UMD Project Management Symposium
The Art of Data-driven Forecasting

William W. Davis[1]

[1]William W. Davis, MSPM, PMP,
5000 NW 5th Avenue,
Boca Raton, FL 33431; famousdavispmp@gmail.com

ABSTRACT

Forecasting when an agile-developed product or service might be released, or how much of the product or service might be ready by a hard delivery date, requires both art and science. The science of forecasting includes statistics calculated from an agile team's historical record of past performance and a statistical model to forecast the uncertainty of the team's future performance. The art of forecasting includes the team's estimates of all remaining work and the subjective judgment of the forecaster. However, forecasters must recognize their own cognitive biases which can adversely affect their forecasts. In this whitepaper, six questions are raised to assist the agile development forecaster in deciding when to apply subjective judgment to adjust or override statistics gathered from historical records.

INTRODUCTION

Forecasting is a critical business capability that helps organizational leaders evaluate return-on-investment (ROI) options and select project portfolios that optimize ROI and the use of scarce organizational resources. In traditional project management, project teams decompose project charter scope statements into a work breakdown structure that later informs the project team's detailed schedule and budget estimates.

In agile project management, project sponsors often agree to fund an agile team for a set duration that is aligned with the project charter's schedule constraint, which itself is a function of the organization's judgment of how long it will take to achieve some larger goal. The resulting project budget is simply a product of the agile team's average weekly cost and the number of weeks that the sponsor agrees to fund the agile team (with necessary adjustments for other costs, such as hardware, software licensing, or costs related to the project endeavor).

Value-stream funding—where an organization chooses to fund value streams rather than projects—is like agile project funding because the organization chooses to fund multiple value-streams for a given time period, such as quarterly, semi-annually, or annually. This whitepaper refers to "agile projects" but applies equally to non-project, value-stream management, too.

While both the *schedule* and *cost* constraints are, for agile development efforts, often inflexibly held, the *scope* constraint is usually the most flexible constraint of the project's so-called "Triple Constraint." Since agile projects expect work to emerge over time, and agile teams decompose only the work that they will imminently bring into their upcoming sprints or iterations, neither the project sponsor nor the agile team can say with certainty *what* the team will deliver or *when* they will deliver completed scope items.

This uncertainty over what will be delivered and when it will be delivered is a source of anxiety for most project sponsors, product managers, product owners, and other interested stakeholders. Product owners using Scrum—the most popular agile method for creating products, services or other results (State of Agile, 2020)—rely on forecasts to maximize the value of a Scrum team's development efforts. Project sponsors and organizational leaders also use forecasts to assess whether the project charter's constraints should be firmly held or relaxed, and where the organization's scarce resources should be deployed now and in the future.

Forecasting an agile development effort is essentially no different than any other type of forecasting. The goals of forecasting are always the same: to align expectations about an uncertain future, and to make better decisions today to optimize the likelihood of maximizing desirable future outcomes.

Forecasters of agile projects use algorithms, historical data and statistics to prognosticate the future performance of an agile team—just like forecasters from any other industry would do. That is because judgmental forecasting without *science* is a largely fruitless effort. Phillip Tetlock, a political scientist, authored a landmark study that found judgmental forecasts alone are usually of dubious accuracy (Tetlock, 2005). In fact, "Human beings who spend their lives studying the state of the world, in other words, are poorer forecasters than dart-throwing monkeys…" (Menand, 2005).

Tetlock's study does *not* mean that all judgmental forecasting is meritless, however. When a forecaster has access to information that is not evaluated by a statistical model and has made similar forecasts within a stable environment, the result informs decision-makers (Kavanagh and Williams, 2014). Good forecasting, then, starts with a good-fitting statistical model and then adjusts the model's algorithms by carefully applying expert judgment to further enhance both the forecast's informativeness and accuracy.

This paper will explore the *art* of data-driven forecasting within an agile product development context using Scrum. Using the built-in statistical functions inside Microsoft Excel® and a freely licensed, pre-built spreadsheet called Statistical PERT® (Normal Edition), an agile forecaster can use a pre-built algorithm to forecast product delivery dates (the *science* of forecasting). Then, the forecaster can employ subjective judgment to improve their forecast while being cognizant to avoid or mitigate common cognitive biases that can degrade the quality of their forecast (the *art* of forecasting).

FORECASTING OBJECTIVES

If the overarching *goals* of forecasting are to align expectations and to make better decisions that maximize the likelihood of desirable future outcomes, then what are the specific *objectives* of forecasting future outcomes?

The forecasting function can have numerous objectives based upon the forecasting context. For instance, meteorologists (weather forecasters) create forecasts with the objective of informing people so they can make good decisions considering upcoming weather changes, especially when a weather storm is threatening property and human life. Operational forecasts help an organization anticipate demand so the firm can ensure it has enough inventory and production capacity to meet that demand.

With agile-developed product forecasting (referred to henceforth as "agile forecasting"), the objective usually is to estimate the delivery date of new product features, or to estimate how many new features an agile team can deliver by an inflexible delivery date. Product owners rely on agile forecasts to align customer and stakeholder expectations and to modify product feature sets that satisfy delivery date constraints whenever they exist.

DATA-DRIVEN AGILE FORECASTING

Agile team forecasters rely on the team's actual performance of converting product backlog items to completed work requests. An agile team may use different units of measure to calculate "done" work requests. Story points, story cards, and ideal days are three such ways that agile teams size their work items. Importantly, agile forecasting does not stipulate the unit of measure to use; any of these units of measure can work well.

In the University of Maryland's 2020 Project Management Symposium, William W. Davis explained how to use a pre-built Microsoft Excel® spreadsheet called Statistical PERT® to create an agile forecast using an agile burn-up chart (Davis, 2020). This visual, data-driven, and probabilistic approach to forecasting requires the forecaster to have access to the agile team's historical record of completed work items from each finished iteration or sprint, and an estimate of all remaining work on the team's product backlog. Excel's built-in statistical functions for the normal probability distribution and Excel's charting capabilities create a visual forecast—a burn-up chart—that sponsors and stakeholders can easily interpret.

The algorithms used in the Statistical PERT spreadsheet are the *science* behind agile forecasting using this particular forecasting technique. But forecasting is a blend of *science* and *art*, and agile forecasters need to employ subjective judgment to improve their forecasts beyond what a data-only forecast can do. However, applying subjective judgment to adjust a statistical model—in hopes of improving the model's informativeness and accuracy—comes with its own set of perils: cognitive biases. Cognitive biases inject irrational conclusions into any kind of forecast that human beings create (Kavanagh and Williams, 2014).

COGNITIVE BIASES

Human beings suffer from a multitude of cognitive biases. Among dozens of cognitive biases that social science has identified since the 1974 seminal work by social scientists Amos Tversky and Daniel Kahneman, two of the most impactful to forecasters are overconfidence in range estimation (*overconfidence*) and *anchoring* (Welsh and Begg, 2018).

With *overconfidence*, forecasters use range estimates that too narrowly describe all future possible outcomes and fail to recognize that the true value may lie outside the forecaster's range estimate. *Overconfidence* hides possible-but-improbable outcomes from decision-makers and makes values within the range estimate appear more probable than they actually are.

With *anchoring*, forecasters are unduly influenced by any initial number that is currently at hand, even if that number is irrelevant to the estimation problem. For agile forecasting, *velocity*—the arithmetic mean of an agile team's historical record of completed work items—acts as an anchor upon which an agile forecast is usually built, even if the agile team's velocity is not a good predictor of their *future* performance. The standard deviation of completed work items is another anchor, although not all teams calculate their standard deviation.

SIX QUESTIONS FOR ARTFUL AGILE FORECASTING

Agile forecasters can improve their forecasts by raising, then answering, six important questions about their data-driven forecasts. When adjusting data-driven forecasts, agile forecasters must avoid the cognitive biases which could result in forecasts that are *worse* than unadjusted, data-driven-only forecasts. With prudent application of subjective judgment, agile forecasters can expand their range estimates to overcome *overconfidence*, and they can choose to override an easy-to-calculate *anchor* when the *anchor* value is not a good predictor of the agile team's future performance.

The six questions every agile forecaster should ask when preparing a forecast for an agile-developed product or service are:

1) How much team history should I include?

2) When should I anticipate reduced team capacity to work?
3) How should I adjust for product backlog growth?
4) How wide should my “cone of uncertainty” be?
5) When should I modify my team’s historical velocity?
6) When should I modify my team’s standard deviation?

HOW MUCH TEAM HISTORY SHOULD I INCLUDE?

Some agile teams work for years on the same project (or value-stream). However, it is generally not a good idea to include *all* the agile team’s historical record of completed sprints when creating a delivery forecast for new product features. Agile teams perform regular self-inspection to evaluate ways where they can improve the way they work together to create a new product or service. Through this process of continual improvement, agile teams ought to be increasingly more efficient and effective in how they work together. This increased efficiency and effectiveness should result in an increased ability to deliver greater value more quickly over time. Therefore, an agile team’s recently completed iterations or sprints are more reflective of how quickly a team converts product backlog items into a “done” increment than what the team completed many months earlier.

An agile forecaster is faced with making a difficult decision of how much team history to include when calculating the velocity for a mature team that has spent many months (or years) working together. The objective is to select as much history of completed iterations or sprints as necessary to calculate a true measure of the team’s current velocity (mean) and standard deviation but avoid including team history that is no longer relevant or reflective of how the agile team is currently working.

This is a difficult, *subjective* judgment call. The agile forecaster must evaluate the team’s historical record to determine how much of that historical data is *relevant* and *reflective* to the conditions under which the agile team currently works.

As a general rule, it is preferable to use all the historical data from the last three-to-six most recently completed months because recently completed sprints are likely to be both *relevant* to current working conditions and *reflective* of how efficiently the agile team converts product backlog items to a new product increment. For agile teams using two-week iterations, using more than six months of history may materially change the delivery dates of new product releases based on old ways of working together. Using less than three months may cause the delivery date range to be too narrow (or possibly too wide) if the most recently completed sprints were consistently (or inconsistently) delivered.

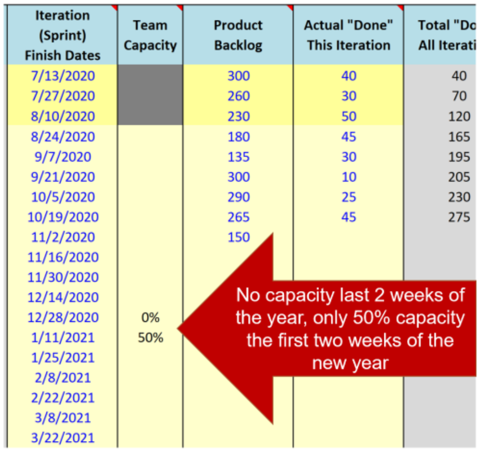
Sometimes there appears to be “outliers” in the agile team’s historical record. These “outliers” may appear to be very small or very large relative to the team’s velocity. Here is another opportunity for the agile forecaster to adjust their forecast by either retaining or excluding data “outliers.” Generally, it is *not* a good idea to exclude “outliers” in the data because the supposed rare conditions under which the outliers were created are often not as rare as agile teams, sponsors, and stakeholders think them to be—this is the *overconfidence* bias that an agile forecaster must guard against.

WHEN SHOULD I ANTICIPATE REDUCED TEAM CAPACITY TO WORK?

Sometimes an agile team's capacity to work in the future is reduced from their recent past. This could be for many possible reasons: upcoming holidays, upcoming school schedule breaks, summertime vacations, non-project training opportunities, business conferences, production support duties, and other, non-project efforts which diminish the team's capacity to work on their agile-developed product.

The agile team's historical record may not reflect these conditions which erode the team's capacity to work on their project endeavor. When that is the case, an agile forecaster must anticipate declines in the agile team's capacity to work and adjust the team's expected, future performance. This is especially true at the end of the calendar year when project teams to have little or no capacity to work because of end-of-year, holiday time-off.

Using Statistical PERT, the agile forecaster indicates the reduced capacity to work by choosing a percentage value under the "Team Capacity" column for the sprints where the team will have reduced capacity to work. For example, choosing "0%" means the team has no capacity to do any work at all, while choosing "50%" means the team expects to work at half its normal capacity.



Iteration (Sprint) Finish Dates	Team Capacity	Product Backlog	Actual "Done" This Iteration	Total "Do All Iterati
7/13/2020		300	40	40
7/27/2020		260	30	70
8/10/2020		230	50	120
8/24/2020		180	45	165
9/7/2020		135	30	195
9/21/2020		300	10	205
10/5/2020		290	25	230
10/19/2020		265	45	275
11/2/2020		150		
11/16/2020				
11/30/2020				
12/14/2020				
12/28/2020	0%			
1/11/2021	50%			
1/25/2021				
2/8/2021				
2/22/2021				
3/8/2021				
3/22/2021				

No capacity last 2 weeks of the year, only 50% capacity the first two weeks of the new year

Figure 1 – Team Capacity Adjustment

HOW SHOULD I ADJUST FOR PRODUCT BACKLOG GROWTH?

Agile teams do not identify all the expected work to complete like a traditional project team would do using a work breakdown structure. Product backlogs are dynamic. Teams will add, reprioritize, and remove backlog items as the project progresses. Usually, work emerges over time and the totality of the work to finish a product is greater than what the team first thought. In traditional project management, this is called "scope creep." In agile projects, product backlog dynamism is welcomed so it is the responsibility of the product owner to ensure that the team's efforts always maximize the value of the product.

Agile forecasters can anticipate work emergence over time. To do that, there are several techniques to use. The simplest technique is to add one or more unnamed epics to the product backlog, sized like other named epics are typically sized. This makes visible the team's expectation that new, as-yet-unknown work will emerge over time.

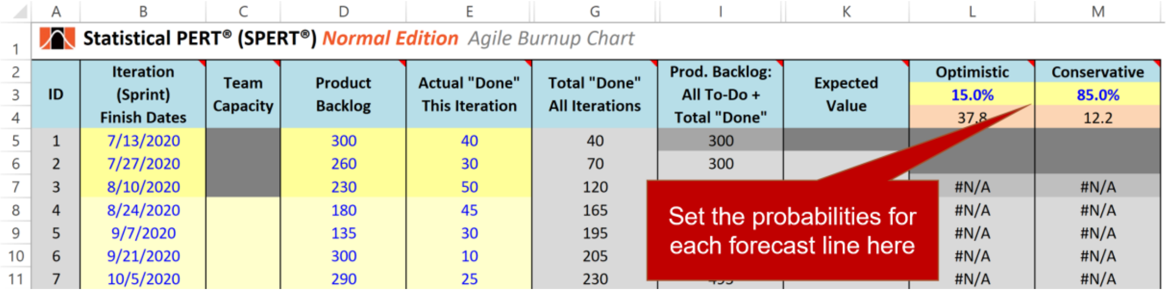
Another technique is to add a growth factor to the existing product backlog. For instance, if the current product backlog has 100 story points of work identified, the agile forecaster can assume 10% growth over the next few sprints and add 10 additional story points to each of the next imminent sprints. Then the total amount of work to complete over, say, the next three sprints is 130 story points (100 + (100 × 10% × 3 sprints). This technique, however, does not make the forecaster's assumption of product backlog growth as visible as does adding unnamed epics to the product backlog.

HOW WIDE SHOULD MY "CONE OF UNCERTAINTY" BE?

The "cone of uncertainty" is determined by two factors: the standard deviation and the confidence interval selected by the agile forecaster. The standard deviation is easily calculated using the agile team's historical record of completed work and Excel's standard deviation function, STDEV.P (standard deviation of a population).

But the agile forecaster also has control over the width of the "cone of uncertainty." Meteorologists who create hurricane forecasts use a "cone of uncertainty" where the probability of the hurricane staying within the cone is only 60-70% (NOAA, n.d.). But since stakeholders may not understand that the actual performance of an agile team may fall *outside* the cone of uncertainty, it is preferable to construct a "cone of uncertainty for an agile forecast that is either 70% or 80% probable, leaving a small amount of uncertainty falling outside the cone.

Using the Statistical PERT burnup chart worksheet, an agile forecaster may specify a confidence interval by choosing "optimistic" and "conservative" percentiles. The example in Figure 2 shows how to create a 70% probable "cone of uncertainty" where the cone's forecast lines are drawn at the 15th and 85th percentiles. Finish dates for future product releases are 30% likely to occur outside the date range determined by the "cone of uncertainty."



Statistical PERT® (SPERT®) Normal Edition Agile Burnup Chart

	A	B	C	D	E	G	I	K	L	M
1										
2	ID	Iteration (Sprint) Finish Dates	Team Capacity	Product Backlog	Actual "Done" This Iteration	Total "Done" All Iterations	Prod. Backlog: All To-Do + Total "Done"	Expected Value	Optimistic	Conservative
3									15.0%	85.0%
4									37.8	12.2
5	1	7/13/2020		300	40	40	300			
6	2	7/27/2020		260	30	70	300			
7	3	8/10/2020		230	50	120			#N/A	#N/A
8	4	8/24/2020		180	45	165			#N/A	#N/A
9	5	9/7/2020		135	30	195			#N/A	#N/A
10	6	9/21/2020		300	10	205			#N/A	#N/A
11	7	10/5/2020		290	25	230			#N/A	#N/A

Set the probabilities for each forecast line here

Figure 2 – Confidence Interval Settings for the "Cone of Uncertainty" Forecast Lines

If an agile forecaster wants to convey a greater sense of uncertainty to their stakeholders, the forecaster can widen the "cone of uncertainty" by using an 80% confidence interval (Optimistic = 10%, Conservative = 90%). This will usually suggest a more conservative finish date on the burnup chart but may not materially change the optimistic finish date.

WHEN SHOULD I MODIFY MY TEAM'S HISTORICAL VELOCITY?

One of the cognitive biases to avoid is *anchoring* bias. Calculating a team's velocity (arithmetic mean) and standard deviation creates two anchors for an agile forecaster. Nearly every agile team knows their velocity—it is their average that is computed from their historical record of completed iterations or sprints.

Since an agile forecaster will judiciously compute statistics using team history that is both *relevant* and *reflective* of their future work capability, this question involves deciding whether the resulting anchor (*historical* velocity) is likely to be reflective of their *future* work performance.

There are many reasons why it may not be. For example, if the team's composition is going to change in the future, their velocity may not be reflective of their future ability to complete work items. If the context is using a scaled agile approach where more than one team is developing a solution, the forecaster must know whether the scaling approach might change in the future—whether new teams will join those already formed, or whether an existing agile team will cease working with other teams to collectively build an agile-developed product using a scaled approach.

Velocity is an anchor value. The agile forecaster must recognize this as an anchor and decide whether it properly frames the range of possible, future outcomes. If it does not, the agile forecaster should carefully use their own judgment to override the anchor so a better-fitting range of possible outcomes are included in their forecast.

WHEN SHOULD I MODIFY MY TEAM'S STANDARD DEVIATION?

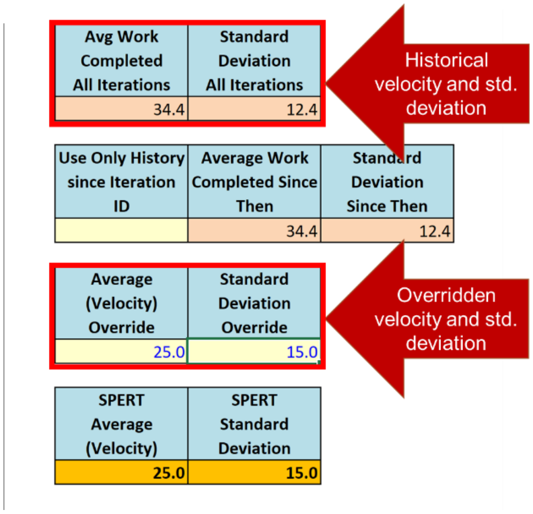
If a team has wide variations in what they finish each sprint, their standard deviation will be larger. If a team consistently delivers finished work items that are comparable to prior sprints, their standard deviation will be smaller. When an agile team has a small standard deviation, the unadjusted result is a narrow "cone of uncertainty" and the illusion that steady *past* performance is indicative of *future* performance. Even without injecting subjective judgment into the forecast, a too-narrow standard deviation may cause *overconfidence* in an agile forecast. The calculated "optimistic" and "conservative" forecast delivery dates may be closer to the mean than what they should be. This is a condition that an agile forecaster must evaluate and sometimes adjust to ensure their forecast does not suffer from *overconfidence*.

There are many reasons why an agile forecaster might want to override the standard deviation that is calculated from the agile team's historical record. Here are examples where an agile forecaster may want to deliberately increase the standard deviation to create a wider "cone of uncertainty" that reflects the forecaster's greater uncertainty about the team's future performance:

- The team's composition of individual contributors is going to change
- The team's availability or capacity to work on the project is subject to unexpected changes
- The team's historical record is, for other reasons, irrelevant or non-reflective of the team's likely future performance

There may also be reasons to override the calculated standard deviation by changing it to a *smaller* value. For instance, if the team's historical record is highly inconsistent but the factors causing the inconsistent performance are identified and now controllable, the agile forecaster may choose to shrink the calculated standard deviation to reflect the expectation that those causal factors will not continue to cause inconsistent future performance. However, overriding an agile team's calculated standard deviation should be done cautiously, since this may be the source of the agile forecaster's *overconfidence* bias, one of the many cognitive biases which degrade the quality of a forecast.

In Figure 3, the agile team's historical velocity of 34.4 and standard deviation of 12.4 is not reflective of their expected future performance. The forecaster has overridden the velocity with a value of 25 and increased the standard deviation to 15. The result of this action is that the expected performance in the future will be about 2/3rd of what the team has historically achieved, and there is greater uncertainty about how consistently they will deliver completed work items in upcoming sprints.



Avg Work Completed All Iterations	Standard Deviation All Iterations
34.4	12.4

Use Only History since Iteration ID	Average Work Completed Since Then	Standard Deviation Since Then
	34.4	12.4

Average (Velocity) Override	Standard Deviation Override
25.0	15.0

SPERT Average (Velocity)	SPERT Standard Deviation
25.0	15.0

Historical velocity and std. deviation
Overridden velocity and std. deviation

Figure 3 – Overriding an Agile Team's Velocity and Standard Deviation

CONCLUSION

Forecasting is a blend of art and science. Statistical models like Statistical PERT will use an agile team's historical record and estimates of all remaining work to calculate statistics to build an initial forecast. By asking and answering six key questions about their forecast, an agile forecaster can use their subjective judgment to adjust—and hopefully improve—their data-driven forecast. They do that by incorporating information which is not evaluated by their statistical model. This extra effort by the agile forecaster can better align stakeholder expectations about the uncertainty of agile development endeavors and can foster improved decision-making that maximizes the likelihood of obtaining desirable future outcomes.

REFERENCES

State of Agile. (2020). *14th annal state of agile™ report.* Digital.AI. https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494> May 26, 2020.

Tetlock, P. (2005). “Expert political judgment: how good is it?” Princeton University Press, Hoboken.

Menand, L. (2005). “Everybody’s an expert.” *The New Yorker,* https://www.newyorker.com/magazine/2005/12/05/everybodys-an-expert (Dec. 5., 2005).

Kavanagh S., and Williams, D. “Making the best use of judgmental forecasting.” *Government Finance Review.* Dec. 2014, 30(6), 8.

Davis, W. W. (2020). “How to forecast delivery dates for an agile-developed product using an agile burn-up chart.” *2020 Project Management Symposium Conference Proceedings.* University of Maryland, College Park.

Welsh, M. B., and Begg, S. H. (2018). “More-or-less elicitation (MOLE): reducing bias in range estimation and forecasting.” EURO Journal on Decision Processes, 6(4), 172.

NOAA (n.d.). “Definition of the NHC track forecast cone.” *National Hurricane Center at the National Oceanic and Atmospheric Administration.* https://www.nhc.noaa.gov/aboutcone.shtml