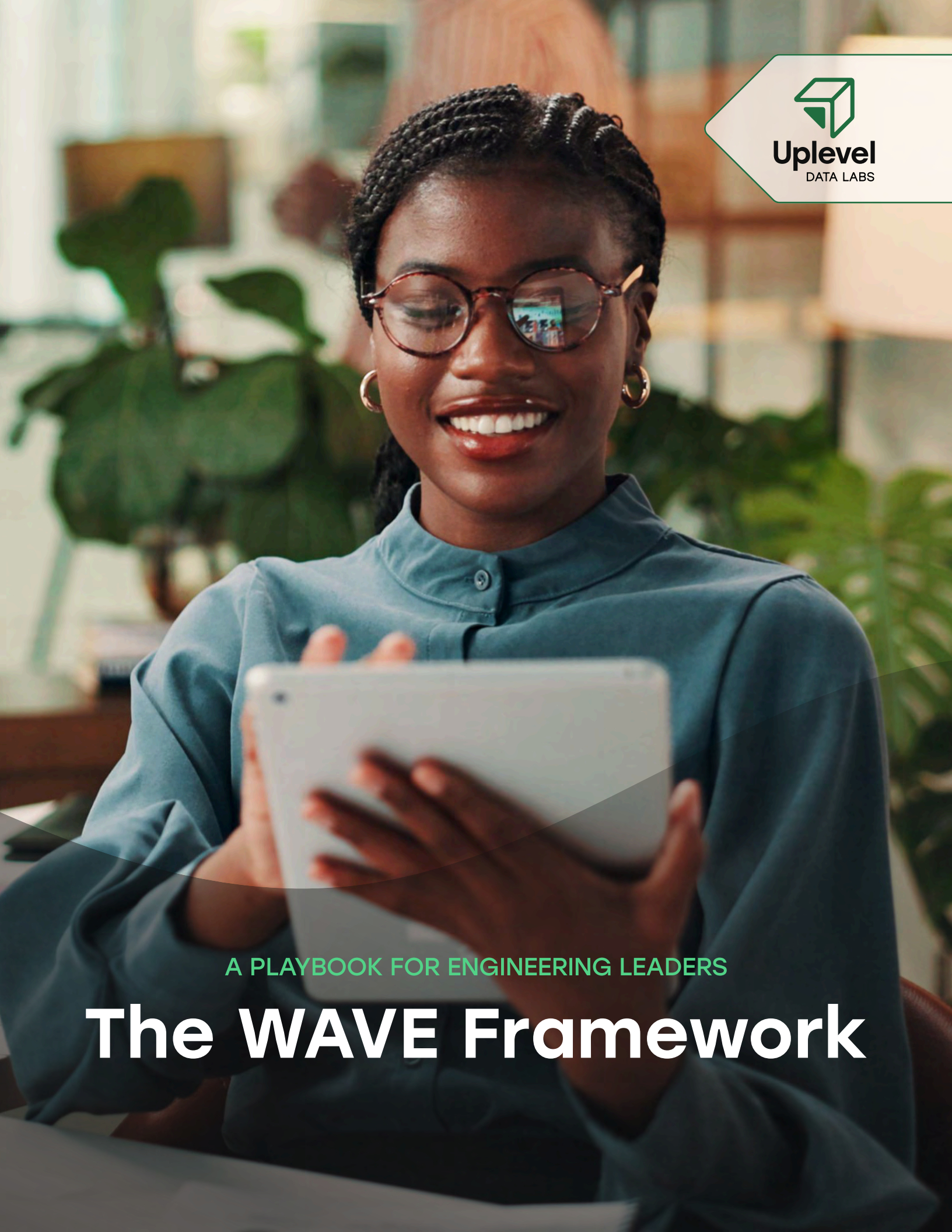


Uplevel
DATA LABS

A PLAYBOOK FOR ENGINEERING LEADERS

The WAVE Framework

Table of contents

The measurement trap

Despite implementing various measurement frameworks, many engineering leaders fall into the trap of measuring what's easy rather than what's meaningful.

Most measurement approaches treat engineering as a purely technical practice that can be optimized through technical metrics alone. But engineering organizations are sociotechnical systems, where human collaboration, communication patterns, and environmental factors are just as important as code deployment statistics.

Most engineering organizations already have plenty of data – but they lack a cohesive framework to interpret that data and drive meaningful change. Uplevel's WAVE Framework can transform your engineering metrics from mere measurements into actionable insights that drive real improvement.

What's the problem with traditional engineering KPIs?

Many organizations collect metrics without a clear understanding of what they're trying to achieve. Traditional KPIs often create an illusion of control, failing engineering leaders in several critical ways:

- **Too much focus on individual output:** Engineering leaders frequently track metrics like PR counts or story points completed. But these metrics are poor proxies for productivity and can lead to detrimental behaviors like artificially inflating PR sizes or submitting unnecessary code changes.

- **Overreliance on lagging metrics:** Frameworks like DORA give you valuable insights, but these are backward-looking measurements. For engineering leaders under pressure to improve future performance, understanding that deployment frequency was low last quarter offers limited actionable guidance on what to change now.

- **Overlooking social dynamics:** Research has demonstrated that team collaboration patterns are often stronger predictors of success than individual technical skills. Yet most organizations focus exclusively on technical metrics while neglecting team dynamics.

- **Little correlation between metrics and business value:** Many organizations measure what's easy to track rather than what drives tangible business outcomes. As a result, they optimize for metrics that don't have a meaningful impact on the organization's success.

- **Limited ability to act on the data:** Research by Dr. Nicole Forsgren (co-author of Accelerate) highlights that without contextual information about organizational structure, team interactions, and environmental factors, technical metrics alone are insufficient for diagnosing performance variations across teams.

These limitations leave engineering leaders with plenty of data but insufficient guidance on what to change.

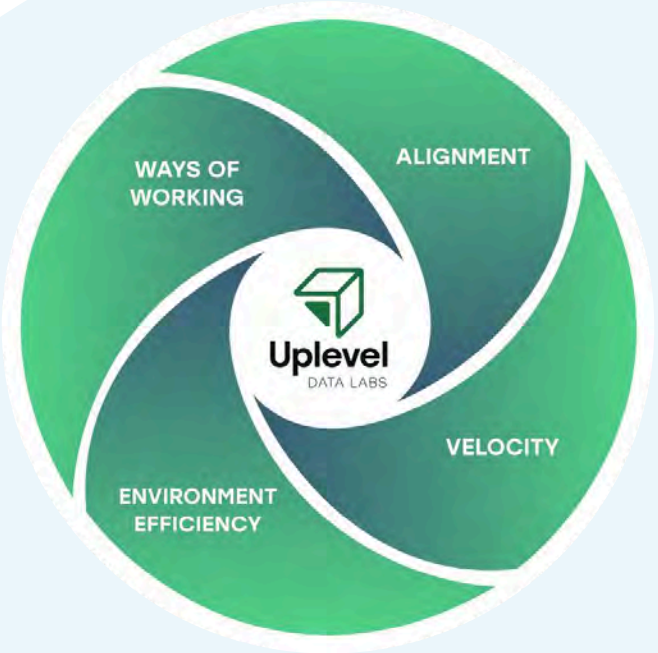
The WAVE Framework: A Holistic Approach to Engineering Effectiveness

Unlike frameworks that focus narrowly on deployment statistics (DORA) or that provide theoretical models without clear measurement approaches (SPACE), WAVE addresses the full spectrum of factors that influence engineering effectiveness. Most importantly, it recognizes that these factors are interconnected: improvements in one area cascade through the entire system.

WAVE is based on our data science findings and deep experience partnering with engineering leaders. Each category below offers a small group of dimensions and metrics that provide opportunities for actionable intervention. WAVE provides manageable clarity while still addressing the complexity of a sociotechnical system.

The WAVE Framework consists of four interconnected components:

- **Ways of Working (W):** Measures how effectively teams collaborate, their overall health, and their ability to focus on deep work
- **Alignment (A):** Captures how well engineering efforts align with business objectives through planning, resource allocation, and user feedback
- **Velocity (V):** Tracks the flow of work through the system, including cycle times, throughput, and lead times
- **Environment Efficiency (E):** Evaluates the quality, deployment process, and overall friction in the engineering environment



WAVE FRAMEWORK; SOURCE

Instead of treating metrics in isolation, WAVE recognizes the interconnections between different aspects of engineering work. Engineering is not just coding – it's all your team's interactions with the product, users, and cross-functional teams.

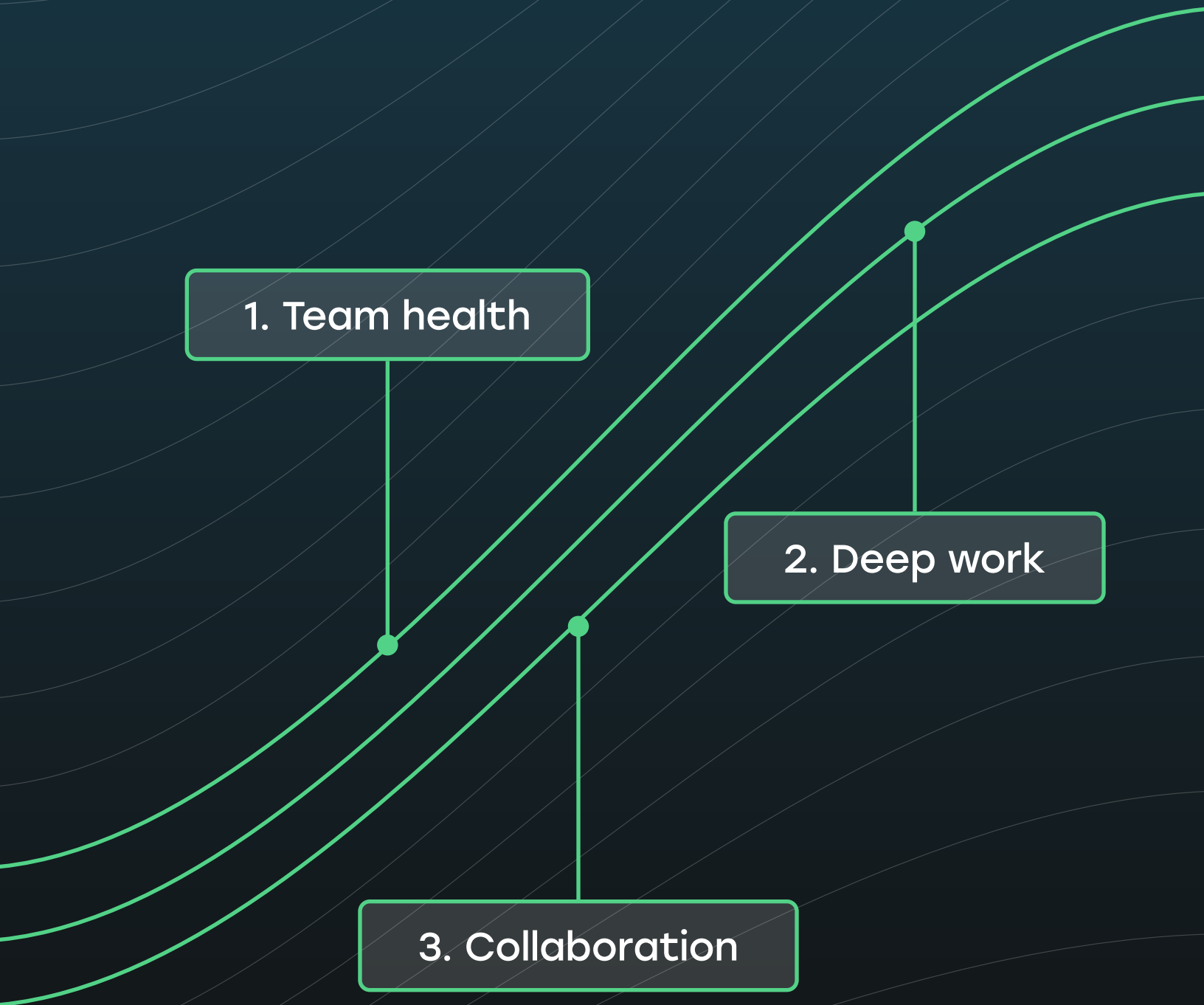
The WAVE Framework creates a diagnostic map that helps engineering leaders understand the relationship between different dimensions of performance, enabling targeted improvements rather than isolated optimizations.

Let's look at the engineering KPIs that WAVE measures.

WAYS OF WORKING	ALIGNMENT	VELOCITY	ENVIRONMENT EFFICIENCY
TEAM HEALTH	**ALLOCATION**	**CYCLE TIME / THROUGHPUT**	**QUALITY**
Composite team health score	• Allocation to new value vs. Allocation to tech debt • Allocation to new value overtime	• PR cycle time • PR throughput • Epic throughput	• Defects by type (bug vs. escaped defect • Defects found by stage
COLLABORATION	**PLANNING**	**WORK IN PROGRESS**	**DEVOPS ENVIRONMENT**
Collaboration satisfaction score	• Sprint completion • Requirements churn	WIP per team	DORA metrics
DEEP WORK	**USER FEEDBACK**	**LEAD TIME**	**FRICTION / FLOW**
Average developer Deep Work hours per day	User feedback cycle time	Epic lead time	• Friction score • Flow efficiency (waiting/total)

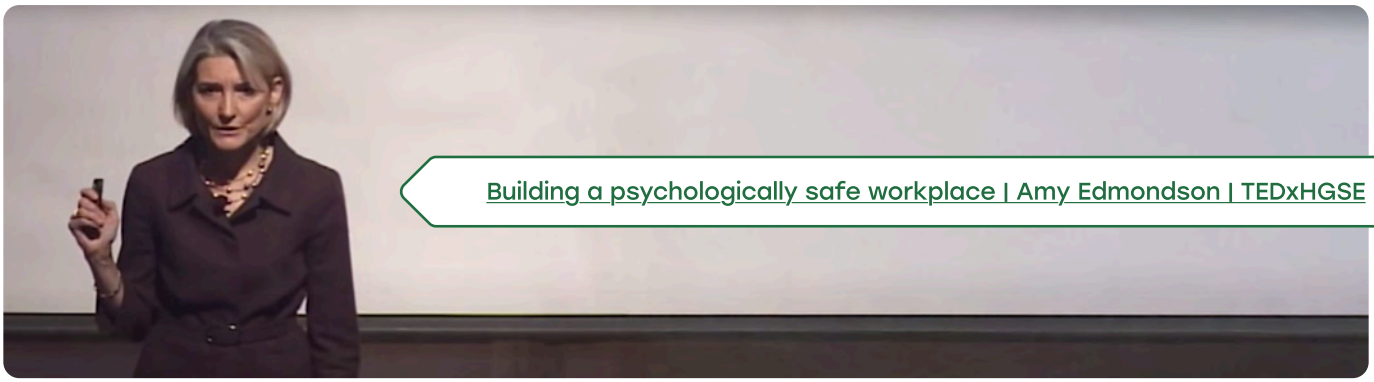
Ways of Working: Team Health and Collaboration

The "Ways of Working" dimension recognizes that engineering performance begins with people, not with code or deployment pipelines.



1. Team health

Team health metrics provide a consolidated view of engineering teams' psychological safety, collaboration effectiveness, and overall engagement. This approach is grounded in Google's Project Aristotle research, which identified psychological safety as the most important factor in team effectiveness.



Building a psychologically safe workplace | Amy Edmondson | TEDxHGSE

In software engineering specifically, a 2024 study in Empirical Software Engineering found that teams with established psychological safety were more invested in software quality, demonstrating "collective problem-solving, pooling their collective intellectual efforts and experience to tackle quality-related challenges."

When you track team health over time, you can identify early warning signs of burnout, disengagement, or collaboration challenges before they impact delivery.

ACTIONABLE INSIGHT:

Rather than treating team health as an HR concern, engineering leaders should view it as a critical engineering effectiveness metric. Implement regular team health assessments focusing on psychological safety, clarity of goals, and meaningful recognition. Create dedicated time for teams to discuss and address issues identified through these assessments.

HOW UPLEVEL MEASURES TEAM HEALTH:

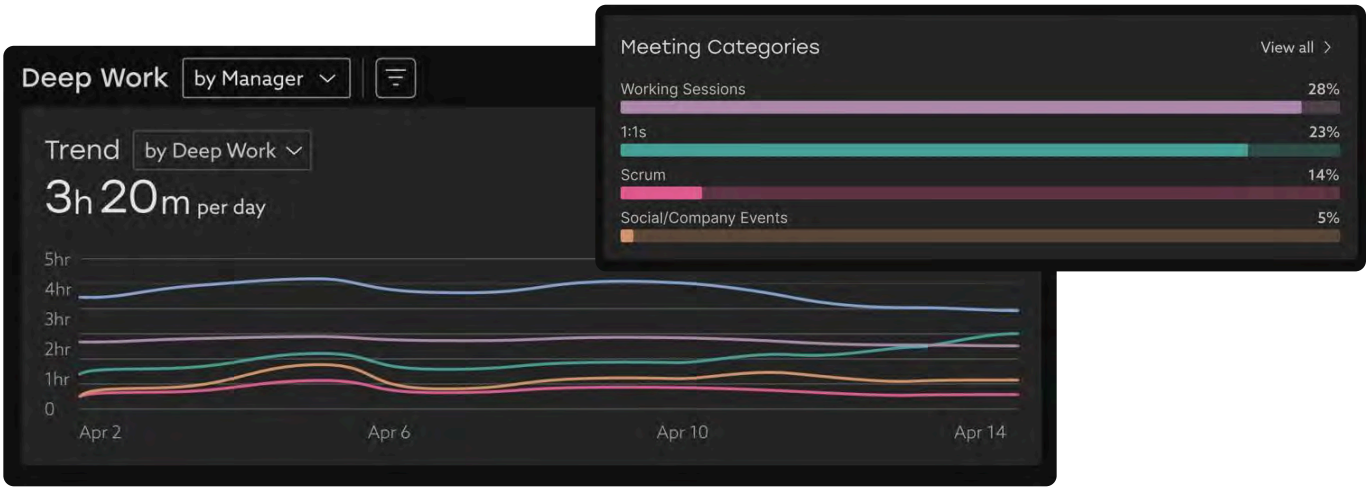
UPLEVEL PRODUCT **UPLEVEL METHOD**

As a quantitative proxy for team health, Uplevel measures “Sustained Always On,” a gauge of sustained work beyond a dev’s normal working hours, which is a leading indicator of burnout. We also ask questions pertaining to psychological safety and team health in qualitative interviews and surveys as part of the Uplevel Method.

2. Deep work

Deep work metrics track the average number of daily uninterrupted hours developers can dedicate to focused coding time. Cal Newport's Deep Work demonstrates the critical importance of uninterrupted focus for complex cognitive tasks like software development.

This concept is further supported by studies from the University of California, which found that after an interruption, it takes an average of 23 minutes for knowledge workers to return to their original task. For software engineers, context switching is particularly costly – frequent interruptions lead to increased defect rates and longer completion times for complex programming tasks.



ACTIONABLE INSIGHT:

Establish your organization's baseline deep work scores via an engineering intelligence platform that ingests and analyzes calendar and collaboration tool data. Consider implementing "no-meeting" blocks across the organization, as companies like Asana and Shopify have done successfully. Create team agreements around communication tools to minimize interruptions during focus time.

HOW UPLEVEL MEASURES DEEP WORK:

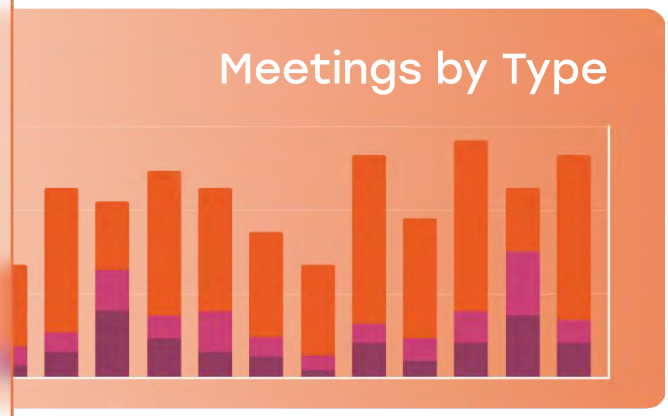
UPLEVEL PRODUCT

Using a combination of chat and calendar data, Uplevel is able to calculate how much uninterrupted time each developer has each day. We define deep work as a time block of at least two hours of focus time without meetings or significant Slack interruptions. Our recommended benchmark is an average of four hours of deep work time per day.

3. Collaboration

Collaboration metrics assess how effectively teams share knowledge, provide feedback, and support each other's work. How teams collaborate can have an outsized impact on delivery: McKinsey describes a company that switched to cross-functional teams halfway through a project. Enabling “more rapid exchange of information, faster requirements clarifications, and speedier problem solving,” this change in ways of working resulted in a 45% decrease in code defects, less rework, and a 20% quicker time to market.

Microsoft's research on remote work during the pandemic also highlighted the critical importance of deliberate collaboration practices for maintaining engineering productivity.



ACTIONABLE INSIGHT:

Measure collaboration through metrics like code review response times, knowledge documentation contributions, and cross-team support activities. Create communication channels specifically designed for knowledge sharing, and recognize and reward collaborative behaviors that lift team performance.

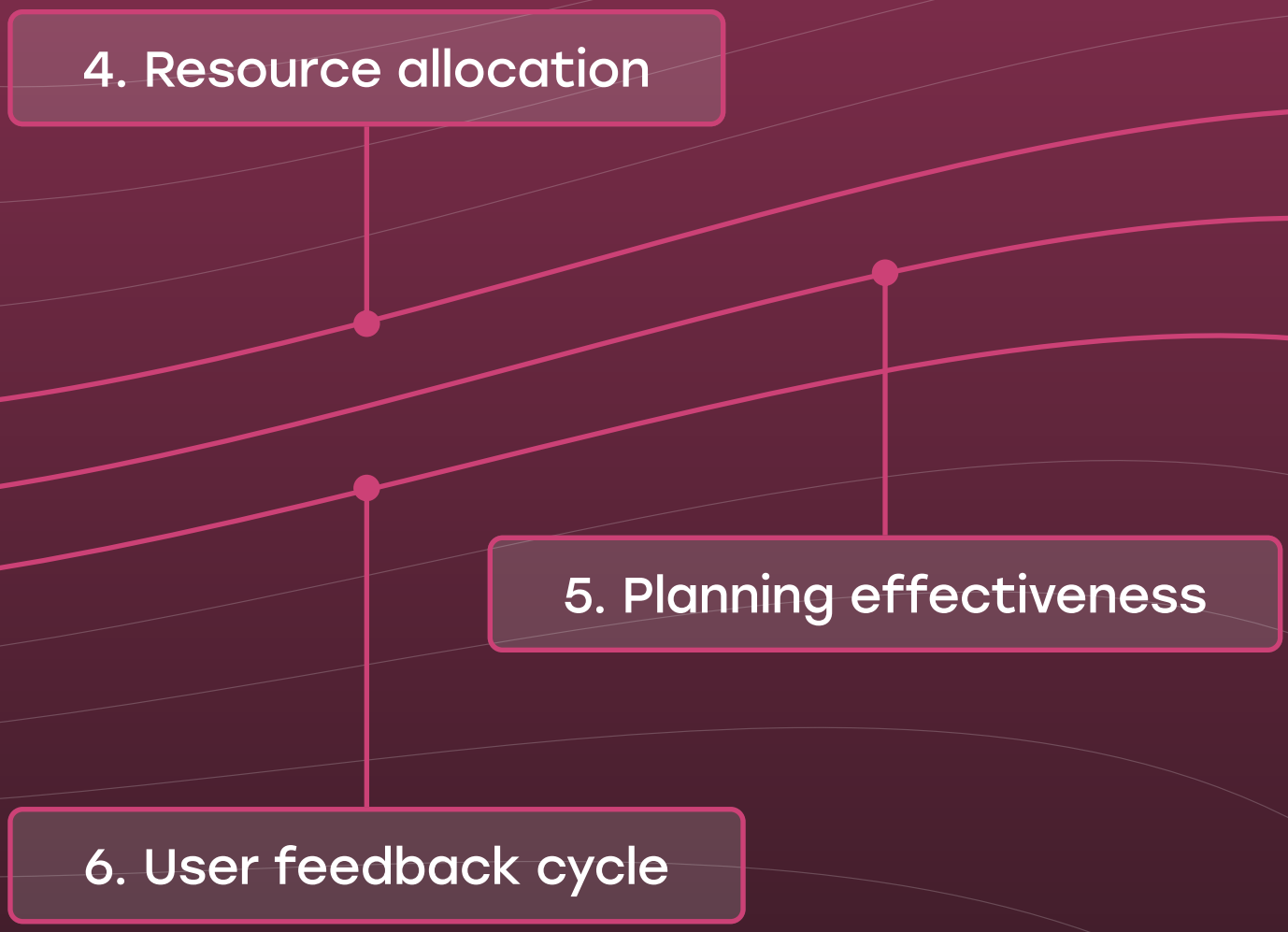
HOW UPLEVEL MEASURES COLLABORATION:

UPLEVEL PRODUCT **UPLEVEL METHOD**

In the product, our meeting classifier helps identify which meetings are working sessions and measures other leading indicators of collaboration like code review response times. We also evaluate collaboration qualitatively in developer surveys and stakeholder interviews.

Alignment: Planning and Resource Allocation

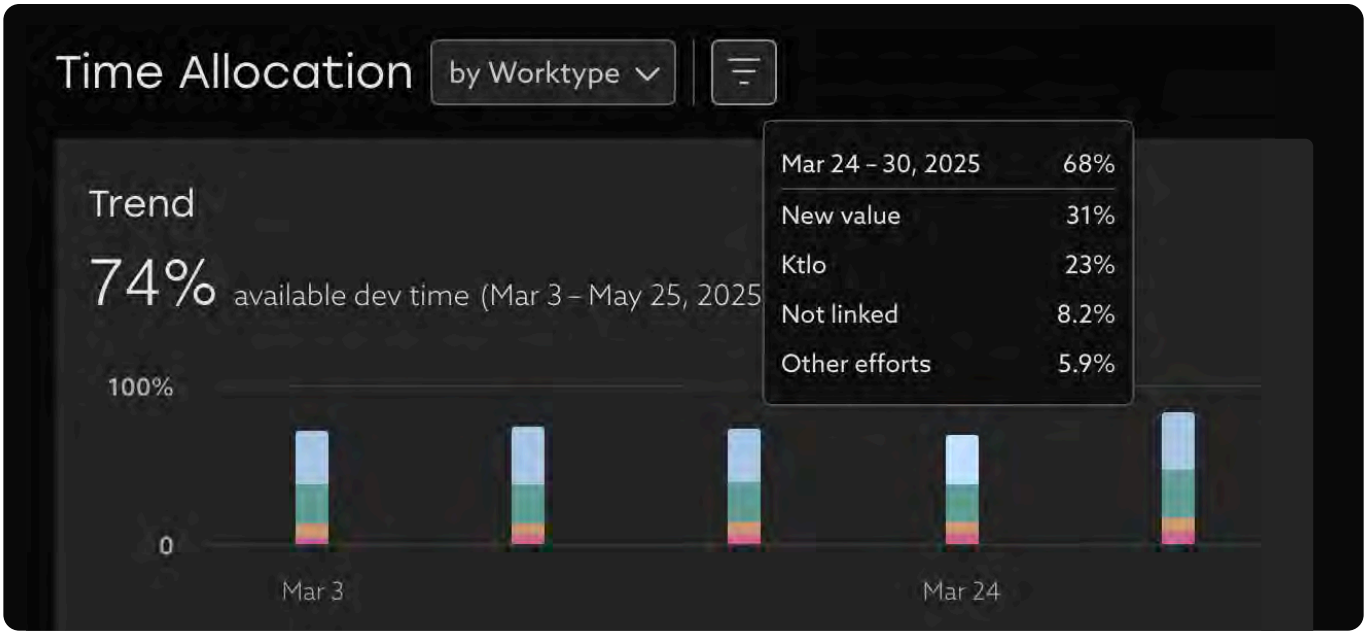
Alignment measures how well engineering efforts connect to business objectives through planning effectiveness, resource allocation, and feedback loops.



4. Resource allocation

Resource allocation metrics track the actual distribution of engineering effort across new value creation, technical debt, and maintenance work. Unlike self-reported time allocations, data-driven measurements provide an objective view of where engineering time is actually spent.

In most organizations, developers believe they spend more time on new features than they actually do when their work is objectively analyzed. Our own research puts the average time spent on new value creation at just under 20% – one day out of five.



ACTIONABLE INSIGHT:

Implement data-driven allocation tracking that objectively measures how engineering time is distributed. Consider adopting a formal budgeting approach for technical debt, setting explicit quarterly targets for system improvement alongside feature development.

HOW UPLEVEL MEASURES ALLOCATION:

UPLEVEL PRODUCT

Uplevel ingests data from your organization's dev and collaboration tools to surface a data-driven estimation of how developers spend their time.

Learn more about our allocation model >

5. Planning effectiveness

Planning effectiveness is a key enabler of value delivery in software engineering and product organizations because it reflects how well teams understand their work, capacity, and alignment with evolving priorities.

Metrics such as sprint completion rates (often referred to as the "say-do ratio") and requirements stability serve as proxies for this understanding. When teams consistently do what they say they'll do, it suggests a healthy balance between ambition and realism. Likewise, stable requirements indicate clarity in what needs to be built – minimizing churn and rework that delay value delivery.

As always, however, context matters. These metrics should not be treated as success criteria on their own. A high sprint completion rate, for instance, could mask underlying issues if teams are playing it safe by undercommitting, or if they are delivering work that is no longer relevant due to shifting priorities. Instead, planning effectiveness is a signal to detect misalignments in team capacity, requirement clarity, or cross-functional communication. When planning metrics fluctuate significantly, it may indicate that teams lack the information or autonomy needed to make reliable commitments, which can delay or derail the delivery of customer value.

ACTIONABLE INSIGHT:

Track sprint completion rates and requirements stability over time to identify patterns and root causes of planning issues. For teams with consistently low planning effectiveness, consider implementing techniques like "confidence voting" during estimation and "pre-mortem" exercises at the start of initiatives to surface potential risks early.

HOW UPLEVEL MEASURES PLANNING EFFECTIVENESS:

UPLEVEL METHOD

Planning effectiveness is evaluated qualitatively as part of the Uplevel Method. During the developer survey, participants are asked about the clarity of vision and priority of the work assigned to them.

6. User feedback cycle

How quickly do teams receive and incorporate user feedback after releasing features? Short user feedback cycles are a leading indicator of software engineering alignment to value because they create a continuous loop of validation between what is being built and what users actually need.

When feedback is rapid and frequent, teams can quickly confirm whether their work delivers meaningful outcomes, enabling faster course corrections and reducing the risk of building features that customers don't use. This responsiveness ensures that engineering efforts remain tightly coupled with business priorities, ultimately leading to higher-impact deliverables, better resource utilization, and increased customer satisfaction.

We find that it's one of the most underrated metrics – if your team doesn't get feedback or gets it too late, information is probably getting locked between departments.

ACTIONABLE INSIGHT:

Implement automated feedback collection mechanisms that capture user responses immediately after feature releases. Create feedback dashboards that make user responses visible to all engineers, not just product managers. Consider adopting techniques like outcome-based roadmaps that focus on user impact rather than feature completion.

HOW UPLEVEL MEASURES USER FEEDBACK CYCLE:

UPLEVEL METHOD

User feedback cycles are evaluated qualitatively as part of the Uplevel Method. Developers are asked how well they understand user needs and goals, and we dive deeper with stakeholders into practices around gathering and incorporating user feedback into engineering processes.

Velocity: Throughput and Lead Time Measurements

Velocity metrics focus on the movement of work through your engineering system, helping to identify bottlenecks and inefficiencies that slow delivery.

7. Cycle time and throughput

8. Work in Progress (WIP)

9. Lead time

7. Cycle time and throughput

PR cycle time measures how long it takes for pull requests to move from creation to deployment, while PR throughput tracks the volume of completed work.

When evaluating metrics like cycle time and throughput, it's tempting to compare teams against each other. However, this approach often leads to misleading conclusions because teams operate under different contexts – varying codebases, workflows, review cultures, and priorities. These factors make true apples-to-apples comparisons across teams nearly impossible.

Instead, comparing a team's current cycle time against its own historical performance offers a far better perspective. This approach allows leaders to:

- **Control for context:** Each team's structure, domain complexity, and work patterns remain relatively consistent over time, making internal trends more meaningful.
- **Identify real improvement:** By comparing against its own baseline, a team can detect genuine progress or regression and understand the impact of process changes or tooling.
- **Encourage healthy behaviors:** Cross-team comparisons can create unnecessary pressure to "compete" on metrics, potentially leading to metric gaming or unhealthy shortcuts. Longitudinal tracking fosters continuous improvement within the reality of each team's workflow.
- **Make metrics actionable:** Teams are more likely to trust and act on data that reflects their own experience rather than abstract benchmarks from other teams.

While some teams look at metrics like PR cycle time at the individual level, the most valuable insights come when these metrics are aggregated at the team level. This shifts the focus away from individual performance and toward systemic improvements that benefit the entire team's velocity and collaboration.

ACTIONABLE INSIGHT:

Establish baseline cycle time and throughput measurements for each team, then identify specific bottlenecks in their workflow. For teams with longer cycle times, examine code review processes, PR sizes, and automation levels. Consider using PR size limits and review service level agreements (SLAs) to improve flow.

HOW UPLEVEL MEASURES CYCLE TIME AND THROUGHPUT:

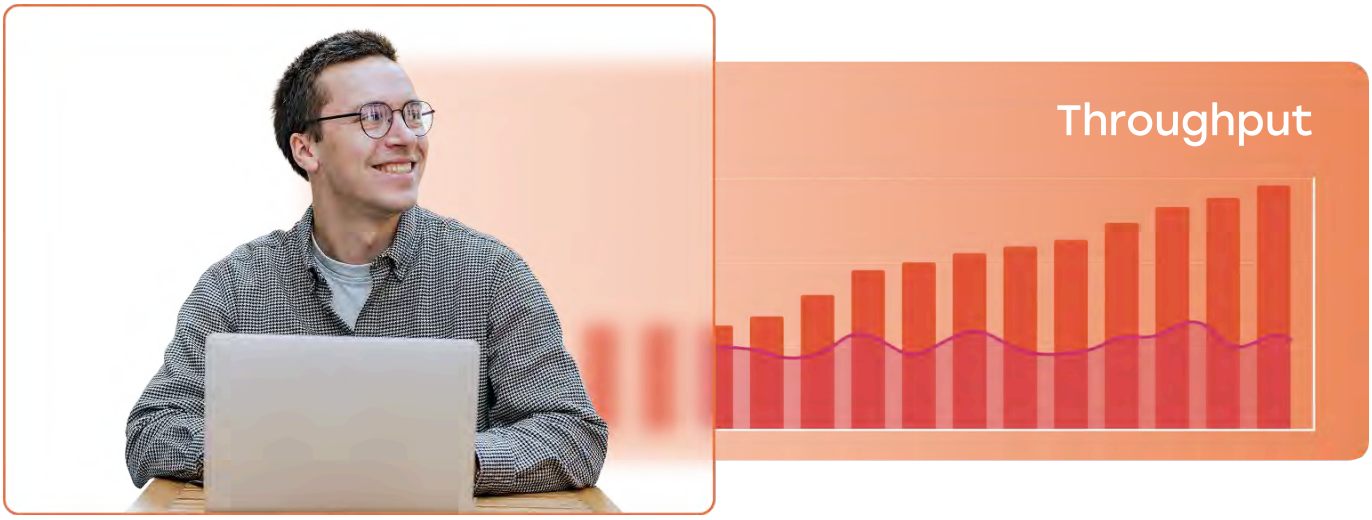
UPLEVEL PRODUCT

Uplevel surfaces PR cycle time and throughput within the platform, including additional context such as time spent in each phase (first commit, waiting, and review) and PR complexity.

8. Work in Progress (WIP)

Work-in-progress (WIP) metrics track how many concurrent items a team is working on at any time.

Organizations often hesitate to measure it because they want to create as much value as possible, but it's important to teams because they can sense when they are getting swamped with concurrent demands. When leaders fail at capacity planning, it's often because they don't take into account the constraints of WIP. High WIP levels mean more context switching, which in turn leads to decreased quality and productivity.



ACTIONABLE INSIGHT:

Implement explicit WIP limits based on team size and work complexity. Start by visualizing all in-progress work to create awareness, then gradually reduce limits. For teams with chronically high WIP, examine upstream processes that may be pushing too much work simultaneously, such as planning practices or stakeholder management approaches.

HOW UPLEVEL MEASURES WIP:

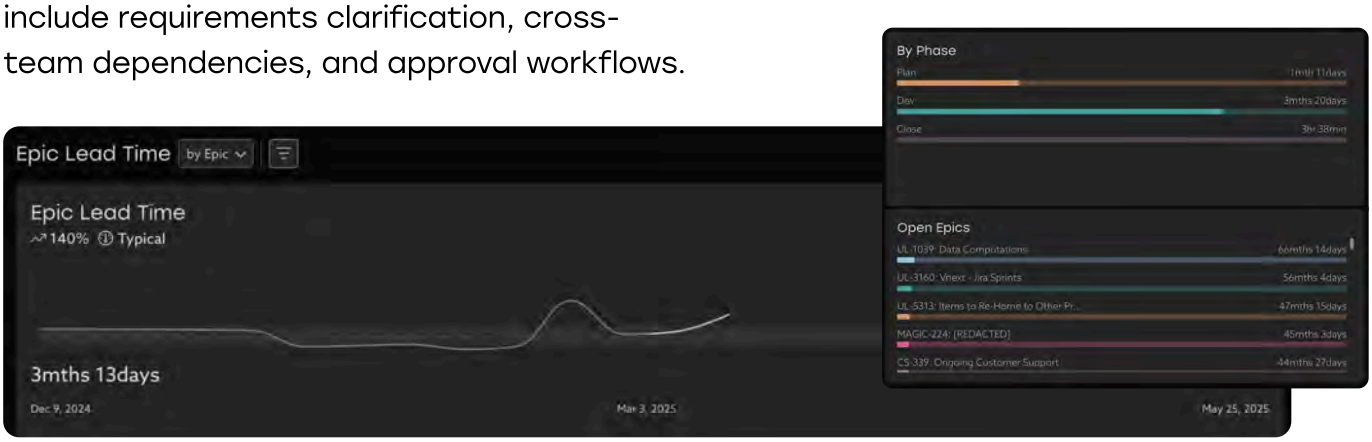
UPLEVEL METHOD

During the assessment portion of the Uplevel method, we enrich data from your dev tools to reveal a holistic picture of your organization's work in progress.

9. Lead time

Epic lead time measures how long it takes to deliver meaningful business value from initial concept to production. Lead time is distinct from cycle time in that it captures the entire journey from concept to customer value, not just the development portion. This metric helps engineering leaders identify systemic bottlenecks in the end-to-end value stream – which is important, as optimizing individual steps without addressing the entire flow often leads to local efficiencies but global ineffectiveness. Some hidden work categories that other metrics might miss include requirements clarification, cross-team dependencies, and approval workflows.

Engineering leaders should track epic lead time variability alongside mean performance to identify process instability and planning risks. High variability signals unpredictable delivery capability, making sprint commitments unreliable and resource allocation inefficient. This data enables targeted process improvements and helps engineering leaders make data-driven arguments for removing organizational impediments that development teams cannot address independently.



ACTIONABLE INSIGHT:

Map your entire value stream from concept to customer to identify where work spends most of its time. Pay particular attention to handoffs between teams and waiting periods. For organizations with long lead times, consider implementing cross-functional teams that can own entire features end-to-end, reducing handoff delays and communication overhead.

HOW UPLEVEL MEASURES LEAD TIME:

UPLEVEL PRODUCT **UPLEVEL METHOD**

Uplevel quantifies epic lead time within the platform and drills deeper into root cause analysis of systemic bottlenecks during the Uplevel Method assessment.

Environment Efficiency: Quality and Flow

Environment efficiency measures how well your engineering ecosystem supports productive work and quality outcomes. These metrics help identify structural impediments to effectiveness that exist beyond individual teams.

10. Quality metrics

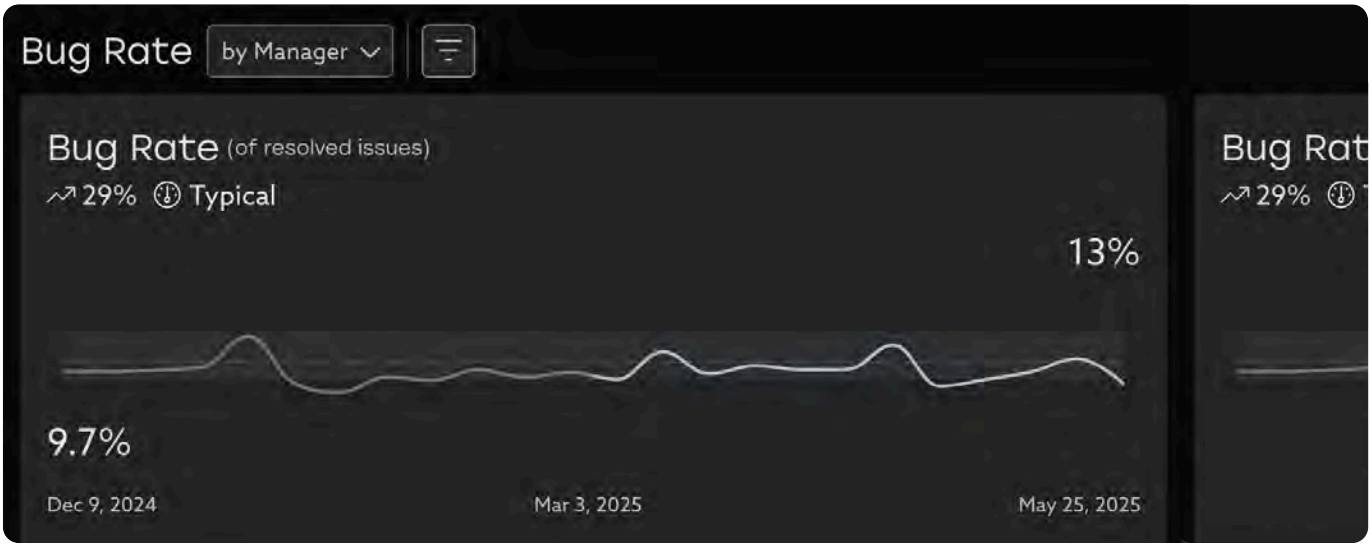
11. DORA Metrics within a Broader Context

12. Friction and flow

10. Quality metrics

Quality metrics include defects by type (bugs vs. escaped defects) and defects found by stage in the development process. It's more focused on outcomes rather than minute processes or output. For example, if your team's code has too many bugs or isn't working properly, you need to understand why that's happening in the first place. Is it because of high WIP, limited time, or too much context switching?

Detecting defects earlier in the development process reduces the cost of remediation by orders of magnitude – defects found in production can cost 100x more to fix than those found during code review.



ACTIONABLE INSIGHT:

Map defects back to their source components and development stages to identify systemic quality issues. Implement structured defect causal analysis sessions to identify and address root causes rather than just symptoms. Consider adopting techniques like "shift-left testing" that catch defects earlier in the SDLC.

HOW UPLEVEL MEASURES QUALITY:

UPLEVEL PRODUCT **UPLEVEL METHOD**

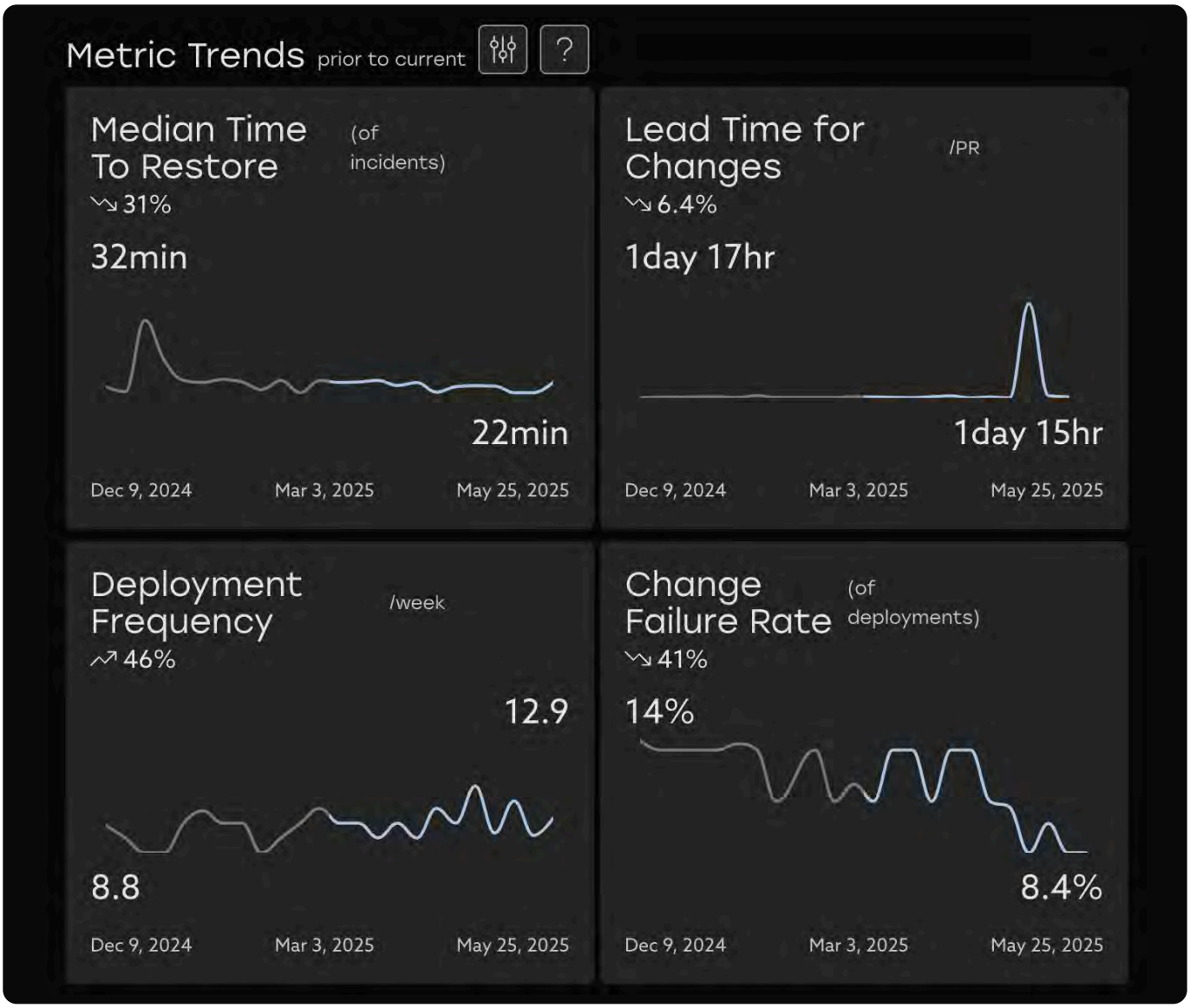
Uplevel surfaces bug rate and change failure rate within the platform. Additional context around specific code quality issues is included in the Uplevel Method assessment survey.

11. DORA Metrics within a Broader Context

DORA metrics (deployment frequency, lead time for changes, change failure rate, and time to restore service) provide valuable deployment pipeline insights, but they're most valuable when contextualized within the broader WAVE framework.

Software delivery performance metric	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one hour	Between one day and one week	Between one month and six months	More than six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Between one day and one week	More than six months
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0%-15%	16%-30%	16%-30%	16%-30%

The DORA research team's own annual State of DevOps reports consistently show that technical practices alone are insufficient for high performance. Their research identifies organizational factors, leadership, and culture as critical elements that enable technical excellence. The 2023 report specifically highlighted that elite performers excel not just in deployment metrics but in how they build healthy engineering cultures.



ACTIONABLE INSIGHT:

Instead of treating DORA metrics as standalone goals, use them as diagnostic tools within the broader sociotechnical system. When a DORA metric shows concerning trends, investigate whether the root cause lies in Ways of Working, Alignment, or other dimensions before implementing technical solutions.

HOW UPLEVEL MEASURES DORA:

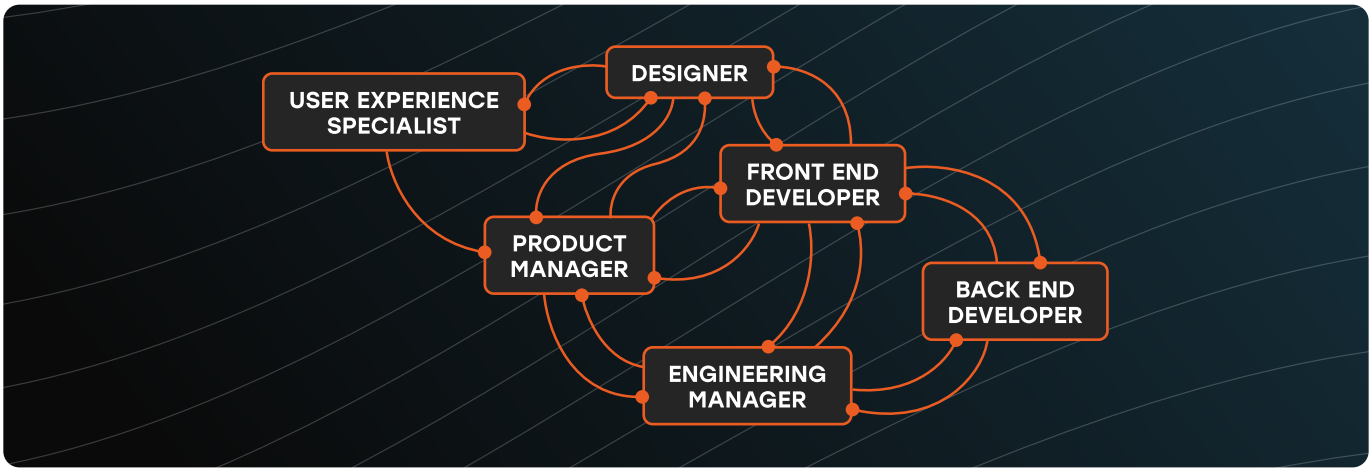
UPLEVEL PRODUCT

Uplevel collects developer-reported data on friction and flow as part of the assessment survey.

12. Friction and flow

Friction scores measure developer-reported friction in the development process, while flow efficiency calculates the ratio of waiting time to total cycle time. These metrics help identify organizational and process bottlenecks that slow delivery.

Friction is usually measured qualitatively through microsurveys. However, flow efficiency is measured by understanding the percentage of the cycle spent waiting. In knowledge work, including software development, items typically spend 70-85% of the time waiting rather than being actively worked on (a flow efficiency rate of 15%). Organizations applying Lean flow principles to software development have demonstrated significant improvements in both delivery speed and quality.



ACTIONABLE INSIGHT:

Map your value stream to identify where work waits the longest. Toyota's "improvement kata" approach, adapted for software by Mike Rother, provides a structured method for addressing flow problems. Focus first on making work visible, then on limiting work-in-progress, and finally on addressing specific bottlenecks in your development process.

HOW UPLEVEL MEASURES FRICTION AND FLOW:

UPLEVEL METHOD

Uplevel collects developer-reported data on friction and flow as part of the assessment survey.

How Uplevel helps engineering leaders implement WAVE

Implementing the WAVE framework doesn’t stop at collecting better metrics. The real change lies in how engineering organizations understand and improve themselves. Sustainable transformation requires both measurement systems and enabling mechanisms for improvement.

As Francisco Trindade, VP of Engineering at Braze, explains: "Having data helps the conversations I have with teams. 'You didn't work on these goals this quarter. Why was that? What can we do to increase the time you're delivering value?' Then we can take action. So that's a lot of the work we're doing with Uplevel."

The WAVE Framework reveals three critical insights about engineering effectiveness that align with current research on high-performing engineering organizations:

1. **Engineering improvement is iterative, not linear.** The concept of iterative improvement has strong roots in both Agile methodologies and Toyota's Kaizen philosophy. Organizations make the most progress when they identify the most impactful dimension, improve it, reassess, and continue this cycle.

2. **The highest leverage improvements often cross organizational boundaries.** Research from Frost & Sullivan found that collaboration is the strongest driver of business performance, accounting for 36% of a company's overall performance. Collaboration showed particularly strong effects on customer satisfaction (41% impact), labor productivity (36%), and financial metrics including profitability (29%), profit growth (26%), and sales growth (27%).

3. **Sustainable improvement requires both measurement and enablement.** Measurement without the capability to change yields little benefit. Organizations need both insights and methods to implement improvements.

As you consider your own engineering organization's effectiveness, ask yourself:

- **Do you have visibility into all four WAVE dimensions?**
- **Can you identify which dimensions currently limit your organization's performance?**
- **And most importantly, do you have a methodology to turn those insights into sustainable improvement?**

Uplevel is the holistic engineering optimization system that makes it easier for tech leaders and their teams to deliver impact.

See it in action >

Uplevel