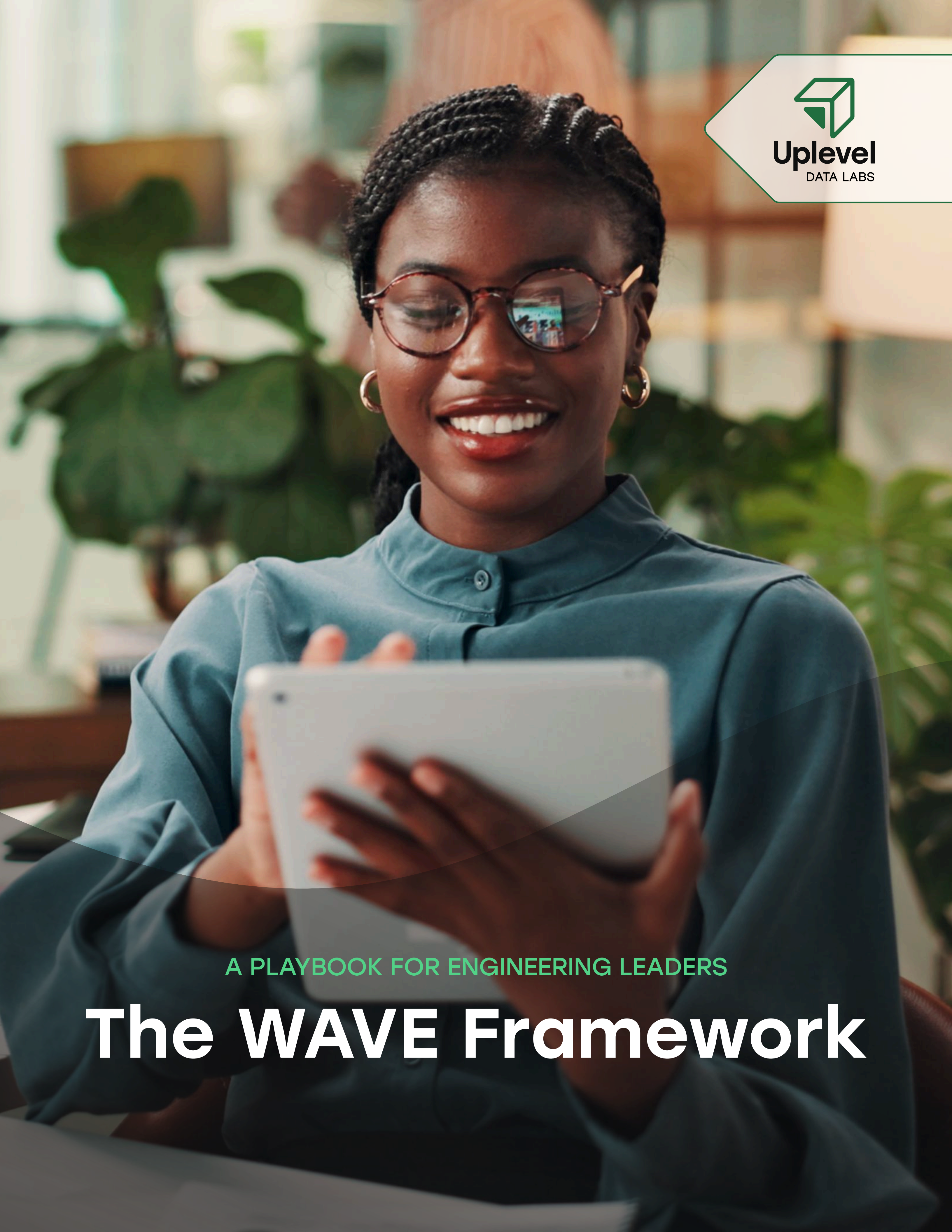


Uplevel
DATA LABS

A PLAYBOOK FOR ENGINEERING LEADERS

The WAVE Framework

Table of contents

The measurement trap 2

What's the problem with traditional engineering KPIs? 3

The WAVE Framework: A Holistic Approach to Engineering Effectiveness 4

Ways of Working: Cultural Factors That Enable Delivery 6

Deep work 7

Team health 8

AI maturity 9

Alignment: Connecting Engineering to Business Value 10

Allocation 11

Planning 12

User alignment 13

Velocity: Throughput and Lead Time Measurements 14

Velocity score 15

Handoffs 16

PR reviews 17

Environment Efficiency: Quality and Flow 18

Recovery 19

Code quality 21

Friction 22

How Uplevel helps engineering leaders implement WAVE 23

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.

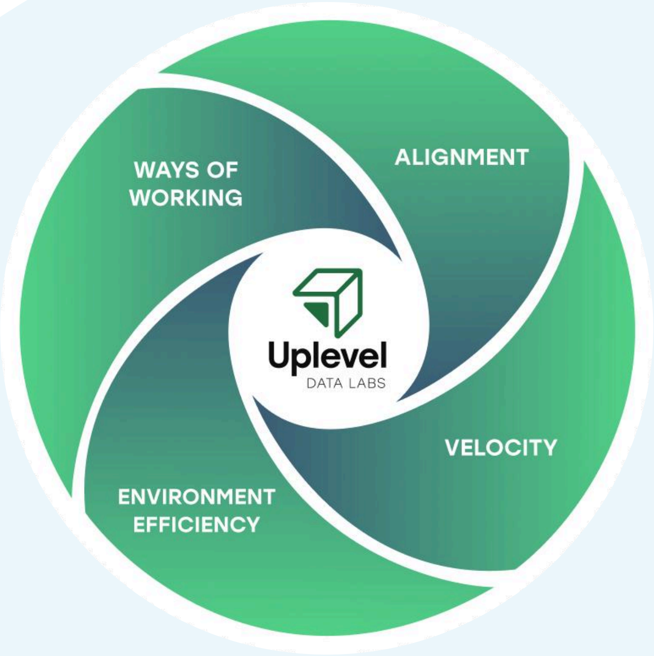
The WAVE Framework: A Holistic Approach to Engineering Effectiveness

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 cultural elements that enable delivery
- **Alignment (A):** Captures how well engineering efforts connect to business value
- **Velocity (V):** Tracks the flow of work
- **Environment Efficiency (E):** Evaluates system quality and friction



Each dimension of WAVE is summarized by a key lagging indicator in that area. This metric is an **output,** an outcome of the inputs that enable good engineering. These leading indicators for performance are captured in the lower section of the table below.

For example, improving **team health inputs** like psychological safety, meeting cadence, and mission alignment and improving **AI perception inputs** like guideline clarity and best practices will naturally result in less overwhelm and more time developers spend in **deep work.**

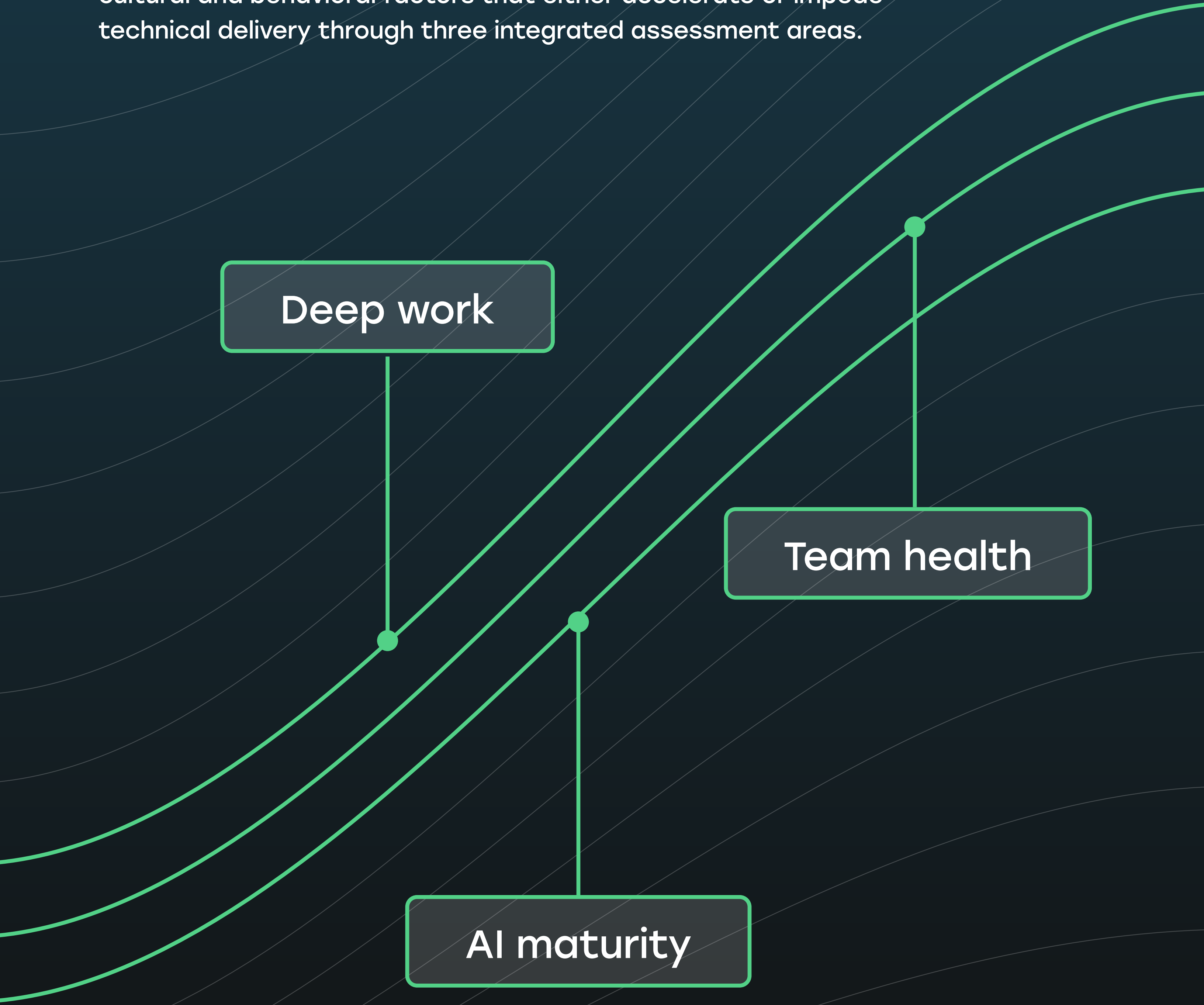
WAYS OF WORKING	ALIGNMENT	VELOCITY	ENVIRONMENT EFFICIENCY
KEY METRIC (LAGGING INDICATORS/OUTPUTS)			
DEEP WORK (page 7) Average deep work hours per day	**ALLOCATION OF EFFORT** (page 11) Allocation to new value vs. allocation to tech debt	**VELOCITY SCORE** (page 15) PR cycle time, issue velocity, and PR velocity	**RECOVERY** (page 19) Mean time to recover and change failure rate
SUPPORTING METRICS (LEADING INDICATORS/INPUTS)			
TEAM HEALTH (page 8) Composite score assessing cultural collaboration factors	**PLANNING** (page 12) Sprint completion, requirements churn, and prioritization	**HANDOFFS** (page 16) Handoff frequency and quality through workflow analysis	**CODE QUALITY** (page 21) Bug rate, code complexity, and support escalations
AI MATURITY (page 9) AI productivity impact, processes, and tooling	**USER ALIGNMENT** (page 13) User feedback cycle time	**PR REVIEWS** (page 17) PR complexity and quality, PR review time	**FRICTION** (page 22) Friction score and waiting/total ratio

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.

Ways of Working: Cultural Factors That Enable Delivery

The Ways of Working dimension recognizes that engineering performance begins with people and team dynamics. This component measures the cultural and behavioral factors that either accelerate or impede technical delivery through three integrated assessment areas.



Deep work

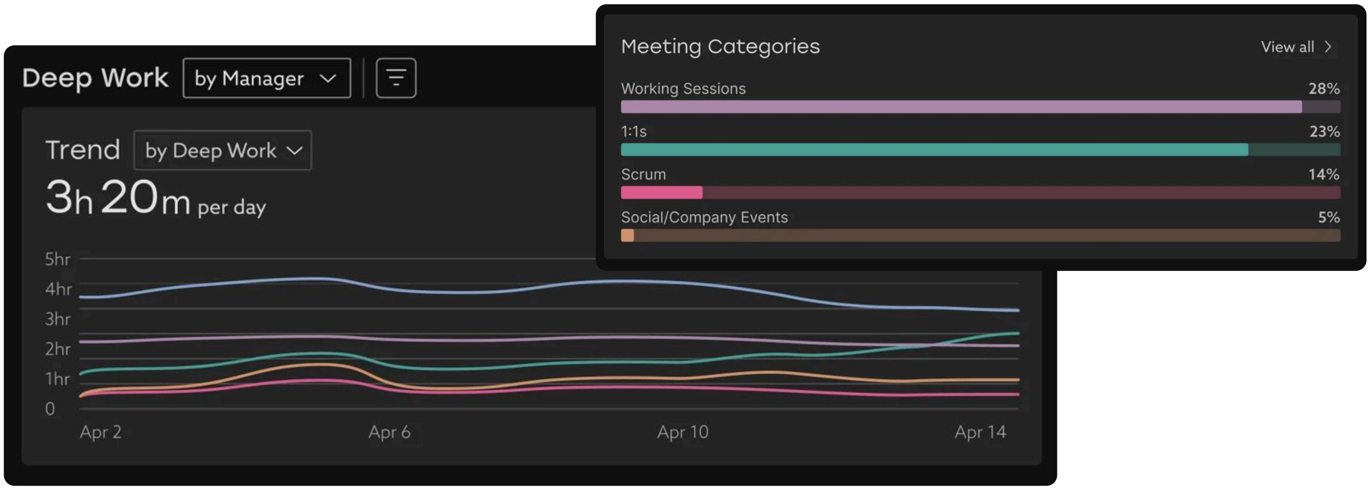
Team health

AI maturity

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.



Meeting Categories	
Working Sessions	28%
1:1s	23%
Scrum	14%
Social/Company Events	5%

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:

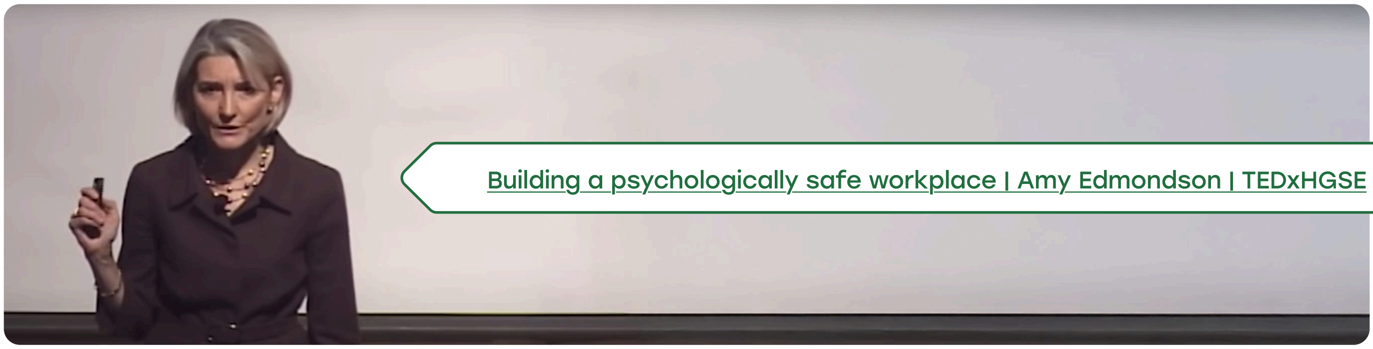
UPLEVEL PRODUCT

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.

Team health

Uplevel’s team health index consolidates mission alignment, psychological safety, collaboration effectiveness, meeting efficiency, and clarity of ownership into a unified score. These elements are interconnected: teams with clear alignment demonstrate higher psychological safety, enabling more effective collaboration.

Google's Project Aristotle research identified psychological safety as the primary factor distinguishing high-performing teams. 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."



Building a psychologically safe workplace | Amy Edmondson | TEDxHGSE

When you track team health over time, you 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 the full spectrum of team dynamics. Create dedicated time for teams to discuss and address issues identified through these assessments.

HOW UPLEVEL MEASURES TEAM HEALTH:

UPLEVEL PRODUCT **UPLEVEL METHOD**

Qualitative interviews and surveys capture psychological safety and team health as part of the 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.

AI maturity

AI maturity captures organizational readiness for artificial intelligence adoption through standard operating procedures, tool coherence, and leadership clarity. As AI performance compounds at 5x every two years (Huang's Law), understanding AI's impact on ways of working becomes critical for organizations seeking competitive advantage.

Organizations and teams with high AI maturity establish clear SOPs for tool usage, identify best practices and recommended use cases, ensure compliance and security, maintain coherent toolsets, and provide unambiguous leadership direction about AI strategy. This creates effective ways of working where the focus remains value creation rather than navigating organizational confusion. Teams with low maturity experience organizational drag, wasting time on tool decisions and worrying about compliance rather than leveraging AI for productivity gains.

ACTIONABLE INSIGHT:

Regularly survey teams about AI tool effectiveness and adoption barriers. Focus on creating coherent AI strategies with clear implementation guidelines rather than allowing ad-hoc adoption that creates organizational friction.

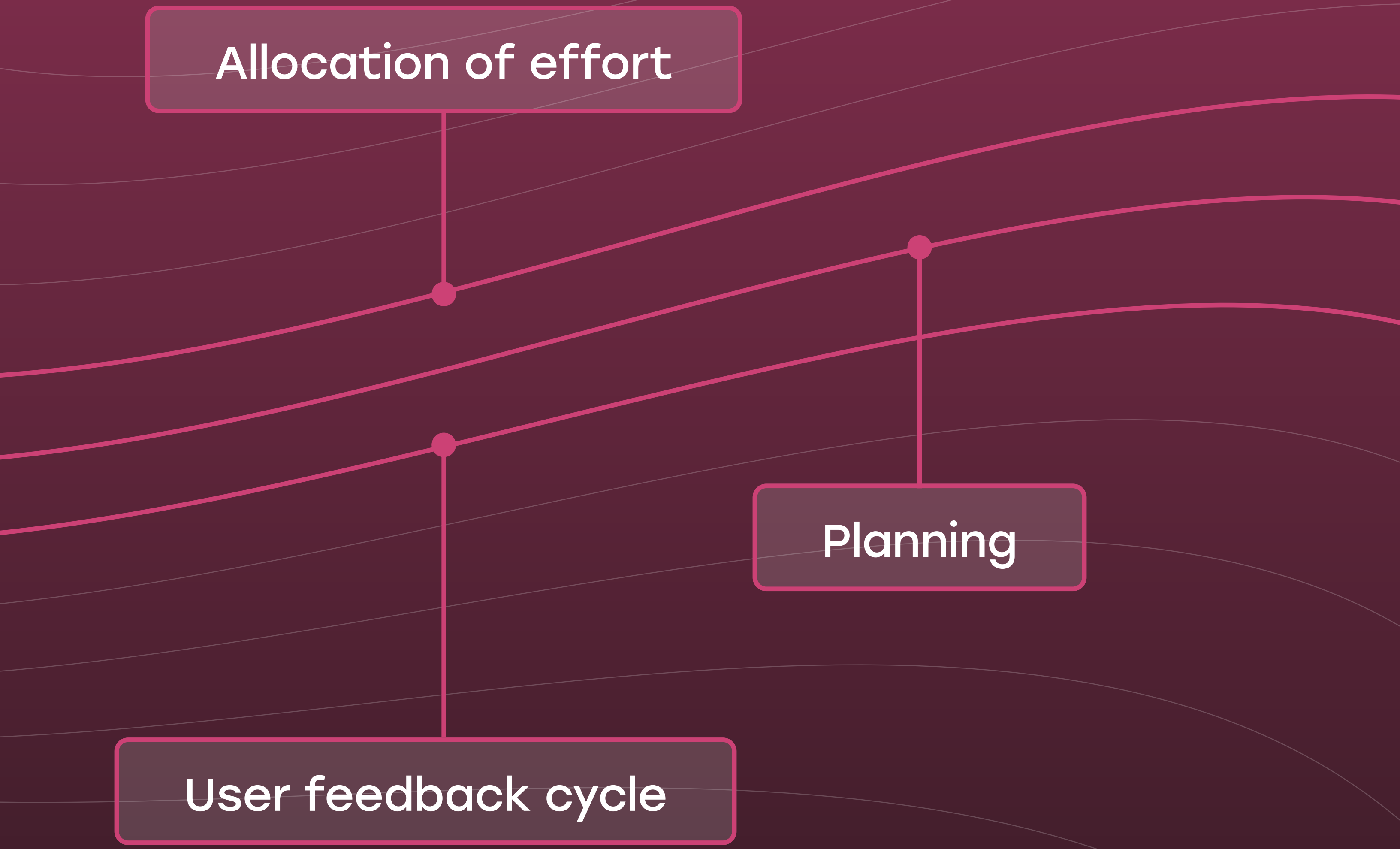
HOW UPLEVEL MEASURES AI MATURITY:

UPLEVEL METHOD

Uplevel evaluates AI maturity qualitatively through developer surveys and stakeholder interviews, capturing organizational readiness, tool standardization, and leadership clarity around implementation strategies.

Alignment: Connecting Engineering to Business Value

Alignment determines whether engineering capacity translates into meaningful business outcomes. This dimension exposes the gap between what teams build and what drives actual value creation.



Allocation of effort

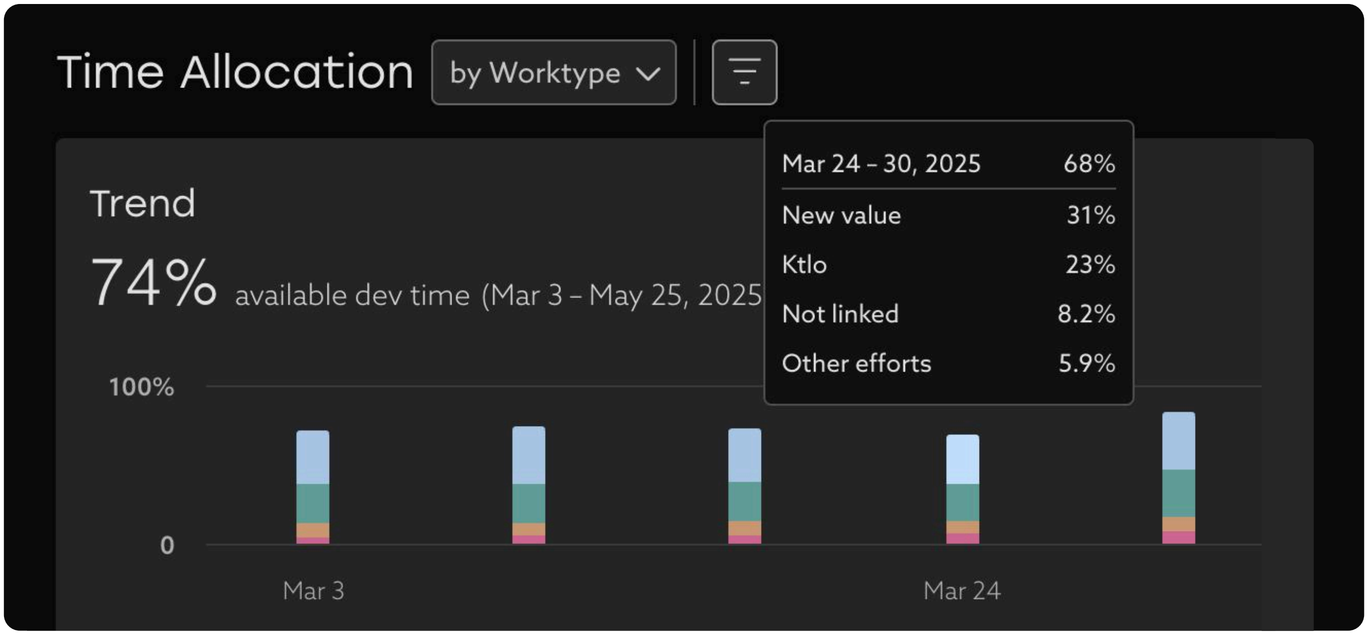
Planning

User feedback cycle

Allocation of effort

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.



Mar 24 – 30, 2025	68%
New value	31%
Ktlo	23%
Not linked	8.2%
Other efforts	5.9%

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 >

Planning effectiveness score

Planning effectiveness reflects how well teams understand their work, capacity, and alignment with evolving priorities through requirements churn, clarity of prioritization, connection to business value, epic lead time, and plan phase duration.

When teams consistently deliver what they commit to, it suggests a healthy balance between ambition and realism. 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 UPLEVEL PRODUCT

Some dimensions of planning effectiveness are 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. More concrete metrics like epic lead time and plan phase duration can be measured within the Uplevel platform.

User alignment

Uplevel's user feedback cycle score measures how quickly teams receive and incorporate user feedback after releasing features through frequency and type of user engagement, user feedback cycle time, and customer satisfaction scores.

Short user feedback cycles are a leading indicator of 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 confirm whether their work delivers meaningful outcomes, enabling faster course corrections.

We find this is 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.

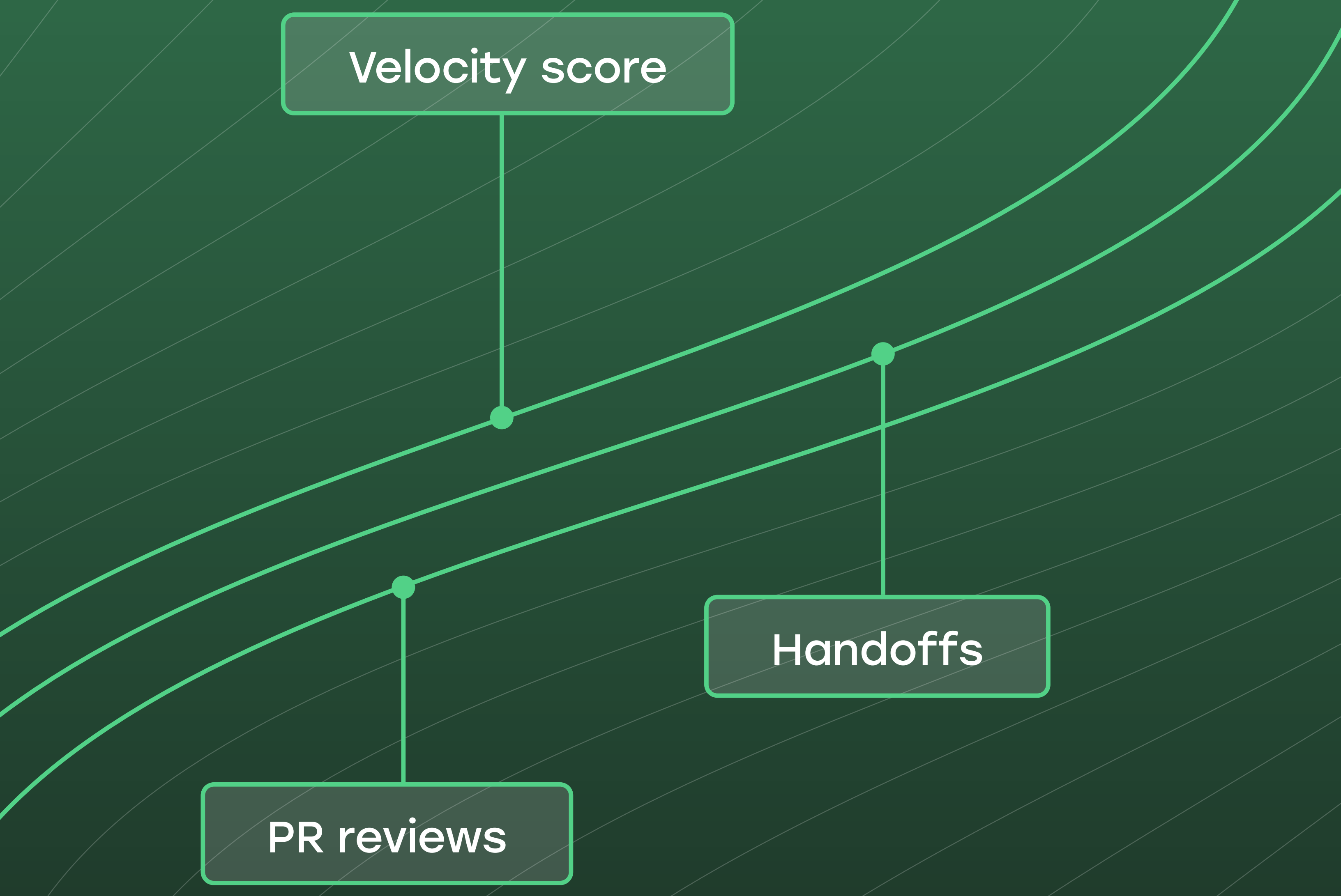
HOW UPLEVEL MEASURES USER FEEDBACK CYCLE:

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 measures how efficiently work moves through your engineering system. True velocity assessment requires understanding both throughput rates and the friction points that create delays, bottlenecks, and coordination overhead in your development process.



Velocity score

Uplevel's velocity score integrates PR cycle time, PR velocity, issue velocity, and deployment frequency (where available) into a comprehensive throughput measurement. This consolidated view reveals whether teams can consistently deliver completed work rather than just generate activity.

When evaluating velocity metrics, avoid comparing teams against each other. Teams operate under different contexts – varying codebases, workflows, review cultures, and priorities make cross-team comparisons misleading. Instead, compare each team's current performance against its own historical baseline to identify genuine improvement opportunities.

The most valuable insights emerge from team-level aggregation rather than individual tracking, shifting focus toward systemic improvements that benefit collective 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 VELOCITY:

UPLEVEL PRODUCT

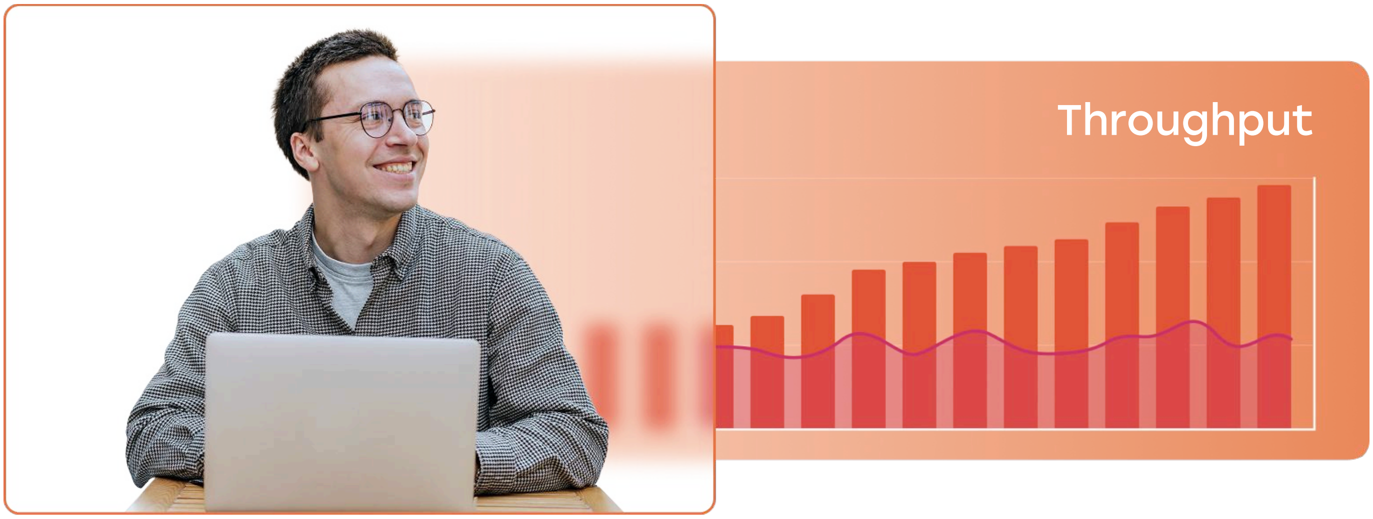
Uplevel aggregates common velocity measures into one composite score and surfaces additional context such as time spent in each phase (first commit, waiting, and review) and PR complexity.

Handoffs

Handoff metrics evaluate both frequency and quality of work transitions between teams, individuals, or process stages. Each handoff introduces coordination overhead and potential communication gaps that slow delivery and increase error rates.

Research shows that minimizing handoffs through cross-functional teams can result in significant improvements – one McKinsey study documented a 45% decrease in code defects and 20% faster time to market after switching to cross-functional teams that reduced coordination dependencies.

Most organizations underestimate handoff costs because the delays appear as waiting time rather than active work, making them invisible in traditional productivity measurements.



ACTIONABLE INSIGHT:

Map all handoffs in your value stream and measure both frequency and quality. Consider implementing cross-functional teams that can own entire features end-to-end, reducing handoff delays. Focus first on eliminating handoffs that require the most coordination time or create the highest error rates.

HOW UPLEVEL MEASURES HANDOFFS:

UPLEVEL METHOD

Handoff frequency and quality are evaluated through workflow analysis and qualitative assessment as part of the Uplevel Method, identifying coordination patterns that create delivery delays.

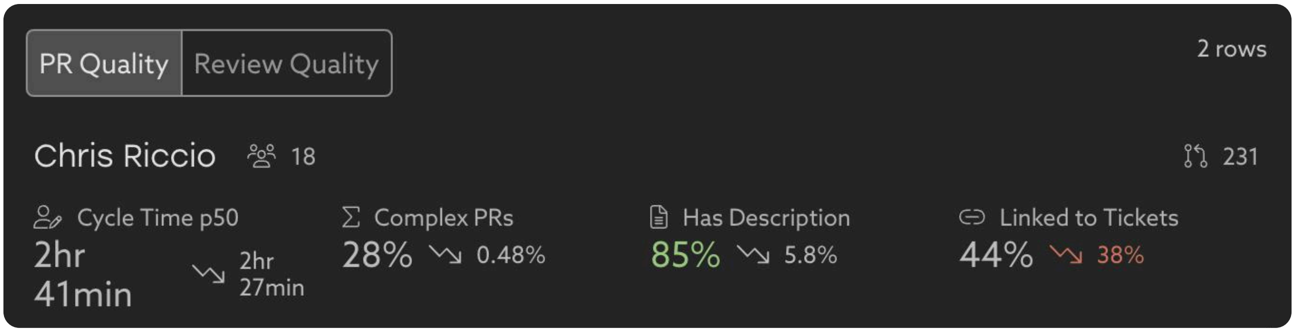
PR reviews

PR review metrics examine PR complexity, PR quality, and review quality/time to assess both work structure and collaborative effectiveness. This measurement reveals whether teams create reviewable code changes and conduct meaningful peer evaluation.

PR complexity tracks oversized changes that create review bottlenecks. Large PRs are harder to review thoroughly, increasing defect rates and cycle times. PR quality measures whether changes include proper descriptions, link to tracking systems, and maintain reasonable cycle times.

Review quality evaluates whether the collaborative process catches meaningful problems. Effective reviews identify functional defects and architectural issues when remediation costs are lowest, not just style preferences that automated tools can address.

Teams with high PR quality and effective review processes ship faster with fewer production issues. Those with poor practices accumulate technical debt and spend more time fixing downstream problems.



Chris Riccio 18 | 231

Cycle Time p50: 2hr 41min ↘ 2hr 27min

Complex PRs: 28% ↘ 0.48%

Has Description: 85% ↘ 5.8%

Linked to Tickets: 44% ↘ 38%

ACTIONABLE INSIGHT:

Set PR size limits to maintain reviewability and establish standards for descriptions and ticket linking. Track cycle times alongside review thoroughness to balance speed with quality.

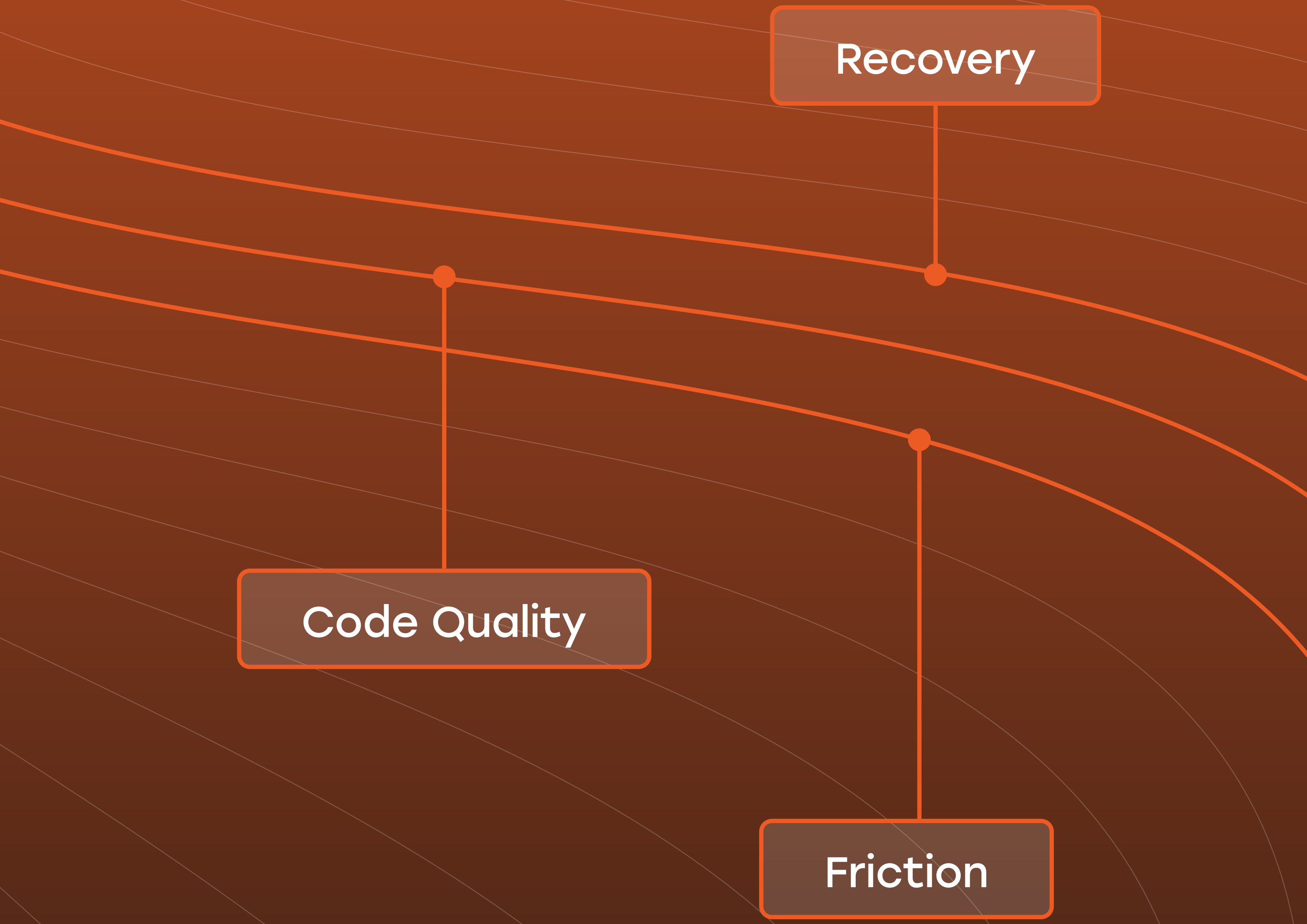
HOW UPLEVEL MEASURES PR REVIEWS:

UPLEVEL PRODUCT

Uplevel tracks PR complexity, cycle times, description quality, and ticket linkage patterns alongside review response times, identifying opportunities to optimize both code creation discipline and collaborative review effectiveness.

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.



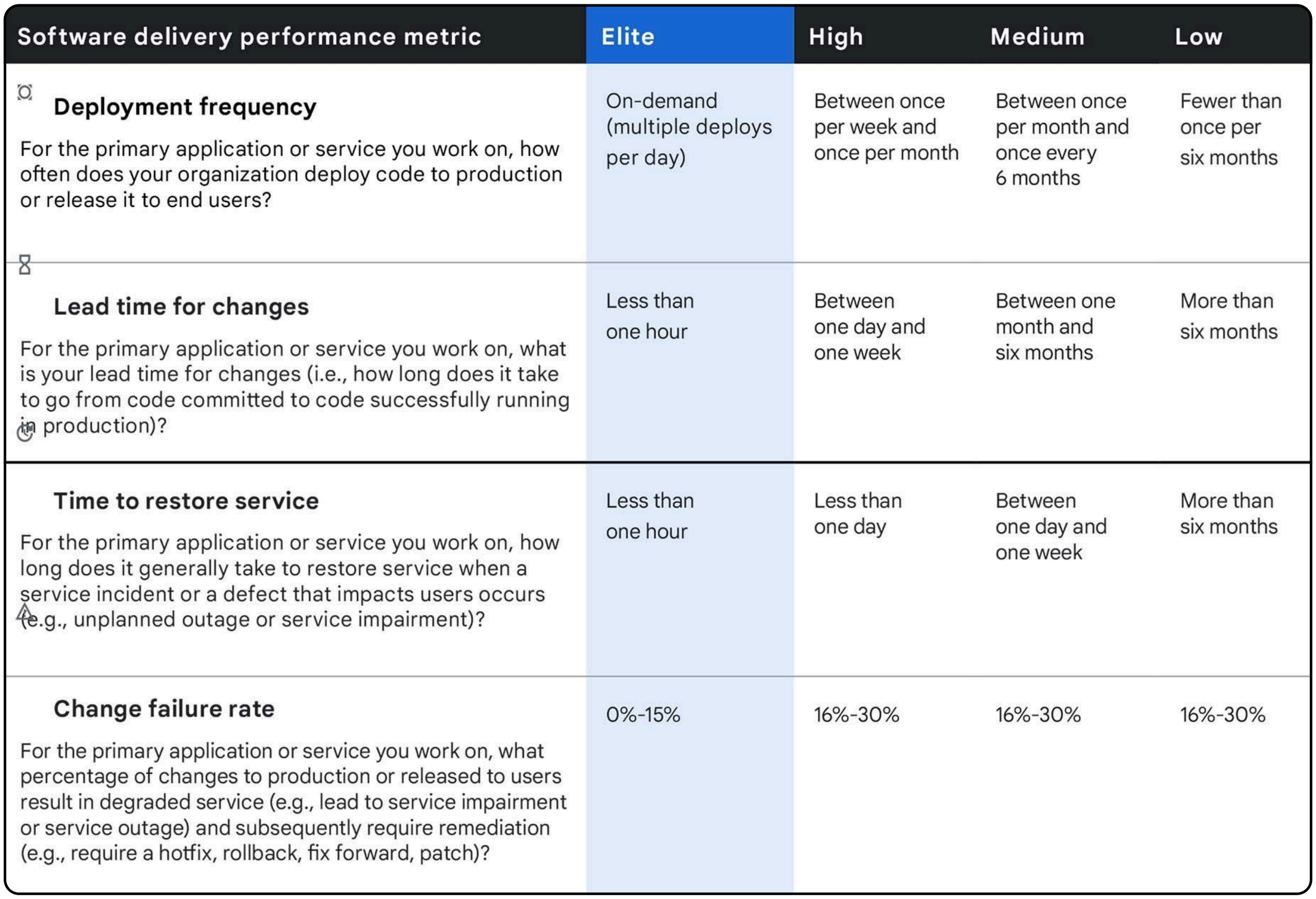
Recovery

Code Quality

Friction

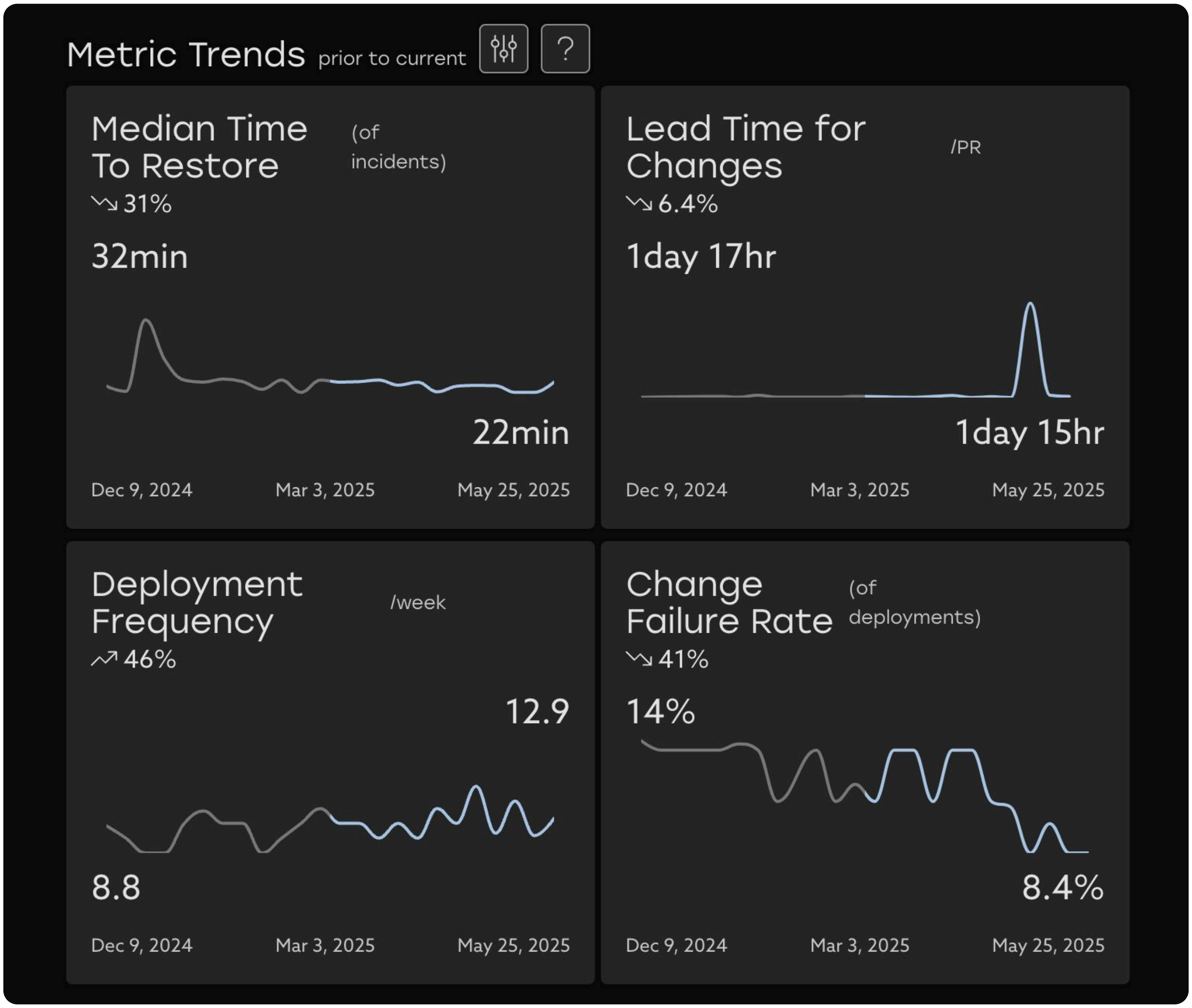
Recovery

Recovery metrics integrate lead time for changes, change failure rate, and mean time to repair (MTTR) – three of the four DORA metrics – to assess system resilience. These measurements reveal how quickly teams can deploy fixes and maintain stability under operational pressure.



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%

Organizations with faster recovery capabilities demonstrate robust testing, monitoring, and deployment automation that enable rapid issue detection and resolution. The 2023 DORA report specifically highlighted that elite performers excel not just in deployment metrics but in building cultures that support sustainable delivery.



ACTIONABLE INSIGHT:

Instead of treating recovery 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:

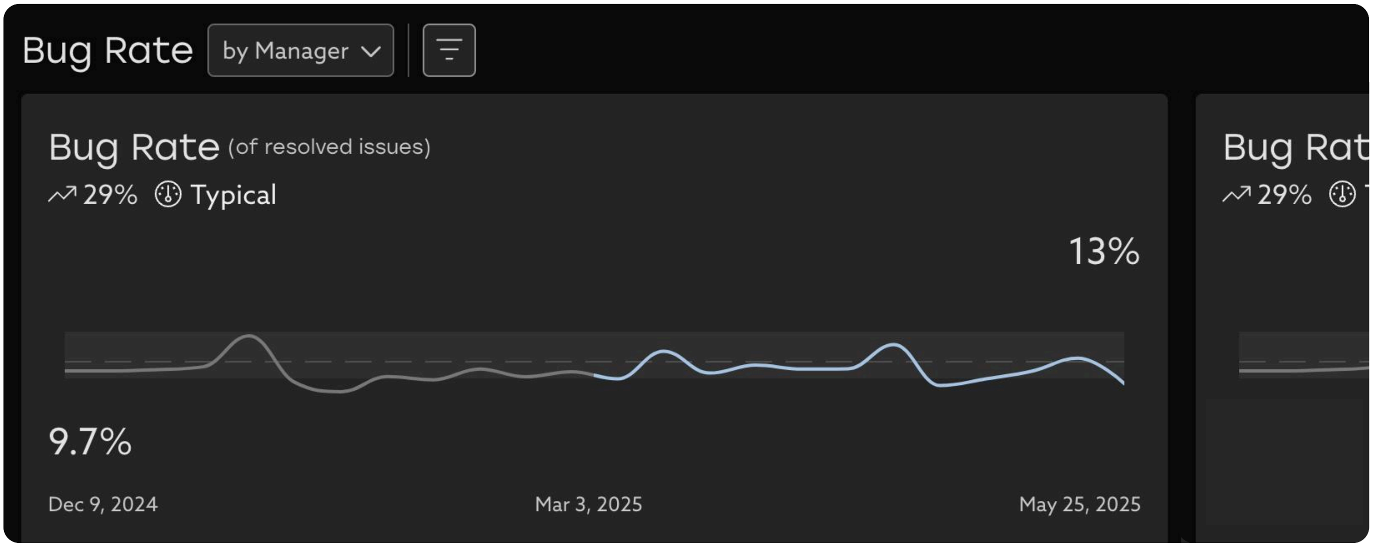
UPLEVEL PRODUCT

Uplevel supports core DORA metrics including lead time for changes, change failure rate, and time to restore service, plus supporting metrics that serve as leading indicators for deployment pipeline health.

Code quality

Code quality consolidates bug rates, customer-found defects, cyclomatic complexity, and support escalations into integrated quality assessment. This recognizes that quality issues compound—high complexity increases bugs, driving support escalations and customer-found defects.

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, making upstream quality investments essential for sustainable delivery.



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:

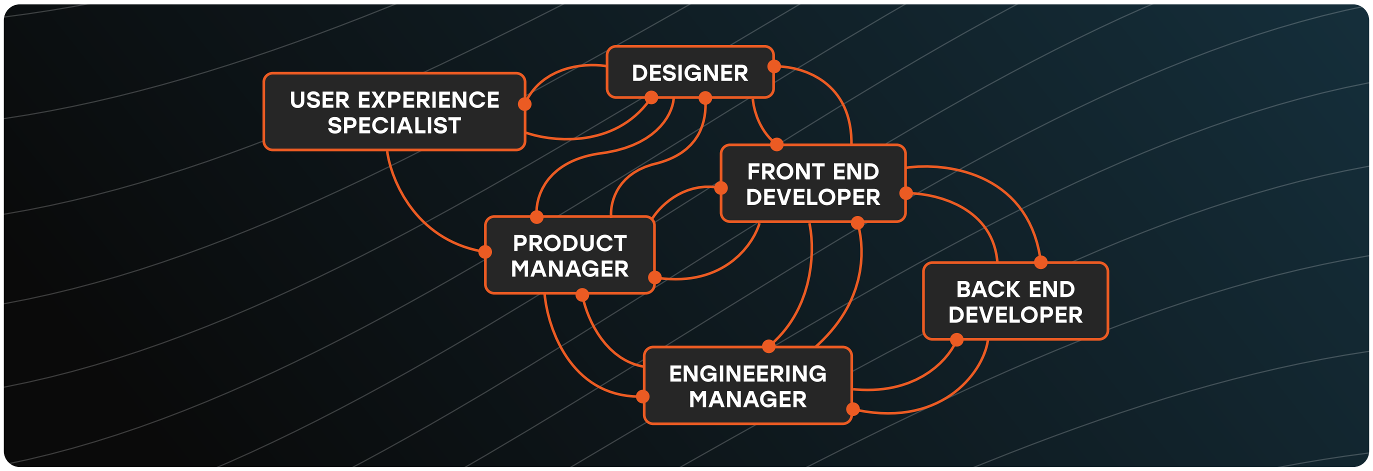
UPLEVEL PRODUCT UPLEVEL METHOD

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.

Friction

Friction measures systemic obstacles through architecture complexity, tooling effectiveness, deployment processes, and flow optimization. These factors determine organizational drag that slows delivery regardless of team capabilities.

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%). This represents massive efficiency opportunities most organizations ignore because waiting appears as white space rather than visible inefficiency.



ACTIONABLE INSIGHT:

Map your value stream to identify where work waits longest. Focus first on making work visible, then limiting work-in-progress, finally addressing specific bottlenecks. Apply systematic improvement approaches to reduce friction in development processes.

HOW UPLEVEL MEASURES FRICTION:

UPLEVEL METHOD

Uplevel collects developer-reported friction data and flow efficiency metrics through assessment surveys, identifying organizational and process bottlenecks that constrain delivery effectiveness.

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