Ways of Working: Executive Summary

The operating model on one page: who owns what, how much process each piece of work earns, and what gets measured.

Ask a product and engineering team who owns production at 2am and you will usually hear three names, or none. This operating model makes that question boring, and here it is on one page: eight principles you can test against your own organisation this week. The structure flexes with your size and the principles hold.

EVERY RESPONSIBILITY HAS ONE NAME AGAINST IT

Product owns the problem and the order of the queue. Engineering owns the solution and the system. Two different people hold them on purpose, because their daily argument is where someone challenges the scope and someone else prices the ambition. Quality, production out of hours and each person's growth have named owners too, and anyone can tell you who. "The team" owns nothing, and neither does "whoever picks it up".

WRITE THE DECISION DOWN WHEN YOU TAKE IT

The problem comes before the solution, the plan before the code, a decision record before anything someone will live with later, and a runbook before anything that can break at 2am. The tracker shows what is in flight, blocked and shipped, so status costs nobody a meeting. If a new joiner has to ask how to deploy, the documentation has failed them.

PROCESS HAS TO EARN ITS PLACE

Work gets a tier before it starts: feature, enhancement or initiative, each with its own discovery, artefacts and decision authority, so a copy change never meets the machinery of a new product line. An engineer accepts every change; where the risk is real, a second person does too, for a written list that includes authentication, payments, personal data, migrations, shared interfaces and anything hard to undo. Only migrations and security changes get their code read.

DEFAULTS BEND; SILENT DRIFT BREAKS

Each process here exists because going without it costs more than following it, and by that test daily standups, story points and review before every merge are absent. A named person can override any default in writing, and the team reviews the override afterwards. Exceptions are healthy. Silent drift is the failure that matters.

SHIP SMALL, AND SHIP OFTEN

Engineers commit small changes straight to the main codebase, switched off behind feature flags until ready, with the automated test pipeline as the gate and production under thirty minutes away. Large work ships in increments, each with a place to stop. Teams of five and up plan in two-week cycles; teams of one to four pull work continuously, with a hard limit on work in progress. The discipline is identical, and only the ceremony changes.

MEASURE THE SYSTEM

Four numbers describe delivery: how often the team deploys, how long a change takes to reach production, how often a change causes a failure, and how fast the team recovers. Next to them sit planned against unplanned work, use of the ring-fenced engineering allocation, and each initiative's outcome against the case its sponsor made. Individual velocity, hours and story points stay off the dashboard: anyone can game them, and they tell you nothing about outcomes.

AI RAISES THE FLOOR; PEOPLE ADD JUDGEMENT

AI drafts problem statements, acceptance criteria, tests, documentation and code, and reviews every change in the pipeline. It accepts nothing, and it acts on its own only where a person set the rule in advance, such as reverting a flag at a predefined threshold. A person owns the outcome of everything it touches, and each use case has a named owner, inputs and outputs. Treat it as a tireless junior with an excellent memory. It is still a junior.

SHRINK THE TEAM, KEEP THE RESPONSIBILITIES

The model is drawn in full for 20 to 45 people. It compresses to a founding team without losing anything that matters, and grouping squads into product areas takes it to around 100. At eight, every responsibility still exists, held by fewer and more senior people, and everyone can still name who holds each one. Roles split back out only when the work triggers it.

Full detail: Team Structure , Product Process , Engineering Process , DevOps Principles , the Founding Team Operating Model for teams of one to four, and the AI Use Case Catalogue .