

Agile Metrics: Velocity, Bug Tracking, and How To Talk About “When Will You Be Finished?”

Ryland Leyton

RylandTheBA@Gmail.com

www.RylandLeyton.com



About Ryland

Ryland Leyton, CBAP, PMP, CSM, is a business analyst, speaker, educator, Agile coach, and technology translator. He has worked in the technology sector since 1998, starting off with database and web programming, gradually moving through project management and finding his passion in the BA field.

Ryland is employed as Lead Business Analyst at Aptos.com.

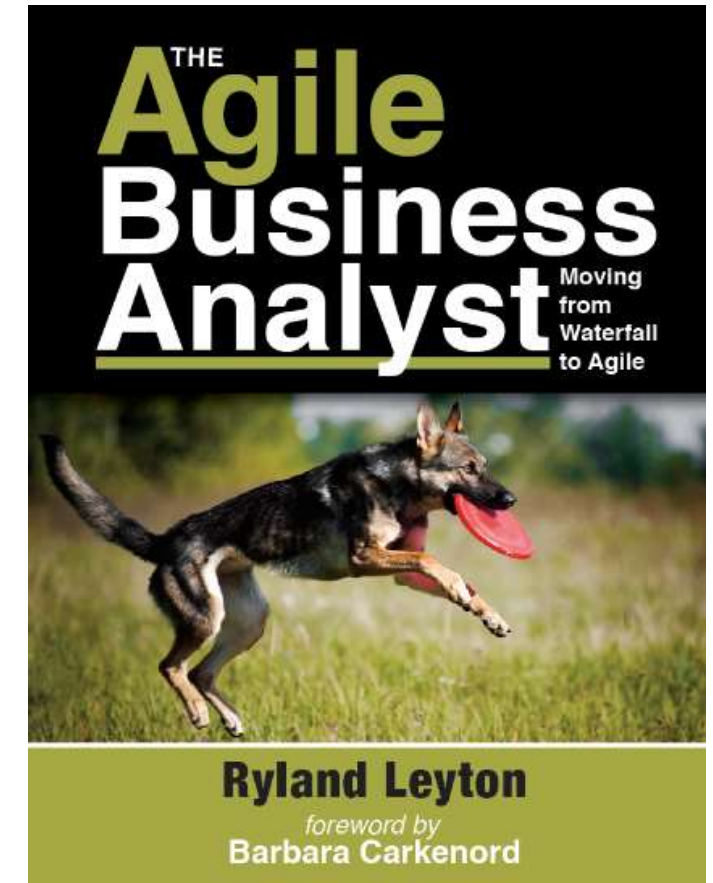
Ryland is passionate about strong analysis practice and prefers Agile environments where possible. He has built both Agile and waterfall SDLC processes for development teams, customizing each one to the challenges facing that particular client group.

He is an active member of his local chapter of the IIBA, speaks at local and national conferences, and serves as an Agile coach and educator.

Ryland is a core team author of *V2 of the Agile Extension to the BABOK*.

Ryland's book, "*The Agile Business Analyst*" has received excellent critical reviews and is available on Amazon.com.

[Available at www.Amazon.com](http://www.Amazon.com)



www.RylandLeyton.com

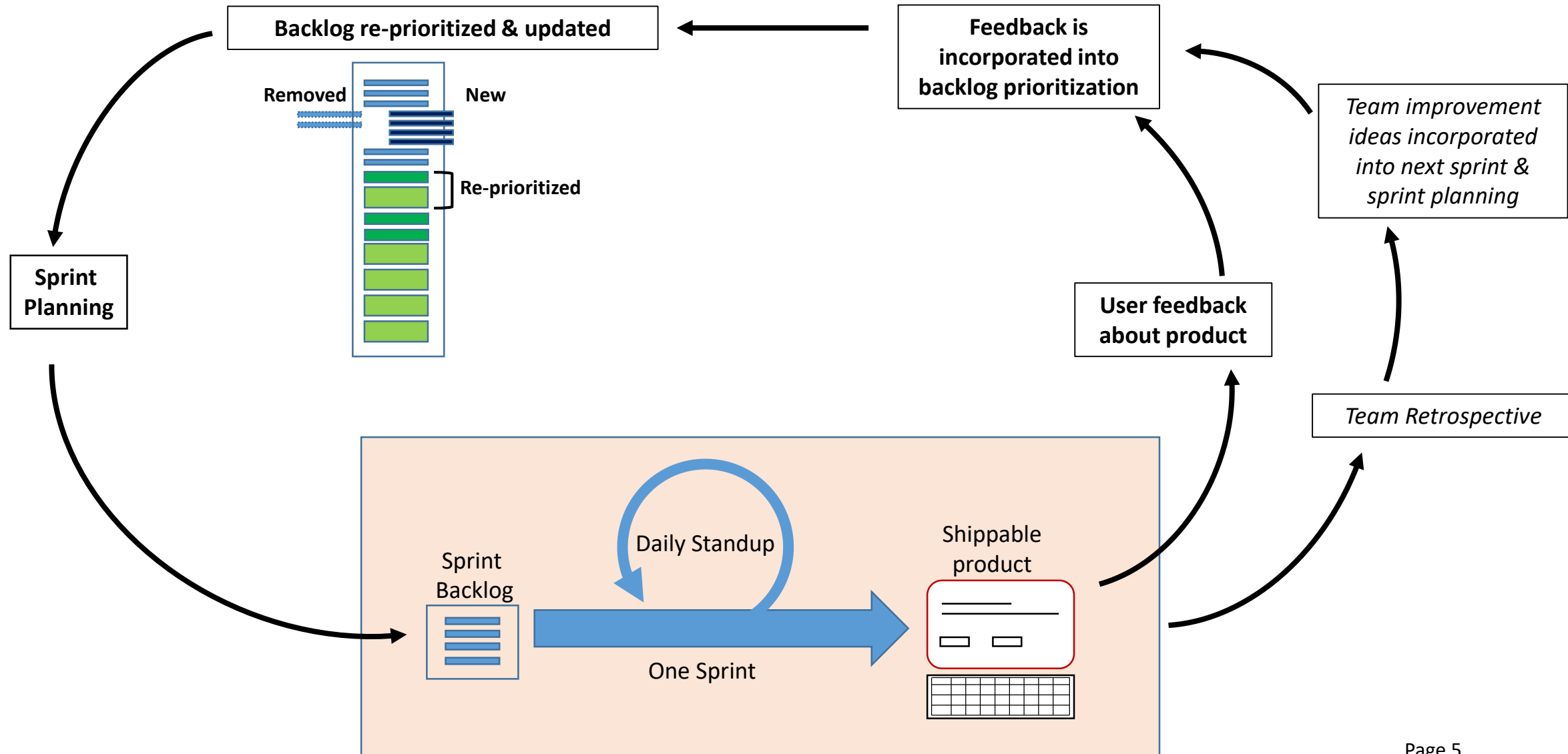
RylandTheBA@GMail.com

Where are we going today?

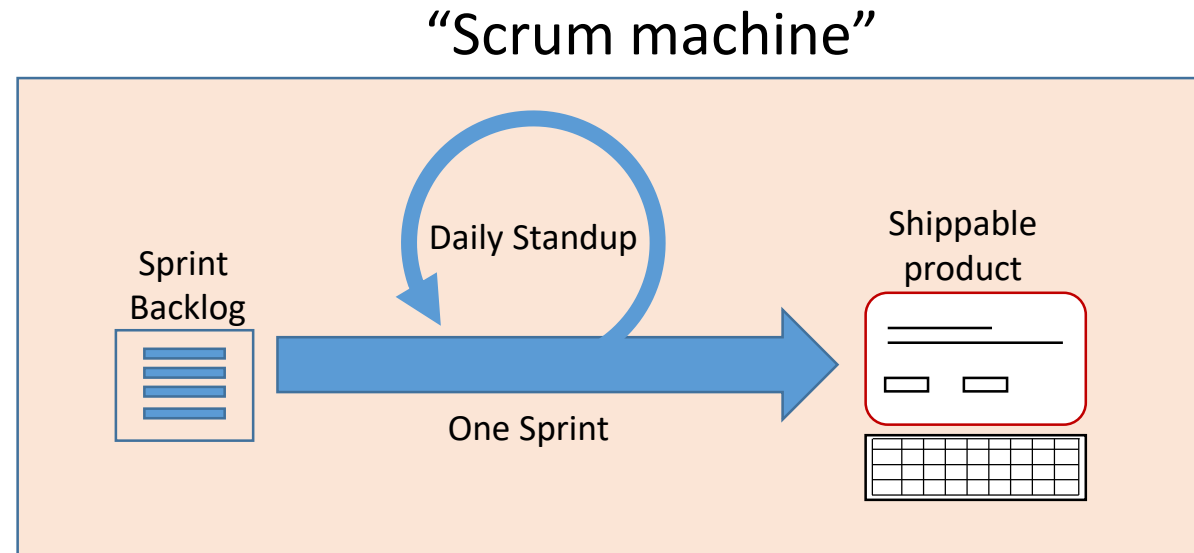
- Understand burn-up charts and their use in project planning and estimation.
- Understand defect tracking and the value for agile teams.
- Consider approaches for communicating project planning issues for stakeholders and leadership.

Velocity & Project Planning

Full agile software development lifecycle



Velocity is a measure of how fast we build shippable product



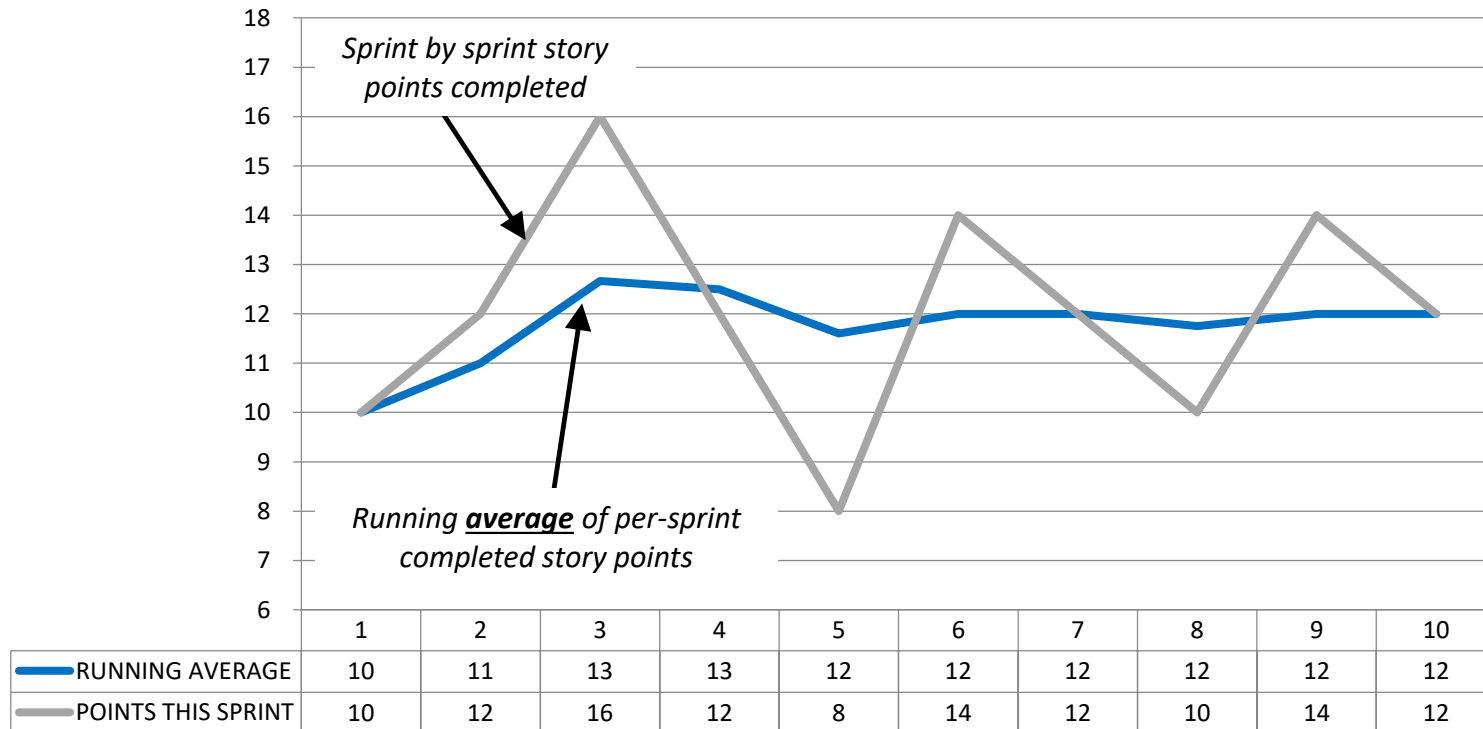
After a team has had several sprints of working together the velocity metric should be *reasonably* accurate.

A few notes about velocity...

- Velocity is specific to each team.
- **NEVER** compare teams by their velocity for any purpose.
 - This will cause all kinds of team and management problems that you do not want.
- Velocity is a good measure of how a team does the kind of work they've been doing.
 - If the nature of the work changes, velocity might change as well.
- Some things that will cause team velocity to “take a hit”:
 - Changing team members - adding or removing, doesn't matter.
 - Changing the type of *work*, the *technology*, or the *product*.
 - Changes to expected work practices or schedules.
 - Changes to definitions of *ready* or *done*.

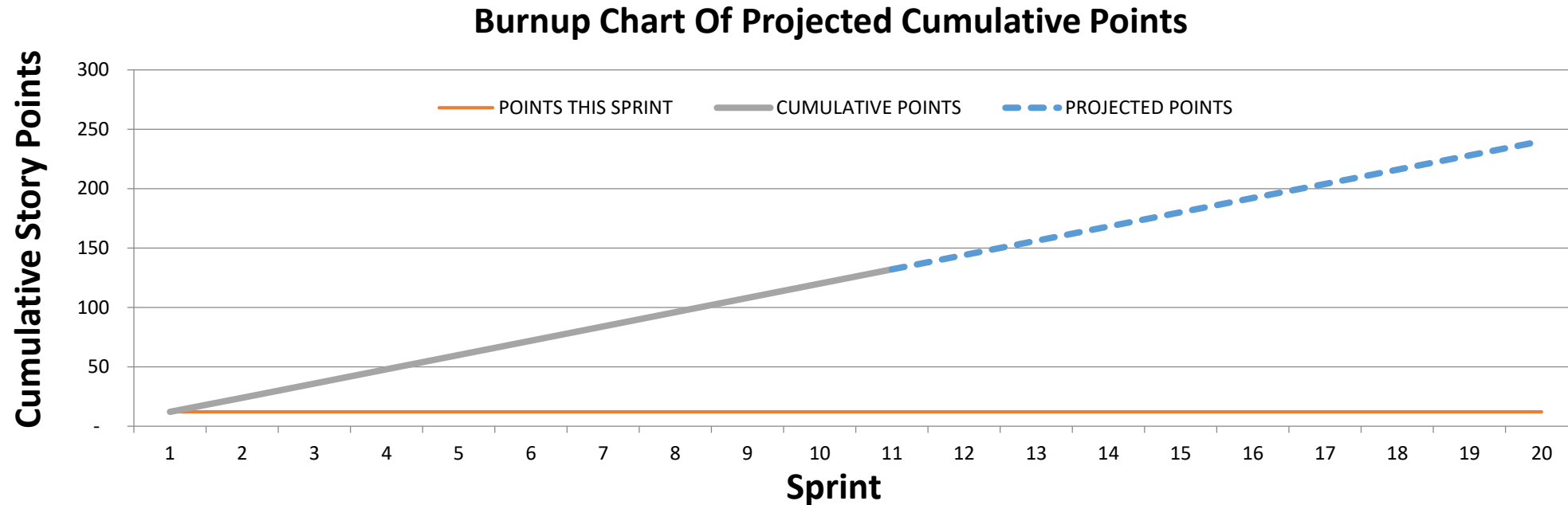
Tracking & estimating velocity

Team Velocity Over Time



- Over time, the story points the team accomplishes each sprint will settle into a fairly reliable average.
- This is referred to as the team's *velocity*.
- This is an estimate of the work the team produces over time.

Burnup chart



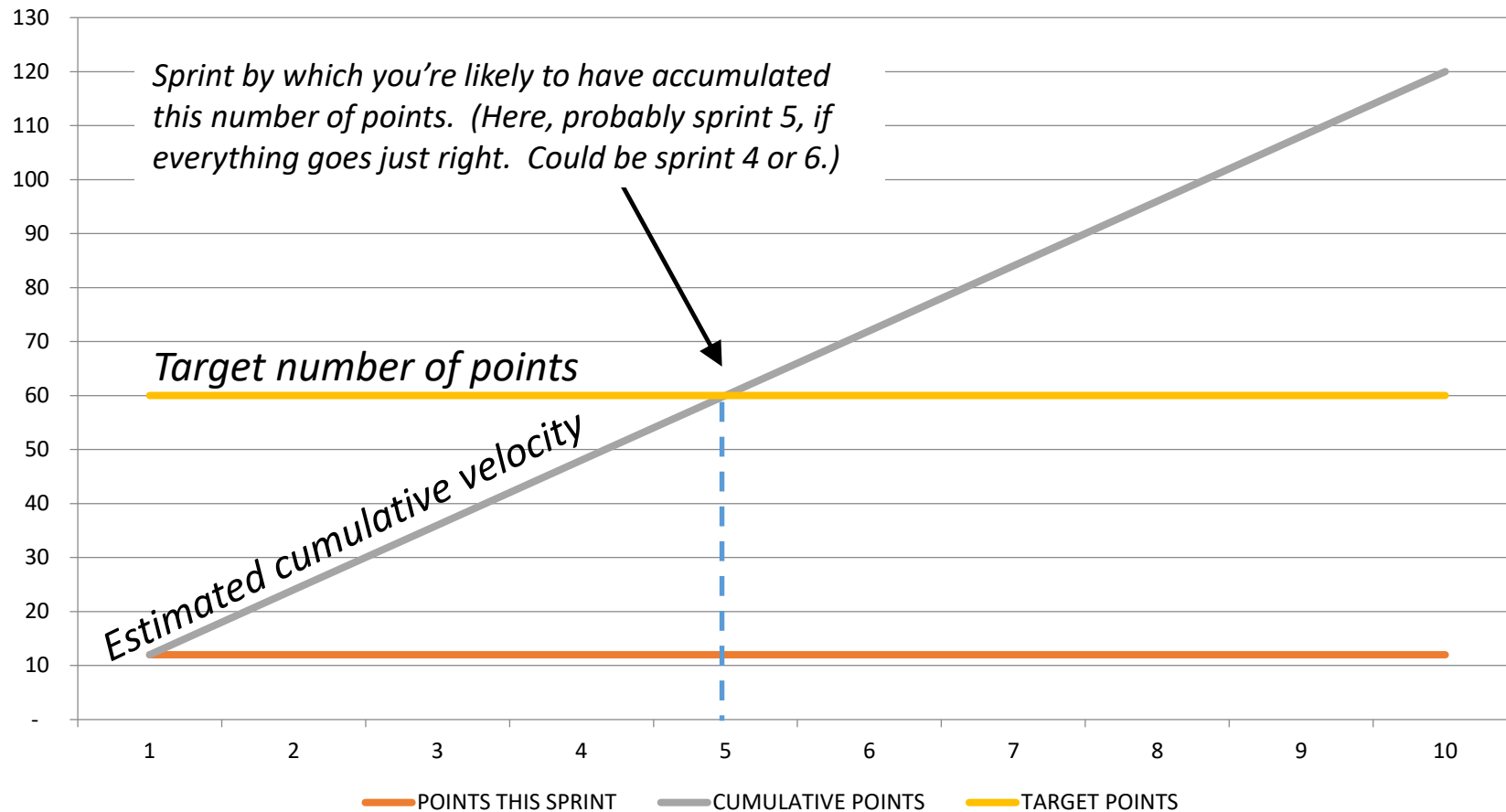
Burnup charts are used to project estimated progress for future sprints.

This is used in conjunction with rough estimation to determine how much the team might deliver in the next several sprints when doing release planning or product road mapping.

The x-axis here is entire sprints of already accomplished work and a projection of what is likely to be possible for the team in the future.

Velocity and release planning

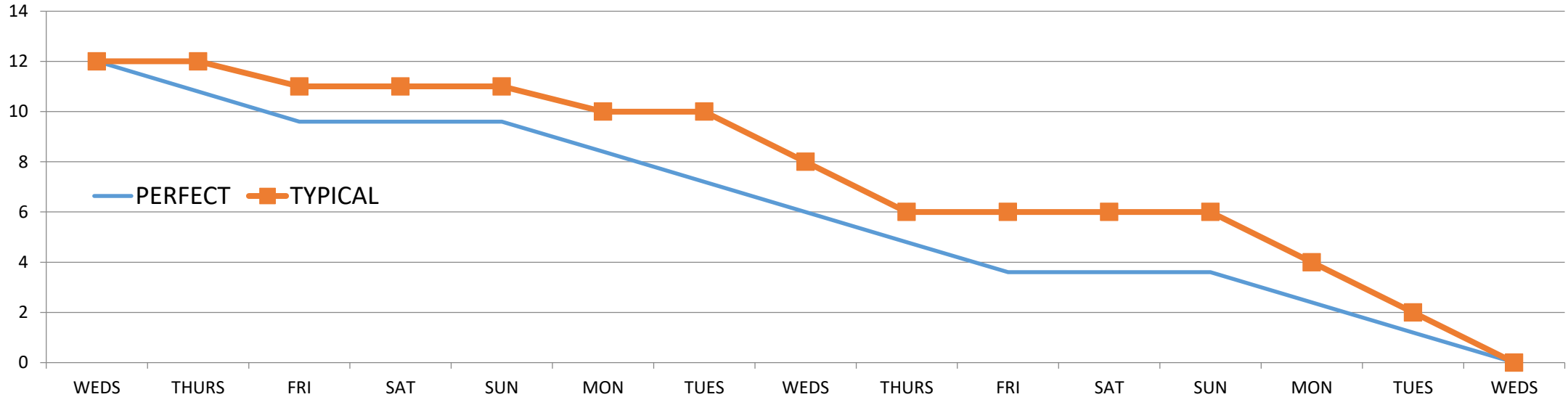
ESTIMATING A RELEASE DATE



- Once a team has a reliable estimate of their velocity, they can use it in release planning
- By dividing the estimate of the release by the number of points they generally achieve each sprint, they can give a reasonable answer about how long it will take to achieve these results
- IF you're doing a *date-driven release*, and this date is past what you want, you'll have to cut scope
- IF you're doing a *feature based release*, this tells you about how soon you can release this feature

Burndown chart

SAMPLE SPRINT BURNDOWN CHART



More an aspect for the scrum master and overall team management **within** a sprint, the burndown chart shows how the team is progressing within the sprint period towards the committed sprint goal.

The two lines are the “straight line” progress (which is almost never how it works!) and the actual plotted progress over the period of the sprint.

Typical shape feature: very little work is completed (“done”) in the first few days, then things speed up as work is checked in and QA passed.

Tracking bugs and defects

Some terms to help us talk about this:

Bug or Defect

- Something that is *not working as designed*.
- If something IS working *as designed* but you don't like how it works that is **not** a bug. That is new work.

New bugs

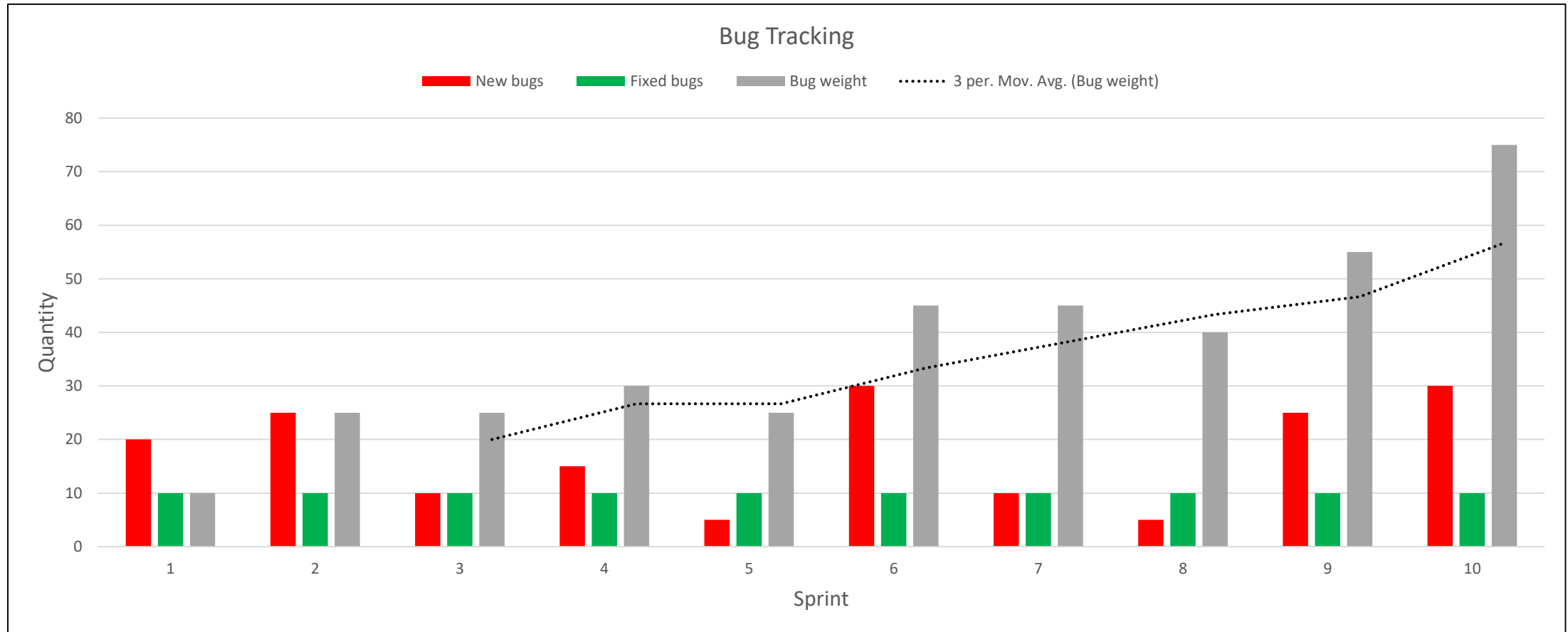
- Newly reported errors.
- Some teams limit this to “validated” bugs, or distinguish between them.

Fixed bugs

- The bugs you have corrected this period.
- Bug “weight”
 - The accumulated number of bugs that are not fixed.

Bug chart

Dotted trendline shown here is a moving 3- sprint average.



A few things about bugs & bug tracking

- Fixing bugs always requires capacity.
- A typical best practice is for teams to fix their own bugs.
- There are several schools of thought about how to approach bugs:
 - Same team as wrote the software.
 - Dedicated portion of capacity vs. fitting them in backlog like any PBI.

Communicating project status

Things you get asked about project status:

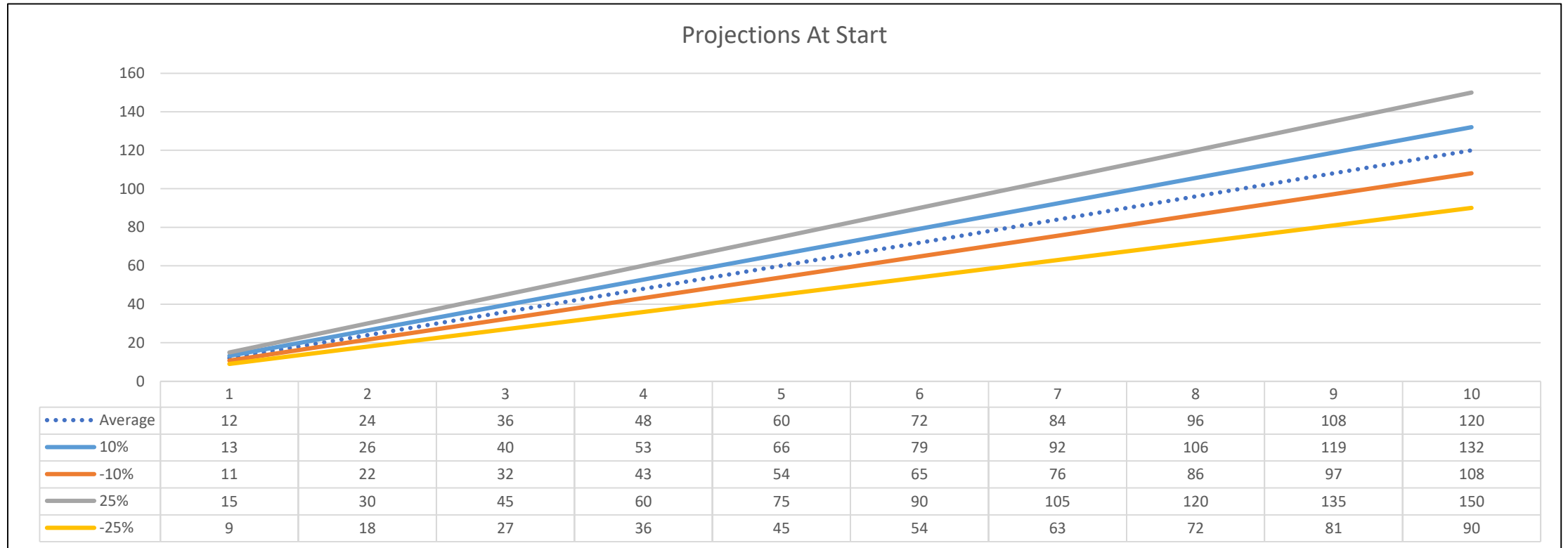
- “How is it going?”
- “When will you be done?”
- “When can we release?”
- “Will you be done with everything by the release date?”
- “How much do you have left?”

...and of course, more variations all having to do with *what & when*.

Your best answer is a range answer

- Range answers communicate the uncertainty inherent in agile work.
- In planning, this is sometimes called “The Cone of Uncertainty”.

Projection & range answers early in the release

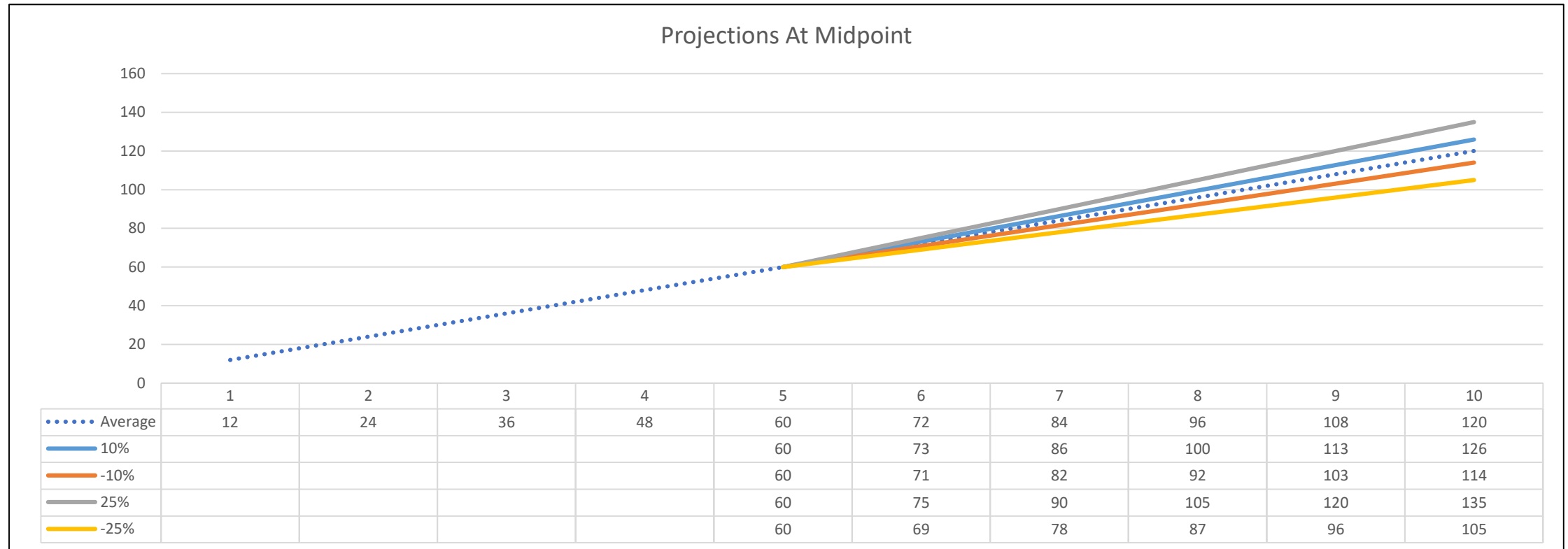


At the beginning, my range answer is:

“In 10 sprints, the team will probably have accomplished between 108 and 132 points.” *(Range of 24 points.)*

“If things are radically different than we expect it could be as low as 90, or as good as 150.” *(Range of 60 points.)*

Projection & range answers midpoint in the release



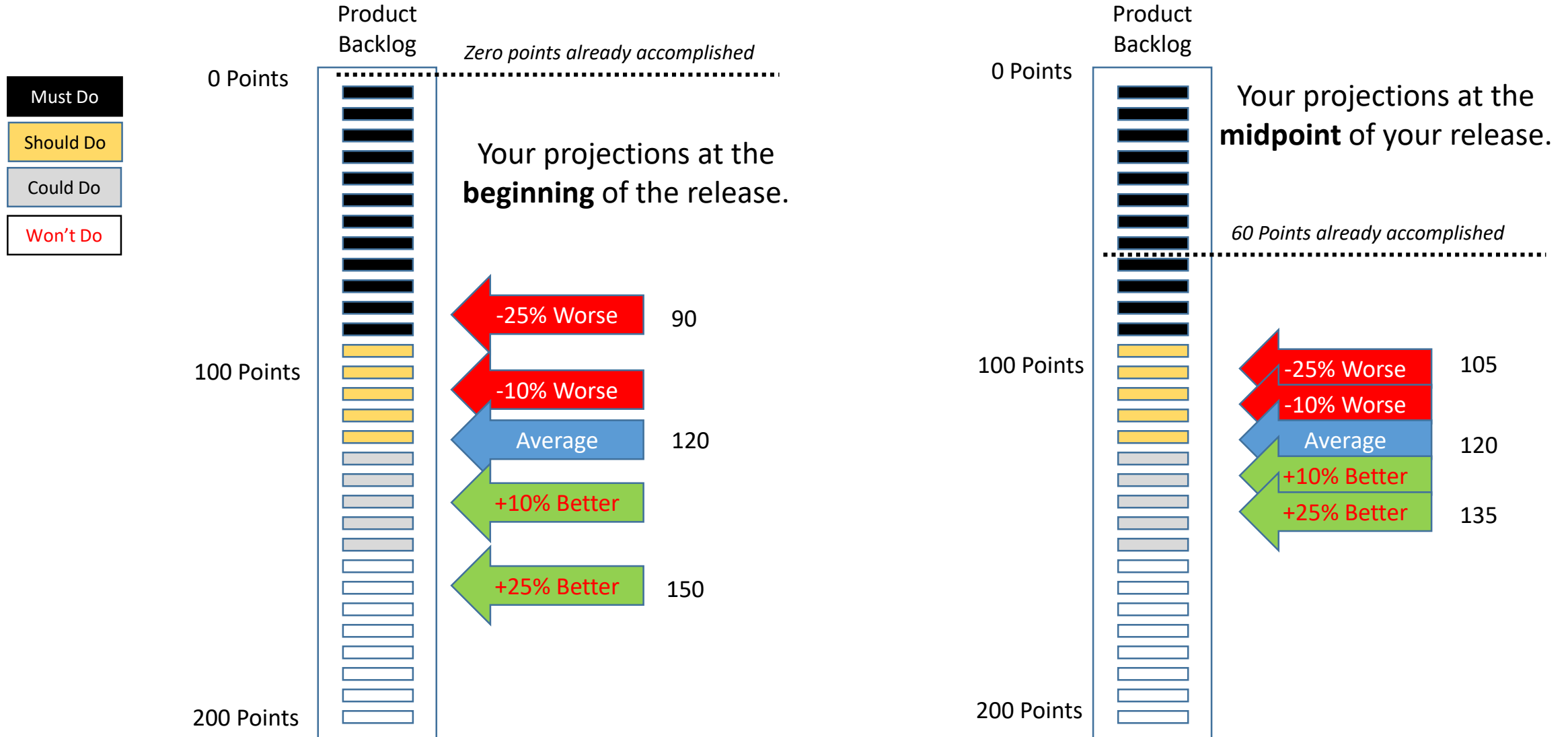
At the midpoint, my range answer is:

“So far, we’ve held a sprint average of 12 points, totaling 60 so far.”

“In 5 more sprints, the team will probably have accomplished a total of between 114 and 126.” *(Range of 12 points.)*

“If things are radically different than we expect it could be as low as 105, or as good as 135.” *(Range of 30 points.)*

“Given our velocity and projections, we are likely to get to [this point] on the backlog....”



And now, how you say it to real people:

Question	Answer
Fixed date release How are you doing?	<p><i>At beginning:</i> We think it is likely we will get through all the “Must Do” items by the end of sprint 10. We are not yet sure if we will get to any of the “Should Do” items, it’s too early to have confidence one way or the other. Ask me again in three sprints and I’ll know more.</p> <p><i>At midpoint:</i> We are almost certainly going to get through all the “Must Do” items, probably by sprint 7 or 8. We are very likely to get through most, if not all, of the “Should Do” items by sprint 10. We probably won’t get further than that.</p>
Feature based release When can we release?	<p><i>At beginning:</i> Assuming we have our usual velocity, you can have the “Must Do” items in roughly 7 to 10 sprints.</p> <p><i>At midpoint:</i> It is sprint 5 and we’ve gotten 2/3 of the “Must Do” items done. We are very likely to complete the rest of them by the end of sprint 7 or 8. If you want some or all of the “Should Do” items, we will probably have to run to sprint 10.</p>

Some things to consider

- If you tackled the hardest, most risky, or novel parts of the work early it is possible that velocity will have been *lower at the start* and will *increase towards the end*. The opposite is also true.
- Factor in holidays, PTO, training, and anything else that would significantly affect the total points you can expect in the release.
 - As best you can, reflect this in your projections of any given sprint.
 - *Example: You know that during sprint 6, several very experienced members of your team are going to a conference for ½ the sprint, so adjust your points expected for that sprint **down**.*

...Remember, always take input from the team about the work projection.

Initial Rough Work Planning *(T-shirt sizes and sprints, not story points)*

DEFINITIONS	LOW	HIGH
Small	0.50	1.00
Medium	1.00	3.00
Large	3.00	5.00
Xtra Large	5.00	8.00
Release Velocity	24	
<i>Three teams, each with 8 Sprints</i>		

EPIC	T-SHIRT SIZE	Item		Cumulative		Cumulative Variability			
		Low	High	Low	High				
1	Large	3.0	5.0	3.0	5.0	2.0	BEST CASE	AVERAGE CASE	WORST CASE
2	Small	0.5	1.0	3.5	6.0	2.5	BEST CASE	AVERAGE CASE	WORST CASE
3	Medium	1.0	3.0	4.5	9.0	4.5	BEST CASE	AVERAGE CASE	WORST CASE
4	Medium	1.0	3.0	5.5	12.0	6.5	BEST CASE	AVERAGE CASE	WORST CASE
5	Medium	1.0	3.0	6.5	15.0	8.5	BEST CASE	AVERAGE CASE	WORST CASE
6	Small	0.5	1.0	7.0	16.0	9.0	BEST CASE	AVERAGE CASE	WORST CASE
7	Small	0.5	1.0	7.5	17.0	9.5	BEST CASE	AVERAGE CASE	WORST CASE
8	Xtra Large	5.0	8.0	12.5	25.0	12.5	BEST CASE	AVERAGE CASE	
9	Large	3.0	5.0	15.5	30.0	14.5	BEST CASE	AVERAGE CASE	
10	Large	3.0	5.0	18.5	35.0	16.5	BEST CASE		
11	Small	0.5	1.0	19.0	36.0	17.0	BEST CASE		
12	Medium	1.0	3.0	20.0	39.0	19.0	BEST CASE		
13	Large	3.0	5.0	23.0	44.0	21.0	BEST CASE		
14	Medium	1.0	3.0	24.0	47.0	23.0	BEST CASE		
15	Medium	1.0	3.0	25.0	50.0	25.0			
16	Large	3.0	5.0	28.0	55.0	27.0			
17	Large	3.0	5.0	31.0	60.0	29.0			
18	Large	3.0	5.0	34.0	65.0	31.0			
19	Medium	1.0	3.0	35.0	68.0	33.0			
20	Small	0.5	1.0	35.5	69.0	33.5			

The units used in this example is SPRINTS, not points.

Read the "Definitions" table like this:

*"A **small epic** will take between one-half and one whole sprint for one team."*

*"A **medium epic** will take between one whole and three whole sprints for one team."*

Register at www.RylandLeyton.com for an excel version of this page.

About Ryland

Ryland Leyton, CBAP, PMP, CSM, is a business analyst, speaker, educator, Agile coach, and technology translator. He has worked in the technology sector since 1998, starting off with database and web programming, gradually moving through project management and finding his passion in the BA field. Ryland is employed as Lead Business Analyst at Aptos.com.

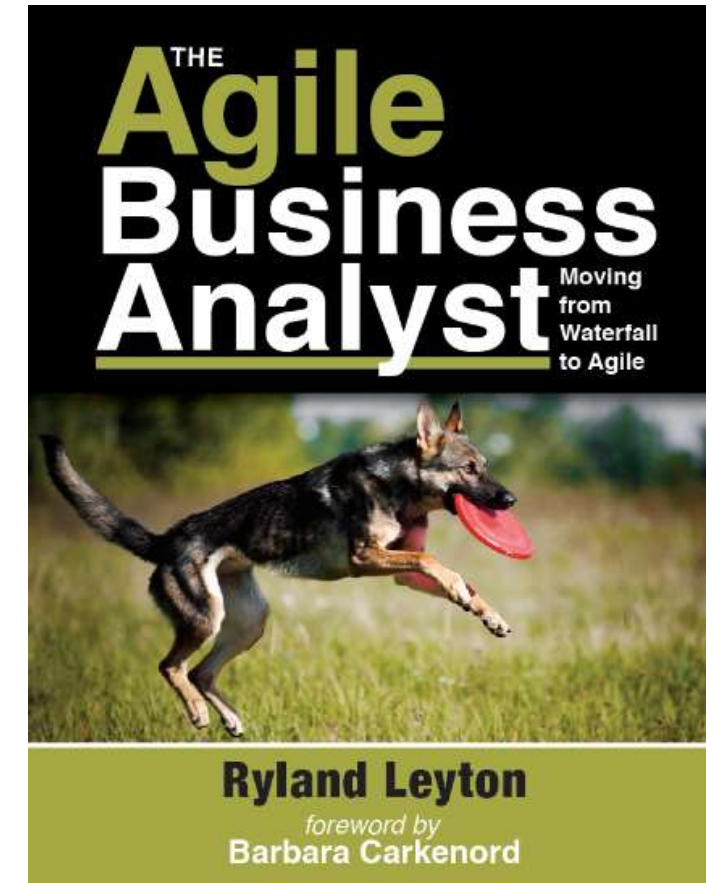
Ryland is passionate about strong analysis practice and prefers Agile environments where possible. He has built both Agile and waterfall SDLC processes for development teams, customizing each one to the challenges facing that particular client group.

He is an active member of his local chapter of the IIBA, speaks at local and international conferences, and serves as an Agile coach and educator.

Ryland is a core team author of *V2 of the Agile Extension to the BABOK*.

Ryland's book, "*The Agile Business Analyst*" has received excellent critical reviews and is available on Amazon.com.

[Available at www.Amazon.com](http://www.Amazon.com)



www.RylandLeyton.com

RylandTheBA@GMail.com