



FIELD GUIDE · 2026

The **AI-Augmented** Data Product Lifecycle

A field guide for data leaders moving from copilots to agents.

30-SECOND READ

Wave 1 is what most data teams call "AI adoption" today: developers using copilots, ~10% productivity uplift, no change to how work flows. Wave 2 - agents doing the work, humans approving at two checkpoints - is achievable on your current stack in 8-12 weeks and cuts developer effort per task by 75%. This guide shows the five waves, what Wave 2 looks like in practice, and a 90-day path to get there.

With a global manufacturing group we cut developer effort per high-complexity task from 8 MD to 0.5-2 MD, moving the team from Wave 1 to Wave 2.

Most teams are still in Wave 1 and don't know it.

Most AI programmes in data teams stall at Wave 1. They buy copilot licences, measure a 5 to 15 percent gain, call it adoption, and the way the team works stays exactly what it was. That is the floor of what is possible, and most teams stop there.

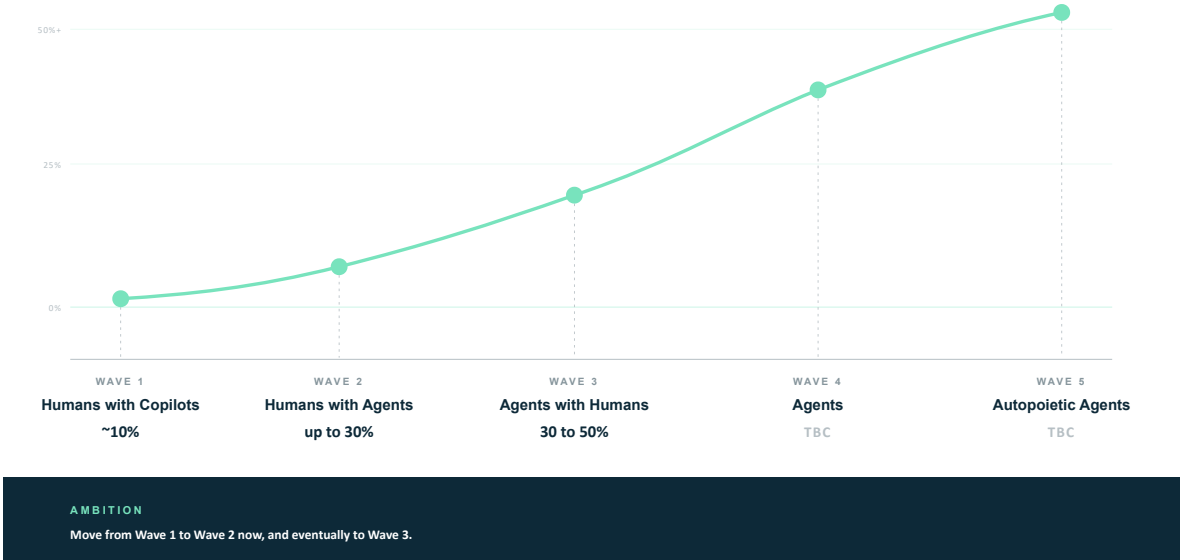
Wave 2 is the territory where agents do the work and humans approve. The components exist today: MCP-capable agents (Claude Code, Cursor agent, Cline), enterprise ticket systems (Azure DevOps, Jira, Linear), spec sources (SharePoint, Confluence, Notion). Most data teams already own all of them. The blocker is process design and approval discipline, not only technology.

Wave 3 is real but further out. The technology is in place; governance maturity is not, and that is the actual work between Wave 2 and Wave 3. Per-repo service principals, audit trails for agent actions, cost controls on autonomous loops: build them, and Wave 3 follows. The realistic horizon is 18 months.

Wave 2 stalls on approval discipline, not only on missing tools.

The Five Waves of data product development.

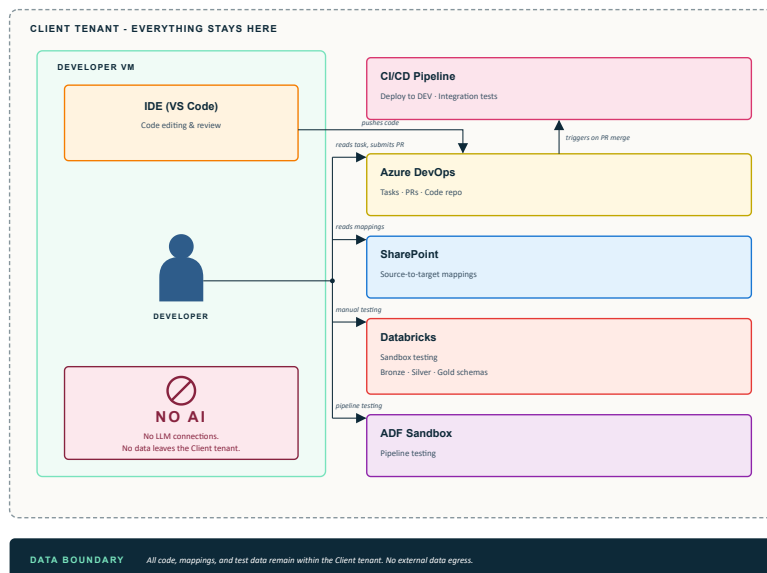
Five waves describe how AI moves through a data team. Wave 1 is where most organizations sit today: individual copilots, no change to the team or the lifecycle. Wave 5 is research. The work between Wave 2 and Wave 3 is where the next 12-18 months of real productivity gains happen, and that is where this guide spends most of its pages.



WAVE	PRODUCTIVITY UPLIFT	TEAM COMPOSITION CHANGE	LIFECYCLE CHANGE
1 - Humans with Copilots	~10%	None	None - same lifecycle, same handoffs
2 - Humans with Agents	up to 30%	None - same roles, new tools	Agents execute tasks; humans approve at two checkpoints per task
3 - Agents with Humans	30-50%	Roles shift toward review and governance	Agents own the loop; humans own the checkpoints
4 - Agents	Unverified	Smaller, more senior teams	No human in production execution
5 - Autopoietic Agents	Unverified	Unknown	Self-modifying agent ecosystems

Wave 4 and 5 are real research directions, but pricing your transformation against them today is fiction. This guide stays inside Wave 1 to Wave 3, which is the territory where both the numbers and the governance are real.

If your team works like this today, you are **with everyone else.**



End-to-end time per high-complexity task

11 MD

Wave 1 baseline

Effort split

Developer: 8 MD (~73%)

analyze, discuss, implement, sandbox testing, PR review

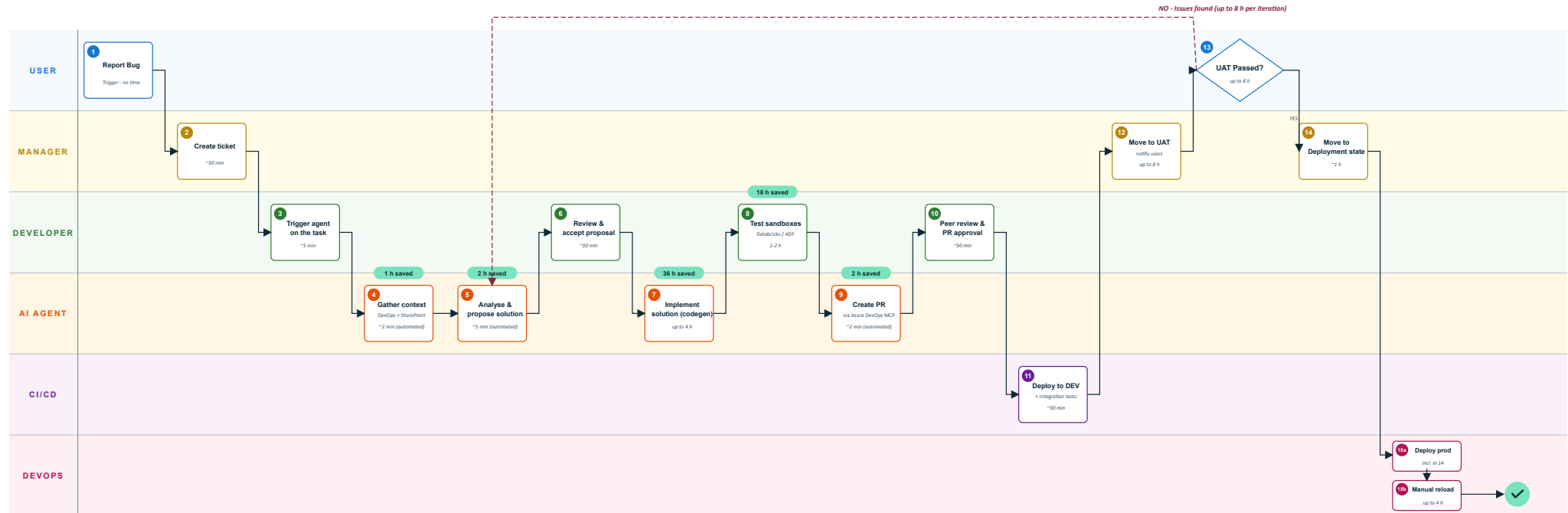
Joint-team validation: 3 MD (~27%)

UAT, deployment, manual data reload

Validation is the joint responsibility of project manager, key user, and the D&A team. It is unchanged across all waves.

Validation stays roughly constant across all waves. The developer's hands-on time is the lever, and Wave 2 pulls it first.

Wave 2: agents do the work, humans approve at two checkpoints.



Same backlog, same stack, same governance boundary. The developer's hands-on time compresses by ~75%.

What changes

Human in the loop on every decision

Developer triggers the agent, accepts the proposed solution, runs sandbox tests, and approves the PR. Two clear approval gates - before code is written and before merge.

Developer effort -75% per task

Sample task, high complexity. 0.5-2 MD effort vs. 8 MD baseline. End-to-end 3.5-5 MD per task, dominated by 3 MD of joint-team validation that does not change.

Agent works from the same artefacts

DevOps task and comments, linked epics, SharePoint mappings. The agent reads them all before proposing a solution. Context is built from the team's existing sources of truth.

Two clear developer checkpoints

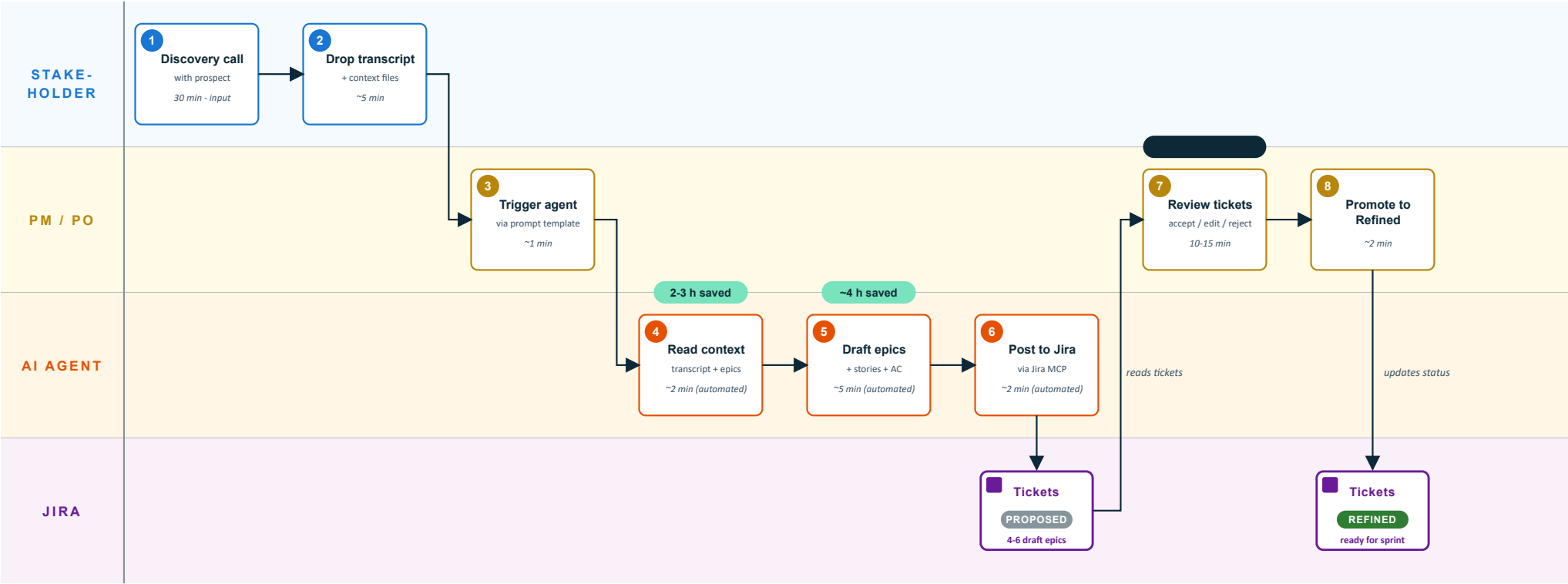
Developer accepts the proposed solution before any code is written, then approves the PR before merge. Between those two gates, the agent does the work and the developer reviews artefacts.

WHAT YOU NEED TO BE IN WAVE 2

- ☐ MCP-capable agent (Claude Code, Cursor agent, Cline)
- ☐ Read access for the agent to your ticket system (Azure DevOps, Jira, Linear)
- ☐ Read access to specs (SharePoint, Confluence, Notion)
- ☐ Sandbox environment per developer (existing CI/CD usually suffices)
- ☐ Per-task human approval gate before any code is written
- ☐ Per-PR human approval gate before merge
- ☐ An honest baseline measurement: how many MD does one high-complexity task take today?

From a 30-minute discovery call to a refined Jira backlog.

Most teams encounter the Wave 2 pattern first in their data engineering workflow. That makes sense. But the pattern is broader. Anywhere a team turns unstructured input into structured work items, the same loop applies: agent reads context, proposes work, human approves, agent executes, human approves output. The work has always been done. The question is who does the typing.



TOTAL Under 1 hour from call to refined backlog · Baseline: 4-6 hours of PM work · PM time repurposed from drafting to review

Worked example: 30-minute discovery call → refined Jira backlog

A Project Manager / Data Product Owner runs a discovery call with the Sales & Operations Director. The company recently acquired a regional retail chain, and the director wants two outcomes from the next data sprint: a redefined "active customer" KPI that consolidates the legacy customer base with the newly acquired stores, and the acquired chain's POS system added as a data source in the gold layer. Thirty minutes of conversation, with side-discussion about which margin reports break under the new KPI definition. In the Wave 1 setup, the PM blocks half a day after the call to draft tickets in Jira, link them to the right epic, tag the right team, and translate the director's asks into INVEST-compliant stories. In the Wave 2 setup, the agent does the drafting and the PM does the judgment.

Input the agent reads

- The call transcript (~6,000 words, exported from Gong, Granola, or Zoom AI summaries)
- The Jira project conventions doc: epic structure, label taxonomy, definition of ready
- Three reference epics from the same project, so the agent learns the team's writing style
- The current product roadmap one-pager
- The data dictionary for the legacy "active customer" KPI and the acquired chain's POS schema

Output the agent produces

- 4-6 draft epics in Jira. Concrete examples from this scenario: *"Redefine active-customer KPI for combined entity"*, *"Ingest acquired chain POS data into bronze layer"*, *"Update margin reports to reflect new KPI definition"*. Each carries a description, a business value paragraph, 3-5 child stories, INVEST-compliant acceptance criteria, suggested labels, suggested priority, and a link back to the relevant transcript timestamp.
- A "rejected ideas" comment per epic. Things the agent considered and explicitly did not include, with the reason. This is the part that speeds up PM review, because the agent's reasoning is visible.

What the PM does

- Reads each draft epic in Jira (10-15 minutes total)
- Accepts roughly 70% as-is, edits roughly 20% (usually scope or priority), rejects roughly 10%
- Promotes accepted tickets from "Proposed" to "Refined"

CYCLE TIME

< 1 hour

vs. 4-6 hours baseline, often spread over 2-3 days

PM TIME REPURPOSED

60-85%

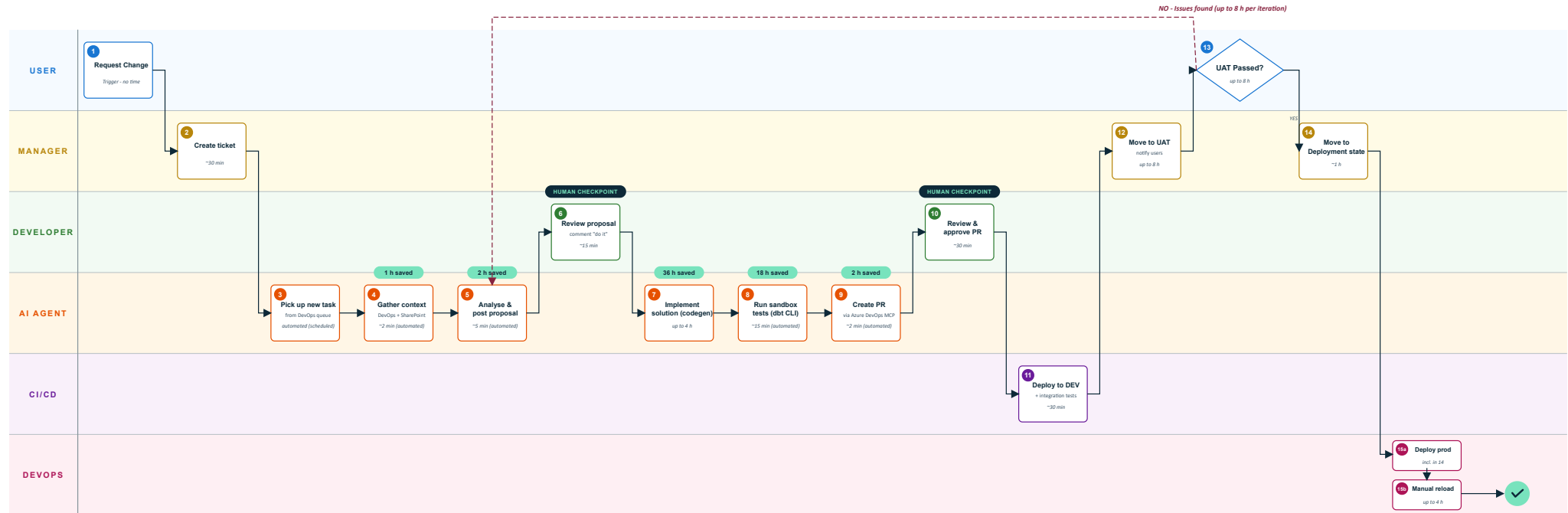
of discovery-to-backlog cycle moves from drafting to review

HONEST CAVEAT

The agent does not remove the need for a good PM. It removes the typing. PM judgment about what to accept, what to reject, and what to escalate to the product lead is the work that matters and the work that stays. A team that pushes the agent's draft straight into Jira without review produces a worse backlog faster. That is Wave 1 with extra steps.

Numbers extrapolated from comparable engagements.

Wave 3: agent owns the loop, human owns the checkpoints.



WAVE 3 RESULT

Developer effort drops to ~0.5 MD per task (-94% vs Wave 1 baseline of 8 MD) · Only 2 human checkpoints: proposal review & PR approval

What changes in Wave 3

Agent owns the loop, human owns the checkpoints

Agent polls Azure DevOps on a schedule (e.g. hourly), proposes a solution as a task comment, implements on "do it" approval, opens a draft PR.

Developer effort ~0.5 MD per task

~94% vs. Wave 1 baseline. Two human checkpoints only: comment approval (~15 min) and PR review (~30 min). End-to-end task time roughly halves.

New control: per-repo service principal

Agent runs under its own identity, with scoped permissions per repo. No destructive rights, no PR merge rights, no production deploy rights. Every action is auditable.

Every step lives in DevOps

The agent's proposed solution, test evidence, and code changes are posted as DevOps comments and pull requests. Audit, compliance, and review use existing tools. No parallel system.

Most organizations are 12-18 months from Wave 3. The work between Wave 2 and Wave 3 is governance maturity: per-repo service principals, agent action audit trails, cost controls on autonomous loops. The technology is already in place. This work cannot be shortcut.

"Why not just use our existing tool's natural-language feature?"

Three tools come up in every conversation: Databricks Genie, Snowflake Cortex, and Microsoft Copilot for Fabric. They are already in most enterprise stacks. The question is fair: why bring in another agent when these are already paid for? The answer is that they do a different job.

	DATABRICKS GENIE	SNOWFLAKE CORTEX	COPILOT FOR FABRIC
What it does well	NL analytics on governed data inside a Databricks workspace	NL queries and ML primitives directly inside Snowflake	Embedded NL across Fabric workloads (Power BI, Lakehouse, Notebooks)
What it does NOT do	Take a DevOps ticket, read SharePoint mappings, generate dbt code, run sandbox tests, open a draft PR	Same gap. Cortex sits in the analyst layer; the dev workflow lives elsewhere	Same gap. Copilot for Fabric is the embedded analyst assistant; the dev workflow lives elsewhere
Where it sits	Inside the data platform	Inside the data platform	Inside the data platform
Where the dev agent sits	Across the full toolchain (tickets, specs, code, tests, PR)	Same	Same

NL-on-data and agentic data product development are different layers of the same stack. They are complementary. The framework we describe here lives above the NL tools, in the layer where dbt code, dbt tests, and merged PRs are produced.

Where is your team on the wave model?

Mark one cell per row. The diagonal pattern shows where your team is today and where the realistic next move sits. Most teams find themselves spread across Waves 1 and 2, which is the expected state at this point in the technology cycle.

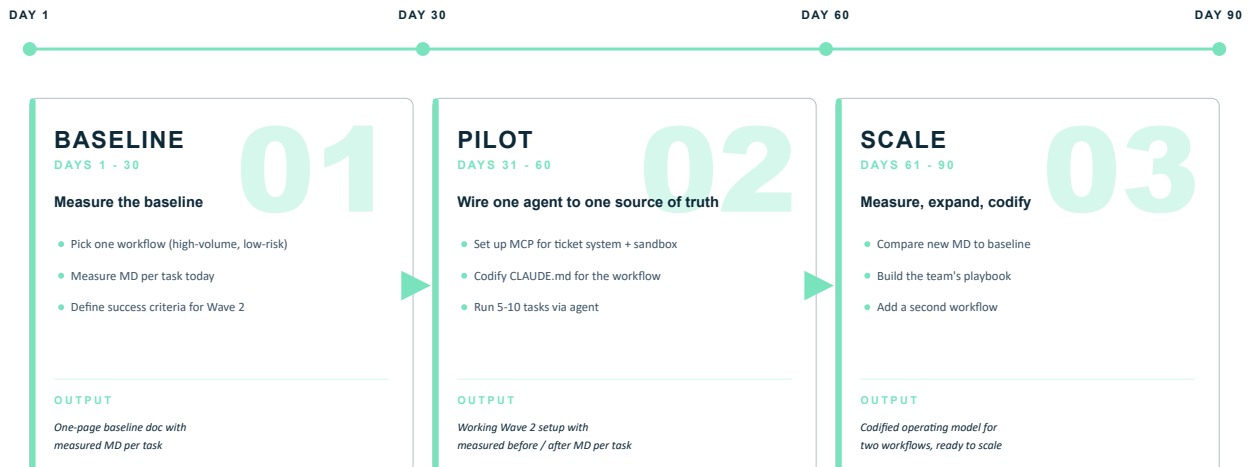
DIMENSION	WAVE 1	WAVE 2	WAVE 3	WAVE 4	WAVE 5
Tooling	Copilots in IDE only	MCP-capable agent connected to tickets, specs, code	Agent runs under service principal, scheduled polling	Multiple specialised agents, autonomous orchestration	Self-modifying agent ecosystem
Process	Ad-hoc AI use, no defined approval gates	Two clear checkpoints per task (proposal, PR)	Two lighter checkpoints, agent owns the loop between	No human in production execution	Process itself is agent-defined
Governance	Informal, "developers use AI on their judgment"	Per-task human approval, agent has read-only access	Per-repo service principal, scoped permissions, audit trail	Continuous monitoring, anomaly-triggered human review	Governance-by-agent
Team skills	Prompt users (intuitive AI use)	Prompt engineers + context curators (CLAUDE.md, KB)	Context architects + agent governance specialists	Agent operators + scenario designers	TBD

NEXT STEP

If you want a 30-minute conversation about where your team sits on this grid and what Wave 2 would look like on your stack, the booking link is on the last page. No pitch deck. We will look at the grid together and sketch your 90-day move.

The 90-day path from Wave 1 to Wave 2.

Most Wave 1 to Wave 2 attempts stall because the team tries to transform too much at once. The pattern that works is bounded: one workflow, one ticket system, one developer team. Measure the baseline before you change anything. The 90 days below assume that scope, and they assume you start on Monday.



TRAPS TO WATCH FOR

- **Context discipline.** Without a curated CLAUDE.md and knowledge base, the agent produces plausible-looking work that does not match your conventions. Treat context as the product.
- **Approval fatigue.** Developers stop reading the agent's proposals carefully after the first week. Build in spot-check audits.
- **Governance ambiguity.** "Who owns this?" is the question that stalls more Wave 2 rollouts than any technical issue. Decide before you start.

Astral Forest.

Strategy and execution from the same team.

Michał Dębski

Michał is Co-founder, Managing Director and Solution Architect at Astral Forest, a boutique data and AI consultancy in Warsaw. He has spent 15 years architecting data platforms for global manufacturers, retailers, and financial services firms. He works with CDOs and Heads of Data who want a peer in the room, someone who builds, not only presents.

Astral Forest

The team CDOs hire when they want strategy and execution from the same group. We architect, we build, and we hand the keys over. Recent work includes a Wave 2 transformation for a global manufacturing group (cited in this guide) and a SOX-compliant finance data warehouse for a global professional services group, with the first release in 90 days.

Three ways to engage

30-MIN CALL

Where is your team on the wave model?

Honest read on your current state and a sketch of the first move.

Free

[Book a 30-min call →](#)

1-DAY WORKSHOP

Data AI in a Day

Hands-on workshop on your stack. Real artefacts shipped by 16:30.

From EUR 9,000 (up to 12 people)

astralforest.com/data-ai-in-a-day →

90-DAY SPRINT

Wave 1 → Wave 2 in 90 days

Full Wave 2 setup for one workflow. Baseline measured, playbook codified.

Per-engagement

[Email Michał →](#)