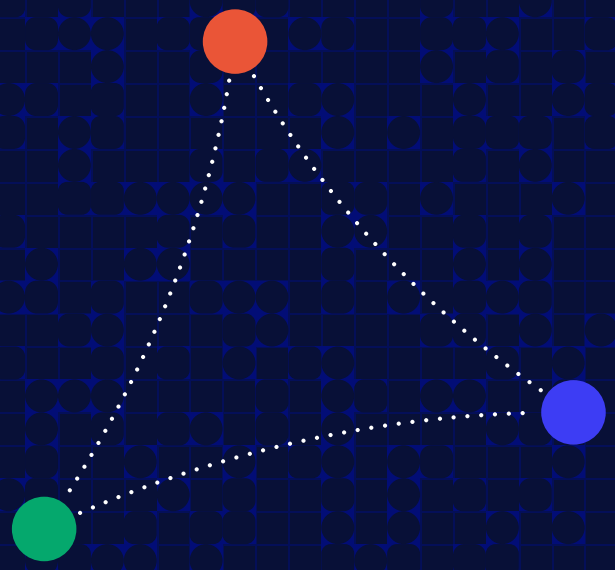
astorg.

WHITE PAPER — SEPTEMBER 2026

Rebuilding the Product & Engineering Operating Model for the AI-first era

WHAT NEEDS TO CHANGE AND WHERE TO START

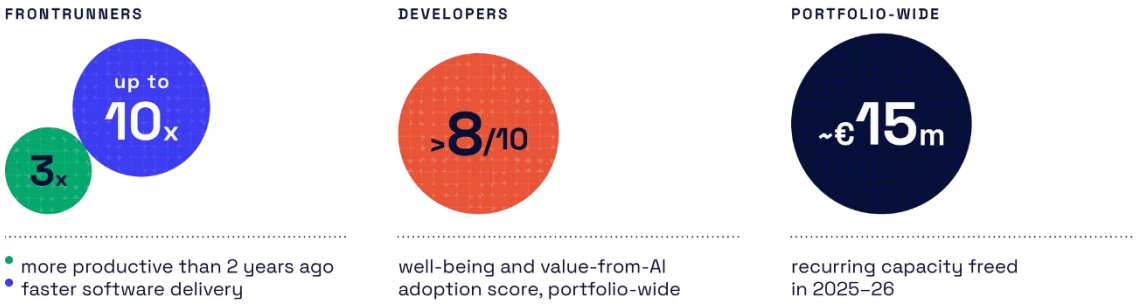


Executive Summary

For the past two years, we have focused our efforts on the one thing an owner can genuinely impact: building the muscle a company needs to create an AI advantage that compounds. It is the skills, capabilities and leadership to keep evolving the operating model. We started in our tech-driven companies because the lesson transfers, their foundations were further ahead, and their delivery loop turns fastest. We are now scaling what we learned across the entire portfolio.

The frontrunner companies in our portfolio now build software up to 10x faster than they did two years ago. Fenergo and Opus 2 ship in weeks what used to take quarters, and are now 3x+ more productive than they were before.[1] Xceptor rolls out in less than three days what once took more than a month. Across the portfolio, these gains have already converted into ~€7 million equivalent of recurring capacity freed in 2025 and a further €8 million being unlocked in 2026, based on productivity gains. This is how they work now, though challenges remain.

Advancements in tooling have made this shift possible, but it took much more than that. The tooling is the easy part. Getting people to use it productively is harder. The gains at that point start to show as individuals get 1.5-2.5x faster. But beyond the team level, these gains flatten; cumbersome handoffs across teams, diluted customer signal feeding the backlog and mismatched talent turn the operating model into the impediment. Teams pull ahead by rebuilding their workflows around AI. Organizations pull ahead by evolving the operating model that lets those workflows run. It is a far harder task, but it is the main source of competitive advantage in AI-first product development.



Not everything however needs to change, and we hope this paper contributes to making that distinction. It sets out the recipe of what needs to change in the operating model within the next year or so to capture the returns from AI-first Product Development Lifecycle. It is based on our takeaways from real production work, in our frontrunner companies. It is rooted in the decisions they have taken or are facing, and what we are implementing at scale across our portfolio. At every step, we call out the no-regret moves we feel most confident about. We will be wrong about many of our assumptions, given the pace of change in this space. Nevertheless, we believe anticipating and preparing for the bottlenecks ahead is what will separate the winners from the losers.

We focused on the four domains where the change is already underway: **Product & Engineering, People & Talent, Governance & Compliance, and ROI**. Before diving in, we want to acknowledge that what decided which companies made the leap was not high budgets, state-of-the-art tooling, or even eager tech talent. It was committed, disciplined and inspiring senior leaders, who role-modeled the change, sat at the keyboard beside their teams to share the learning journey and the risk. We are lucky to have had so many, who were also generous enough to contribute their experience to this paper.

[1] Cycle time falls faster than productivity rises, as delivery parallelizes and handoffs collapse.

Table of Contents

This paper was written by the Astorg Digital, Tech & Data Operations team in close collaboration with Product, Engineering and Transformation leaders across our portfolio companies. We are deeply grateful for their continuous involvement over the past two years. Their expertise and commitment have been instrumental in designing, planning and executing our shared acceleration of AI across Product and Engineering.

INTRODUCTION

Setting the direction

In mid-2024, one of the clearest places for AI to create value across our portfolio was in the way our software, tech, and data-driven companies design, develop, and operate their products. The new tooling was spreading fast but had not yet reached companies of the scale we invest in. We chose to lead rather than let each company find its own way there. Everything since has run on the same logic of finding what is keeping our companies from scaling, clearing it, and moving to the next bottleneck.

Two years in, the results are material. Individuals using agentic AI inside their existing workflow deliver 1.5-2.5x faster than before. The frontrunners in The Forge, our acceleration program, have rebuilt the work as an AI-first loop and cut delivery cycle times by up to 10x.[2] Fenergo's pods now ship features once scoped at six to nine months in a matter of weeks, at roughly 3x the productivity; at Xceptor, features that took more than a month to build before AI now roll out in under three days, repeatedly. At Opus 2, the most architecturally complex AI-first build in the company's roadmap was estimated at ten months and delivered in three. Sprint by sprint, these teams have taken on bigger scopes, from single features to value streams to re-platforming their legacy solutions.

From curiosity to daily use

In 2024, we set the ground for individual adoption, through introduction of the tooling to our portfolio, training of our tech talents and implementation of a common telemetry.

First, by putting the tools in everyone's hands. Through our alliance with Microsoft we secured a partnership with GitHub, the obvious enterprise-grade choice at that time. We centralized the rollout and negotiated exclusive terms leveraging the size of our portfolio.

Second, by making sure the tools were used well, and this was never a checklist exercise for us. Rather than generic blanket sessions, we trained more than 400 developers in cross-portfolio cohorts tailored by role and coding language. Once that common layer was in place, we met people where they were. We ran office hours where developers brought whatever was blocking them on real backlog items and worked through it live in pair-programming sessions. That hands-on support built genuine conviction in the early days, when skepticism was still the norm.

Third, we measured what was happening. We built an adoption dashboard for the whole portfolio, giving us and each technology leader a clear view of who was adopting, how fast, and against what benchmark. Qualitative KPIs such as DevEx were deployed to secure change management. Live,

[2] Cycle time falls faster than productivity rises, as delivery parallelizes and handoffs collapse.

centralized line of sight on a portfolio's engineering activity was rare for a PE fund at the time. And we quickly found the first two levers were working: 62% adoption, 12 points above the then market average, and the number we cared about most, a developer survey score above 8/10 on well-being improvement and value impact from adoption of AI tooling.

Along the way we built our most valuable asset, a real community of practice. We set up cross-portfolio cohorts of product and engineering people, held office hours and online Meet-Ups. This quickly created a strong cross-portfolio momentum, a sense of belonging to a network of practitioners facing the same challenges, and above all, a safe space to share learnings, failures and wins. This community multiplied how fast the whole portfolio learned and helped us shape the support our companies needed.

From point solutions to a new operating model

In late 2025, that community came together and raised the ambition. Astorg's DevDay gathered tech leaders and engineers from across the portfolio to take stock of what we'd done and what would hold us back next. Individual adoption within existing workflows was no longer the constraint. So we set a more ambitious direction on two fronts: carrying AI into every stage of the lifecycle, not just coding but product, design, QA, DevOps and SRE; and pushing past autocomplete and one-off chatbot help towards real autonomy, with agents running work in loops that span the product development lifecycle (PDLC) end to end. Our measurement had to evolve too, tracking these new axes of adoption and extending into business outcomes and like-for-like productivity on real tasks. We also recognized that our companies started from different points.

The first track, fixing the basics, aimed to get the whole portfolio AI-ready. Once agents operate on their own, gaps in the underlying tech, legacy DevOps tooling or thin automated testing, cause far more friction than before. Solid, modern foundations are also the safest bet when the frontier shifts every few months. So we worked through the portfolio company by company, helping each find and fix what was slowing it down. Most started strong and moved quickly; those carrying more legacy or built through rapid acquisition had a harder road.

The second track emerged naturally. Our frontrunner companies were ready tech-wise and had strong momentum in moving from individuals using agents to full agentic loops, so they were also the first to feel the need to evolve their operating model. To help them anticipate the change, we built The Forge, an acceleration program to front-load value creation and derisk the transformation through iterative, fact-based and collaborative design of the target operating model.

Indeed, enthusiasm and good tooling take you a long way, but the gains stall unless someone has the explicit mandate to work differently. The Forge puts a small pod of skilled, motivated product people and engineers, backed by outside expertise where the company lacks it, onto a real production feature with real deadlines and CEO-level sponsorship. Teams ship production software while getting the room to step back and experiment in ways day-to-day delivery never allows. Alongside sits a scaling team focused on designing the target operating model, drawing in people

from every function to work out how the new ways of working land across the company. Rooted in real-life evidence and experimentation, the target operating model becomes the obvious path.

The leaders who got this right role-modeled the transformation. They sat at the keyboard alongside their developers and took the risk themselves, doing what others had said couldn't be done, or pairing with the engineer who swore a task would take three days to see whether it really had to. In doing so, they made it safe to try, and to be wrong on the way to getting it right. A 2026 BCG survey found that in the companies getting most value from AI, 88% of managers actively model its use themselves, against 25% elsewhere.[3] People embrace a change this big not when they are told to, but when someone they respect shows them it is possible.

"Engineers calibrate their leaders constantly. When they hear tech leadership discussing context management, hooks and orchestration in technical terms, they conclude that the people directing the change understand its trade-offs. When they hear only the vocabulary of adoption and productivity, they conclude, usually correctly, that they are being asked to follow a trend.

Hassan Lachgar, VP of Engineering, EcoVadis

Hassan leads the global engineering organization and the development and evolution of its technology platform. He has driven the scaling of its engineering capabilities and, more recently, the adoption of agentic AI across product and technology.

We have been lucky to have many such leaders across the portfolio. They made this transformation happen. **This paper sets out the Target Operating Model we have been shaping with them.** We wrote it for the product, engineering and people leaders in our own portfolio, as a pragmatic take on the questions we found most critical and most blocking as we scaled across our frontrunners. It is not an exhaustive list of everything a company must fix. We anchored each domain on the no-regret moves a company can start now, backed by evidence from the market and our own portfolio, while

[3] BCG, "AI Transformation Is a Workforce Transformation", 4 February 2026, bcg.com/publications/2026/ai-transformation-is-a-workforce-transformation.

flagging the obstacles we see coming. The paper is organized around four domains: Product & Engineering, People & Talent, Governance & Compliance, and ROI.

It is too soon for definitive answers in a field moving this fast, and we will adapt as we go. We share it anyway, because the obstacles ahead are cheaper to prepare for than to discover late. We hope it proves useful to leaders who want to embrace AI in the way they design, develop and maintain their products.

01

Product & Engineering

When delivery becomes cheaper and faster, where does the constraint move?

Agentic AI loops now draft requirements, code, tests and documentation faster than the surrounding product and engineering operating model can absorb. As velocity in these tasks rises, new bottlenecks emerge in areas that were previously hidden, or where the trade-offs used to be less stark. Their nature and relative weight vary across our portfolio companies, depending on each one's product context, and some will surface only once the AI-first delivery cycle is fully scaled. Even so, we can already point to three constraints where the pressure is mounting, and to the operating-model changes our frontrunners are making to address them: choosing what to build; giving agents a backbone to work through, spanning both the engineering platform underneath and the AI-specific layer of context, instructions and models on top; and verifying the output fast enough and reliably enough. These three decide how much of the speed-up translates into business outcomes rather than bloat and rework.

Outcomes matter more as delivery accelerates

Seemingly low cost of delivery means companies should double down on customer-centricity. As delivery becomes cheaper and faster, the risk of dispersing focus away from what drives material outcomes for customers rises. This matters more as AI-first engineering moves to self-improving loops which can propose, implement and refine variations faster than the organization can decide whether the next increment is worth it. This failure mode is already visible at the frontier of adoption: in a June 2026 interview, OpenAI's Codex product lead described uncoordinated internal teams routinely building competing, ship-quality versions of the same feature, making curation the scarce resource.[4] The strain shows in delivery data too. In February 2026, as agentic coding traction mounted, CircleCI reported average workflow throughput up 59% while median main-branch throughput fell 7% and merge success hit a five-year low, across more than 28 million workflows.[5] Having more work in motion needs stronger upstream and downstream market-fit signals, coupled with a more reactive roadmap that links directly to value creation, for example through Objectives and Key Results (OKRs). Rapid prototyping predates AI; it is now cheap and fast

[4] Andrew Ambrosino (OpenAI, Codex), interviewed on Lenny's Podcast, "OpenAI Codex lead on the new shape of product work", 28 June 2026, lennysnewsletter.com/p/openai-codex-lead-on-the-new-shape.
[5] CircleCI, "2026 State of Software Delivery", 18 February 2026, analysis of 28M+ CI/CD workflows, circleci.com/blog/five-takeaways-2026-software-delivery-report.

enough to become standard practice. Fastmarkets built a fully functional prototype of its new AI-first workflow, including synthetic data, in weeks; showing it to clients surfaced unexpected monetization options that were built into full-scale production from the start.

Feeding a fast-moving backlog requires deeper, more reliable discovery signal. The near-term answer is to increase the granularity and frequency with which already available signal, such as customer tickets, call transcripts, product analytics or competitive intelligence, directly feeds the backlog. AI-driven automation is also an opportunity to broaden the sources of signal, e.g., Trustpilot ratings, community forums and social channels. Enhanced discovery then feeds the top of the AI-first PDLC funnel, as agents convert the insight into high-fidelity requirements. In our portfolio, Fenergo's Forge pods run a six-agent PM intake architecture, Opus 2 started by codifying its product strategy, operating context and OKRs into structured documentation its agents read directly, so agents work from shared company context while humans keep exercising taste and judgment at each transition from ideation and refinement into delivery. The step from analytics insight to automated backlog intake is still maturing.

Validation also has to move earlier and become more outcome-based. There are two distinct tests. The first is directional: showing credible customers the product direction early enough for their feedback to change the build. We see this across the portfolio. At Xceptor, customer validation is now pulled forward to proof-of-concept stage, after only a few days of building, rather than the weeks or months into development that used to be typical; Opus 2 and Fenergo are increasing engagement with customer advisory boards on their roadmaps. Opus 2 also leans increasingly on feature flags and preview distributions to power users, pulling validation of new and experimental features forward. Synthetic users and customer panels could raise the volume, frequency and granularity of insight further. The second test is hard-numbers validation: checking whether shipped work moved the key result it was meant to move, whether that is usage, revenue, retention or cost. Validation at sustained AI-first velocity is still unproven, which is exactly why the instrumentation for it has to be built now.

Roadmap cadence then needs to change around the tighter signal loop. A purely annual or semi-annual roadmap cannot absorb learning quickly enough once delivery units move in days or hours. Rigid sprint-level prioritization can also become too heavy if every smaller delivery loop demands a full prioritization ceremony. The ceremonies are collapsing first: at EcoVadis, a refinement that took four hours went to two, then one, then thirty minutes, and is now a file generated automatically, with teams drifting to a board-and-pick model. The likely model is two-speed: quarterly objectives or OKRs at the strategic level, with more continuous prioritization below. At the lower level, especially in increasingly autonomous, continuously agentic loops, prioritization needs to be encoded as a reward function. This is where the validation instrumentation above becomes crucial, because it lets agents recognize when the nth optimization iteration has hit diminishing returns on the outcome metrics they are set to optimize.

> At EcoVadis, backlog refinement used to take **4 hours**, went to **2**, then **1**, then **30 minutes**. Now it generates itself, and teams are drifting to a board-and-pick model.

Release-to-GTM alignment becomes a visible problem as release cadence accelerates. Faster release helps only if sales, customer success, implementation and support understand what changed, why it matters and how to explain it to clients. As release frequency rises, enablement cannot rely on manual catch-up. Opus 2, which over the last couple of quarters moved from six-month cycles to weekly releases, has automated the generation of all its release notes, training materials and customer collateral. The end state is controlled, agent-operable release management, operating at the same cadence as agentic delivery. Beyond generating artifacts faster, product and engineering should also be embedded deeper in early client conversations, to keep the product discussion current and capture feedback sooner.

In two quarters, Opus 2 moved from **six-month** release cycles to **weekly** ones.

"Weekly releases aren't an engineering achievement, they're a go-to-market commitment. Support, Marketing, Sales, Ops, Product, Engineering and Design built our new release process. It's a truly cross-functional achievement.

Tiama Hanson-Drury, Chief Product and Technology Officer, Opus 2

Tiama is Chief Product and Technology Officer at Opus 2, the London-based legal technology company serving the world's leading litigation teams. She leads product and engineering, where AI now sits at the center of how the platform is built and how it works.

Product and engineering decision rights need to sit closer together. The speed and streamlining of handoffs that AI drives mean the two areas now have to operate as one loop more than they did in the past. Split agendas and waterfall decision-making at executive level can quickly become the bottleneck in the cycle. In many software companies this will point towards a CPTO model with strong product and engineering leads underneath. Where the roles stay distinct, the two leaders need a broader capability overlap than today, a highly technical CPO and a product-minded CTO, and a set of decision gates they run jointly, for example on prioritization, against shared outcome targets.

• NO-REGRET MOVES

Identify existing sources of discovery signal available to the company and make them accessible to agents. Use rapid prototyping to pull customer validation forward at least on your highest-value product development streams.

Agents don't scale on a fragmented backbone

Weak or fragmented engineering stacks were a known problem well before agents arrived, but the trade-offs were often judged acceptable. Migrations are disruptive, and an aging system could be run down on a longer timeline. Agentic delivery changes that calculation, because it amplifies existing strengths and accelerates the problems of weak foundations. Parts of the infrastructure backbone that hold up today are also bound to become bottlenecks as volume rises. As stated by Uma Namasivayam of Dropbox in a June 2026 interview, "If your code throughput goes up by 3x tomorrow, would your SDLC be able to absorb it? We have seen some early signs it is not the case. It breaks everywhere."[6] Both point to the same conclusion. This work should start now.

The cost of a fragmented or outdated stack rises when agents enter the workflow. PDLC systems sprawl already carries a maintenance, onboarding and support burden; in an AI-first setup with autonomous agents, it becomes a scaling constraint. Agents depend on well-defined interfaces and repeatable rules, so fragmentation multiplies the surfaces that have to be explicitly defined and maintained, across versioning systems, CI/CD platforms, artifact managers and the rest. It also stops learning from compounding across teams, because a pattern that works in one repository or pipeline does not necessarily replicate cleanly to another. Legacy tooling adds a further constraint. Some of it lacks the modern primitives that let agents operate safely, such as pre-Git source control, bespoke CI or manual release steps. That was survivable while agents were little more than an advanced chatbot inside the old workflow. Agents working asynchronously need that machinery to act through, and modern engineering platforms increasingly ship with more of it built in, including workflow orchestration, permissioning, code scanning and review hooks.

Volume will put pressure even on currently well-scoped infrastructure. Pipelines, test environments and merge queues are typically sized for human-scale change volume and speed, and not always comfortably even then. Agentic delivery changes the equation, both by letting human engineers parallelize work and by adding autonomous loops that fan out to build and test many iterations of the same change. For some of our portfolio companies the constraint is already apparent. At EcoVadis, teams found themselves queuing longer for shared full-stack environments as upstream acceleration and proliferation of tests, now cheaper to write, drive demand. Across multiple companies in our portfolio, we have concluded that neither scaling infrastructure linearly, nor incrementally optimizing the current setup would be enough. A deeper rethink of both the

[6] DX, "From PR Throughput to Product Velocity: How Dropbox Is Rethinking Productivity in the Agentic Era", Engineering Enablement podcast, 29 June 2026, newsletter.getdx.com/p/from-pr-throughput-to-product-velocity.

testing frameworks and the stack is underway as a result. For the companies that are not facing constraints in the short term, the focus should be identifying the parts of current infrastructure would take more than a few weeks or months to scale significantly when demand spikes.

Upgrading the tech backbone is the highest-confidence no-regret move. Models, providers and agentic capabilities will keep changing every few months, and much of the AI-specific layer will need reworking. A cleaner workbench carries higher option value, because each future capability can use the same source-control, CI/CD, test and deployment surfaces. Across the portfolio we are therefore prioritizing stack modernization wherever the base is fragmented, close to end of life or too bespoke to govern. Our companies were already progressing here, and we made a deliberate choice to make sure these efforts were comprehensive and to accelerate them across the full portfolio. The frontrunners in particular are finalizing consolidation of their source control, CI/CD and artifact-management platforms. Crucially, the AI-driven PDLC can itself accelerate this backbone work. At EcoVadis, a remotely running autonomous agent dedicated to maintenance and mechanical work has shipped over 300 pull requests, and one squad completed its full .NET 10 migration while delivering the rest of its roadmap ahead of plan. AI-accelerated migration patterns enabled Xceptor to modernize its repositories and delivery pipelines estate in bulk.

At EcoVadis, a remotely running autonomous agent dedicated to maintenance and mechanical work has shipped **300+** pull requests.

"Untangling our existing pipelines would not have been possible at this scale without AI. The impact on the way we ship software is revolutionary.

Michael Kinloch, SVP Engineering, Xceptor

Mike oversees Xceptor's Engineering strategy, focused on building and operating the secure, scalable, high-quality software, SaaS, and cloud services that underpin Xceptor's Data Automation Platform and products, including Post-Trade Operations and Tax. Mike also pioneered Xceptor's AI-native product development lifecycle to deliver significant efficiency gains across the business.

• NO-REGRET MOVES

Prioritize source-control and delivery-platform consolidation and upgrade where the base is fragmented or close to end of life; baseline current infrastructure load; expand capacity where it is at or close to becoming the bottleneck.

The AI-specific layer stays company-owned and portable

The other half of the backbone is what the company tells its agents. This is the context about its own business and the instructions on how they should work. Owned and governed as code, that layer becomes durable IP, and the model providers underneath become a sourcing decision the company can revisit as the market moves.

Context and instructions should be treated as code assets and core IP. The context layer makes the knowledge on which agents depend easily discoverable: the company, its products and customers, the code architecture, and the rationale behind past implementation choices. The instruction layer, expressed for instance as skills, tells agents how to work, such as which design patterns and quality standards to apply. When context and instructions live in scattered documents or individual prompts, agents risk acting on stale or contradictory information, and burn tokens re-deriving the same business logic session after session; versioned and reviewed as code, they become a maintainable asset and part of the company's IP. At full agentic volume the review conversation itself moves to this layer; as the creator of Anthropic's Claude Code put it in a July 2026 note, "Did you read the code?" becomes "what context was the model missing and how do we solve it for next time?"[7]

Ownership works best as internal open source: a central baseline with local extensions. A fully central model becomes inaccurate, because local context differs significantly across the organization and changes fast. A fully local model creates prompt sprawl and contradictory standards. Even federated models risk instruction bloat, context rot and friction between global and local standards if they are not deliberately governed and maintained. The pattern the Forge companies are implementing is closer to internal open source: expert maintainers own a central baseline covering security, interfaces, quality gates and staleness review, teams extend or fork it for local context, and useful changes flow back through pull requests. Fenergo runs a code-controlled Agent Engineering Playbook with four named reviewers, spec.md libraries attached to every build item, and pre-tool and post-tool hooks. Xceptor has a similar setup; EcoVadis's is close in mechanics but differentiated by division. On the evidence of our portfolio, modern versioning tools are enough to start with, while the capabilities for more advanced needs, such as lineage and evaluation of the instructions themselves, are currently available only as specialized tooling but will likely mature fast and fold into existing model harnesses.

[7] Boris Cherny (creator of Anthropic's Claude Code), public note, July 2026, threads.com/@boris_cherny/post/Da4CkQnEea-.

The choice of a model provider should not be set in stone. State-of-the-art suppliers keep leapfrogging one another every few months on capability, pricing, model access, safety posture and geopolitical exposure. We have not mandated a single provider or harness across the portfolio, but after broad early experimentation our companies are converging organically on a small sanctioned set, at most one to three providers, with volume concentrated on one. Interest in running open-source models, including locally, has been limited so far but is growing, driven by both cost and dependency considerations. Historically the overhead of running these setups made them impractical at the scale of the companies we hold. That calculation could change sooner rather than later, as the guardrails, evaluation harnesses and deployment tooling around open-source models mature. Whatever the provider choice, what we focus on with our companies is maintaining interoperability, keeping the company-owned context and instruction layer portable, and avoiding vendor-specific standards wherever a neutral format is feasible.

- NO-REGRET MOVES

Create context and instruction repositories with named expert maintainers, PR-based contributions and staleness checks; define which standards are mandatory and which teams may extend or fork. For any third-party delivered solution, ensure these artifacts are parts of the deliverables.

Verification becomes machine-first and tiered

An operating model that is AI-first in everything except review cannot work. A human reviewer clears roughly 500 lines of code an hour; an agent produces as much in minutes[8]. Volume is only half the constraint. In an AI-first cycle, humans direct outcomes rather than writing the code themselves, and they progressively lose the line-level context of each implementation, so the depth of diff-level review expected today cannot be assumed at agentic volume. Coping with both the quantitative and qualitative change means embedding quality controls earlier, increasing the share of automated controls, and setting explicit thresholds and triggers for selective human review.

Agents make thorough specs affordable, turning the specification into the first verification gate. When test-driven development tried to embed quality requirements at the source, it hit limits of economic affordability and human patience in how thorough those requirements could be. Agents have made it affordable to enumerate a spec's edge cases and define matching test plans granularly and consistently. In the current optimization paradigm, with planning assigned to the most capable models and execution delegated to cheaper, more drift-prone ones, it is especially important that specs act as a contract the downstream pipelines can check against. Within The Forge we are progressing rapidly here. At Fenergo, for instance, agents decompose each feature into 20 to 40 sub-items with extensive acceptance criteria, capturing requirements that would not

[8] The New Stack, "QCon chat: Is agentic AI killing continuous integration?" (CircleCI's Michael Webster; Microsoft's Daniel Doubrovkine), 27 January 2026, thenewstack.io/qcon-chat-is-agentic-ai-killing-continuous-integration.

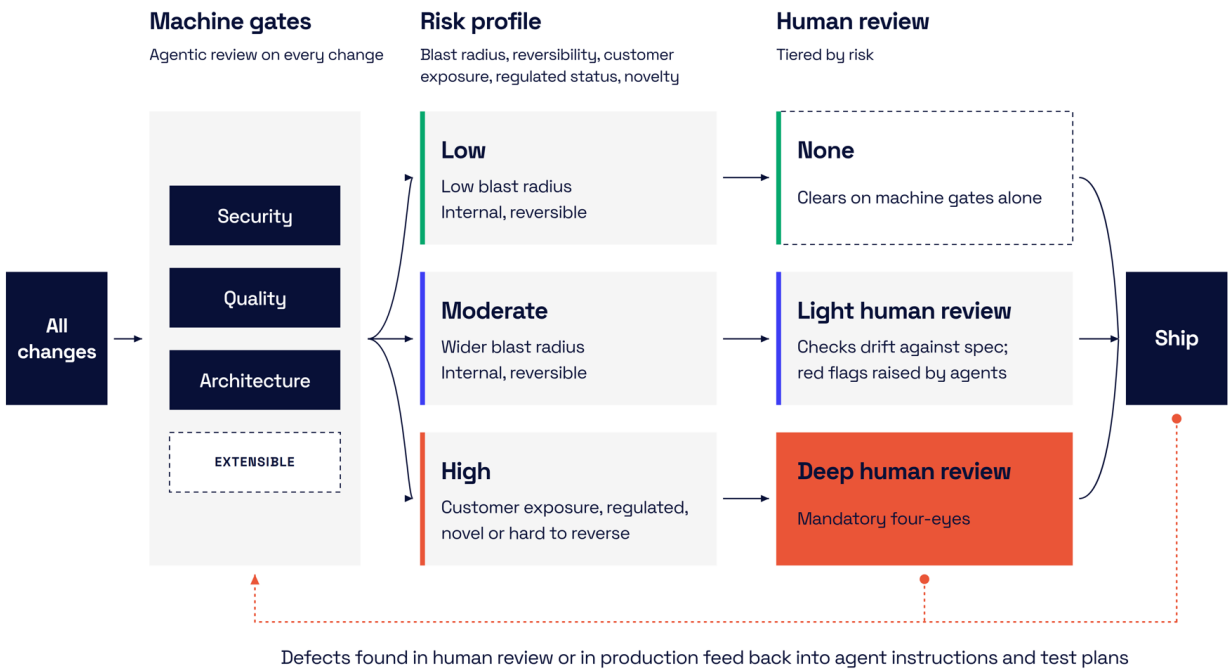
previously have made it into user stories and lifting the quality bar. The thoroughness of the spec is itself validated by an independent, automatically enforced agentic gate. Xceptor does the same, preventing agents from starting a build until the feature document, architecture document and testing strategy have been reviewed, a happy trade-off: "spending 30 minutes in dialogue with the agent during feature creation can save hours of rework during build."

> At EcoVadis, PRs on a 15-year-old legacy scope are now 90%+ AI-authored, with **zero rollbacks**.

Machine verification has to sit ahead of scarce human attention. Across our portfolio, companies are making AI code review mandatory on every pull request, as an organization-wide quality gate. The standing objection is agents marking their own homework: tests generated alongside the code inherit its assumptions and can be as flaky as any human-written suite. The answer is twofold, independence and feedback loops into instructions. Within The Forge we have already generalized independent review by specialized agents that sit in separate loops from the ones implementing the code. Fenergo puts dedicated code-review, test and security agents on every pull request, ahead of a minimum of two human reviews. Xceptor wraps each agent change in hooks that compile, test and lint before and after. EcoVadis has gone further, running delivery through a governed harness which can run in full, lean or quick mode, to adjust the machine-gating density vs. the risk of the change. The result is that for hundreds of merged PRs and stories shipped to production across a 15-year-old legacy scope, of which >90% were AI-authored, there have been zero production rollbacks. The next step is building feedback loops from defects caught in human review or in production, triaged to the layer that should have stopped them and back to the agent instructions that generate the test plans and checks. Currently, this still relies on human-led iteration across our portfolio, and automating it is a clear direction of travel.

The depth of human review then has to be tiered to stay effective. Blast radius, reversibility, customer exposure, regulated status and novelty of the change should dictate the tier. An internal, reversible change can clear on machine gates alone, or with light human review for drift against spec. A customer-facing change, or any regulated scope requiring mandatory four-eyes review, should still trigger thorough human review. Cloudflare already routes this way, auto-classifying each merge request as trivial, lite or full, with security-sensitive paths always forced to full review.[9] Across our portfolio, which is largely regulated or mission-critical, human review currently stays mandatory for most changes: "autonomy ends at the merge button" for the Forge agents. To keep that human gate effective, our Forge teams are already experimenting with automated review briefs that help decide quickly which parts of a pull request most deserve human attention. At Fenergo, the deepening use of agentic engineering has been matched by stronger automated quality, security and review controls. The objective is not simply to generate software faster, but to increase delivery capacity while preserving the controls expected by regulated financial institutions.

[9] The Cloudflare Blog, "Orchestrating AI code review at scale", 20 April 2026, blog.cloudflare.com/ai-code-review.



Tiered verification: machine gates apply on every change, human review depth is tiered by risk level

● NO-REGRET MOVES

Enforce multi-dimensional agentic review (security, quality, architecture) ahead of human review and track both false positives and false negatives to improve the filter over time; write down the risk criteria that set the level of human review for each change, keeping strict human merge ownership and four-eyes where it is regulatorily mandated.

02

People & Talent

What skills and team shapes does the AI-first PDLC require, and how can companies close the gap?

The people challenge starts when the AI-first Product & Engineering model begins to scale. The previous section described how agents entering the workflows require reinforcing the links from product strategy to execution and from execution to validation. Those changes require updating the whole workforce system and reconsidering which skills matter, their distribution across roles, teams and agents, and how much capacity the organization now needs. Setting the target is the easy part. Workflows now change weekly or monthly, while hiring plans, learning programs, career ladders and performance systems still move in quarters or years; this is why People & Talent was the first domain our frontrunners tackled.

Roles shift from execution to judgment and stewardship

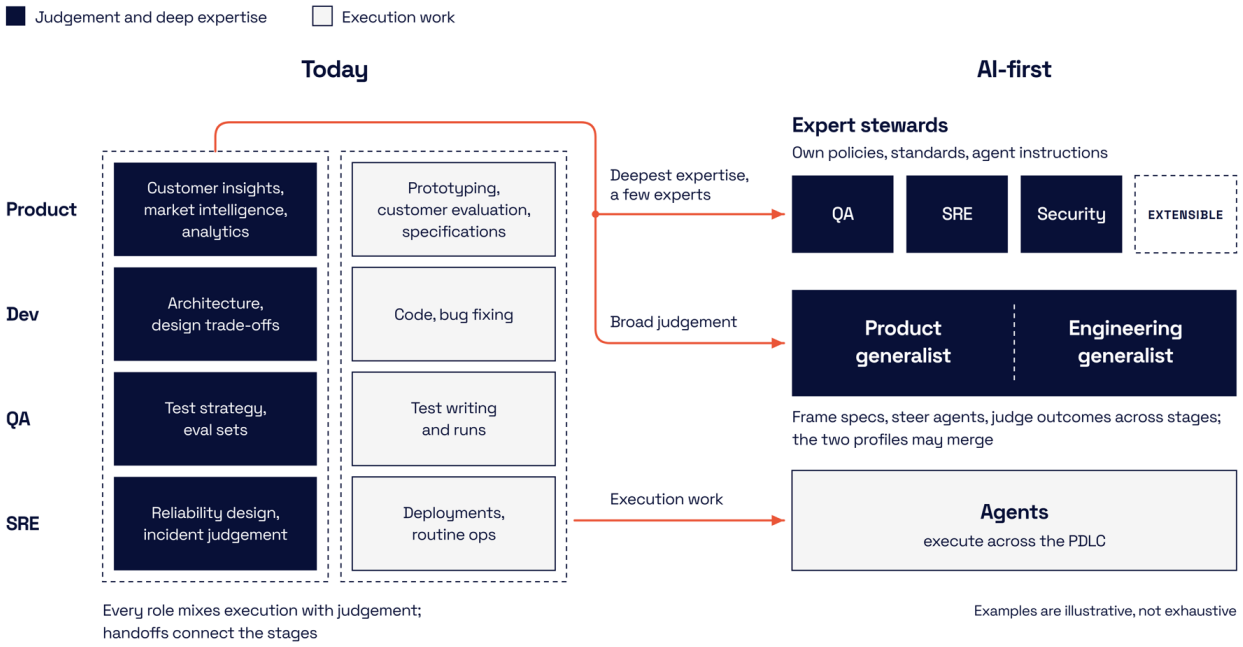
The market has already cycled through several AI-specific role labels, from prompt engineer to context engineer and agent manager. Each reflected a temporary technical constraint of the tools rather than a coherent, durable set of skills. Our guidance to the portfolio is to avoid title-specific redesign until these skills emerge more clearly, and to focus instead on how existing roles absorb AI-first capabilities, assigning responsibilities explicitly across human teams and agents.

Execution shifts to agents, while humans move to judgment and stewardship depending on their profile. The skill mix and its distribution change as the work done by humans moves up a level of abstraction. In this respect AI accelerates, and widens to new roles, a pattern already long-running in engineering, where work and skills progressively moved to higher-level languages and frameworks. In the current shift, agents can increasingly handle the executable part of the work. The consequences are threefold:

- Judgment skills, such as choosing the right problem, framing it correctly and assessing trade-offs, become more crucial for humans. Anthropic's latest Claude Code research shows that across roughly 400,000 sessions, humans made about 70% of planning decisions but only 20% of execution decisions.[10]

[10] Anthropic, "Agentic coding and persistent returns to expertise", 16 June 2026, anthropic.com/research/claude-code-expertise.

- Generalists can now oversee a wider variety of execution tasks that used to require specialist skills, such as writing a test suite or reviewing incident logs, provided they can exercise sufficient judgment across this larger surface.
- A smaller group of humans, typically senior experts, need to become stewards of the agents' execution skills to ensure their continued effectiveness.



The skill shift: generalists steer agents across stages; a small group of expert stewards owns the instruction layer.

QA illustrates the pattern well. Agents can draft and run routine tests, so QA execution work can shift more decisively left and be overseen by generalist programmers, who now need the skill to validate what should be tested and whether a generated test is meaningful. At the same time, a much smaller group of human QA specialists may own the quality release policies, test strategy, eval sets and associated agent instructions. The same logic applies to areas such as SRE and architecture.

At EcoVadis, non-engineers have raised **1,100+** pull requests.

True end-to-end generalists, able to span both product and engineering, are individual exceptions rather than the norm in our portfolio today, since that skill combination is rare on the market. They may become more widespread if, on the engineering side, agent reliability and automated quality gates improve to the point where product-oriented individuals can safely validate technical implementation. At EcoVadis, non-engineers have raised over 1,100 pull requests, mostly internal tools and prototypes. Product changes shipping without engineering support remain the exception, but the first cases are already through. Conversely, the engineering-translation layer of product work, such as drafting specifications, can already be executed by agents overseen by generalist engineers. Those engineers are typically familiar with the working artifacts and trade-offs of such

tasks, and can therefore judge the agents' output. As a result, product profiles can focus on the market and customer-facing layer of the work. Agents will be able to execute traditional BA work here, such as verbatim or acquisition-funnel analysis, but also to deliver sophisticated analytics on more diverse data sources than a mid-sized company can typically generate today. Human product roles will therefore need to retain expert-level judgment in behavioral science, customer experience and market signal, while also stewarding the new data-science capabilities the agents unlock. Opus 2 has for instance made the call to dissolve data science as a standalone function, absorbing the capability into product and engineering roles; agents execute much of the analytical work, so hiring refocuses on people close to the customer problem who can frame the right question and challenge the output.

Behavioral readiness matters as much as skills. Beyond identifying who already has the skills a generalist or specialist role requires, companies should first assess who is already leaning into the new model, who can be coached into it, and where there is a real gap. The people who thrive in the AI-first PDLC are curious, comfortable with uncertainty and willing to learn in public. They challenge agent output without reverting to old execution habits. This mindset shift is not a given. At Fenergo, the difference was not whether people used agents but how: some kept them inside their old workflow, micro-managing step-by-step execution – change this button, add this test – while others framed specs and reviewed whole-workflow outcomes.

- NO-REGRET MOVES

Assess which part of the current workforce already matches, or can be trained into, the generalist AI-augmented profile, the expert-steward profile, or neither.

Delivery consolidates into small, senior teams that own outcomes

Smaller units can deliver the same value faster, if they own the process and the outcome. As agents shrink execution time and let generalists carry out more types of task, teams can cut coordination overhead and therefore cycle time while maintaining output quality. Fenergo unlocked 3x effective engineering capacity, as users got more productive (from 1.3 to 4.9 PR/week/user in 3 months[11]) and as backlog started parallelizing into many smaller delivery streams, each assigned to teams of one to three people. Xceptor's acceleration is likewise structured around teams of two to three. This is easier where teams already work against a value stream or are driven by OKRs, because they have already been through the hard change management needed to embed the ownership, discovery principles and the fast-iteration culture those frameworks require.

> Fenergo unlocked **3x** effective engineering capacity. The backlog split into smaller streams, each run by a team of **one to three.**

[11] Anthropic enterprise usage export for Fenergo (Claude telemetry), January to July 2026.

Fewer management layers are required. The market is heading towards fewer middle-management roles in AI-first PDLC organizations. We already see managers in our portfolio being steered back towards senior individual-contributor roles. This should not be read as an immediate reduction in management load. In the short term, line managers who are technically credible and comfortable with uncertainty are absorbing a large share of the change management the transformation demands. Over time, as small delivery units stabilize and agents take on more monitoring and administrative load, fewer management layers will be needed. Management will remain necessary for people leadership, including coaching and conflict resolution, and for delivery leadership, including objectives, prioritization and judgment escalations. The corollary is the need to build credible career paths for senior individual contributors that do not lead into line management. At Fenergo, the move back to senior IC roles is seen positively, as career progression historically pushed strong ICs into management. Xceptor has rebuilt around this shape: many small teams sit under a leaner management layer. Engineers with the deepest expertise gain more space to tackle the hardest technical problems and more time to spend with product teams, while continuing to contribute directly to development. To anchor the new roles, Xceptor rewrote job descriptions and both its technical and behavioral competency frameworks, and new hires will be assessed to fit this model.

> Xceptor rewrote its **job descriptions** and **both competency frameworks**, technical and behavioral. New hires will be assessed against them.

A structural premium on senior staff requires explicit junior onboarding mechanisms. The shift in the human skill mix from execution to judgment favors a senior-skewed pyramid. These capabilities tend to correlate with experience, which is why the market is moving away from the graduate-heavy model, with new graduates falling to roughly 1 in 10 hires. The senior-to-junior ratio that will ultimately prevail remains an open question. What is increasingly clear is that organizations need a deliberate apprenticeship pathway to keep onboarding and productively employing junior colleagues. One approach being trialled at Fenergo sets out explicit "onboarding pods", in which a senior mentor temporarily oversees a few juniors and deliberately transmits the judgment skills they would once have picked up by trial and error while doing the execution work agents now handle.

- NO-REGRET MOVES

Align your teams to OKRs or a similar outcome-oriented framework; identify the teams that will be hardest to reshape into small delivery units and start reorganizing them proactively: carve out a first pod of one to three people with end-to-end ownership of a well-bounded outcome, then extend the pattern progressively

Core people processes evolve incrementally, on evidence

AI-driven capacity gains assumptions should be rooted in evidence. For workforce planning, the question is how much capacity uplift to assume in headcount plans. A conservative speed-up, for example 10-20%, can confidently serve as the base case[12], with upward revisions conditioned on results observed on average teams and representative projects, once the strongest teams have proven the impact on delivery speed, quality and employee satisfaction.

The hiring process should now explicitly test judgment in working with AI. Most corporate hiring processes have not changed since pre-LLM standards. Technical interviews, notably, still test at length a candidate's ability to build artifacts or make execution choices that AI-driven PDLC has now automated, such as raw coding-ability interviews for engineers. Besides being easily gamed in remote settings, these processes give little meaningful signal on the target skills. Our portfolio has not yet settled on a best-practice approach, and neither has the market: in a recent survey, 71% of tech leaders said AI had made assessing skills harder[13], but only 30% had updated their assessment processes. Even so, two complementary patterns are emerging. One is to include assessments where candidates use AI tooling to reach a given outcome, with interviewers probing their understanding of the outputs and their judgment in steering the agents through iterations. The other is to reintroduce in-person assessments at later recruitment stages, to test systems thinking independently of AI support.

Learning should mostly happen within workflows. Generic AI-101 trainings do little to shift day-to-day practice beyond an initial adoption spike, and role-targeted curricula that could have more lasting impact are proving hard to structure, given how fast the material goes out of date. After trying several paths, we now recommend two complementary approaches, the common lesson being that enablement should sit where people do the work:

> Opus 2's QA teams used peer-programming sessions to build test-suite agents. Their cycle time is now **5–10x faster**.

- Couple autonomous upskilling on off-the-shelf training from the tool vendors, which is updated frequently, with practical proprietary bootstrap kits. Xceptor packaged onboarding into the toolchain with dedicated onboarding, setup and session-unblock skills; Fenergo likewise packaged its agentic loops into pre-installed setups.
- Provide generous live coaching as pods shift to AI-first delivery. Across the portfolio we set up peer-programming office hours and ad-hoc coaching embedded in teams, provided by external partners and progressively owned by internal champions. These helped teams move from individual adoption to more structured workflow deployment. Opus 2's QA teams, for

[12] DX, "State of AI Impact in Engineering: Q2 2026" (500+ organizations, July 2025 to June 2026): median weekly throughput per engineer rose 37% over four quarters. Even the slowest industries within that achieved ~20%.
[13] Karat, "Engineering Interview Trends 2026", 7 January 2026, karat.com/engineering-interview-trends-2026.

instance, used the peer-programming sessions to build test-suite agents and achieve 5-10x cycle-time acceleration.

Explicit redesign of incentives should start with leadership. AI transformation should sit in the objectives of the CTO, CPO or CPTO and the management layer below them. Each leader should define, with their first circle, the evidence that proves material progress, and that evidence should be written into the objectives. It varies by business, but could include roadmap or OKR overachievement, stronger team capability evidenced in telemetry (the flow, outcome and quality metrics we discuss in the ROI section, below), or transformation progress. Vanity metrics such as adoption targets or percentage of code authored by AI should be avoided.

At the operational level, redesigning incentives is harder. The performance gap between AI-proficient talent and the rest of the workforce is widening. Claude Code research showed that novice-rated sessions reached verified success about 15% of the time, against roughly 30% for more proficient ones.[14] In our portfolio, however, we have not yet observed a consistent compensation-pressure pattern for full-time staff. Especially during the transition, it may be enough to reward the expected behaviours within existing performance grids. Our approach has also leaned heavily on non-monetary incentives, including board-level visibility for teams driving AI-first work, access to the most interesting delivery problems, and external recognition through public speaking. Additional monetary measures may need to be considered, depending on whether the performance gap proves permanent and competition for talent accelerates.

- NO-REGRET MOVES

Select a few frontrunner teams to go AI-first. Add a practical AI test to the recruitment process for the highest-volume role family first. Build self-starter kits with packaged AI setups, and provide coaching options. Define additional leadership objectives for the next review cycle. Check that incentives for individual contributors do not encourage behaviours contrary to the new operating model, e.g., focusing on output volume.

[14] Anthropic, "Agentic coding and persistent returns to expertise", 16 June 2026, anthropic.com/research/claude-code-expertise.

03

Governance & Compliance

What controls need to change when agentic workflows enter the PDLC?

Governance shifts from documents to code

Agentic delivery changes what governance has to check. Approvals, artifact trails, quality gates: most of that evidence is a by-product of the operating model described above.

The portfolio pattern follows that line. Where governance can be expressed as code, in gates, hooks and repositories, our companies run ahead of most of the market; where it requires legal analysis or organizational decisions, such as threat models, contract reviews and regulatory positioning, the work is only starting. Two fronts demand attention first. One is the control surfaces autonomous agents need; the other is the external obligations that move on regulators' and customers' calendars rather than the company's.

Controls embedded in existing development toolchains are still catching up with the sprawl of non-human identities, which the Cloud Security Alliance now estimates at roughly 45 for every human one, with agent credentials likely to fall outside existing audit scopes[15]. Our companies are actively front-loading three questions - what agents may touch, what evidence exists of what they did, and what new failure classes they introduce.

Identity, auditability and threat models need reinforcing

Identity and access governance have to be reinforced and automated further as agent autonomy increases. Any part of identity and access governance that is manual or ill-defined today will hurt when applied to automatically spawned agents, leaving them either too constrained to be useful or over-permissioned and riskier by default. Tools that let agents inherit their user's permissions, limited to the surfaces they work on, are already reaching the market and should make this much easier over the coming months. In the meantime, the portfolio has run ahead of the problem, governing agents through dedicated toolchain automations that hold at current scale. Xceptor, for instance, distributes agent access to repositories and work items through an admin-controlled plugin, so permissions are set and updated centrally.

[15] Cloud Security Alliance, "The Non-Human Identity Governance Vacuum", 20 May 2026, labs.cloudsecurityalliance.org.

Agent-level auditability can already be enforced programmatically. In our Forge companies, agents' intermediate artifacts are themselves versioned and tracked automatically. Fenergo, for example, adds hooks that enforce changelog entries and version bumps on every build, so the record is captured consistently and effortlessly. EcoVadis has made an audit trail at the agent decision layer a prerequisite for scaling agents into production. More granular lineage, tracing a production change back to the model version, prompt context and agent invocation that produced it, remains harder to enforce today. Commercial agent platforms and monitoring software editors are however starting to ship increasingly sophisticated audit features that will make lineage less bespoke to assemble for companies relying on off-the-shelf solutions. It will remain a key concern for those developing their own agent harness, of which we have a few examples in the portfolio for more bespoke workflows, mostly outside the PDLC.

> At EcoVadis, an **audit trail** at the agent decision layer is a **prerequisite** for scaling agents into production.

Threat models will need upgrading for agent-specific risks. The OWASP agentic Top 10[16] maps ten agent-specific attack classes, from goal hijack and tool misuse to memory poisoning and rogue agents, several of which have already caused real incidents in the market, even at this relatively early stage of autonomous-agent adoption. Our portfolio's cybersecurity teams are actively reviewing and upgrading the existing controls. In so volatile a space, the effort will have to continue, scaled to how widely agents are deployed and to the permissions they hold.

Contracts and regulation lag, so involve legal early

In a B2B world, contractual conditions can constrain AI-first delivery. At one portfolio company, the requirements data itself is treated as client IP under existing contracts, so exposing that data to AI tools requires explicit amendments first. More generally, data-use, IP, audit-rights and subprocessor clauses were not written with agentic toolchains in mind and can block AI-first delivery outright, which is why reviewing and updating the relevant clauses is crucial.

Regulatory obligations do not yet fully account for non-human identities. Compliance frameworks such as SOC 2, ISO 27001 and PCI DSS assume human identities in access control, as do four-eyes review obligations in regulated paths. AI-specific regulation is recent and evolving rapidly. The EU AI Act's transparency obligations for deployers apply from August 2026, though the machine-readable marking requirement for generative content already on the market is deferred to December 2026.[17] The Act's text does not yet include detailed provisions specific to autonomous agents, but the Commission's June 2026 draft guidelines have begun to address them, confirming that agentic systems interacting directly with people must identify themselves as AI. The FCA has flagged human-in-the-loop protocols and audit trails as live issues, with practical guidance on how Consumer Duty and the SMCR apply to AI now expected by end 2026; and at Bank of England

[16] OWASP, "Top 10 for Agentic Applications", 9 December 2025, genai.owasp.org.
[17] Gibson Dunn, "EU AI Act Omnibus Agreement: Postponed High-Risk Deadlines and Other Key Changes", 27 May 2026, gibsondunn.com/eu-ai-act-omnibus-agreement-postponed-high-risk-deadlines-and-other-key-changes.

roundtables in early 2026, supervised firms challenged whether blanket human-in-the-loop review can scale.[18]

Bringing legal and compliance in early ultimately accelerates scaling. These functions are sometimes treated as a brake. In our experience, bringing them in earlier has let us identify and fix gaps before they became blockers during scale-up. The same holds for auditability, cheaper to embed early and by design in the workflow than to retrofit later.

- NO-REGRET MOVES

Keep versioned logs of agents' delivery activity; audit which IAM processes are still manual or ill-defined; and scope agents by default to least-privilege, task-bound access.
Run a light, scoped legal check: is there anything in current contractual conditions or regulatory obligations that prevents exposing particular data to AI tools, or that limits how autonomously AI tools can be used?

[18] Bank of England, "Summary of AI roundtables", February 2026, bankofengland.co.uk/minutes/2026/february/summary-of-ai-roundtables-feb-2026.

04

ROI

How can companies govern AI spend and frame ROI decisions as usage scales?

Three questions follow about the meaning of delivery metrics once agents inflate them, the AI consumption as the variable production cost to be planned, and the steering of J-shaped returns.

Volume metrics inflate, so measure flow and quality instead

Operational metrics need to match the new production model. Metrics built on raw output volume, whether lines of code, pull-request counts or story points, assume production is scarce. Once agents make production cheap, volume metrics inflate without a clear understanding of whether value moved. Quality metrics are more resilient to the change, though their definitions need to evolve. For instance, code maintainability for agents is not the same as maintainability for humans. Our portfolio has already started retiring current volume KPIs wherever agentic delivery runs, while reinforcing flow and quality metrics discipline. Fenergo excluded lines of code and story points from its velocity definition and reports cycle time, deployment frequency, defect rate and feature throughput to the board; its AI-first pods decompose work into 20 to 40 same-session issues, a granularity at which story points stop meaning anything. Xceptor collapsed quality reporting into a single quality-health indicator, with industry-standard metrics underneath.

Isolating AI impact is hard, but pragmatic triangulation is possible. Existing baselines are rarely clean or granular enough to rigorously isolate the effect of AI on a specific KPI. We have adopted a pragmatic option and track the overall trend in a broad set of metrics against the baseline before widespread individual adoption of AI and before the scaling of AI-first PDLC. We have paired this with like-for-like comparisons on well-known, repeated activities, such as a type of build the team has delivered many times. Measuring systems (e.g., DX, Jellyfish, Plandek) now provide a richer, more structured way of collecting these insights, and have helped us refine our understanding of the impact across the portfolio. Opus 2 shows what this instrumentation looks like at company level: it has implemented a layered measurement stack combining quarterly developer-experience surveys across every squad, DORA delivery metrics, sprint-level flow measures such as volatility, predictability and spillover, and AI-specific impact data, all benchmarked against peer cohorts across industry rather than internal history alone. This setup allows the engineering management to make timely tradeoffs as AI-native development impacts all those metrics at once.

> **"AI-authored code was running 20% above the top-decile cohort while merge frequency sat 30% below it. That told us generation had outpaced absorption, and to invest in the release pipeline, not in more generation.**
>
> Tiama Hanson-Drury, Chief Product and Technology Officer, Opus 2

As we have already discussed throughout this paper, the conversation is now rapidly shifting from productivity ROI to business ROI. Telemetry should therefore extend past the PDLC, ultimately linking spend on delivery efficiency to the outcomes it served, e.g., revenues from faster time-to-market, reduced customer churn, lower cost to serve.

• NO-REGRET MOVES

Stop using raw output metrics for decisions, gear instead to flow, outcome and quality metrics.

AI consumption is now a variable production cost

AI tooling entered most budgets as a per-seat license but should be seen as a metered industrial input. Spend scales with parallel agent runs, context size, verification loops and test compute rather than with headcount, and the curve is non-linear because agentic sessions grow their own context as they work[19]. The market shows what linear extrapolation from a pilot does; across 15 companies canvassed by practitioners in April 2026, token spend rose roughly 10x in six months and the per-user caps set at the start had to be raised or scrapped.[20] Uber exhausted its 2026 AI-coding budget in four months, then imposed a $1,500 monthly per-tool cap[21]. Fenergo started with a deliberately low monthly cap per developer and a come-and-talk-to-us trigger; developers hit their caps faster and faster, several in a single day, and engineering raised the cap by an order of magnitude. Human attention is still the larger cost, so caps exist to create visibility and trigger a conversation when consumption jumps. But the aggregate is real and grows with every cohort, which is why the visibility matters now. It is cheap per feature today but non-linear in aggregate at full rollout, which is why generous caps are paired with telemetry.

The flat cap is a blunt instrument, a stopgap rather than the target state. The first refinement already visible in the portfolio is tiering by role before rollout. Fastmarkets is baselining expected

[19] Vantage (Casey Harding), "Agentic coding costs", 15 April 2026, vantage.sh/blog/agentic-coding-costs.
[20] Pragmatic Engineer, "The Pulse: token spend breaks budgets – what next?", 30 April 2026, newsletter.pragmaticengineer.com/p/the-pulse-token-spend-breaks-budgets.
[21] TechCrunch, "Uber caps employee AI spending after blowing through budget in four months", 2 June 2026, techcrunch.com.

spend per user group and rolling the tiers up into a cost forecast for the function, with delivery roles in the lowest spend tier, followed by engineers above them and the data and innovation team ring-fenced at the top; a second portfolio company has since adopted the approach. A second layer tiers by objective rather than role: EcoVadis ring-fences a dedicated quota for high-impact experiments, its "mission to Mars", above the default "mission to Moon" allowance, protecting exploration while telemetry matures. Where a company starts on this spectrum is a risk-appetite choice: Fastmarkets deliberately began with block-then-approve and role-based usage expectations, while EcoVadis deliberately began open, to establish a baseline of what is possible.

Fastmarkets tiers expected spend **by user group**: delivery roles lowest, engineers well above, the data and innovation team ring-fenced at the top.

"Ultimately it's horses for courses, based on the value we can unlock.

Richard Peers, Chief Technology Officer, Fastmarkets

Richard is Chief Technology Officer at Fastmarkets, a leading global commodity price reporting and market intelligence agency, where he leads the global technology team, driving growth through large-scale digital transformation, product innovation, data and AI.

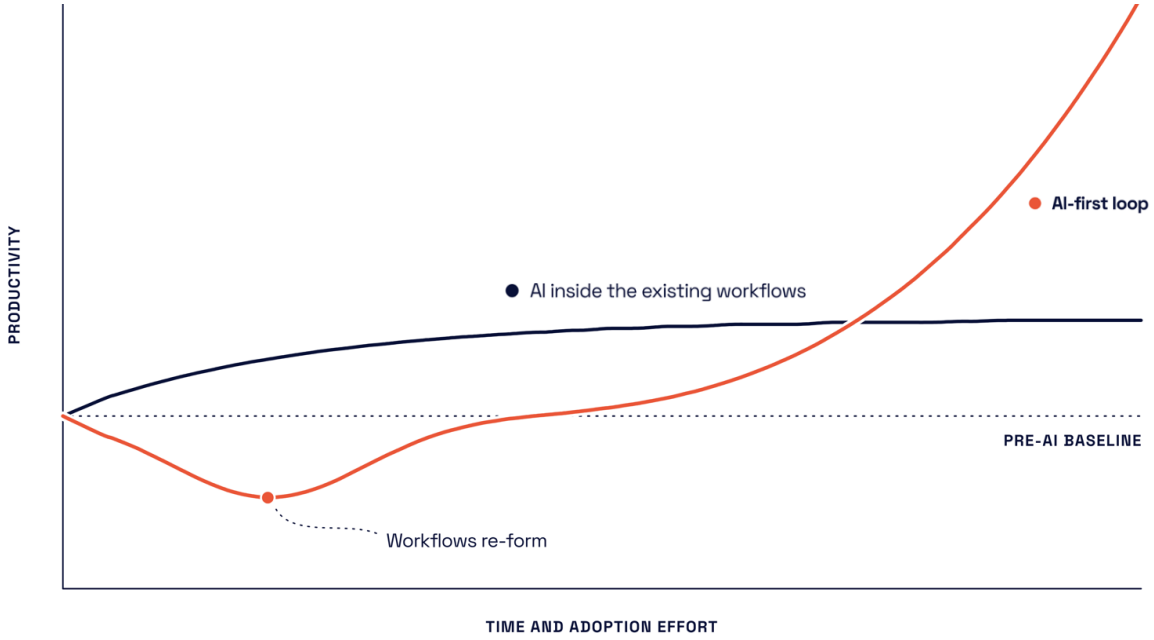
But tiering shares the flat cap's blind spot: it controls how much each role can spend, not what the spend bought. The end state is telemetry enabling granular attribution of spend, tagging it to the feature or repository and, ultimately, outcome it served. Cost engineering for agentic delivery is new for everybody, but several of our portfolio companies are starting to instrument granular telemetry. Xceptor, for instance, measures token spend per feature and per PDLC skill. For its connector feature builds, the overall cost came in ~80% below pre-AI-PDLC estimates. Within that, tokens accounted for only about 2% of total costs, with the rest reflecting the human time spent. While the telemetry that allows cost interventions rather than blanket caps matures, broader FinOps levers ranging from model routing to provider negotiation can already be rolled out. We are supporting this throughout our portfolio via our alliances with hyperscalers and partners. EcoVadis has taken this furthest, standing up a dedicated cost-optimization guild that crunches usage telemetry across agents, skills, connectors and engineers, in dollars rather than tokens, prioritizes optimization actions, pilots them with a small group and generalizes only where impact is measured without quality loss.

- **NO-REGRET MOVES**

Set tiered usage caps, kept generous, while building towards attribution, with spend telemetry tagged ideally per-feature. Implement basic FinOps levers such as setting cheaper default models.

ROI follows a J-curve, paying off at scale

The return lags the adoption. The DORA ROI model[22] describes a J-curve with productivity dips while workflows re-form, then recovers once they settle. Where a company sits on that curve depends on whether it has made the full-workflow switch. A team still running AI inside its old process has not reached the upturn, so its ROI looks limited because it is being measured at the bottom of the curve. The self-diagnosis of our frontrunners before we started the Forge program captures where most companies stand: "we have adopted faster than we have changed". Past the upturn, the harder work is scaling the new ways of working from proven pods to the full organization, where most of the return sits.



The ROI J-curve: productivity dips before compounding for teams that rebuild AI-native workflows

Reading the curve requires care as pilot numbers will often overstate the return, based on hand-picked teams and estimated baselines for lack of better options. Long-term progress needs to be judged on a broad baseline, over time. The assumptions behind the ROI view tend to move fast on both sides of the ledger. Capabilities improve with every model generation, and costs swing with releases and competition among providers, so the ROI read should be revisited at least quarterly. Either way, agentic delivery is now the standard way software gets built. If returns disappoint even after accounting for the switch and the scale-up, the answer must be to rethink and fine-tune the approach, e.g., to re-sequence adoption scope, adapt tooling choices, tighten usage discipline.

[22] DORA, "ROI of AI-Assisted Software Development", May 2026, reported by InfoQ, infoq.com/news/2026/05/dora-roi-ai-assisted-dev-report.

Conclusion

We hope that what we have gathered here proves useful to leaders setting out on a transformation of their own. We have deliberately avoided being overly prescriptive about the change itself, because every company starts from a different place. But across our interactions with our portfolio leaders, a few hallmarks of successful transformation have less to do with method than with posture.

- **Lead from beside**

Good leaders do not sit in front of their team when they are facing a problem. They sit next to them, so they see the problem from the same angle. The leaders who moved their companies sat at the keyboard and did the work themselves, taking the same risk and using the same tools, they were asking everyone else to use. Being first to try, and first to be wrong, is what made it safe for their teams to follow.

- **Hold the vision and the execution tighter**

Few leaders are equally strong at keeping sight of both where the technology and the business are heading and at solving daily execution challenges. AI tightens the loop between vision and execution, and leaders now need to stretch on both ends. As if this was not enough of a challenge, CPTOs are also expected to drive the implementation of the required transformation enablers by working with other functions, e.g., aligning with GTM, embarking customer support, and working with HR to redesign the People and Talent strategy.

- **Prepare the rest of the C-suite**

In our portfolio, CPTOs often led the change because AI capabilities grew fastest in product and engineering, closest to their craft. But AI is now reshaping the daily work and the operating model of every function, from Sales to Operations. The stretch that has been asked of CPTOs, between setting the vision and role-modeling how the job itself must change day-to-day, is coming for the rest of the C-suite.

The field will keep moving, and our answers will move with it. What we are confident will not change is the intensity of the effort required of management. Someone has to go first, take on the risk they are asking of others, and show that on the far side of the cliff lies not a fall but flight. Identify that leader, give them a strong mandate, and the rest of this paper becomes a question of disciplined execution.

APPENDIX

The no-regret moves at a glance

Each subsection of this paper closes on a no-regret move. Collected here, they are the actions to start now, whatever direction the field takes next.

Product & Engineering

1. Identify existing sources of discovery signal and make them accessible to agents; use rapid prototyping to pull customer validation forward on the most high-value development streams.
2. Consolidate and upgrade source control and CI/CD platforms where the base is fragmented or close to end of life; baseline infrastructure load and expand capacity where it is close to becoming the bottleneck.
3. Create context and instruction repositories with named expert maintainers, PR-based contributions and staleness checks; define which standards are mandatory and which teams may extend or fork; for third-party-delivered solutions, make these artifacts part of the deliverables.
4. Enforce multi-dimensional agentic review (security, quality, architecture) ahead of human review and track false positives and false negatives to improve the filter over time; write down the risk criteria that set the level of human review per change, keeping strict human merge ownership and four-eyes where regulatorily mandated.

People & Talent

1. Assess which part of the current workforce already matches, or can be trained into, the generalist AI-augmented profile, the expert-steward profile, or neither.
2. Align teams to OKRs; identify the units hardest to reshape into small delivery teams and start reorganizing them proactively, beginning with a pilot pod of one to three people owning a single outcome.
3. Go AI-first with a few frontrunner teams; add a practical AI-steering test to recruitment for the highest-volume role family; package self-starter kits and coaching; set leadership objectives on the transformation; check individual incentives do not reward output volume.

Governance & Compliance

1. Keep versioned logs of agents' delivery activity; audit which IAM processes are still manual or ill-defined; scope agents by default to least-privilege, task-bound access.
2. Run a light, scoped legal check on whether current contractual conditions or regulatory obligations prevent exposing particular data to AI tools, or limit how autonomous AI tools are used.

ROI

1. Stop using volume metrics for decisions; steer on flow, outcome and quality metrics against a pre-AI baseline captured before the old measures stop being collected.
2. Instrument per-developer and per-feature AI spend telemetry; set tiered guardrails on top, deliberately generous where the bottleneck sits; implement basic FinOps levers such as cheaper default models.
3. Keep exploration generous; gate large-scale deployments on success and rollback criteria written before scaling, judged on trend against baseline and re-checked on the planning calendar.

APPENDIX

Glossary

Artifact — Any working output of the lifecycle: a specification, a test plan, or a package built for release.

Blast radius — How much breaks if a change goes wrong.

CI/CD — Continuous integration and continuous delivery. The automated pipeline that builds, tests and ships every change to the code.

Cycle time — Elapsed time from starting a piece of work to it being live. Not the same as productivity, which is output per person.

Deployer — Under the EU AI Act, the organization using an AI system, as distinct from the provider that builds it. Obligations differ.

DevEx — Developer experience. How easy the day-to-day of building software is, measured by survey.

DevOps — The tooling and practice that automate building, releasing and operating software.

DORA metrics — Set of widely-adopted delivery measures, including KPIs like deployment frequency, lead time, change failure rate and mean time to restore.

FCA — The UK Financial Conduct Authority. Consumer Duty is its rule requiring firms to deliver good outcomes for retail customers.

IAM — Identity and access management. The systems governing who, or what, may access what.

Least privilege — Granting only the access a task strictly requires, and no more.

Lineage — In this context, tracing a change in production back to the model, context and agent run that produced it.

LLM — Large language model. The underlying AI model that everything else is built on.

Non-human identity — A credential belonging to a machine or an agent rather than a person.

OWASP — Open security community whose Top 10 lists are the industry reference for attack classes.

PDLC — Product development lifecycle. Idea and discovery through design, build, test, release and operation.

Pull request (PR) — A proposed change, reviewed before it is merged into the shared codebase. The standard unit of review.

QA — Quality assurance. The testing discipline.

Repository — One codebase held in source control. A fork is an independent copy of it.

SDLC — Software development lifecycle. The engineering part of the PDLC: build, test and release.

SMCR — Senior Managers and Certification Regime. UK rules making named senior individuals personally accountable.

SOC 2, ISO 27001, PCI DSS — The security standards enterprise customers audit against: US service providers, international, and payment card data.

Source control — The system holding the code and every change ever made to it. Git is today's standard.

SRE — Site reliability engineering. Keeping systems running reliably in production.

Telemetry — Data collected automatically on how systems, agents and teams are working.

Contributors

ASTORG DIGITAL, TECH AND DATA OPERATIONS TEAM

Valérie Legat
Head of Digital, Tech & Data Operations

Andrea Siani
AI & Data Operating Director

Anastasiia Ivankova
Associate

PORTFOLIO CONTRIBUTORS

Michael Kinloch
SVP Engineering
Xceptor

Niall Twomey
Chief Technology Officer
Fenergo

Hassan Lachgar
VP of Engineering
EcoVadis

Krishna Corinaldesi
Director of Engineering
Fastmarkets

Tiama Hanson-Drury
Chief Product and Technology Officer
Opus2

Richard Peers
Chief Technology Officer
Fastmarkets

Matt Rhodes
Chief Transformation Officer
EcoVadis