UMD Project Management Symposium
When Will It Be Done?
How to Forecast Answers
To Your Toughest Agile Questions

William W. Davis[1]

[1]William W. Davis, MSPM, PMP,
5000 NW 5th Avenue,
Boca Raton, FL 33487; famousdavispmp@gmail.com

ABSTRACT

Just like a plan-driven project that uses a predictive project management methodology, agile product development is subject to the same project management "triple constraint" of scope, time/schedule and cost/resources. Although agile teams may no longer create detailed schedules, organizational leaders still want to know when their agile teams expect to deliver new value to their customers.

This paper will explore how to create a probabilistic forecast using a Statistical PERT® Normal Edition spreadsheet. Statistical PERT spreadsheets are freely licensed, Microsoft Excel®-based files that use the built-in, statistical functions inside Excel. Agile teams who use a probabilistic forecast are better equipped to align expectations with key organizational stakeholders. Armed with information from a probabilistic burn-up chart, organizational leaders and agile teams can make better decisions *today* to achieve desirable outcomes *tomorrow*.

INTRODUCTION

Agile teams do not work with detailed project schedules. Yet, most organizations continue to use projects to fund new software development efforts (the alternative is to continuously fund value streams without projects, much like organizations fund their core operational functions). For project-oriented organizations, their project charters authorize how long an agile team can work together on a specific project. The project funding should be equal to the cost of funding a dedicated agile team (or multiple teams for larger efforts) for the full duration of the project's schedule constraint. For the scope constraint, it is common for the project charter to offer high-level scope objectives for the agile team to complete and the business value that the organization expects to receive. The charter may also offer a high-level release plan or key milestones for expected delivery cycles, but these are often little more than guesswork on the part of those who drafted the charter.

Agile teams are (usually) long-lived teams and not subject to the same kinds of fluctuations in team member participation as project teams working on traditional, so-called "waterfall" projects. Agile teams, ideally, have all the skills necessary to plan, design, create, test, and continuously deploy new software into their organization's production environment.

Therefore, two of three parts of project management's so-called "Triple Constraint" are well understood for endeavors using an agile approach for product development: the schedule and cost constraints. What is unknown to the agile team and the sponsoring organization is how much scope can be completed within the time and budget constraints imposed by the project charter.

Specifically, organizations using an agile approach to new development must answer these scope-related questions:

- When will the project's scope *really* be finished?
- When can new, major releases of the product be ready for production deployment?
- How much scope (that is, new and/or enhanced features and capabilities) can be delivered by a certain date?

Answering these questions is important to project sponsors and organizational executives. Executive leadership of an organization is accountable for how they use the organization's available project resources, especially their employees' time and effort.

To make informed decisions on which projects to fund (and not to fund) and which product features to pursue (and not to pursue), product owners, product managers, and organizational leaders must have enough information available to assess the expected return on investment for each project and each new product feature. To do that, they need agile teams to provide reliable estimates for how long they think they need to work together to convert ideas listed on a product backlog into working software.

By providing product owners, product managers, and organizational leaders with a delivery estimate, it is easy to compute how much each major product release may cost:

Agile team weekly cost × Number of expected weeks to complete a product release

Although the equation is simple, what is *not* simple is accurately estimating how long it will take an agile team to convert ideas on a product backlog into working software.

This paper explores how to estimate when an agile team will complete new development efforts, whether that is an entire project or just a subset of the project's expected scope. Using the built-in statistical functions inside Microsoft Excel® and a freely licensed, pre-built spreadsheet called Statistical PERT® Normal Edition[1], an agile team can create product development forecasts that align delivery date expectations with product owners and other key stakeholders. These forecasts inform organizational decision-makers so they can make better decisions earlier to achieve desirable future outcomes.

ESTIMATING, PREDICTING, AND FORECASTING

In his book, *Upgrading Leadership's Crystal Ball,* healthcare economist and futurist, Dr. Jeffrey C. Bauer, makes a distinction between a **prediction** and a **forecast**. According to Dr. Bauer, these distinctions are highly relevant to business decision-makers who must make informed trade-offs about which opportunities to pursue—and which not to pursue.

Dr. Bauer offers that a *prediction* is "a specific estimate of the expected value of a key variable at a future point in time" (Bauer, 2017, pg. 4). He explains that a *forecast* is "an estimate of the probabilities of the possibilities for a key variable at a future point in time" (Bauer, 2017, pg. 4).

Note that Dr. Bauer uses the word "estimate" in both definitions. The Merriam-Webster dictionary defines an estimate as, "a rough or approximate calculation" or "a numerical value obtained from a statistical sample and assigned to a population parameter" (Estimate, n.a.).

A quick search on the Internet reveals that *estimate, prediction,* and *forecast* are variously defined, sometimes used interchangeably, and can differ from Dr. Bauer's definitions. However, it is possible to simplify Dr. Bauer's definitions without losing their distinctive appeal and develop accessible definitions for *estimate, prediction,* and *forecast.*

For the purposes of this white paper, an *estimate* is, simply, **somebody's idea about the true value of something unknown**. In the context of project management and agile product development, what is unknown is how long an agile team will work together to create working software out of the product owner's wish-list items appearing on the team's product backlog. If an agile team can estimate how long it might take to convert product backlog items into working software, the cost of that effort is simply the result of multiplying the team's weekly burn-rate (that is, the cost it takes to fund the agile team's working together for a week) with the number of expected weeks it will take to achieve some desirable objective, like completing a major, new feature request.

Similarly, a *prediction* is **somebody's *single* idea about the true value of something unknown**. This is a *deterministic* estimate because it represents just one possible outcome

[1] Statistical PERT® spreadsheets are freely licensed and available at https://www.StatisticalPERT.com

(usually the expected or most likely outcome).

Finally, a *forecast* is **somebody's *varied* idea about the true value of something unknown**. A forecast varies in the *possibilities* of what may occur and their respective *probabilities* of occurrence.

Most often, estimates of the future are created as deterministic estimates—single-value predictions of the future. But predictions of the future have two severe drawbacks. Firstly, a predicted value does not indicate how likely the prediction is. There is no sense of certainty or uncertainty about the prediction because it is only a single value with no corresponding probability of occurrence. Secondly, a predicted value does not reveal other possible outcomes. There is no sense of variation of alternate possible outcomes.

Without understanding the sense of certainty and uncertainty about project predictions, and without knowing the span of other possible outcomes that might occur, key stakeholders and agile teams cannot be well-aligned. Decision-makers who evaluate predicted values risk making poor decisions by not recognizing the complexity and uncertainty with creating new product features and the level of effort it may take to create new solutions.

Rather than share predictions with key stakeholders, agile teams can learn to forecast uncertain future outcomes. By sharing their forecasts with organizational leaders, agile teams can effectively convey their sense of risk, uncertainty, complexity, and required effort to convert product backlog items into working software.

METHODS OF ESTIMATING PRODUCT BACKLOG ITEMS

Agile teams commonly use relative sizing called *story point estimation* to estimate product backlog items. Large product backlog items which are too big to bring into the team's next iteration are called "epics." Smaller product backlog items that comfortably fit into the team's iteration are usually called "user stories." An "epic" is simply a too-big user story that, at the right time, must be decomposed into smaller user stories.

Another way to estimate product backlog items is not to estimate them all. Instead, some agile teams simply decompose "epics" into roughly the same-sized "user stories" before they begin working on them. Then, at the end of an iteration, they count the number of completed user stories. By knowing about how many same-sized "user stories" they can complete each iteration, these agile teams can estimate future performance by estimating how many decomposed user stories are represented on their product backlog.

It is beyond the scope of this white paper to analyze how teams estimate their product backlog items. Irrespective of how they estimate their work, they need a way to create projections—forecasts, in particular—to answer scope-related questions from key stakeholders. This paper will use *story point estimates* for all examples, but the technique in this paper works equally well using other estimation approaches, like *user story counts*.

Once an agile project has begun, a product owner populates the product backlog with ideas for a new or enhanced product. The agile team works together to understand these feature requests. Once understood, the team can use whatever estimation method they choose.

Rather than decomposing all their work up-front, an agile team is content to use rough-order-of-magnitude estimates for their epics. They will only decompose high priority product backlog items on which they expect to work in the next iteration or two. Once they have estimated all the known work in the product backlog (epics and decomposed user stories), they are in position to do release forecasting.

STATISTICAL PERT® NORMAL EDITION

Statistical PERT® Normal Edition is a freely licensed spreadsheet file that uses the built-in statistical functions inside Microsoft Excel®. Statistical PERT spreadsheets are free to download, use, modify, and share with others under the terms of the GNU General Public License, a copyleft license agreement created by the Free Software Foundation (Free Software Foundation, 2007).

Using Statistical PERT—SPERT® for short—an agile team can answer many questions by forecasting answers to them. SPERT does not use Monte Carlo simulation to create forecast results. Instead, SPERT spreadsheets model uncertainties with bell-shaped risk properties using Excel's built-in statistical functions, like the NORM.DIST (normal distribution) and NORM.INV (normal inverse) functions.

WHEN WILL THE PROJECT SCOPE REALLY BE FINISHED?

There are several ways SPERT users can forecast an answer to the question, "When will the project scope *really* be finished?" One way is to use the "Agile Forecast" worksheet (one of several worksheets inside a SPERT workbook). Using this worksheet, an agile team can express a three-point estimate (minimum, most likely, maximum) for how much work they can complete in a sprint or iteration using the team's preferred estimation unit-of-measure.

The unit-of-measure the team uses to estimate items on their product backlog does not matter. The team can estimate using story points, user story count, number of workdays or hours, or some other unit-of-measure to assess the work effort, complexity, and uncertainty represented by their product backlog items.

To create a project finish date forecast, an agile team must know the following:

- The **starting date** for their project (or product release)
- The **number of weeks in their sprint** or iteration planning cycle
- Their **velocity**, which is an average number of what the team can complete in each sprint or iteration planning cycle, expressed using the team's unit-of-measure
- A **subjective judgment** about *how likely* the most likely outcome (that is, their velocity) will occur in future iterations
- A **best-case scenario** for how much the team might possibly achieve in an iteration
- A **worst-case scenario** for how little the team might possibly achieve in an iteration
- The **total amount of work** represented on their product backlog for the whole project or just a subset of features needed for the next major product release

Once these inputs are added to the "Agile Forecast" worksheet, the forecaster can stipulate with what level of confidence they would like to calculate a finish date. Common choices are 10-25% probable to find an aggressive/optimistic finish date, 50% probable to find an expected finish date, and 75-90% probable to find a conservative/pessimistic finish date.

SPERT Normal Edition uses the normal (sometimes called Gaussian) probability distribution to represent the uncertain finish date for a project. Although the normal distribution may not be the best-fitting distribution, using the normal distribution has, in practice, served to fulfill the two main goals of estimating agile project uncertainties: align key stakeholder expectations and make good business decisions earlier.

Since the normal distribution is unbounded (that is, the probability distribution is distributed to infinity on both the left- and right-side of the probability distribution bell-curve, exceeding the boundaries specified by choosing minimum and maximum point-estimates), choosing forecast values should be at least 5% probable and no greater than 95% probable to minimize forecasting distortions from using the normal distribution to model bounded uncertainties.

Below, Figure 1 is an example of the SPERT Normal Edition's "Agile Forecast" worksheet. Using example data, the projected finish date for this project is November 2, 2020. The projected finish date is only 50% probable, meaning, that the true finish date for the project may finish on or before November 2; there is a 50% chance the true finish date will be after November 2.

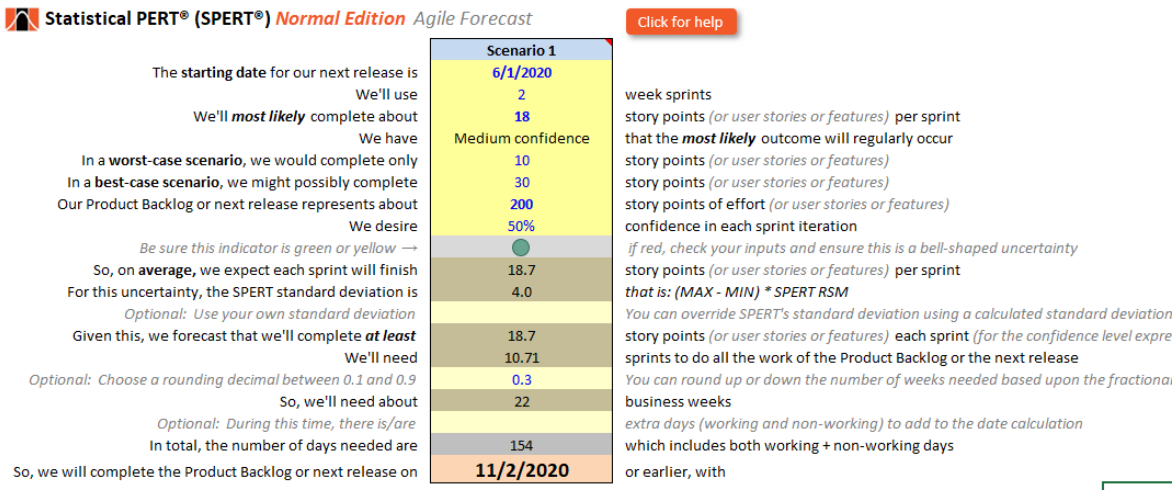

Figure 1. "SPERT® Normal - Agile Forecast" worksheet at 50% confidence

To find a higher-confidence project finish date, the user needs only to change the percent of confidence to a more desirable choice.

For example, in Figure 2 (below), if the forecaster changes the confidence level from 50% to 80%, the resulting finish date shifts from November 2 to November 30. This makes sense; if an agile team wants to be more certain that they will finish all the work on their product backlog, they will need more time to handle the unexpected issues that may arise during their working together.

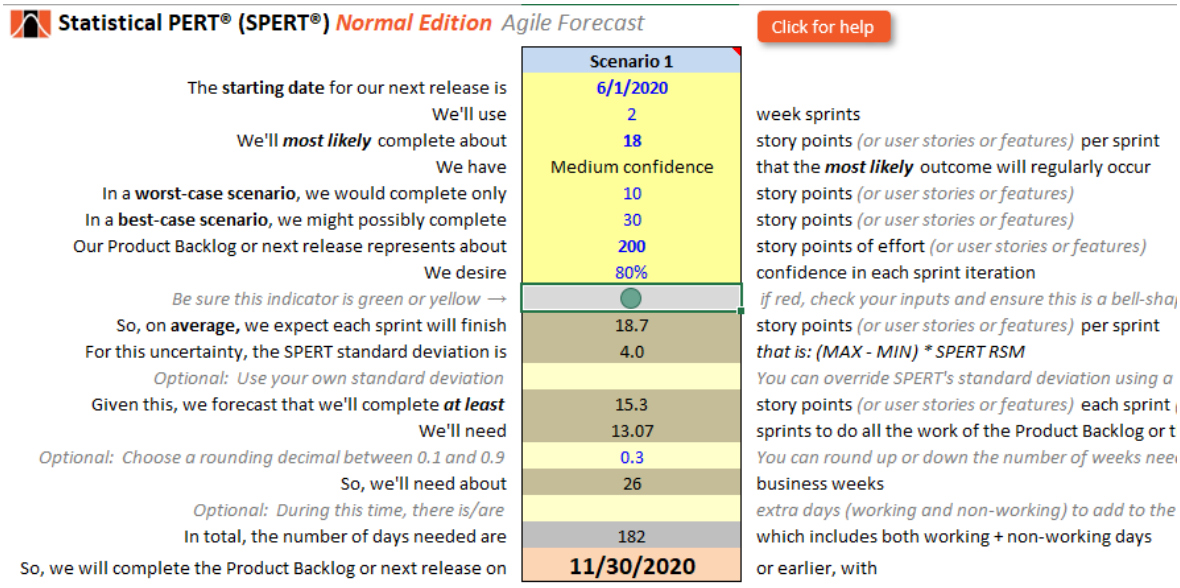

Figure 2. "SPERT® Normal - Agile Forecast" worksheet at 80% Confidence

A key feature from using a SPERT spreadsheet is that the forecaster can apply their own subjective judgment about the uncertainty they are estimating. The "Agile Forecast" worksheet lets the user choose from among 10 different subjective judgments about *how likely* the most likely outcome really is. If an agile team has had difficulty completing their planned work at the end of their iteration time-box, then the forecaster may choose a subjective choice like "Low Confidence" or "Very Low Confidence" that the team will match or come close to matching their historical velocity. Conversely, if the team has established a regular cadence and experiences relatively little variation in what they are able to achieve in each iteration, the forecaster may have "Medium-High Confidence" or "High Confidence" that the team will match or come close to matching their historical velocity.

The forecaster's subjective judgment about *how likely* the most likely outcome really is will change the underlying standard deviation that the SPERT spreadsheet automatically creates to represent the uncertainty's variation. The size of the standard deviation influences the implied shape of the normal distribution bell-curve, and that affects the probabilities associated with the range of possible outcomes.

If a forecaster knows how to calculate the standard deviation from the team's historical velocity (by simply using Excel's STDEV.P function), they can optionally insert the true standard deviation value into this worksheet (row 14) and the explicit standard deviation will override the SPERT-generated standard deviation.

VISUAL FORECASTING USING A PROBABILISTIC BURN-UP CHART

The Statistical PERT® Normal Edition's "Burn-up Chart" worksheet allows a user to explore different what-if scenarios using different inputs assumptions. For an agile team that has actual recorded history of performance, using a burn-up chart can show what happens when future performance mimics past performance. But what happens when past performance is not a good indicator of future performance? What if the team's future performance declines? What if it improves?

Using a probabilistic burn-up chart shows more than just the expected outcome for when the agile team will finish their work. Using Statistical PERT® Normal Edition's "Burn-up Chart" worksheet, an agile team can create a flexible, visual forecast to model a range of future possibilities that might occur. Figure 3 is an example of this burn-up chart.

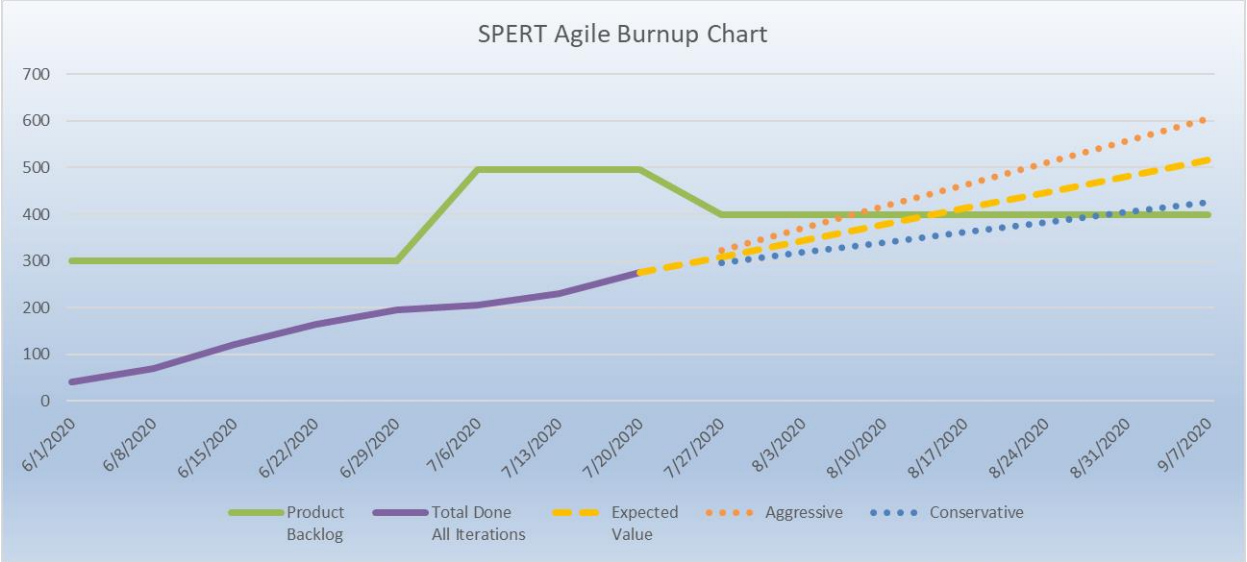

Figure 3. "SPERT® Normal - Burn-up Chart" worksheet

In Figure 3, the horizontal line that begins at Y-axis value of 300 represents the team's estimate to complete a set of work represented on their product backlog. It could be a major product release, a key feature, or even all the work on the product backlog. Beyond completing backlog items, product backlogs change over time—by removing items that the product owner no longer wants and/or by adding new ideas that the team discovers togethers—the horizontal line may rise and/or fall over time. The product backlog line represents a time-series sum of both "done" and "undone" product backlog items.

Examining the calendar date values on the X-axis, the product backlog work was increased between June 29 and July 6 but then decreased slightly between July 20 and July 27. This is normal behavior for product backlog management.

The other solid line—a slightly wavering line rising from the bottom-left corner of the chart—represents the amount of work completed by this agile team. Since this line has not intersected the horizontal line (which represents the product backlog), the inference is that this agile team still has work to complete. Wherever these two solid lines intersect, that is the point that the agile team has completed all the work included in the forecast.

The dashed and two dotted lines on this probabilistic agile burn-up chart constitute the forecast of the agile team's possible future performance. The dashed line is the expected value line; given the team's historical performance as a baseline, the expected value line is the most likely outcome for future performance. Where the dashed line intersects with the product backlog's horizontal line (August 17) is when the agile team most likely will complete all the work included in this forecast. This outcome is a 50/50 forecast. There is a 50% probability that the agile team will finish on or before August 17 and a 50%

probability they will finish after August 17.

The two dotted lines show two other finish date possibilities. One dotted line intersects the product backlog's horizontal line on August 10; this is an aggressive/optimistic possibility which is only 15% probable. The other dotted line intersects the product backlog's horizontal line on August 31; this is a conservative/pessimistic possibility which is 85% probable, meaning that true finish will be on or before August 31 with an 85% likelihood of occurrence. Importantly, the user of a Statistical PERT® spreadsheet can choose any probability values for the aggressive/optimistic and conservative/pessimistic outcomes, and the SPERT® spreadsheet will automatically adjust the burn-up chart accordingly.

With a probabilistic burn-up chart, agile teams can align with key stakeholders on the range of possible and probable outcomes for when they will finish a defined set of work on the product backlog. If key stakeholders do not like what they see in this visual forecast, they and the agile team can work together to find ways to improve the forecast. For instance, the team might need to remove items on the product backlog, de-scope items that stay on the product backlog, add additional team members, or find other, simpler ways to fulfill items that stay on the product backlog.

To create the probabilistic forecast, the forecaster will interact with a table that is on the SPERT "Burn-up Chart" worksheet (Table 1):

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

ID	Iteration (Sprint) Finish Dates	Product Backlog	Actual "Done" This Iteration	Total "Done" All Iterations	Prod. Backlog: All To-Do + Total "Done"	Expected Value	Aggressive	Conservative
							15.0%	85.0%
							47.2	21.6
1	6/1/2020	300	40	40	300			
2	6/8/2020	260	30	70	300			
3	6/15/2020	230	50	120	300		#N/A	#N/A
4	6/22/2020	180	45	165	300		#N/A	#N/A
5	6/29/2020	135	30	195	300		#N/A	#N/A
6	7/6/2020	300	10	205	495		#N/A	#N/A
7	7/13/2020	290	25	230	495		#N/A	#N/A
8	7/20/2020	265	45	275	495	275	#N/A	#N/A
9	7/27/2020	125			400	309	322	297
10	8/3/2020	125			400	344	369	318
11	8/10/2020	125			400	378	417	340
12	8/17/2020	125			400	413	464	361
13	8/24/2020	125			400	447	511	383
14	8/31/2020	125			400	481	558	404
15	9/7/2020	125			400	516	605	426
16								

Table 1. Input table on the "SPERT® Normal - Burn-up Chart" worksheet

The burn-up table requires these inputs:

- **Iteration (Sprint) Finish Dates**: *The finish dates for future iterations/sprints*
- **Product Backlog**: *The current work represented on the product backlog that is within the scope of this forecast*
- **Actual "Done" This Iteration**: *When known, this is a count of actual "Done" work for each iteration, summing product backlog item estimates for "Done" work*

In Table 1, the current sprint has a finish date of July 27, and there are 125 story points of

work left to do. Importantly, to correctly represent future work on the burn-up chart, the current work left to do in the "Product Backlog" column must be copied downward until the last iteration finish date entered under the "Iteration (Sprint) Finish Dates" column.

Notice the two cells below the labels "**Aggressive**" and "**Conservative**" (top-right corner in Table 1). These are user-enterable probabilities that correspond to the meaning of an aggressive/optimistic finish date or a conservative/pessimistic finish date. By default, they are set at 15% and 85% respectively, but the SPERT forecaster can change these values.

As each iteration completes, the forecaster will enter the sum of estimates for completed product backlog items into Table 1 (under the column heading, "Actual Done This Iteration"). The burn-up chart will update automatically to reflect all completed work, and the forecast of how the agile team might perform in the future will also be updated. Naturally, as the team gets closer to completing all their work, the gap between the "Aggressive" and "Conservative" forecast lines (where they intersect with the "Product backlog" line) will grow narrower as these possibilities converge with the "Expected Value" forecast line because there is less uncertainty about the imminent future.

CONCLUSION

Forecasting is superior to single-outcome estimates. Forecasting shows both the range of possibilities and the probabilities of those possibilities. Forecasting aligns agile teams, organizational leaders, and other key stakeholders by making explicit what the team believes is feasible and infeasible and what their sense of certainty is with respect to their effort in converting product backlog items into working software.

By using Statistical PERT Normal Edition's "Agile Forecast" and "Burnup Chart" worksheets, agile teams can model their planned development efforts and quantitatively answer common questions that organizational leaders have about when new features will be available for deployment or when the project's original scope will be delivered.

REFERENCES

Bauer, J. C. (2017). *Upgrading Leadership's Crystal Ball: Five Reasons Why Forecasting Must Replace Predicting and How to Make the Strategic Change in Business and Public Policy*, Productivity Press, Boca Raton, FL.

Free Software Foundation. (2007). "GNU General Public License." *GNU GPL Version 3.* Retrieved February 28, 2020, from https://www.gnu.org/licenses/gpl-3.0.html.

Merriam-Webster. (n.d.). Estimate. In Merriam-Webster.com dictionary. Retrieved February 28, 2020, from https://www.merriam-webster.com/dictionary/estimate