Appfire Flow

# Beyond DORA

## Find the signals that predict delivery performance




appfire

# Contents


## Part 1: Why DORA alone isn't enough

DORA metrics show delivery problems without explaining them ..... 4

Why lagging metrics can't drive leading decisions ..... 5

## Part 2: The leading indicators behind each DORA metric

Deployment frequency ..... 8

Lead time for changes ..... 8

Change failure rate ..... 8

Mean time to restore service ..... 8

## Part 3: Where to start

The four phases of your Flow metrics journey ..... 10

How to use this guide with your team ..... 11

Part 1

# Why DORA alone isn’t enough

# DORA metrics show delivery problems without explaining them

## Deployment frequency drops. Lead time increases. Your Change failure rate spikes.

DORA metrics became the industry standard because they provide a consistent way to measure software delivery performance. They help engineering organizations track reliability, speed, and operational stability over time.

DORA metrics reflect outcomes after delivery performance has already shifted.

A change in Deployment frequency, for example, may trace back to slower reviews, shifting sprint priorities, or interruptions that reduced engineering focus weeks earlier.

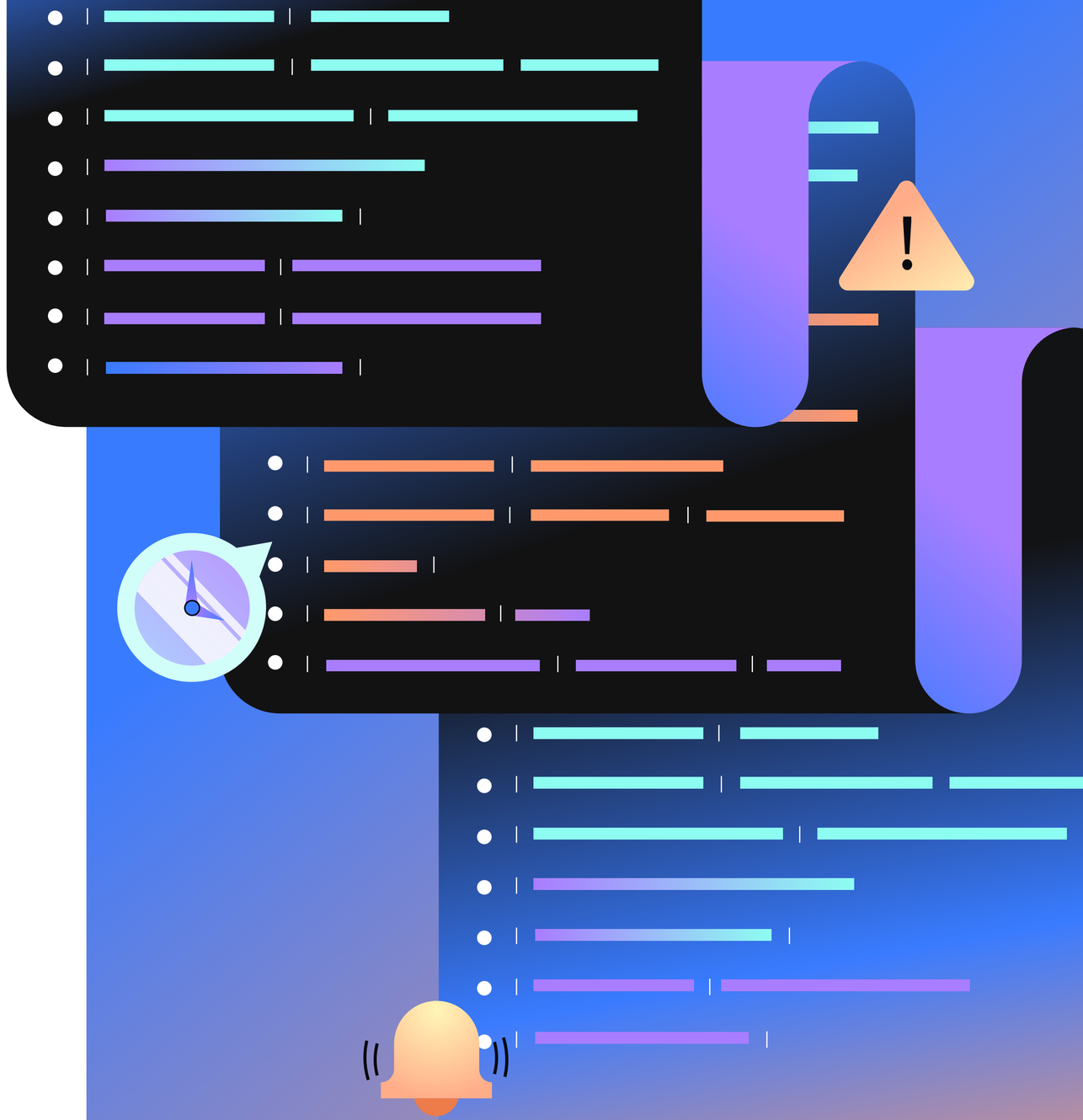
The pattern repeats across the other DORA metrics. By the time performance changes appear in reporting, teams are often working backward to reconstruct what happened, which slows decision-making.

Teams investigate the metric, test a likely fix, and wait to see whether performance improves. Meanwhile, new bottlenecks emerge elsewhere in the workflow.

> **Without earlier operational signals, teams spend more time reacting to performance changes than preventing them.**

Leading indicators help close that gap. They provide visibility into the workflows, delivery patterns, and collaboration behaviors influencing DORA outcomes before those outcomes decline.

This guide maps the leading indicators connected to each DORA metric and shows how to use them to spot delivery friction earlier.



# Why lagging metrics can't drive leading decisions

## DORA metrics measure four outcomes:

- **Deployment frequency** – how often you ship
- **Lead time for changes** – how long code takes to reach production
- **Change failure rate** – how often deployments cause issues
- **Mean time to restore service** – how fast you recover

These are powerful signals. But they're downstream. By the time a DORA metric shifts, the root cause happened days or weeks earlier in your commits, reviews, planning, or prioritization decisions. That creates a familiar loop:

- A metric dips, you investigate, you fix something … you wait
- And the cycle starts again

Leading indicators break that loop. They give you visibility into the habits, workflows, and decisions shaping your outcomes so you can act while there's still time to change them.

In Appfire Flow, these signals map to four phases of maturity:

| Phase | Focus | What it reveals |
|---|---|---|
| Activity | Commit patterns and coding consistency | Whether your team has a stable engineering rhythm |
| Collaboration | PR workflows and knowledge sharing | Whether quality and resilience are sustainable |
| Planning and delivery | Throughput, Cycle time, Sprint completion rate | Whether delivery is predictable |
| Portfolio optimization | Investment alignment and outcomes | Whether delivery translates to business impact |

Part 2

# The leading indicators behind each DORA metric

# The leading indicators behind each DORA metric

In Appfire Flow, each DORA metric maps to a primary phase where its strongest leading indicators appear. This section focuses on those core signals so teams can identify delivery friction earlier and take action sooner.



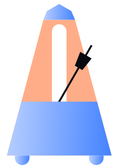
| | Deployment frequency | Lead time for changes | Change failure rate | Mean time to restore service |
|---|---|---|---|---|
| What it measures | How often your team deploys | Time from commit to production | % of deployments causing issues | How fast you recover |
| Why it matters | Smaller, more frequent changes reduce risk and speed up feedback. | Lead time reveals where your system is slowing down. The core drivers of lead time sit in collaboration, especially how quickly work is reviewed, discussed, and approved. | Speed without quality creates risk. | Recovery speed determines the cost of failure.<br><br>Recovery speed depends on how quickly your system can respond — not just during incidents, but in everyday delivery. |
| Leading indicators to watch | **Activity phase**<br>• **Coding days** — Consistent coding activity supports steady delivery<br>• **Commits per day** — Smaller, steady commits help teams ship more consistently<br>• **Impact** — Meaningful commits drive measurable progress<br>• **Efficiency** — Rework slows delivery over time | **Collaboration phase**<br>• **Time to first comment** — Delays add directly to lead time<br>• **Time to merge** — The most direct lever to improve speed<br>• **Iterated PRs** — Signals clarity vs. rework<br>• **Review distribution** — Bottlenecks slow everything | **Planning and delivery phase**<br>• **Efficiency** — Rising rework increases delivery risk<br><br>**AI adoption signals**<br>• **Code acceptance rate** — Healthy range: 26-35%. Higher rates often signal unreviewed AI output and increased risk.<br><br>This is where teams mistake output for impact during AI adoption. | **Planning and delivery phase**<br>• **Velocity** — Teams ship fixes faster when incidents occur<br>• **Cycle time** — Shorter cycles speed up fixes<br>• **Throughput** — Teams that ship more recover faster |

Part 3

# Where to start

# The four phases of your Flow metrics journey

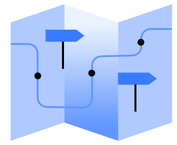
Appfire Flow's four phases help teams improve delivery in a sequence that builds consistency over time.



**Activity** focuses on engineering rhythm and commit consistency



**Collaboration** focuses on review quality and knowledge sharing



**Planning and delivery** focuses on predictability and delivery flow



**Portfolio optimization and scale** connects engineering outcomes to business priorities

The most effective transformations build progressively across activity, collaboration, and planning and delivery before shifting attention toward portfolio-level optimization.

# How to use this guide with your team

## Start with activity metrics first, even if your team's immediate focus is delivery performance.

Teams often focus on delivery outcomes before the underlying workflows are stable. Appfire Flow's four phases help teams build consistency step by step, using different layers of engineering data to understand where delivery patterns are improving or slowing down.

### 1. Start with activity metrics and Deployment frequency

**Focus on commit activity signals:**

- **Coding days**
- **Commits per day**
- **Impact**
- **Efficiency**

These metrics reflect the consistency of day-to-day engineering activity and often provide the earliest signal that delivery patterns are shifting.

If Deployment frequency is low, start here first.

### 2. Move into collaboration metrics and Lead time for changes

**Focus on pull request and review signals:**

- **Unreviewed PRs**
- **Time to first comment**
- **Time to merge**
- **Iterated PRs**
- **Thoroughly reviewed PRs**

This phase helps teams improve review responsiveness, reduce bottlenecks, and spread knowledge more consistently across the organization.

When Lead time or Change failure rate starts rising, collaboration patterns often explain why.



## 3. Expand into planning and delivery metrics and Mean time to restore service

**Focus on delivery and workflow signals:**

- **Throughput**
- **Cycle time**
- **Sprint completion rate**
- **Velocity**

These metrics help teams improve predictability and identify where delivery work begins to stall.

If delivery remains inconsistent after collaboration improves, planning and delivery workflows are usually the next area to examine.

## 4. Use portfolio optimization and scale to connect engineering outcomes to business priorities

**Focus on portfolio and organizational signals:**

- **Executive summary**
- **Investment profile**
- **Developer sentiment**
- **DORA metrics**
- **Proficiency**
- **Scenario Builder**
- **AI adoption visibility**

At this stage, teams can evaluate whether improvements in the earlier phases are translating into stronger delivery outcomes across the organization.

The strongest transformations build progressively across activity, collaboration, and planning and delivery before shifting attention toward portfolio-level optimization.

Over time, the focus moves from reacting to delivery problems to improving the systems and behaviors influencing them.



# Next steps

You've seen how leading indicators help explain the story behind your DORA metrics. The next step is seeing how those signals appear in practice across **reviews, delivery flow, planning, and AI adoption**.

The earlier your team can spot friction, the faster you can improve delivery performance before small issues become expensive ones.

See Appfire Flow in action






Experience Flow in action appf.re/flow-demo