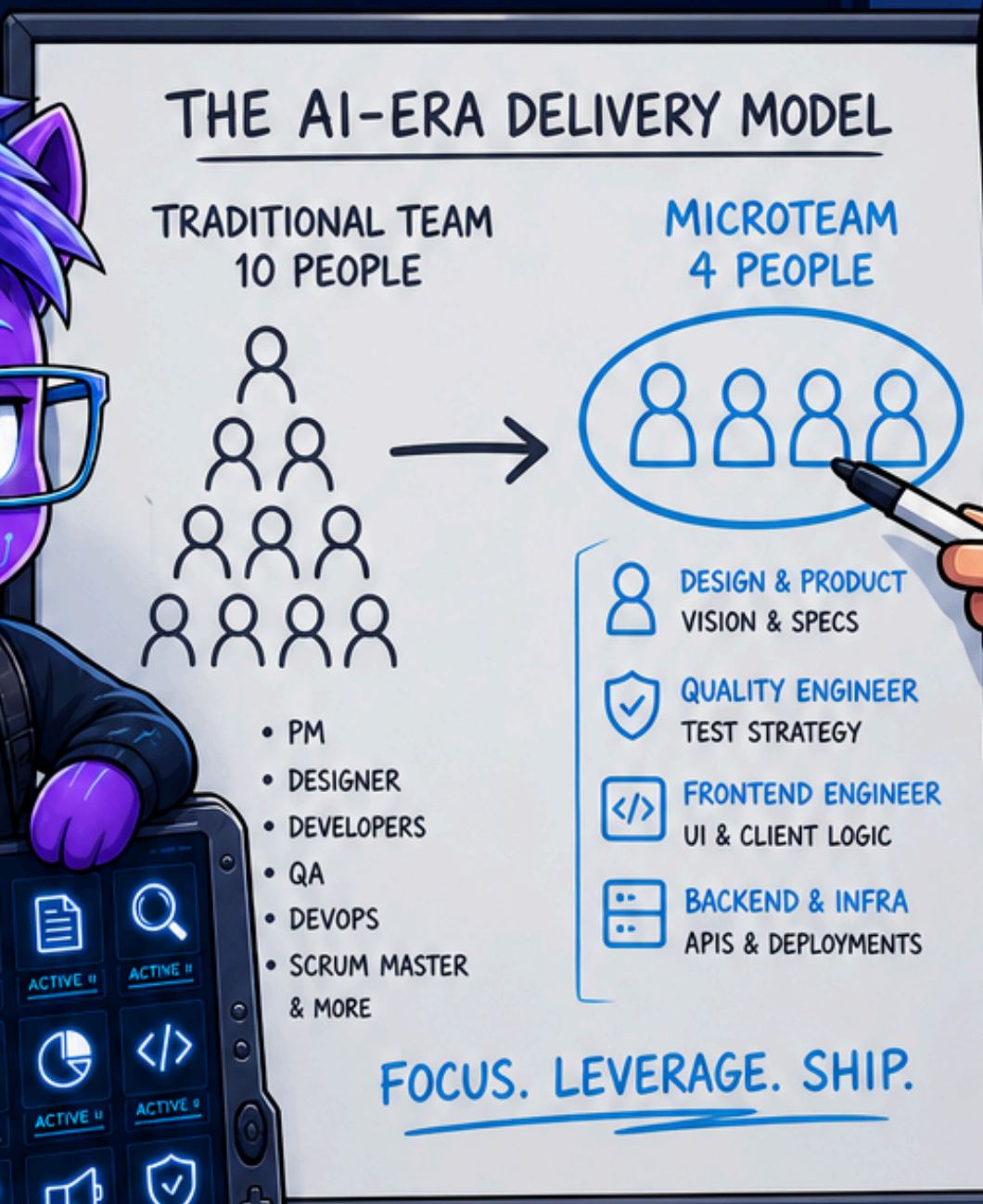
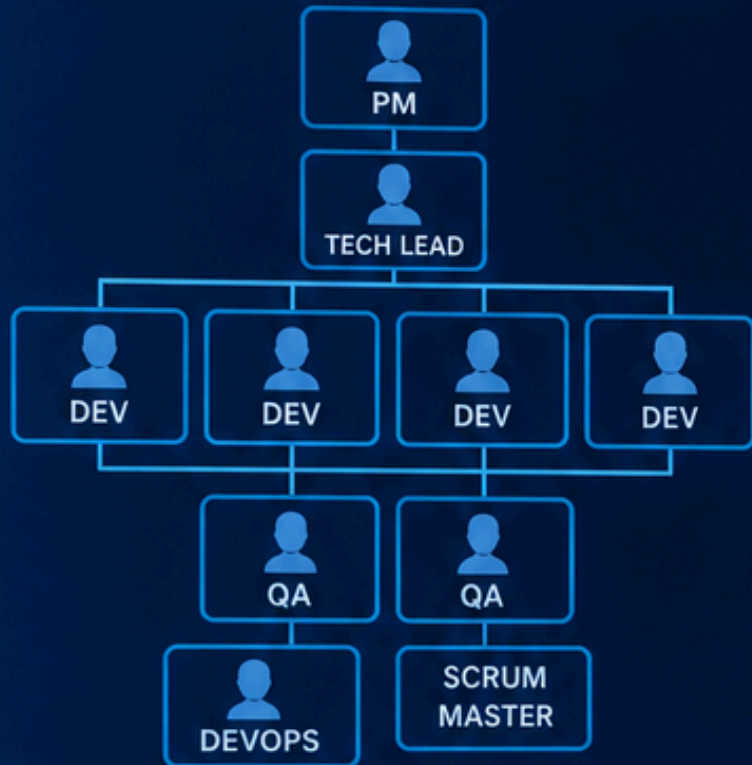


Microteams: The AI-Era Delivery Model That Actually Ships



Why the 10-person sprint team is starting to look like a legacy cost

THE OLD 10-PERSON TEAM



THE MICROTEAM (4-5 PEOPLE)



- ⚡ AI TOOLING EMBEDDED AT EVERY STAGE
- ⚙️ EXECUTION LAYER AUTOMATED
- 🎯 OUTPUT DRIVEN NOT HEADCOUNT DRIVEN



What's actually changed to make this viable

The honest answer is: the tools got genuinely capable, and the data is catching up.



AI tool adoption among developers has grown sharply in recent years. Surveys from GitHub, Stack Overflow, and JetBrains have consistently tracked rising usage, with AI-assisted code generation now accounting for a substantial and growing share of what developers ship — though precise figures vary by methodology and population surveyed. **That's not a marginal shift. That's a fundamental change** in how codebases are scaffolded, drafted, and extended.



But raw code generation isn't the story. The more important shift is at the system level. **A small team guides a coordinated system of AI agents that can deliver an entire application end to end** — from design to code to testing to integration — raising only the decisions that truly require human judgment. The result, in well-structured engagements, is significant leverage: a few senior practitioners delivering what once required a much larger department.



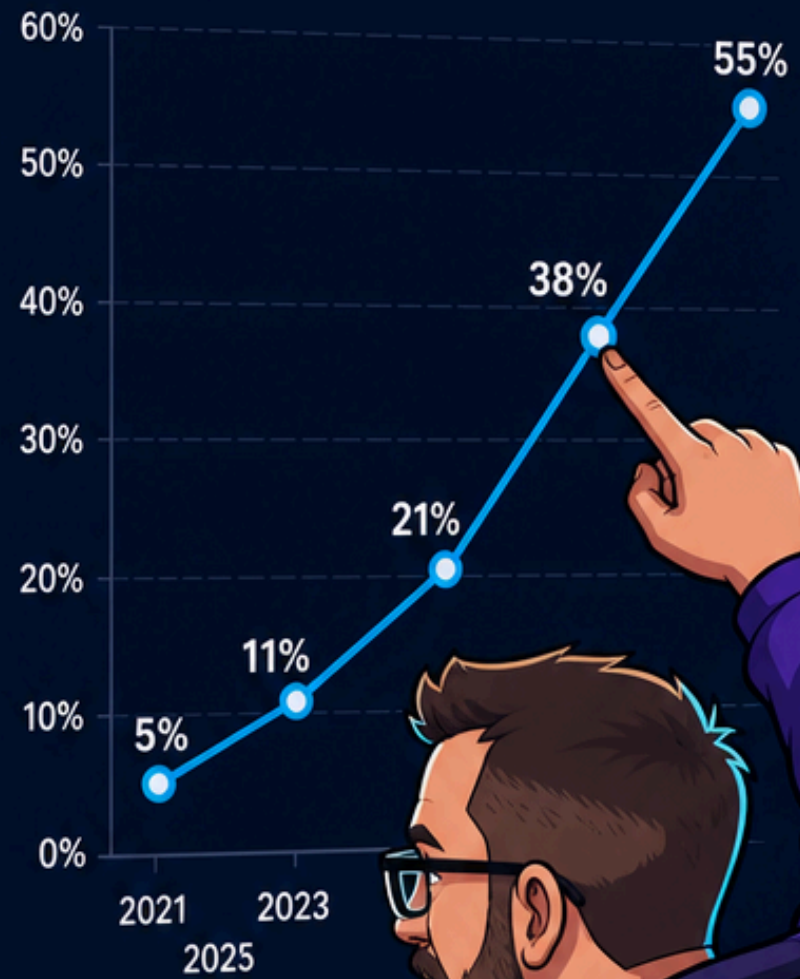
There's an important caveat here. The productivity gains don't show up automatically. **The success of AI in software engineering depends less on the sophistication of the tools and more on the strength of the organisational systems surrounding them.** Engineering culture, platform capabilities, development workflows, and internal knowledge systems ultimately determine whether AI improves productivity and delivery outcomes or simply accelerates complexity.



Put plainly: a badly structured microteam with AI tools is just a badly structured team that moves faster into the wall. **The structure has to be right first.**

AI tool adoption among developers

% of developers using AI-assisted code generation



Source: GitHub, Stack Overflow, JetBrains (2021–2025 surveys)



ROLE CLARITY IS THE NON-NEGOTIABLE FOUNDATION

DECISION-MAKER



- ✓ Final say on scope, architecture, and priority.
- ✓ Sets the boundaries AI and executors operate within.
- ✓ Owns the go/no-go calls.

EXECUTOR



- ✓ Writes specs, builds features, runs, deployments.
- ✓ Directs and coordinates multiple AI agents.
- ✓ Verifies output, catches errors, ensures architectural coherence.
- ✓ Operates at the cintersection of senior engineering judgment and AI tooling fluency.

AI-AUGMENTED REVIEWER



- ✓ Owns validation, not execution.
- ✓ AI generates test suites; humans define what to test.
- ✓ Evaluates whether results actually matter.
- ✓ Value is judgment about what passes and what doesn't.

THE HUMANS IN THE POD OWN THE DECISIONS.
THE AGENTS EXECUTE THE WORK THAT FEEDS
AND FOLLOWS THOSE DECISIONS.



THE CONCRETE EXAMPLE: A FOUR-PERSON DELIVERY TEAM

HERE'S WHAT THIS LOOKS LIKE IN PRACTICE.
NOT THEORY — A REAL TEAM STRUCTURE
EVOTRON STUDIO RUNS ON CLIENT ENGAGEMENTS.

THE TEAM HAS FOUR ROLES:



DESIGN AND PRODUCT

WRITES SPECS, OWNS THE USER JOURNEY,
DEFINES ACCEPTANCE CRITERIA,
HOLDS THE VISION



QUALITY ENGINEER

WRITES AUTOMATED TEST CASES
AGAINST THE SPEC BEFORE A LINE
OF FEATURE CODE IS WRITTEN



FRONTEND ENGINEER

IMPLEMENTS UI AND CLIENT-SIDE
LOGIC, CONSUMING THE SPEC AND
TEST SUITE AS THE SOURCE OF TRUTH



BACKEND AND INFRASTRUCTURE

OWNS DATA MODELS, APIS, DEPLOYMENTS,
AND THE CI/CD PIPELINE

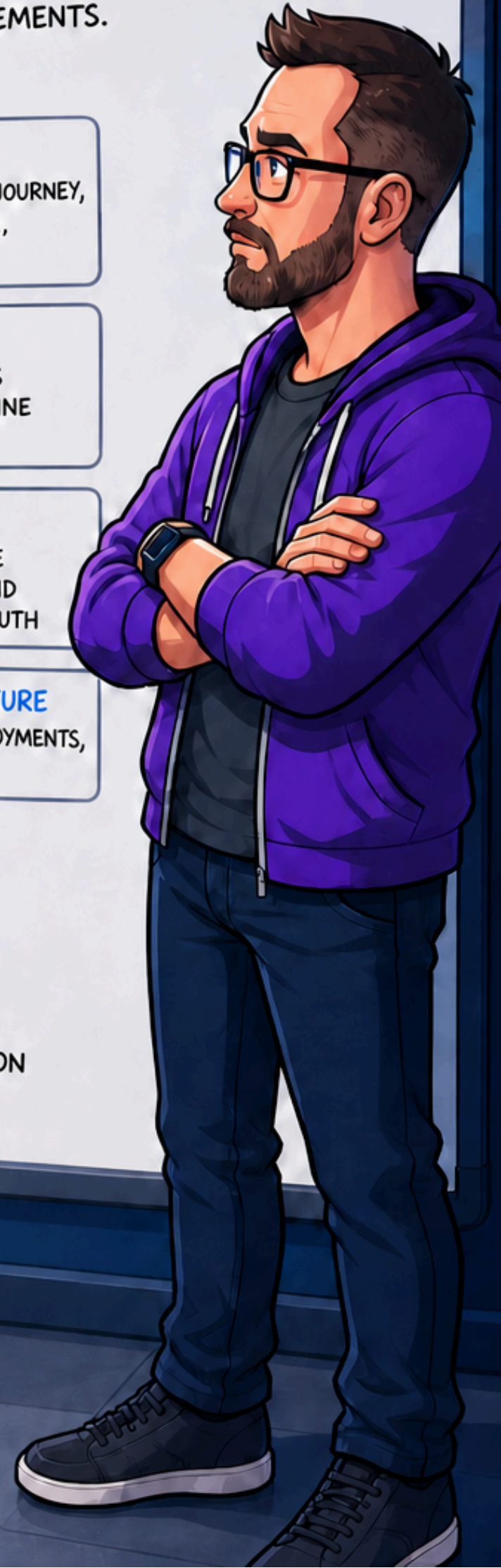
FOUR PEOPLE.

NO SCRUM MASTER.

NO PROJECT MANAGER.

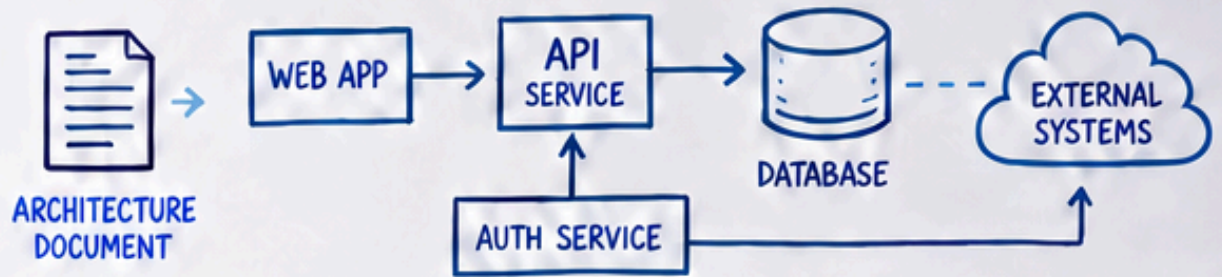
NO ACCOUNT DIRECTOR.

THIS WORKS BECAUSE THE PROCESS
IS ENGINEERED TO REMOVE COORDINATION
OVERHEAD, NOT MANAGE IT.



THE FIVE-STAGE PROCESS

1 ESTABLISH THE SOLUTION ARCHITECTURE



2 BOOTSTRAP WITH ESTABLISHED FRAMEWORKS AND SET UP CI/CD



3 SET UP AI CONTEXT GUARDRAILS AND KNOWLEDGE SHARING PROTOCOLS



4 ESTABLISH THE SPEC-TEST-BUILD-DEPLOY PIPELINE



5 AGILE TEAM PROCESSES FOR CONTINUOUS DELIVERY



Async-first isn't a culture preference, it's a structural requirement

Distributed microteams don't have the luxury of a shared office to absorb miscommunication. Every piece of context that isn't written down is a liability.

Documentation is the fuel for asynchronous communication. If it's not written down, decisions don't exist in a distributed team. <cite index="24-5,24-6">

This matters especially for AI tooling. When context lives in someone's head rather than a document, AI agents can't access it. The spec, the architecture decisions, the compliance constraints, the naming conventions — all of it needs to be in writing, versioned, and accessible to every member of the team and every tool the team uses.

Well-structured written handoffs remove the "waiting for the meeting to clarify" cycle that bleeds time in synchronous teams. That's not because async is inherently faster — it's because the discipline of writing things down forces the clarity that verbal communication often defers.

In a four-person team: the spec is the handoff from Design/Product to Quality Engineer and the engineers. The test suite is the handoff from QA to the build stage. The CI/CD pipeline is the handoff from engineering to deployment. Meetings don't carry the work. Documents do.



MEASURING WHAT ACTUALLY MATTERS

Metrics that tell you whether the team is shipping working software that solves **real problems**.



LEAD TIME TO VALUE

How long from validated spec to deployed feature?
Not story points. Actual time.

7.3 days



CHANGE FAILURE RATE

What percentage of deployments cause a production issue?
This is the **quality signal**.

2.1%



CODE DURABILITY

Proportion of code that remains substantially unmodified after 14 or 30 days.
Emerging quality signal.

81%



SPEC-TO-DEPLOY CYCLE

How long does a single feature take from written spec to deployed build?
The microteam's primary throughput measure.

5.4 days



Every KPI should tie to a business outcome.
If you can't explain how a metric informs decisions, stop tracking it.



GOVERNANCE DOESN'T DISAPPEAR IN A MICROTEAM, IT GETS PERSONAL



DECISION LOGGING



AI AUDIT TRAILS



CLEAR ESCALATION PATHS

ID	DECISION	CONTEXT	DECISION MAKER	DATE	AI vs HUMAN CONTRIBUTION
DEC-042	Switch vector DB to Pinecone	Performance + scalability	Marko	Oct 28	AI 40% HUMAN 60%
DEC-043	Expand scope: User analytics	Customer feedback	Evvo	Oct 29	AI 25% HUMAN 75%
DEC-044	Add compliance telemetry	Audit requirement	Marko	Oct 30	AI 70% HUMAN 30%
DEC-045	Adopt AI code review agent	Dev velocity + quality	Evvo	Oct 31	AI 60% HUMAN 40%



OWNERSHIP IS CLEAR.
DECISIONS ARE DOCUMENTED.
ACCOUNTABILITY IS BUILT IN.

TRACKING AI VS HUMAN CONTRIBUTIONS.

DECISION LOG, WITH REASONING.



THE HONEST TRADE-OFF

HIGHER AVERAGE SENIORITY

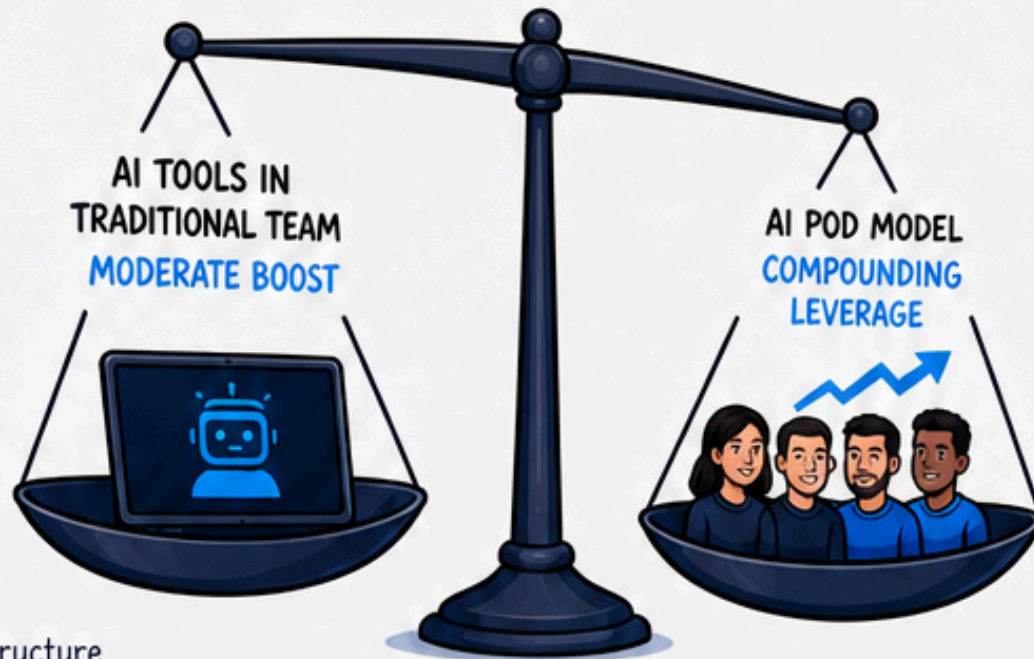
Nobody junior to absorb overflow.

DISCIPLINE IN DOCUMENTATION

Too small to absorb tribal knowledge.

CLEAR PROCESS

Four people without structure don't magically self-organise.



THE DIFFERENCE ISN'T THE TOOLS.
IT'S THE STRUCTURE.

If you're running corporate innovation programs, evaluating whether to build an internal capability or route work through an external partner, or trying to understand what "AI-augmented delivery" actually looks like in practice – the microteam model is worth understanding before you make that decision.

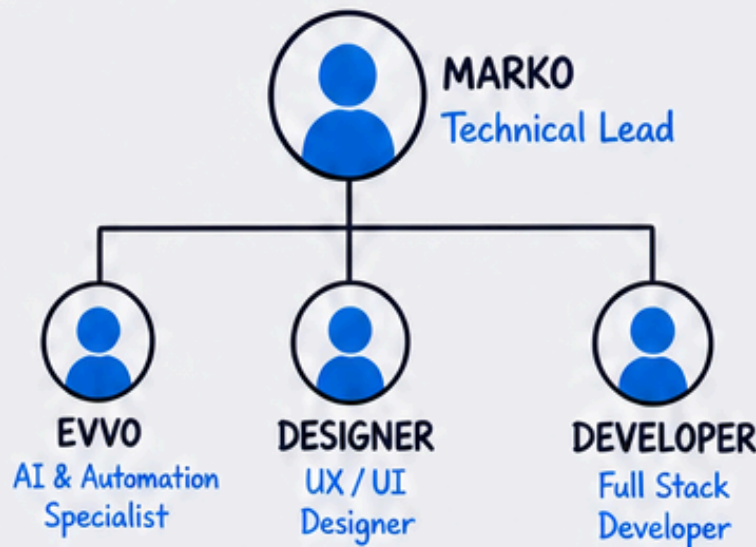
📄 1-22, 1-23

	TRADITIONAL TEAM	MICROTEAM (AI POD)	
SENIORITY	LOWER	HIGHER	✓
KNOWLEDGE CAPTURE	HARDER	REQUIRES DISCIPLINE	✓
PROCESS NEED	MODERATE	CRITICAL	✓
PRODUCTIVITY IMPACT	MODERATE BOOST	COMPOUNDING LEVERAGE	✓
OUTCOME	INCREMENTAL GAINS	SUSTAINED ADVANTAGE	✓



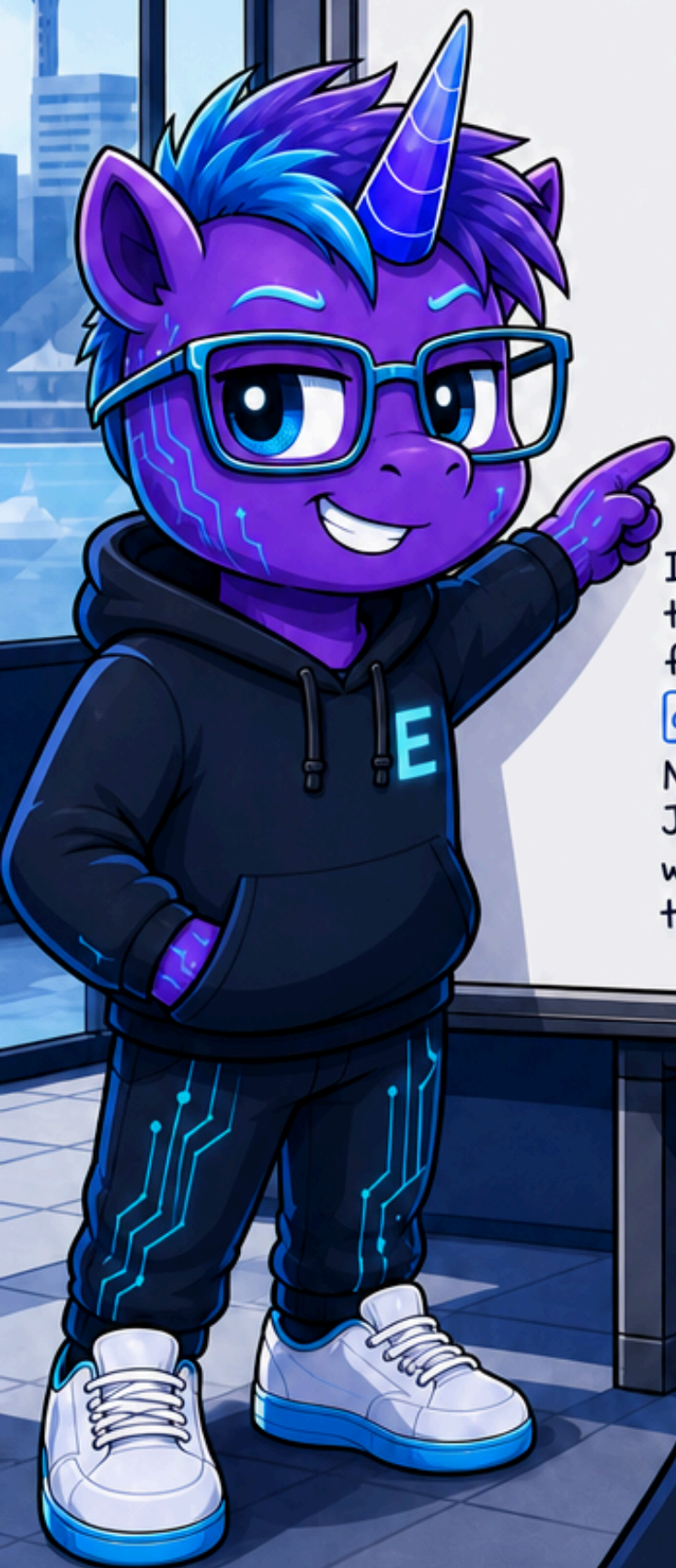
What this looks like at Evotron Studio

This isn't theoretical for us. The four-person team structure above is the model Evotron Studio runs on client engagements typically on bigger projects.



If you want to see the model in action, or talk through whether a microteam approach fits your innovation initiative, [Start the conversation at evotronstudio.co.nz](https://evotronstudio.co.nz).

No deck. No account manager.
Just a senior operator who can tell you whether this is right for what you're trying to build.





<https://evotronstudio.co.nz>