Azure DevOps

Testing in Production

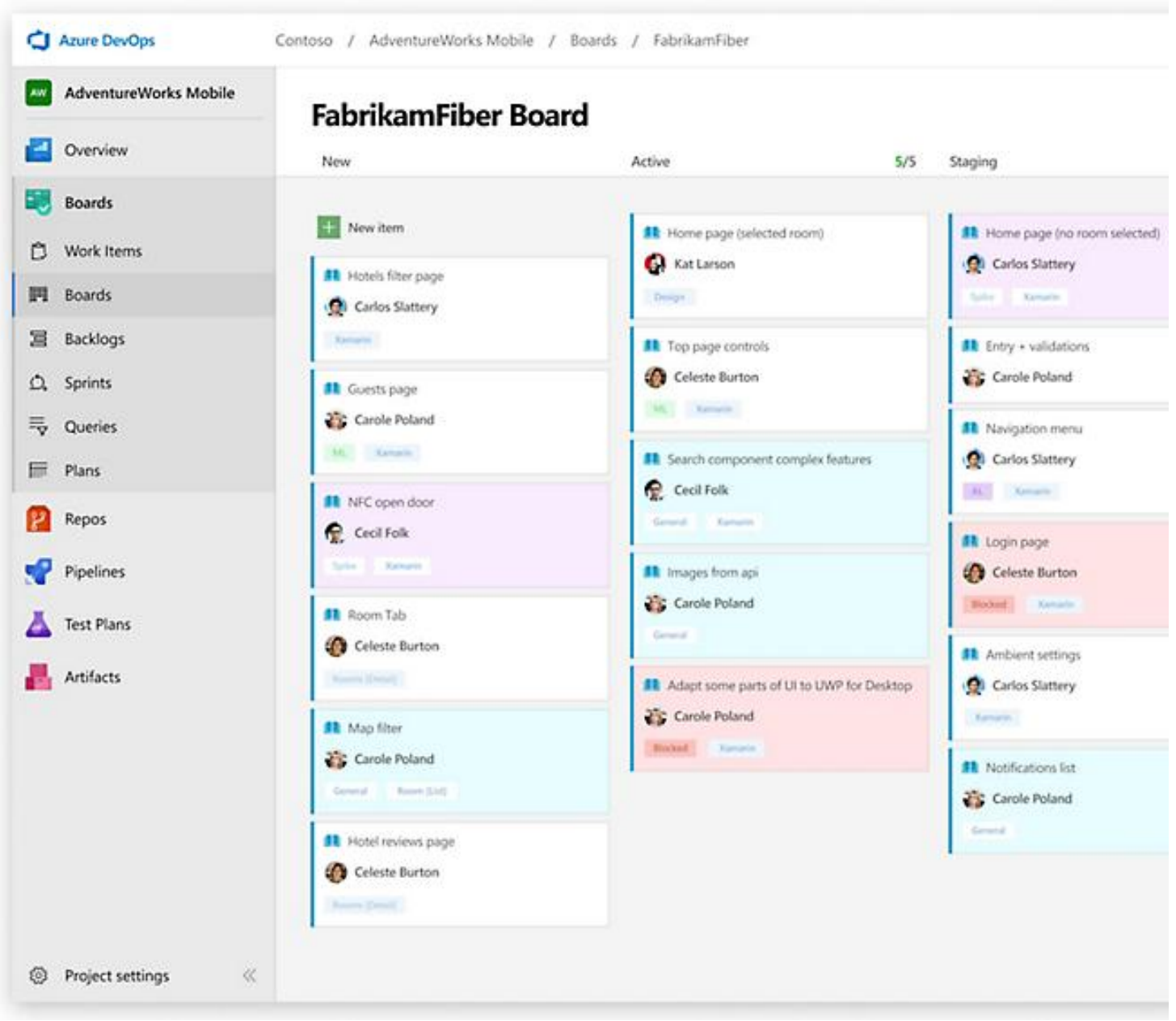
How we fixed New Boards Hub performance in Azure DevOps

Dave Paquette
Principal Software Engineer - Microsoft

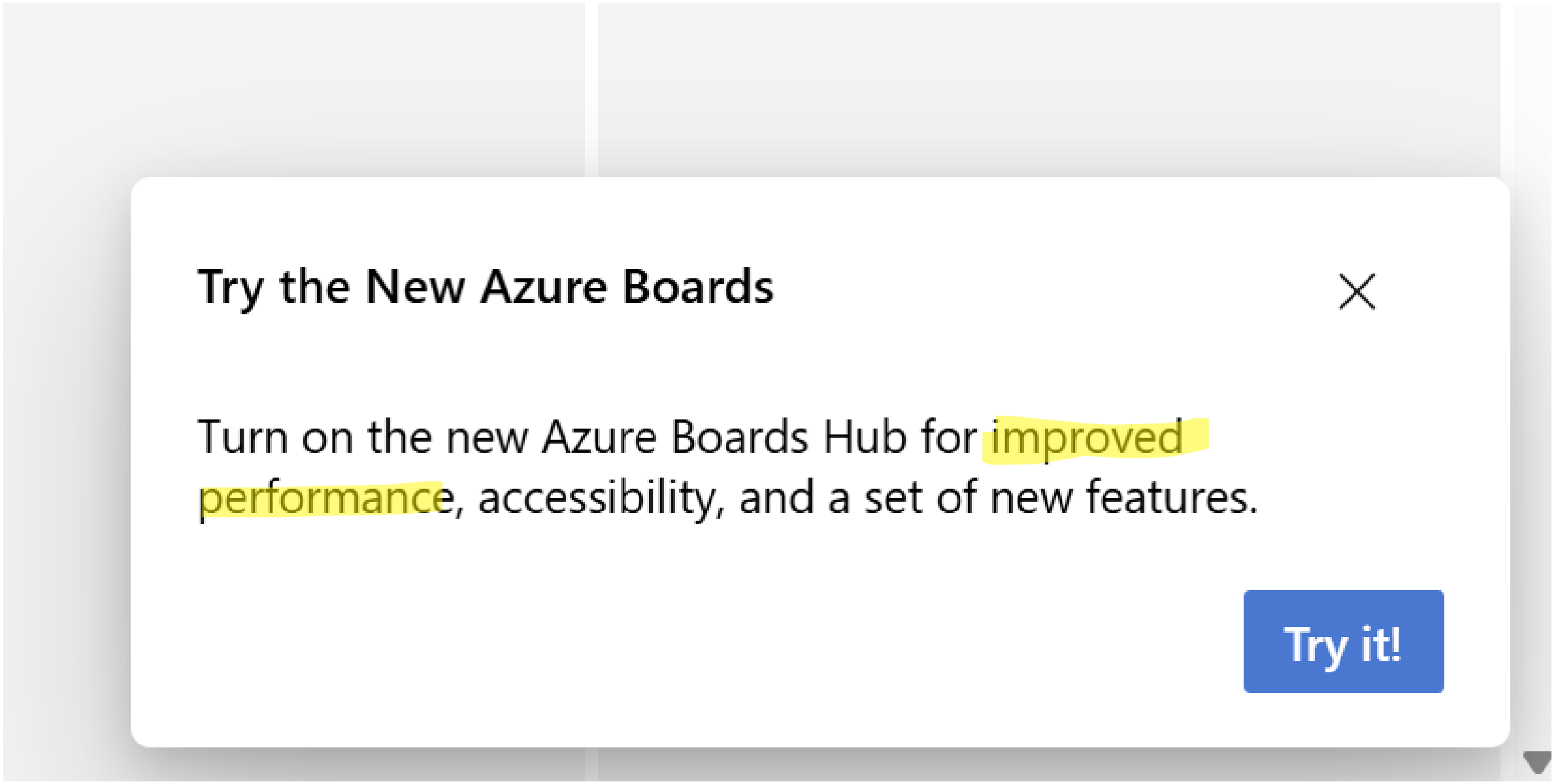
Microsoft

Azure DevOps

What is Boards?



Why did New Boards need fixing?

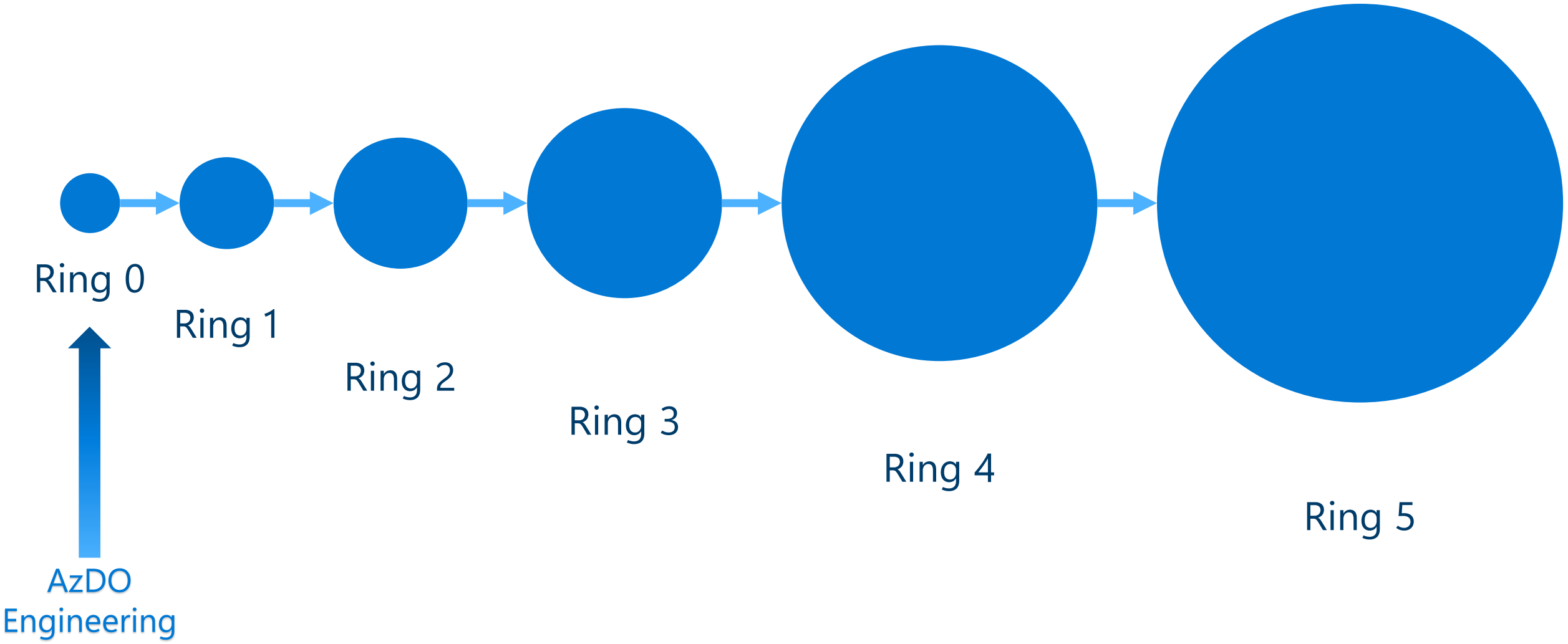

Try the New Azure Boards
Turn on the new Azure Boards Hub for improved performance, accessibility, and a set of new features.
Try it!

Safe Deployment Practices

- **Safe Deployment through rings**
- Feature Flags
- User Opt-In
- Care about Quality Signals
- Deploy Often (and during work hours)

https://learn.microsoft.com/en-us/devops/operate/safe-deployment-practices

Azure DevOps Deployment Rings



Safe Deployment Practices

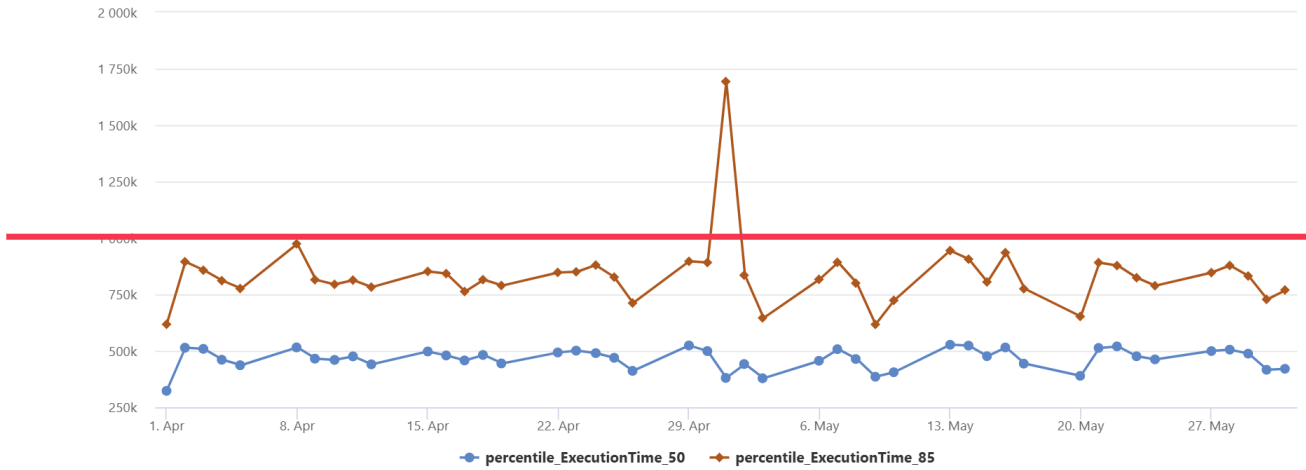
- Safe Deployment through rings
- **Feature Flags**
- **User Opt-In**
- Care about Quality Signals
- Deploy Often (and during work hours)

https://learn.microsoft.com/en-us/devops/operate/safe-deployment-practices

Azure DevOps

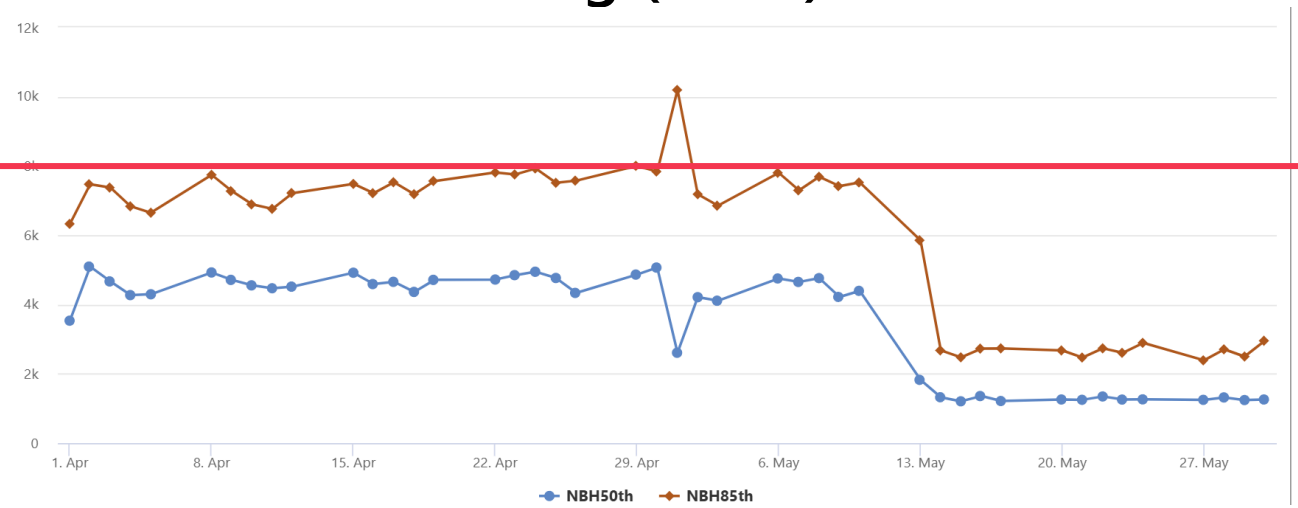
How do we measure performance?

Server Response Time



1 second

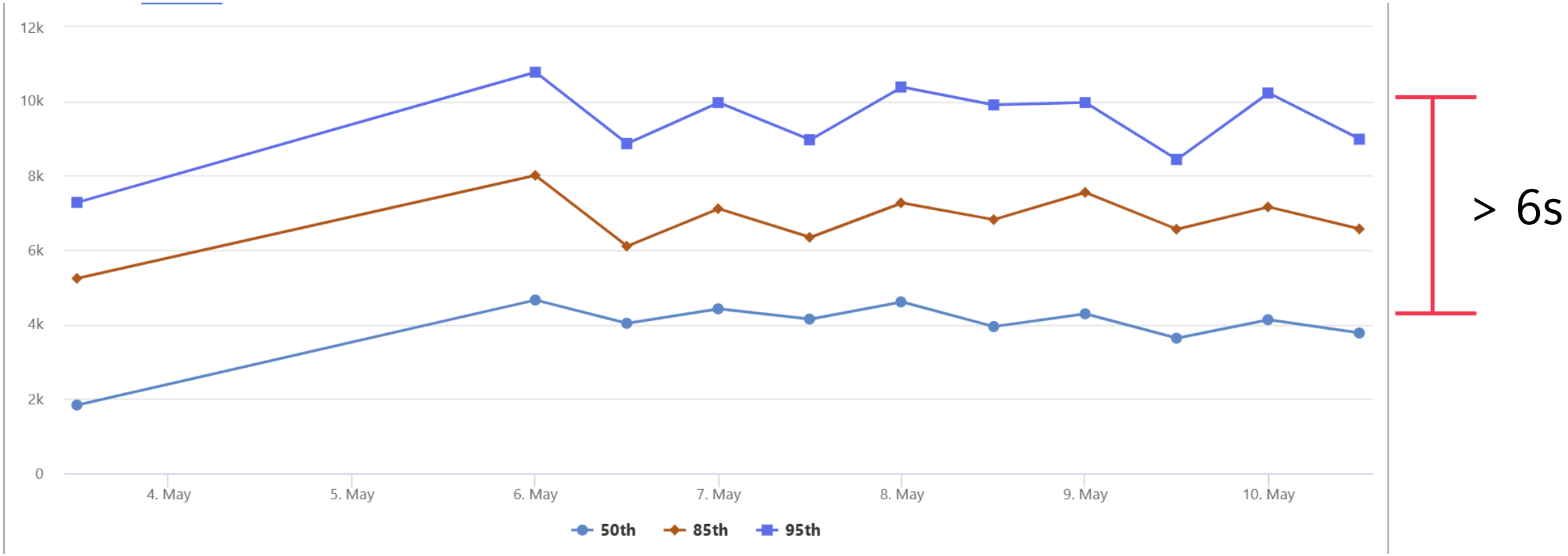
Real User Monitoring (RUM)



8 seconds

Time to Interactive

But TTI can vary



Apdex

3 Buckets

- 😍 Satisfied < 1s
- 🙄 Tolerating 1s – 4s
- 😡 Frustrated >4s

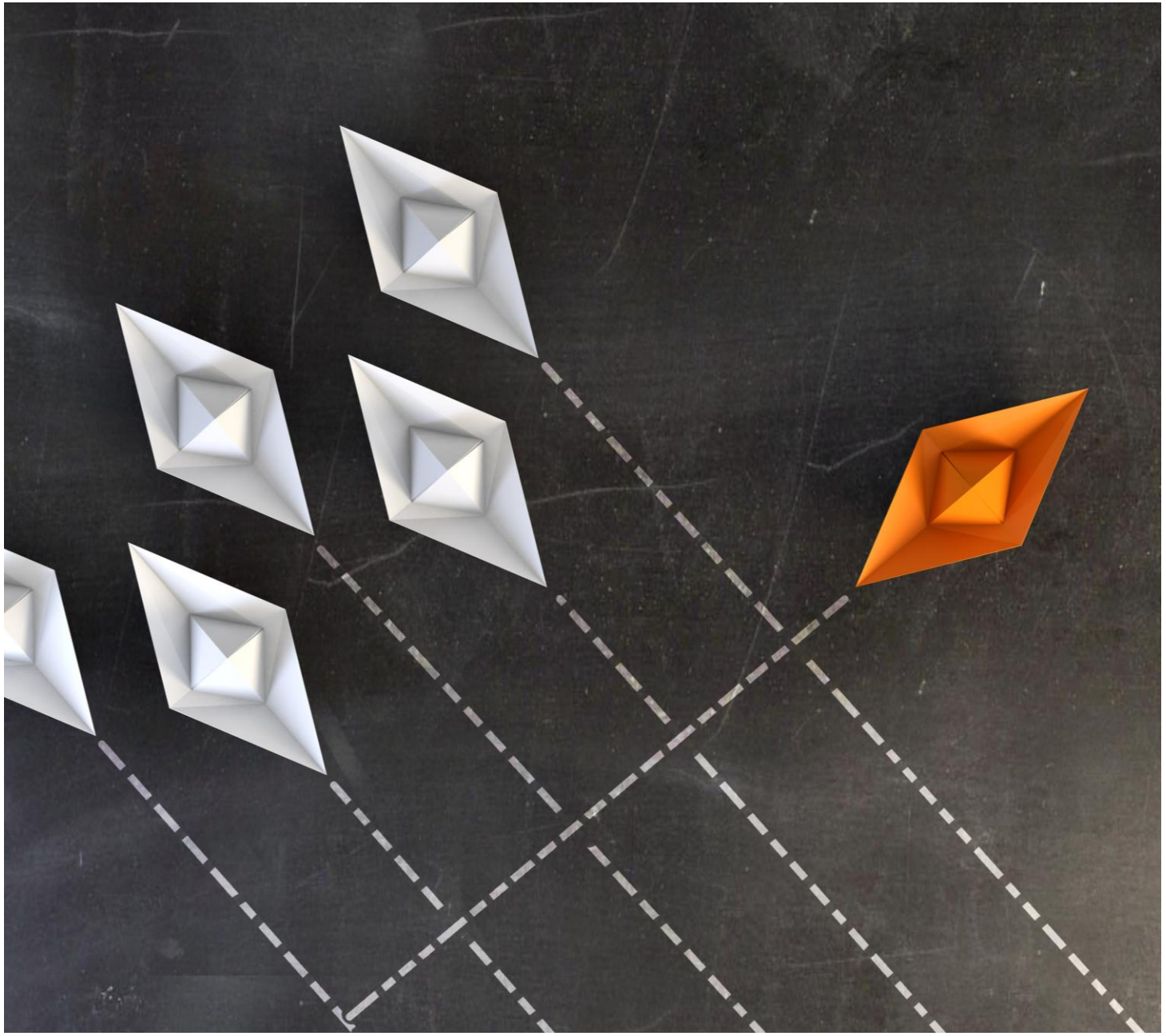
$$Apdex = \frac{😍 + \frac{🙄}{2}}{TotalSamples}$$

What Happened with New Boards?

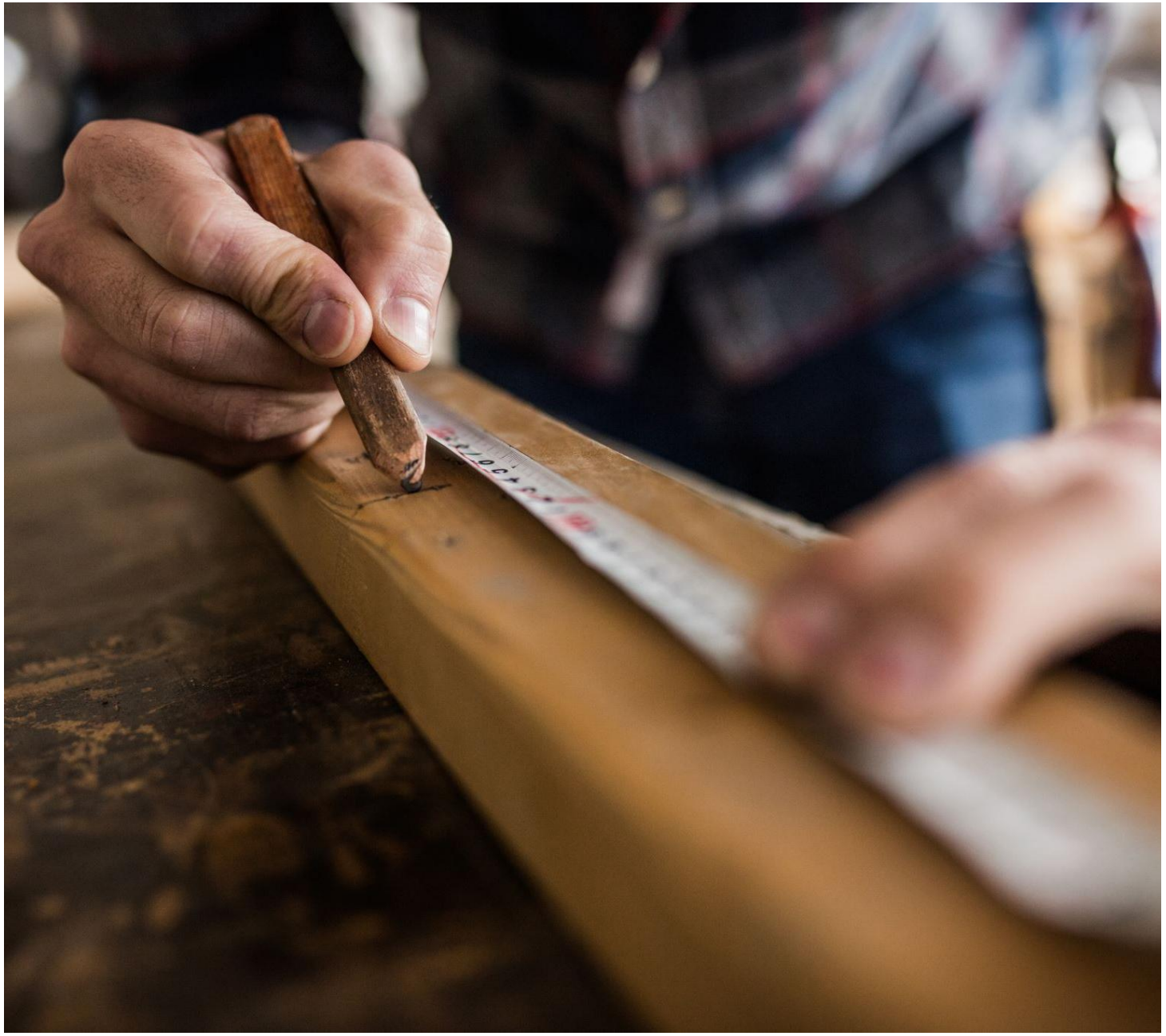
- Misleading telemetry
- Huge variance depending on data shape
  - Larger customers with biggest perf problems were in later rings
- Works on my machine
- Client-side memory usage was not an area we were considering

How can we fix this?

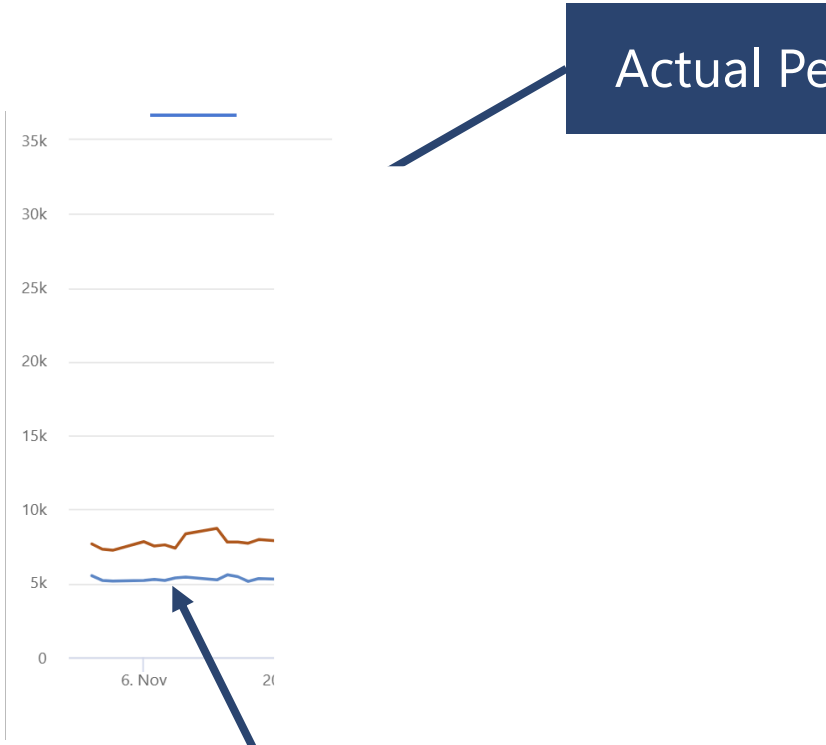
Our Strategy

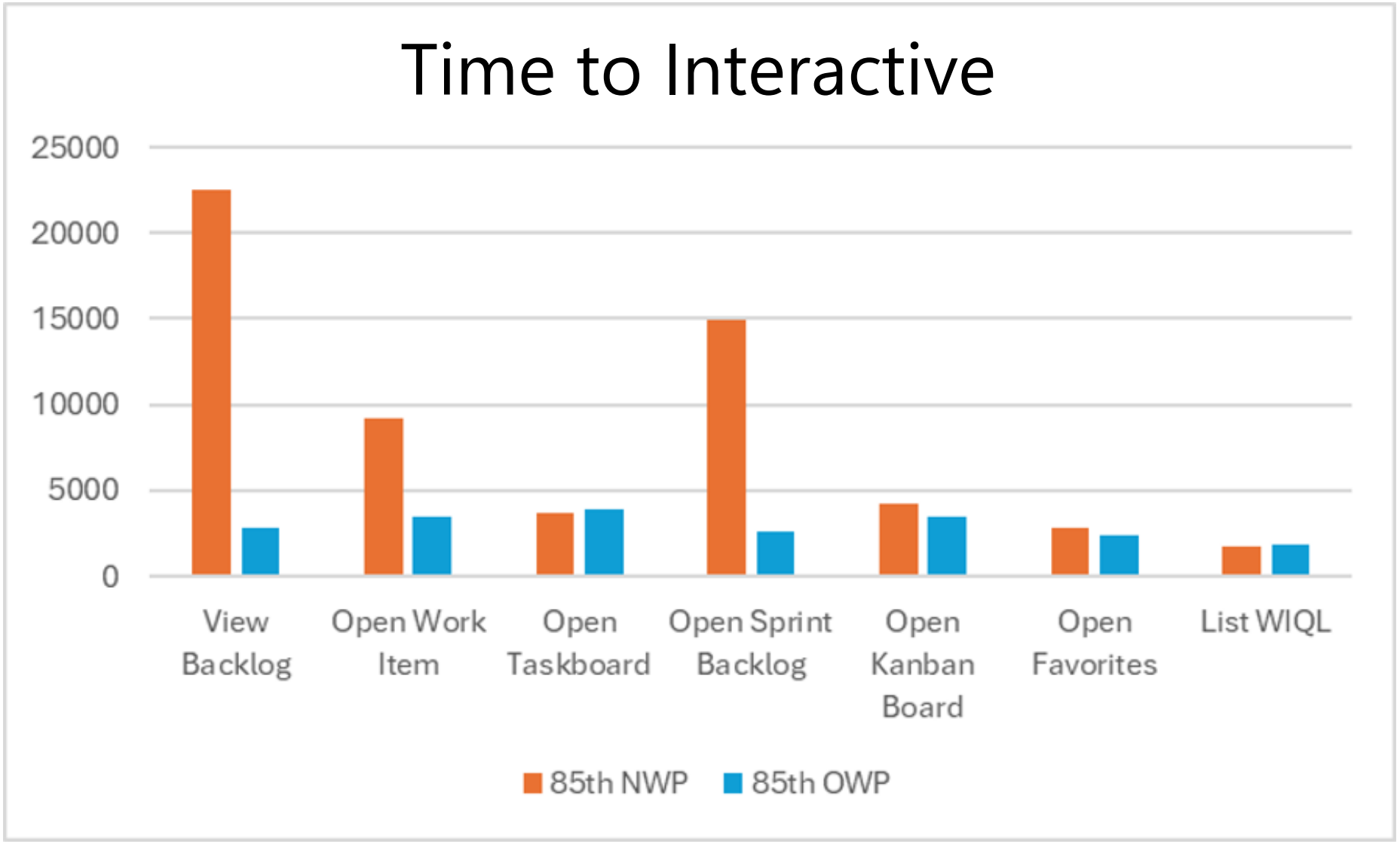


Step 1 – Fix Telemetry and Build Dashboards

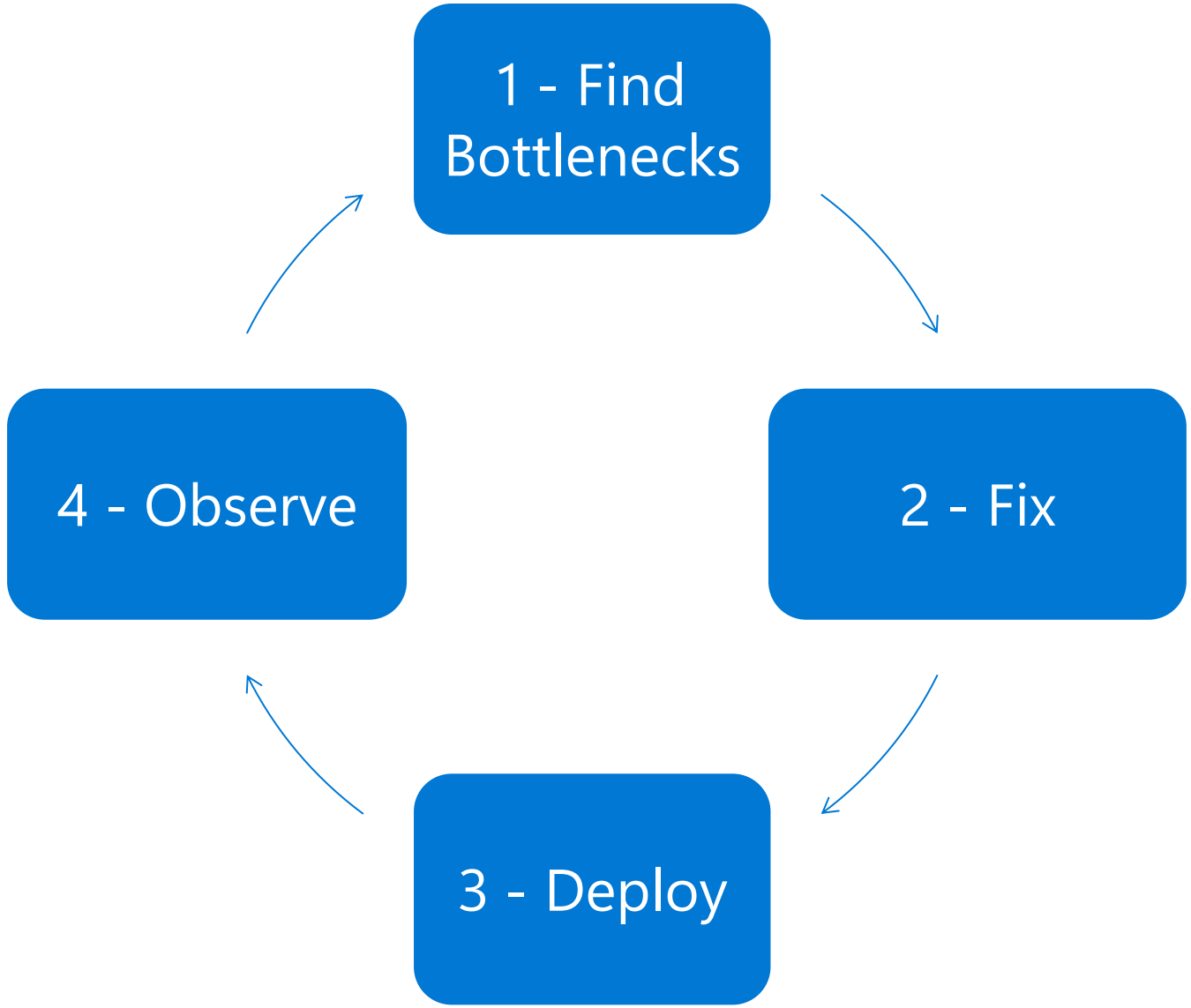


Reviewing TTI markers




Time to Interactive
25000
20000
15000
10000
5000
0
View Backlog
Open Work Item
Open Taskboard
Open Sprint Backlog
Open Kanban Board
Open Favorites
List WIQL
85th NWP
85th OWP

Step 2
Iterative Data Driven Improvements



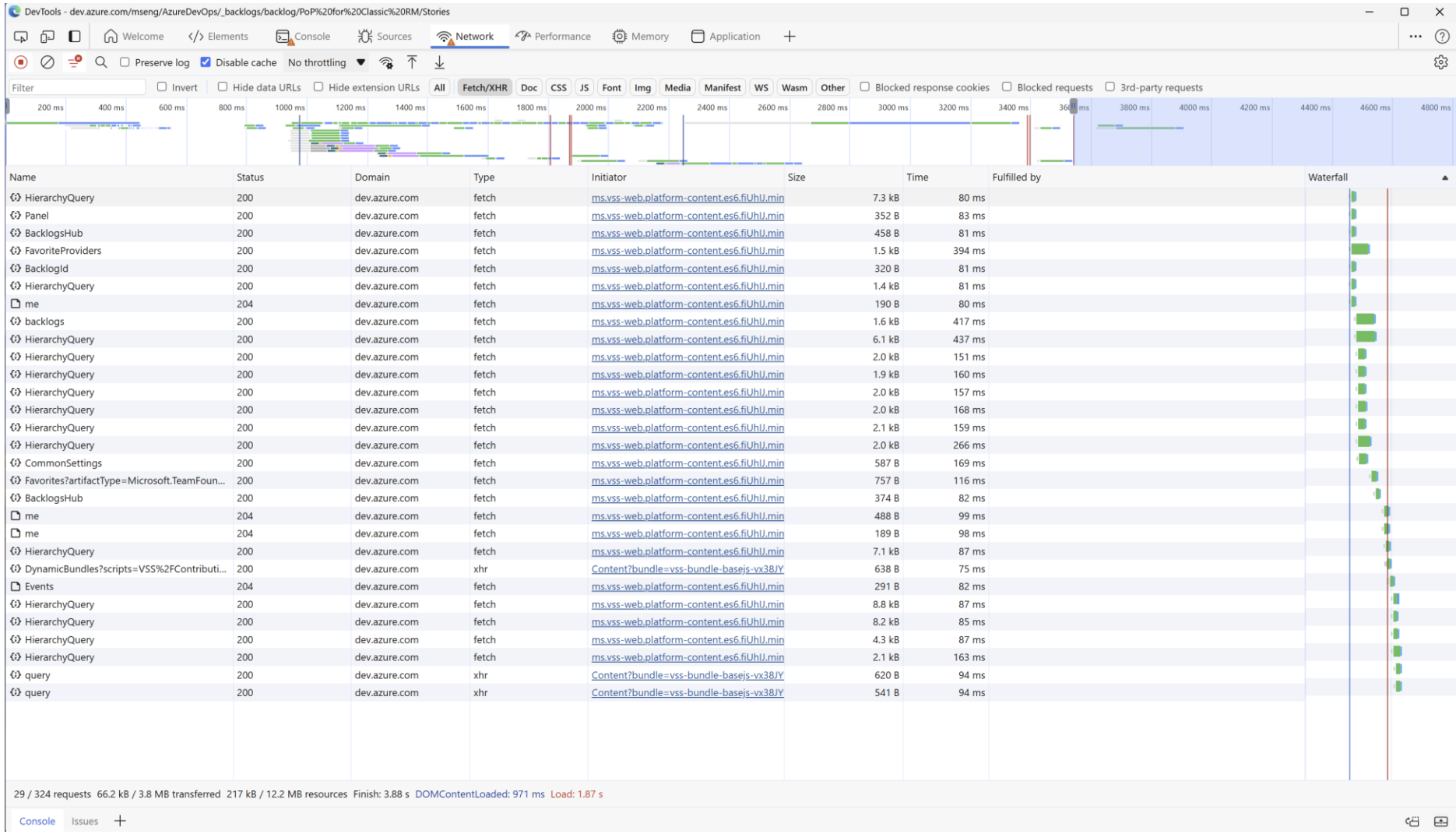
Finding Bottlenecks

- Telemetry
- Performance Bar
- Browser Dev Tools
  - Network Tools
  - Memory Profiler
  - Performance Profiler

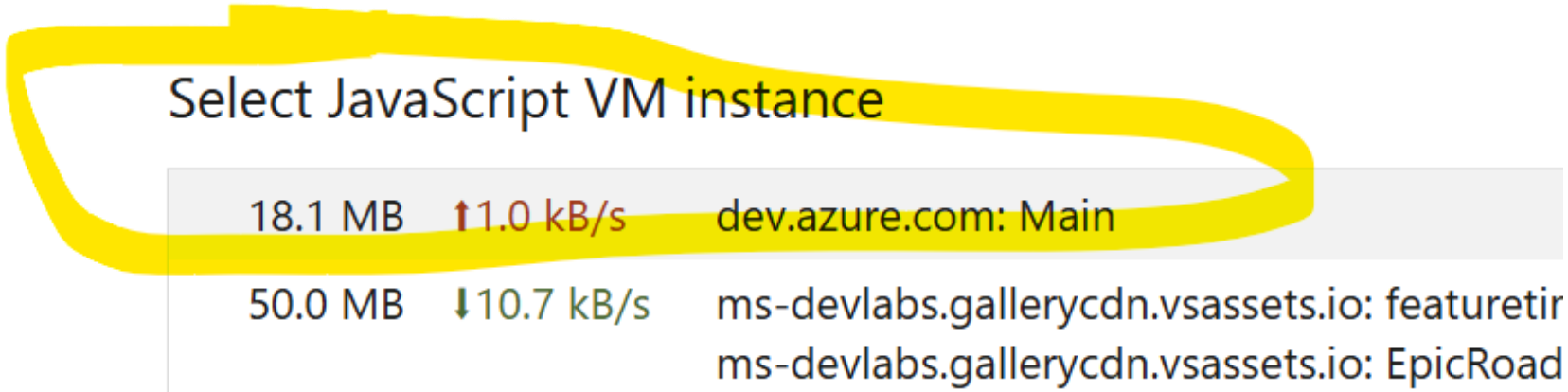
Telemetry

Performance Bar

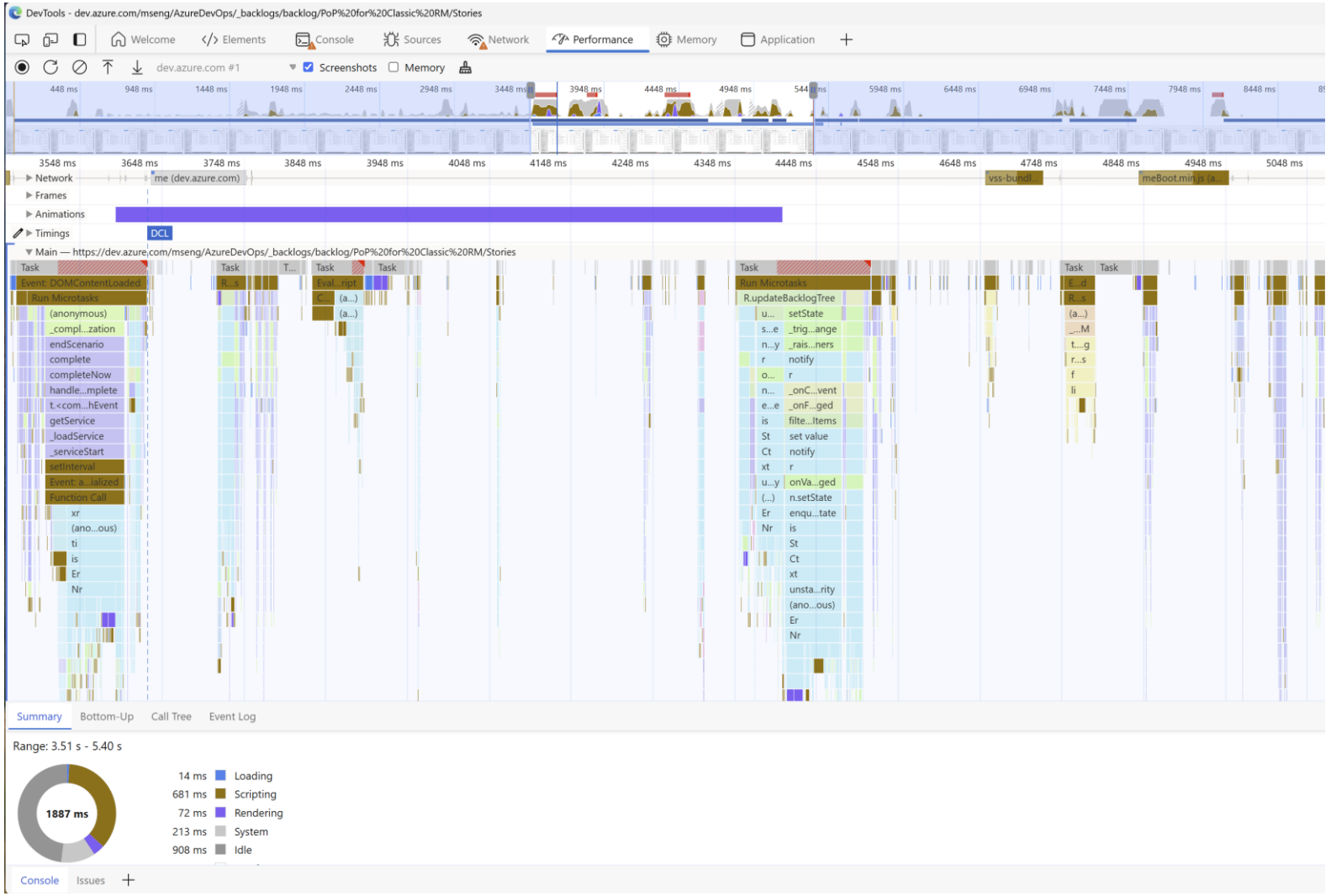
Network Tools



Memory Profiler



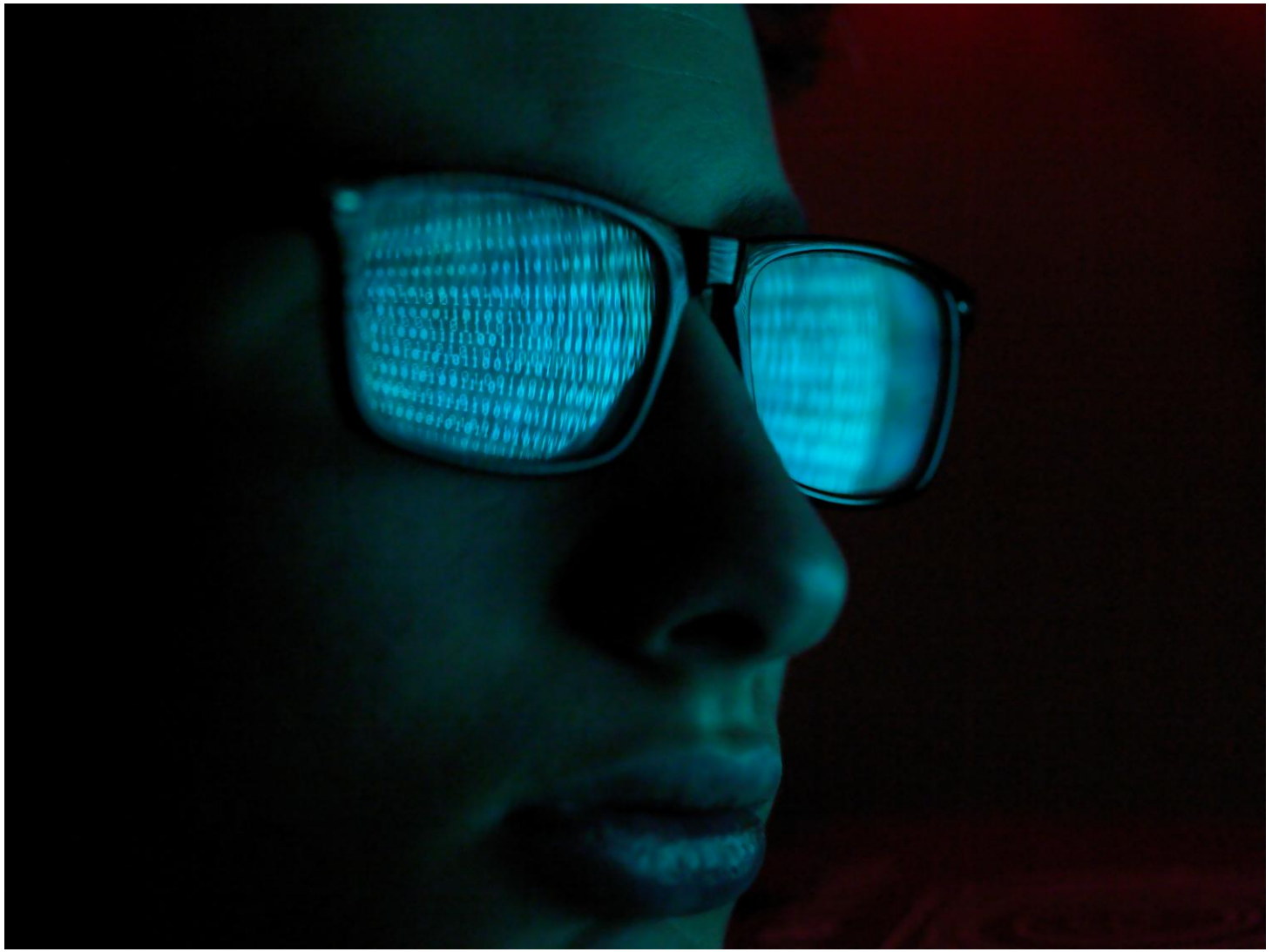
Performance Profiler



Types of Bottlenecks

- Inefficient client-side code
- Inefficient server-side code
- Loading too much data
- Loading data too early
- Loading data too often

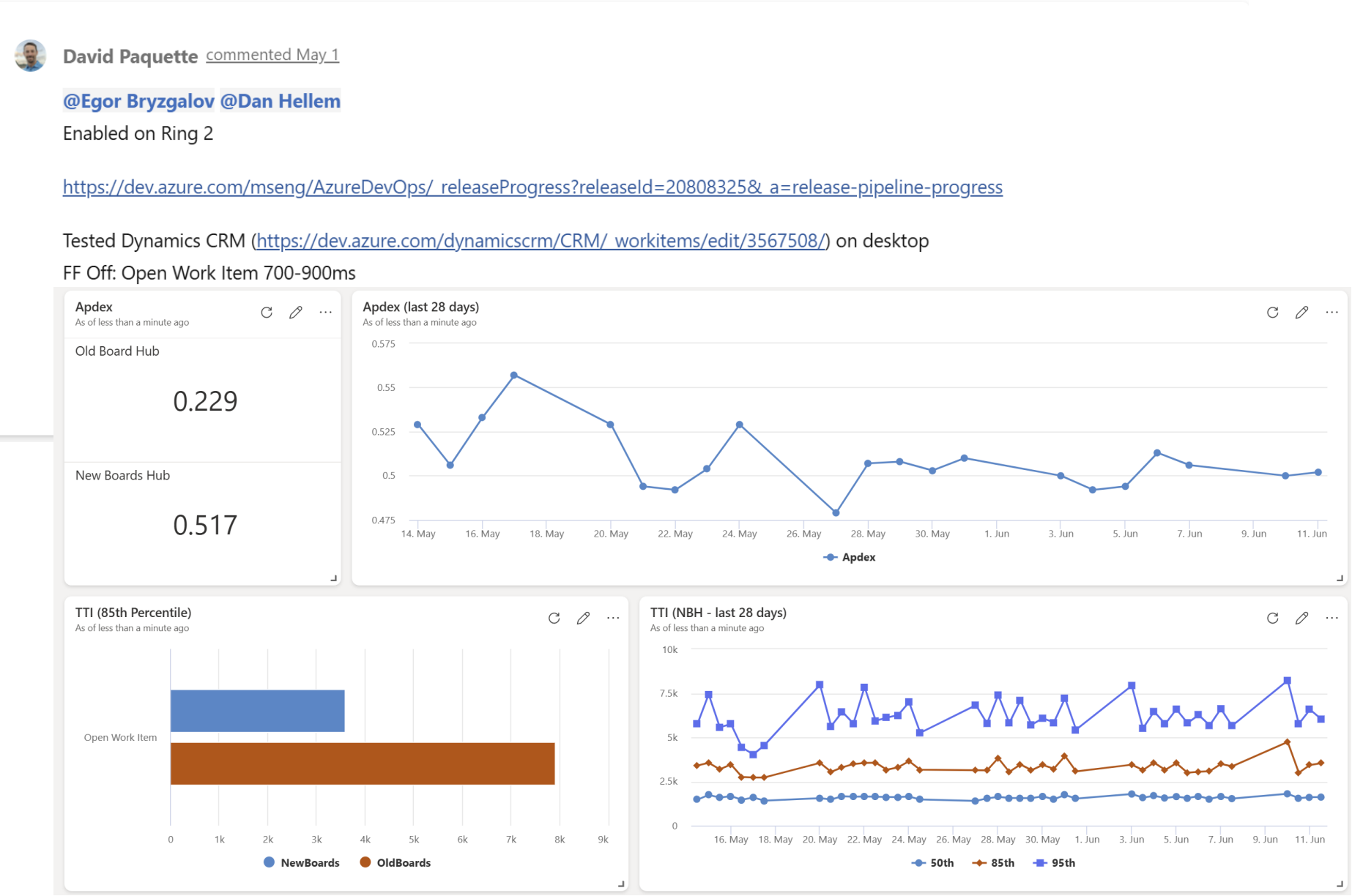
Fix



Deploy

- Feature Flag EVERYTHING
- Aggressively back-ported
  - FF Off by default

Observe


David Paquette commented May 1
@Egor Bryzgalov @Dan Hellem
Enabled on Ring 2
https://dev.azure.com/mseng/AzureDevOps/_releaseProgress?releaseId=20808325&_a=release-pipeline-progress
Tested Dynamics CRM (https://dev.azure.com/dynamicscrm/CRM/_workitems/edit/3567508/) on desktop
FF Off: Open Work Item 700-900ms
Apdex
As of less than a minute ago
Old Board Hub
0.229
New Boards Hub
0.517
Apdex (last 28 days)
As of less than a minute ago
0.575
0.55
0.525
0.5
0.475
14. May
16. May
18. May
20. May
22. May
24. May
26. May
28. May
30. May
1. Jun
3. Jun
5. Jun
7. Jun
9. Jun
11. Jun
Apdex
TTI (85th Percentile)
As of less than a minute ago
Open Work Item
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
NewBoards
OldBoards
TTI (NBH - last 28 days)
As of less than a minute ago
10k
7.5k
5k
2.5k
0
16. May
18. May
20. May
22. May
24. May
26. May
28. May
30. May
1. Jun
3. Jun
5. Jun
7. Jun
9. Jun
11. Jun
50th
85th
95th

Secret Weapon - Internal Orgs

Ring 1: SQL

Ring 2: Dynamics

Ring 3: Office or Onedrive

Ring 4: Azure

Ring 5: Microsoft

Example 1: Memory Improvements in Backlog View

Categories

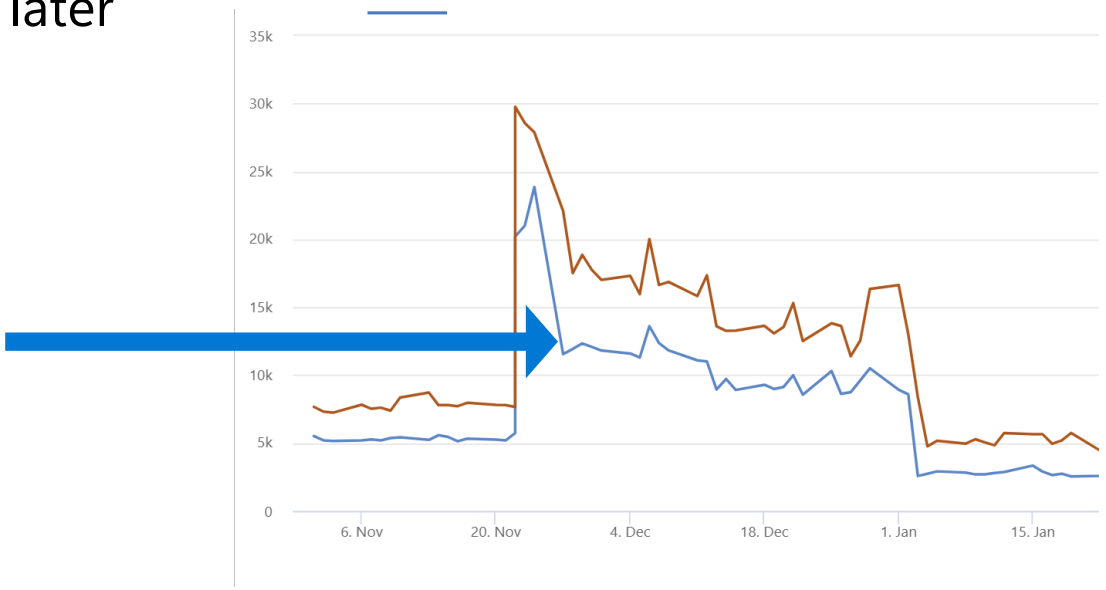
- Inefficient client-side code
- Loading too much data

Fix

- Optimize client-side code
- Deferred the too much data part to later

Results (Azure)

- Memory usage down from 1.3GB to 100MB
- Page load down from ~25s to ~12s



Example 2: Classification Nodes Meta Data Cache

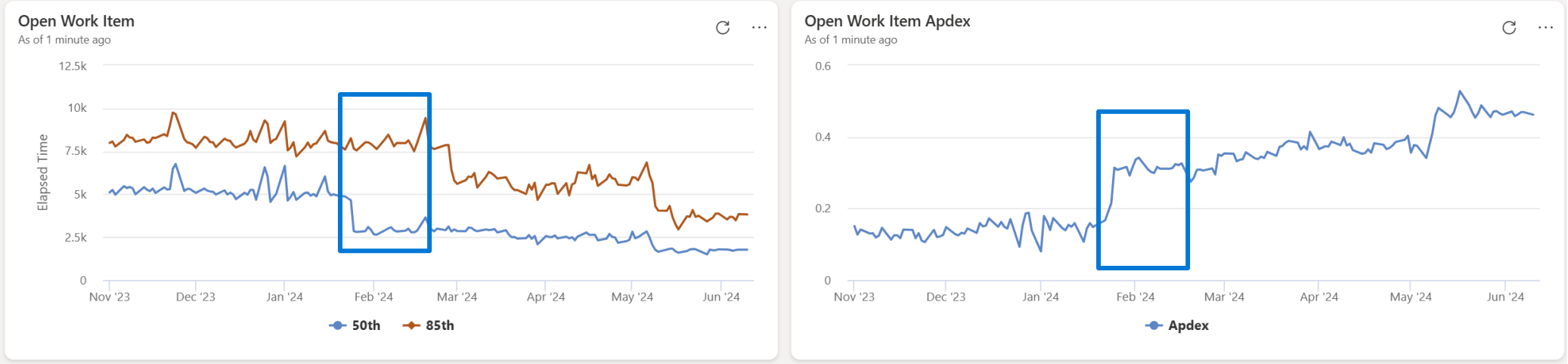
Categories

- Load data too often
- Loading too much data

Fix

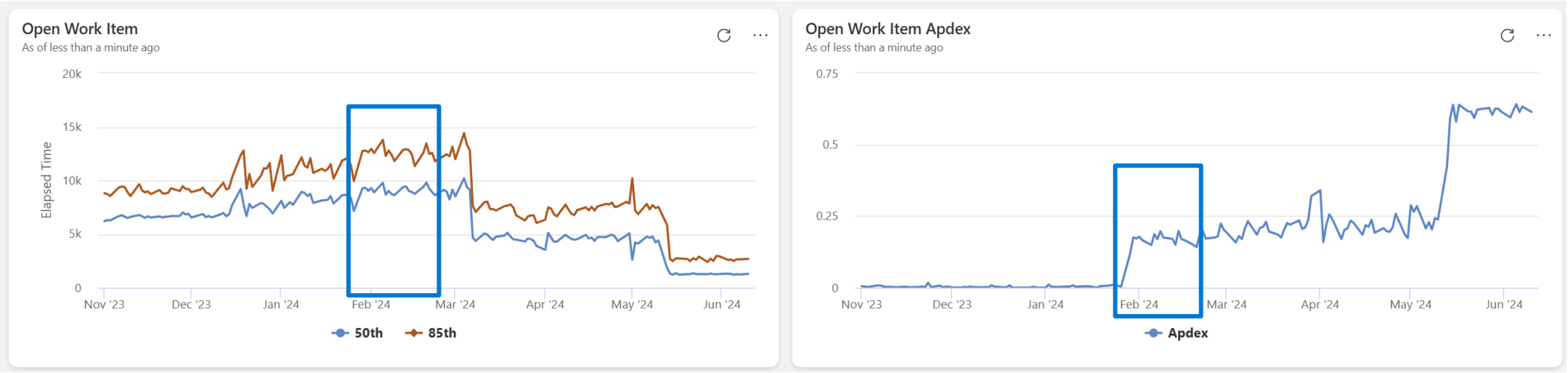
- Leverage Meta Data Cache
- Deferred loading too much data

Results (Azure)



Example 2: Classification Nodes Meta Data Cache

Results (BigBankCo)



Example 2: Classification Nodes Meta Data Cache

Example 3: Classification Nodes Data Reduction

Categories

- Loading too much data

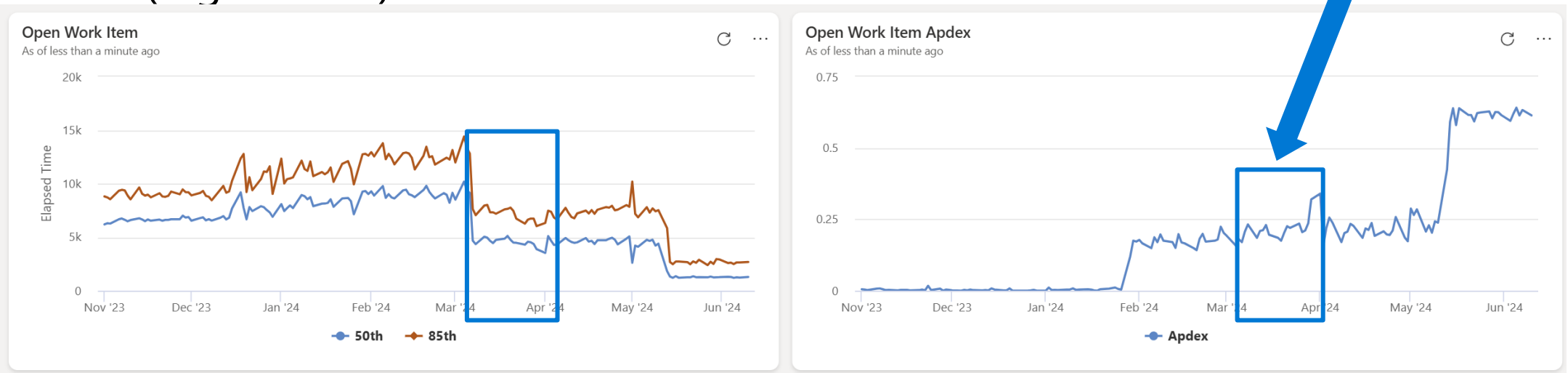
Fix

- Server was returning properties that were not used

Results (Azure)

- Uncompressed payload size reduced by almost 1/2
- Reduction in server load

Results (BigBankCo)



Example 4: Work Item Model Refactoring

Categories

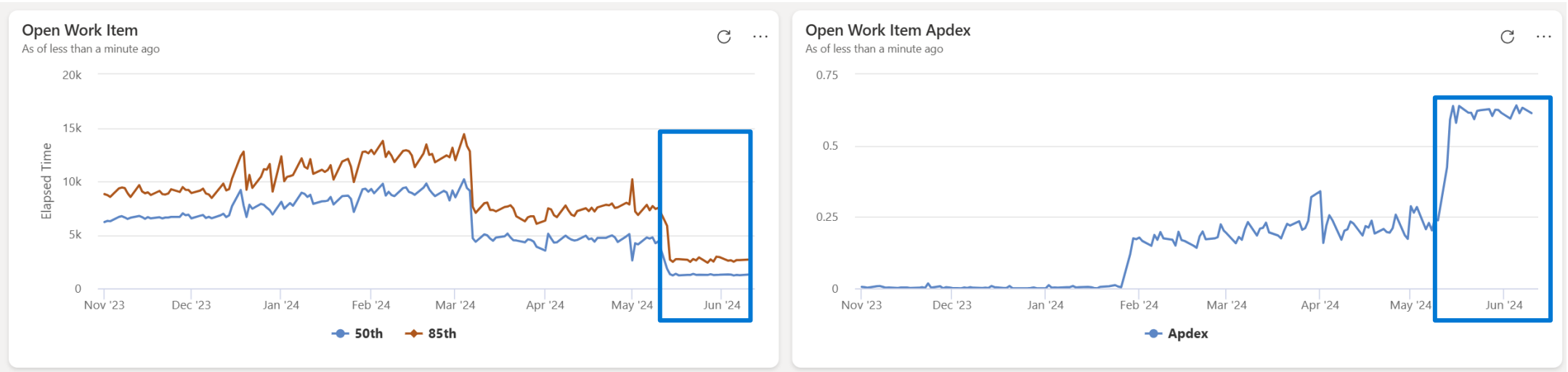
- Inefficient client-side code
- Loading data too early

Fix

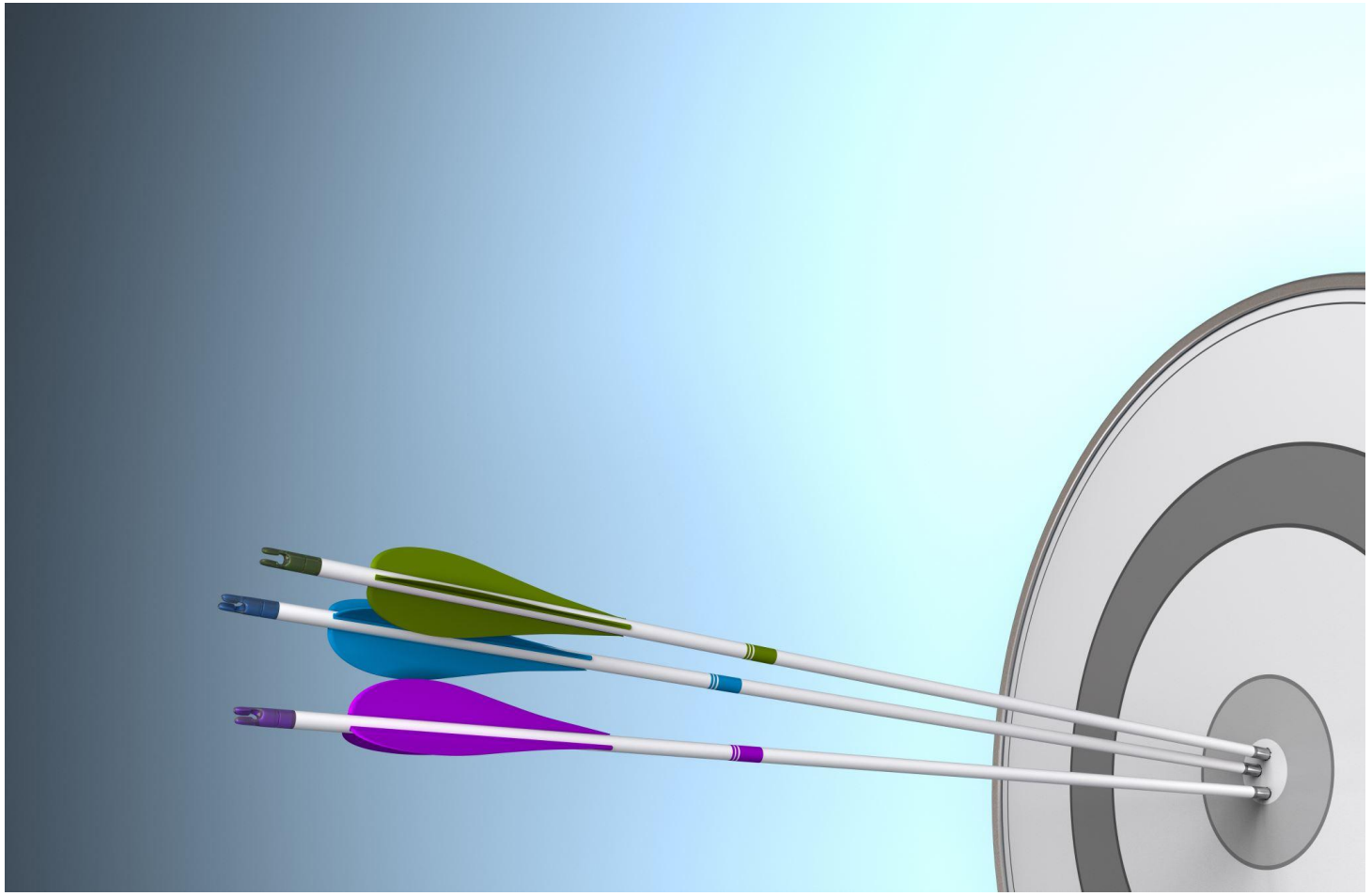
- Refactor client side WorkItem class
- Refactor data loading so it doesn't block UI rendering

Results (BigBankCo)

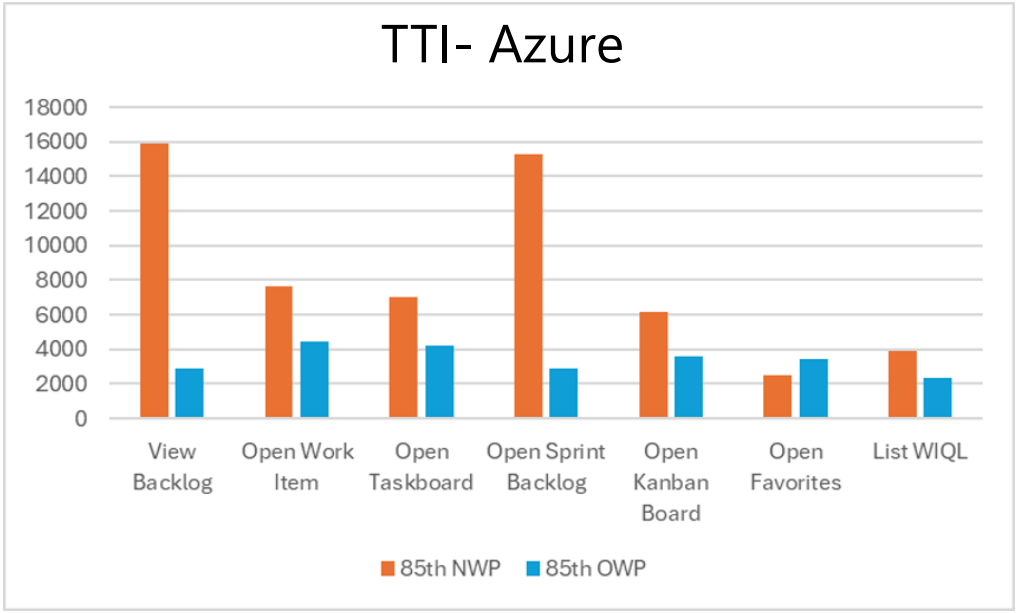
- Finally faster than Old Boards!

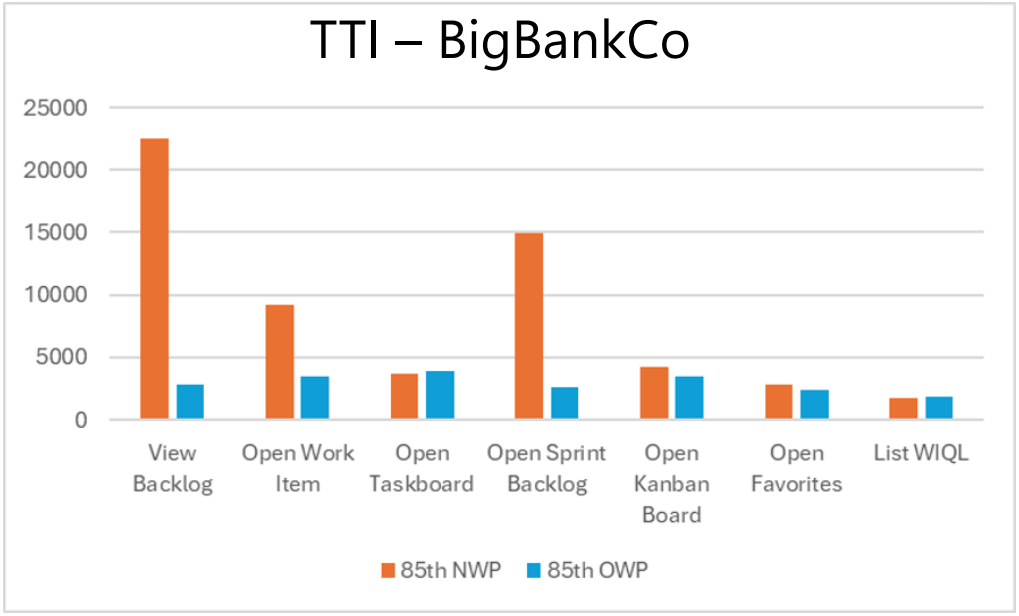


Results

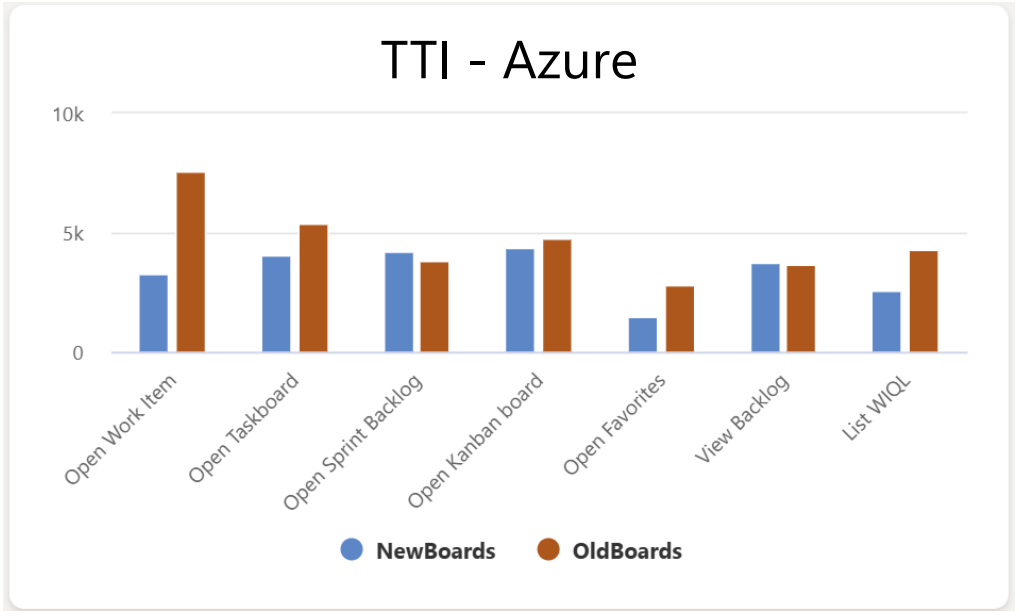


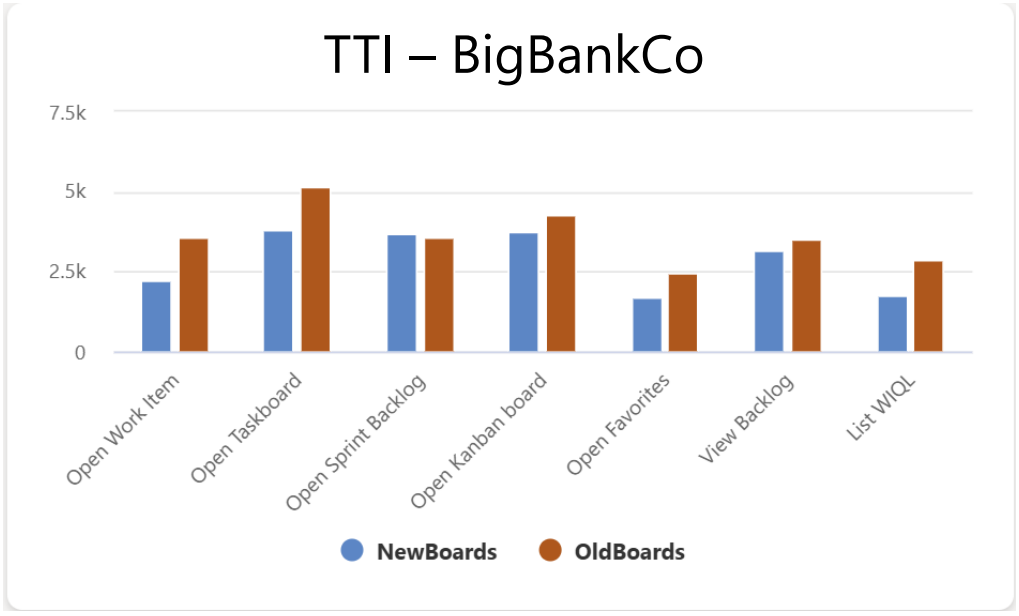
BEFORE



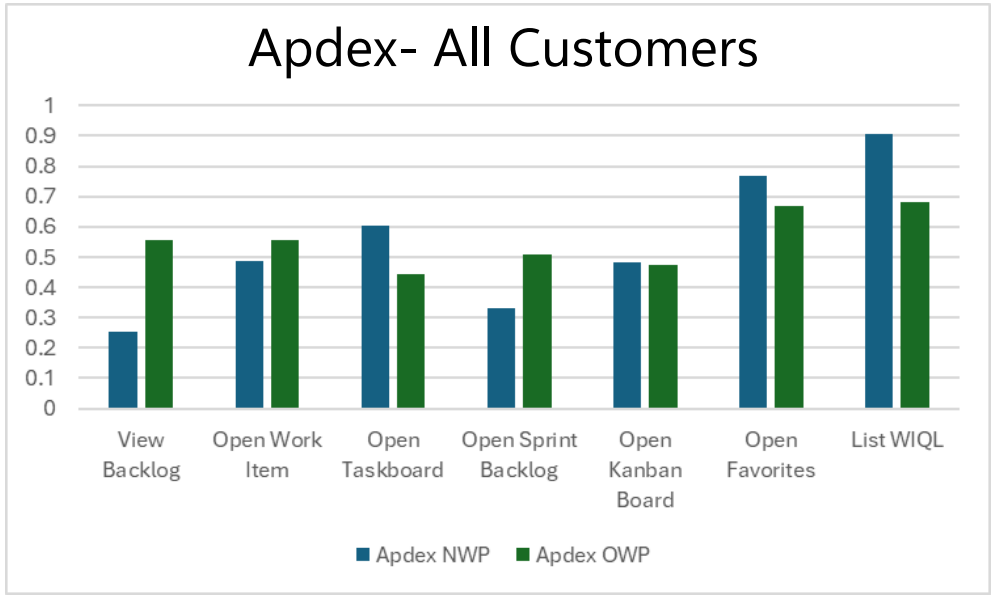


AFTER

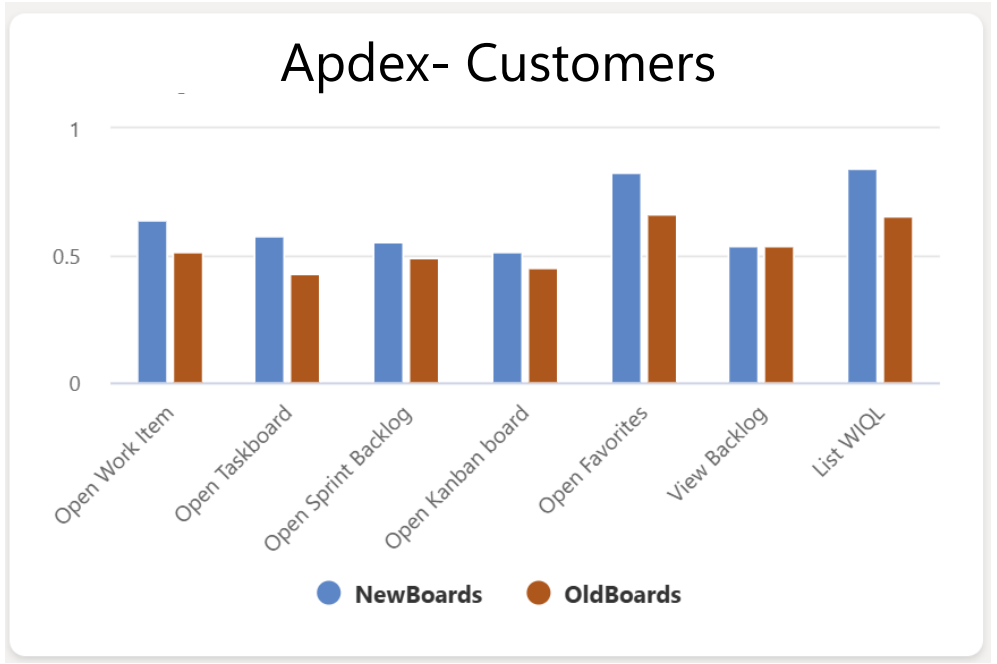




BEFORE



AFTER



Apdex

As of less than a minute ago

Old Board Hub

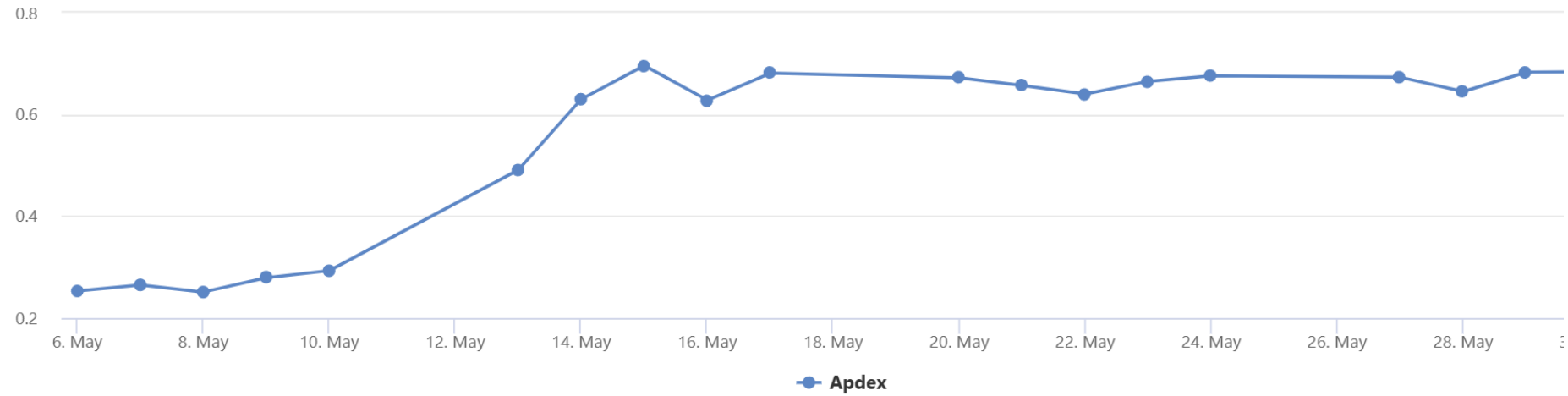
0.546

New Boards Hub

0.67

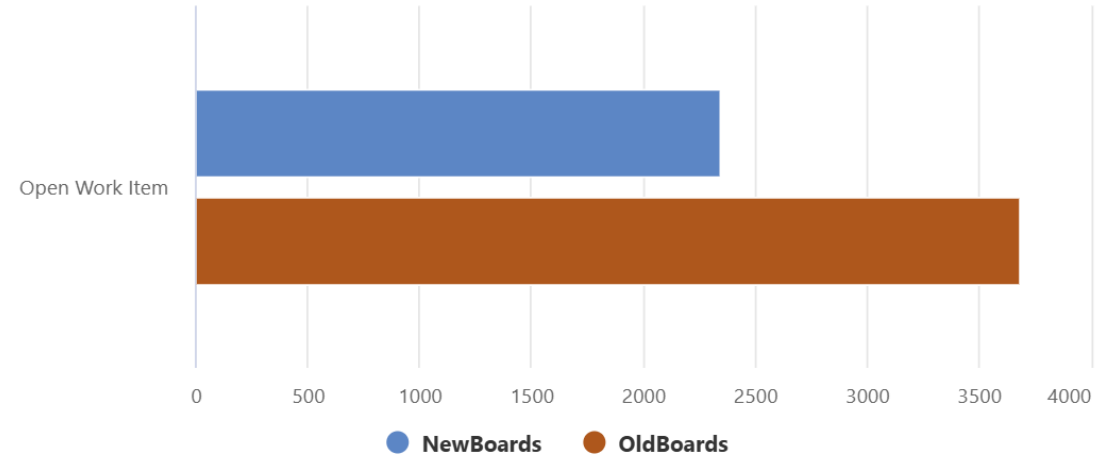
Apdex (last 28 days)

As of less than a minute ago



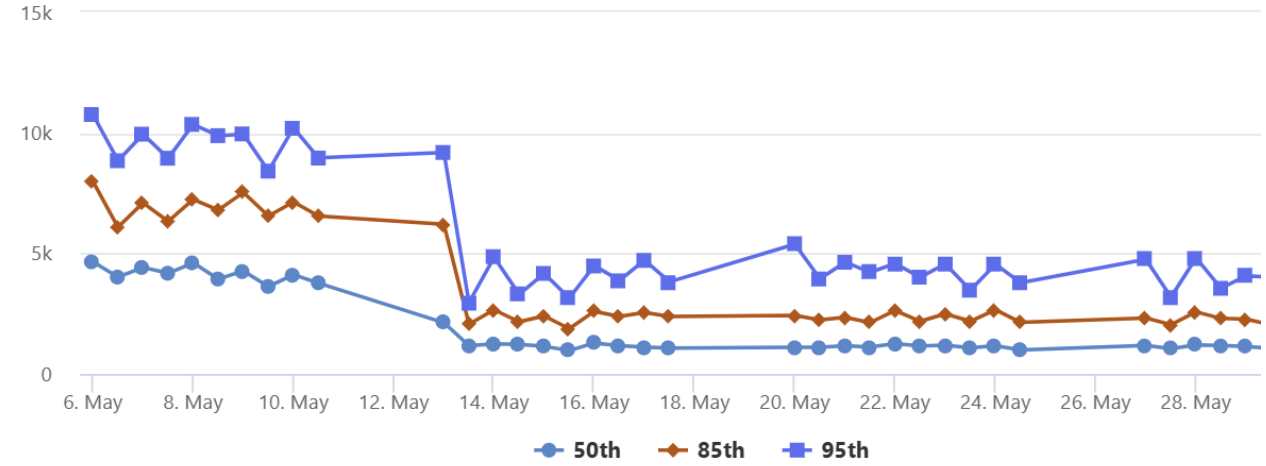
TTI (85th Percentile)

As of less than a minute ago

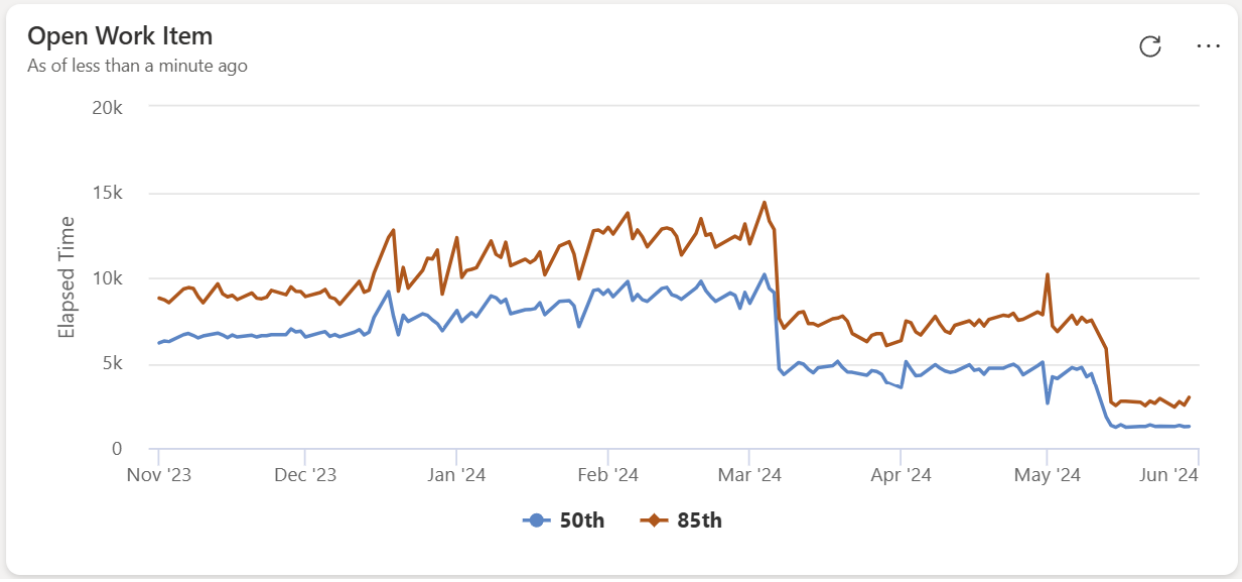


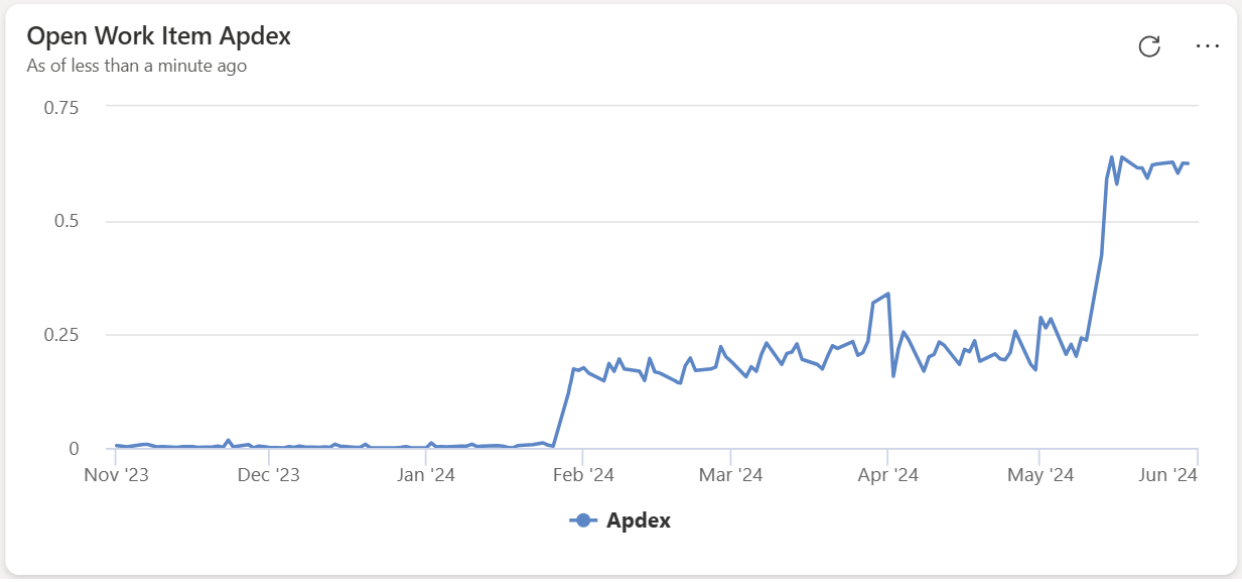
TTI (NBH - last 28 days)

As of less than a minute ago

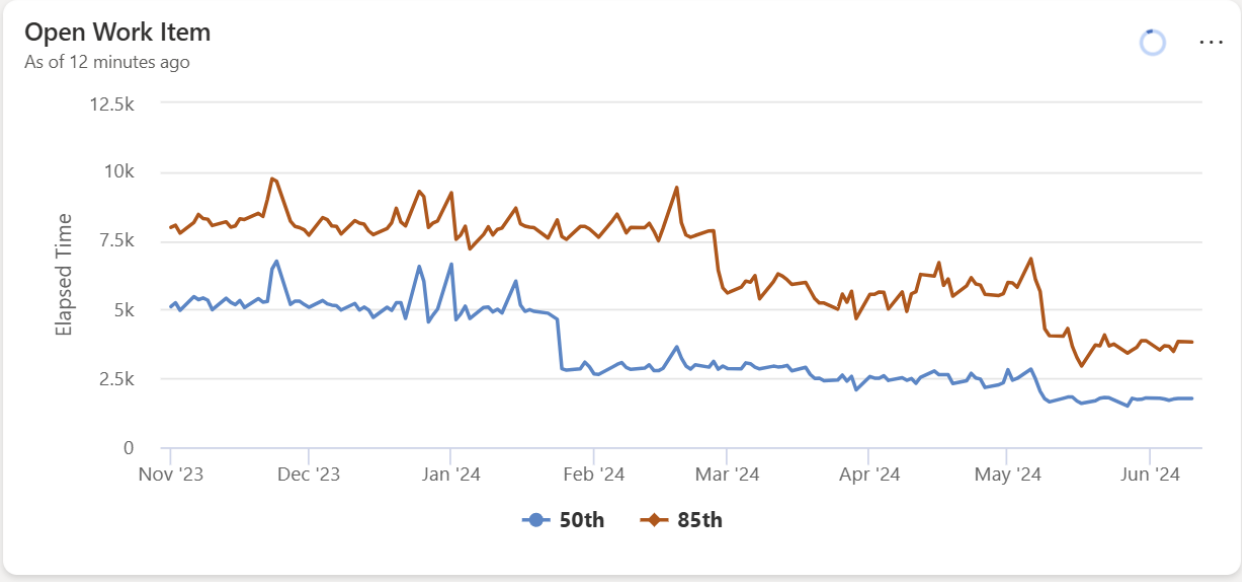


BigBankCo





All Customers



What did we learn?

Load the smallest set of data needed to render the UI

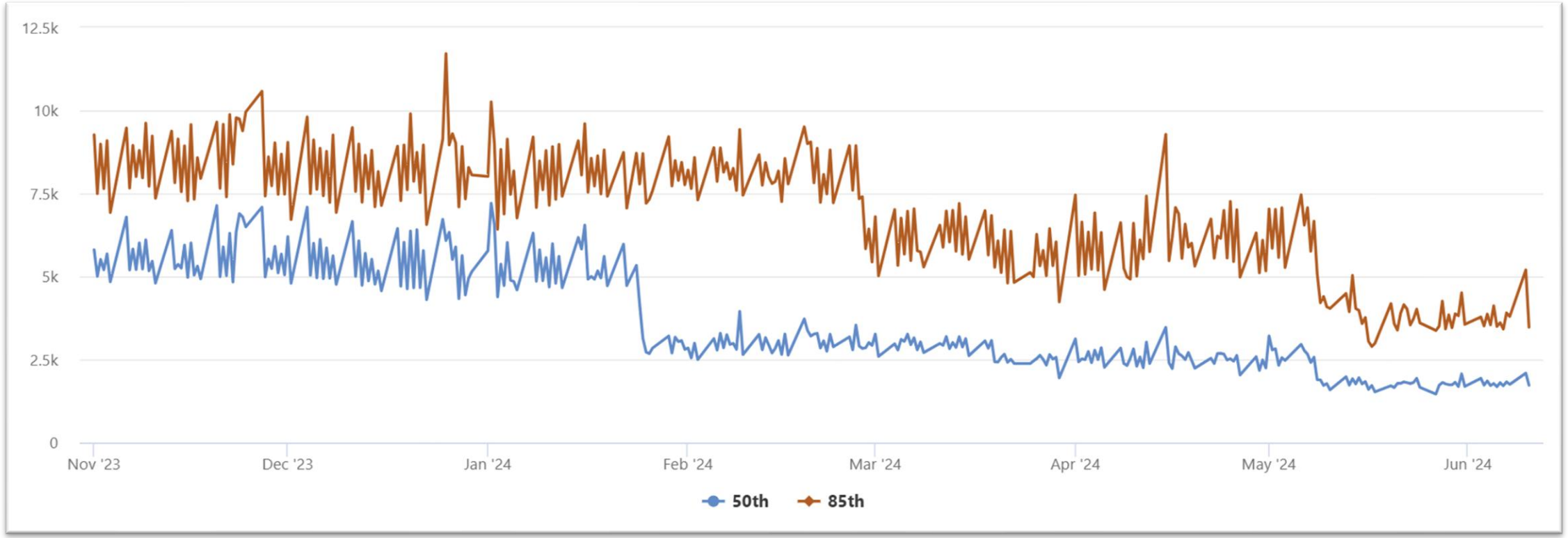
Delay loading large payloads

Check telemetry

Don’t rely only on local testing

What to watch for

- Don’t assume network calls are fast
  - Local != Production
  - AzureDevOps != Azure
  - North America != India --Try binning data by 12h instead of 1d



Azure DevOps

Thank you

Dave Paquette
Principal Software Engineer - Microsoft

Microsoft