

Where your engineering time really goes

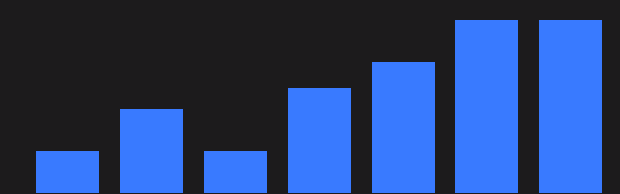
(and what to do about it)

Team health insights

Deployment frequency

3.4
Average

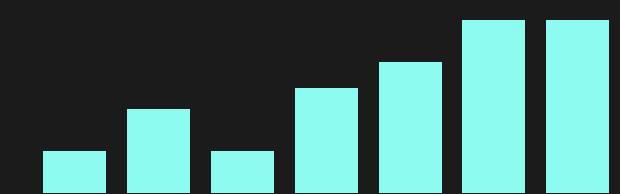
5.0
Target



Reviewed PRs

22%
Average

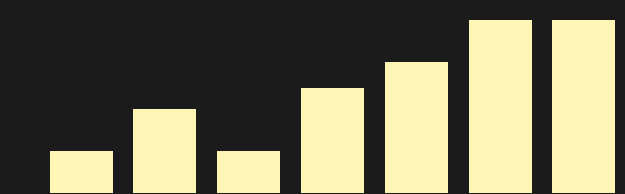
18%
Target



Time to merge

21
Hours

24h
Target



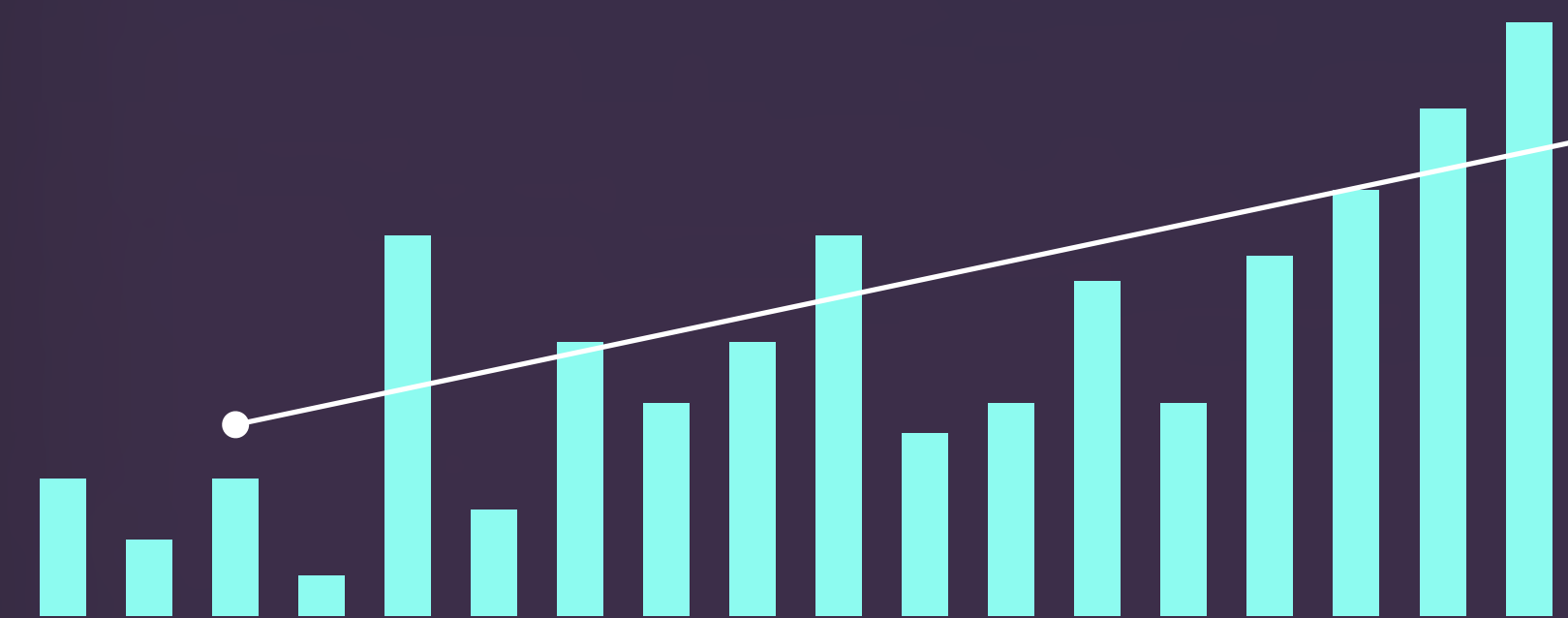
Weekly impact

Average

231

Trend

+23%



Introduction

Software delivery is anything but linear. From concept to production, it moves across teams, priorities, and timelines, with constant collaboration along the way.

The features your team ships today could drive real revenue tomorrow. But not all impact is immediate. Sometimes, a small bug fix matters more than a big feature, especially when it affects user trust or retention.

In fast-moving orgs, leaders need to make smart decisions about where to invest engineering time. That means understanding not just what’s getting done, but what it’s worth. Team leads need to justify their teams’ work to three audiences: leadership, finance, and developers themselves. Everyone wants proof that time is being spent in the right place.

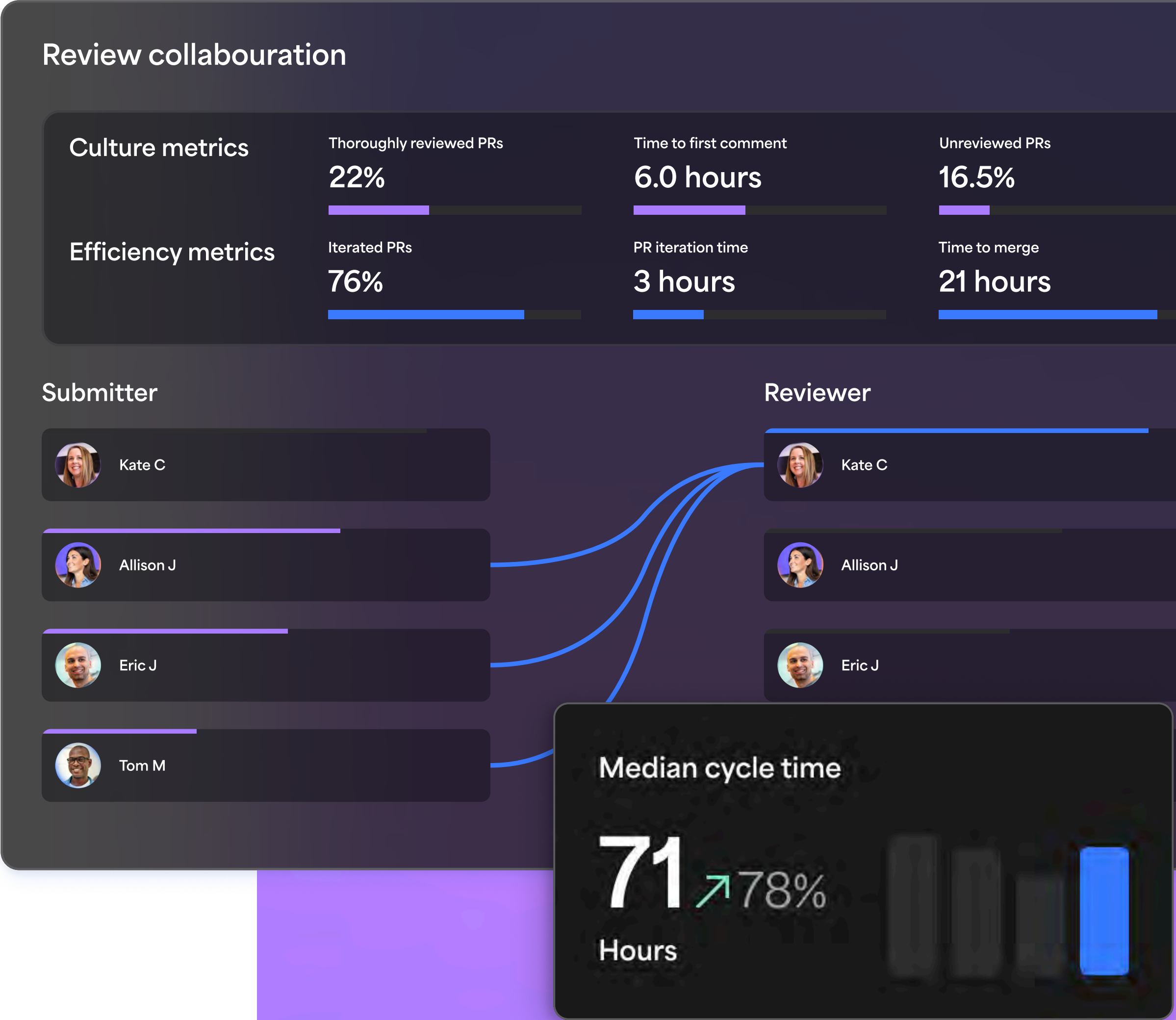
That’s where Flow’s **Investment Profile** comes in.

This capability gives you visibility into where engineering effort is going, so you can advocate for your team, align with strategic priorities, and back it all with data. For finance and accounting, **Investment Profile** pairs easily with salary data to show the actual cost of engineering resources. No more digging through spreadsheets or estimating time by gut feel.

Whether you’re a manager, a CTO, or an accountant, you need the same thing: clarity. Flow lets you view work allocation by team, individual, epic, or timeframe, so you can move from “what are we working on?” to “is this work moving the business forward?”

Flow shows how engineering time is spent. Investment Profile helps you decide if that time is well spent.

Let’s explore how this data-driven approach helps engineering leaders allocate more effectively, improve team health, and align effort to impact.



Capitalization vs. allocation

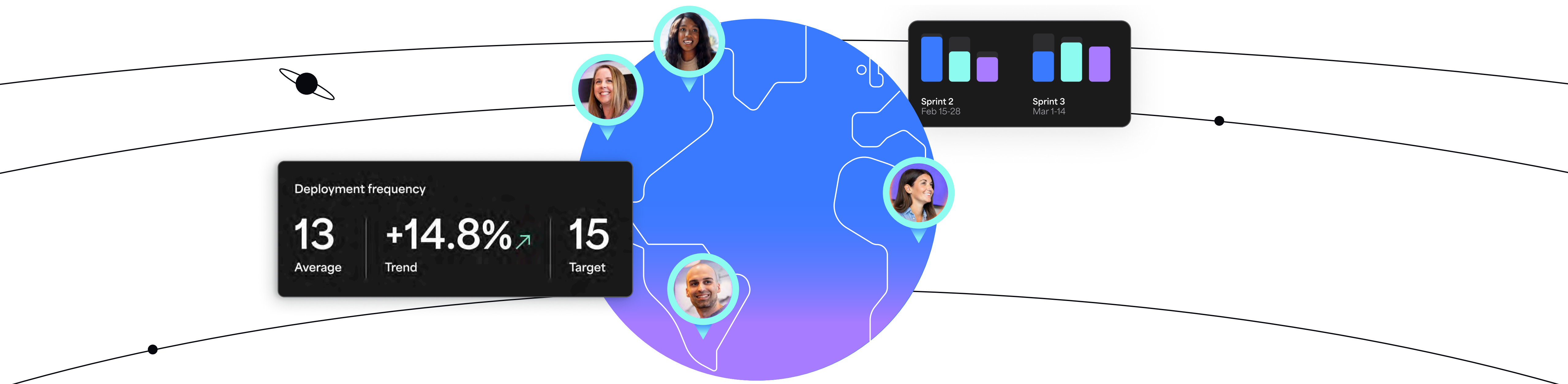
Allocation and capitalization both help you understand where engineering time is going, but for different reasons.

Allocation helps teams manage day-to-day priorities. It answers questions like: Who's working on what? Are we spending time on the right things?

Capitalization, on the other hand, connects engineering work to business value. It helps finance teams track which features and products are driving revenue, and which development efforts can be counted as long-term investments.

Once engineering work is expected to deliver value, accounting teams mark it for capitalization. That step matters. It shapes how companies report costs, forecast returns, and make funding decisions.

And while the process is complex, the goal is simple: understand what's adding value, and invest accordingly.



Allocate across projects to focus on work that matters

Understanding how your team allocates time across epics helps you see what's getting prioritized (and what's not).

But expectations and reality don't always match. You might think your team is focused on feature work, only to realize bugs or maintenance tasks are eating up time. **Investment Profile** helps bridge that gap with data you can act on.

To measure what matters, ask:

What type of work is going into this epic?

How many tickets are being completed over time?

Is this epic truly being prioritized, or is other work taking over?

Most epics start with a clear purpose. A new feature epic might be entirely "feature" work. A bug fix epic should focus on defects. But in practice, things shift. A feature may trigger rework. A fix might spark an improvement. That's normal, but it makes measurement essential.

Tracking how different types of work show up within an epic can reveal hidden blockers, quality concerns, or shifting team priorities. Over time, this insight helps you deliver with more confidence and less chaos.

Let strategy (not urgency) drive priority

Different goals call for different types of focus. Are you racing to beat a competitor to market? Prioritize feature work. Are you working to reduce churn? Then bugs may need to come first. Starting a new team? Maintenance work is a great onboarding path.

But what if two strategic initiatives feel equally urgent?

Ask these four questions to break the tie:

Can I split these epics across two teams? Keeping teams focused avoids context switching and speeds up delivery.

How quickly can my team complete each epic? If one initiative can wrap in a sprint and the other will take a quarter, ship the fast win first.

Which epic aligns better with our long-term strategy? Protect the roadmap. Don't let short-term pressure derail big-picture goals.

What's the business impact of delay? If a high-value customer is waiting on a fix, that's a clear signal to reprioritize.

Don't just guess your top priority: Prove it

It sounds obvious: your most important epic should receive the most effort. But competing work and reactive tasks can quietly pull your team off track.

Investment Profile shows you where story points and tickets are going, so you can verify that your team's time lines up with your top goals.

See what's *really* being prioritized. Use **Investment Profile** to compare epics by story points, ticket count, and time spent.

- Catch misalignment early.
- Refocus effort with confidence.
- Reinforce strategic priorities in every sprint.

Allocate across time to see the full picture

Workload balance isn't static: It shifts over time. To keep priorities aligned, you need to zoom out and see not just what your team is working on, but when.

Maybe your team focused heavily on new features to meet a big customer request. But while that was happening, maintenance got deprioritized. Now your product is lagging, and you're losing other users.

That's why it pays to track how effort is distributed over time across feature work, maintenance, and bug fixes. Patterns reveal whether your team is building sustainably or heading toward burnout or backlog creep.

Track the time balance

Use **Investment Profile** to review how your team's effort shifts week to week or quarter to quarter.

- Catch overinvestment in one area.
- Make space for maintenance before it becomes urgent.
- Spot trends early and rebalance with confidence.

Adapt your plan to shifting priorities

Some priorities need to spike temporarily. For example, if you've got a backlog of critical bugs, that might be your team's focus for a sprint or two. But don't let urgent needs crowd out long-term goals, like developing a feature your roadmap depends on.

When you review effort over time, you can validate that urgent work isn't starving strategic work. You can also adjust in real time, not just in retros.

Use the past to forecast the future

Historical data can do more than tell you where time went: it helps you plan where it should go next.

Knowing how long your team typically spends on different types of work helps you:

- Estimate future delivery timelines more accurately.
- Set more realistic sprint goals.
- Scale the team based on actual performance, not guesswork.

You can even define a healthy mix of work, like allocating 60% to features, 25% to maintenance, and 15% to bugs, then monitor whether that mix holds up in practice.

Turn past sprints into future insights

Use historical work data to set smarter goals, balance workloads, and prioritize with precision.

- Know your team's capacity
- Build realistic plans
- Adjust without overloading

Allocate across teams to improve focus and collaboration

It makes sense that every engineering team has its specialties. But when work stays siloed too long, it can limit collaboration and slow delivery. The fix? Introduce more variety and visibility across teams.

When teams take on a mix of work – not just feature delivery but bugs, maintenance, and reviews – they upskill passively, share the load, and ship faster. This cross-team balance boosts delivery velocity, improves team morale, and helps reinforce a culture of shared ownership.



We start to work together on something that we both care about, and we can better understand our roles and what makes us a team as opposed to what separates us.”

Dr. Tracy Brower
Sociologist and author

The delivery gains are real

When multiple teams contribute to feature work, bugs, and pull requests, two things happen:

- Releases move faster.
- Support work gets cleared more quickly.

That means fewer backlogs, shorter queues, and smoother roadmaps.

Having each team focus on one release at a time also minimizes distractions. Teams stay aligned, execution improves, and every sprint feels more purposeful.

Find the right balance between focus and variety

There’s a sweet spot between keeping teams focused and keeping work interesting.

Assigning each team a primary work type, for example, features, bugs, or maintenance, reduces context switching and builds momentum. But over time, monotony can lead to disengagement.

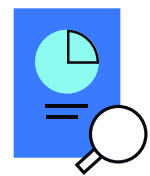
This is where time-based planning comes in. Use short, focused time windows. Let teams complete one type of work, then rotate to the next. It keeps the work fresh and the team energized.

A key benefit of Investment Profile

Get clarity on whether teams are working on the most strategic initiatives, or if reactive work is taking over

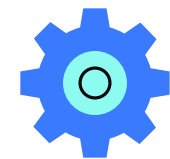
- Eliminate silos
- Stay aligned with strategy
- Catch drift before it spreads

How to allocate across teams



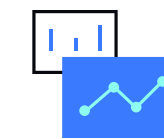
Compare team workflows over time

Look at historical data from past months, quarters, or years



Customize work categories

Use naming that maps to your team's language and strategic concepts



Measure both effort and proportion

Track ticket count, story points, and percentage breakdown by work type



Review work by epic

See which epics each team is contributing to, and how much time is going where

Allocate across individuals to improve experience and output

Happy developers ship better software. When individuals are supported, challenged, and protected from burnout, teams move faster, quality improves, and delivery becomes more predictable.

Work allocation plays a big role in that. When done right, it balances expectations, reduces overload, and gives everyone space to grow.

Spot burnout before it happens

Investment Profile helps you see who's doing what, and how much. If one engineer is handling significantly more tickets or story points than their peers, that's a signal. It could mean they're at risk of burnout, or that your team's workload isn't being shared fairly.

The sooner you catch the imbalance, the faster you can rebalance effort and avoid long-term fatigue.

Burnout alert

Use **Investment Profile** to compare effort across engineers by ticket volume and story points.

- Identify overextended contributors.
- Adjust workloads quickly.
- Protect team health and retention.

Mix it up to keep people engaged

Variety matters as much as volume. Monotonous work leads to disengagement. A mix of projects, tasks, and responsibilities keeps developers interested and accelerates upskilling.

Of course, some tasks require senior engineers, but that doesn't mean junior team members should be sidelined. When everyone has a chance to contribute, they stay connected to the bigger picture, and learn faster.

Give people meaningful work

Developers thrive when they're trusted with work that matters. That means giving them opportunities to build features and improve products, while also sharing ownership of less glamorous tasks like fixing bugs or reviewing code.

When teams understand how their work ladders up to company goals, engagement goes up, and so does delivery speed.

Allocate with ease

Great allocation starts with visibility. Without it, even the best strategy gets buried in spreadsheets or lost across tools.

That's why **Flow's Investment Profile** brings everything into one place. You can see work allocation across individuals, teams, time, and epics – all in a single, unified view.

It's more than just convenience. It's clarity that helps you act with confidence.

One report, every perspective

See how work is distributed, spot gaps quickly, and shift focus without guesswork

- Save time
- Stay aligned
- Scale smarter

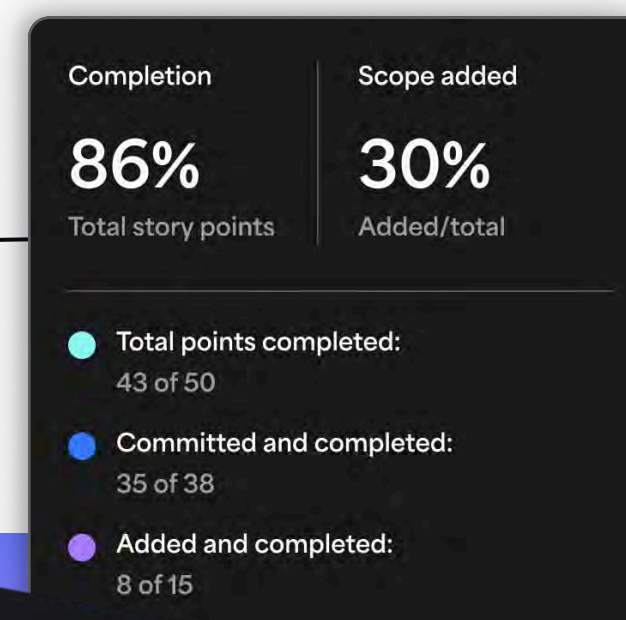
When your allocation is clear:

- Your CTO sees targets hit.
- Your CRO gets clean data for investment planning.
- Your engineers stay engaged with meaningful, varied work.

It's visibility that works for everyone.

Measure what matters, fix what's blocking, deliver what counts.

[See Flow in action](#)





Experience Flow in action appf.re/flow-demo