 2013(2), 37-44. Retrieved from http://www.earnedschedule.com/Docs/Crumrine%20Article%20-%20Measurable%20News%20(May%202013).pdf

Lipke, W. (2008). Project Duration Forecasting: Comparing Earned Value Management Methods to Earned Schedule. *CrossTalk*. Retrieved from http://www.earnedschedule.com/Docs/Project%20Duration%20Forecasting%20%20-%20Walt%20Lipke.pdf

Lipke, W. (2009). Earned Schedule. Raleigh, North Carolina: Lulu Publishing.

Lipke, W. (2016). Examination of the Threshold for the To Complete Indexes. *PM World Journal*, V (III). Retrieved from http://www.earnedschedule.com/Docs/PMWJ-Examination%20of%20Thresholds%20for%20To%20Complete%20Indexes.pdf

Michell, J. (1986). Measurement Scales and Statistics: A Clash of Paradigms. *Psychological Bulletin*, 100 (3), 398-407.

PMB (i.e., Performance Measurement Baseline). (2013). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Newtown Square, Pa: Project Management Institute. Note: The phrase "time-phased plan" is from Baseline. (2004). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Newtown Square, Pa: Project Management Institute, p. 352.

Relative Estimation. (n.d.) Retrieved from https://www.agilealliance.org/glossary/relative-estimation/

Sauro, J. (2011, June 01). Should You Care If Your Rating Scale Data Is Interval Or Ordinal? Retrieved from https://measuringu.com/interval-ordinal/

Sil, K. (2013, November 13). How to Estimate Quickly and Efficiently. Retrieved from https://www.scrumalliance.org/community/articles/2013/november/success-story-how-to-estimate-quickly-and-efficien

Singh, V. (2016, January). Agile Estimation Techniques. Retrieved from https://www.scrumalliance.org/community/articles/2016/january/agile-estimation-techniques/

Stevens, S.S. (1946). On the Theory of Scales of Measurement. *Science*, 103 (2684), 677-680.

Uebersax, J.S. (2006). Likert Scales: Dispelling the Confusion [Blog post]. Retrieved from http://www.john-uebersax.com/stat/likert.htm

Van De Velde, R. (2015a, October 31). ES Statistical Analysis [Blog post]. Retrieved from http://www.projectflightdeck.com/cESExchange000.php?blog_archive=2015-10

Van De Velde, R. (2015b, November 11). ES Statistical Analysis in Action [Blog posts]. Retrieved from http://www.projectflightdeck.com/cESExchange000.php?blog_archive=2015-11

Van De Velde, R. (2015c, December 8). ES Statistical Analysis Pro and Con [Blog post]. Retrieved from http://www.projectflightdeck.com/cESExchange000.php?blog_archive=2015-12

Van De Velde, R. (2017, February 27). ES for Agile Projects: Step 2 Baseline the Schedule [Blog post]. Retrieved from http://www.projectflightdeck.com/cESExchange000.php?blog_archive=2017-02

Vanhoucke, M. and Vandevoorde, S. (2007, October). A Simulation and Evaluation of Earned Value Metrics to Forecast the Project Duration. *Journal of the Operational Research Society*, 58, 1361-1374.

Vanek, C. (2012, April 24). What is a Likert Scale? Retrieved from https://www.surveygizmo.com/survey-blog/likert-scale-what-is-it-how-to-analyze-it-and-when-to-use-it/

Velocity. (n.d.) Retrieved from https://www.agilealliance.org/glossary/velocity/

ABOUT THE AUTHOR

Robert Van De Velde owns and operates ProjectFlightDeck.com, a company focused on Earned Schedule products and services. As a project manager, Rob has a 30-year track record of delivering IT programs and projects in a variety of domains. He has successfully used ES on both plan-driven and Agile projects. Rob holds a PhD, a PMP, and a Black Belt in MS Project. In 2014, he became a Certified Scrum Master. Rob posts regularly on LinkedIn's ES group and on his blog, EarnedScheduleExchange.com.

CONTACT INFORMATION

Robert Van De Velde
Owner/Operator
ProjectFlightDeck.com
Robert.VanDeVelde@ProjectFlightDeck.com
3416 Sawmill Valley Drive
Mississauga, Ontario
Canada   L5L 3A4
(905) 828-0508