GistLabs^AI

Managing Uncertainty in Work and Velocity

John Heintz, CTO and Founder, GistLabs^AI

> "Foresight is not about predicting the future, it's about minimizing surprise.",
>
> Karl Schroeder

Introduction

Project managers must deal with both the uncertainty of work estimates and also uncertainty of team velocity. This whitepaper describes the general approach to manage these factors with an example of two teams working together to reach a goal. We expose the key choice teams must make:
Either update estimates for finished work, or accept velocity changes

Both of these can result in changes to the project forecast and we describe and calculate the implications in ways that can be communicated and actioned. We show below an example where slower velocity puts one team at high risk of delay, and updating estimates puts the other team at high risk. Either slower velocity puts TeamB at risk of 32 days delay, or the updated estimates put TeamA at risk of 26 days delay.

Table of Contents

https://gistlabs.ai/whitepapers/managing-uncertainty-work-velocity
Version 1

Velocity is Uncertain

Velocity, the team's pace of getting work done, is uncertain. Any number of things could slow down a team from getting done the work they planned. Multitasking, illness, an internet service outage, attrition, demotivation, and so on. Accepting and modeling this uncertainty is necessary to model and steer projects. In this paper we use the following assumptions:

- The team expects that their work day will probably be a full day, but will often range from 10% slower to faster. Further, they think it’s possible but unlikely that their day may range from zero to 2x.
- This is treated as a triangular of (0.9, 1, 1.1) and uniform from [0, 2]

The next important step is measuring a team’s velocity to update future predictions. This is where small data updates are learned (Bayesian refinement) and the model is updated in-place to *revise* forecasts and enable steering the project with updated information. This will be explained in detail below in Learning the New Team2 Velocity.

Work Estimates are Uncertain

Estimating work items has much prior art, and we just summarize here and provide our own style of encoding the estimates. Important factors in estimating work items include: a) future work is never precisely known, b) range based estimates are very useful to capture team knowledge[1], c) development work has more uncertainty than routine work and manufacturing work.

[1] Hubbard, D. W. (2007). How to measure anything: Finding the value of "intangibles" in business.

“How long do you think it will take?”

“Well, best case is 3 days. At most it could be 8 days, and probably 5 days”

This dialog shows the range based estimation approach to produce a triangular probability distribution that represents the random variable for this estimate. See Figure 1:

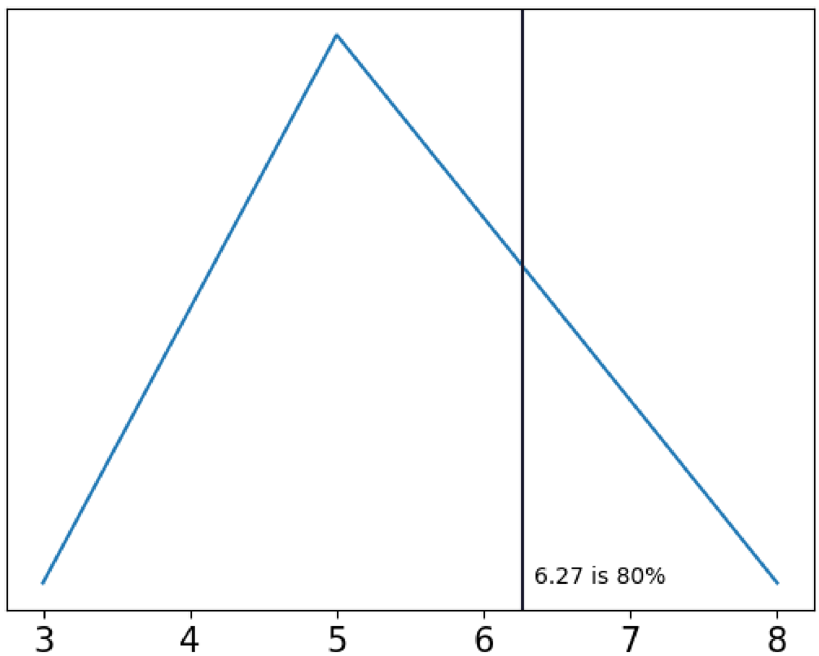

Figure 1: Triangular probability distribution for 3, 5, 8 with 80% likely at 6.27

Finally, we define classes of work as similar types of work tasks. In particular, estimation of these similar tasks will tend to correlate. If we find that a class of work is easier or harder than initially assumed, that new information can be applied to improve the estimates of other tasks of that same class of work. By “improve” we mean to make more accurate and more correctly reflect our latest knowledge. This could of course mean that estimates get bigger, but that is still an improvement because it helps us see what is really likely and avoid late surprises.

Sample Project: 2 teams, 11 tasks, 2 classes of work

For this paper we describe two teams each working on a list of tasks. The tasks are spread across two classes of work. We assume the teams are equally skilled in performing work on both classes of work, and that the teams estimations are comparable. Notice the following details that help setup this example:

- The first task for each team is expected to take 5 days.
- TeamA starts with tasks of class "Y", and finishes with "X". TeamB is the opposite.

TeamA Estimates	bestCase	expected	worstCase	classOfWork
Task1	3	5 days	8	Y
Task2	4	6	12	Y
Task3	5	8	15	X
Task4	3	8	13	X
Task5	5	8	13	X

TeamB Estimates	bestCase	expected	worstCase	classOfWork
Task6	3	5 days	8	X
Task7	2	3	4	X
Task8	5	7	12	Y
Task9	4	4	10	Y
Task10	5	8	12	Y
Task11	6	10	14	Y

Forecasting the Project on Day Zero

At the beginning of the project we run simulations to forecast the finish likelihood and odds of hitting our goal. This "day zero" plan analysis is an important step in planning realistic goals. This approach is similar to the classic Pert formula but used Monte Carlo simulations. One common use of day zero analysis is to set the project buffer to a reasonable amount based on total risk of missing the deadline. We set the buffer in Adding Buffer to the Project below.

Given our initial work estimates and velocity estimates we can forecast the likely (here we choose 80% confidence) outcome of each team's finish in Figure 2.

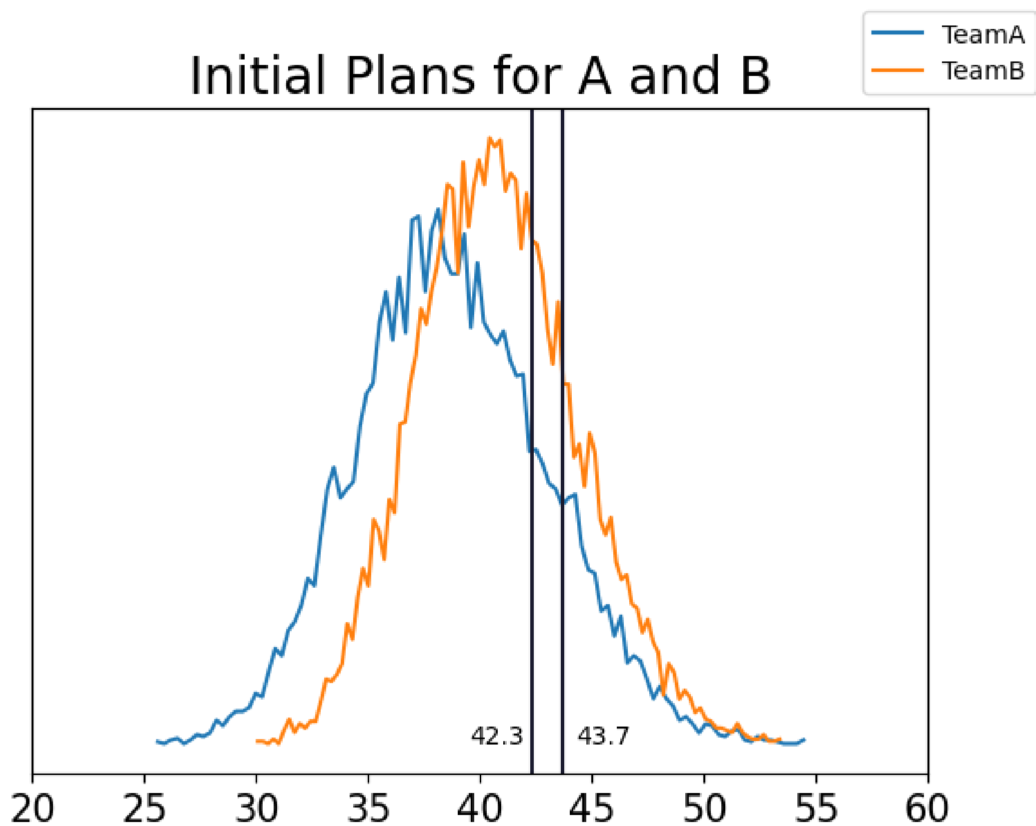

Figure 2: Likely (80% confidence) finish for teams is 42.2 and 43.7

The combined outcome of both teams finishing is shown in Figure 3, and the result is 45 days with 80% confidence (rounded).

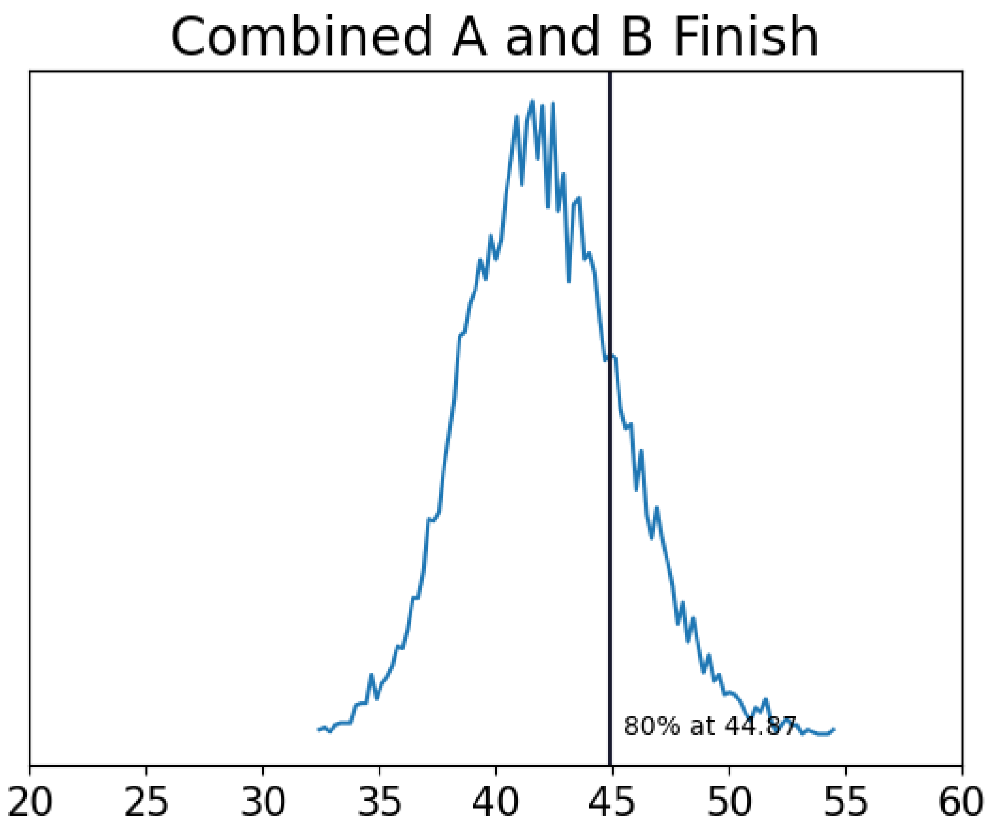

Figure 3: Likely (80% confidence) finish for both teams is 44.89 rounded to 45

Adding Buffer to the Project

Assuming the team and stakeholders agree that 80% is a useful planning confidence, then we can calculate the buffer for each team's schedule that should be added to the project plan given a 45 day finish forecast.

TeamA has an expected duration of (5 + 6 + 8 + 8 + 8) == 34, and TeamB of (5 + 3 + 7 + 4 + 8 + 10) == 37. Therefore TeamA will have 11 days of buffer, and TeamB 8 days of buffer.

At the end of the first week, TeamB is 50% complete. Now What?

At the end of the first week of the project both teams update status. TeamA has finished the first task and will start the second task on the 6th day of the project. TeamB has completed 50% of Task6 and has 50% work remaining. What do we do with this information?

At this point in the project we know we have risk, and likely the teams still have hope they will catch up. But, without further analysis and interpretation of the data, there isn't any clear action or update that can be provided to key stakeholders. Without more we can't make data-driven decisions and we have to wait longer to steer the project.

Was the Task Misestimated or Was the Team Slow?

The key question that only the team and leaders can answer is where the delay is coming from: root cause analysis is important here. Was the team working normally, but the task was misestimated? Or, was the task estimated fine but the team was multitasking/away/slow/other? There are no "bad" answers here, getting to the truth and dealing with it appropriately is the goal.

The next sections will analyze the implications of both of these potential sources of risk.

Learning the New TeamB Velocity

TeamB originally thought their velocity would most likely be around 1, plus/minus 10%. As the updates are made at the end of the first week we first assume the work estimate was accurate and analyze the impact on velocity for TeamB given the new evidence that they made 50% progress.

Applying Bayesian refinement and marginal learning[2] we derive a new velocity for TeamB shown in Figure 4. The updated velocity peaks around the 1 day region, but clearly has shifted down to a slower pace. Now the 50/50 odds of work per day is 85% of a full day.

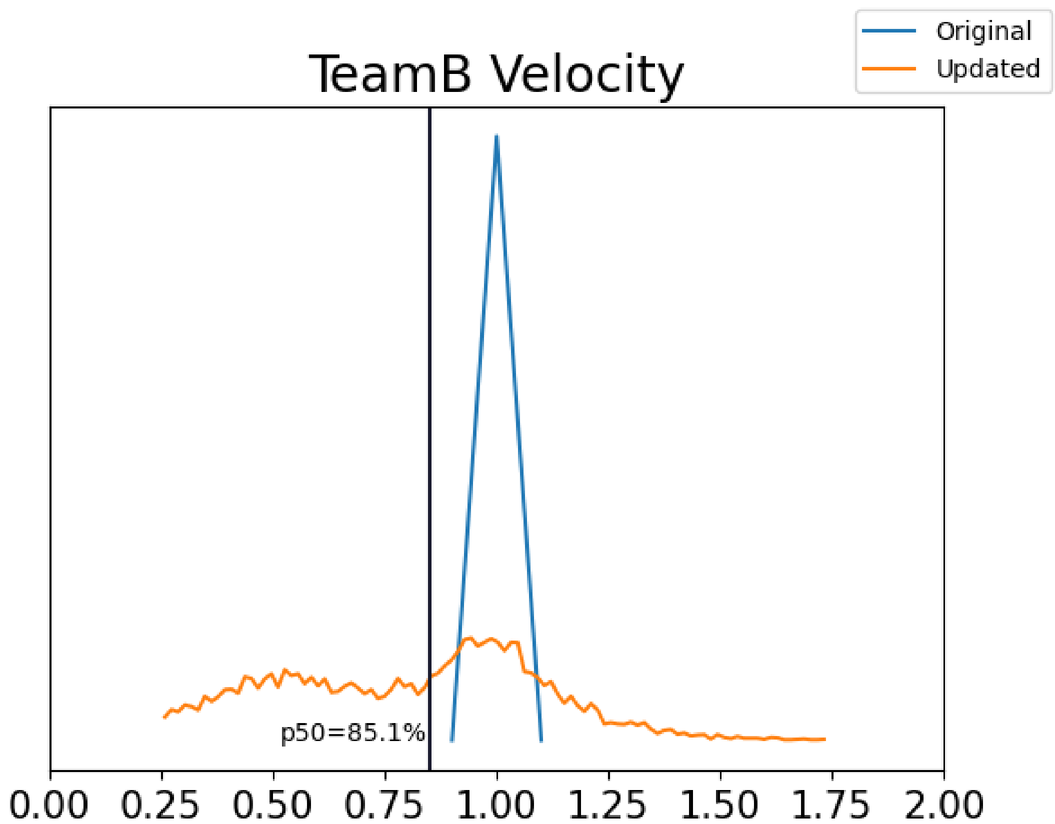

Figure 4: Learned velocity for Team B. Now mostly likely pace is 85% of a "day"

Velocity Risk: TeamB at high risk of 32 day delay

We can compare the new TeamA and TeamB forecasted finish predictions. See Figure 5 and notice that TeamA is nearly the same as before, while TeamB has a significantly later 80% confidence forecast.

[2] Cantor, M. (2023). An Introduction to Bayesian Parameter Learning

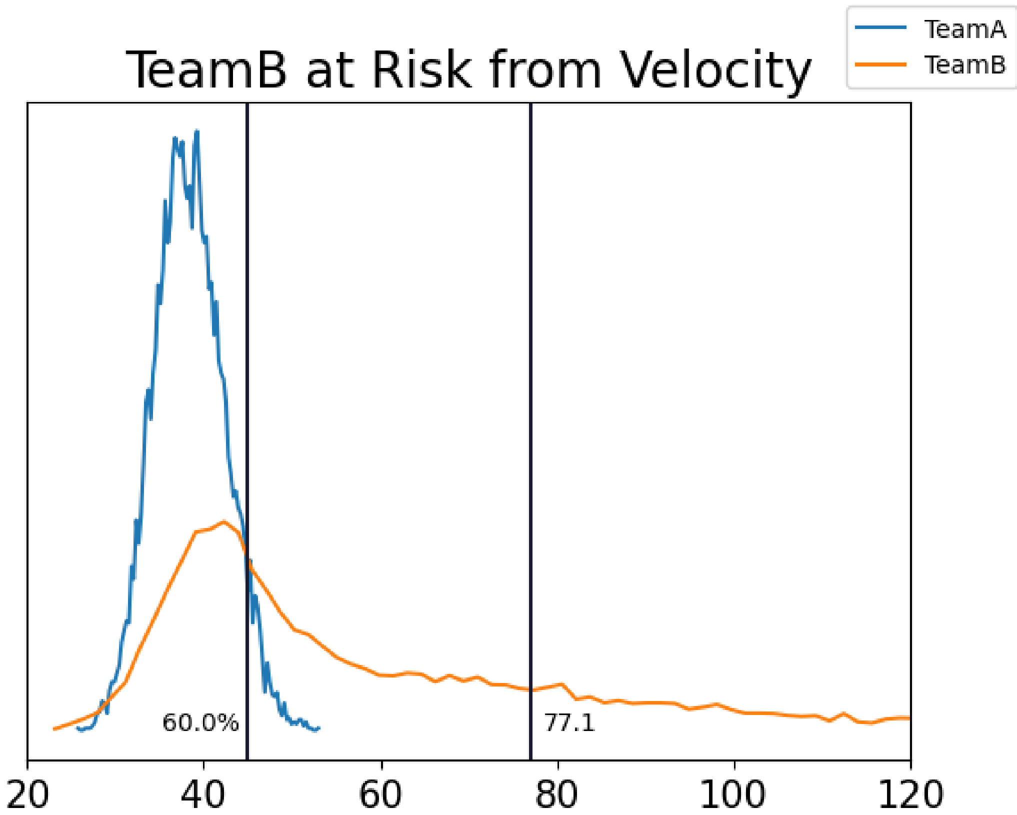

Figure 5: 80% confidence finish date is now 77 days due to TeamB velocity

This gives us two key pieces of actionable information:

1. The updated confidence of hitting day 45 is now 60%
2. The new 80% confidence target date is pushed back by 32 days to day 77

Both of these pieces of information are very important for steering and communication decisions. Also this information is available right away, instead of simply waiting for more data and losing time to make changes

Some of the choices we could make with this information:

- The stakeholders could accept a 60% likelihood of finishing
- "What if" scenarios where we shift work from TeamB to TeamA
- Root cause analysis, why is the team going slower?

Updating the Estimates for Class of Work “X”

Above we keep the original work estimate and the implications are entirely on TeamB finish forecast. Instead we can change the estimate of Task6 to be expected 10, ranging from 6 to 16. Further, all estimates for tasks of class of work “X” should be re-examined. Here, we simply double all the estimates. This has some effect on TeamB:

- The team’s velocity is maintained, not updated to 85% of “1” with wide range
- Some more work (the extra estimated effort) is added to TeamB backlog, but only for 2 tasks.
- TeamB has a slightly longer finish confidence now

Estimate Risk: TeamA at high risk of 26 day delay

The effects this has on TeamA is significantly more impactful. It’s important to note that the bulk of TeamA tasks are of class “X” and those also were the larger estimated tasks for this team. This significantly increases the backlog of work for TeamA and pushed their forecast 80% finish back 26 days to 71 days. See Figure 6 where TeamB and TeamA forecasts are shown.

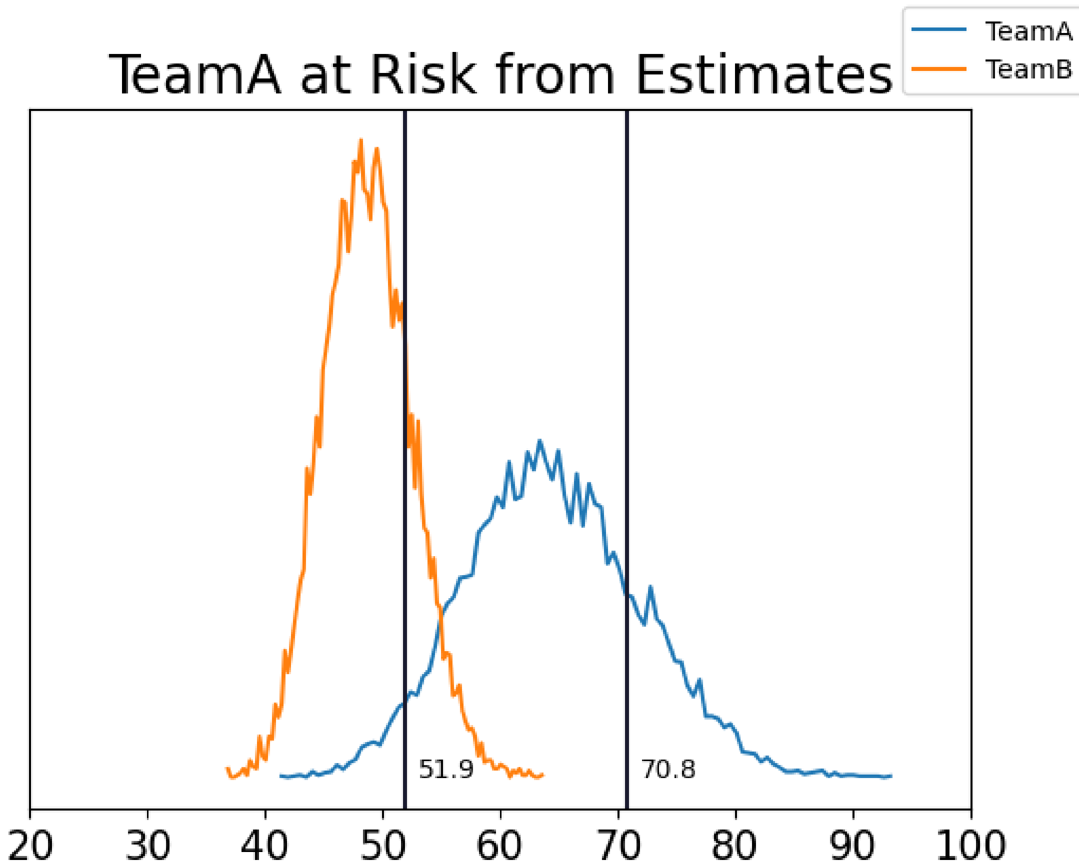

Figure 6: TeamA is now significantly longer schedule, with 80% confidence of day 71

Recommended: Update Estimates as We Learn

We recommend teams update estimates as they make progress. This enables us to use the available information we have to improve forecasts. Work items can be specifically designed to provide opportunities for learning and revising future plans. “Design spikes” from Extreme Programming, research activities, and prototypes all offer opportunities to improve future work estimates and planning.