The following is a user guide for navigating the PLT Metrics Dashboard. It includes the definition and criteria for each metric collected in the PLT Metrics Summary, PLT Flow Predictability and PLT Flow Efficiency Metrics Dashboard.

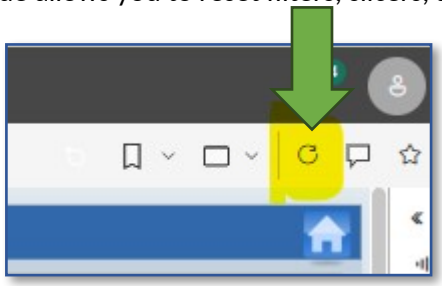
Important Note: All numbers and visuals on this dashboard display maximum of up-to 13 months of data. Data for the previous month will be visible on the dashboard within first week of the current month. This is done to ensure iteration releases are accurately closed out & validated by product teams.

Report Tabs:

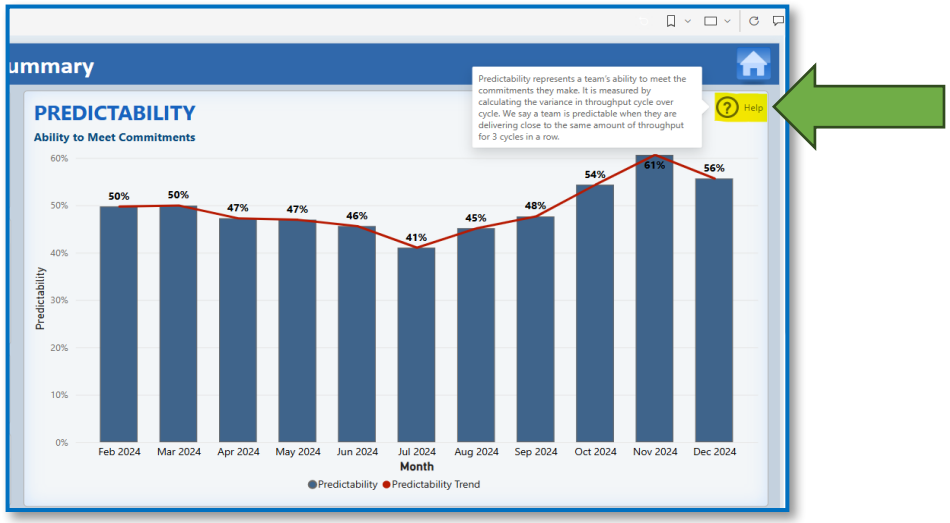
Report Tab	Description
PLT Metrics Summary	This section provides a high-level view of key metrics across all Product Line Teams (PLTs). This section aggregates data across Product divisions to present an executive summary of critical metrics, allowing leadership and stakeholders to monitor the trends. This helps provide a bird's-eye view of overall Agile maturity across all Product teams. It helps identify broad trends in delivery capability, efficiency, and quality.
PLT Flow Predictability	Flow predictability measures the consistency and reliability of a product team's delivery maturity over a specified period. It assesses how well the product teams meet planned delivery expectations by analyzing key indicators such as feature throughput, effort variance, and backlog readiness. A predictable flow ensures teams can anticipate delivery timelines, manage stakeholder expectations, and align work execution with business priorities. By tracking metrics like feature throughput (volume and effort), throughput variance, and feature readiness backlog, teams gain insights into delivery stability, process bottlenecks, and areas requiring adjustment. Strong flow predictability reduces delivery risk, enhances planning accuracy, and fosters continuous improvement in MTA IT's Agile at scale execution.
PLT Flow Efficiency	Flow efficiency measures how effectively a product team translates planned work into completed deliverables, ensuring optimal use of resources and minimizing delays. It showcases the alignment between planned and delivered work - tracking key indicators such as feature completion ratios (count and effort), release success rates, batch size consistency, and release frequency. A high Flow Efficiency indicates optimized workflows and improved throughput, while a lower efficiency signals bottlenecks, resource misalignment, or excessive work-in-progress (WIP). By continuously monitoring these metrics, teams can enhance predictability, improve Agile execution, and maximize the value delivered to our business stakeholders.
User Guide	Displays the User Guidelines/Knowledge Base Article
Release Notes	Displays the Release notes

Data Filtering: The report data on the dashboard can be filtered by different data points by clicking on a graphic (such as the year "2025" of the "Metrics Collection Dates" visual) or by using the dedicated filters, e.g., Product Category Team (PCT), Product Line Team (PLT) below/next to the Date filter. Multiple filters can be layered at once.

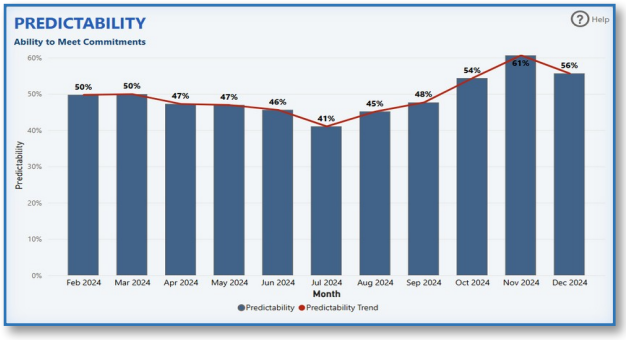
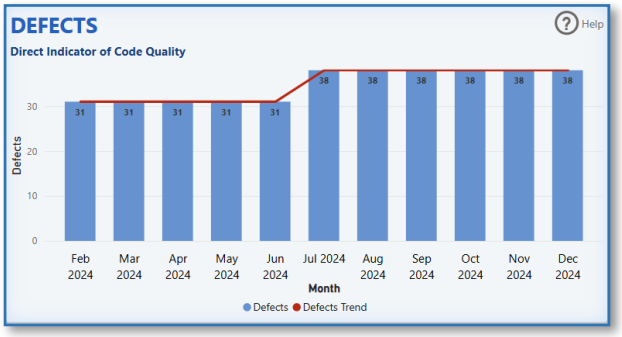
Important Note: The reset button on the upper right-hand side allows you to reset filters, slicers, and other data view changes.



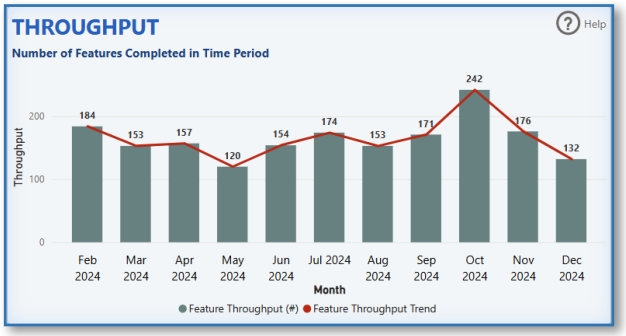
Help Text: Each metric has a help icon in the upper right-hand corner. Hovering over the icon "?" displays a pop-up with the definition of the specific metric. See below.



Metrics Descriptions/Formulas and Other Details:

Predictability 	**Metrics Description:** **Predictability** represents a team's ability to meet its commitments. It is measured by calculating the variance in throughput cycle over cycle. We say a team is predictable when they consistently deliver close to the same estimated effort for three consecutive months. **Formula Description:** Inverse of Feature Throughput Variance (Effort)
Defects 	**Metrics Description:** **Defects** are a direct indicator of the quality of the code. We measure it by counting the number of defects in the open state in a specified period. A reduction in defects over time represents an improvement in quality. **Formula Description:** Count of Defects Active in a Month

Throughput



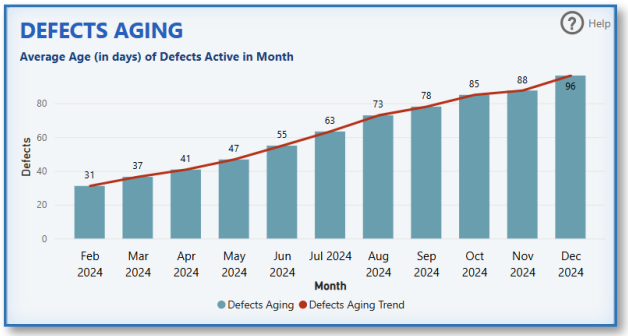
Metrics Description:

Throughput measures the amount of work completed in a period. For Product Line Teams, this means Features. We measure throughput to understand how much work can be delivered in a given time period based on actual data. We then use that information when creating our Product roadmaps to create a realistic plan for when work will be delivered.

Formula Description:

Sum of Delivered Features Per Period

Defect Aging



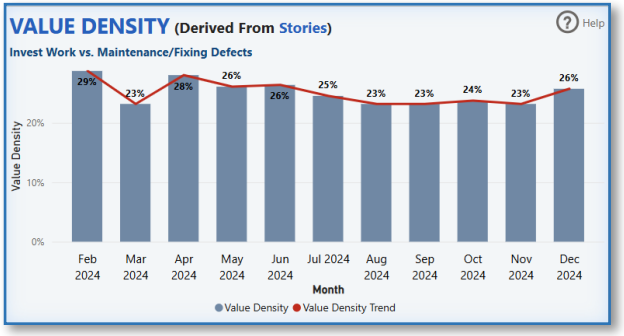
Metrics Description:

The **Defect Aging** average is a good proxy for understanding the team's ability to resolve issues. The goal is to prevent the value from growing as it accumulates quality waste in the system that reduces the team's capacity to complete other valuable work.

Formula Description:

Average of Active Bugs Age (Days)

Value Density



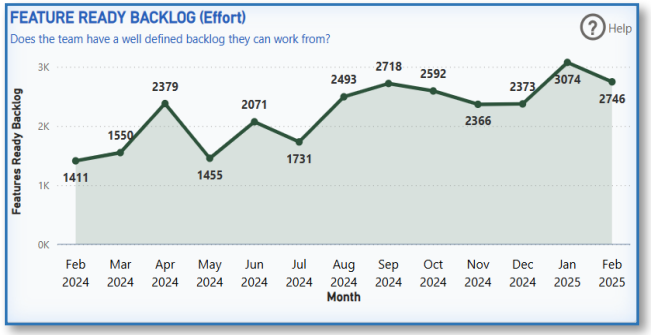
Metrics Description:

Value Density highlights how much of a team's work is invest work and how much of it is maintenance and fixing defects. We use value density to guide decisions about priorities and ensure the team is spending its time working on items in line with the business priorities.

Formula Description:

The Invest Work vs. Maintenance/Fixing Defects

Feature Ready Backlog (Effort)



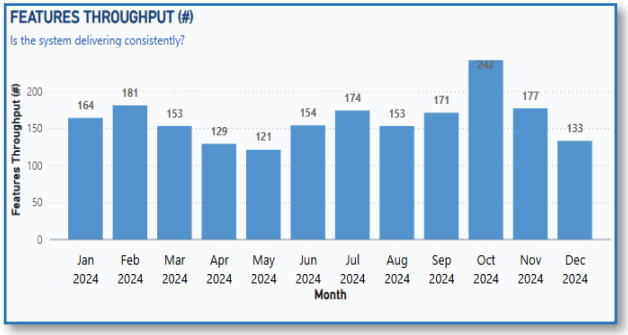
Metrics Description:

This chart shows whether a team has enough backlog to plan and provide optionality or is overplanning.

Formula Description:

The Sum of Effort of Features in Ready Stage at the End of the Month

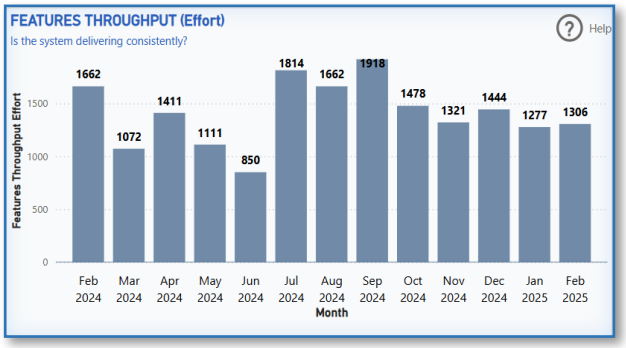
Feature Throughput (#)



Metrics Description:
The **Feature Throughput (#)** measures the amount of work completed in a period. For Product Line Teams, this means Features and the period is one month. We measure throughput to understand how much work can be delivered in a given time period based on actual data. We then use that information when creating our Product roadmaps to create a realistic plan of when work will be delivered.

Formula Description:
Count of Features Completed

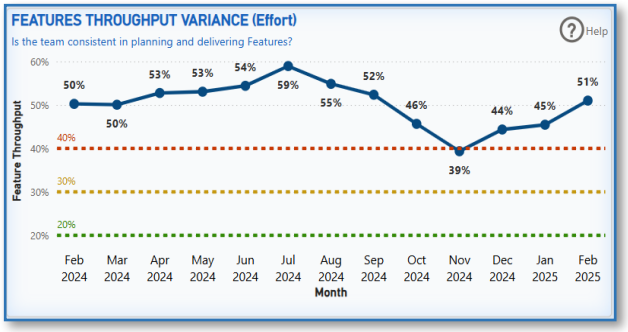
Feature Throughput (Effort)



Metrics Description:
This chart shows the throughput measured in estimated effort for each month. It measures the team's ability to assess its planned effort.

Formula Description:
Sum of Feature Effort Completed

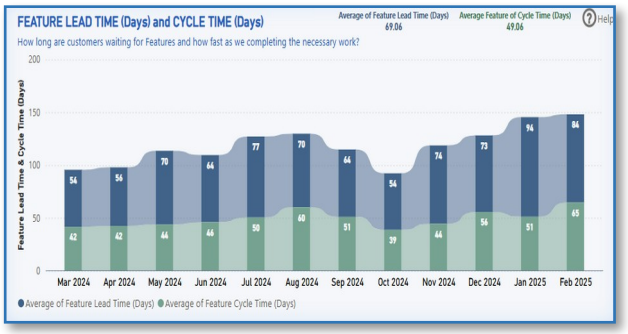
Feature Throughput Variance (Effort)



Metrics Description:
The **Feature Throughput Variance (Effort)** indicates the team's ability to accurately estimate the work required to complete features. It's important for the teams to be able to estimate the effort correctly to make and meet commitments to customers and stakeholders. The target variance is below 30%, while variance above 40% is considered a critical condition.

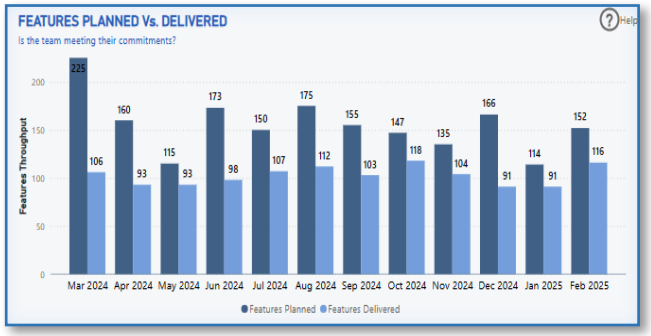
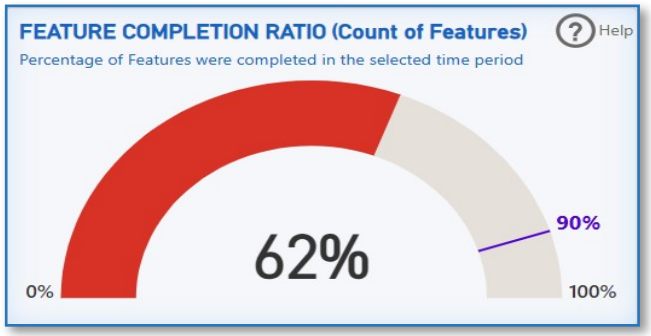
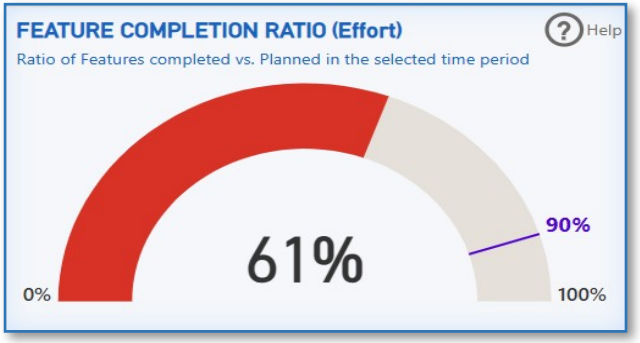
Formula Description:
Percent Variation from the Last Three Months' Throughput Effort

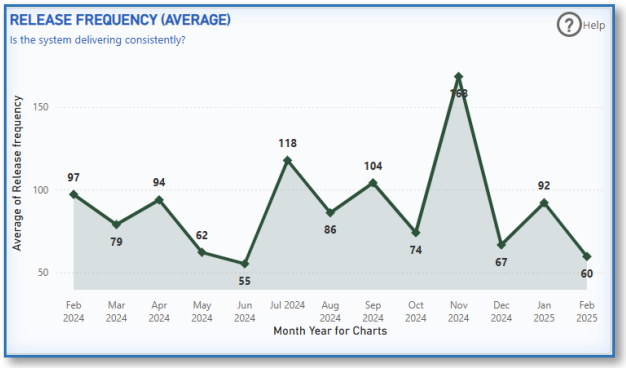
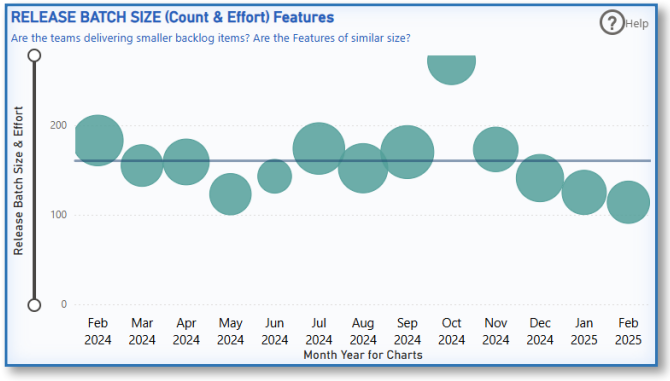
Feature Lead Time (Days) and Cycle Time (Days)



Metrics Description:
The **Feature Lead Time (Days) and Cycle Time (Days)** measure the average time to complete work from the customer's and internal perspectives, respectively. Responding quickly to changing requirements requires reducing both Cycle Time and Lead Time over time through system efficiencies.

Formulas Description:
Feature Lead Time (Days): Average Days Features Spend from Solution Design to Completed
Feature Cycle Time (Days): Average Days Features Spend from in Progress to Completed

Features Planned Vs. Delivered  	**Metrics Description:** The chart shows the number of Features released in the month compared with the Features with a Target End in the same month (Planned for the same month). These are the two components of the Feature Completion Ratio (Count). The goal is a ratio between 90% and 100%. **Formula Description:** Count of Features Planned to be Released in the Month and Count of Features Released
Feature Completion Ratio (Count of Features)  	**Metrics Description:** The **Feature Completion Ratio (Count of Features)** measures the ratio of Features released in the selected period divided by the Features with a Target End in the same selected period (planned for the same selected period). The target ratio is between 90% and 100%. Risk conditions **(Yellow)** are ratios between 80% and 89%, and Alert conditions **(Red)** are ratios above 100% (overestimated effort during planning, injected work accepted and completed) and under 80% (underestimated effort during planning, injected work replacing planned work, and other risks). **Formula Description:** Count Features Released ÷ Count Features with Target End in Month
Feature Completion Ratio (Effort)  	**Metrics Description:** The **Feature Completion Ratio (Effort)** measures the ratio of the estimated effort necessary to complete the Features released in the selected period divided by the Features with a Target End in the same selected period (planned for the same selected period). The target ratio is between 90% and 100%. Risk conditions **(Yellow)** are ratios between 80 and 89%, and Alert conditions **(Red)** are ratios above 100% (overestimated effort during planning, injected work accepted and completed) and under 80% (underestimated effort during planning, injected work replacing planned work, and other risks). **Formula Description:** Sum Effort of Features Released ÷ Sum Effort of Features with Target End in Month
Release Date Success 	**Metrics Description:** The **Release Date Success** measures the percentage of releases deemed completed on time by the team. **Formula Description:** Percent Releases Delivered by Target Date

Release Frequency (Average)  	**Metrics Description:** The **Release Frequency (Average)** metric measures the ability of the team to deliver consistently month after month. **Formula Description:** Days From the First Feature in Progress to the Actual Release Date
Release Batch Size (Count & Effort) Features  	**Metrics Description:** The **Release Batch Size (Count & Effort) Features** measure the average size of the releases for each month. The team's goal is to work on smaller backlog items to reduce the cost of changing requirements and provide incremental solutions to customers that can be validated early. **Formulas Description:** Release Batch Size (Count): The Average Number of Features in a Release Release Batch Size (Effort): The Average Sum of Effort of Features in a Release