CASE STUDY · THOUGHTFOX × OCUCO · AGENTIC SDLC

95% agent-written code. *140 engineers.* In production.

How Ocuco rebuilt its engineering model around agentic coding – and what it means for the organisations that follow.

€6.1M	8×	2.4×	20×
Annualised new engineering value · Year 1	Return on Claude Code consumption cost	Increase in pull request rate	Growth in AI-accepted lines over 6 months

THE SITUATION

A 140-strong team. A roadmap *ready to accelerate*.

Ocuco is one of the world's leading software companies in the optical and eyecare sector — a *140-strong engineering and product team* supporting thousands of clients across the UK & Ireland, Netherlands, Nordics, and North America. A mature, mission-critical codebase handling PII and PHI in regulated markets — with data governance at the centre of every architectural decision.

In late 2024, Ocuco made a deliberate decision to transform how software was built — not because the team was failing to deliver, but because the ambition of the product roadmap demanded a step-change in capacity. The organisation spent 12 months building AI fluency, establishing the foundation that would make the next step possible.

In Q4 2025, Ben Churchill — Ocuco's Head of AI, working in partnership with ThoughtFox — ran proofs of concept using Claude Code against the mature, complex codebase. They confirmed what the team had anticipated: the complexity was not a barrier. The right context infrastructure was the answer.

> *"We're now over 95% agent-written code across a 140-strong engineering team."*



IARHFLAITH KELLY
CTO, OCUCO

THE ROLLOUT

By April 2026 the transformation was complete — delivered **four weeks ahead of schedule**. 19 teams were restructured into agentic pods; GitHub Copilot was retired as the team moved fully to the agentic model, supported by an AI enablement team and regular Communities of Practice.

150+

Claude Code seats deployed org-wide across all engineering teams

23

Agentic pods — 19 teams restructured into pods of 2–3 developers

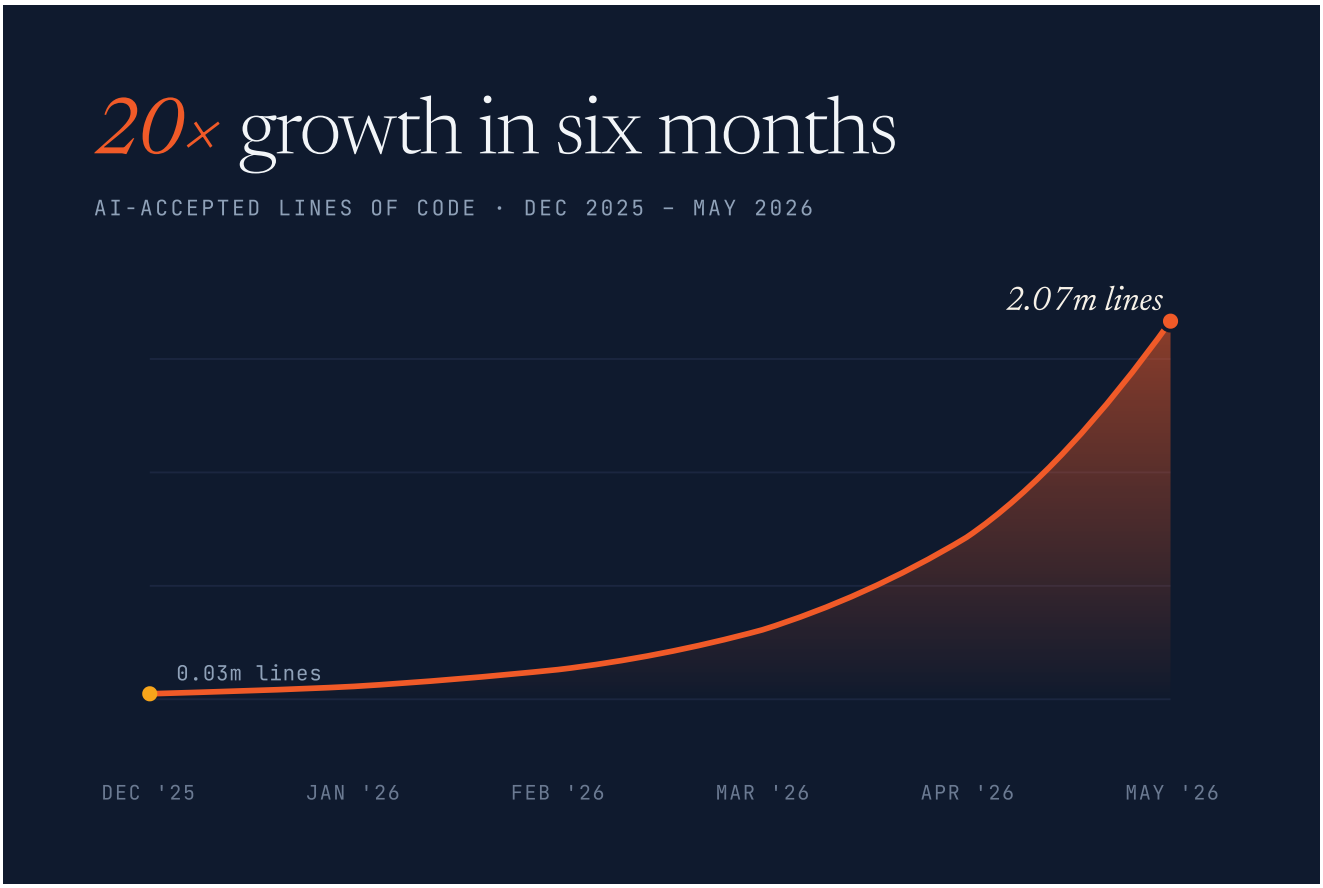
4 wks

Ahead of schedule — full rollout completed April 2026

THE SCALE OF CHANGE

In six months, the shape of the work *changed entirely*.

AI-accepted lines of code grew from **0.03 million in December 2025 to 2.07 million by May 2026** — a 20× increase as the agentic model moved from pilot to production across all 140 engineers.



Dec '25 and May '26 are measured. The path between follows the growth curve.

THE TRANSFORMATION

Three *structural changes* — not a tool rollout.

The step-change came from three structural changes that together created an entirely different operating model.

01

CONTEXT LAYER

Hive — the universal context layer

Ocuco built Hive: an internal MCP server aggregating 37 other MCP servers. It connects live Jira tickets with full comment history, vectorised archives of historic tickets, developer resource libraries, and documentation tooling. Engineers stopped copying context into prompts. Agents understood not just the code, but the ticket, the history, the institutional reasoning. PII and PHI are handled via redaction; Claude Enterprise & Teams plans ensured no data left the organisation. Beyond context, Hive runs active agents across the SDLC — a dedicated bug-triage agent classifies and routes every incoming bug automatically. Hive operates at a **92% success rate** across 14,400 calls per 30-day period.

02

SDLC REDESIGN

Rebuilding the SDLC from the ground up

The initial Tiger Team revealed the real insight: when AI writes the code, the bottleneck moves. Planning became the critical investment — rich tickets, Claude in planning mode producing complete specs before any code. QA shifted left to product planning. TDD finally became the path of least resistance. Code review became the primary quality gate.

TDD now the default

QA shifted upstream

Review primary quality gate

03

ENTERPRISE CONFIGURATION

Enterprise-grade workflows and skills

Claude Code out of the box is fundamentally different from Claude Code configured for enterprise software. Ocuco built 126 custom skills, slash commands and agent behaviours calibrated for the codebase — made available to every engineer. Engineers moved from writing code to directing and judging AI output.

126 skills

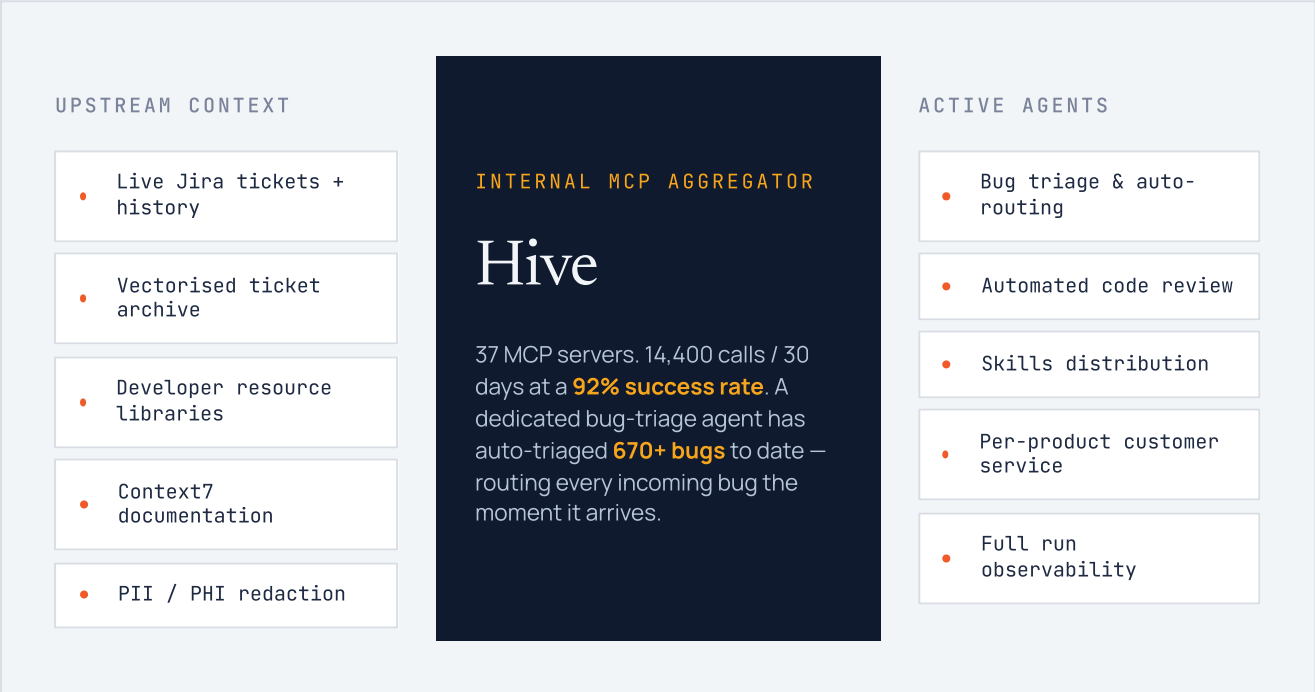
37 MCP servers · 92%

150+ engineers active

CONTEXT INFRASTRUCTURE

The answer to *"too complex for AI"* was never the AI.

Hive aggregates 37 MCP servers into one context layer — and runs active agents across the lifecycle. Engineers stopped manually assembling context; agents gained the ticket, the history, and the institutional reasoning, with full observability across every run.



THE INSIGHT

The right question is not *"can AI handle our codebase?"* — it is *"have we given AI enough context to navigate it?"*

INSIDE THE TIGER TEAM

Users of a system — or *architects* of one.

The Tiger Team was selected from what were expected to be the most AI-fluent engineers. The first lesson came quickly: assumed AI fluency and actual readiness to work in a fully agentic model are different things. The team needed a few days of structured training to calibrate — but once they did, adoption was fast and genuine.

What happened next was the signal that mattered. Engineers did not just use the new tools — they started building with them. Within the first phase, team members were writing their own skills, commands, and workflows and sharing them across the business. They had stopped being users of an AI system and started being architects of one.

The Tiger Team shipped everything they planned to. Then they ran a bug blitz.

THE PEOPLE REALITY

Seniority was a *poor predictor*. Character was not.

Senior engineers with deep expertise were often the most resistant — not because they feared being replaced, but because they genuinely underestimated what the technology could do. Their scepticism was professional, not personal. Once they saw the capability, many became the strongest advocates.

More junior engineers found it easier to embrace the new model, having less prior framework to unlearn. But seniority turned out to be a poor predictor either way. The real dividing line was character.

> *"The question isn't will my experienced engineers accept this? It's which of my engineers are genuinely adaptable? The answer often reveals more about your team than the transition itself."*

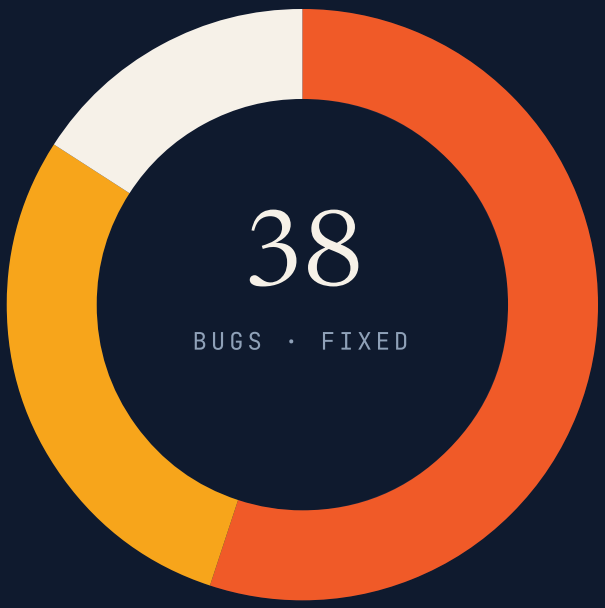


BEN CHURCHILL
HEAD OF AI, OCUCO / CO-FOUNDER, THOUGHTFOX

THE BUG BLITZ

38 unplanned bugs *fixed* on top of full sprint delivery.

The Tiger Team's first bug blitz — a proof of concept for what agentic bug resolution looks like at pace.



55% Fixed completely **automatically** — no human intervention required — FULL AUTONOMY

29% Fixed with **partial autonomy** — agent drives, engineer steers — PARTIAL

16% **Human supervised** — complex cases requiring deep engineer judgement — SUPERVISED

BEFORE

5 days

Average bug fix time across the Ocuco organisation, historically.

AFTER

20–30 min

Same categories of bugs, under the agentic SDLC.

That is not an improvement in efficiency.
It is a different category of engineering capability.

SCALED ORG-WIDE · APRIL 2026

A proof of concept became an *operating standard*.

When the methodology was rolled out across all 140 engineers, the results scaled with it. The April 2026 org-wide bug blitz closed *404 priority bugs* and produced a 50% decrease in skipped tests.

404	−50%	140
Priority bugs closed in the April 2026 org-wide bug blitz	Decrease in skipped tests across the organisation	Engineers running the same methodology, in parallel

More significant is what happened at the quarterly level. **Q2 2026 is the first quarter in Ocuco's history where bugs resolved exceeded bugs created.** That is not a one-off improvement in efficiency — it is a structural inflection point.

494 > 437

RESOLVED VS. CREATED · Q2 2026

For the first time, Ocuco is **reducing its total bug backlog** while simultaneously shipping more features than at any point in its history. The agentic model did not trade one for the other — it moved both in the right direction at once.

BREADTH OF EVIDENCE

Not an elite cohort. The *whole organisation*, moving.

The headline numbers capture magnitude. The breadth of the evidence is what makes them defensible. In Q2 2026, the organisation delivered *35.8% above commitment* — against a 1.6% baseline in the same period one quarter earlier (Q1 2026). A 34-point like-for-like delta.

17/20	+22%	850 SP
Teams overdelivered — the whole organisation, not a top cohort	Sustained delivery forecast above commitment across Q2 2026 (range +15–29%)	Additional quarterly capacity — ~37% QoQ story-point growth

LIKE-FOR-LIKE DELTA

The Q2 2026 result — 35.8% above commitment versus 1.6% in the identical sprint slot one quarter earlier — is a clean comparison. Same sprint slot, same teams, a 34 percentage-point improvement. The full-quarter forecast confirms this is a sustained shift, not a spike — and it holds across 17 of 20 teams, not a self-selecting top cohort.

Ocuco plans on a quarterly cadence (internally, Planning Intervals): Q3 2025, Q4 2025, Q1 2026, and Q2 2026 — the current quarter. All comparisons above are like-for-like across these quarters.

QUALITY ENGINEERING

Moving faster did not mean *accepting more risk*.

The common concern with agentic coding is quality regression — more code, shipped faster, with less human scrutiny. The Ocuco data shows the opposite. The team did not just move faster; they rebuilt the quality infrastructure while doing it.

26→85%	-70%	74 min
Test automation coverage (Q3 2025 → Q2 2026)	Reduction in manual regression effort over the year	Test pipeline runtime — down from 180 minutes

Manual regression is now run by **three testers across the entire organisation** — down from one to two testers per team historically. The defect detection percentage (DDP) for 2025 was **79%**. UAT tickets per deployment fell **49%** and regression bugs fell **36%**.

NAMED AGENTS IN ACTIVE USE

- INSPECTOR OAF
- OAF CODE REVIEW
- OAF TEST DEV

IN THEIR WORDS

"TestRail (a widely-used test case management platform) couldn't keep pace with how modern QE teams work. Building Quexa on Claude Code saved us over **€60,000 a year in licensing and €200,000 in manual effort** *– but the bigger shift was structural. AI is now a first-class participant at every stage: Claude reads the story, writes the test cases, builds the plan, and flags gaps before release. Not a bolt-on. Built in."*



NIRMAL YESUDAS
SR. QE MANAGER, OCUCO

ENGINEERING PERFORMANCE · DORA

Industry-standard measures. *Unambiguous* improvement.

DORA (DevOps Research and Assessment) metrics are the industry's standard framework for measuring software delivery performance. Across all three key indicators, Ocuco has moved materially.

METRIC	BEFORE	AFTER	CHANGE
Deployment frequency Deployments per quarter	42	51	+21% target 60
Mean time to recover MTTR after a failed deployment	29 days	8 days	-72% recovery time
Change failure rate Deployments requiring remediation	42%	32%	-10pts failure rate

29 → 8 days

MEAN TIME TO RECOVER

The MTTR improvement is the standout figure. A **72% reduction in recovery time** changes the risk calculus for deployments — teams ship more often because the cost of a failed deployment has fallen dramatically.

— THE RESULTS

Productivity increased by 50% almost immediately. As teams learned the nuances, that grew to 2–3×. The floor was extraordinary. The ceiling kept rising.



IARHFLAITH KELLY · **CTO, OCUCO**

ECONOMIC RETURN

€6.1M

Annualised new engineering value, Year 1.

COST RETURN

8×

Return on Claude Code cost, stabilising.

THROUGHPUT

2.4×

Increase in pull-request rate (Q1 → Q2 2026).

ROADMAP RECOVERY

The 850 extra story points of quarterly capacity were not abstract headroom. They absorbed a major *473 story-point market-fit programme* into the existing schedule — same headcount, no reduction in force. Two consecutive PIs have now recorded 100%+ sprint completion.

CUSTOMER IMPACT

Complex customer issues that previously took weeks of investigation are now resolved in hours. Automated bug resolution — 55% of bugs fixed with no human intervention — is now a routine operational capability. The €6.1M figure compounds: same headcount, same cost, more output every quarter.

FOR EVERY ENGINEERING LEADER

Four things *every leader* needs to understand.

01

Code completion is not transformation.

12 months of Copilot and AI assistance produced incremental gains. The step-change came only when the entire SDLC was rebuilt around the assumption that AI writes the code. These are not the same approach at different intensities — they are categorically different operating models.

02

The constraint shifts. You have to shift with it.

When AI takes over implementation, the critical work moves to planning upstream and review downstream. Organisations that do not redesign around this create a faster way to produce poorly-specified, under-reviewed software.

03

"Our codebase is too complex" is almost always wrong.

Ocuco's codebase complexity was real. The argument that it was too complex for AI was not — it was an argument about poor context infrastructure, not about AI capability. Hive was the answer. The right question is not "can AI handle our codebase?" It is "have we given AI enough context to navigate it?"

04

Out-of-the-box is not enterprise-grade.

Claude Code configured for enterprise software — purpose-built skills, workflows, governance design, data controls, and full run observability — is a fundamentally different capability from the default tool. The configuration and operating model matter as much as the underlying model.

WHAT 95% REALLY MEANS

"95% agent-written code" is not a metric about AI. It is a statement about *where engineering cognition goes*. Ocuco's engineers are no longer spending their expertise on implementation detail. They spend it on what code does, how it is tested, how it performs, and whether it meets the standard.

— WHAT OCUCO BUILT. WHAT IT PROVES.

What Ocuco built. *What it proves.*

Ocuco's transformation is one of the most complete agentic engineering programmes in the industry. *95% agent-written code across 140 engineers.* DORA metrics moving materially across every indicator. A quality engineering function rebuilt from the ground up. A roadmap back on track — with capacity to spare.

ThoughtFox partnered with Ocuco to design and deliver the programme. The methodology — Flow Engineering — is available to engineering organisations ready to make the same move.

Flow Engineering

A Virtual CAIO mandate that rebuilds the SDLC around agentic coding. **Hive-style context infrastructure. DORA-based measurement. Tiger-team activation.** Day-90 target: 95% agent-written code.

THOUGHTFOX.AI →

CONTACT

Ben Churchill

ben@thoughtfox.ai

ROLES

Head of AI, Ocuco
Co-founder, ThoughtFox
AI Transformation Consultancy

PROGRAMME

Tech Team AI Activation
thoughtfox.ai