Memfault

Boosting IoT Product Performance and Quality with Device Reliability Engineering

François Baldassari
CEO & Co-Founder, Memfault

François Baldassari

CEO / Co-Founder, Memfault

- Passion: tooling and automation in software engineering
- Previously a Firmware Engineer @ Pebble, Oculus, Sun Microsystems
- Can find my thoughts and content on Memfault's Interrupt blog (interrupt.memfault.com)

pebble Memfault oculus

Hardware - it works

It wasn’t always the case!

How did we get there?

Quality engineering:

- Stringent materials requirements and inspections
- Characterization
- Tools
- Processes
- Management focus
- Quality assurance checkpoints
- Root-cause analysis and fast iteration

IoT = New Problems

Families are LOCKED OUT of or INSIDE their homes as [redacted] 'smart' security app crashes leaving dozens stranded

- Households up and down the UK were unable to lock or unlock their doors

[redacted] customers were also unable to turn their alarms on or off with sirens wailing

[redacted] Smart Home Living App was down for more than 24 hours due to glitch

- **Have you been affected by the [redacted] crash? Email lara.keay@mailonline.co.uk**

Software is a major driver of defects in IoT devices

- No “input qualification”: lots of vendor code is unaudited and provided “as-is”
- High complexity: connectivity, OS, and other layers are 10^6 lines of code or more!
- Performance less well characterized than e.g. that of a given aluminum alloy

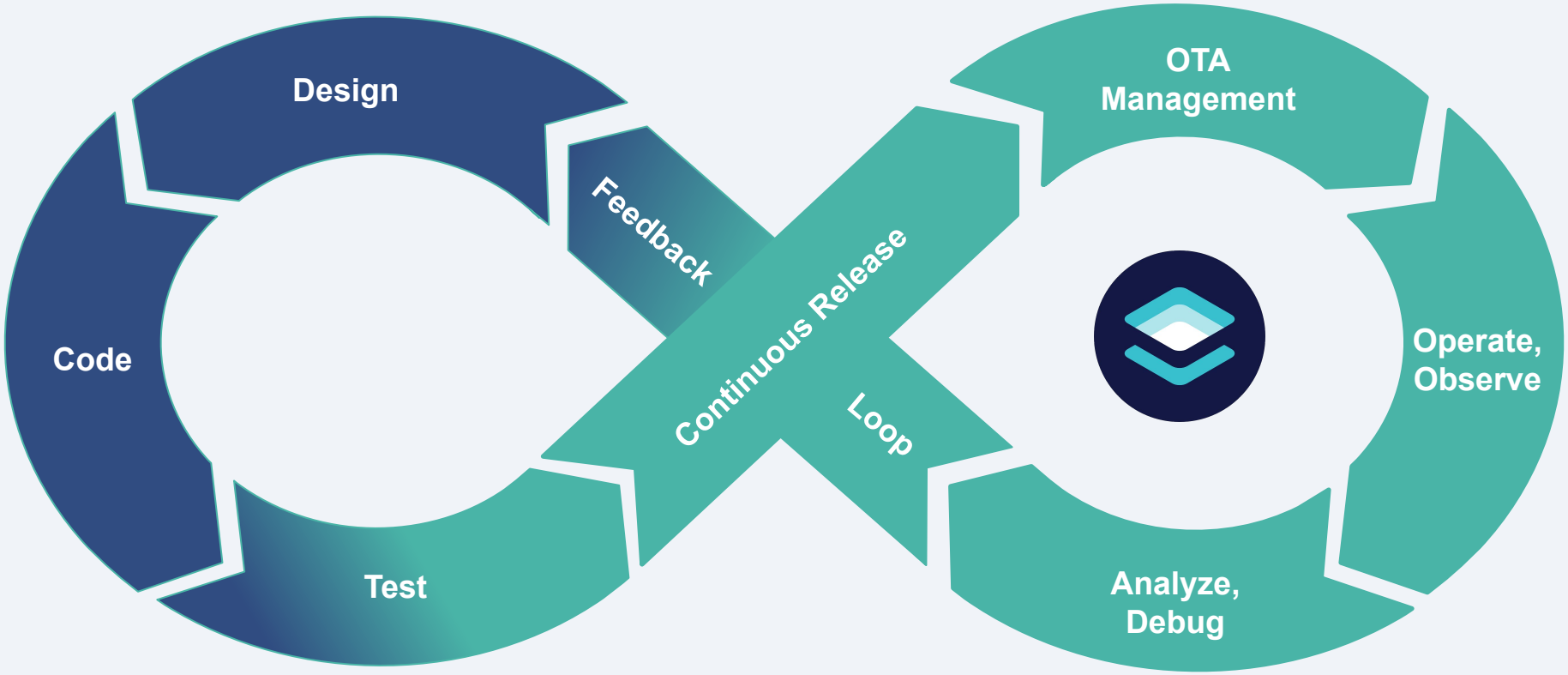
👉 At Pebble, we found that software issues were driving the majority of RMAs

It does not have to be this way!

Software does not rust, decay, wear out, or yield.
It should be our most reliable component!

Make software the most reliable part of the IoT.

Device Reliability Engineering



Improving quality with DRE

Robust OTA & DeviceOps

Performance Metrics

Remote Debugging

Poll #1

Over-the-Air Updates

Software can be improved in the field

Unlike hardware components, software can be improved in the field

This greatly accelerates your rate of iteration

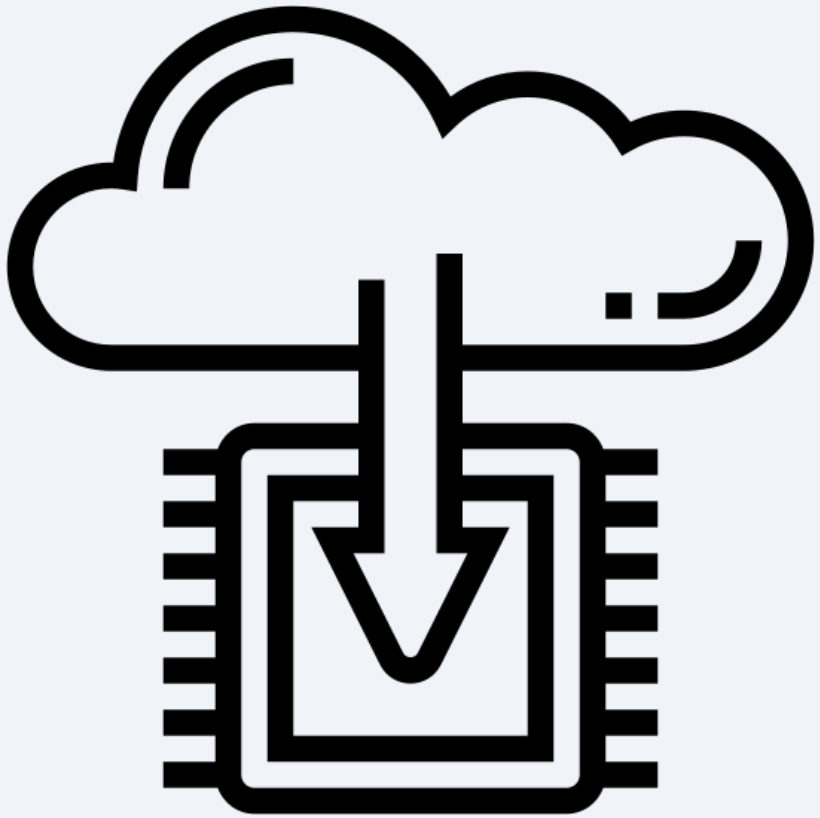
No need to wait for the next manufacturing run!

Robust OTA

OTA is your insurance policy against issues

It needs excellent test coverage!

At the very least, your system should support **cohorts**, **staged rollouts**, and **must-pass-through releases**



Cohorts

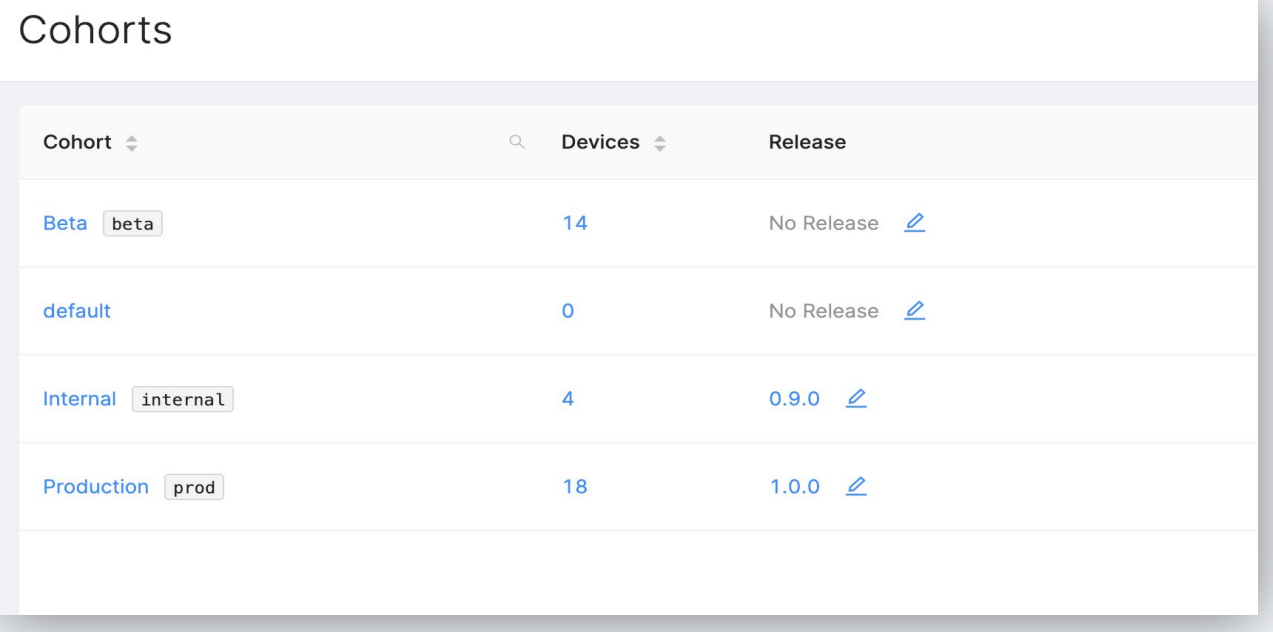
What it is

Grouping your devices, and updating each group separately.

Why You Need It

Enable beta tests, A/B tests, and other forms of experimentation for methodical & low-risk release management.

Cohorts with Memfault:



Staged Rollouts

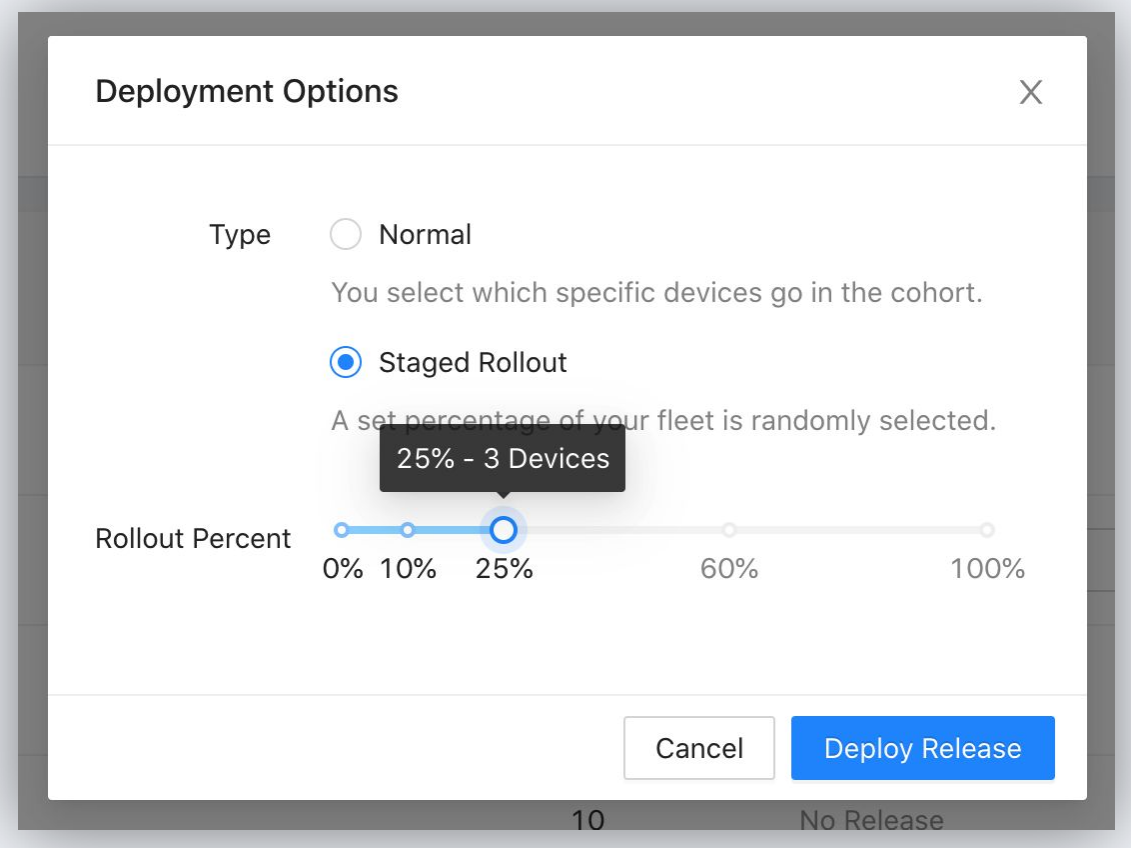
What it is

The ability to roll out a new release to an incrementally larger subset of the fleet.

Why You Need It

Every release introduces risk. By rolling out updates incrementally, you limit the blast radius of any new issue that comes up.

Staged rollouts with Memfault:



Must-Pass-Through

What it is

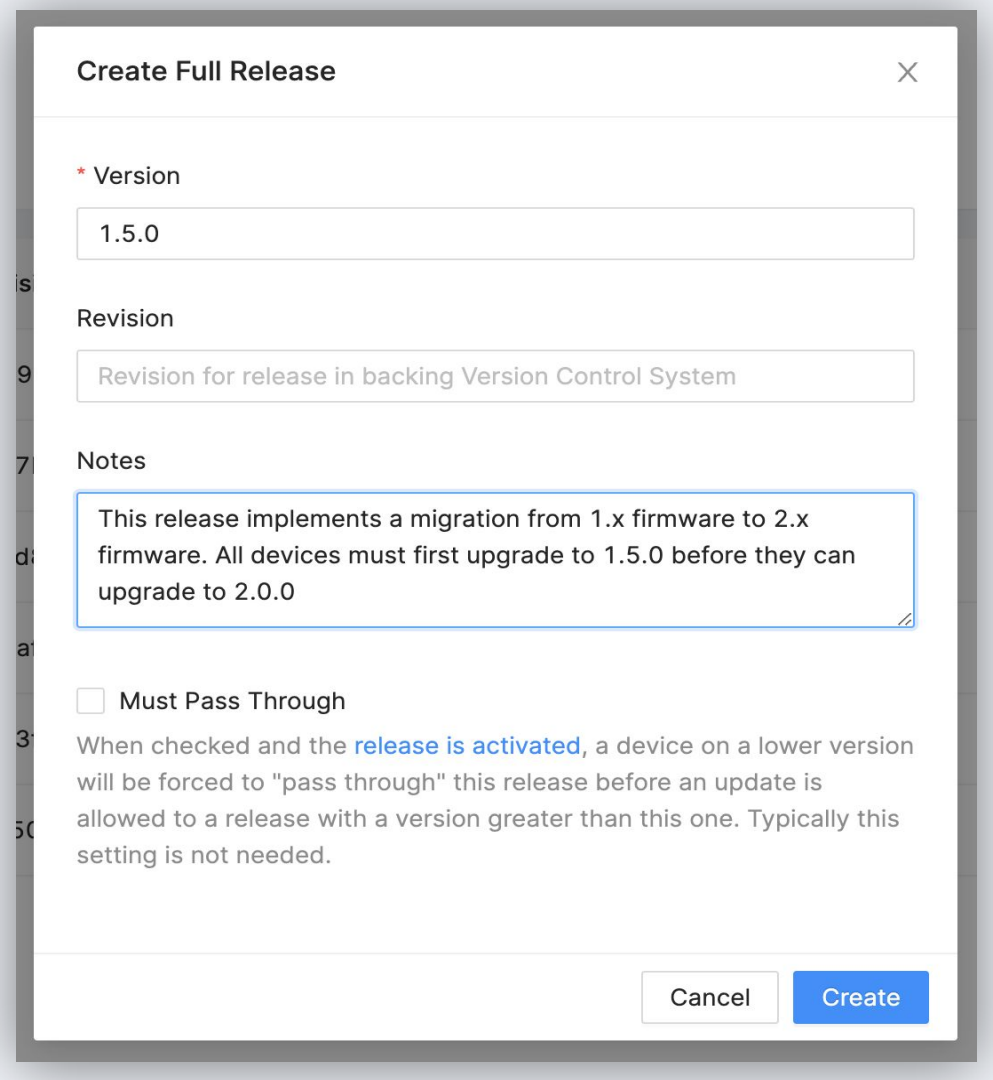
A release which must be loaded on the device before future releases can be installed.

Why You Need It

Gives you control over the upgrade path for complex migrations that may not be forward-compatible.

Ex. upgrading from 1.2 to 3.8 might require multiple steps:
1.2 → 2.0 → 3.0 → 3.8

Must-pass-through with Memfault:



Performance Metrics

Performance Metrics

"How are my devices doing?"

- Connectivity
- Battery Life
- Memory Usage
- Sensor Performance
- System Responsiveness

This system must be:

1. Low overhead (no device impact)
2. Easy to extend
3. Privacy preserving

Individual Device Metrics

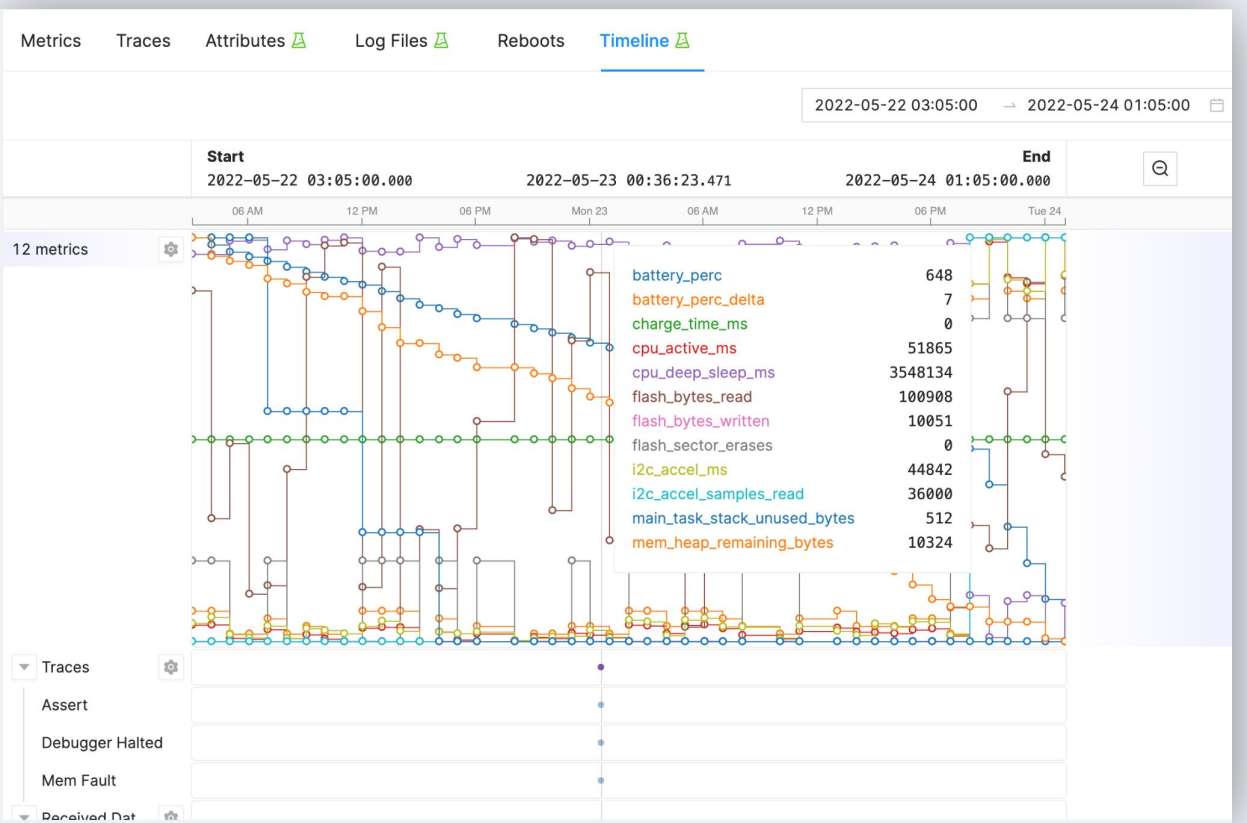
What it is

Collection of data points from devices at regular intervals.

Why You Need It

For customer support or engineering teams to investigate specific reports of devices misbehaving.

Device Metrics with Memfault:



Aggregates and Dashboards

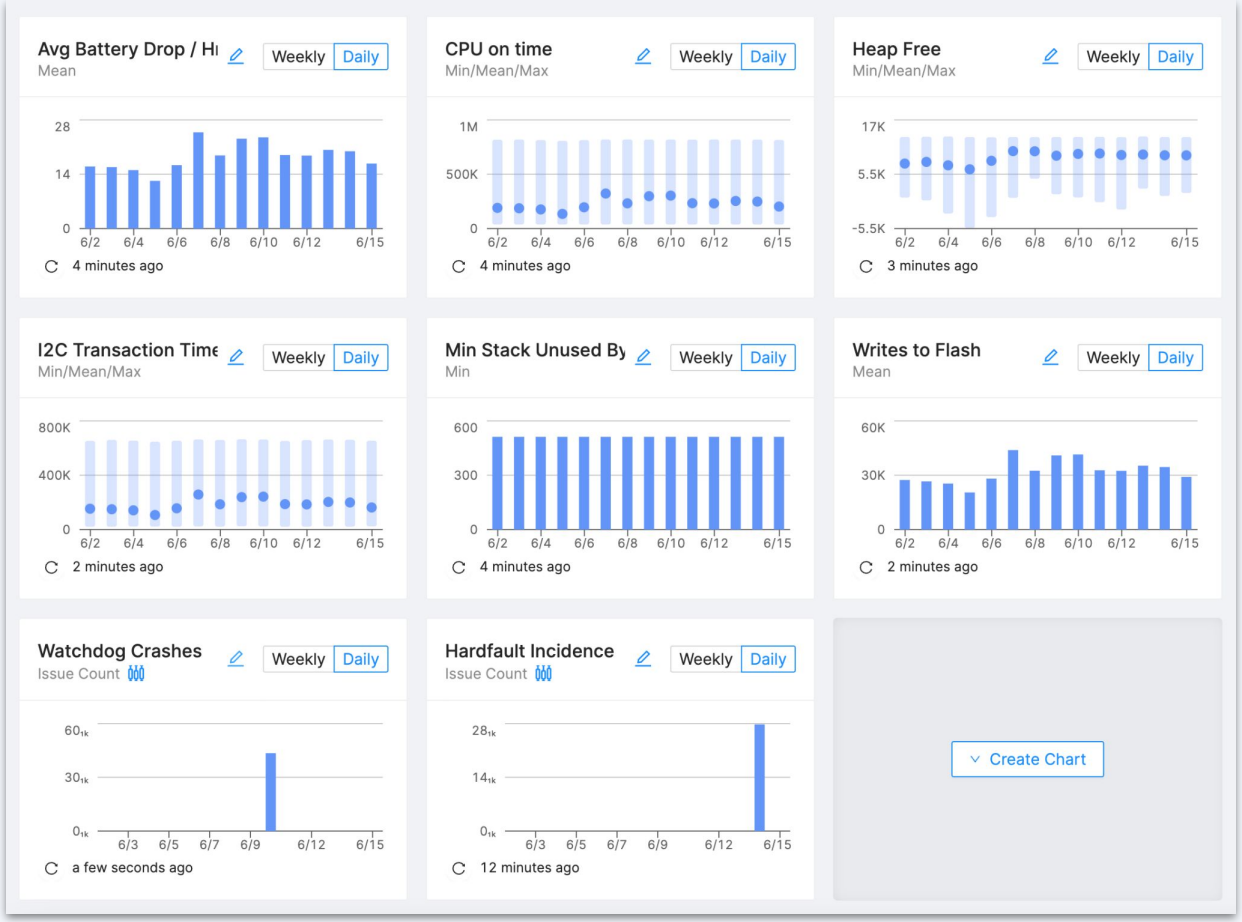
What it is

Dashboards aggregating individual data into high-level charts.

Why You Need It

To understand overall fleet performance and quickly identify trends in the data.

Dashboards with Memfault:



Alerts

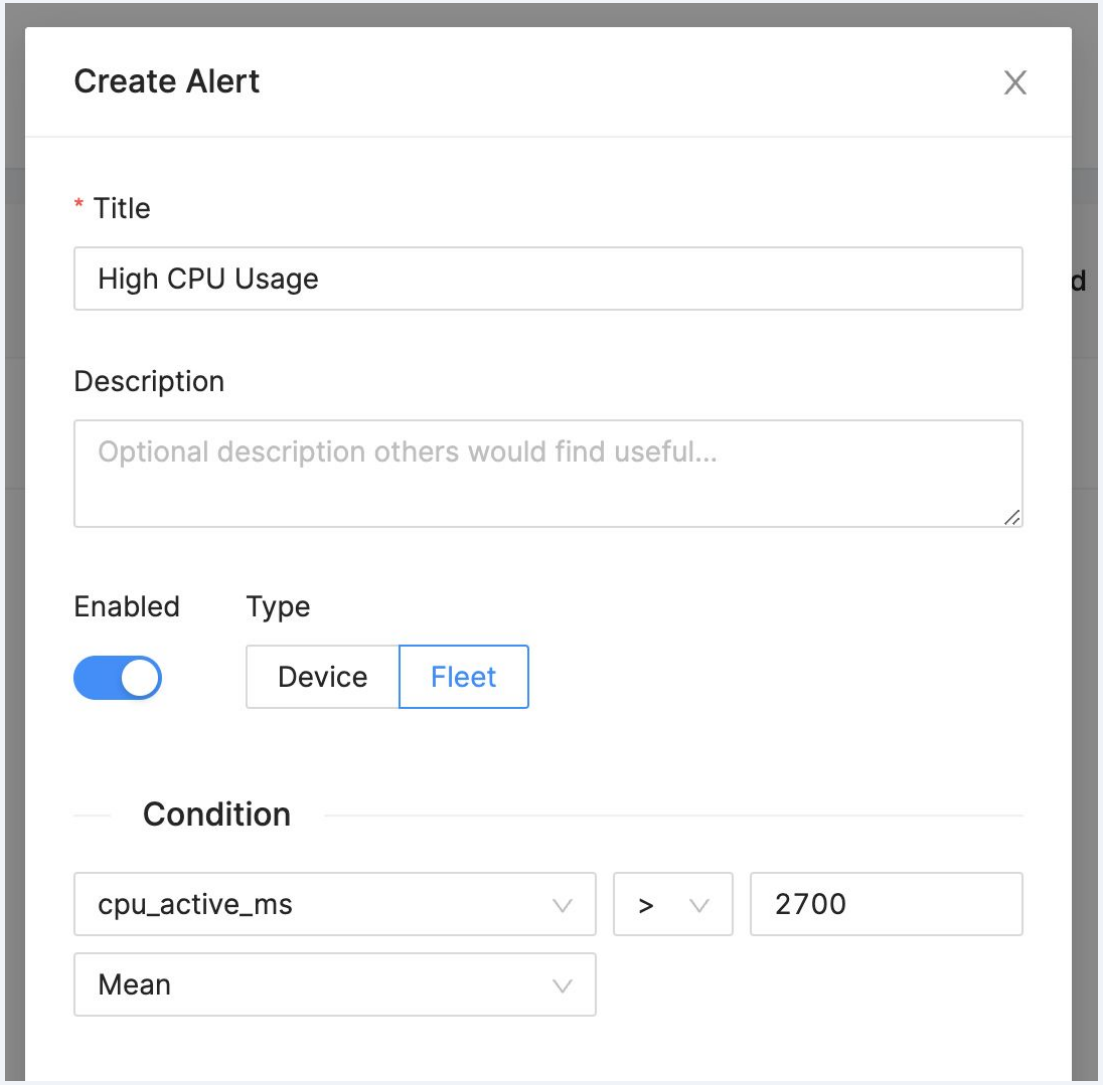
What it is

Alerts to email, slack or incident management platforms when certain conditions are met

Why You Need It

To bring issues to your team's attention quickly, so you can act fast rather than wait for a team member to view the charts.

Alerts with Memfault:



Case Study: Silvertree

silvertree

Challenges

- Write-worn, battery powered safety device for older adults
- Complex trade-offs in design between LTE, GPS, and CPU on time vs. battery life

Results

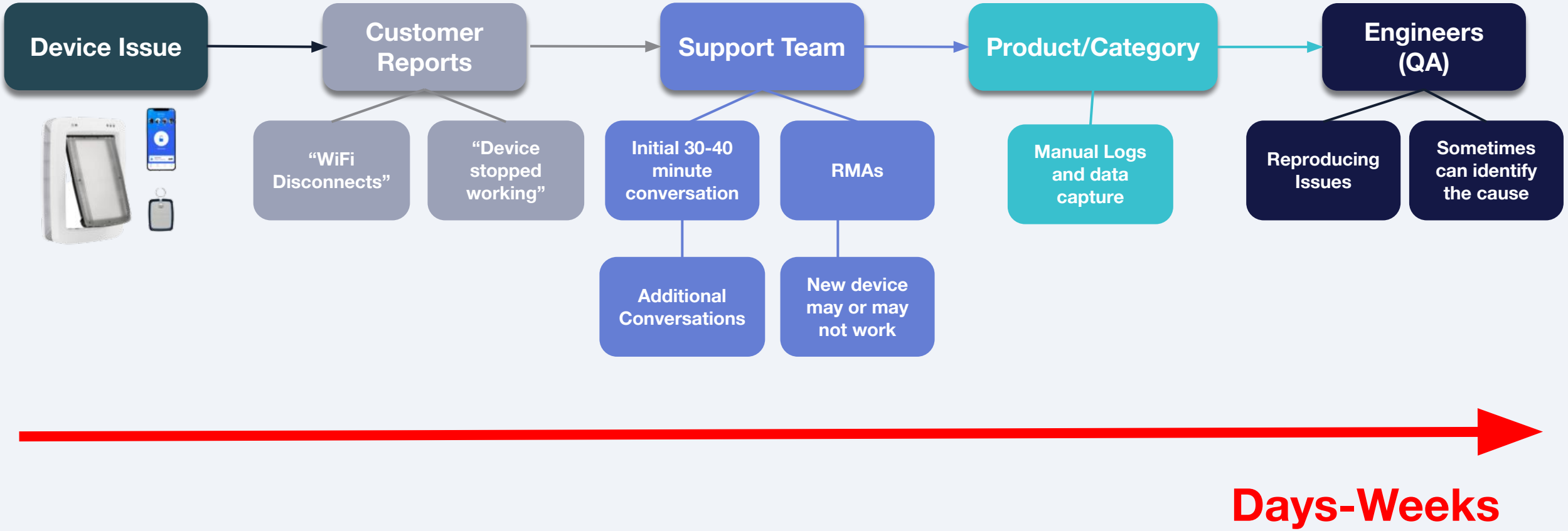
- Monitoring battery level, memory usage, and other metrics enabled Silvertree to make product decisions on key features, like **GPS**
- Data collected by Memfault revealed that Silvertree could reduce GPS on-time by **50%**, saving battery but without jeopardising performance.

> ***To this date, integrating Memfault was probably one of the best decisions I've made for this company because it saved us so many headaches and so much time.***

Konstantin Klitenik
Head of Engineering

Remote Debugging

Traditional methods slow & labor intensive



… and fail to catch 1/10000h bugs

- Bug occurs once every 10,000 hours
- Takes **416 days** to see it on a single device

*With 10,000 devices that issue is hit **every hour***

*With 1 million devices…**every 36 seconds***

Catastrophic issues might *never* be seen during internal testing

Coredumps

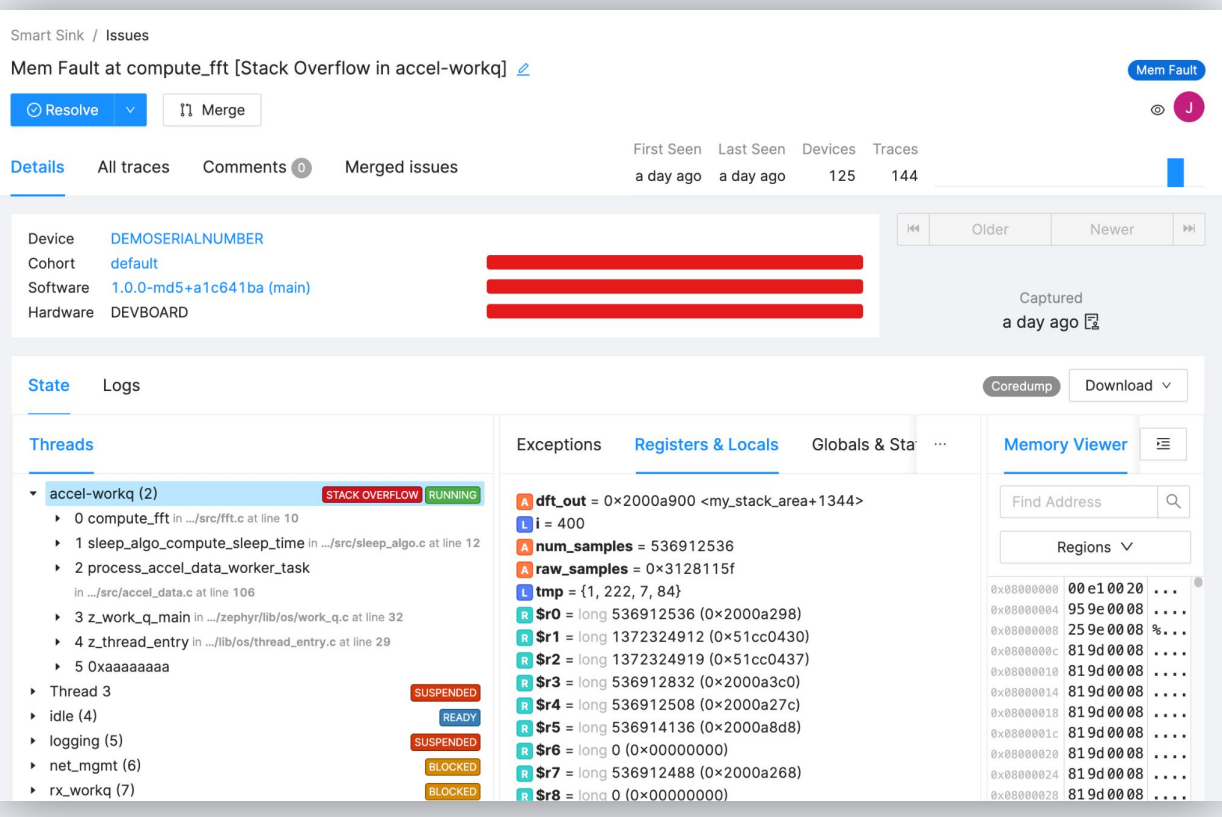
What it is

Automatically collect detailed diagnostic data as soon as an issue occurs.

Why You Need It

Give your engineers the information they need to resolve the problem quickly, without an RMA or sending out a technician.

Coredumps with Memfault:



Case Study: Diamond Kinetics

Challenges

- No crash logs or stack traces on older smart ball and bat products
- Support tickets had minimal detailed information, making it difficult to reproduce the issue locally
- All devices had to be updated at once to fix an issue on a device

Results

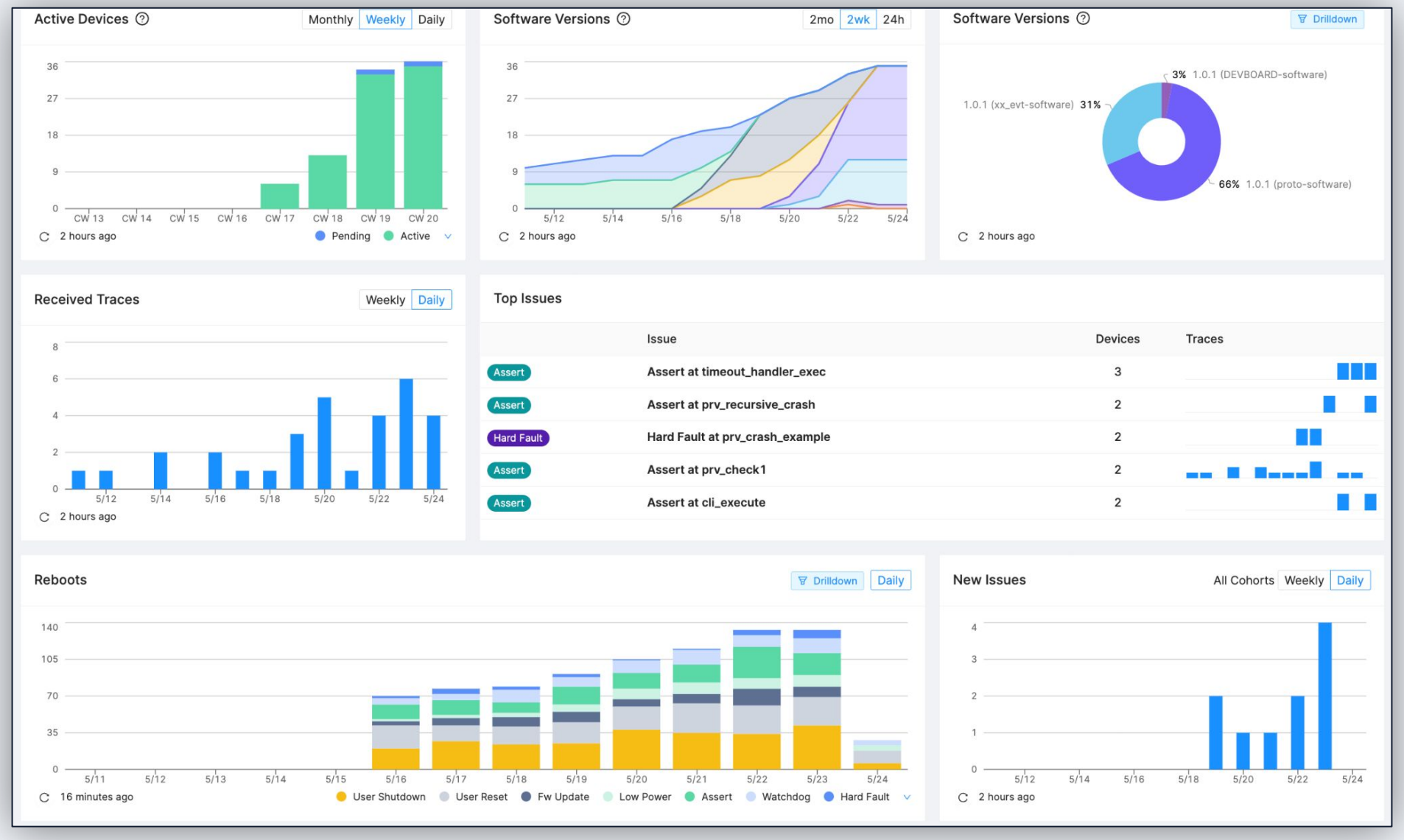
- Reduction in resets per device by **90%** in six months
- **Fixed #1 crash**, a connectivity bug in Bluetooth LE chip
- Faster time to market by accelerating the development cycle

> ***Every step of the way, the Memfault SDK has been reliable and easy to integrate. I wouldn't go to market on an IoT device without Memfault in place.***

Mike Ressler
CTO

Poll #2

Memfault: IoT Reliability Platform



Memfault