



Agile University: Delivery School

Agile Metrics



High performance. Delivered.

Strategy | Digital | Technology | Operations

Agenda

Agile Philosophy of Metrics

Key Agile Metrics

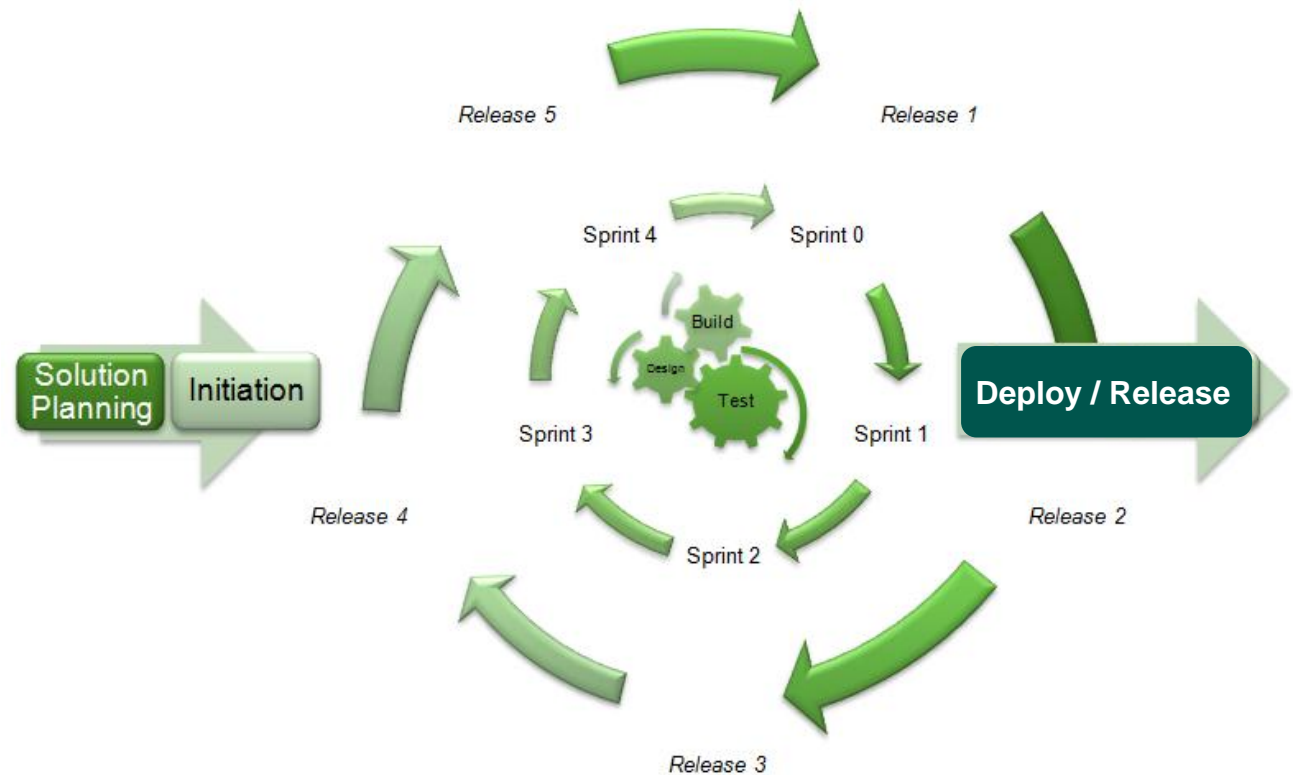
Agile Philosophy of Metrics

- **Minimum number** of metrics that will provide all information necessary to meet business goals
- Measure **outcomes**, not outputs
- Measure **results**, not activity
- Measure work items **completed**, not time spent per task
- Assess **trends**, not snapshots
- Provide feedback on **frequent** and **regular** intervals
- Metrics should be **easy to collect**



Metrics in Agile at Accenture

- Combination of Managed Delivery and Agile Metrics
- Captured at: **Project**, **Release** and **Sprint** level



Agenda

Agile Philosophy of Metrics

Key Agile Metrics

Velocity

Description It is the amount of features “done” in a sprint by a team

Unit Story Points

Formula Sum of story points of user stories completed (aka ‘Done’) in the sprint

Exclude any Defect Stories that are considered Technical Debt (defects part of features being developed)

Example

User Story Title	Estimate (Story Points)	Priority Category	“Done” ?	“Done” in Sprint?	Planned for Sprint
Search Video by Popularity	2	Must-Have	✓	Sprint 1	
Add to Shopping Cart	2	Must-Have	✓	Sprint 2	
View Shopping Cart	2	Must-Have	✓	Sprint 2	
Search Book by Author	3	Must-Have	✓	Sprint 3	
Remove from Shopping Cart	2	Must-Have	✓	Sprint 3	
User Registration	3	Must-Have			Sprint 3
Register Billing Address	3	Must-Have			Sprint 4
Bill using Visa Card	8	Must-Have			Sprint 5
Bill using Master Card	5	Could-Have			Sprint 6

Team Velocity is the velocity of last completed Sprint

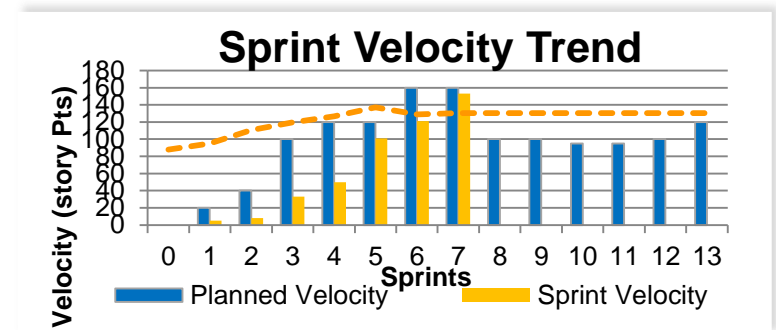
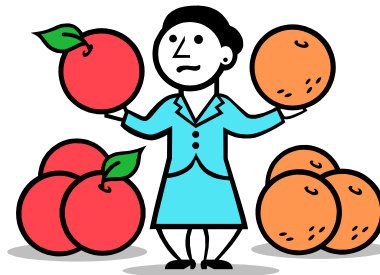
For example, in this case, **Sprint 3** is the last completed Sprint

Team Velocity
= Sum of estimate of ‘Done’ stories
= 3+ 2 = 5

Comparing Velocity

My Team's Velocity Vs. Your Team's Velocity

As velocity is based on story points, what constitutes one story point for one team or project, would not be the same as what it is for another team or project. So, velocity cannot be used to compare productivity of different teams or projects.



Velocity Trends

However, Velocity can be very useful for a team to compare itself at different points in time

See the Velocity Trend chart shown on the right, comparing the Planned, Actual & Required Velocity

Required Velocity

Description This is the average velocity that the team would require in the remaining sprints, to complete at least the remaining “Must-Have” category user stories, so that a Minimal Marketable Product can be deployed to Production.

Unit Story Points

Formula Sum of Estimate of Remaining “Must-Have” stories / No. of Sprints Remaining



User Story Title	Estimate (Story Points)	Priority Category	“Done” ?	“Done” in Sprint?	Planned for Sprint
Search Video by Popularity	2	Must-Have	✓	Sprint 1	
Add to Shopping Cart	2	Must-Have	✓	Sprint 2	
View Shopping Cart	2	Must-Have	✓	Sprint 2	
Search Book by Author	3	Must-Have	✓	Sprint 3	
Remove from Shopping Cart	2	Must-Have	✓	Sprint 3	
User Registration	3	Must-Have	✗		Sprint 4
Register Billing Address	3	Must-Have	✗		Sprint 5
Bill using Visa Card	8	Must-Have	✗		Sprint 6
Bill using Master Card	5	Could-Have	✗		Sprint 7
Search Book by Title	3	Must-Have	✗		Sprint 7

In this example,

There are 4 Remaining “Must-Have” priority category stories of sizes 3, 3, 8 and 3 story points

Of 7 sprints in the release, 3 are complete and 4 sprints remaining

Required Velocity = $(3+3+8+3)/4 = 4.25$

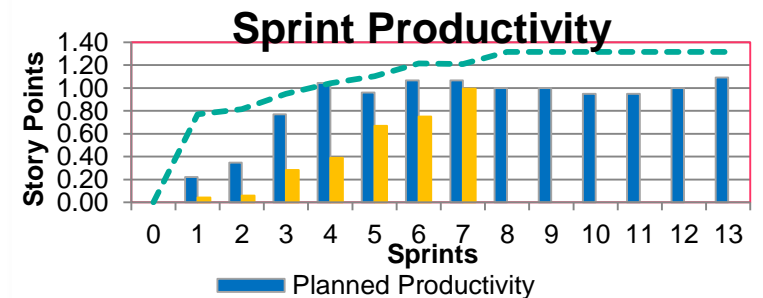
Productivity - Velocity Based

Description	It is the average amount of work “done” in one day of a sprint by one team member
Unit	Story Points Per Person Day
Formula	Velocity of Sprint / Actual Effort spent on Sprint
When Useful	<p>Often, comparing velocity only across sprints may not provide an accurate measure of team’s improvement in productivity. Reason:</p> <ul style="list-style-type: none"> ➤ Change in team size between sprints: For example, if the team size in Sprint 5 was 5 team members, but Sprint 7 had 8 team members ➤ Vacations/Holidays in the sprint: For example, a 2 week Sprint with 2 holidays <p>This metric can also be used to check if the team’s current productivity is sufficient to produce a Minimal Marketable Product (MMP) within the remaining sprints. Release Slippage check: Is Current Productivity > = Required Productivity?</p>



Other Related Productivity Metrics

Planned Productivity	= Planned Velocity / Planned Effort in Sprint
Required Productivity	= Sum of Estimate of Remaining “Must-Have” stories / Total Planned Effort in Remaining Sprints



Sprint Burn-Down Chart



Description

- ✓ This chart is used to track the progress of the on-going Sprint.
- ✓ It tracks the work that remains to be completed in the sprint backlog.
- ✓ It is used to analyze the variance on any day with respect to an ideal Burn-down.
- ✓ A forecast trend-line can be used to predict the likely variance at sprint completion and take appropriate actions during the course of the sprint.

Frequency

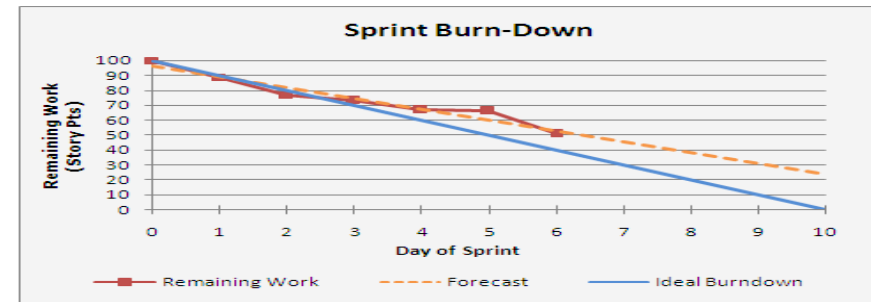
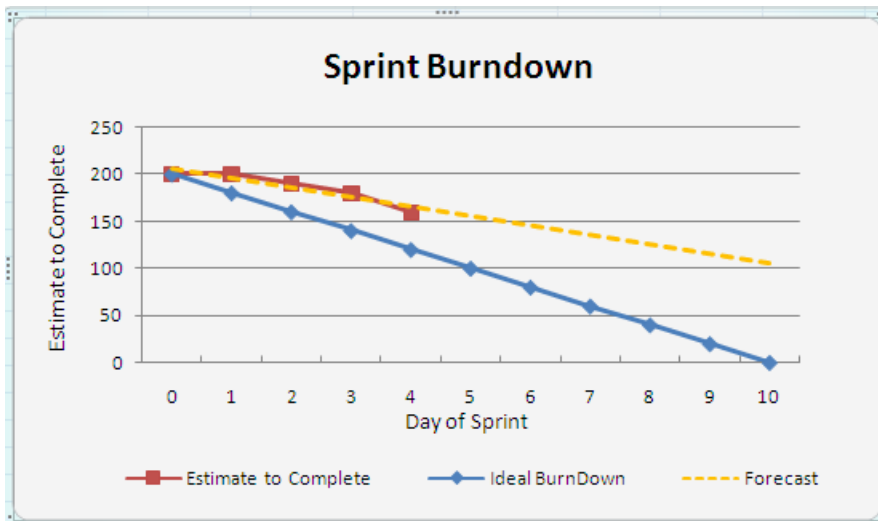
Updated & Used Daily

Alternate Usage

The X-axis is used to track the day of the Sprint.

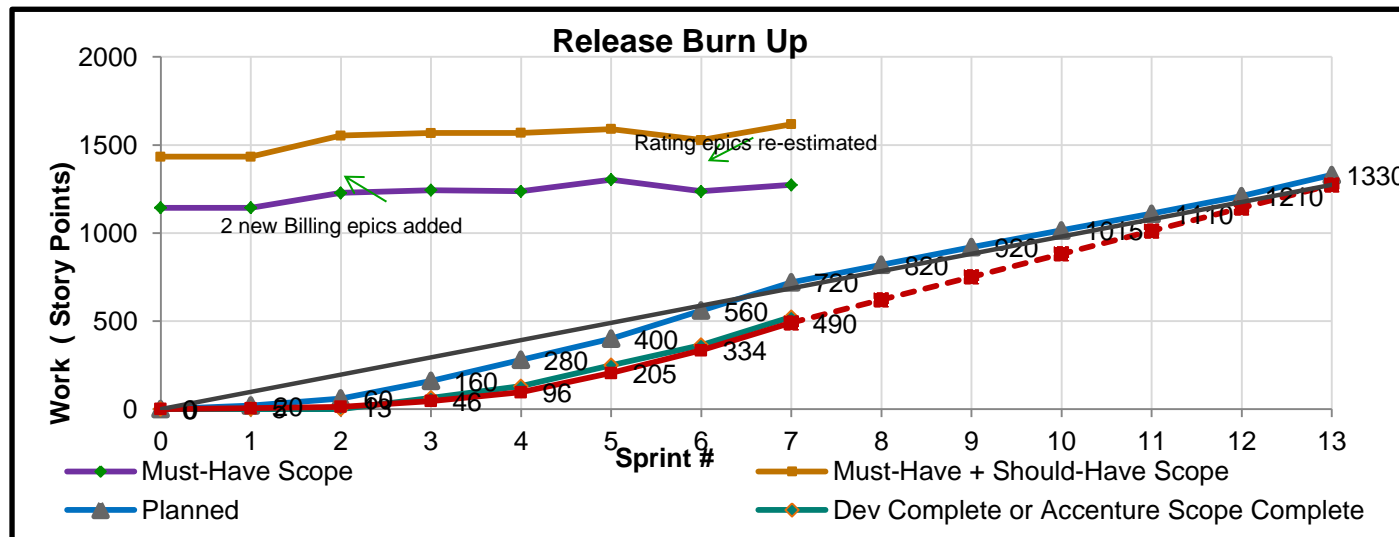
The Y-axis measures the Remaining work.

- (i) Remaining work is often calculated in terms of hours of work remaining. Each day, the sum of Remaining effort of all Sprint tasks is plotted on the chart.
- (ii) The number of story points remaining to be completed in the sprint (as per sprint commitment) can also be used alternatively to measure the Remaining work.

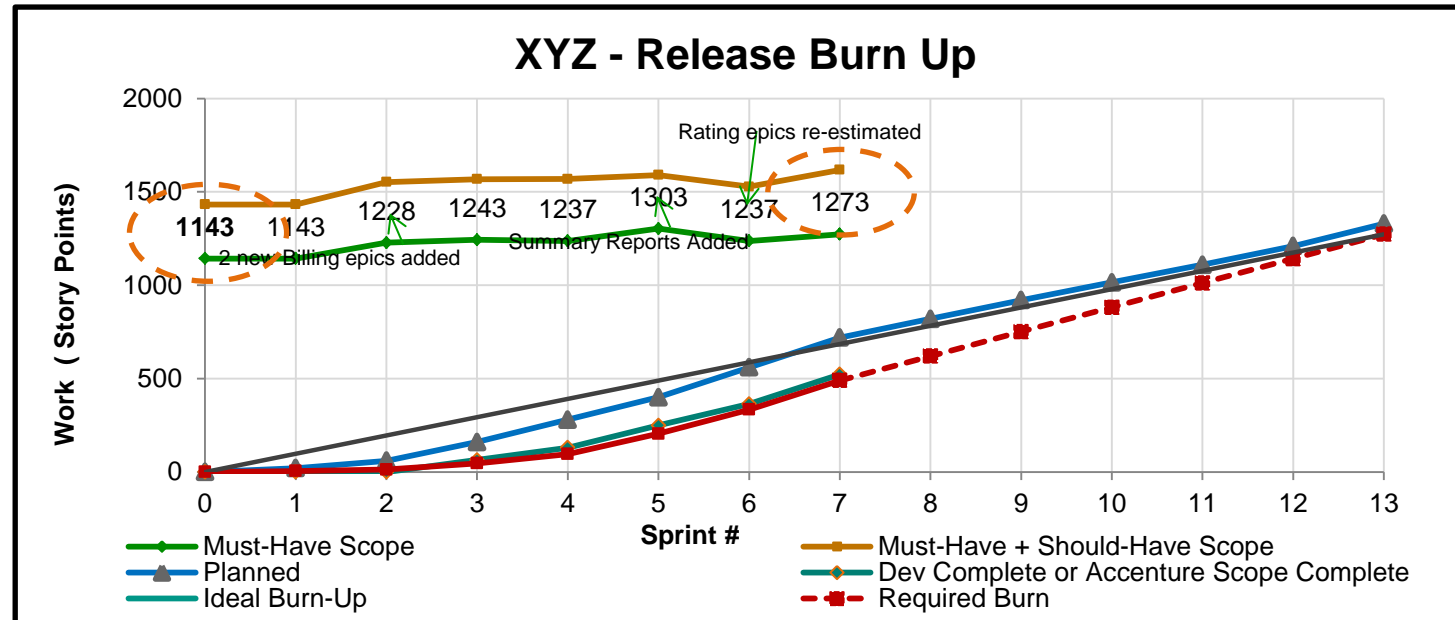


Release Burn-Up Chart

Description	<ul style="list-style-type: none"> ✓ This chart is used to track the progress of the Release. ✓ This depicts the amount of work “done” so far in the release (cumulative velocity). ✓ It compares variance against the ‘Ideal’ and ‘Planned’ release burn-up trend lines. ✓ Additionally, the same chart can be used to track release scope volatility
Frequency	At the end of Every Sprint of the Release, this chart is updated and analyzed
Y Axis	<p>“Must-Have Scope” line: Plot the Total Story Points of all Must-Have stories in Release Backlog</p> <p>“Planned” line: Plot the total Story Points of work planned to be completed by that Sprint for the Release</p> <p>“Test Complete or Done” line: Plot the <u>Cumulative</u> Story Points of stories completed (ie ‘Done’), for the Release, at the end of that Sprint</p>



Release Burn-Up Chart - Tracking Scope Volatility



Scope
Volatility
(for Must-
Have's)

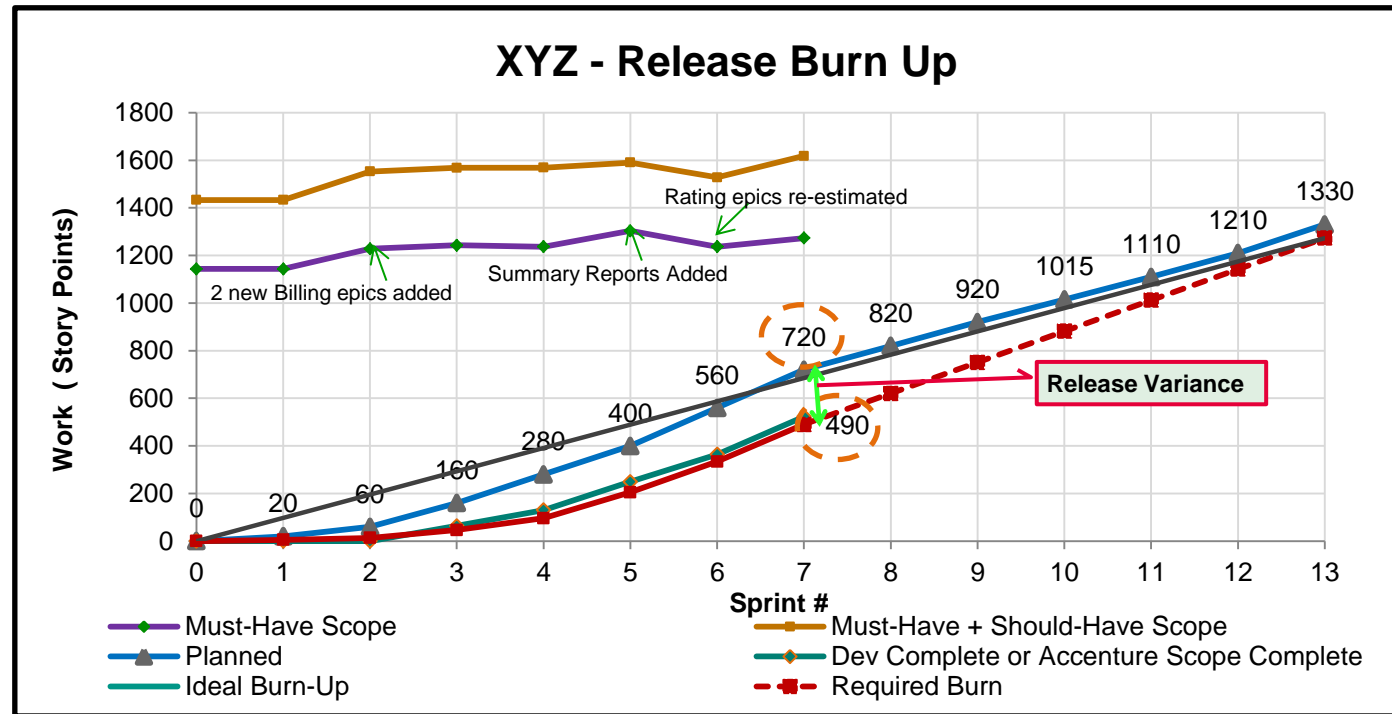
Depicts change in scope size for the Must-Have scope of release

= (Current Size of Must-Have – Original Size of Must-Have) / Original
Size of Must-Have * 100 %

In the example above,

Scope Volatility = $(1273 - 1143) / 1143 * 100 = 11\%$

Release Burn-Up Chart - Tracking Scope Variance



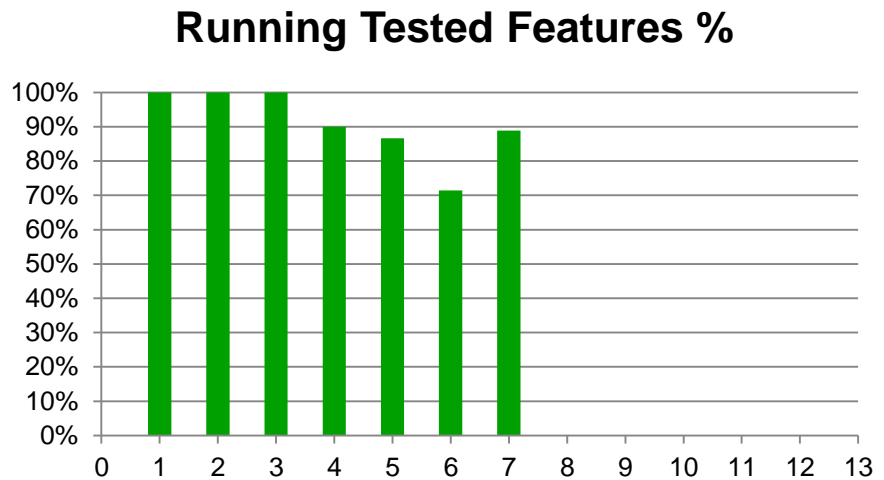
Release Variance

- Depicts % variance in planned work to completed work, for on-going release

$$= \frac{(\text{Size of work planned to be completed so far} - \text{Size of work completed so far})}{\text{Size of work planned to be completed so far}} \times 100$$
- In the example above, at the end of Sprint 7, Release Variance = $(720 - 490) / 720 \times 100\% = 47\%$
- There are NO specific limit such as +/- 10% & must always be analyzed with this trend chart.
- The variance could be higher (even a 50%) during initial 3 sprints, but must be very carefully watched from mid-release onwards. Actions may include negotiations towards removal of non-Must Have scope for release, addition of resources or another iteration etc.

Running Tested Features

- Indicates the % of working software over software built so far
- Running Tested Features =
$$\frac{\text{No. of 'Done' stories that passed last regression test}}{\text{Total no. of 'Done' user stories}} \times 100 \%$$
- Used to check how much of software built in previous sprints are still working, or if the team is breaking old code while building new code.
- If a story/feature/product backlog item is not tested in the latest regression test, then it cannot be counted in the numerator.
- Frequency: Sprint End



# Features built	90
# Features tested working in last regression	80

Questions & Answers

