



Actionable Agile Metrics

Chris Schmitt
September 30, 2020

Agile metrics

- *metrics relevant to our Agility*

Actionable

- *metrics we can use:*
 - *to better know when we be done?*
 - *to understand how we are doing?*
 - *and see if we can improve on that*

So, what should we measure?

Hint from Agile Principle #3

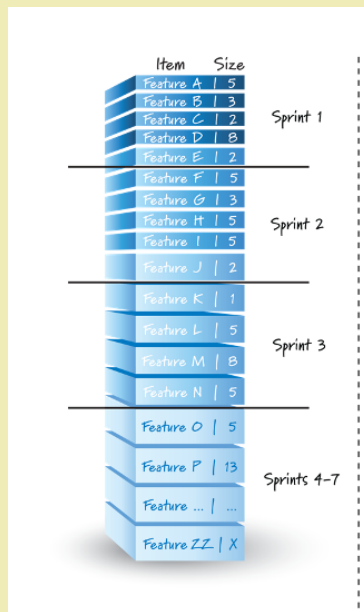
*Deliver working software frequently,
from a couple of weeks to a couple of months,
with a preference to the shorter timescale.*

Our delivery goals are mostly bigger things

Feature Delivery



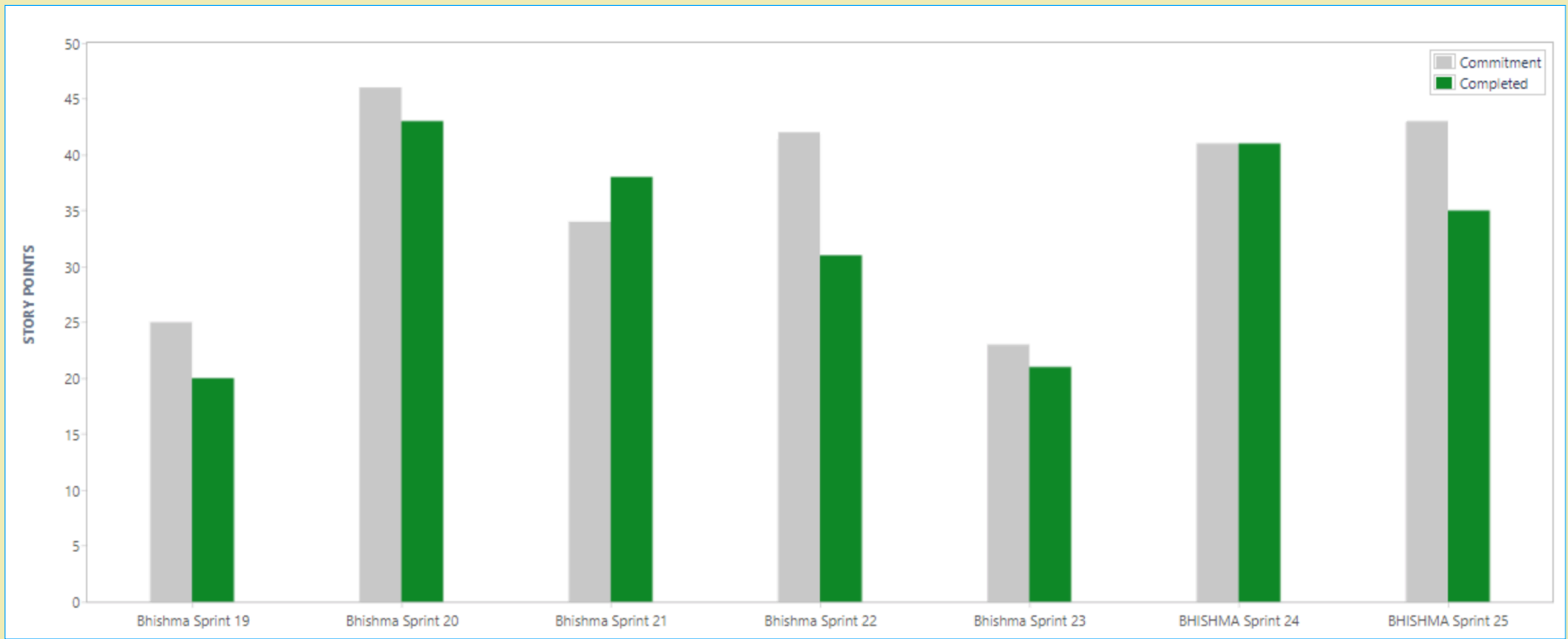
*Those bigger things consist of a number of stories.
How many stories until we're done?*



courtesy ©2013 Comakers LLC and Jeff Patton

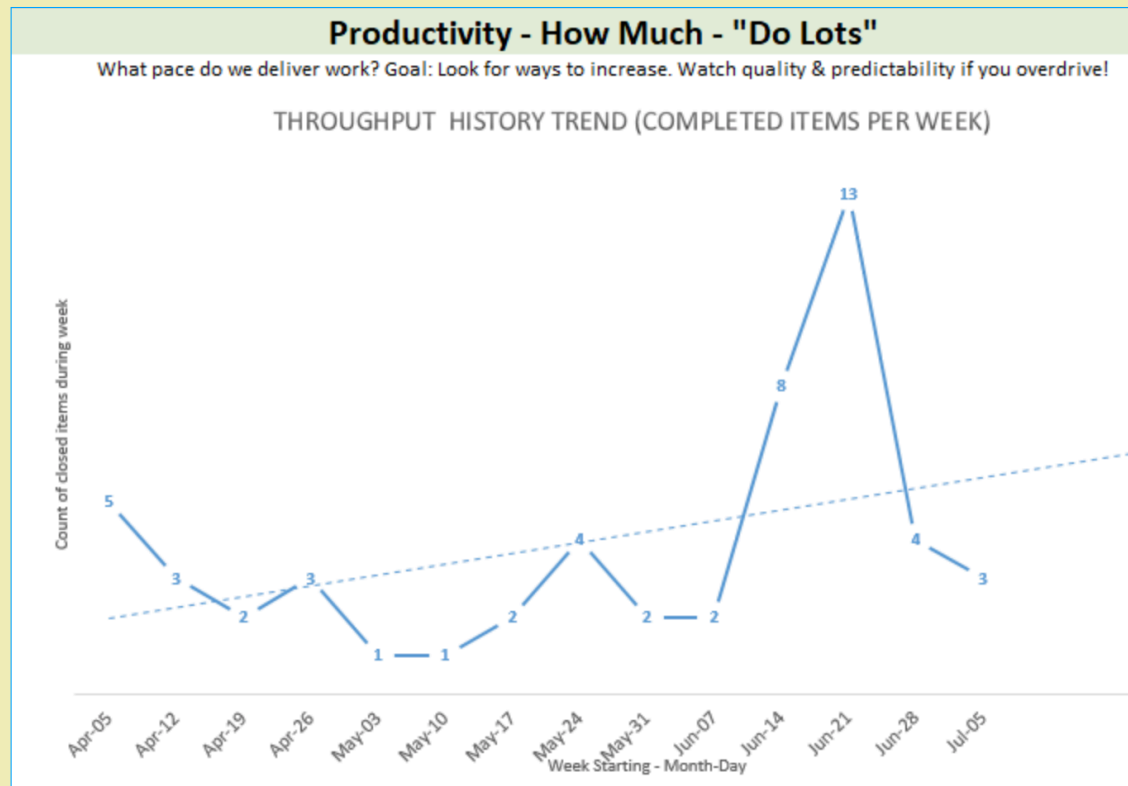
How many stories can I get done each week?

Actionable: Sprint Velocity Chart



Velocity helps us understand how many storypoints we deliver per Sprint

Actionable: Weekly Throughput Chart

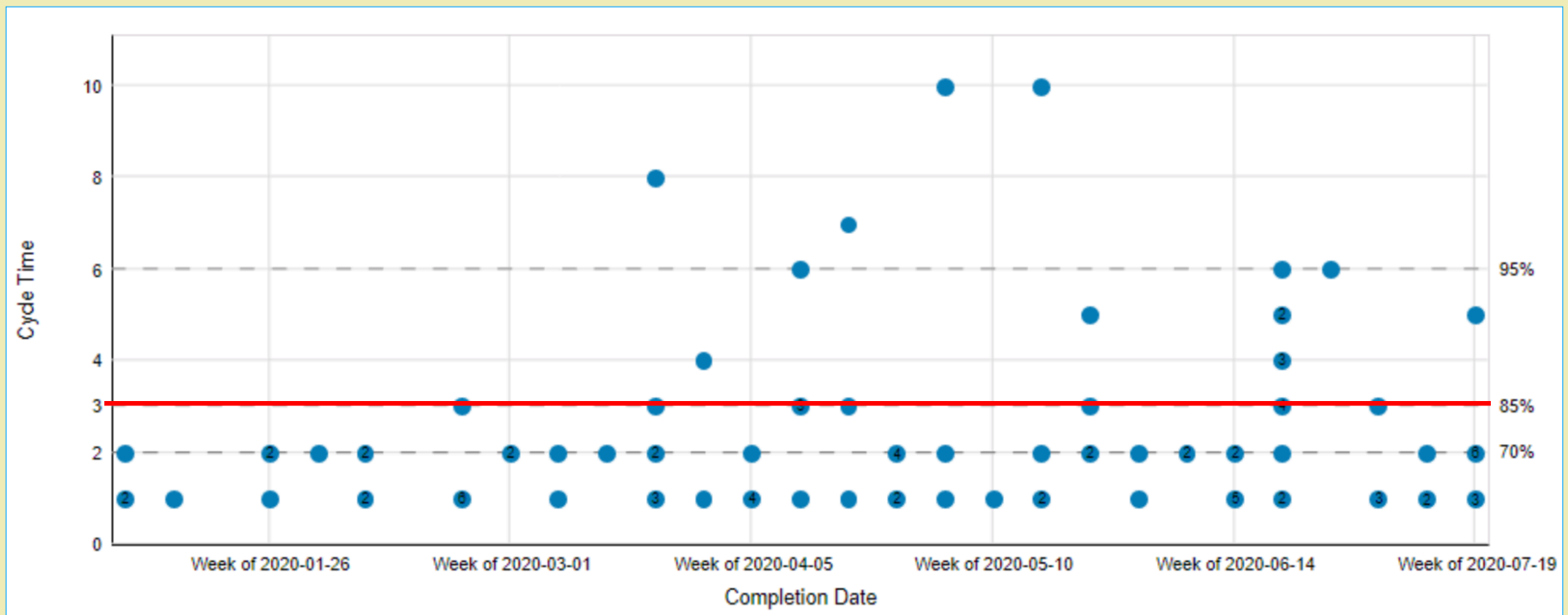


Similarly, Throughput tells us how many stories we finish per week

But how long do Stories really take?

This chart shows done stories and how long each story takes

This chart shows done stories and how long each story takes



Gives us a baseline to understand and seek to improve how long stories take

Knowledge Work

Lots of variation in it due to:

- Degree of understanding and preparedness
- Size
- Complexity
- Risk
- Wait time

We want to take out or at least minimize variation in order to improve predictability and performance

What else influences our delivery time?

**Turns out there is a intimate relationship
that influences delivery...**

✓ **Throughput**

- (how many stories we deliver per week/Sprint?)

✓ **Lead/Cycle time**

- (how long do stories take to finish?)

Work in Progress (WIP)

- (how many stories are we working on at same time)

Known as Little's Law

$$AverageThroughput = \frac{AverageWorkInProgress}{AverageCycleTime}$$

It speaks to the system dynamics between these measures...

and of what is possible

Actionable: Leveraging Little's Law

$$AverageThroughput = \frac{AverageWorkInProgress}{AverageCycleTime}$$

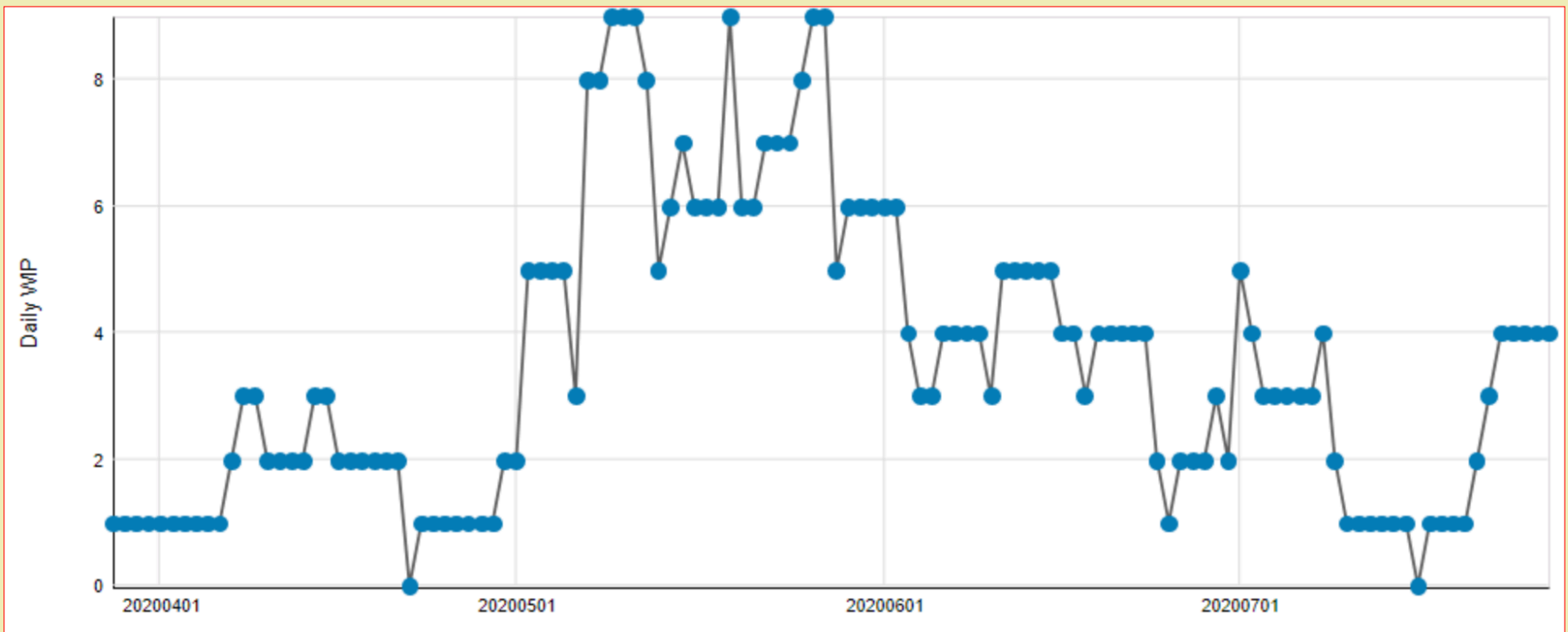
- If our Throughput is fairly steady,
and **if we reduce or limit our Work in Progress (WIP)**
story cycle time should get shorter (do stuff faster)

and **if our story cycle time gets shorter,**
our throughput should go up (more stuff delivered)

*Let's look a little closer at Work in Progress
(WIP)*

Actionable: WIP Run Chart

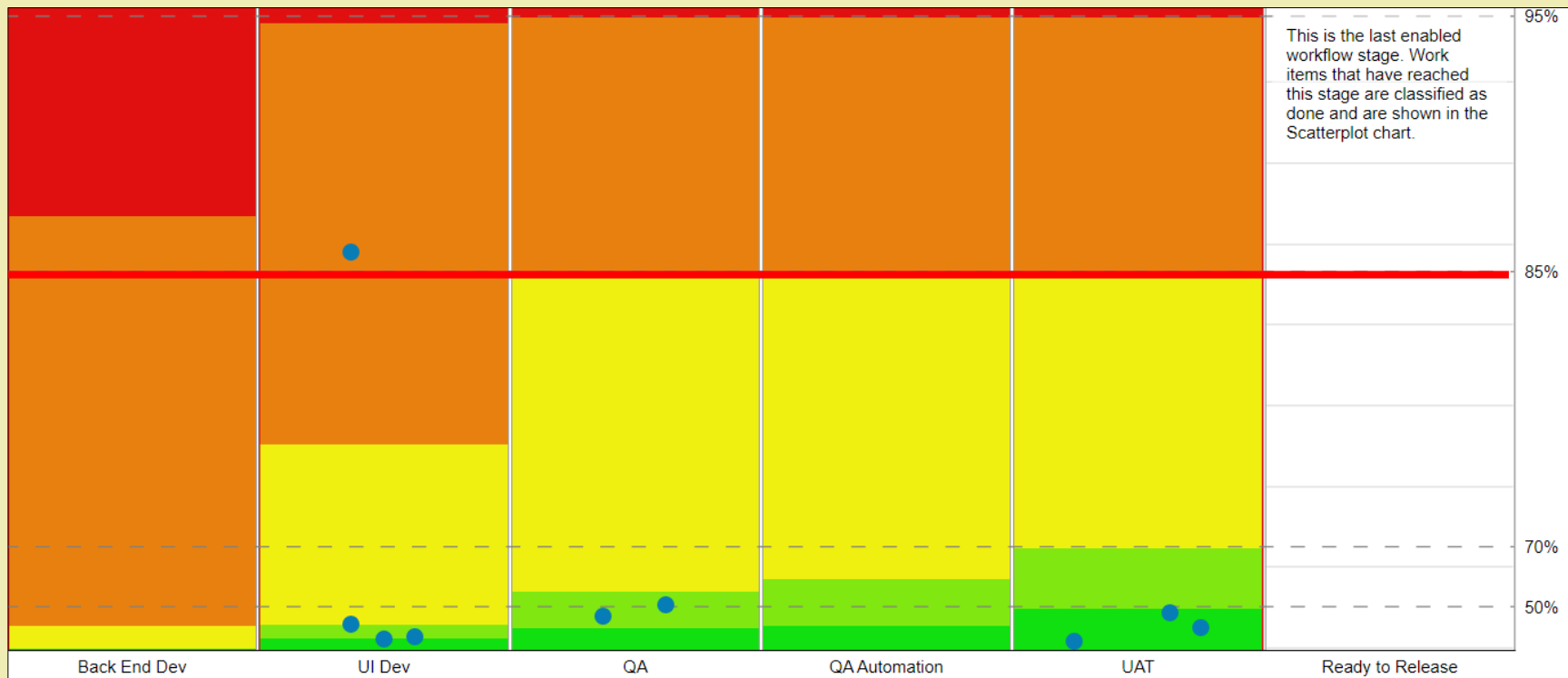
This chart shows variation in how many stories (WIP) we are working on



If we can stabilize/limit WIP, we will become more reliable, etc.

Actionable: WIP Aging tracks story progress

This chart shows how much stories in progress (WIP) are in time or drifting



- Team can swarm on drifting stories - tightening up their cycle time
- Team could also see if one stage is a bottleneck

What else slows down or impacts delivery?

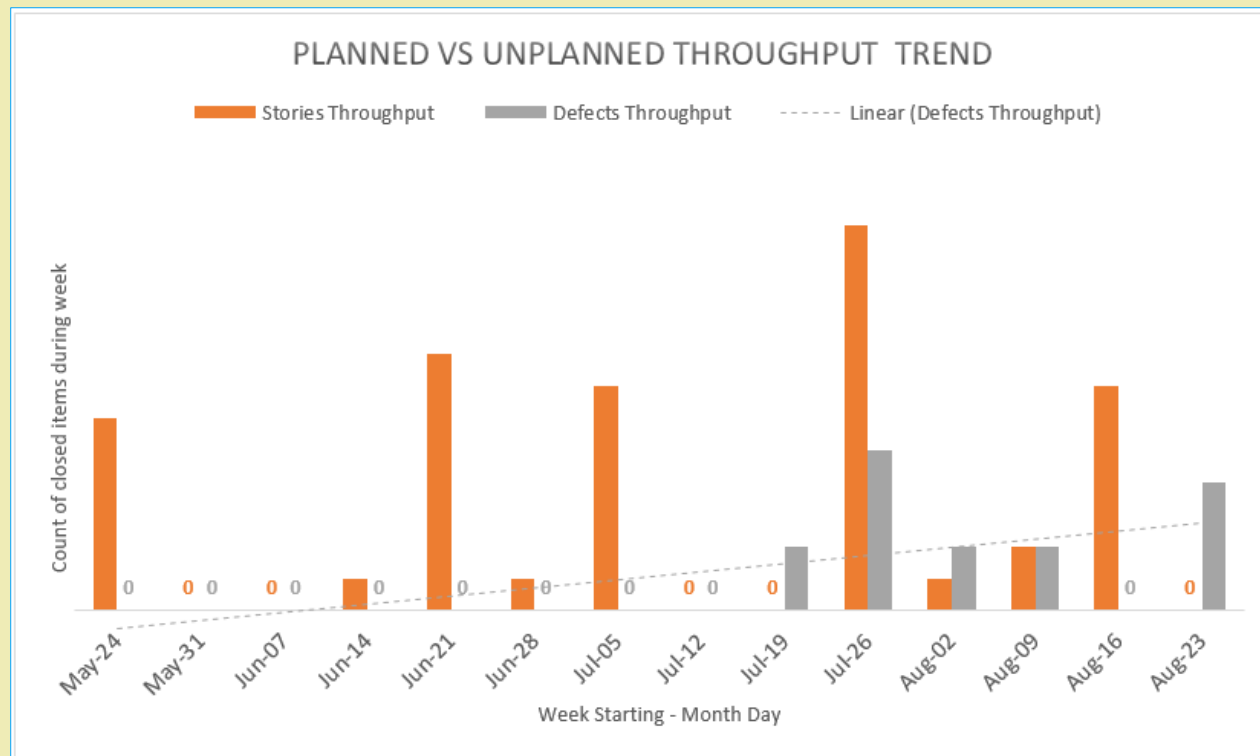
The Thing about Defects

- Yes, Defects are quality problems that should not get out to the customer
- But they also represent some gap in our process or understanding. Do we catch them early? Why did it appear at all? Missing Acceptance Criteria? Misinterpretation of them?
- Along with Impediments and Delay, they slow our forward progress by making us work on unplanned stories. Their presence and variation negatively impact predictability.



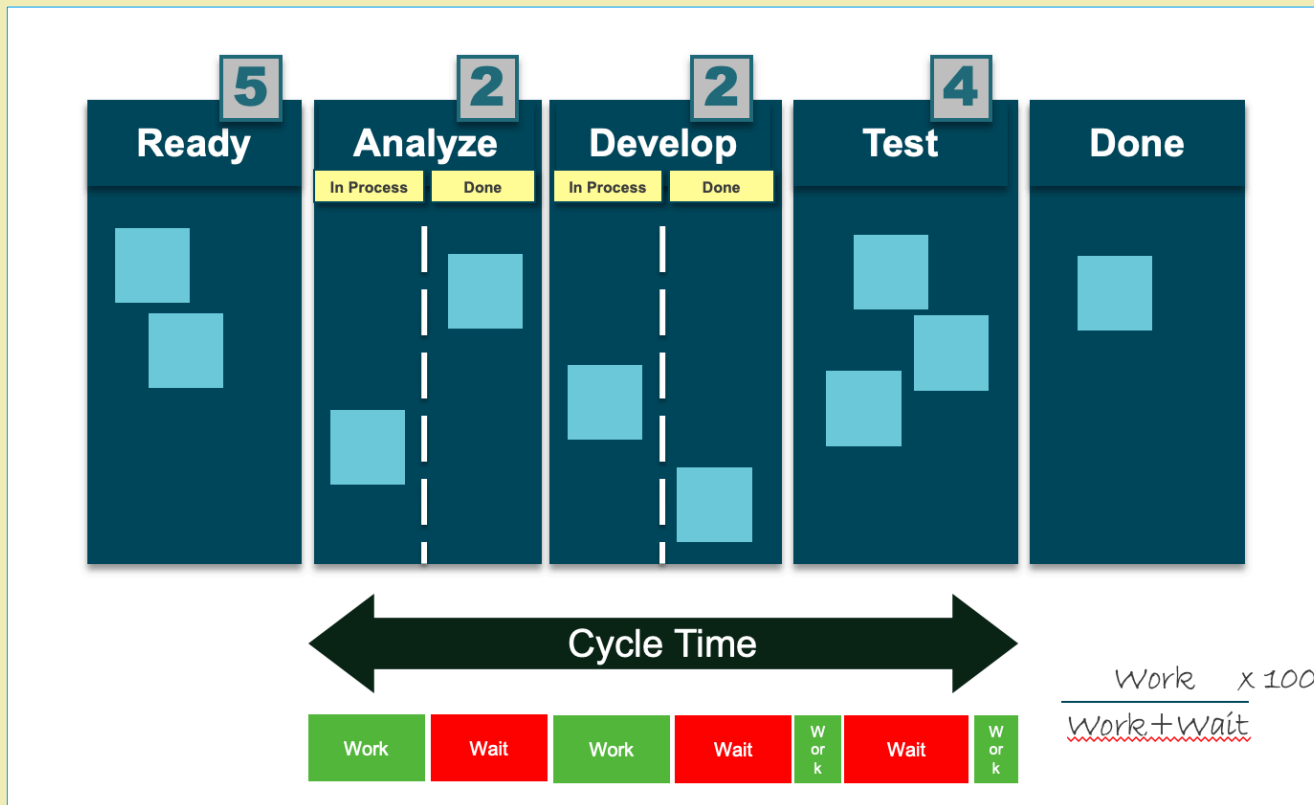
Actionable: Weekly Defect Load

Orange shows story completions, Gray shows defect completions



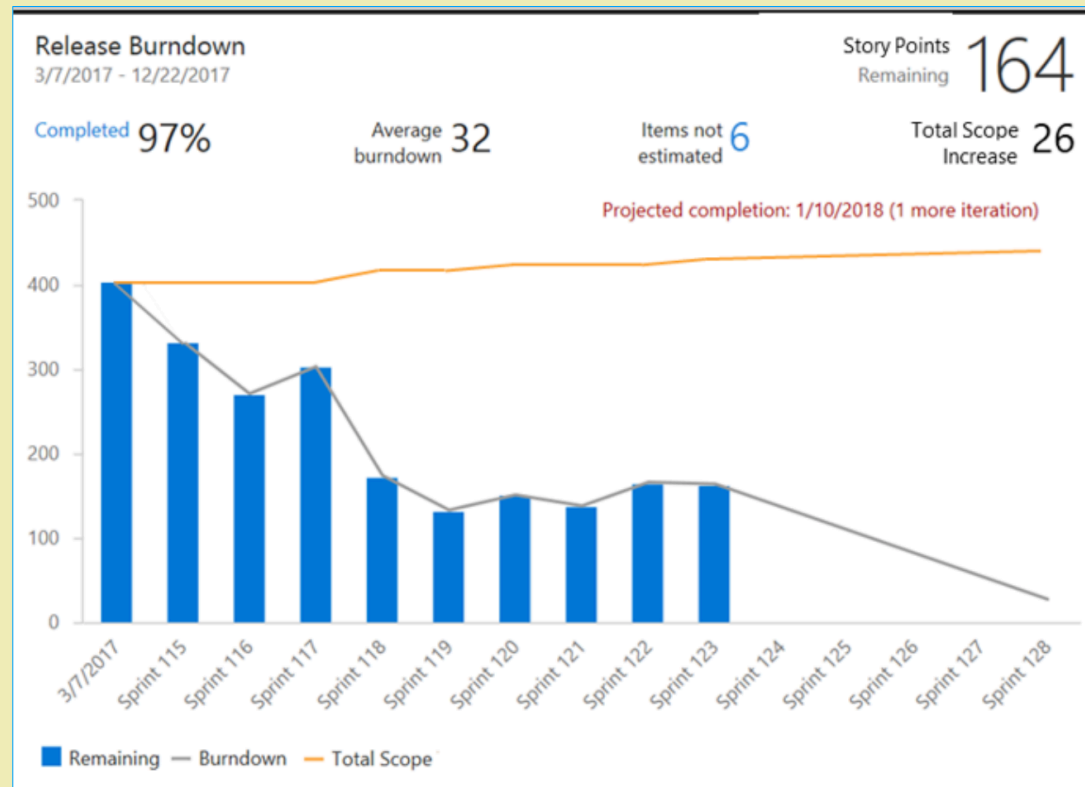
Investigate why defects/quality occurs and eliminate unplanned work load

Actionable: Flow Efficiency



Are we on track?

Actionable: Burndown Charts measure Progress



Are we on track to meet our Delivery date?

*Use Metrics in your Retros and Standups
to Reflect and Grow*

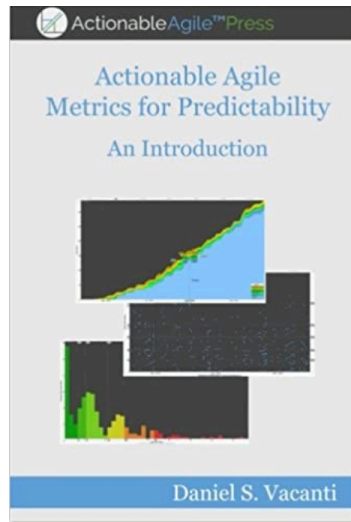
Summary

- Agile is about delivering quality Features frequently
- **How many stories in a Feature?** – Storymap/Backlog
- **How many stories per week?** – Velocity/Throughput
- **How long do they take?** – Lead/Cycle Time
- **Little's Law shows:** limiting WIP - can reduce Cycle time - increasing Throughput – WIP run/Aging

Summary (cont.)

- **Reduce defects to increase Throughput – Defect Load**
- **Watch Flow Efficiency to improve Cycle Time**
- **Find ways to watch your progress – Burndown/up**
- **Share metrics during Retros and Standups**

Further Reading

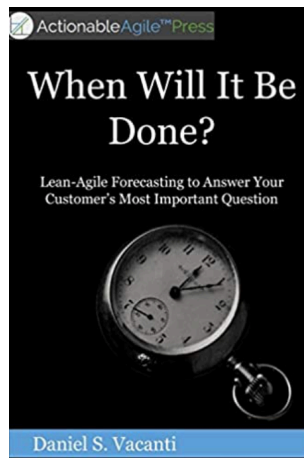
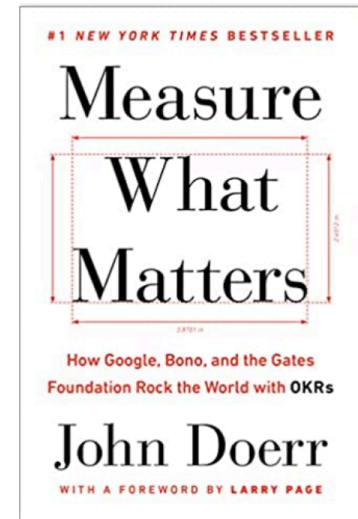


Focused Objective

Tools, consulting and training
for Agile Forecasting and Data

UPCOM

<https://www.focusedobjective.com/w/support>



<https://kanbanize.com/agile/project-management/agile-metrics>

Who is this guy?



- Chris Schmitt
- Lives in Pennington
- Has a wife, two daughters, two very cool cats
- Business Agility Consultant/Coach
- Doing Lean-Agile for almost 15 years
- Works currently for TD Ameritrade (Charles Schwab)
- Also consults privately through AgileDance
- cschmitt@agiledance.com



Credits

- ✧ Slides in this presentation contain items from the Visual AGILExicon®, which is a trademark of Innolution, LLC and Kenneth S. Rubin.
- ✧ The Visual AGILExicon is used and described in the book: ***Essential Scrum: A Practical Guide to the Most Popular Agile Process.***
- ✧ You can learn more about the Visual AGILExicon and permitted uses at: <http://innolution.com/resources/val-home-page>



Visual AGILExicon®