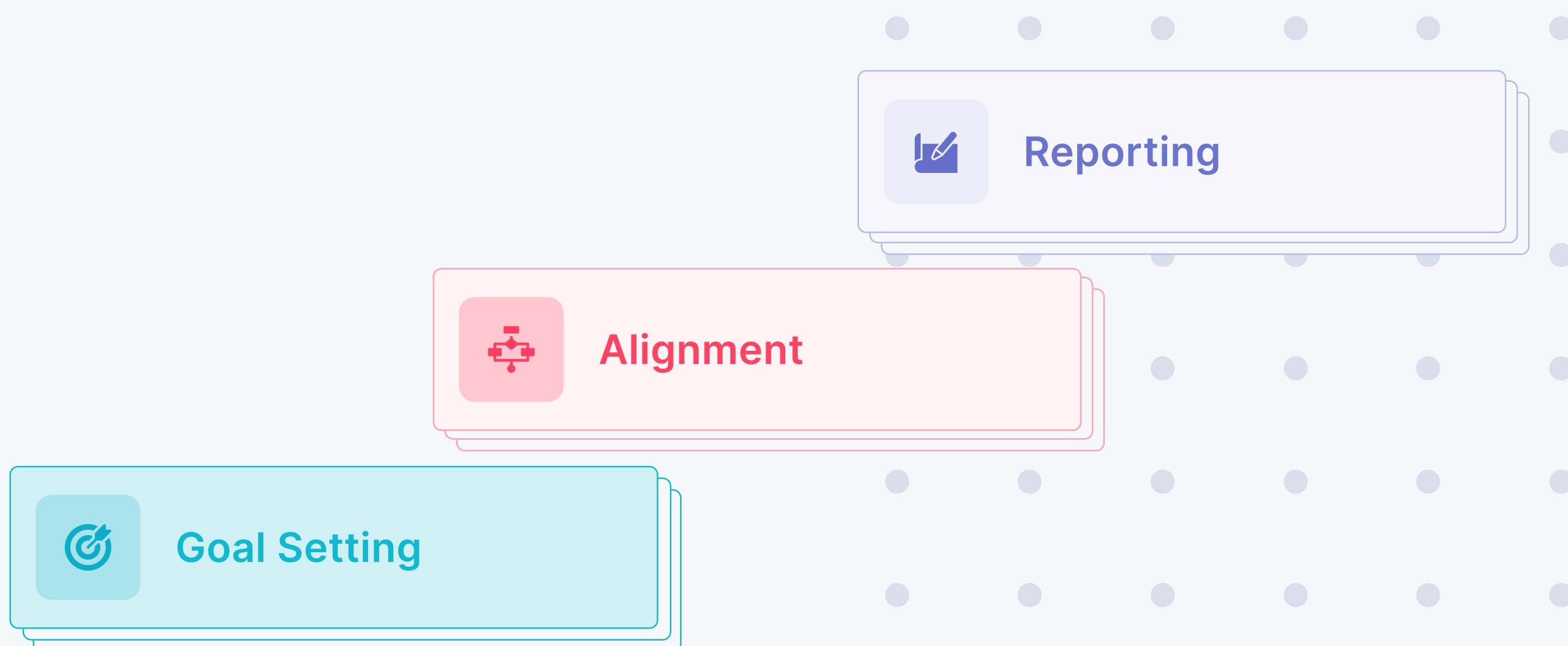




# Engineering Leader's Guide to Goals and Reporting

---

Metrics and benchmarking are foundational steps to improving engineering execution and aligning R&D to business priorities. But engineering leaders cannot stop at visibility if they want to actively improve. The next step is to use this insight to set tangible improvement goals, adopt workflow automation to drive improvement, and align stakeholders by reporting progress against stated goals.



In this guide—part three of the five-part engineering success model—you’ll learn how to set operational-and-business-focused improvement goals and build reporting for specific audiences.

If you're just getting started with your engineering metrics program we recommend beginning with the Engineering Leader’s Guide to **Building a Metrics Program** and **Resource Allocation**. Here’s what we’ll cover in the pages ahead:

- **Goal setting:** Tactical guidance on setting (and achieving) organizational and team goals
- **Alignment:** Bringing engineering stakeholders and business leaders into the fold while ensuring improvement goals/ reporting delivers operational and business outcomes
- **Reporting:** How to standardize your presentation and cadence for communicating goal attainment and improvement results to different audiences



# Table of Contents

- 1. The Duality of Goal Setting for Engineering Leaders**
- 2. How to Set Engineering Improvement Goals**
  - Start with Organization Level OKRs
  - Move to Team-Level OKRs
  - Keep Everyone Aligned to Goals with Dashboards and Automation
- 11. Reporting on Goals, Success, and Further Opportunities**
  - Reporting to Sr. Leadership, C-Suite, and other business leaders
  - Reporting to Engineering Organization and Teams
- 17. The universal, measurable impact of goal setting**
- 19. Conclusion**

# The Duality of Goal Setting for Engineering Leaders

Engineering leaders have a dual mandate to: (1) deliver operational excellence (2) drive business impact. This concept permeates every aspect of an engineering leader's remit and working to improve your organization with goal setting is no different. When setting goals, engineering leaders have to look at both business outcomes and operational outcomes and how they cascade into each other.

Setting (and achieving) these goals requires visibility into impact metrics and an understanding of industry standards benchmarks to know how far you have to go and what to shoot for.

Let's dig into the tactical process of setting improvement goals for your engineering organization.

We'll cover this in more detail in a bit, but here is a high-level example of how to set a goal that encompasses both business and operational outcomes and their relationship to each other:

## Holistically improve your engineering practice by aligning goals at every level

### Business outcomes

- Increase ROI with efficiency improvements (presented as either a multiplier or in actual dollar amounts)
- Foster higher attraction/retention of talent with better DevEx
- Deliver more value to the business with predictable delivery of high impact projects

### Org level goals

- Ex. Higher efficiency and quality to drive higher merge frequency
- Ex. Increased focus on high impact work that creates value (like new features)
- Ex. Reduce organizational cycle time by 10% for Q1 FY24

### Team level goals

Are fed by organizational OKRs and are communicated down from leadership (VPEs) to managers to team leads

- Ex. "Team A to reduce team cycle time by 15%" over Q1 FY24
- Ex. "Team B to reduce CFR by 20%" over Q1 FY24



# How to Set Engineering Improvement Goals

Before you ever put pen to paper on your improvement goals, you need visibility and context. If you don't already have an **engineering metrics program** in place, start there. You'll also want to be sure to include **Resource Allocation** visibility (and goal setting) as part of your metrics program.

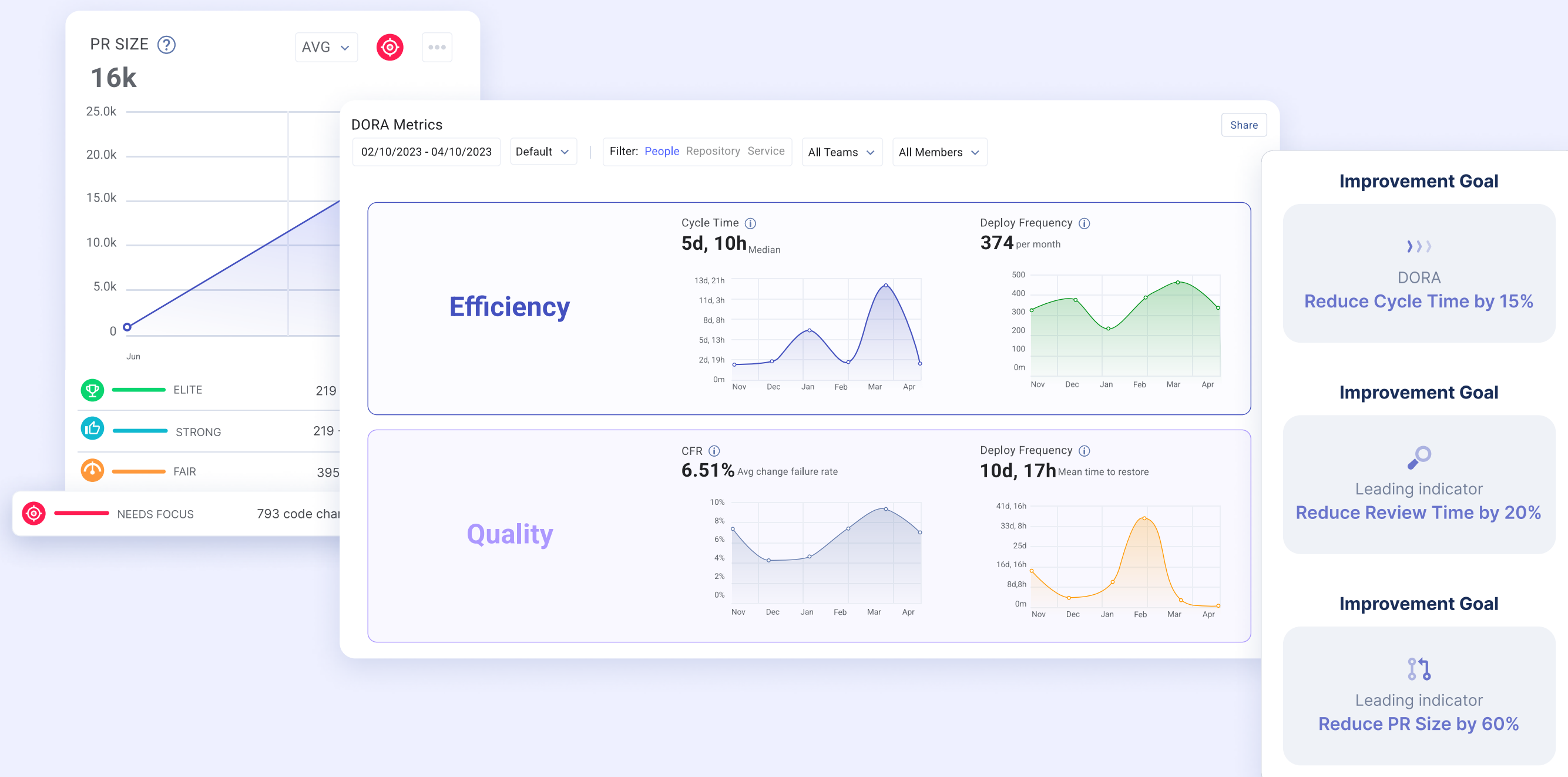
Remember that goals should be tightly coupled at the org and team levels, as well as aligned to business priorities.

This ensures that everyone is working toward the same outcomes.

## Start with Organization Level OKRs

Goal setting should be done at the highest level first: your entire engineering organization. Start by compiling the most fundamental metrics for your whole organization—DORA metrics and associated leading indicators are ideal candidates.

Anchor initial improvement goals around industry standard measures of engineering performance and their leading indicators



# How to Set Engineering Improvement Goals

Use operational metrics, resource allocation visibility, and engineering benchmarks to identify bottlenecks



For many organizations, these metrics will be great initial goals to anchor improvement initiatives around. LinearB also makes it easy to dig a little deeper and set even more granular OKRs—like **keeping Bugs in Progress consistently** under a defined threshold. This is done through customizable metrics dashboards that can track your work however it’s categorized—by epic, story, task, custom field, labels, tags, and more.

In addition to efficiency/quality/DevEx, you’ll want to incorporate visibility into business outcomes as well—like whether you’re prioritizing high impact work like developing new features.

Once you have visibility in place, look at your organization metrics and work to identify your bottlenecks. Using **industry benchmarks** will help you understand if your organization’s 7-day cycle time, 200 diff PR size, or current investment of 35% on “New Value” work, needs to be a focal point of your improvement goals.

Now that visibility and context are in place, you can identify bottlenecks and set your organization-level improvement goals. We recommend aligning to the OKR structure on your goals and using the SMART framework (Specific, Measurable, Achievable, Relevant, Timebound).

# How to Set Engineering Improvement Goals

LinearB is great for setting SMART OKRs because it aligns to the framework out of the box and has built-in functionality for:

- Getting visibility into the metrics that matter most
- Contextualizing metrics with performance benchmarks to guide focus
- Ensuring business alignment
- Setting goals and measuring attainment with simple dashboards
- Automating workflows to help you achieve the results you want to see

## Pro Tip(s)

Set 1 or 2 organization improvement goals MAX. Trying to improve everything all at once is a recipe for failure.

Make sure to get as granular as possible with your metrics when setting goals. This granularity will help you get to the root cause of your bottlenecks.

Here are some examples of business and operational OKRs that you can set at the organization level that use investment focus, DORA metrics, and leading indicators:

## Setting goals for every level of your engineering organization

### OBJECTIVE

**Deliver on my promises to the business this quarter**

#### Key Results

Maintain >75% Planning Accuracy, >85% Capacity Accuracy, and focus on planned work (fitting in unplanned/added work where possible)

Further tactical key results to get there: Keep PR size to <150 diffs, reduce cycle time by 20%, decrease CFR by 15%

### OBJECTIVE

**Improve the efficiency of my engineering organization by End of Q1 (Objective)**

#### Key Results

Reduce cycle time by 25%, 14% increase in deploy frequency

Further tactical key results: Keep PR size to <150 diffs, route PRs appropriately via automation (gitStream)

### OBJECTIVE

**Allocate more of my engineering organization to high value/high impact projects (Objective)**

#### Key Results

Maintain ~40% investment in New Value and ~30% investment in Feature Enhancements

*We covered resource allocation in detail in the Engineering Leader's Guide to Resource Allocation*

# How to Set Engineering Improvement Goals

## Move to Team-Level OKRs

After you've set org-level goals, communicate these OKRs to your management team and team leads. Then they can work to set team-level goals using the same process outlined in the previous section:

- 1 Investigating metrics (like DORA)
- 2 Identify bottlenecks
- 3 Set 1-2 improvement OKRs at a time and follow the SMART framework

## An overview of engineering health

### Efficiency Metrics

Cycle Time *	4d 1h
Deploy Frequency	9.2/ week
Merge Frequency	19.2/ week
PR Size	359
Coding Time	1h 36m
PR Pickup Time	2d 5h
PR Review Time	1d 19h
Deploy Time	23h

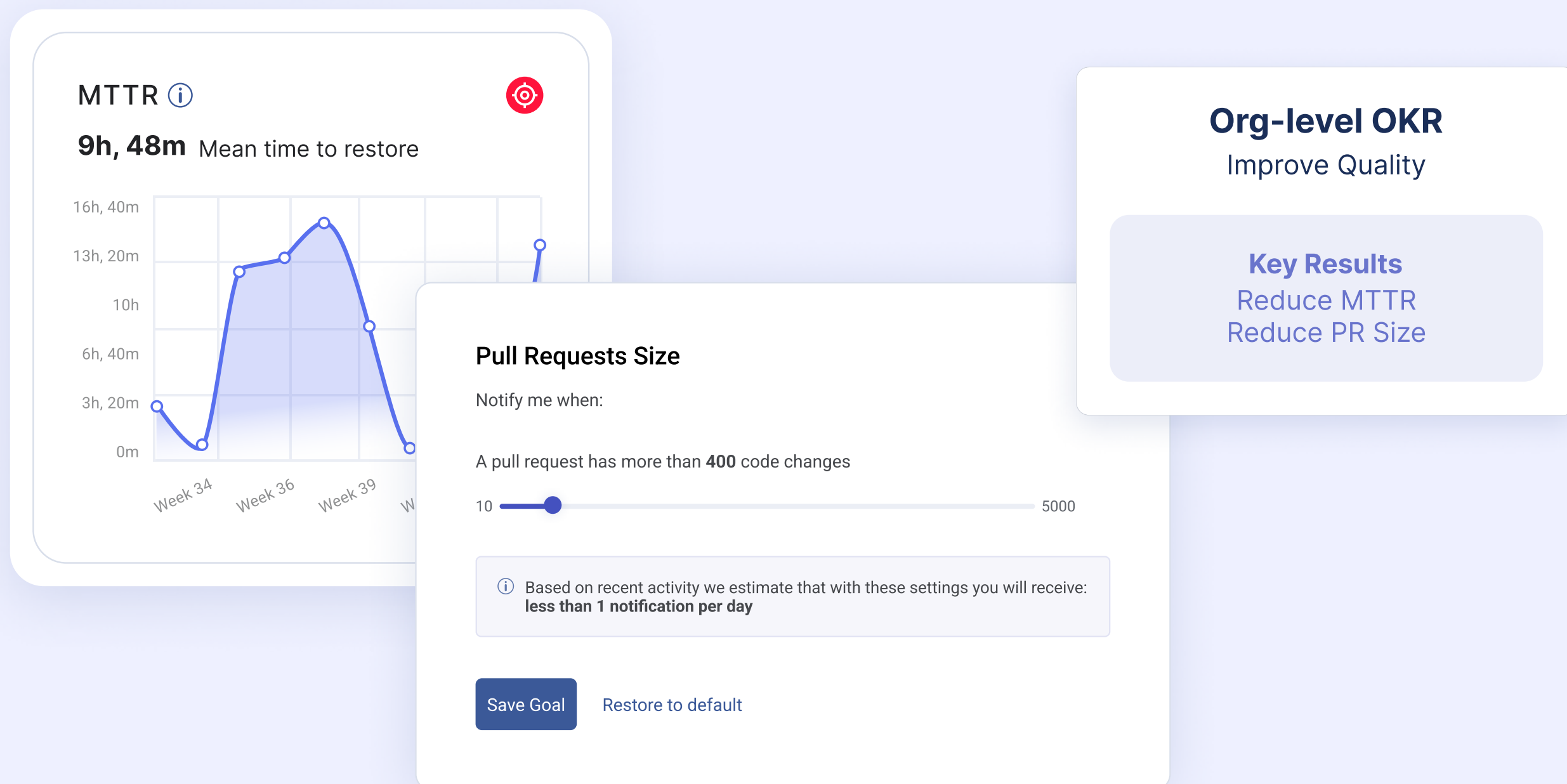
### Quality Metrics

Change Failure Rate*	7.47%
PRs Merged w/o Review	3.7/ week
Review Depth	1.03 comments/PR
PR Size	359
Rework Rate	17.7%
Non-functional work (%)	3.8%
Bug Investment	4%



# How to Set Engineering Improvement Goals

## Outcomes, goals, and key results



At the team level, be sure to leverage views into metrics to set these goals and be sure to align to org-level OKRs. Here is an example of what this looks like in practice:

Team ABC has an org-level OKR to improve overall quality. The key result to make that happen is to reduce MTTR by 2%.

First stop for Team ABC is to investigate what their MTTR is.

In the example shown above, MTTR is a weak point for the team. They'll start by investigating that activity that's causing the spikes. Once they have this context, they can set additional goals as necessary.

In the case of MTTR reduction, a great secondary goal would be to keep PRs under a certain size. PRs that are smaller are able to be reviewed faster, which means that problems are found faster, which means fixes are applied faster. All of which will reduce MTTR.

# How to Set Engineering Improvement Goals

Repeat the process for every team in your organization ensuring that any team goals are always aligned to organizational goals. Here is an example of some of the efficiency and quality bottlenecks that a team may find and set improvement goals against:

The final step in setting goals is to ensure that the entire organization maintains visibility and alignment to stated improvement goals and has the tools necessary to achieve them.

## Identify your bottlenecks based on Benchmarks

### Efficiency Metrics

Cycle Time *	⚠ 4d 1h
Deploy Frequency	9.2/ week
Merge Frequency	19.2/ week
PR Size	⚠ 359
Coding Time	1h 36m
PR Pickup Time	⚠ 2d 5h
PR Review Time	1d 19h
Deploy Time	23h

### Quality Metrics

Change Failure Rate*	7.47%
PRs Merged w/o Review	⚠ 3.7/ week
Review Depth	1.03 comments/PR
PR Size	⚠ 359
Rework Rate	⚠ 17.7%
Non-functional work (%)	3.8%
Bug Investment	4%



# How to Set Engineering Improvement Goals

## Keep Everyone Aligned to Goals with Dashboards and Automation

Once everyone is informed of the mission(s) at hand for the org and individual teams, it's time to (1) build team goals dashboards that all stakeholders have access to and (2) set up automated workflows and PR-policy-as-code to help you achieve those goals. **LinearB makes the process of setting team goals and creating dashboards easy.**

## Building Your Dashboards

These dashboards will ensure visibility, keep stated goals top of mind, and act as an ideal talking point for recurring check-ins/ceremonies to make them more productive and data driven.

You can set up team dashboards that include standardized goals that help you align to engineering best practices. Some examples of these goals include “Avoiding merging high risk work” –

(PRs with more than 100 code changes and a refactor or rework rate of 50% or more) and “Don't merge PRs without a review” (applied to PRs with over 20 code changes).

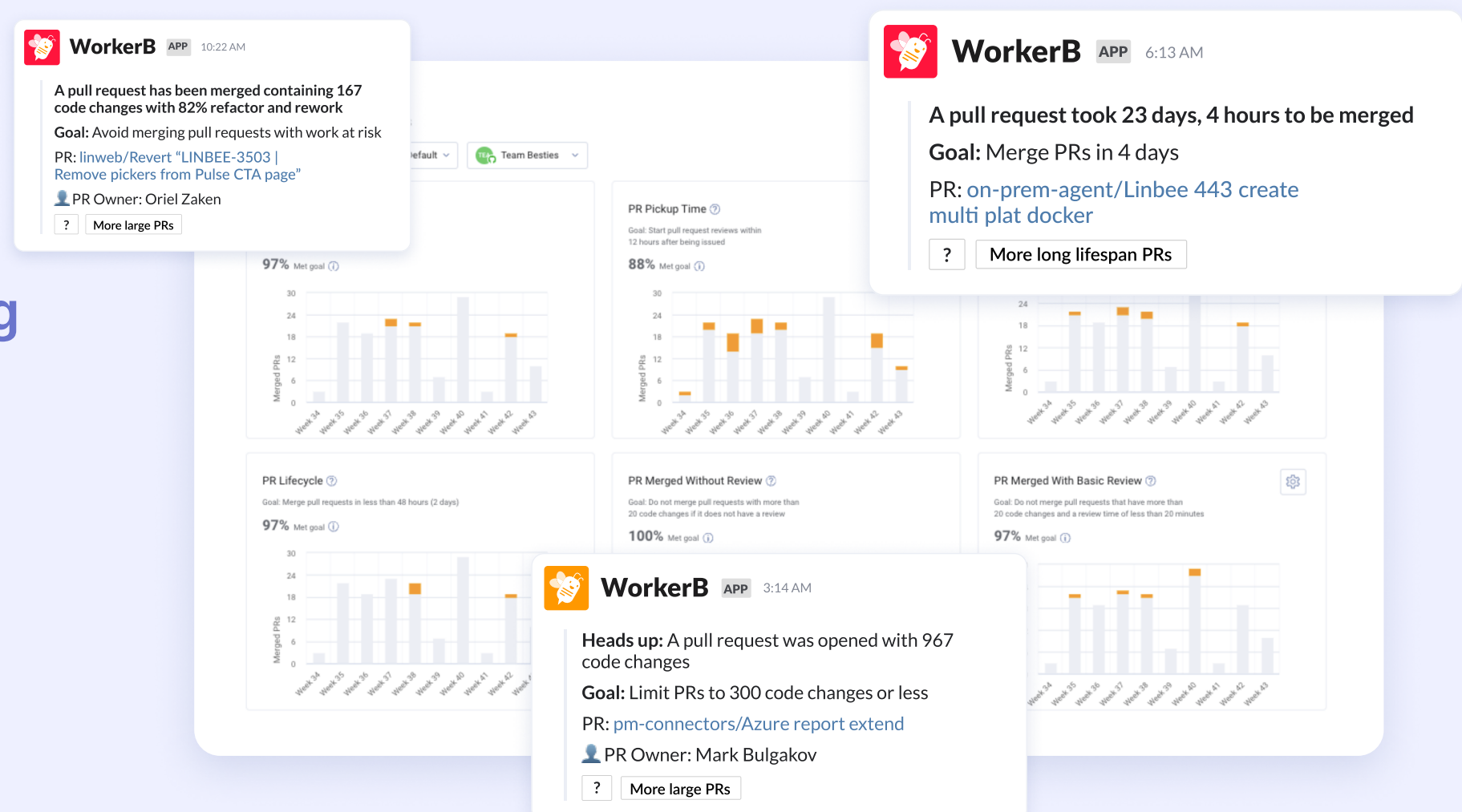
You can then analyze goal attainment for the team on a specified timescale, bring this data to your team(s), and form a plan to improve.

### Pro Tip

When building these dashboards, discuss with your team to determine which goals you'd like to appear and set automated alerts against. LinearB enables teams to **choose their alert thresholds**, toggle goals on/off on a team-by-team basis, and select which goals appear on a dashboard.

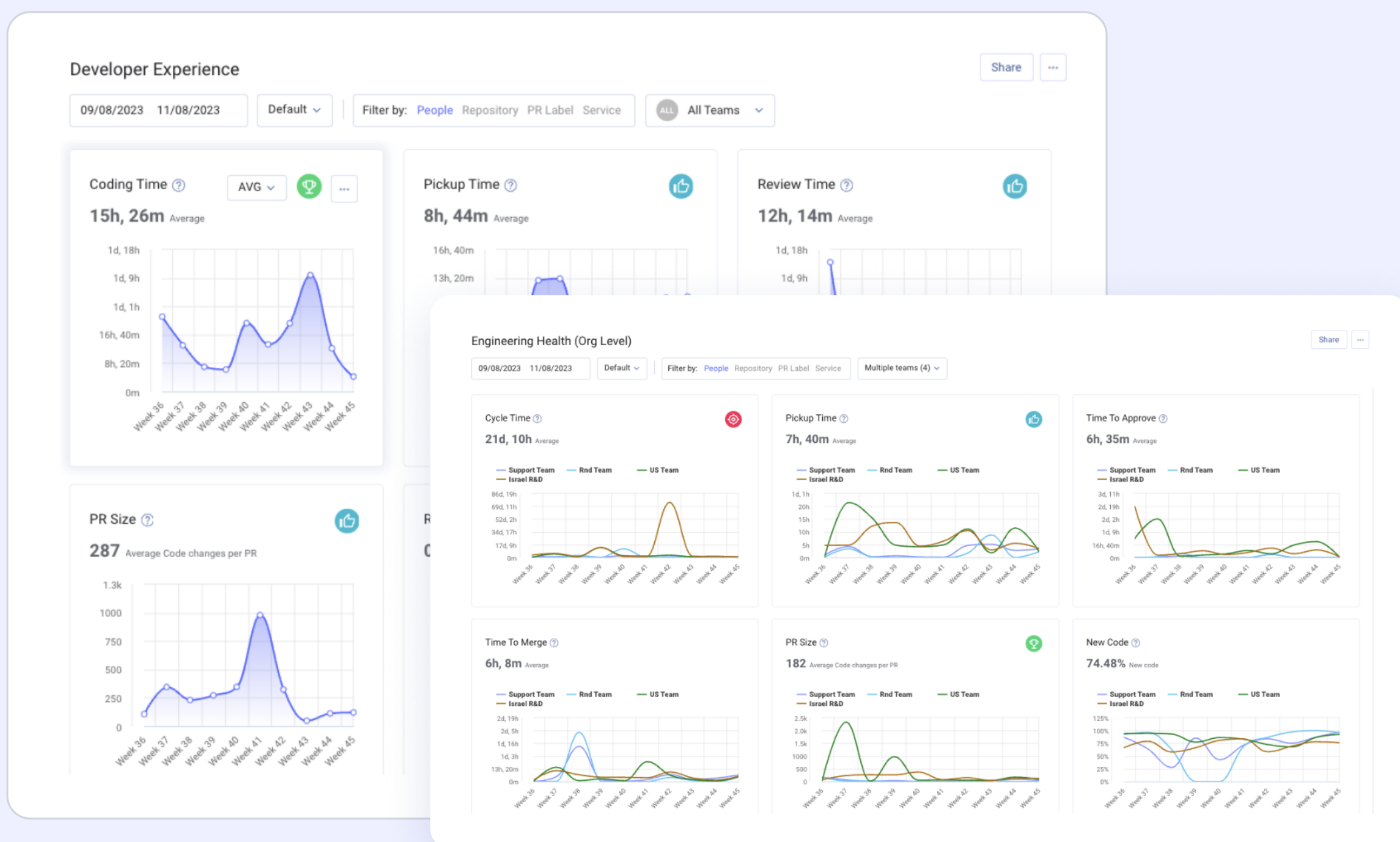
If a team sets a goal to keep PRs under 300 diffs, and a member of the team creates a PR that exceeds that number, they'll receive an automated alert. It's this automated alerting that keeps teams focused and provides a low friction mechanism to achieve their goals.

See how your team is doing against goals they set and keep them aligned with automated alerts.



# How to Set Engineering Improvement Goals

## An engineering health dashboard & A DevEx dashboard



Another great way to set and track improvement goals for your organization is to build custom dashboards using grouped metrics that align to an outcome you want to see. In the examples shown above, assume that an engineering organization wants to see the following outcomes:

1. **A great developer experience**—which fosters talent retention, predictable delivery, and an overall good company culture

2. **Good engineering health**—a great indicator of high code quality, dev efficiency, and delivery predictability

These custom dashboards include the metrics and benchmarks that will inform improvement goals and help teams achieve those outcomes.

# How to Set Engineering Improvement Goals

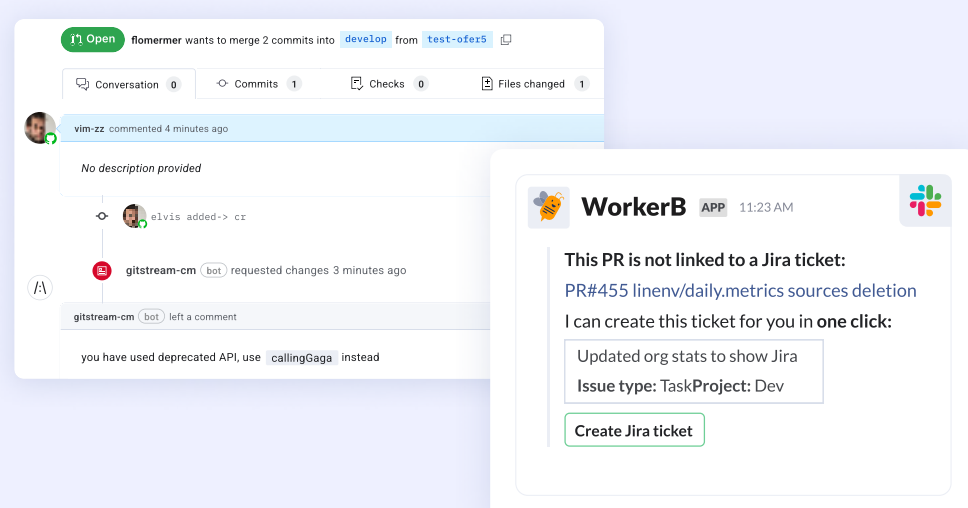
## Automation

In addition to goal-specific automated alerting, programmable workflows are crucial for an engineering organization to achieve stated improvement goals. We'll cover workflow automation in detail in a future guide, but it is the key to unlocking more efficiency, higher quality, and better business results.

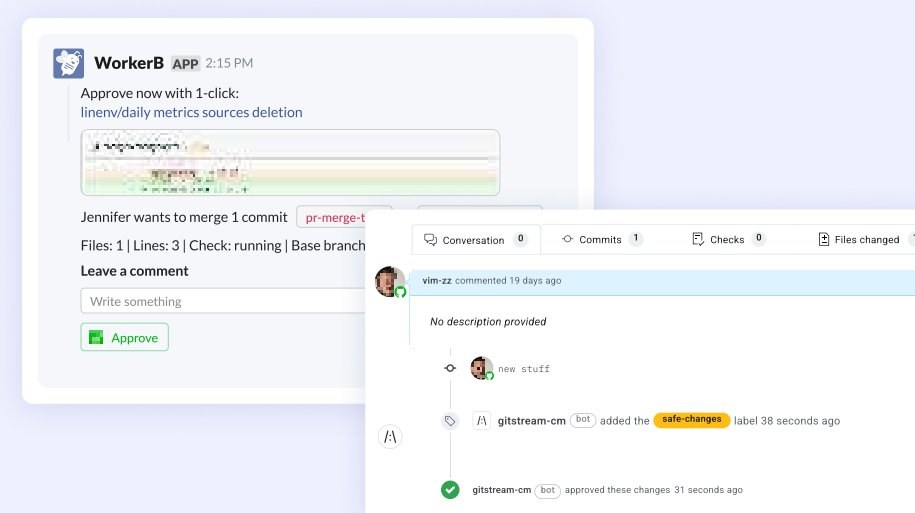
LinearB provides two extremely powerful automation tools to help in this regard:

- **WorkerB:** A developer bot that automates pre-merge manual tasks devs hate, speed up code reviews, reduce toil/idle time, and help your team prioritize their day.
- **gitStream:** A holistic automation and rules engine that organizations can apply to repos (or the whole org) that classify and route PRs appropriately, automate security checks, add context, and more.

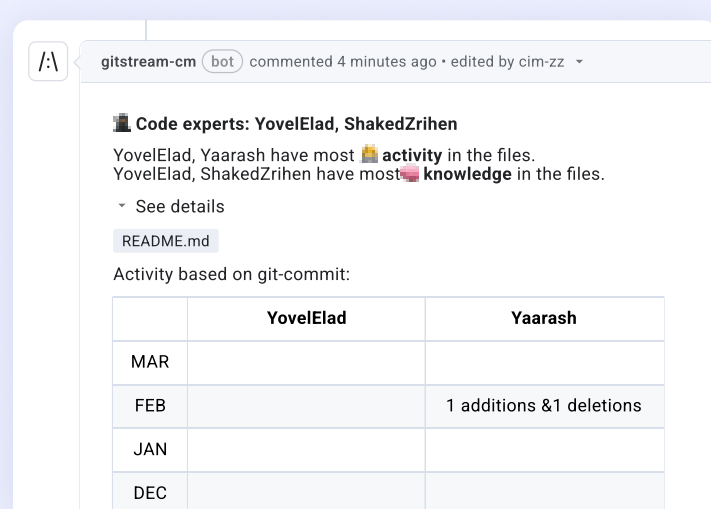
## Workflow automation is how teams improve



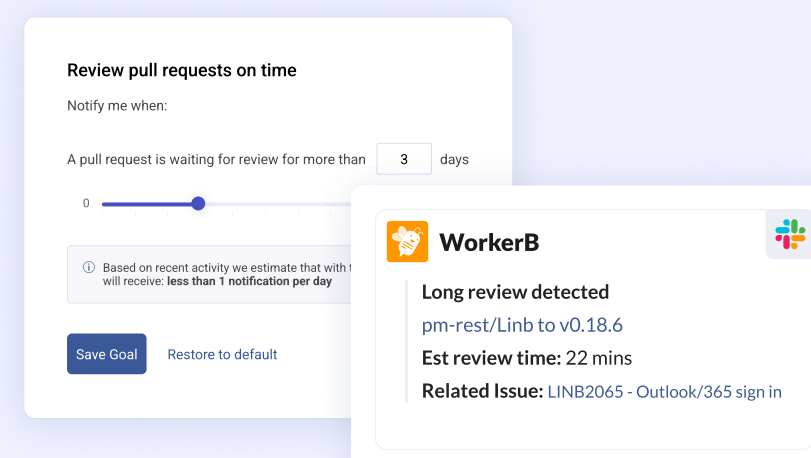
Ensure quality and adherence to best practices



Streamline the code review process with automation and one-click workflows



Meet quality and efficiency goals by automatically assigning reviews to code experts



Set alert thresholds for goals to keep teams aligned to improvement efforts



# Reporting on Goals, Success, and Further Opportunities

This is the fun part: looking into how you're doing as an organization, evaluating progress against your stated goals, and determining how far you still have to go.

When reporting to various stakeholders, understanding your audience is crucial to determine what data you want to present and how you want to present it.

## Reporting to Sr. Leadership, C-Suite, and other business leaders



**"Brevity is the soul of wit."**

Shakespeare

When providing an engineering update to business stakeholders, think of yourself as a translator—in this context, that means putting **everything** in terms of **business relevance and impact**.

By sticking to the talking points and readouts presented below, you'll be able to confidently report on engineering health, resource allocation, project status, delay risks, and (most importantly) how all of this will impact the bottom line. Best of all, you'll be able to do it with data instead of gut feeling!

### Pro Tip

Make your update to this audience as clear and concise as possible. Keep technical details to a minimum and only provide them when necessary or requested.

# Reporting on Goals, Success, and Further Opportunities

Here's what this audience wants to know:

## Updates on key deliverables and recommendations:

Provide a quick overview of the status or ETA of a new feature or enhancement that the business is expecting. Communicate any significant progress or blockers that will impact promised delivery dates.

Always provide next steps and how you plan to address any gaps in execution. You can streamline a large part of your update by [using this slide template](#) and plugging your metrics into it.

### Key Engineering Investments

Q1, 2023

Initiative	Business Impact	Q1 Actual	Engineering Execution	Q2 Recommend
ABC-123 Lorem ipsum dolor sit amet	HIGH	8.4 \$315K	<b>STRONG</b> 4d 1h 1.8	Boost
PRJ-14 Morbi id purus vitae risus sagittis	HIGH	7.8 \$283K	<b>FAIR</b> 6d 2h 1.1	Improve
MOB-456 Morbi id purus vitae risus sagittis	MED	7.5 \$281K	<b>ELITE</b> 1d 7h 2.9	Maintain
ABC-223 Integer gravida erat vitae rutrum	LOW	6.2 \$233K	<b>NEEDS FOCUS</b> 10d 2h 0.8	Reduce
CS-111 Sed feugiat enim elementum	HIGH	-	-	Start
<All other work>		7.6		

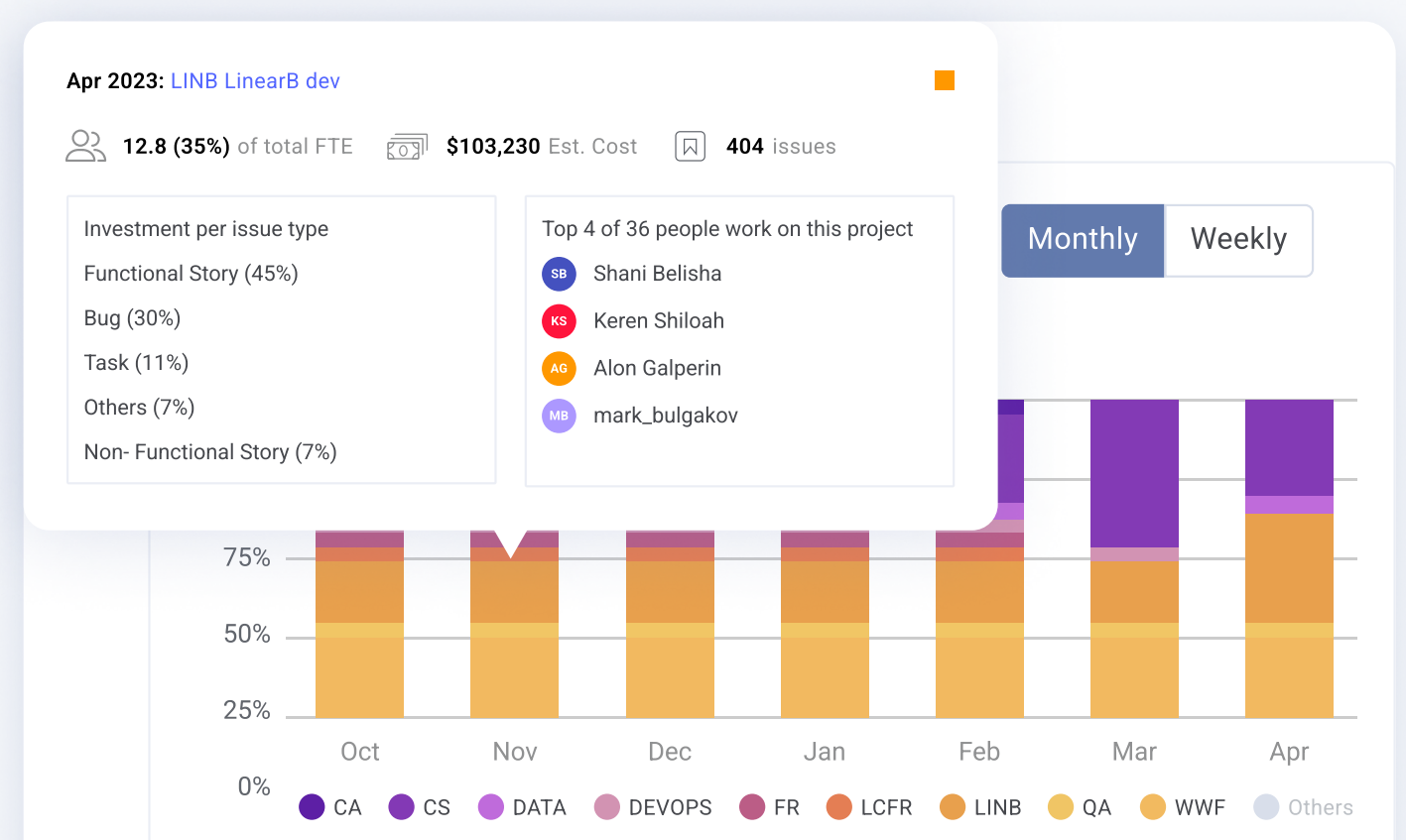
FTE | Estimated Cost (avg. cost of \$150K/dev) | Cycle Time | Merge Frequency (Merged PRs/Dev/Week)

Unified update on key projects, execution, and recommended strategy

## Where/how you're investing resources:

Outline how projects are resourced and your overall investment mix.

Make sure to surface how this aligns to any business outcome goals you've set for engineering.

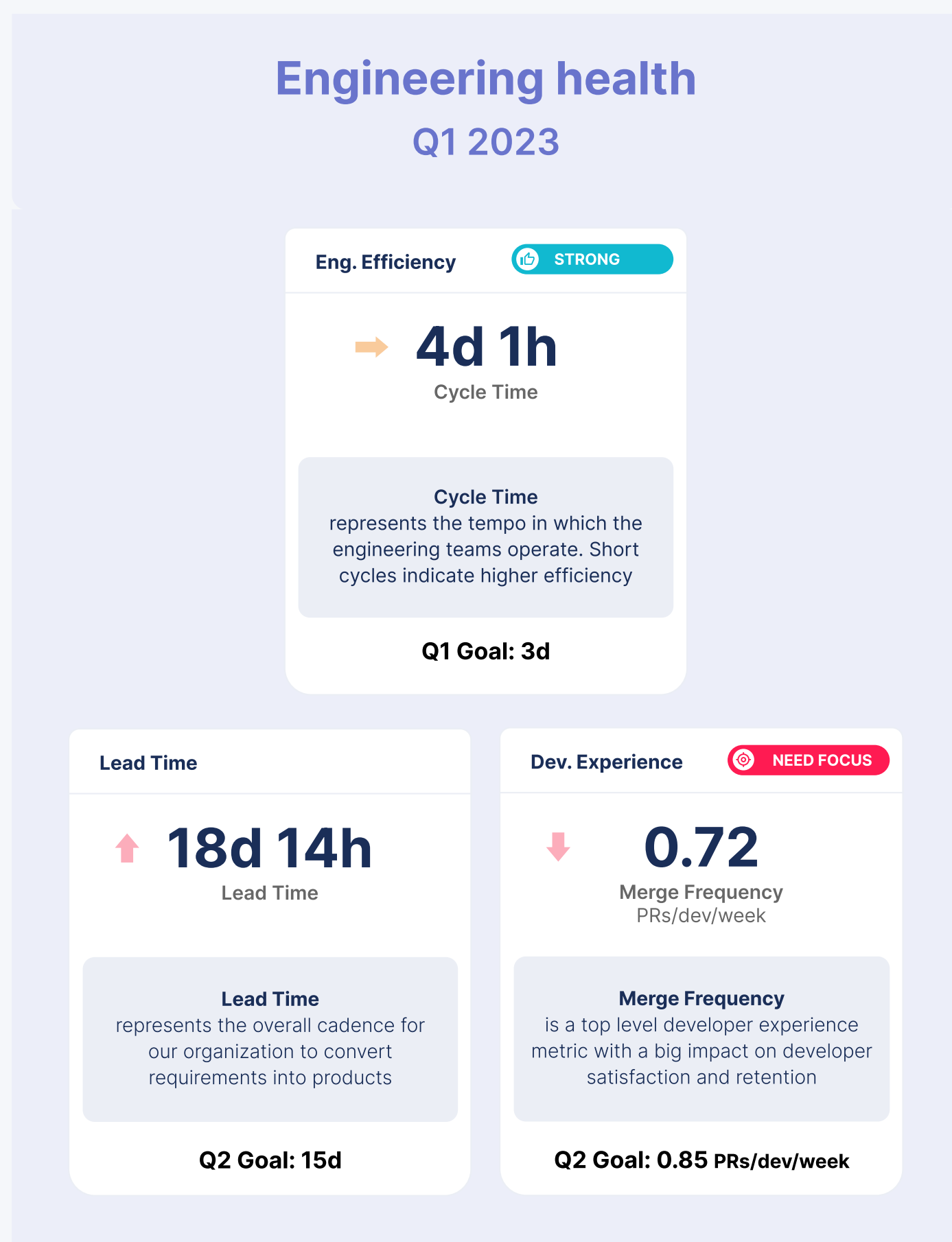


A single dashboard that combines projects, resourcing, work breakdown, and total cost

# Reporting on Goals, Success, and Further Opportunities

## Overall engineering health:

Provide a quick update on efficiency, dev experience, and execution data, and any delay risks (like an uptick in unplanned and/or added work).



## Expected business impact:

This is an update that will likely make this audience sit up and take notice. We covered this in detail in the Engineering Leader's Guide to Resource Allocation (link) but if you've reallocated dev cycles or budget to a high impact project and initiatives classified as new value, the impact on the business is likely significant. [Learn more about this relationship with the ROI Calculator.](#)





# Reporting on Goals, Success, and Further Opportunities

## Reporting to Engineering Organization and Teams

This is your comfort zone. You've been in these meetings a lot in your career in various roles. The venue is probably your weekly engineering all-hands, managers meetings, team-level meetings, and recurring ceremonies (like retrospectives). When presenting on goal attainment and updates, supporting technical details and getting into the nitty gritty is great. You're likely addressing the following people:

- ✓ Engineering managers
- ✓ Team leads
- ✓ Sr. Developers
- ✓ Product leaders

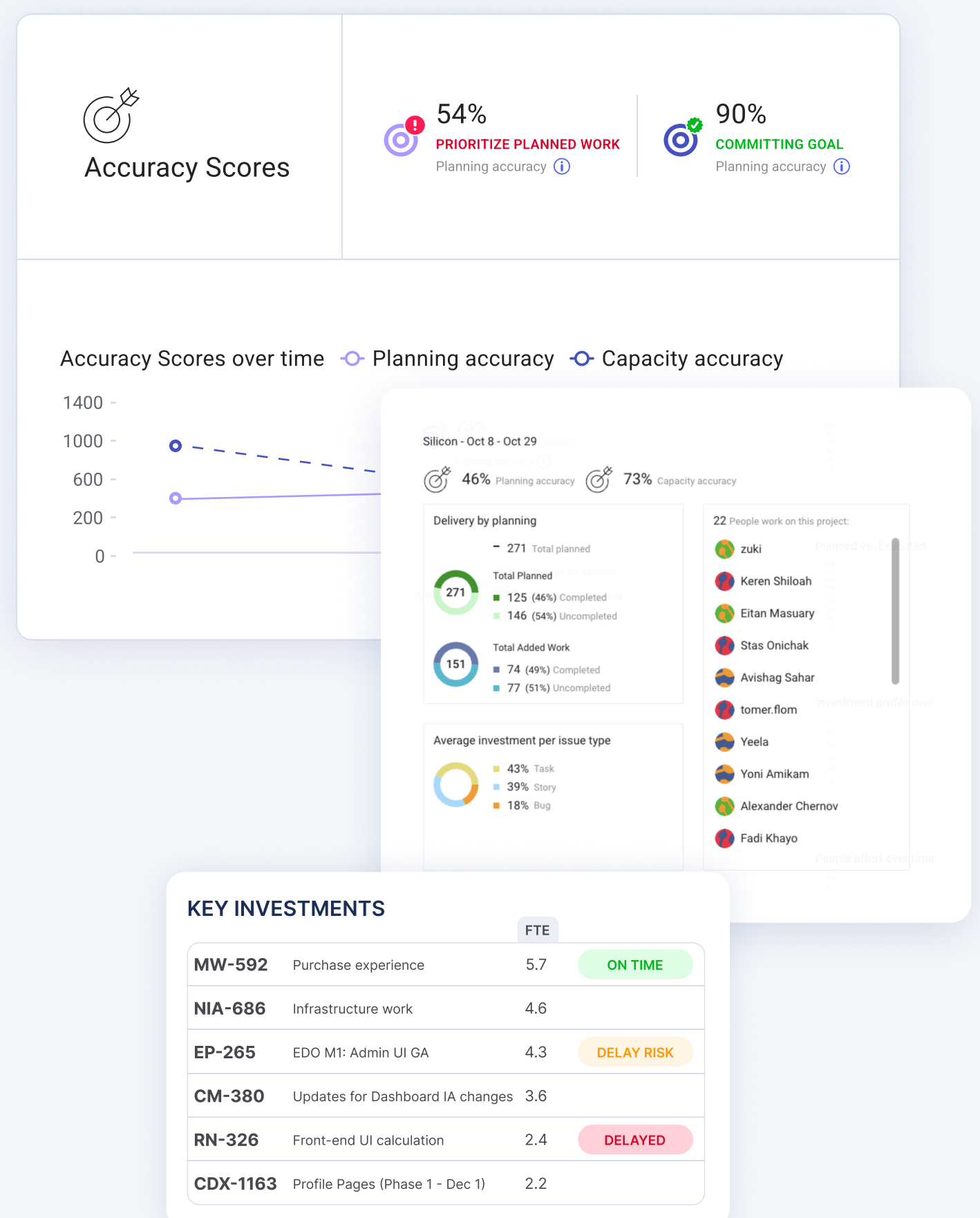
They can then take this data to their team(s) to share and come up with ways to refine their strategy and meet their goals.

One thing to highlight and emphasize with your team is that this is about improving as an organization/team and mentoring, not monitoring. Big brother is not watching—it's through the process of metrics, benchmarks, goal setting, and (re)evaluation that we improve.

Here's what this audience wants to know:

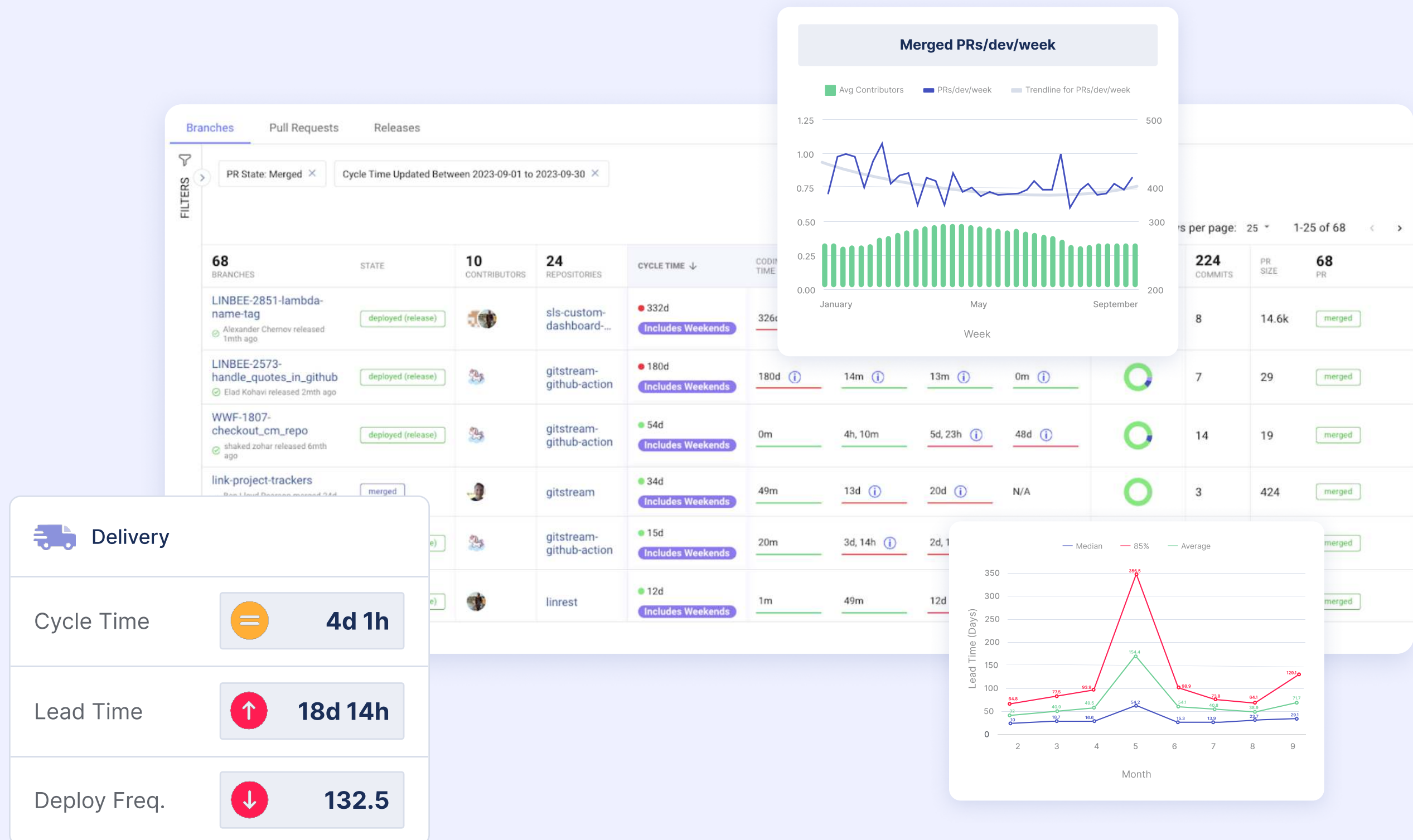
### Updates on promised deliverables and overall project health:

You can start with a general overview on project status (basic Red, Yellow, Green) but you can also dig into execution and project data as well. If something is trending Red, surface the reason(s) why.



# Reporting on Goals, Success, and Further Opportunities

Deep dive into operational data with your engineering teams



## Project operations

Dive into DORA, leading indicators, idle time, context switching, WIP, bugs (technical vs. functional), security incidents, DevEx, and anything else that needs to be talked about. Use any of the dashboards

(goals, metrics, project, activity) to make your point about goal attainment and what is helping or hurting your progress.

# Reporting on Goals, Success, and Further Opportunities

## Any progress (or risk) with stated operational goals:

Make sure you're taking time to remind the team of goals that they're working towards, celebrate successes, and surface any risks.

### Efficiency Goals



#### Org Level OKR

Decrease Cycle Time by 25%

#### Team Goals

PRs under 200 diffs  
PR Pickup Time < 5 hours

### Quality Goals

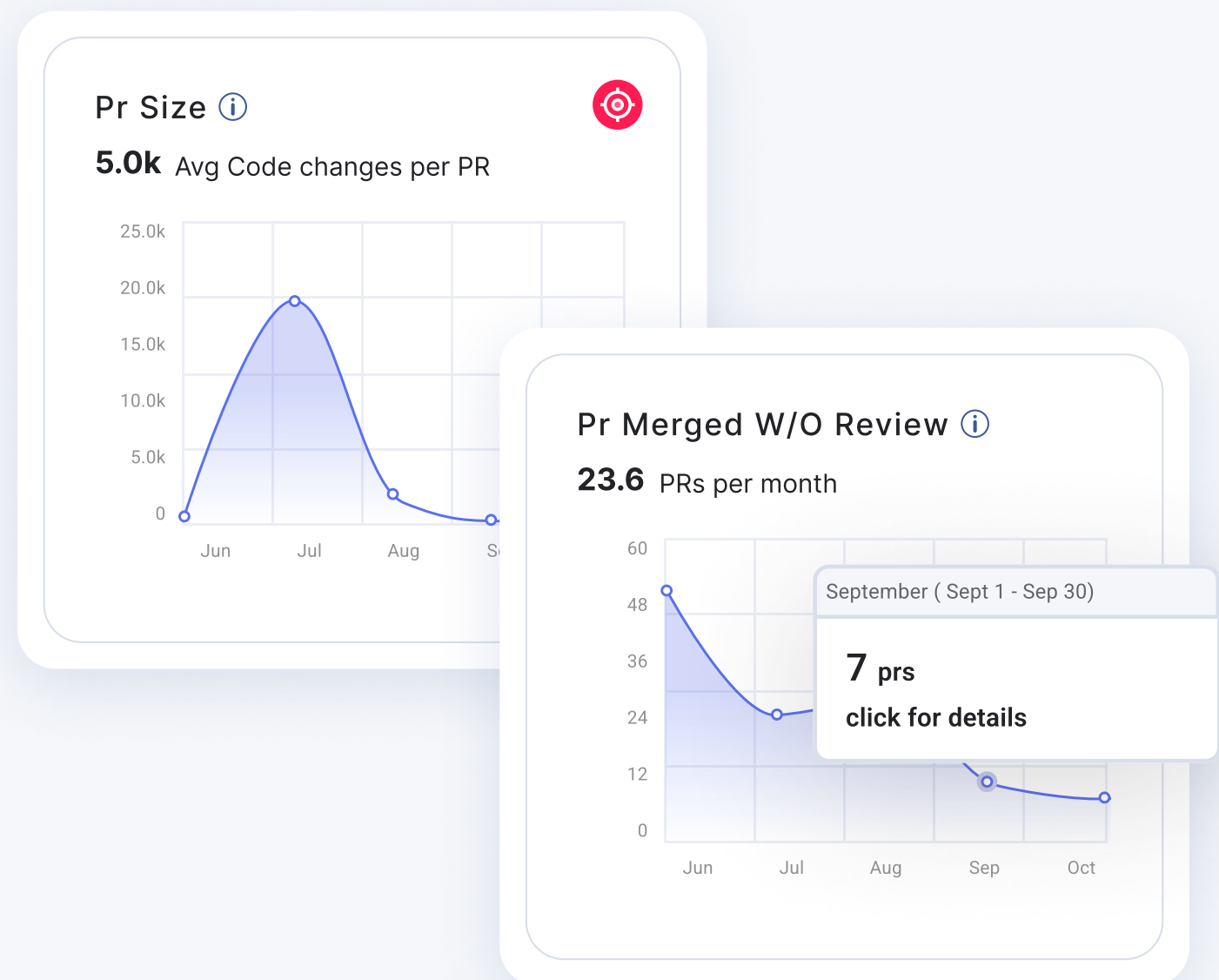


#### Org Level OKR

Decrease Rework by 20%

#### Team Goals

PRs under 200 diffs  
0 PRs merged w/o review



## Your recommendations on how to move forward:

Just like the reporting session for business stakeholders, you'll want to figure out how to update strategy and tactics to achieve your stated goals. That might mean moving devs from one project to another, reallocating resources to a high impact project, deprioritizing a lower impact project, or adopting a new tool that can help with quality or efficiency.

Establish a rhythm for how these recurring checkpoints should go and come up with a template using the views above for presenting overall engineering health, progress on goals, and identified risks/bottlenecks, and plans to mitigate them. Practice makes perfect.



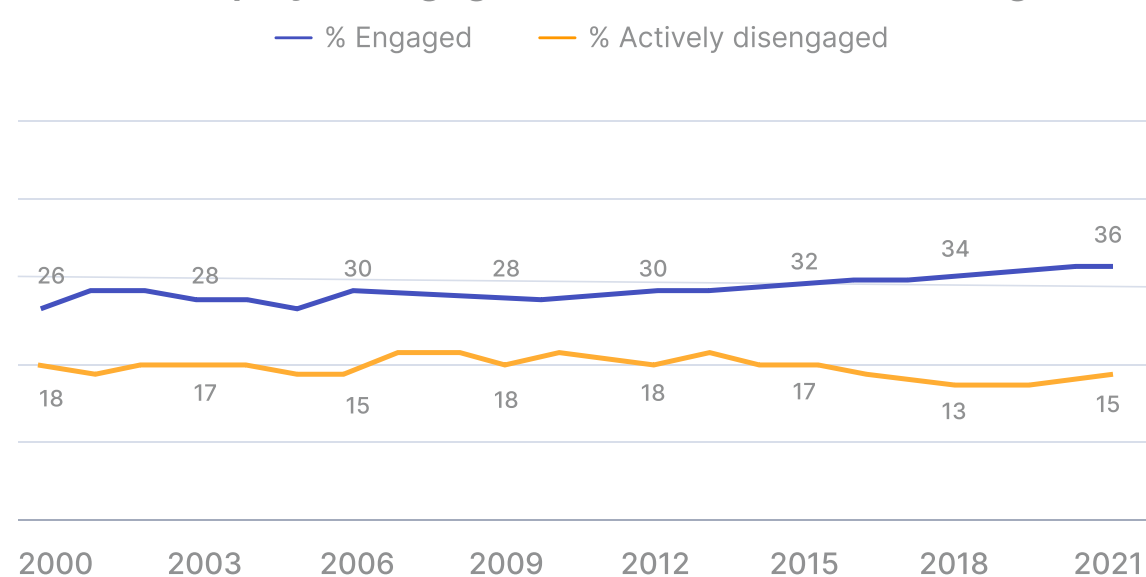
# The universal, measurable impact of goal setting

While it may be stating the obvious: Performance and improvement goals are crucial for every business—regardless of industry vertical, size, or structure.

## Goals help motivate people:

According to a 2022 Gallup survey, only 36% of US employees are actively engaged at work. That figure drops to 20% on a global level. Even more staggering is the fact that 15% are actively disengaged. Having a goal helps to drive more engagement because it gives people something to work toward—necessitating that they are more engaged at work

U.S. Employee Engagement Trend, Annual Averages



Note: 2021 results are for January through June.  
GALLUP

## Goals help organizations avoid misunderstandings:

Goals provide a rallying point for everyone on your team and within your organization. It's crucial to set short-term, long-term, and when appropriate, Big Hairy Audacious Goals. They align teams and ensure everyone is on the same page.

## Goals increase performance /productivity/value:

Goals help the individual and the organization improve performance and increase value by:

- Driving effort and keeping you motivated to do the work
- Directing effort and fostering persistence because they provide something to come back to when challenged and force you to think creatively/strategically

# The universal, measurable impact of goal setting

A LinkedIn study found that goal setting netted an average 18% increase in the value of output. In the example provided, a software developer that makes \$80,000 a year has “do your best” goals. After setting challenging, specific goals, the company could expect the value of the increased productivity from that developer to be worth about \$14,400. For a 30 person startup where the average salary is \$80,000, that translates to about \$432,000 a year in additional output. And that impact compounds over time. Every year the company adopts a rigorous goal-setting process, the benefit accumulates.

## Goals maintain accountability:

When goals are set, everyone becomes accountable to them (from the top C-suite to the individual developer). It also becomes easier to maintain a positive working relationship when goals are in place—managers can tell their reports how they’re doing, and offer support when they need it.

## Goals lead to better decision making:

Past goals and performance are a fantastic way to inform strategy and decisions for the future. If you set a goal and achieved it, stay on the path. If something is working, keep doing it. If you set a goal and didn’t hit it, alter it or try something new.

“

**“Those who cannot remember the past are condemned to repeat it.”**

George Santayana

# Conclusion

Congratulations, your metrics program and general strategy now includes tangible, measurable improvement goals that will drive increased efficiency, higher quality, and better business outcomes.

Look out for the next guide in this series where we'll deep dive into workflow automation.



# Conclusion

LinearB's software delivery management platform is ideal for building an engineering metrics program from scratch and aligning engineering to the needs of the business. After onboarding and initial configuration/integration with your existing software stack, you'll get the metrics, dashboards, insights, and automation discussed in this guide right out of the box.

You can **get your DORA metrics with a free forever DORA Dashboard** and begin building your engineering metrics program today! If you'd like to discuss any of what was covered in this guide in more detail, or you want to see Goals and Reporting in action, you can **schedule a demo**.

We'd love to hear from you!