waydev

ENGINEERING LEADER’S INTELLIGENCE REPORT

Leading Engineering In The Age of AI

Table of Contents

Foreword

The data is no longer ambiguous. The transformation is no longer theoretical.

We are living through the most significant shift in software engineering in a generation. The question is no longer whether AI will reshape how software is built, it already has.

What remains uncertain is whether you are measuring it, governing it, and leading through it, or simply watching your dashboards report impressive-looking numbers while your business outcomes remain flat.

This report is built from Waydev's research, field experience with hundreds of engineering organizations, and the most rigorous analysis of AI-era engineering available today. It’s written for CTOs, VPs of Engineering, and Engineering Directors who need more than inspirational rhetoric. You need metrics, frameworks, and clarity.

> *The core paradox of AI-era engineering: activity is skyrocketing, velocity looks extraordinary, and yet for many teams, real business outcomes remain stubbornly unchanged. The problem is not your tools, it is what you are measuring.*

1. The State of AI Adoption

What is actually happening inside the world's leading engineering organizations.

The Numbers

Across the industry's most advanced engineering organizations, AI is no longer a pilot program. It is the operating reality.

95% of OpenAI engineers use Codex daily	200% productivity increase per engineer at Anthropic	4% of all public GitHub commits authored by Claude Code
70% more PRs by engineers who fully embrace AI	59% increase in average engineering throughput	55% faster task completion with GitHub Copilot

These are not projections. These are measurements from production engineering organizations today.

Boris Cherny, Head of Claude Code at Anthropic, reported that 100% of his code has been written by AI since November 2025, with zero manual edits, while shipping 10 to 30 pull requests per day. His assessment: coding, as a mechanical bottleneck, is largely solved. The question is now what to build, not whether it can be built.

OpenAI's Sherwin Wu described a team where engineers are no longer typing code. They are running 10 to 20 parallel AI threads, steering agents like fleet commanders. PR review time has collapsed from 10–15 minutes down to 2–3 minutes, because AI pre-reviews everything.

> *Claude Code now accounts for 4% of all public GitHub commits, and is predicted to reach 20% by the end of 2026. The trajectory is not gradual. It is exponential.*

The Structural Shift: From Authorship to Orchestration

The most productive engineers today are not the best typists, they are the best directors of AI capability. This shift has profound implications for how teams are structured, how individuals are evaluated, and what skills drive competitive advantage.

By the end of 2026, industry observers expect the title of Software Engineer to evolve. The emerging descriptor *Builder* reflects a broader, more generalist capability set: thinking across disciplines, identifying latent demand, and leveraging AI fluently.

Anthropic's internal operating philosophy crystallizes this. The organization deliberately under-resources projects, pairing a single exceptional engineer with massive problems and unlimited AI compute. The principle: more compute often beats higher headcount.

The McKinsey Context

Top-quartile engineering organizations grow revenue 4 to 5 times faster than their peers. AI adoption is accelerating this divergence; because the organizations that effectively measure, govern, and optimize AI usage compound advantages over time.

The organizations that fail to develop measurement infrastructure now will not simply fall behind in AI adoption metrics. They will fall behind in revenue growth, product velocity, and competitive position.

Why Most Teams Are Not Seeing These Numbers Yet

The headline numbers are real - but they describe the leading edge, not the average. Developer research consistently shows that real-world gains are meaningful but structurally constrained. Understanding those constraints is what separates teams that capture AI's upside from those that keep wondering why their velocity metrics look great but outcomes do not move.

The first constraint is structural: coding represents only around 16% of how engineers actually spend their time. Even a dramatic acceleration in code generation can only move overall throughput so much. The rest of the time - planning, alignment, scoping, stakeholder communication, code review - AI has not yet materially touched. A four-day task becoming a three-day task is a real gain. It is not a 3x increase in shipped value.

The second constraint is organizational. Isolated solo adoption does not produce meaningful gains. AI productivity compounds when teams develop shared practices, shared prompting approaches, and collective muscle memory. That organizational work is slower and harder than purchasing a license - and most organizations have not done it.

The third constraint is contextual. AI tools perform well on problems that are self-contained and well-documented. Most real engineering work is neither. The knowledge that matters - why a system was designed a certain way, what the implicit contracts between services are, what business constraints apply - lives in people's heads, not in any document AI can access. Until that institutional knowledge is made explicit, AI will keep hitting a ceiling on the hardest and most valuable problems.

> *The gap between the headline numbers and your team's experience is not a failure of your tools or your engineers. It is a measurement and organizational problem. Closing it requires focusing on the parts of the job AI has not touched yet.*

2. What Traditional Metrics Can't See

Traditional metrics still work, but they weren't built for what AI makes visible, or invisible.

The Blind Spot

Traditional engineering metrics (lines of code, commits, story points, PR velocity, cycle time) were designed to measure human activity. They remain valid signals of human work. But when AI begins producing the majority of that activity, these numbers inflate dramatically, and a new set of outcomes falls completely outside their view.

The result is what Waydev calls the engineering leader's paradox: the metrics that historically indicated a healthy engineering organization now routinely show excellent performance in teams where actual business outcomes are flat or declining. You are measuring activity. You are interpreting it as impact. The gap between those two things is where strategy goes to die.

+15.2%	−6.8%	72 min
Throughput on feature branches	Throughput on main branch	Mean time to recovery, up 13% YoY

CircleCI's 2026 State of Software Delivery report, drawn from over 28 million CI workflows, surfaces the clearest evidence. AI-assisted development drove a 59% increase in average engineering throughput, but the median team saw main branch throughput decline 6.8%. More code is entering the pipeline. Less is making it to customers.

> *For a team pushing 500 changes per day, a mean time to recovery of 72 minutes costs the equivalent of 12 full-time engineers annually. AI generated the productivity. An unretooled measurement system let it drain away.*

What Traditional Metrics Miss Specifically

- **Lines of code and commits:** When AI generates thousands of lines in seconds, raw volume metrics measure AI output, not engineer effectiveness. A team with lower AI adoption appears less productive.
- **Story points and PR velocity:** Teams that fully embrace AI open 70% more pull requests. If those PRs are not driving outcomes, higher velocity is noise, not signal.
- **Cycle time:** AI compresses implementation time while review and governance remain human-paced. Implementation speed can mask growing downstream bottlenecks.
- **Individual contribution metrics:** Engineers orchestrating AI agents can produce the output of many. Measuring individual lines contributed becomes meaningless.
- **Adoption metrics alone:** License utilization and rollout percentages measure inputs, not outcomes. Organizations report 60–70% AI-assisted commits while simultaneously experiencing flat delivery and rising rework.

What the Top 5% Do Differently

The top 5% of engineering organizations in CircleCI's research nearly doubled throughput year-over-year and increased main branch throughput by 26%. They were not operating with better AI tools. They had better visibility: tracking the metrics that determine whether AI is actually delivering results at production.

METRIC	TOP 5% BENCHMARK	INDUSTRY AVERAGE
Main branch success rate	90%+	70.8%
Mean time to recovery	< 60 minutes	72 min (+13% YoY)
Main branch throughput	+26% YoY	−6.8% YoY
AI tool active usage	80%+	~60%

3. The New Metrics Architecture

A three-tier framework for AI-era engineering intelligence.

Tier 1: Outcome Over Activity

The primary shift required is from activity to outcome. These metrics replace the inflated signals:

- **Time from decision to deployment:** Not commit to merge. The full cycle from strategic decision to shipped customer value.
- **Iteration velocity on strategic bets:** How quickly can the organization test, validate, and iterate on its highest-priority initiatives?
- **Decision quality:** What percentage of shipped features actually move the business needle?
- **Main branch success rate:** Benchmark: 90%. Industry average: 70.8%.
- **Mean time to recovery:** For AI-generated code specifically. Target: under 60 minutes.

Tier 2: System Efficiency

Where AI creates value, and where it creates new friction:

- **Intent-to-value velocity:** How fast does the organization turn a product decision into shipped, validated customer value?
- **AI-generated PR friction:** Are over-review processes slowing down code that AI has already pre-reviewed?
- **Branch vs. main branch throughput delta:** A large gap indicates governance or review bottlenecks.
- **Human intervention frequency:** How often are engineers overriding or correcting AI work? High rates signal prompting quality or tool mismatch.

Tier 3: Leading Indicators

The inputs that predict future performance:

- **AI adoption impact:** Tool usage correlated specifically to delivery wins, not just license counts.
- **AI-augmented performance:** Which engineers are 3x-ing output through orchestration? What characterizes their approach?
- **AI contribution by tool:** Which tools' output actually survives the full delivery cycle to production?
- **AI rework rate:** What percentage of AI-generated code requires significant human revision before merge?
- **Cycle time delta on AI PRs:** Are AI-generated pull requests moving faster or slower through review than human-authored ones?

> *The organizations measuring AI impact now will compound advantages over the next 2 to 3 years. The ones flying blind will wonder why improved velocity isn't translating to business outcomes.*

The Three Dimensions of AI Measurement

DIMENSION	KEY METRICS	PURPOSE
Utilization	DAUs/WAUs, % AI-assisted PRs, % AI-generated code, tasks assigned to agents	Establish baseline adoption
Impact	AI-driven time savings, PR throughput, change fail %, code maintainability, developer satisfaction	Measure real productivity and quality effects
Cost	AI spend per developer, net time gain, agent hourly rate (HEH/AI spend)	Optimize ROI and justify investment

4. The Agentic Shift

Governing, measuring, and leading when autonomous agents write code.

The Platform as Organizational Operating System

When AI agents enter the software development lifecycle, the platform changes its fundamental nature. It stops being the road and becomes the rulebook; the place where the organization encodes how work is allowed to happen: security policies, compliance boundaries, cost controls, quality gates, deployment rules, governance.

This is not a nuance. It is a strategic reorientation that most organizations have not yet made.

> *Agents without platforms create chaos. Agents without governance create risk. Agents without measurement create blind spots.*

Four Classes of Platform Users

- **Engineers writing and reviewing code:** Now increasingly acting as orchestrators rather than individual contributors.
- **Business users building workflows:** Product managers, designers, and analysts using AI to generate working software. The definition of developer is expanding.
- **Data teams running AI pipelines:** Requiring platform capabilities traditional developer platforms were not designed to provide.
- **Autonomous agents:** The newest class. Agents require the same things developers do (identities, permissions, audit trails, guardrails, observability) but at machine speed.

What Agent Observability Requires

For every agent-generated pull request, the organization must be able to answer: which model generated it, which policy rules were evaluated, which prompts triggered the action, and what reasoning path produced the outcome. Without that visibility, autonomous systems cannot be trusted. Trust is the limiting factor of AI adoption at scale.

AGENT METRIC	WHAT IT MEASURES
Decision volume	SDLC decisions made by agents vs. humans
Policy adherence rate	% of agent actions compliant on first pass
Human intervention frequency	How often engineers override or correct AI work
Behavioral anomaly rate	Agent actions outside expected parameters
AI contribution by tool	Which tools produce code that reaches production
Model deployment frequency	How often prompts and configs change system behavior

Governance at Machine Speed

Traditional governance operates on human time: weekly reviews, quarterly audits, sprint retrospectives. AI agents operate at machine speed, potentially thousands of actions per hour. An agent attempting to access sensitive customer data needs to be evaluated against privacy policies instantly. A workflow exceeding a cost threshold needs to be throttled before the invoice arrives.

This means the tacit knowledge that used to live inside people - the staff engineer who knows the deployment rules, the security lead who knows which services can access customer data, must be formalized, written down, and enforced automatically through the platform. The platform becomes organizational memory.

5. The ROI Question

Proving AI's value to boards. With data, not anecdotes.

The Gap Every Engineering Leader Must Close

Every department in a modern company has analytics. Sales has Salesforce. Marketing has HubSpot. Finance has forecasting tools. But engineering, the team building the product, has historically had almost no visibility beyond Jira tickets.

You can see what tasks exist. You cannot see how software is built, how long it takes to go from the first line of code to production, where teams get blocked, or which AI tools are actually making a difference.

> *Most executives still cannot answer a simple question: which AI tool is actually improving business outcomes? Adoption is not impact. License counts are not ROI.*

The ROI Calculation Framework

- **Direct time savings:** AI-driven time savings per developer per week. Intercom, by nearly doubling AI adoption, achieved a 41% increase in developer time savings.
- **Net time gain per developer:** Time savings minus AI spend. The true productivity delta accounting for tool cost.
- **Delivery cycle compression:** Booking.com deployed AI tools to over 3,500 engineers and achieved a 16% throughput increase within months.
- **Before/after comparative analysis:** Because Waydev captures longitudinal engineering data, organizations can precisely compare performance pre- and post-AI adoption across 150+ metrics.
- **Agent hourly rate:** Human-equivalent hours completed by agents, divided by AI spend, enabling direct cost comparison with human engineering capacity.

What to Show Your Board

BOARD QUESTION	REQUIRED METRIC	WAYDEV CAPABILITY
Which tools produce production code?	AI contribution by tool to main branch	Commit attribution analysis
Is AI improving or inflating?	Delivery cycle before/after AI adoption	Longitudinal comparison, 150+ metrics
What is AI actually costing?	Net time gain per developer	Time savings minus AI spend, per engineer
Which engineers leverage AI best?	Output delta by AI adoption level	AI-augmented performance benchmarking
Where are the new bottlenecks?	Branch vs. main branch throughput delta	Real-time pipeline visibility

6. Implementation Guide

Rolling out AI metrics without breaking culture.

The Human Factor

Measuring developer activity in the context of AI is a sensitive undertaking. The risk of malicious compliance is real. Metrics like code generation volume are particularly susceptible to gaming. Engineers who optimize for the metric rather than the outcome will produce data that is useless at best and misleading at worst.

There is also a deeper human dynamic at play. Engineering teams are often polarized between those who have embraced AI and those who remain skeptical. In many organizations it is genuinely unclear whether it is high or low status to talk openly about using AI tooling. Without explicit leadership on norms, ambiguity fills the void - and ambiguity slows adoption, skill development, and ultimately outcomes.

Leaders should not underestimate the skill gap either. Using AI effectively is its own discipline. Developers early in that learning curve get substantially less out of the tools than those with significant practice. The ability to decompose a complex problem into a clear, well-scoped prompt takes time to develop. Measurement programs that do not account for this will misread low-output engineers as unproductive rather than as early in a learning curve.

> *Strongly caution against top-down mandates or using AI metrics for individual performance evaluation. The goal is to understand how AI-assisted work affects developer experience and software quality, not to micromanage output.*

Three-Phase Rollout

Phase 1: Establish Baselines. Weeks 1 to 4

- Connect to your engineering stack: GitHub/GitLab, Jira, CI/CD systems
- Establish pre-AI baselines: PR throughput, lead time, change failure rate, time to restore service
- Document AI tool landscape: which tools are in use, by whom, at what frequency
- Communicate measurement purpose clearly: learning and investment optimization, not performance surveillance

Phase 2: Measure Adoption and Initial Impact. Weeks 5 to 12

- Track utilization: DAUs/WAUs per tool, % AI-assisted PRs, % AI-generated commits
- Begin direct impact measurement: AI-driven time savings per developer per week
- Compare branch-level vs. main-branch throughput to identify bottlenecks
- Identify early adopters and high-leverage use cases for organizational learning

Phase 3: Optimize and Govern. Month 3 Onwards

- Shift focus to outcome metrics: decision-to-deployment time, main branch success rate
- Build agent governance: policy adherence rates, intervention frequency, anomaly tracking
- Implement cost optimization: net time gain per developer, AI spend per engineer
- Begin board-level reporting: connect AI metrics to revenue acceleration and risk reduction

Communication Principles

- **Reinforce non-punitive intent:** AI metrics will not be used in individual performance evaluations. Point to the existing performance review process as evidence.
- **Frame as resource allocation:** Data guides which tools and workflows deliver real value. Not surveillance.
- **Share results bidirectionally:** When measurement reveals a team struggling with AI workflows, that is a signal to provide support - not assign blame.
- **Build from DORA:** AI context adds a layer to existing frameworks. It does not replace them.

7. The 18-Month Roadmap

Where engineering leadership goes from here.

HORIZON	TIME FRAME	PRIMARY FOCUS	KEY DELIVERABLE
Immediate	0–3 months	Establish baselines, drive adoption	Before/after comparison infrastructure
Near-term	3–9 months	Measure impact, optimize workflows	AI ROI reporting for executive leadership
Strategic	9–18 months	Agent governance, platform as control plane	Decision transparency and real-time governance

The Decisions That Cannot Wait

- **Invisible ROI:** Without measurement infrastructure, you cannot demonstrate to your board what your AI investment is returning. You are making significant budget decisions on intuition rather than data.
- **Invisible risk:** AI-generated code entering production without governance tracking creates compliance exposure and security risk that may not surface until it is expensive to address.
- **Invisible inefficiency:** Without visibility into where AI workflows create friction, you cannot optimize your AI investment.

The Compounding Effect

The organizations building AI measurement infrastructure today are not just solving a current problem. They are building a capability that compounds. Every month of longitudinal data makes before/after comparisons more precise. Every quarter of agent behavior tracking makes governance frameworks more effective.

The gap between organizations with measurement infrastructure and those without it is widening at the same rate that AI's role in engineering is expanding. That rate is accelerating.

> *In the next 2 to 3 years, most engineering metrics dashboards will need a full reinvention. The organizations that move now will be setting the benchmarks everyone else measures against.*

8. About Waydev

Waydev is an AI-native engineering intelligence platform that analyzes real development activity across your stack to show how the SDLC actually operates. It helps organizations measure AI adoption, quantify impact, and understand the ROI of AI tools based on production outcomes. Waydev acts as the system of record for the modern SDLC, providing clarity across both human and AI-driven work.

CAPABILITY	WHAT IT DOES
AI contribution tracking	What % of merged code is model-generated, by tool and by team
Before/after analysis	Compares 150+ metrics pre- and post-AI adoption with statistical rigor
Agent detection	Identifies which AI agents operate across repos and classifies contributions
Vendor performance comparison	Which tool actually reduces incidents and improves deploy frequency
ROI per AI-active team	Return on AI investment at team, and org level
Predictive simulation	Models what changes would reduce cycle time with concrete recommendations

Visit waydev.co to learn more

waydev

TRUSTED BY **FORTUNE 500** ENTERPRISE COMPANIES

citi VENTURES | WOM | TATA HEALTH | NOWCOM | SOVOS | Yellowbrick | pwc

innovaccer | AEROMEXICO | BlueCross BlueShield | Carrier | AON | Capgemini | genpact