

AI Unfiltered: From Experimentation to Delivery



enabl'is.

How Organisations Move from AI
Experimentation to Operational Impact

Executive Summary

enablis.

The reality is most organisations are experimenting with AI, few are seeing consistent, production-level impact. The gap isn't tools; it's delivery capability. AI delivers value only when it is embedded into the way work is delivered.

We're seeing



AI rapidly identifying bottlenecks



Increasing domination of AI-native development



Transformation of testing and delivery teams

What's changing



Faster development cycles



More output per engineer



New bottlenecks in testing, review and release

Why initiatives stall

// THE PROBLEM



Disconnected pilots
No ownership
No metrics

// THE RESULT



No adoption
No accountability
No impact

How Enablis helps

We move organisations from experimentation > capability > impact



Strategic discovery
(30 / 60 / 90 roadmap)



Hands-on delivery with real workflows



Adoption scaling with governance and enablement

Who this report is for



CTOs & Engineering Leaders



Heads of Delivery / Platforms



CIOs driving AI transformation

Conclusion

Winning organisations aren't experimenting more. They are embedding AI into delivery.

Our Perspective

enablis.

Enterprise AI conversations often start in the wrong place. They focus on risk, governance and restrictions, rather than on the practical question of how AI can improve the way work actually gets delivered.

Across engineering-led organisations, AI is shortening feedback loops, reducing delivery friction and unlocking work that has sat in backlogs for years. The organisations creating advantage are those embedding AI directly into the engineering delivery model, not treating it as a separate innovation programme. AI becomes part of how delivery operates, supported by clear ownership, governance and

measurable outcomes. This whitepaper is written from that perspective. It draws on what we see across client engagements and on our CTO, Ed Marshall's direct experience leading AI-enabled engineering. It is not a guide to managing AI risk. It is an argument for building AI capability, deliberately, at pace, with the governance in place to sustain it.

AI capability is no longer optional for modern organisations. The organisations that treat AI as a delivery capability will outperform those that treat it as an experiment to contain. That is the view this whitepaper takes, and we have not written it to be neutral about it.



The goal isn't to write more code.

It's to deliver features to market faster and more robustly."

Ed Marshall, CTO
Enablis



AI is transforming delivery

AI is already accelerating parts of the engineering lifecycle, from requirements and coding to testing and release. The organisations seeing real impact are embedding AI directly into everyday workflows rather than treating it as a side experiment.



Capability requires governance, ownership and enablement

The organisations stalling on AI are not short of ambition or tools. They are short of ownership, governance and a clear line between how AI is being used and what outcomes it is expected to deliver. Fixing that is an engineering leadership problem, not a technology problem.



AI changes the economics of engineering

AI-assisted development increases what experienced engineers can deliver. As cycle times shrink, the economics of software delivery shift from scaling teams to increasing

01

AI is already integral to the engineering pipeline



Google and Microsoft CEOs say 20–30% of code in Microsoft repositories is written by AI.

These signals show the direction of travel across modern engineering organisations. (Source: [CNBC](#))

AI is already embedded across the delivery pipeline, from requirements and design through to testing and release.

Requirements and scoping

AI is helping teams clarify business requirements earlier in the process. It can analyse proposed changes, identify potential impacts on the codebase and help teams understand the scope of work before development begins.

This support is not limited to engineers. Product managers, delivery leads and analysts are increasingly using the same tools to structure requirements and reduce ambiguity.

AI-assisted coding

Developer effectiveness has improved significantly through AI-assisted coding, particularly when organisations adopt tools deliberately and securely.

Commonly standardised tools include:



GitHub Copilot with Visual Studio Code, often enabled through Microsoft enterprise licensing



Cursor, where teams are free to select best-fit development environments



Claude Code, which is gaining interest for scaling AI-enabled workflows

Tool choice alone does not create advantage. Advantage comes from integrating those tools into real engineering workflows.

As development accelerates, constraints shift downstream

Review, integration and release

When development accelerates, constraints move downstream.

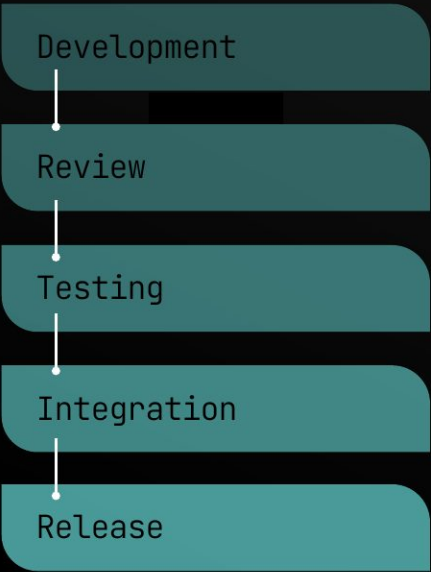
Organisations are now applying AI across CI/CD pipelines, peer review workflows and release processes to prevent new bottlenecks from forming.

Testing

AI is increasingly used for test generation, validation and execution. As build velocity increases, testing capability must increase with it.

Principle: Delivery pipelines behave like systems

Accelerating one stage of the delivery pipeline does not remove the constraint. When AI increases development speed, the bottleneck typically shifts to:



High-performing organisations optimise the **entire pipeline**, not individual stages. As organisations begin embedding AI across the engineering pipeline, a new challenge emerges: turning early experimentation into sustainable capability.



As soon as you unblock one bottleneck, you'll get another."
- Ed Marshall, CTO Enablis

The shift from assistant to agent

For the past two years, AI in engineering has operated as an assistant. That model is now changing.

// ENGINEERING IS CHANGING

Focus Area	Assistant Model	Agent Model
Role of AI	Supports tasks	Executes workflows
Human role	Writes + prompts	Directs + validates
Workflow	Step-by-step	Autonomous, multi-step
Output	Suggestions	Completed work
Impact	Incremental gains	Step-change acceleration

A new generation of agentic systems is moving from suggestion to execution. This can include:



Writing code and running tests



Raising pull requests



Iterating on failures autonomously



Operating without step-by-step human direction

Agentic workflows are already operating inside production environments. Tools such as GitHub Copilot, Cursor and Claude Code demonstrate how AI is moving beyond suggestion into execution within real delivery workflows.

Spotlight: enablis.

**Enablis built its own PSA platform
with agents - in 26 days**

The situation

Enablis needed a PSA platform to run the consultancy from first lead to final invoice. Build vs buy. Off-the-shelf meant configuration debt, lock-in, and a roadmap we didn't control.

The approach

We built it. Agentic delivery was the chosen route, not a test.

- ✓ 2-person human team set goals, not tasks
- ✓ Agents planned, built, tested and raised PRs
- ✓ Guardrails built in from day one (boundary checker + full test suite failing the build on anything off-limits)

The outcome



**1 every
40 minutes**

How often our agent team was merging pull requests over a 2-day period

- ➔ 404 commits
- ➔ 380 pull requests through full continuous integration
- ➔ 21 domains
- ➔ 2,606 tests
- ➔ 52 pull requests in a single 2-day stretch
- ➔ 2 developers steering several workstreams in parallel

What it demonstrates

Direct and validate, let the controls do the gatekeeping. This is what build-vs-buy looks like when AI is part of the engineering system.

The role of the engineer doesn't disappear - it *evolves*.

The implications for delivery are immediate:



Engineers move through backlog work faster, with fewer manual steps



Throughput increases across the delivery pipeline



Value shifts to defining constraints, validating outputs and ensuring alignment with architecture, security and delivery requirements

Engineers who can direct AI systems effectively become more leveraged within the delivery model. Those who cannot adapt to this mode of working may see their relative impact decrease.

Organisations are now beginning to consider what this shift means operationally.



How workflows operate when AI generates code independently



How governance applies to AI-generated changes



How delivery adapts as throughput increases

The organisations addressing these questions now will be better prepared as agentic workflows become more common across engineering environments. For many organisations, that shift is likely closer than current planning assumptions suggest.



02

Capability requires governance, ownership and enablement

Most AI pilots fail because they are disconnected from real engineering environments and no one owns adoption.

What typically causes pilots to stall

Challenge	What happens	What changes the outcome
Context mismatch	Pilots disconnected from real delivery constraints fail to build trust	Demonstrate AI inside the live delivery environment
Risk-only ownership	AI becomes a governance topic without delivery accountability	Assign ownership for enablement and risk
Delivery-only ownership	Productivity gains lack oversight	Introduce guardrails alongside experimentation
Abstract pilots	External examples fail to resonate internally	Use real codebases and workflows

Integration complexity, security controls and release policies quickly expose gaps between experimentation and production.

Adoption accelerates when AI is demonstrated inside the organisation's delivery workflows, where engineers can see how it affects day-to-day work



“When we take their codebase and demonstrate how it can be used, that's the light bulb moment.”

Ed Marshall, CTO
Enablis

AI pilots don't fail because of technology. They fail because organisations aren't ready to adopt them.

In one financial services organisation, an AI pilot produced promising results in a controlled environment but exposed gaps when assessed against the realities of the organisation's data estate, governance processes and operational ownership. The pilot demonstrated that the technology could work, but not that the organisation was ready to deploy it at scale. Rather than

forcing deployment, the focus shifted to building the foundations for adoption - aligning AI initiatives with the data strategy, establishing governance forums to evaluate use cases, and clarifying accountability for risk and adoption.

This gap between experimentation and production delivery is where governance and ownership become critical.

Governance enables sustained AI adoption

AI adoption only scales when organisations establish clear guardrails around how the tools are used. Governance is not about slowing experimentation. It is what allows organisations to move quickly while maintaining control over quality, security and cost.

Teams need:



Guidance on acceptable use



Access to the right tooling



Visibility into how AI is being applied across the organisation

Without those guardrails, adoption becomes inconsistent and risks are difficult to manage. With them, teams can move quickly while leadership maintains oversight of quality, security and outcomes.



Governance isn't the most sexy word in the world... but for us governance is about the guardrails – support, tooling and reporting that allow a business to manage adoption effectively."

Ed Marshall, CTO
Enablis

Turning experimentation into capability

Most organisations already know the tools exist.

The real challenge is using them inside real delivery workflows and holding teams accountable for results.

AI adoption only becomes meaningful when teams apply the tools to live codebases, real backlog items and real delivery constraints. That requires clear ownership, defined usage boundaries and measurable outcomes.

The organisations making progress are not discussing AI in isolation. They are applying it directly to how software is built, reviewed and released.

```
led to connect to the databas
user data
():
g user data..."
data loaded.")
("jane.doe", "securePassword
registration
r(username, password):
stering user: {username}..."
ame or not password:
egistration failed: Invalid
d = hash_password(password
ername} registered succo
username, hashed_pass
```

Phases of applying AI in delivery

Phase	Focus	What it looks
Before	Define the real constraint	Identify where delivery slows, where quality drops and where handoffs create delay
During	Apply AI to real friction	Work against real workflows or codebases and define acceptable usage
After	Convert insight into capability	Enable tooling, assign ownership and track adoption and delivery outcomes

A recent example is Yoti, a regulated digital identity platform (full case study below).

Rather than enabling AI tools and observing uptake, Yoti set explicit adoption targets and aligned leadership and governance early.

60% Yoti engineers embedded AI into their workflows **within 30 days**

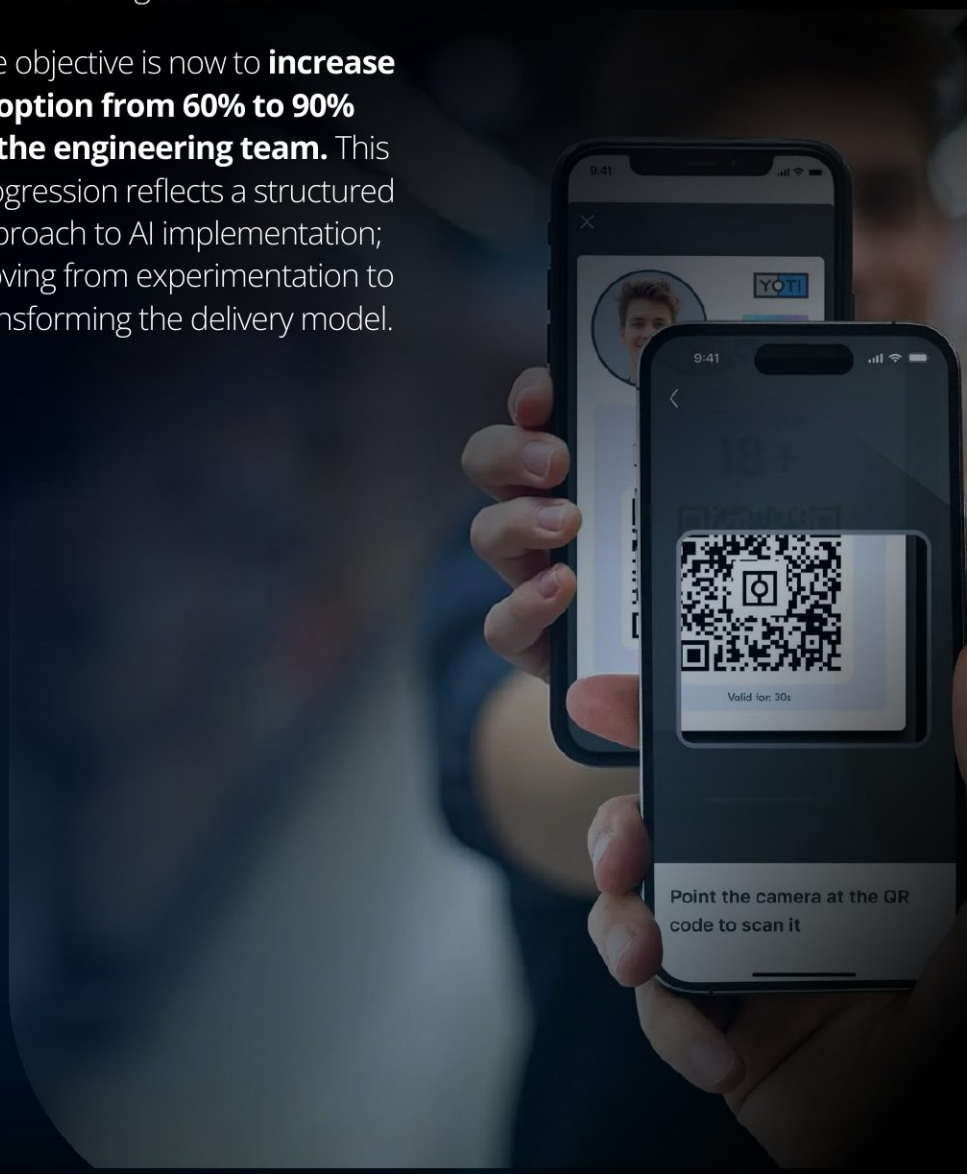
Phase 2 target:

90%
adoption

Within **30 days**, more than **60% of engineers** were **actively using AI tools** across real engineering workflows.

That momentum triggered **Phase 2 of the programme**, focused on deeper enablement, governance and scaling adoption across the organisation.

The objective is now to **increase adoption from 60% to 90% of the engineering team**. This progression reflects a structured approach to AI implementation; moving from experimentation to transforming the delivery model.



03

AI is changing the economics of engineering

More than productivity, AI adoption changes the economics of how software is delivered.

When development cycles compress, organisations can deliver more features with smaller teams and fewer coordination delays.

AI-enabled environments show three structural changes:

Smaller teams	→	greater delivery output
Senior engineers	→	increased leverage
Accountability	→	stronger quality, testing and security

This pattern is already visible in AI-native organisations.

Anthropic has published guidance on how its engineers use Claude Code as a core part of their development workflow. Rather than using AI as an occasional assistant, engineers work with Claude Code to handle multi-step tasks autonomously, including writing, testing and iterating on code within a single workflow. The tooling is not layered on top of how software is built. It is integrated directly into the development process. For engineering leaders, the implication is structural. The challenge is not simply introducing AI tools into existing delivery workflows, but considering how those workflows evolve when AI becomes part of the engineering system.

As AI increases development throughput, review, testing and release capacity must increase alongside it. Teams therefore need clear acceptable-use guidance, appropriate tooling provision and visibility into usage, so quality does not erode as speed improves.

What happens when AI increases development throughput



Organisations that adopt AI-assisted development quickly see a secondary effect: the volume of code moving through the delivery pipeline increases.



As engineers produce code faster, pull request volume rises and review cycles become more demanding. Testing workloads also increase as more changes move through CI/CD pipelines.



In one engineering organisation, the introduction of AI-assisted development tools significantly increased the rate at which engineers could generate code and submit pull requests.



While this improved implementation speed, it quickly shifted pressure to the review stage of the delivery pipeline. Review queues grew, and testing and validation processes became the new constraint on delivery. The technology had successfully accelerated code generation, but the surrounding processes had not yet adapted to the increased throughput.

Industry signals already reflect this shift. Microsoft's research on GitHub Copilot adoption found that **completed tasks increased by 26% across 4,867 developers** working in enterprise environments. Faster code generation translated directly into more code moving through review and integration workflows.

Teams that plan for this increase review capacity, strengthen automated testing and ensure release processes can absorb the higher change volume.

Why senior engineers outperform in AI-enabled teams

Senior engineers tend to outperform in AI-enabled environments for three reasons:



Research supports the variability of productivity gains.

A 2024 controlled study estimated developers using AI coding assistants completed tasks around 21% faster on average.

The gap between senior and junior engineers in AI-enabled environments is wider still – experienced engineers get more from the tools, faster.



1. Tool direction

They translate ambiguous requirements into precise instructions and constraints.



2. Risk detection

They recognise architectural, security and operational risks earlier.



3. Reduced handoffs

Experienced engineers move work from concept to implementation with fewer coordination delays.

Rethinking engineering economics

AI tool spend can appear expensive when viewed in isolation. It looks very different when measured against delivery cost.



"Good engineers will quite easily burn through \$150 of credits in a single day. That might scare some businesses because they think costs are going out of control. But that's misunderstanding what the cost is.

If you bring a feature down from two weeks to a day, the total cost of development is significantly cheaper."

Ed Marshall, CTO
Enablis

When cycle times compress, delivery economics shift from headcount scaling to capability density. A smaller AI-enabled team may incur higher daily tooling cost but deliver significantly more value per engineer. The question leadership should be asking is not "what are we spending on AI?" It's "what is each feature costing us to deliver, and is that number going down?"

For that shift to work, governance and economics must align. Organisations need:



Clear guardrails on acceptable use and data handling.



Training and enablement so engineers can use tools confidently.



Visibility into spend and delivery outcomes such as cycle time, defect rates and rework.

Cultural change inside engineering teams

As AI accelerates throughput, the source of professional reward also changes.



"Once teams become effective at delivering like this, some engineers can feel they've lost part of the role they enjoyed... the shift is moving that satisfaction closer to product outcomes."

Ed Marshall, CTO
Enablis

Engineers increasingly derive satisfaction from **features released, user feedback and measurable customer impact**, rather than from writing code itself.

The organisations that get this right will not just move faster. They will carry a structural cost advantage that compounds over time. Under-investing in AI tooling is not the cautious choice. It is the expensive one.

Spotlight: enablis. x



Structured AI enablement in practice

The situation

Yoti is a digital identity verification platform operating in a highly regulated environment where security and trust are non-negotiable. Yoti wanted to explore how AI could support product innovation and operational efficiency without compromising regulatory and ethical standards. The challenge was not enthusiasm; it was clarity. Where should AI be applied first? And how could value be measured safely?

The approach

Enablis facilitated a focused, delivery-led AI workshop designed to:

- ✓ Identify friction points across the delivery lifecycle
- ✓ Separate viable opportunities from speculative ideas
- ✓ Align AI experimentation with measurable outcomes
- ✓ Define governance guardrails before scaling

The emphasis was not on tool demos, but on real operational constraints.

The outcome

60%

Yoti engineers embedded AI into their workflows **within 30 days**



Phase 2 target:

90%

adoption

That early momentum triggered Phase 2 of the programme, focused on deeper enablement, governance and scaling adoption across teams. The objective of Phase 2 is to increase adoption from 60% to 90% of the engineering organisation, embedding AI-assisted development into everyday engineering workflows.

Beyond adoption metrics, the engagement also delivered:

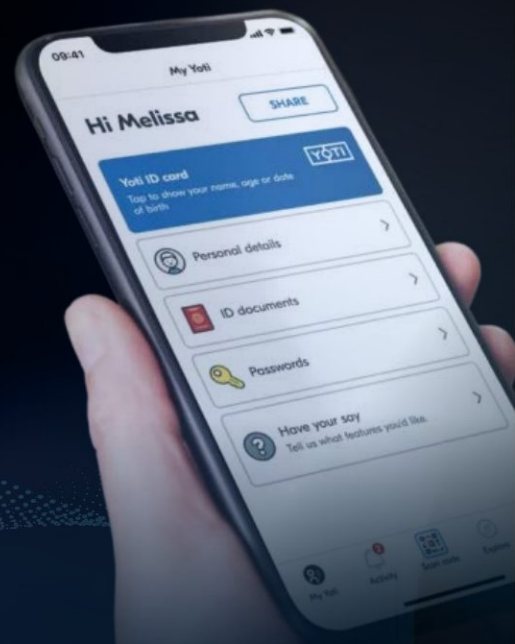
- A prioritised AI roadmap
- Defined governance boundaries for safe experimentation
- Clear next steps tied to delivery metrics

AI moved from experimentation to structured capability inside the engineering organisation.



“The shift wasn’t about adding more AI. It was about identifying where it would genuinely move the needle.”

Ed Marshall, CTO
Enablis



Spotlight: enablis. x nexus

Structured AI adoption beyond engineering

The situation

Nexus, the industry-leading vehicle and plant hire solution, recognised the potential of AI to improve product innovation and operational efficiency. However, like many organisations operating in highly controlled environments, the challenge was not ambition - it was clarity. AI exploration risked becoming fragmented without a clear understanding of where it would deliver value or how it could be applied safely within real delivery constraints.

- ➔ No clear prioritisation of use cases
- ➔ Uncertainty around safe adoption within regulated systems
- ➔ Risk of experimentation without measurable outcomes

The approach

Enablis supported Nexus through a structured **6-8 week programme** combining strategy, workshops and practical experimentation.

Cross-functional sessions brought teams together to examine:

- 📄 How work was delivered
- ⚡ Identify operational friction
- ✨ Prioritise where AI could add value.

Weekly workshops focused on mapping workflows and highlighting manual or time-intensive processes. The programme combined training with rapid experimentation, using internal hackathons to move quickly from idea to application.

Two hackathons built momentum: the first identified AI opportunities across the business, and the second selected a high-impact use case and developed it into a working prototype within a single day.

What it demonstrates

A working model for how AI capability is built across an organisation - within the departments where delivery friction actually exists, not just in engineering.

The programme starts with pain, not technology. Teams identify where work slows, effort is wasted and low-value activity consumes capacity. AI solutions emerge from these real constraints, not from a catalogue of tools.

Creating structure and visibility

- ✓ A prioritised backlog of AI opportunities
- ✓ Clear sequencing and ownership
- ✓ Leadership visibility and control





“What made the difference was having a team that could work across the whole business, not just with our engineers, but with finance, operations and contact centre. Enblis helped us move from scattered AI experiments to a structured programme with a real backlog of deliverable ideas. We went from conversation to deployed solution in a day. That’s the pace we needed.”



Pete Ennis,
CTO, Nexus

The outcome

AI opportunities were not treated as isolated ideas, but structured into a prioritised backlog aligned to real operational needs. This allowed Nexus to move quickly from exploration to delivery—taking a high-impact use case from backlog to working prototype in a single day.

// EXPLORATION



Identifying opportunities



Manual processes



Early experimentation

// PRIORITISATION



Structured backlog of AI use cases



Clear ownership and sequencing

// DELIVERY



Prototype built from backlog



Rapid validation in real workflows

// SCALE



Deployment into production



Continuous, organisation-wide capability

Over time, this approach established a repeatable model for identifying, testing and prioritising AI opportunities across the organisation.



How Enablis helps

enablis.

The organisations that will lead on AI delivery in the next two years are making foundational decisions right now.

Not about which tools to buy. Not about governance frameworks to adopt. About how AI is operationalised, where ownership sits, what measurable change looks like in 30 days, and what the delivery model looks like when AI is infrastructure rather than experiment.

That shift does not begin with a workshop or a pilot. It begins with an honest assessment of where delivery is actually slowing down, where quality erodes under pressure, and where your engineering capability can absorb AI adoption without creating new risk.

From that starting point, the path forward is usually one of three things: a Strategic Discovery session to map the opportunity and build a 30/60/90-day roadmap; a hands-on delivery engagement working directly on live backlog items and real codebases; or an Adoption Acceleration programme for teams already using AI but not yet seeing consistent value across the engineering organisation.

The right starting point depends on where you are. The wrong starting point is waiting.

If you want to understand what that looks like in your environment, **get in touch**.

About Enablis

Enablis is a tech consultancy specialising in strategic implementation and technology delivery for ambitious enterprises and SMEs.

We help teams turn AI transformation into real business impact through practical delivery, risk management, and enterprise-ready adoption.



Ed Marshall

Chief Technology Officer

Extensive financial services experience across Sainsbury's Bank, Tesco Bank, Tandem Bank, Experian and Starling Bank. Ed was the technology lead for Sainsbury's Bank's DevOps transformation, driving the cloud architecture and integrated toolset that connected stakeholder teams to technology delivery.

// CONTACT US

AI becomes valuable when it is built into delivery. That shift requires alignment across engineering, leadership and governance from the start.

If you would like to explore how this applies in your environment, we'd love to talk.

hello@enablis.co

1st Floor, 1 York Place, Leeds LS1 2DR

+44 (0)113 323 9900

LinkedIn: Enablis