AI tools don't create ROI.

The way you build software does

nearform.com

Nearform

Enterprises are investing heavily in AI tools for software development. But without the right practices, architecture, and oversight in place, productivity gains remain inconsistent and hard to measure.

To generate measurable returns, AI must be integrated into the software delivery lifecycle, not treated as a standalone tool.

AI tools are everywhere. Impact is harder to find.

What we're seeing

AI is already part of the software development toolkit. Across enterprises, AI generates code, agents assist with testing, and automation tools accelerate routine tasks across the development lifecycle.

Yet many organisations still struggle to show that these investments are improving delivery performance.

The issue is not adoption; it's that most initiatives introduce new tools without changing how software is built, tested, and operated. Workflows, architecture readiness, production integration, and governance will all typically remain unchanged.

Developers may use AI every day, but *without system-level modifications, they will encounter great difficulty* when trying to measure improvements in delivery speed and productivity.

The result

The outcome is activity without meaningful impact. Many organisations remain stuck between experimentation and real adoption.

Those making progress take a different approach. They treat AI as part of the delivery system, measure what matters, and keep focus on outcomes that drive the business.

This shift isn't incremental. It changes how engineers and AI deliver work together, through redesigned workflows and more coordinated execution.

It requires upfront investment in defining workflows, configuring systems, and establishing supporting frameworks. While this introduces short-term overhead, it's where the most consistent improvements in delivery performance are realised.

How to prove AI ROI in engineering

Measure first_

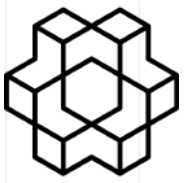
Introducing AI tools without establishing a baseline is a common mistake. Without one, it becomes impossible to determine whether AI is improving delivery performance. Teams may feel more productive, but subjective impressions are not enough to justify investment.

Before introducing AI into delivery workflows, capture the metrics that show how the system currently performs. Typical signals, aligned with DORA metrics and delivery performance indicators, include:

- Lead time for changes, from commit to production
- Deployment frequency
- Pull request cycle time and review latency
- Change failure rate and mean time to recovery, MTTR
- Production incident rate and defect escape rate

These indicators reveal where time is currently being lost. Once AI is introduced, they provide a clear way to measure whether delivery is improving.

If those metrics improve, AI is contributing to measurable engineering impact. If they do not, it often indicates that tools alone are not addressing underlying delivery constraints.

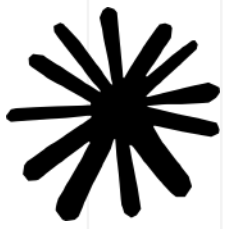


Keep experienced engineers in the loop

AI can accelerate many parts of development, from generating scaffolding and documentation to proposing fixes and producing tests. But speed does not remove the need for engineering judgement.

Reducing human oversight of AI-generated code introduces significant technical risk.

AI-generated code may be syntactically correct while still containing performance issues, architectural inconsistencies, or security weaknesses.



Strong discipline must remain in place.

Best practice includes:

- Routing AI-generated code through normal pull request reviews and CI pipelines
- Maintaining static analysis and security scanning processes
- Applying existing coding standards and architectural guidelines
- Ensuring senior engineers remain responsible for key design decisions

A critical factor in making this effective is ensuring AI tools have access to the right context at the right point in the workflow.

This includes structured specifications, relevant parts of the codebase, and well-defined prompts, often integrated through existing tooling. Without this, outputs can be inconsistent or misaligned with how the system is designed to operate.

The most effective teams treat AI as an amplifier for experienced engineers rather than a replacement. AI may accelerate execution, but human expertise keeps systems resilient and maintainable.

Focus on outcomes not output

Another common mistake is **measuring the wrong signals.**
Many organisations attempt to measure AI adoption using activity metrics such as:

- Lines of code generated
- Number of prompts executed
- Time spent using AI tools

These metrics show how frequently tools are used. They do not show whether delivery performance is improving. Measurable AI returns appear in system outcomes.

Organisations should instead look for signals such as:

- Reduced lead time for changes
- Increased deployment frequency
- Lower change failure rate
- Faster recovery from production incidents
- Improved system reliability in production

When AI is integrated properly into delivery workflows, these outcomes begin to move. Delivery accelerates, maintenance burdens shrink, and teams can spend more time on product innovation.

Treat security as a design requirement

Security can easily become a blocker if it is treated as an afterthought. Many AI pilots succeed technically but fail to reach production once security and risk concerns surface late in the process.

AI systems interact with data, services, and decision logic in ways that traditional architectures were not designed to handle. When governance, access controls, and auditability are not built into the engineering process from the start, organisations are often forced to pause deployments while those risks are addressed.

Security itself does not slow AI adoption, but retrofitting it afterwards can.

Why AI ROI often fails to appear

Despite strong enterprise interest in AI, several patterns repeatedly prevent organisations from realising measurable value.

Tool-first adoption
Many organisations begin by purchasing AI tools and encouraging experimentation. Without changes to workflows, governance, or measurement practices, these tools rarely translate into measurable improvements.

Pilot fatigue
AI initiatives often become stuck in small experiments. Teams test new capabilities in isolation but struggle to translate them into improvements across production systems.

Automating the wrong work
AI is often applied to high-profile innovation initiatives because they attract attention. In practice, the fastest returns usually come from automating low-profile repetitive work that consumes developer time.

Tasks such as testing, refactoring, documentation, and dependency upgrades frequently deliver the earliest productivity gains.

AI-native engineering drives real ROI

Without changing how software is delivered, it is difficult to measure the results of introducing AI tools into the development lifecycle. Productivity may increase in isolated tasks, but this provides little value if system outcomes remain unchanged.

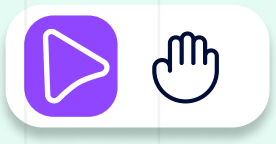
Organisations demonstrating real AI returns take a different approach. They **treat AI as part of the delivery system itself.**

That means:

- Measuring delivery performance before introducing automation
- Maintaining strong human oversight of AI-generated work
- Focusing on outcomes such as faster releases, improved reliability, and reduced technical debt

This is the principle behind AI-native engineering.
In practice, this builds on established DevOps and platform engineering approaches, extending them to integrate AI into developer workflows, delivery pipelines, and governance from the start.

The result is measurable improvements in how software is built, delivered, and operated.



Unaccounted cost and risk_

AI adoption introduces additional cost and risk that are often underestimated. These include tool licensing, infrastructure and inference costs, as well as risks related to security, data exposure, and code provenance. Without accounting for these factors, organisations may overestimate the return on AI investments.

Where AI is helping and where it is not

If AI is increasing output but engineering performance is unchanged, the issue may not be the tool.

Our **60-minute AI-native engineering** Inspire session shares what we're seeing across enterprise environments — where AI is improving delivery outcomes and where it is introducing friction.

Book a session to get a clearer perspective on how delivery systems are evolving and where AI can drive meaningful impact in your organisation.

Book your session

nearform.com

Nearform