BITS Pilani
Pilani Campus

BITS Pilani presentation

K.Anantharaman
kanantharaman@wilp.bits-pilani.ac.in

innovate achieve lead

BITS Pilani
Pilani Campus

SE CZ 544 , Agile software processes – Agile Metrics and Tools

Agile Metrics

- Examples:
  - Velocity, Lead time, Cycle time, Charts, Escape defects and so on.
- Helps to assess the quality of a product and track team performance.

- The Concept:
  - Define Metrics that can be used by Agile teams and Team management, Agile metrics that matter.
- The Opportunity
  - Reduced costs, Increase Product Quality, Increased team satisfaction
- The Potential
  - Auto Generate using exposed APIs provided by various PM tools.

Source: Prachi Maini, Manager, QA Engineering, Morningstar, Inc.

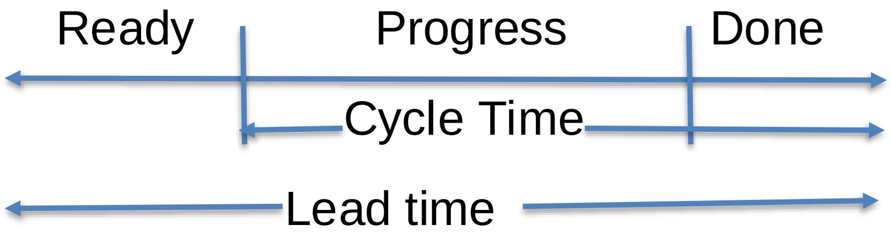
Quantitative & Qualitative Metric

- Quantitative Metric
  - Measurement number: Lead time, Number of defects ….
- Qualitative Metric
  - Based on subjective opinion: Maintainability, Team happiness index …

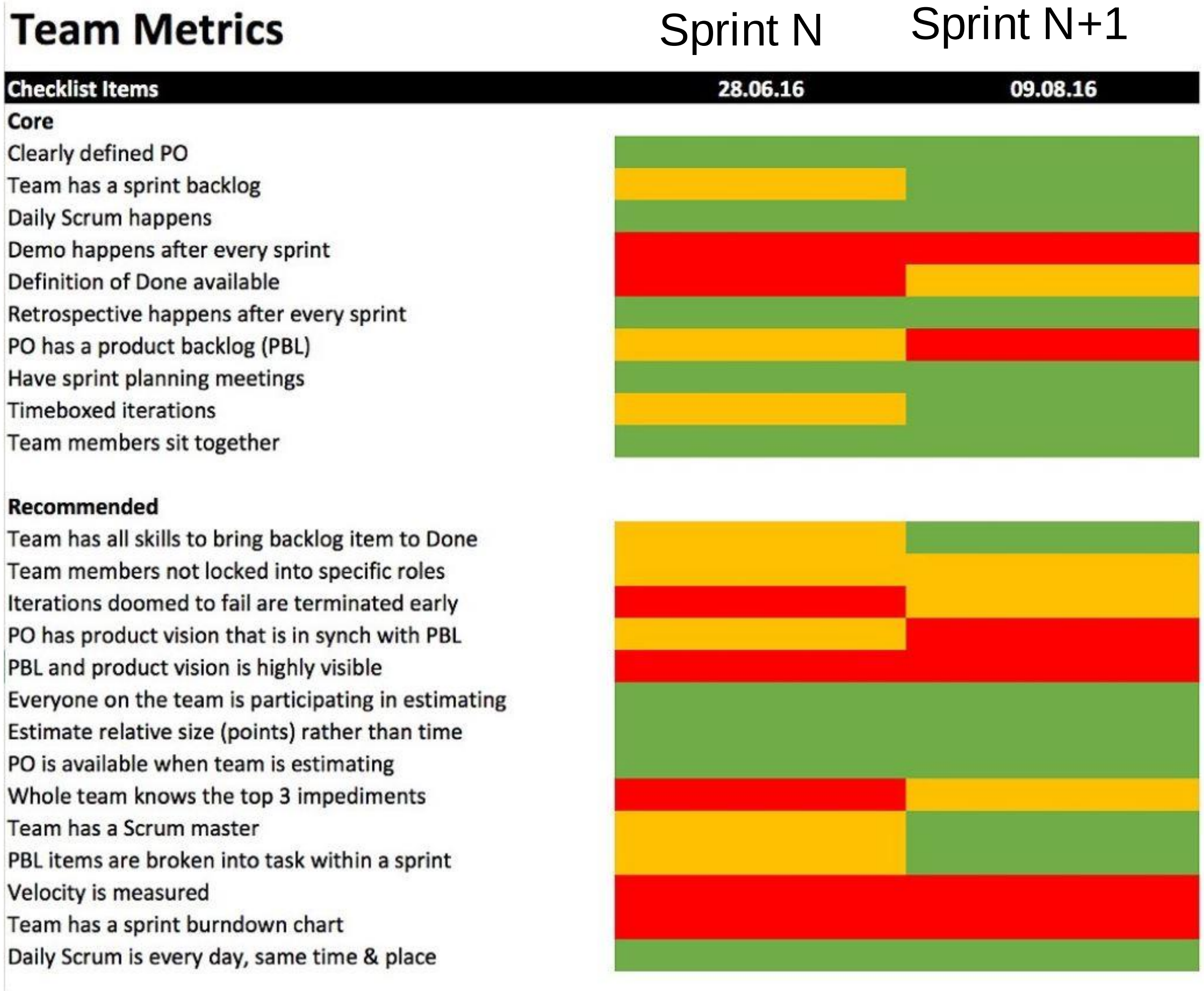
Source:https://www.infoq.com/articles/metrics-agile-teams/

An example Quantitative Metric

- Example: **Lead time** is a useful quantitative statistic for evaluating team performance.
- Determinant metrics:
  - A set of measurements related to a specific measurement
  - **Associated metrics:**
  - Flow efficiency (wait time)
  - Speeding tickets(%), (Tickets moves through multiple statuses)
  - Total sprint completion (Committed vs Actual Story points)
  - Defects returned from QA(%)
  - Escape defects(%)
  - Bug fixing Vs working on feature (% time)



An example - Good Qualitative Agile Metrics: Team Adoption to Agile



Team Metrics

Checklist Items	Sprint N 28.06.16	Sprint N+1 09.08.16
Core		
Clearly defined PO	🟩	🟩
Team has a sprint backlog	🟧	🟩
Daily Scrum happens	🟩	🟩
Demo happens after every sprint	🟥	🟥
Definition of Done available	🟥	🟧
Retrospective happens after every sprint	🟩	🟩
PO has a product backlog (PBL)	🟧	🟥
Have sprint planning meetings	🟩	🟩
Timeboxed iterations	🟧	🟩
Team members sit together	🟩	🟩
Recommended		
Team has all skills to bring backlog item to Done	🟧	🟩
Team members not locked into specific roles	🟧	🟧
Iterations doomed to fail are terminated early	🟥	🟧
PO has product vision that is in synch with PBL	🟧	🟥
PBL and product vision is highly visible	🟥	🟥
Everyone on the team is participating in estimating	🟩	🟩
Estimate relative size (points) rather than time	🟩	🟩
PO is available when team is estimating	🟩	🟩
Whole team knows the top 3 impediments	🟥	🟧
Team has a Scrum master	🟧	🟩
PBL items are broken into task within a sprint	🟧	🟩
Velocity is measured	🟥	🟥
Team has a sprint burndown chart	🟥	🟥
Daily Scrum is every day, same time & place	🟩	🟩

Green: It worked for the team.

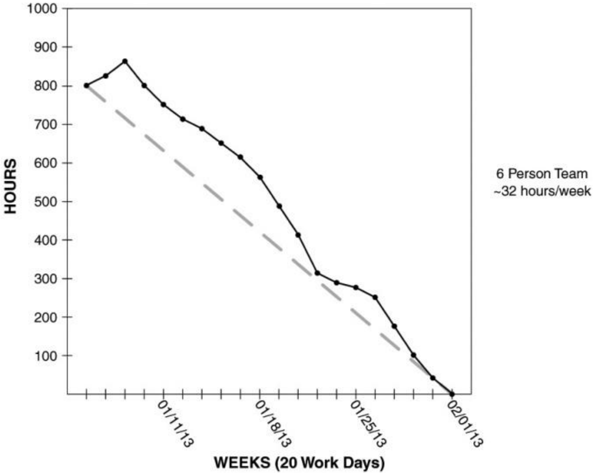
Orange: Room for improvement.

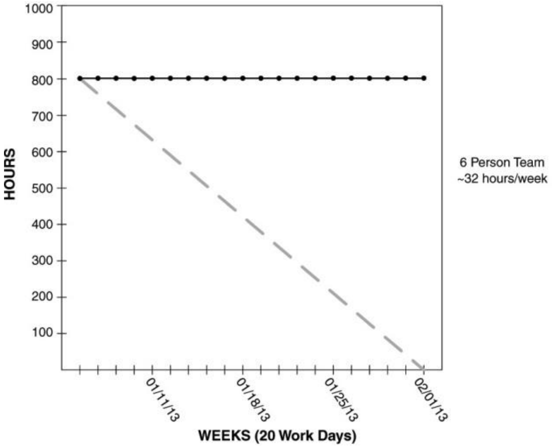
Red: Didn’t apply or the practice is failing

https://www.crisp.se/wp-content/uploads/2012/05/Scrum-checklist.pdf

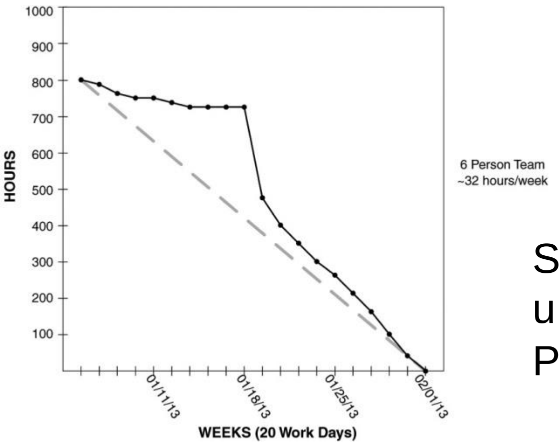
Metrics: WHAT ARE SOME TRENDS OF BURNDOWN CHARTS AND WHAT DO THE PATTERNS INDICATE?

Uptick: New tasks/Stories added. Issue if continues.



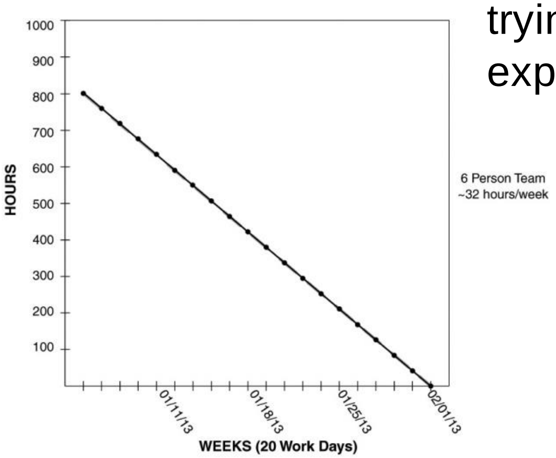


Flatline: Multiple reasons. Impediments, Task/Stories added at the same rate as work complete.



Sharpdrop: Team not updating the chart/ Pointed removed

Perfect line: Team trying to align with expectations



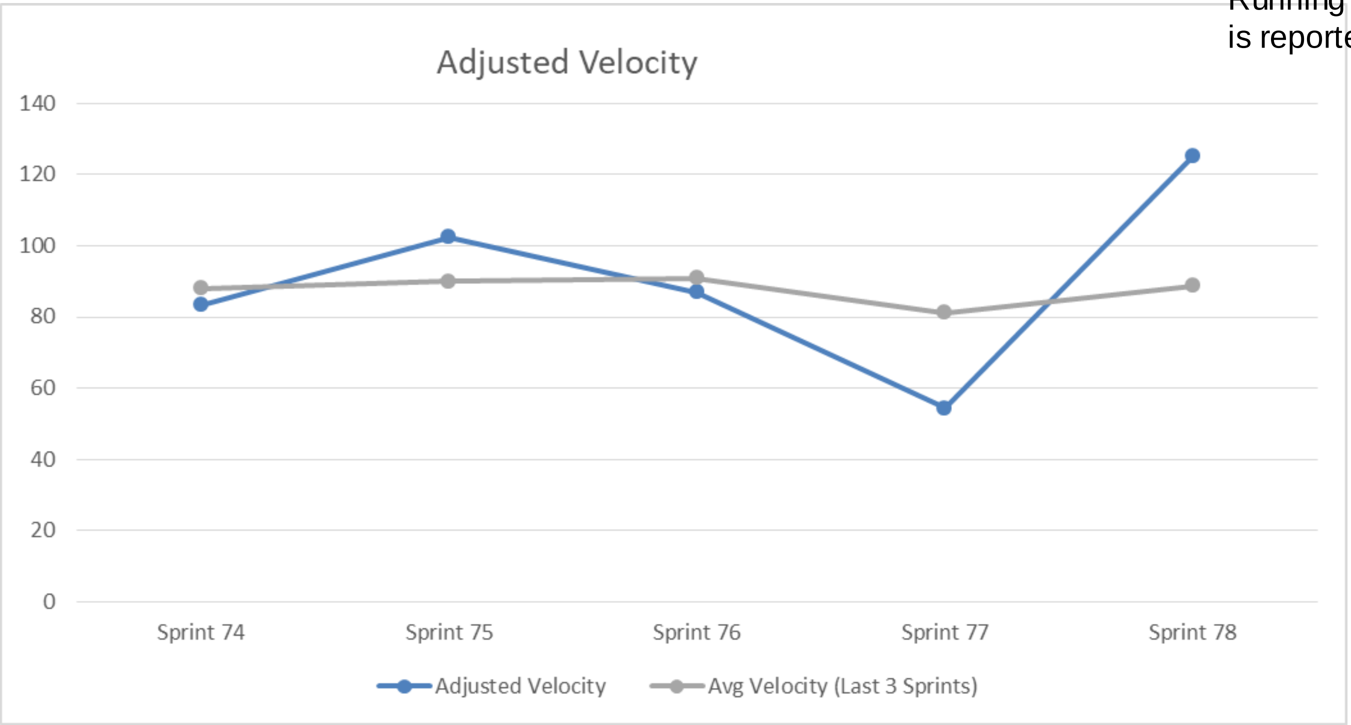
Velocity

Capacity	Sprint 74	Sprint 75	Sprint 76	Sprint 77	Sprint 78
Team Size	8	8	8	8	8
Available Days	80	80	80	80	80
Unavailable Days	10	12	11	5	0
Net Days (Capacity)	70	68	69	75	80
Velocity/					
Adjusted Velocity	83	102	87	54	125
Avg Velocity (Last 3 Sprints)	88	90	91	81	89

Velocity: Points of work completed by an agile team within a given sprint

Adjusted Velocity:
Points of work completed by an agile team accounting for holidays, team absence, improvements etc..

Running average of last three sprints is reported.



Source: Prachi Maini, Manager, QA Engineering, Morningstar, Inc.

What to Watch for?

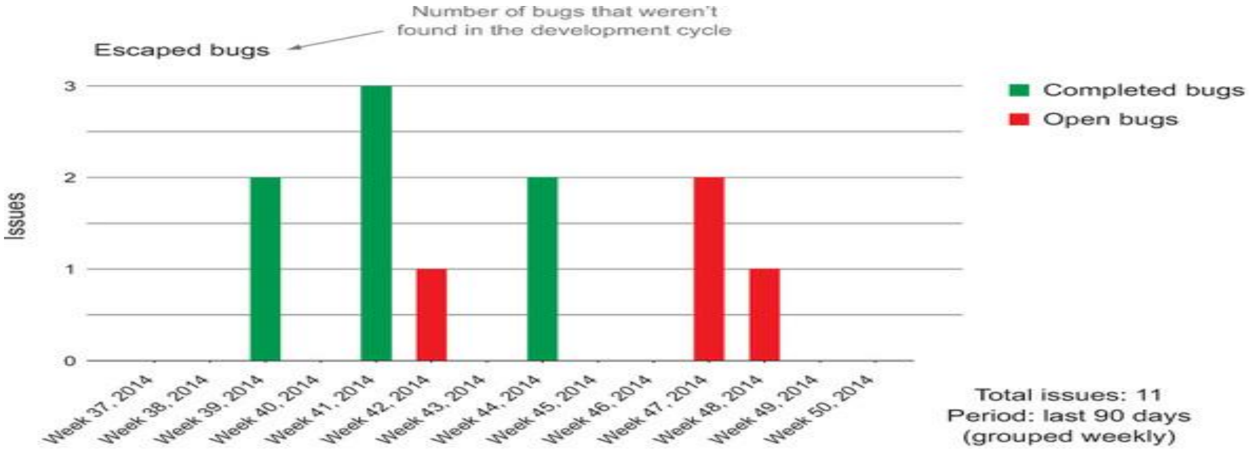
1. An erratic average velocity over a period of time requires revisiting the team’s estimation practices.

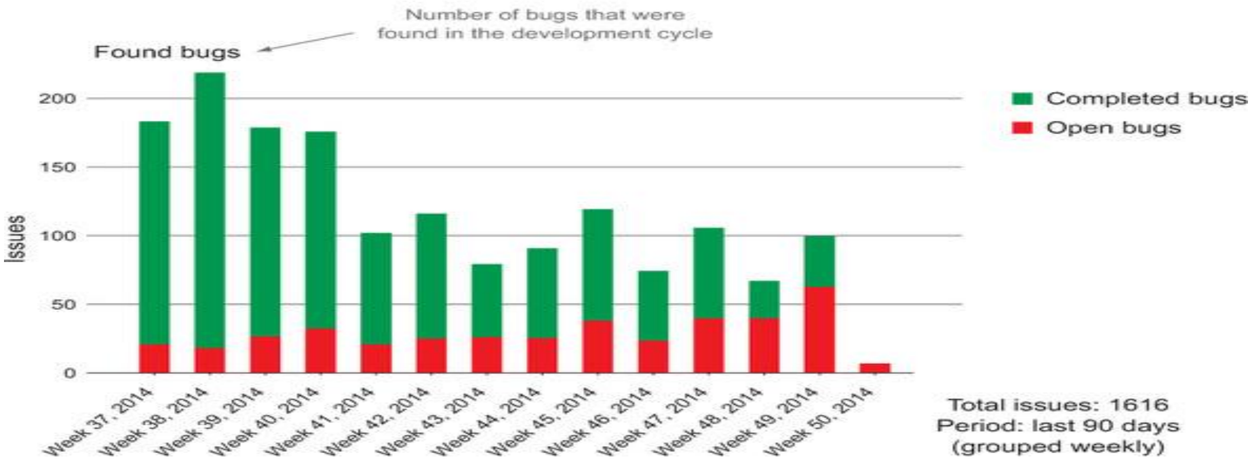
2. Are there unforeseen challenges not accounted for when estimating the work

DONOT

- Use velocity to compare two different teams since the level of work estimation is different from team to team
- Use velocity to identify lower performing teams.

Bug counts





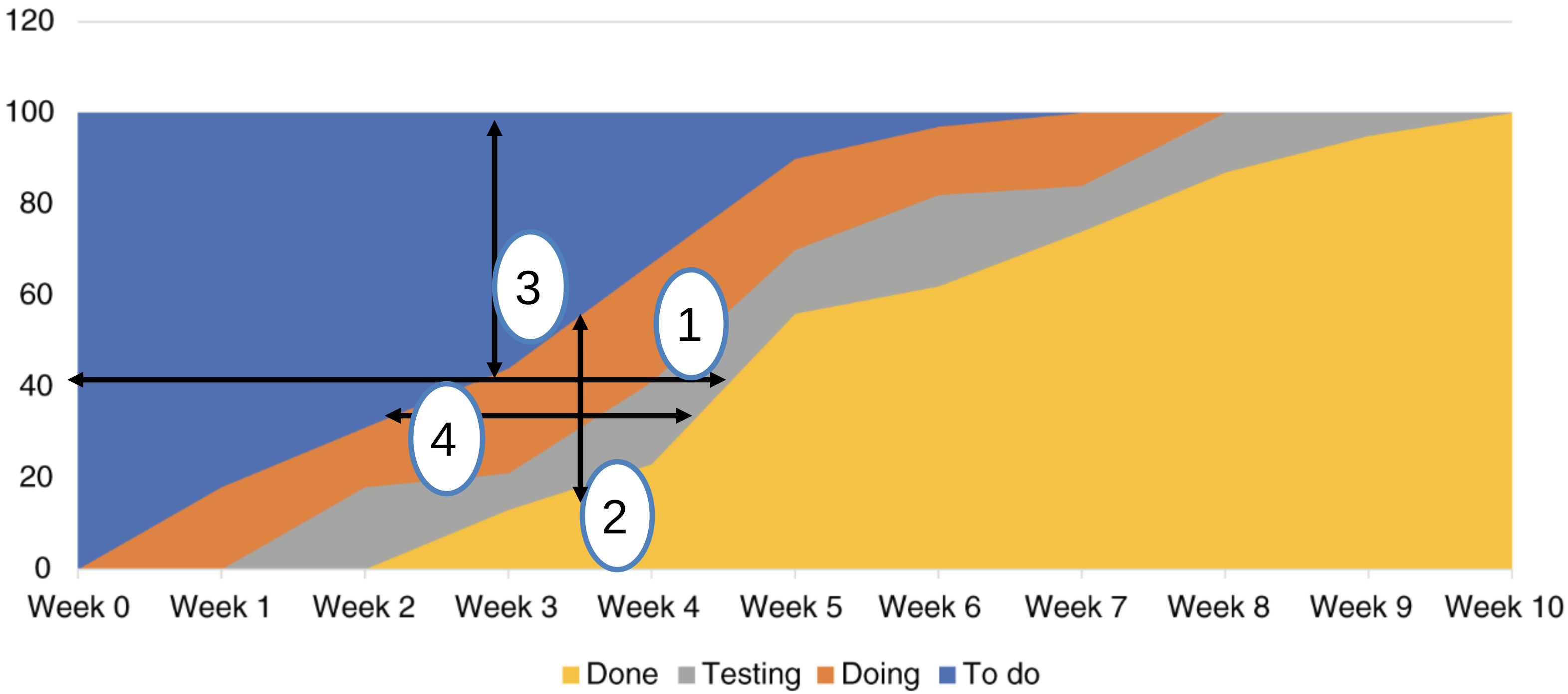
Source: Agile Metrics in Action: How to measure and improve team performance by Christopher W. H. Davis , Published by Manning Publications, 2015

Summary

- Metrics never convey the whole picture. Management by metrics and dashboards needs to be supplemented with management **by context and conversations**.
- Stop measurements that lead to counterproductive behavior and stop at measurements (i.e., don’t continue to targets) that lead to desired behavior.
- Prefer outcome-oriented metrics to activity-oriented ones. Prefer aggregate metrics to fine-grained ones.
- Get comfortable with lagging (or trailing) indicators. When fast feedback is available, lagging indicators are a reliable alternative to speculative leading indicators.

Quiz

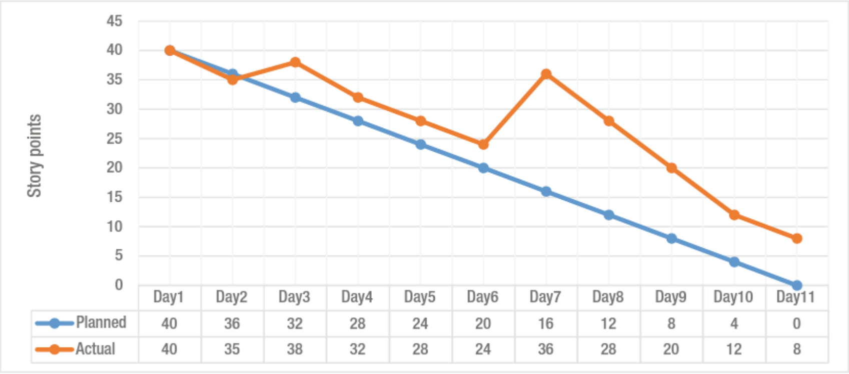
Cumulative flow diagram
Q1: What to Watch for?



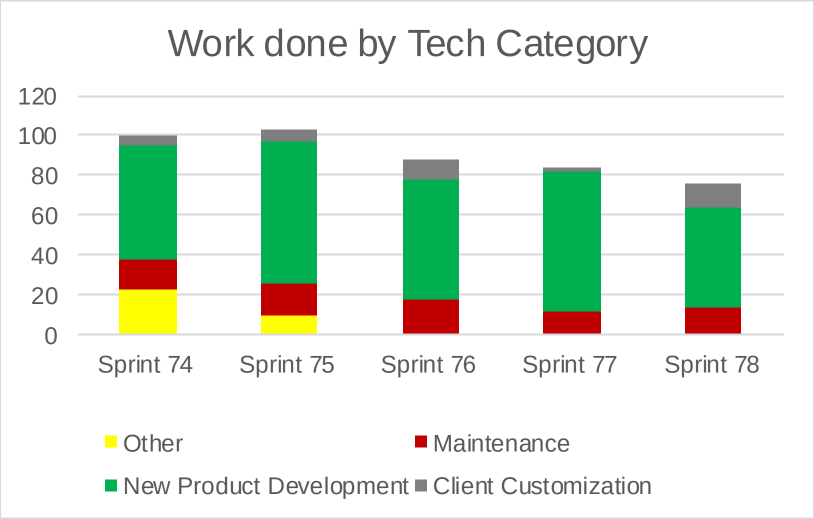
1- Lead time , 2.Work in Progress, 3. Backlog (week3), 4. Cycle time

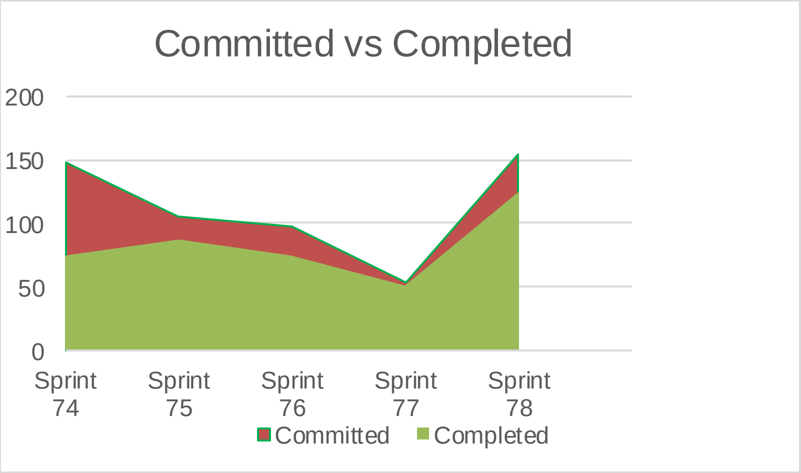
Project Progress:
Q1: What to Watch for?

Burndown chart



	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11
Planned	40	36	32	28	24	20	16	12	8	4	0
Actual	40	35	38	32	28	24	36	28	20	12	8



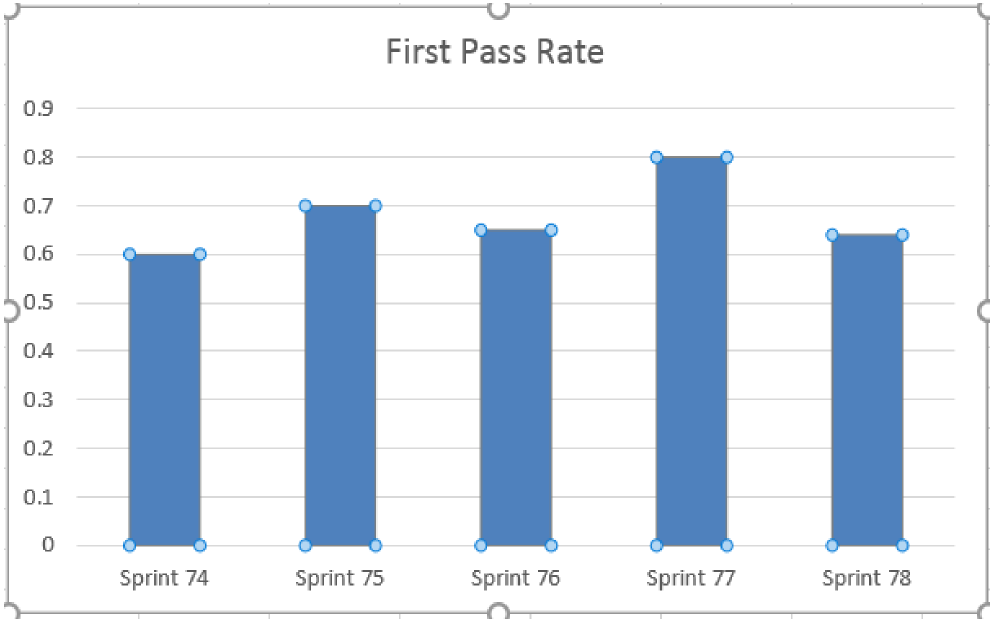


Q2

Quality of Code

- First Pass Rate
  - Used for measuring the amount of rework in the process
  - Defined as number of test cases passed on first execution.

    FPR = Passed\Total on First Execution
  - For stories that deal with the development of new APIs or Features
  - For stories that deal with addendums to APIs or Features FPR should include regression.



How do you connect this measure to production defects?

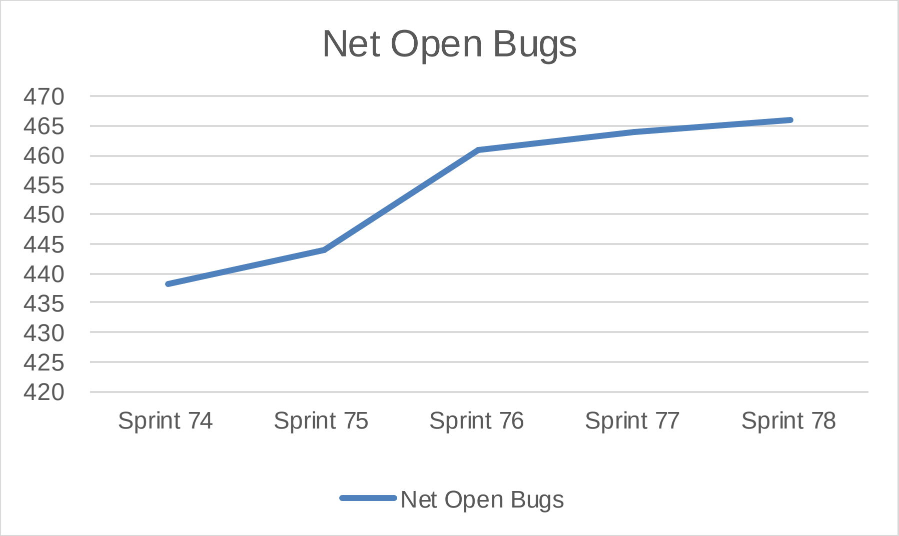
What To Watch For

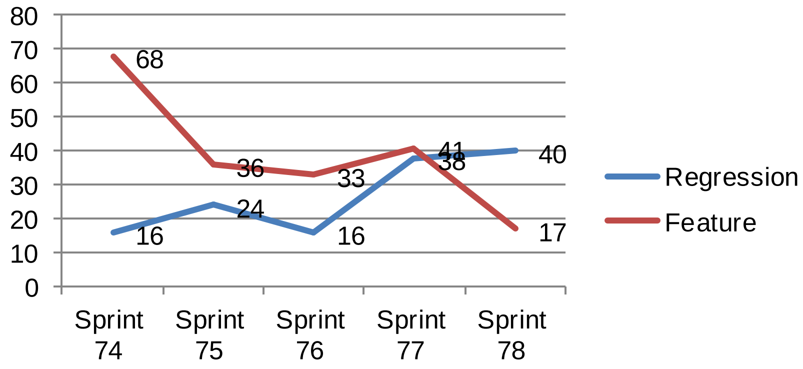
- Lower first pass rates indicate that Agile tools like desks checks , unit testing are not used sufficiently.
- Lower first pass rate could indicate lack of understanding of requirements.
- Higher first pass rate combined with high defect rate in production could indicate lack of proper QA.

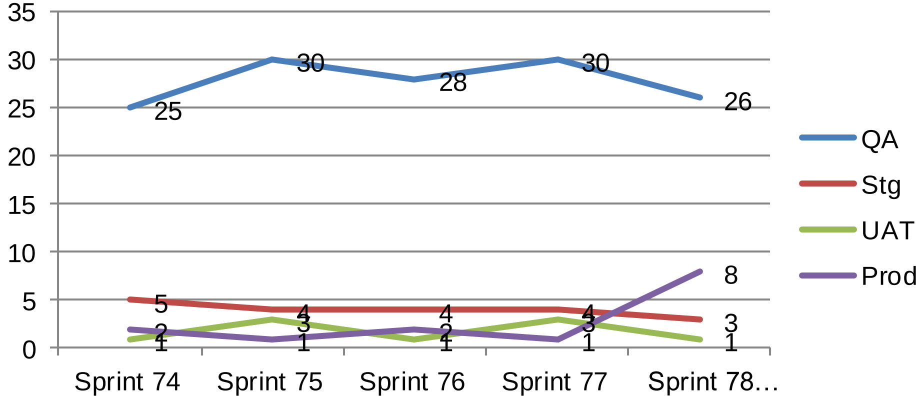
Q3

Bug Dashboard

- **Net Open Bugs / Created vs Resolved**
  - This gives a view of the team flow rate. Are we creating more technical debt and defects than what the team can resolve.
- **Functional Vs Regression Bugs Trend**
  - This helps identify the defects found in new development vs regression.
- **Defects Detected in**
  - This helps identify the environment in which the defect is detected. (QA, Staging, UAT, Production)
  - For defects detected in environment higher than QA, an RCA is needed.







What To Watch For

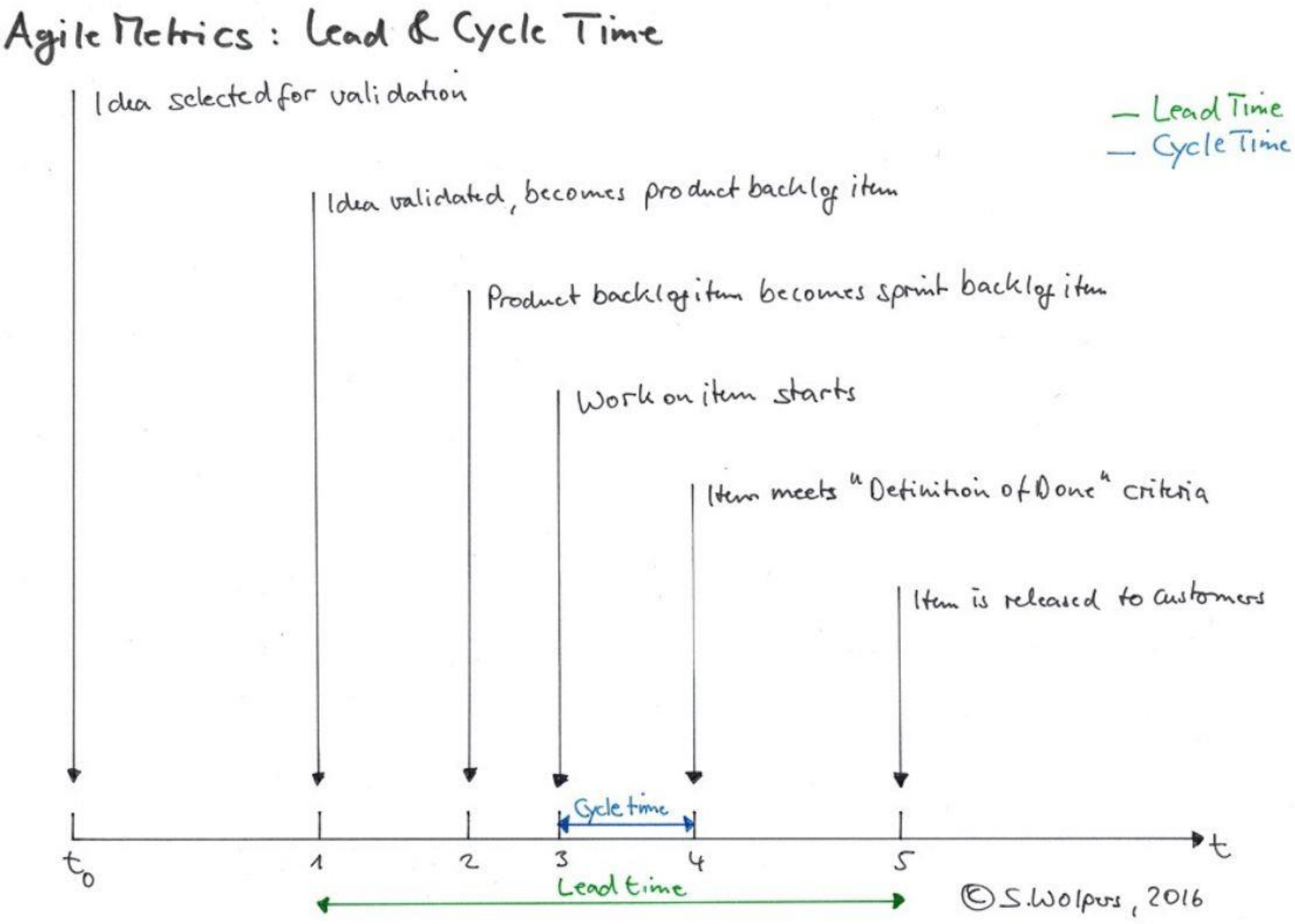
- An increase in regression bug count indicates the impact of code refactoring.
- An increase bug count in non-QA environment due to environment differences requires revisiting the environment strategy.
- An increase bug count in non-QA environment due to QA oversight requires revisiting the testing strategy.

BITS Pilani

Pilani Campus

Additional Notes - Measuring Agile Performance

Good Quantitative Agile Metrics: Lead Time and Cycle Time


Agile Metrics : Lead & Cycle Time
Idea selected for validation
— Lead Time
— Cycle Time
Idea validated, becomes product backlog item
Product backlog item becomes sprint backlog item
Work on item starts
Item meets "Definition of Done" criteria
Item is released to customers
Cycle time
t_0
1
2
3
4
5
t
Lead time
©S.Wolpers, 2016

Important to measure the things that drive/determine Lead Times. Levers that teams can actively (e.g. determinant metrics like Flow Efficiency).

Most common and meaningful metrics for Team System Improvement(SI)

Simple SI metrics	Metric	Comment
Overall SI goal metrics	Lead Time, Cycle Time	Good measures of overall Time to Value
Determinant metrics:		
Speeding Ticket	% Speeding Tickets	Tickets that have been moved through multiple Statuses (e.g. in Jira) after the event (so there is no real visibility of workflow stages)
Timing accuracy	Total Sprint Completion (%)	Percentage of completed story points for a given sprint(s). The factor takes into account story points added once a sprint has started.

Source:https://www.infoq.com/articles/metrics-agile-teams/

Most common and meaningful metrics for Team SI …

Simple SI metrics	Metric	Comment
Productivity	Flow efficiency (%)	Percentage of time spent active versus inactive within a workflow
	Return rate (%)	Percentage of tickets returned from QA (for whatever reason) during the dev process. This generates Rework.
	% Time bug fixing (**The ratio of fixing work to feature work.)**	Percentage of time a team spends bug fixing versus feature contribution.
	Number of defects escaping to production.	This is category of fixing the work.

Source:https://www.infoq.com/articles/metrics-agile-teams/

Most common and meaningful metrics for Team SI …

Simple SI metrics	Metric	Comment
Team Wellness	Team Happiness Team Sprint Effectiveness Rating	Self Assessment tests: Individual engineers polled each Sprint/cycle.

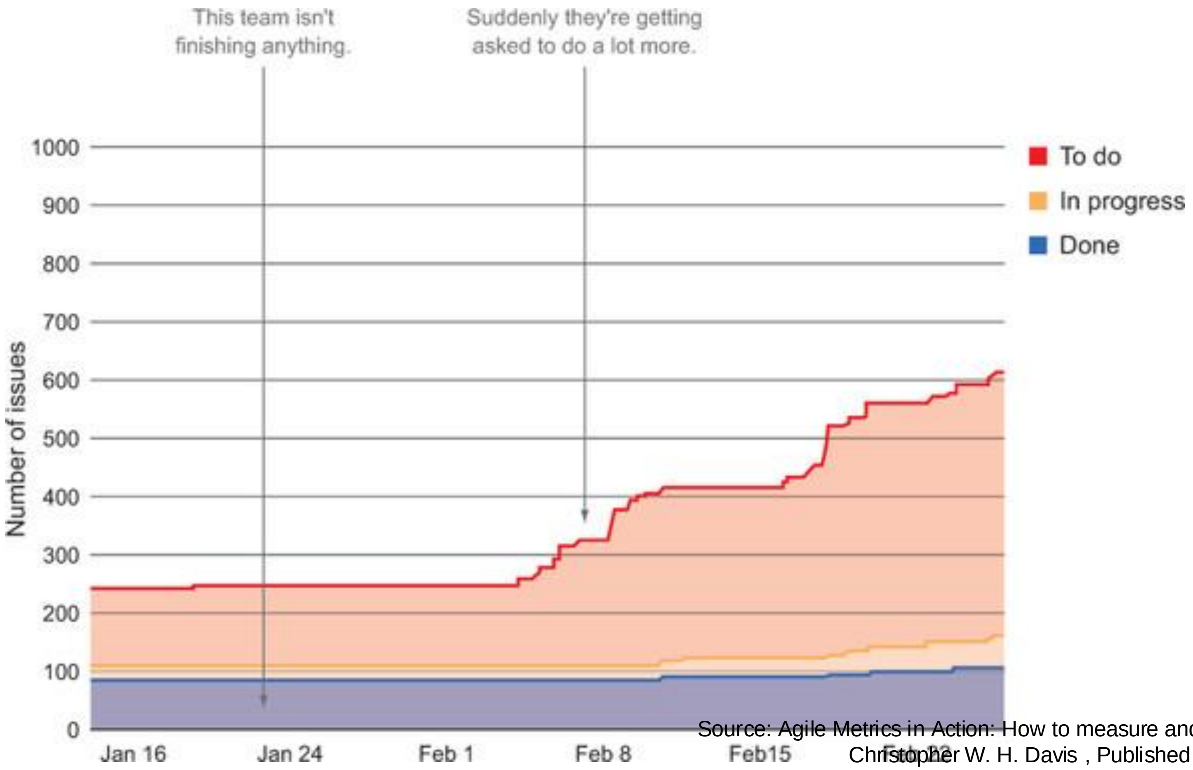
Source:https://www.infoq.com/articles/metrics-agile-teams/

The Importance of Metrics to Agile Teams

- The philosophy of Continuous Improvement (CI) is central to Agile.
- CI- should not be imposed and driven top-down – instead it should be led by Agile teams themselves, so Self-Improvement (SI) is a more suitable terminology.
- SI is hard requires organization leadership long term support, recognition and suitable framework. Crucially,
  - **A set of meaningful and agreed Agile metrics to** track performance improvement over time; **and a means to surface these metrics in near real time**, with minimum/no effort involved for the teams themselves.
  - Keep metrics **simple and deterministic (no ambiguity).**
  - For each of these metrics, it is the *trend* that is important, not an absolute number. The trend will tell you if your attempts at improvement are having an effect.

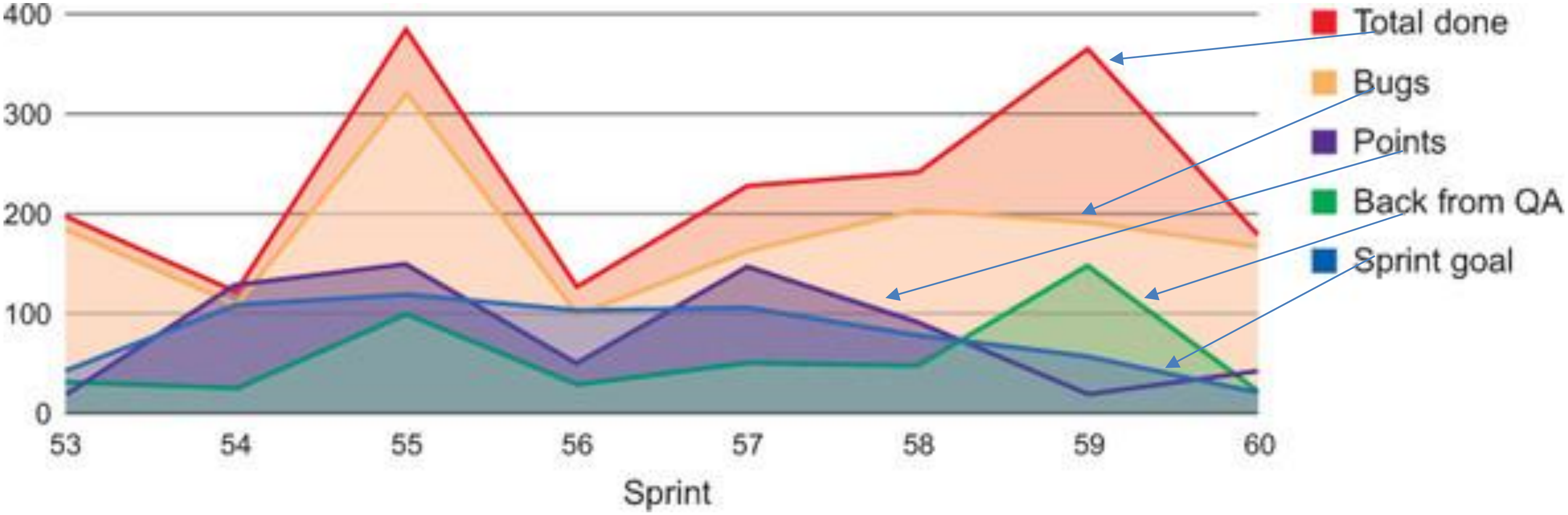
Cumulative Flow

- An example cumulative flow diagram showing a team getting asked to do a lot more than they have been able to accomplish historically



Source: Agile Metrics in Action: How to measure and improve team performance by Christopher W. H. Davis , Published by Manning Publications, 2015

Example - Combination of data

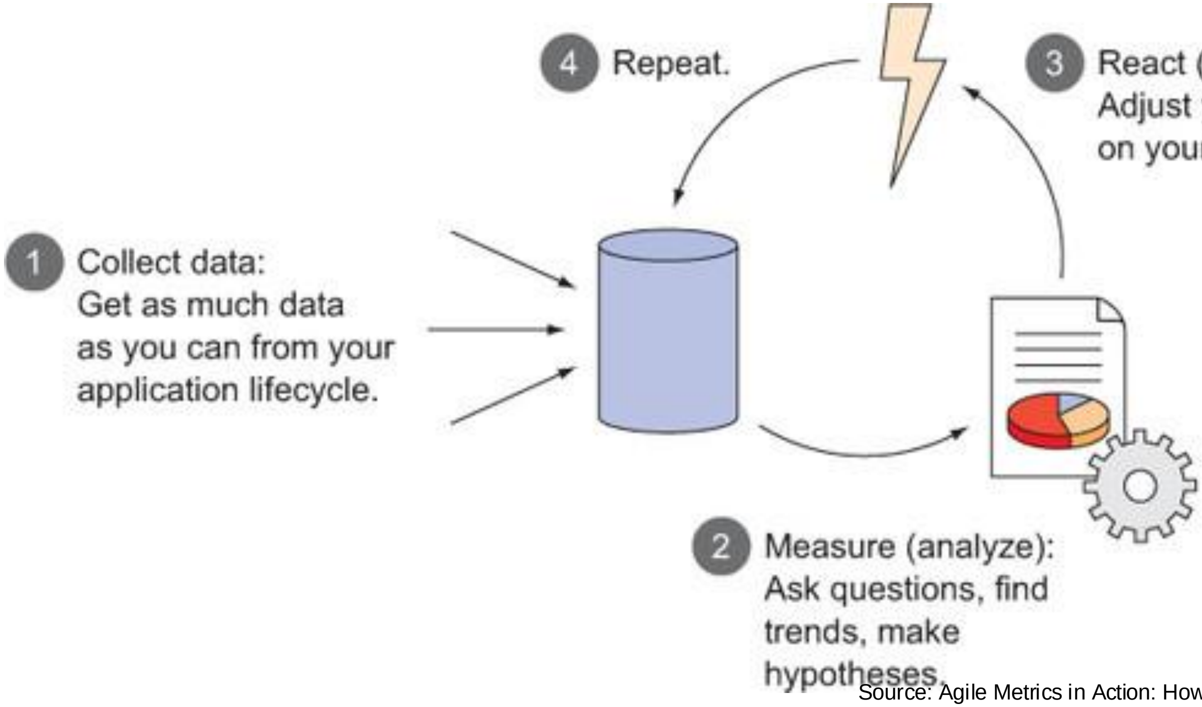


- **Spikes in this data point can indicate potential problems:**
- There’s a communication gap somewhere on the team.
- Completion criteria (a.k.a. done) are not defined clearly to everyone on the team.
- Tasks are being rushed, usually due to pressure to hit a release date.

Source: Agile Metrics in Action: How to measure and improve team performance by Christopher W. H. Davis , Published by Manning Publications, 2015

COLLECT, MEASURE, REACT, REPEAT—THE FEEDBACK LOOP

- There isn’t a silver-bullet metric that will tell you if your agile teams are performing as well as they can.
- Collecting and analyzing data in the form of metrics is an objective way to learn more about your team and a way to measure any adjustments you decide to make to your team’s behavior.



1 Collect data: Get as much data as you can from your application lifecycle.

2 Measure (analyze): Ask questions, find trends, make hypotheses.

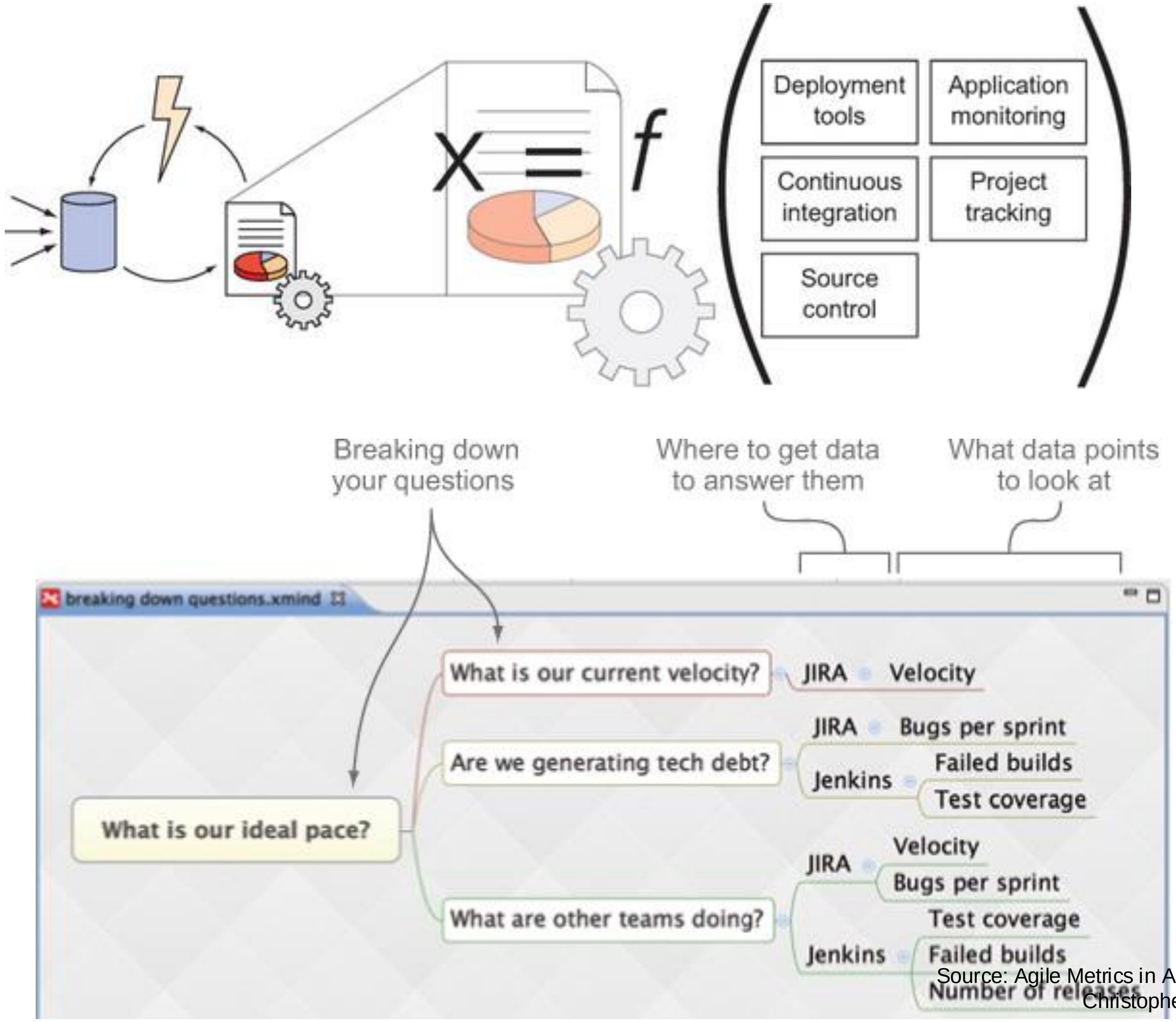
3 React (apply): Adjust your team based on your findings.

4 Repeat.

Source: Agile Metrics in Action: How to measure and improve team performance by Christopher W. H. Davis , Published by Manning Publications, 2015

Figuring out what matters

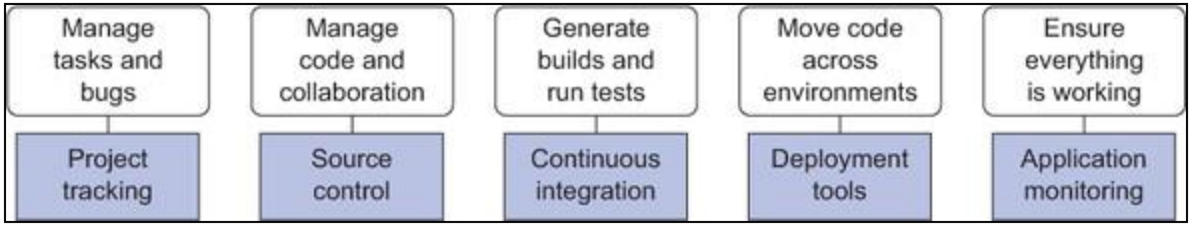
(X is what you want to answer; some combination of your data can get you there.)



- Mind mapping is a brainstorming technique where you start with an idea and then keep deconstructing it until it's broken down into small elements. XMind (www.xmind.net/),

Source: Agile Metrics in Action: How to measure and improve team performance by Christopher W. H. Davis , Published by Manning Publications, 2015

Project performance data



Manage tasks and bugs – Project tracking

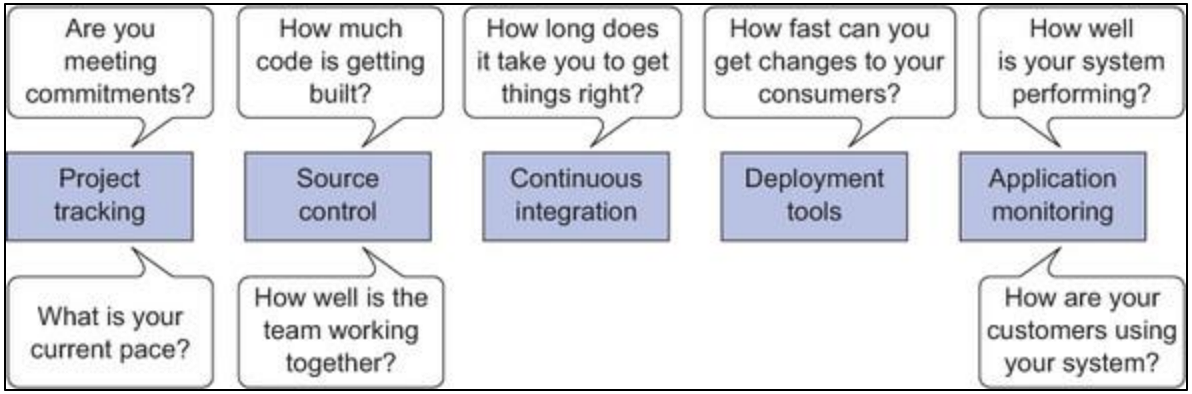
Manage code and collaboration – Source control

Generate builds and run tests – Continuous integration

Move code across environments – Deployment tools

Ensure everything is working – Application monitoring

Data is all over the place without a unified view



Are you meeting commitments? – Project tracking – What is your current pace?

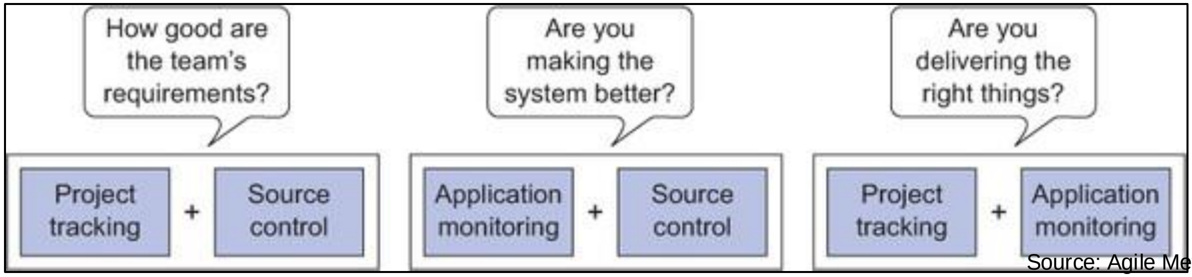
How much code is getting built? – Source control – How well is the team working together?

How long does it take you to get things right? – Continuous integration

How fast can you get changes to your consumers? – Deployment tools

How well is your system performing? – Application monitoring – How are your customers using your system?

Questions you can answer with data from systems in your SDLC.



How good are the team's requirements? – Project tracking + Source control

Are you making the system better? – Application monitoring + Source control

Are you delivering the right things? – Project tracking + Application monitoring

Adding data together to answer high-level questions

Source: Agile Metrics in Action: How to measure and improve team performance by Christopher W. H. Davis , Published by Manning Publications, 2015

innovate achieve lead

BITS Pilani
Pilani Campus

Thank you