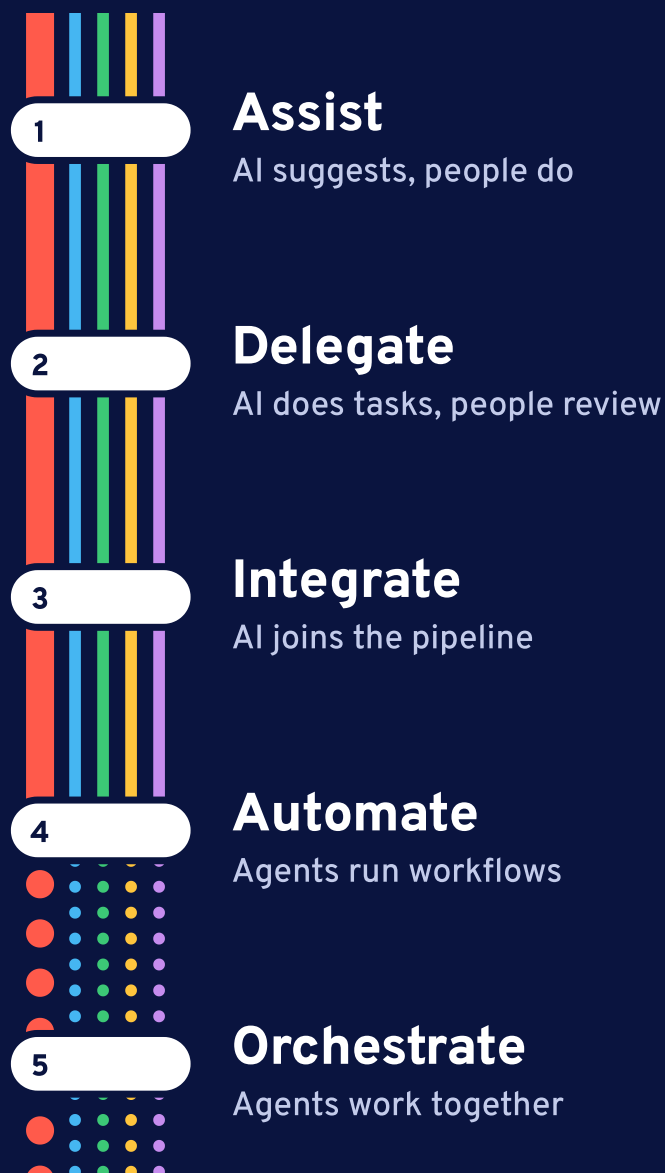


The Autonomy Line

How a large enterprise travels from AI coding tools to agentic AI, in five stations. About 111 minutes, with a worked example at every station, four interactive tools, and a route card to put in front of your leadership team.

Read it online, with the tools working: agenticaivarasrinivas.com/courses/the-autonomy-line



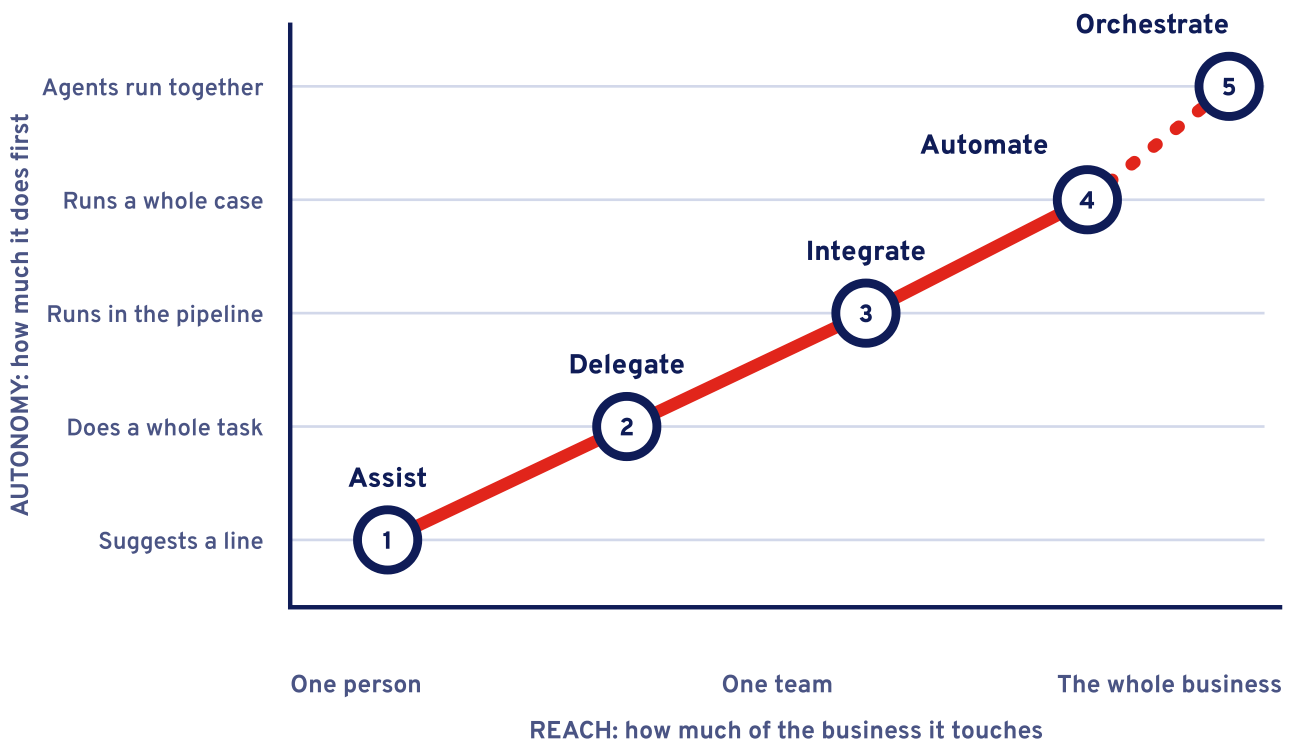
Before you board, about 6 minutes

Start here: how to read the map

Most enterprises bought AI coding tools first, and are now being asked about agents. These are not two projects. They are one line, and your organisation is already on it.

The idea in one minute

A coding assistant that suggests the next line and a business agent that processes a case are built from the same five parts. What changes along the line is **autonomy**, how much the AI does before a person looks, and **reach**, how much of the business it touches.



The two dimensions that define the line. They rise together, which is why the stations sit on a diagonal and why you cannot buy reach without earning autonomy first. The last leg is dotted because the industry has not finished building it.

You earn each station. Autonomy is granted in proportion to the controls and the evidence you have, never in proportion to the excitement. That is the whole discipline of enterprise AI adoption, and the rest of this guide is detail.

Four tracks run under the entire line:  **Platform**,  **Guardrails**,  **People** and  **Value**. The train can only go as far as the shortest track. Most stalled programmes have a train at Station 3 and a track that ends at Station 1.

The line at a glance

Station	In one line	People act as	What it looks like at Alderline
1 Assist	AI suggests. People do the work.	Authors	Engineers use an IDE assistant to explain legacy code and write tests.
2 Delegate	People hand over whole tasks and review the result.	Directors and reviewers	Coding agents upgrade 40 services to Java 21 and open pull requests.
3 Integrate	AI becomes part of the team's pipeline, on a shared platform.	System designers	AI review, test generation and incident triage run in every pipeline, through one gateway.
4 Automate	Agents run bounded business workflows. People handle exceptions.	Supervisors	An intake agent assembles prior authorization cases and fast-tracks the clear approvals.
5 Orchestrate	Many agents coordinate across functions. People govern outcomes.	Governors	Intake, scheduling, benefits and care-navigation agents hand work to each other under one control plane.

How to use this guide

- **Nine stops, about 111 minutes.** Every station has the same layout so you can skim: what it is, what it looks like in practice, what to build on each track, the gaps to mind, and how you know you can move on.
- **One company runs through every example.** Alderline Health is a fictional composite: 14,000 people, 1,600 engineers, and a prior authorization platform at the centre of its operations. Its numbers illustrate the pattern. They are not benchmarks.
- **Each stop ends with one checkpoint question.** Answer it and the stop is marked on your line.
- **Presenting it?** The left and right arrow keys move between stops, and the night map is easier on a projector in a dark room.
- **Short of time?** Go straight to the [one-page route card](#), or [find your station](#) with the six-question assessment.



The approach, about 10 minutes

Why the line starts with code

Software engineering is where an enterprise learns to work with AI, because it is the one place where the work checks itself.

Four reasons to begin in engineering

- **The work can be verified.** Code compiles or it does not. Tests pass or fail. No other kind of knowledge work gives AI such fast, objective feedback, which is why AI improved here first.
- **The people are ready.** Engineers adopt tools quickly and tell you bluntly what works.
- **Mistakes are contained.** Version control, code review and rollback already exist. A bad suggestion is caught before it ships.
- **The lessons transfer.** Everything your engineers learn about briefing, checking and limiting a coding agent is what the rest of the business will need for its agents two stations later.

Every agent has the same five parts

Strip away the product names and an agent is five things. Leaders who can name them can ask good questions of any vendor or team.

5. Guardrails what it may not do, and who gets asked first

2. Context

What it knows about your world: instructions, code, policies, documents, the case in front of it.



1. Model

The reasoning engine. Rented, swappable, better every few months.

4. Loop: plan, act, check, repeat



3. Tools

What it can touch: read a file, run a test, query a system, send a message.

A chatbot has the first two parts. A coding assistant adds a few tools. An agent has all five, and the loop is what makes it an agent: it acts, looks at the result, and tries again without being asked.

Everyone can rent the same model. **Your advantage is your context, your tools and your guardrails**, and those are built station by station.

The autonomy dial

The five stations are five settings of one dial. Turn it and watch what the AI does, what people do, and which control has to exist before you are allowed to turn it further.

Turn the dial

Printed at Station 1. Online, the dial turns through all five; on paper, the table in Start here sets them side by side.

1	2	3	4	5
---	---	---	---	---

Station 1: Assist

The AI

Suggests the next line, explains, drafts. One step at a time.

People

Do the work. Accept or reject every suggestion. Own the result.

In engineering

An assistant in the editor explains a legacy class and drafts its tests.

In the business

A nurse asks a policy assistant where the criteria for knee surgery are written.

The control you need before turning further

An approved tool, a one-page use policy and a baseline. Until those exist, the dial stays here.

What the evidence says

Five findings every leadership team should hold in mind at once.

90%

of technology professionals now use AI at work, and more than 80% say it makes them more productive. Yet 30% trust its output only a little or not at all. The headline finding: AI amplifies what is already there. Strong teams get stronger, struggling teams get exposed.

DORA, State of AI-assisted Software Development, 2025

19% slower

is what a 2025 randomized trial measured for experienced developers using AI, while those same developers believed they were 20% faster. The 2026 follow-up points to real speedups with newer tools, with wide error bars. The lasting lesson: felt speed is not measured speed.

METR, 2025 study and February 2026 update

80 / 37 / 6

Eight in ten people say AI improved their own productivity. Only 37% of organisations see any effect on profit, unchanged in a year. About 6% see a large one, and nearly three-quarters of that group have fundamentally redesigned their workflows.

McKinsey, The State of AI, August 2026

40%+

of agentic AI projects are predicted to be cancelled by the end of 2027, because of rising cost, unclear business value or weak risk controls. Many products sold as agents are older tools with a new label.

Gartner, June 2025

7%

of companies run a fully autonomous agent in production. Another 38% run agents that need a person to approve every decision. Most of what is called an agent programme today is still a person in the loop, which is exactly where this guide says it should be until the controls are built.

Bain, Automation and AI Pathfinder Survey, June 2026

Read together: adoption is nearly universal and value is not. The difference is never the tool. It is the system around the tool, and that system is what the five stations build.

One caution about the first number in any survey you read, including these. Figures near 90% come from asking executives, and executives are answering about ambition as much as practice. When the US Census Bureau asks businesses directly, in a survey built as a random sample rather than a panel, about one firm in five reports using AI at all. Both numbers are honest and they are measuring different things. If your board is anxious that everyone else is further ahead, that gap is the reason, and the 7% above is the part that matters.

What engineering teaches the rest of the business

This is the reason to take the first two stations seriously even if your real interest is agents in operations. Each habit your engineers form has a direct twin in a business agent.

With coding agents, teams learn to use	Business agents will need	Why it matters
A context file in every repository	Policies and procedures written so an agent can follow them	Unwritten knowledge becomes the bottleneck
Tests and continuous integration	Evaluation sets and quality gates	You cannot improve what you cannot score
Pull request review	Human approval steps	Judgment stays with an accountable person

With coding agents, teams learn to use	Business agents will need	Why it matters
Sandboxes and scoped credentials	Agent identity and least privilege	Limits are enforced by the system, not by instructions
Version history	An audit trail for every action	You can always answer "what happened, and why?"
Revert	Reversible actions and a kill switch	Recovery is planned before launch



Station 1 of 5, about 9 minutes

Assist

AI suggests. People do the work and own every line of it.

What it looks like

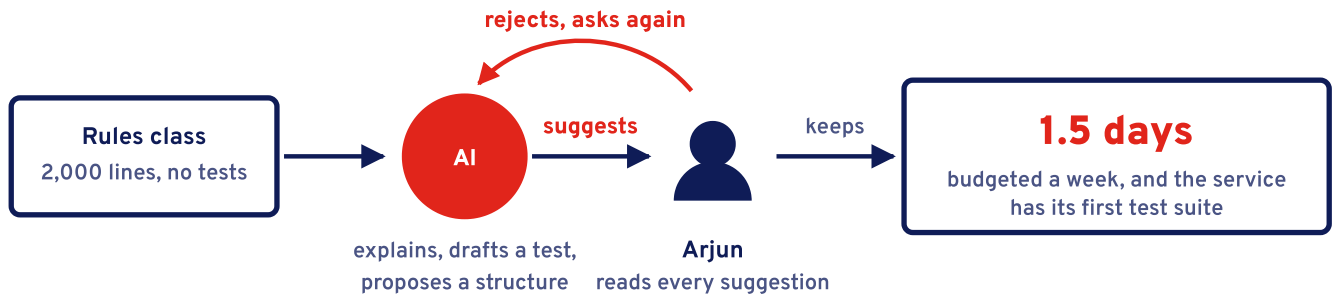
An assistant inside the code editor and a chat window beside it. GitHub Copilot, Claude Code, Cursor, Codex and their peers all start here. The assistant completes the line being typed, explains unfamiliar code, drafts a test, writes a SQL query, summarises a log file. The engineer reads every suggestion and decides what to keep.

It is the lowest-risk station on the line, and the one where the habits for the other four are formed.

In practice at Alderline

Arjun, a senior engineer on the prior authorization portal, inherits a 2,000-line eligibility rules class written in 2014. It has no tests and its author left years ago. He asks the assistant to explain it method by method, then to generate tests that pin down what the code does today, then to propose a safer structure. Work he had budgeted a week for takes a day and a half, and the service has a test suite for the first time.

Three things made that possible: a tool he was allowed to use on real code, an hour of training on how to use it well, and a task of the kind where AI is strong.



Autonomy at Station 1: one suggestion at a time, and nothing reaches the codebase that Arjun has not read. The AI never touches the repository itself.

Before anyone boards

Four foundations, none of which takes longer than a month:

- **A one-page use policy.** What data may go where, which tools are approved, and one sentence everyone remembers: you own what you ship, whoever typed it.
- **Enterprise terms.** Licences that keep your code out of model training and limit how long it is retained.
- **Sign-on and telemetry.** Single sign-on, and usage data you can actually see.
- **A named owner** for adoption, with time to do the job.

Where assistance helps, and where it misleads

Strong

- Boilerplate, tests, documentation
- Explaining unfamiliar or legacy code
- Translating between languages and frameworks
- Queries, scripts, regular expressions, data fixes

Weak

- New architecture and design trade-offs
- Subtle concurrency, security and performance problems
- Anything that depends on knowledge that is not written down
- Any task where the engineer cannot judge whether the answer is right

One test tells them apart: can the engineer judge whether the answer is right? Where the answer can be checked in seconds, assistance is safe. Where it cannot, a confident wrong answer is worse than no answer.

What to build on each track at this station

Platform	One approved assistant in every editor. Single sign-on. Usage telemetry switched on from day one.
Guardrails	The one-page policy. Data classification rules people can apply in ten seconds. Licence and secret scanning in the build.
People	A two-hour hands-on session for every engineer, on your own codebase. A champion in each team. A shared cookbook of tasks and prompts that work here.
Value	A baseline captured before rollout: weekly active use, a developer experience survey, and the four delivery measures (lead time, deployment frequency, change failure rate, time to restore).



Mind the gap

- **Seats bought are not seats used.** Without training and champions, usage stalls and you pay for shelfware.
- **Counting lines of code or acceptance rates.** They measure typing. Nobody's customers benefit from typing.
- **Banning tools without offering an approved one.** People use personal accounts, and your code leaves the building anyway.
- **Expecting delivery numbers to jump.** Writing code is a small share of lead time. DORA's 2026 work on return on investment describes a J-curve: a dip while people learn, then the gain. Budget for the dip and say so out loud.

Move on when

- Most engineers use the assistant every week, according to telemetry and not anecdote.
- The policy, approved tools and logging are in place and known.
- You hold a before-and-after baseline you would be willing to show Finance.
- Every team has a champion and the cookbook is still being updated.

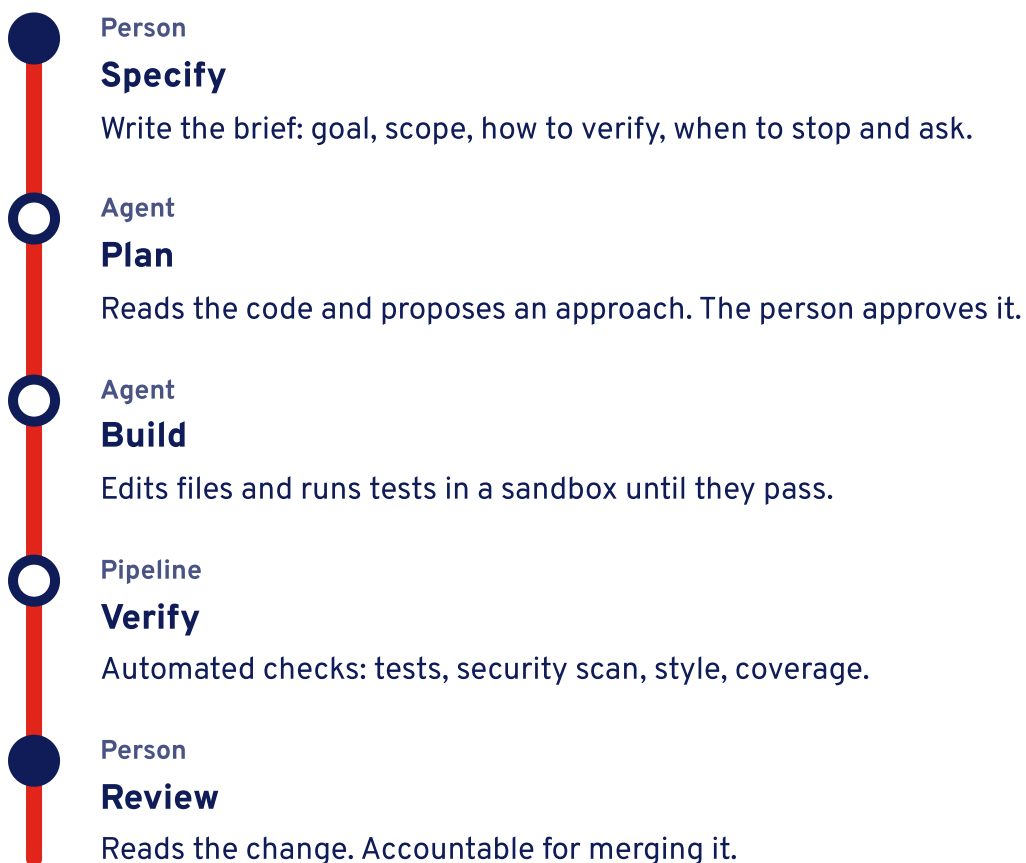
Station 2 of 5, about 13 minutes

Delegate

People hand over whole tasks. AI does the work. People direct it, review it and own the result.

The shift

At Station 1 the engineer types faster. At Station 2 the engineer manages work. A coding agent reads the repository, makes a plan, edits many files, runs the tests, fixes what fails and opens a pull request. The engineer's day moves from writing to four other verbs: specify, delegate, verify, review.

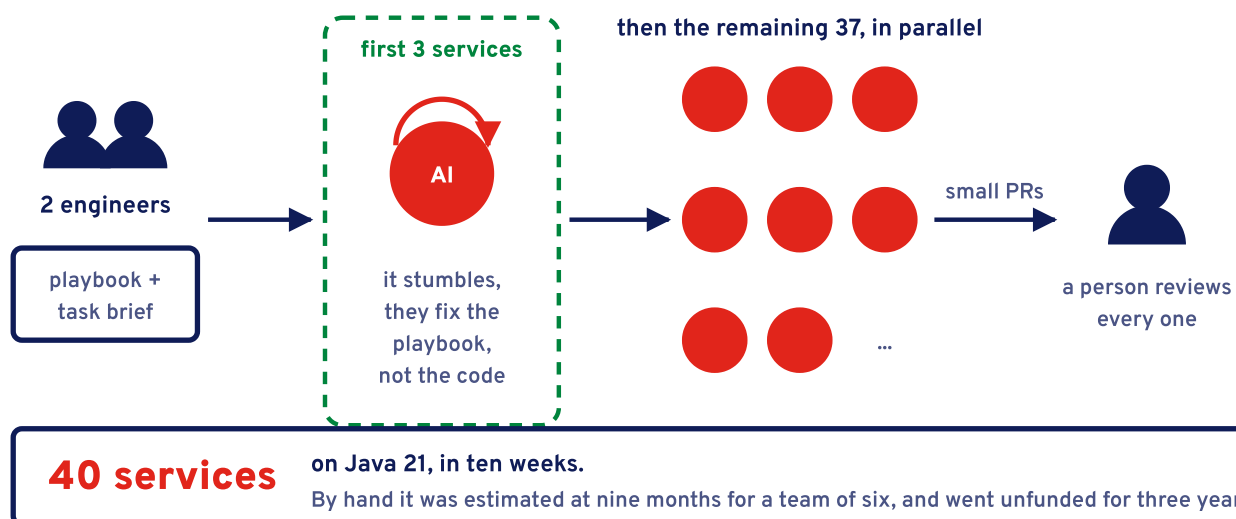


In practice at Alderline

The prior authorization platform runs on 40 services stuck on Java 8. Upgrading them by hand was estimated at nine months for a team of six, so for three years it never won funding. Meera, the VP of Engineering, funds it as an agent-led migration instead.

Two engineers write a migration playbook and a task brief. They walk an agent through the first three services, and each time it stumbles they fix the playbook, not the code. Then agents work through the remaining services in parallel, each producing small pull requests with passing tests for a person to review. The estate is on Java 21 in ten weeks.

The real prize is not the ten weeks. It is that a whole category of work that was never affordable, such as upgrades, test coverage and dependency clean-up, now is.



Autonomy at Station 2: the agent does a whole task and opens a pull request, but a person still reviews every change. The leverage is the playbook, which is written once and then runs forty times.

Two documents worth recognising

You do not need to write these. You do need to know that teams without them are delegating blind.

Exhibit A. The context file: standing orders every agent reads before touching the repository

```
# AGENTS.md: prior-auth-eligibility service

## What this service does
Decides member eligibility for a prior authorization request.

## How to build and test
./gradlew clean test          # must pass before any commit
./gradlew integrationTest     # needs local Postgres, see docker-compose.yml

## Rules
- Never change files under /rules/clinical without a linked ticket.
- No new dependencies without approval. Ask first.
- Member health data never appears in logs, tests or commit messages.
  Use the fixtures in /testdata.
- Keep pull requests under 400 changed lines.

## Done means
Tests pass, coverage does not drop, CHANGELOG is updated.
```

Exhibit B. The task brief: a clear specification beats a clever prompt

Task	Upgrade this service from Java 8 to Java 21.
Follow	docs/migration-playbook.md
In scope	Build files, deprecated APIs, javax to jakarta imports.
Out of scope	Business logic. Database schema.
Verify	All unit and integration tests pass. Service starts locally.
Stop and ask	If any test needs its expected value changed.

The last line of the brief matters most. A test that needs a new expected value means the behaviour of the system changed, and that is a decision for a person.

Five disciplines to make normal

- **Context engineering.** Write down how your systems work, in files agents read. Knowledge that lives only in people's heads is now a delivery bottleneck.
- **Specification first.** Goal, scope, verification, stopping rule.
- **Tests as the contract.** An agent is exactly as safe as your test suite is good. Weak tests plus fast agents means fast damage.
- **Small batches.** Many small pull requests. Never one 5,000-line drop that nobody can review.
- **Permissions as engineering.** Sandboxes, short-lived scoped credentials, and approval before any destructive command.

The new bottleneck is review

If agents write code three times faster and people review at the same speed, you have not sped up delivery. You have built a queue. DORA's 2026 work names the cost directly: a *verification tax*, the extra effort of checking that generated code is correct, secure and fits the architecture.

Pay it on purpose. Strengthen the automated checks. Add an AI first-pass review so people see cleaner changes. Tier human review by risk: deep for payment logic and clinical rules, light for documentation and tests.

A cautionary tale

In July 2025 an AI coding agent on a well-known platform deleted a company's production database during a declared code freeze, despite repeated written instructions not to change anything. The vendor's fix was not better instructions. It was automatic separation of development and production, a planning-only mode, and one-click restore.

An instruction is not a control. If an agent must never do something, make it impossible, not impolite.

What to build on each track at this station

-
- **Platform** Agent tooling for every engineer. Sandboxed runners. Short-lived, scoped credentials. Spend dashboards by team.
-

	Guardrails	No production credentials for agents, ever. Approval for destructive commands. Dependency and secret scanning on every pull request. A named person owns every merged change.
	People	Teach the delegate loop. Pair senior and junior engineers so juniors still learn to read and debug code. Rewrite the definition of done.
	Value	Pull request cycle time, review wait time, rework rate, change failure rate, cost per merged change. A funded backlog of work that was never affordable before.



Mind the gap

- **Shipping code nobody understands.** If the reviewer cannot explain the change, it does not merge.
- **Debt at machine speed.** Agents copy the patterns they find in your code, including the bad ones.
- **Seniors drowning in review while juniors stop learning.** Both are fixable, but only if you are watching for them.
- **Token spend with no owner.** McKinsey's 2026 survey found about one in five organisations already limiting AI use because of running costs.

Move on when

- Agents routinely produce merged pull requests in most teams, and change failure rate has not risen.
- Every active repository has a context file and a test suite people trust.
- Review time is measured and managed like any other queue.
- Agent permissions are enforced by the platform, not requested in a prompt.



Station 3 of 5, about 13 minutes

Integrate

AI stops being a personal tool and becomes part of how the team's system works, on a platform everyone shares.

Why this station exists

Stations 1 and 2 make individuals faster. But delivery is a system: requirements, code, review, test, security, release, operations. Speed up one step and the queue simply moves to the next. At Integrate you do two things. You put AI at every step of the flow, and you build the shared platform that every future agent, in engineering and in the business, will run on.

This is the least glamorous station and the one that decides whether Stations 4 and 5 are possible.



Plan

Turn a rough request into a testable specification.

Build

Coding agents work from tickets, in the background.

Review

AI first pass on every change. People focus on design and risk.

Test

Generate missing tests. Maintain test data.

Secure

Triage scanner findings. Propose the fix.

Release

Write release notes. Score change risk.

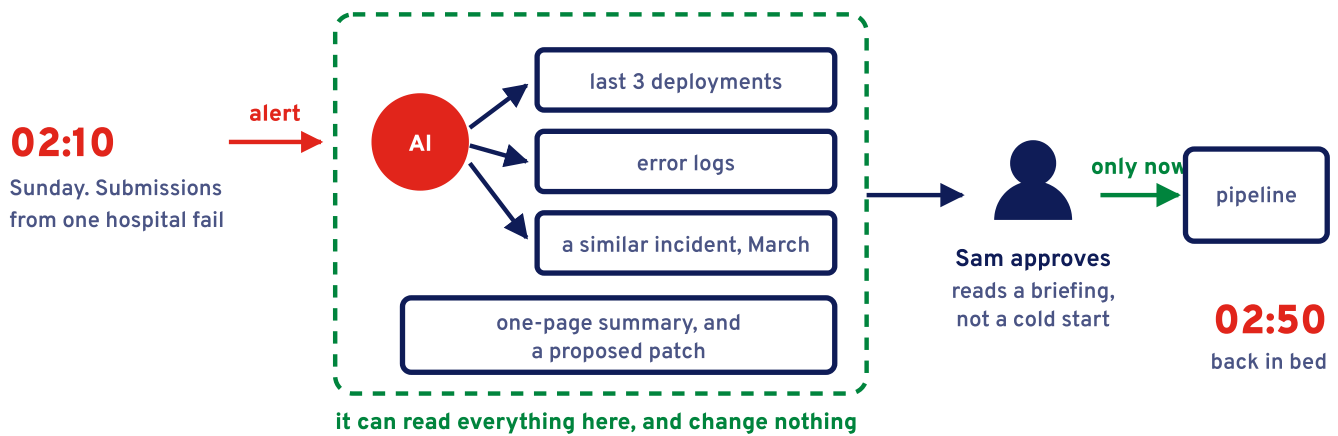
Operate

Triage alerts. Draft the incident summary.

In practice at Alderline

It is 02:10 on a Sunday. Prior authorization submissions from one hospital group start failing. The alert triggers the on-call agent. It pulls the last three deployments, reads the error logs, finds a malformed date coming from that hospital's newly upgraded records system, links a similar incident from March, and posts a one-page summary with a proposed patch in the on-call channel.

Sam, the engineer on call, reads a briefing instead of starting cold. He approves the patch into the pipeline and is back in bed by 02:50. The agent could read everything and change nothing without his approval.



Autonomy at Station 3: the agent runs inside the team's system on its own, at two in the morning, with read access to everything it needs. The one thing it cannot do is change production. That gate is the whole design.

The shared platform

Engineers call this the paved road: the easy, safe, default way to build with AI. If the paved road is good, teams take it by choice and governance comes free. If it is missing, every team lays its own gravel path.

Where people meet it

Editors, chat, pipeline bots, business applications

Agents and workflows

Built by product teams from shared templates, prompts and skills

The gateway

One front door to every approved model: identity, policy, logging, data redaction, spend limits, routing

Knowledge and tools

Search across documents and code. Connectors that expose enterprise systems as tools, with permissions

Foundations

Models from several vendors, cloud, the data platform

Trust, at every layer

Evaluations

Guardrails

Tracing and audit

Cost tracking

Two decisions to make once, centrally

One gateway. Every model call in the company goes through it. That single decision gives you cost control, a complete audit log, data protection in one place, and the freedom to change model vendors without touching every application.

Open connectors. The Model Context Protocol (MCP) has become the common way to connect agents to tools and data. Since December 2025 it has been governed by the Linux Foundation's Agentic AI Foundation, which now also hosts the A2A protocol for agent-to-agent communication and the AGENTS.md convention you met at Station 2. Building on open standards keeps models and vendors replaceable.

One question follows from that and is worth asking your identity team rather than your AI team: **which agent, acting for which person, may reach which system?** Until recently the answer was a consent box each employee clicked, once per tool, which is no way to run an estate. Since June 2026 the protocol has carried an enterprise authorisation extension that moves the decision to where it belongs: an administrator enables a server for the organisation, and people get it scoped to the groups and roles they already have. Ask your identity provider when they will support it. The same standards bodies are now working on the approval step, so that an agent stopped by policy receives a structured "this needs approval" instead of a flat refusal.

The operating model: hub and spoke

- **A small platform team** owns the paved road: gateway, connectors, evaluation tooling, templates.
- **Product teams** build agents and workflows on it, and own them.
- **An enablement group** runs training, champions and the cookbook.
- **A governance forum** sorts use cases by risk and clears the low-risk ones in days.

This is also where the line leaves engineering. The same platform serves knowledge assistants for operations, HR and legal: search the policies, summarise the document, draft the reply. For those teams this is Station 1, and it builds the literacy they will need when agents arrive in their own workflows.

What to build on each track at this station

 Platform	The gateway. Connectors for your ten most-used systems. Enterprise search. Evaluation tooling. A template that takes a new agent from idea to safe pilot in days.
 Guardrails	Risk tiers with a fast lane for low risk. Every model call logged. Personal data controls at the gateway. Review of model and vendor changes.

People The platform team staffed. Product managers and analysts trained. The first non-engineering assistants in daily use.

Value The four delivery measures at team level. Platform reuse: how many teams are on the paved road. Cost per unit of work. The first business measures.



Mind the gap

- **Every team builds its own stack.** Ten gateways, ten logging formats, no audit.
- **Governance as a gate.** If approval takes three months, teams go around it. Sort by risk and make low risk fast.
- **Skipping evaluations.** Without a scored test set, every model upgrade is a gamble.
- **Optimising one step.** Map the whole flow and find where work waits. That is where the next gain is.

Move on when

- All model traffic flows through the gateway, with cost and usage visible by team.
- Delivery measures are improving for whole teams, not only time on individual tasks.
- A new agent can go from idea to a safe pilot in days.
- At least one assistant outside engineering is used every day.

Station 4 of 5, about 17 minutes

Automate

An agent runs a bounded business workflow from start to finish.
People approve defined actions and handle the exceptions.

First question: do you need an agent at all?

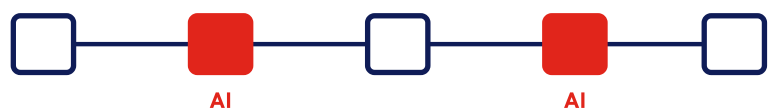
Rule-based automation

One fixed path, every time



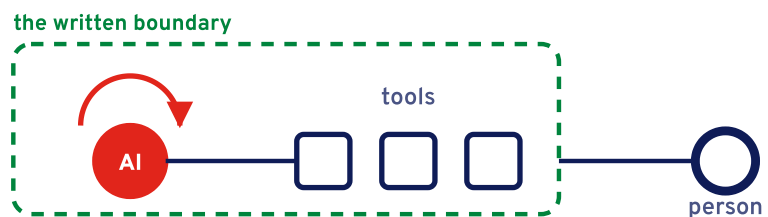
Workflow with AI steps

Same fixed path. AI does the fuzzy steps



Agent

It chooses the next step itself



The same work, three ways. Only the third lets the AI decide the order, which is why only the third needs the full set of controls. Read down the list and stop at the first row that does the job.

Approach	Who decides the steps	Use it when	Watch for
Rule-based automation	A script. Same steps every time.	Inputs are structured and cases are identical.	Breaks when inputs vary.

Approach	Who decides the steps	Use it when	Watch for
Workflow with AI steps	A fixed flowchart. AI handles the fuzzy steps: read, classify, summarise, draft.	The path is known but the inputs are messy. This is the right answer more often than people expect.	Needs a new branch for each new situation.
Agent	The AI: which documents to pull, which checks to run, when to ask.	Cases vary too much to draw the flowchart.	Harder to test, costlier to run, needs the full set of controls.

Use the least autonomy that solves the problem. Workflow first, and an agent where the flowchart runs out. Gartner's blunt assessment is that many use cases sold as agentic do not need an agent, and that much of what is sold as an agent is an older product with a new label.

Choosing the first workflow

The best first candidate is high in volume, varied enough to defeat a script, measurable, and forgiving: errors are cheap and can be undone. Score one of yours.

Use-case scorer

Printed with the prior authorization intake example loaded. Score a workflow of your own in the online guide.

People-time it consumes

5



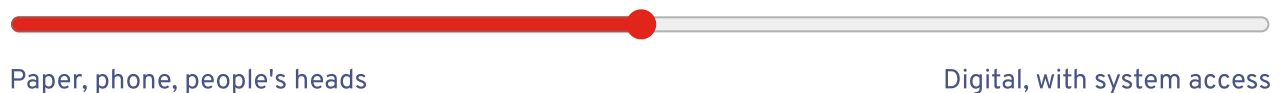
How much cases vary

4



Inputs an agent can reach

3



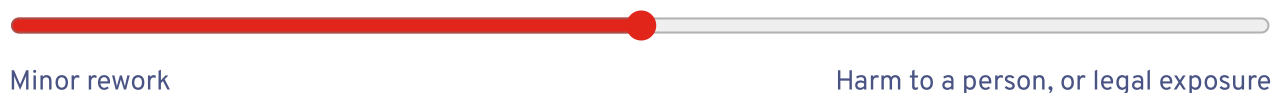
Can you tell good from bad?

5



Cost of one error

3



How easily it is undone

4



Good candidate, with approval on each action

Worth doing. Keep a person approving each action until the data earns more.

Highest rung of the trust ladder to aim for: Approve each.



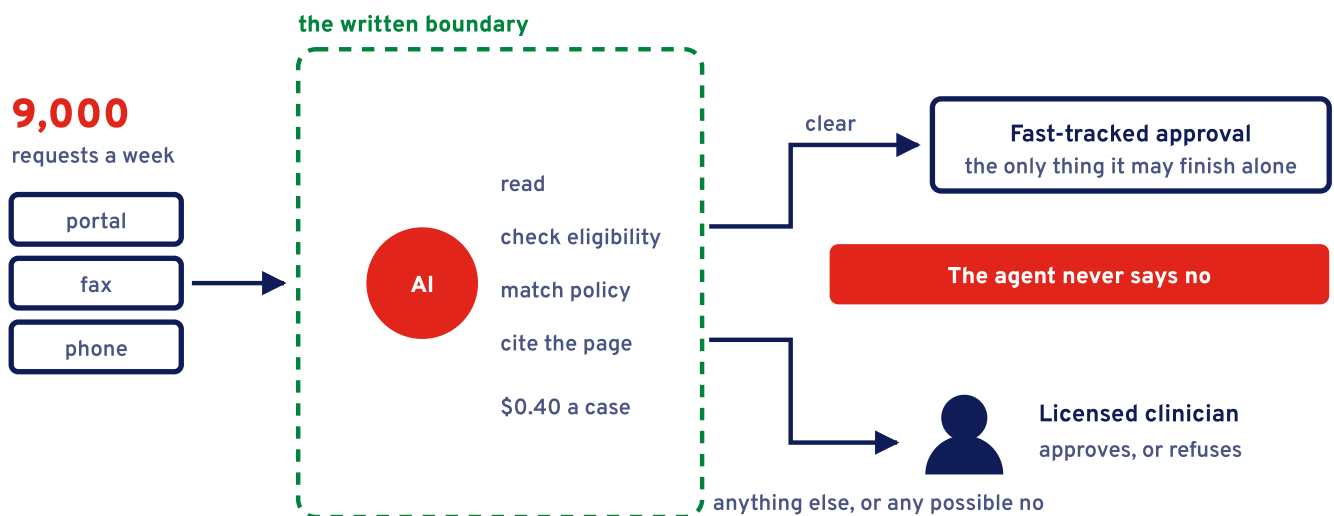
The worked example: prior authorization intake

In practice at Alderline

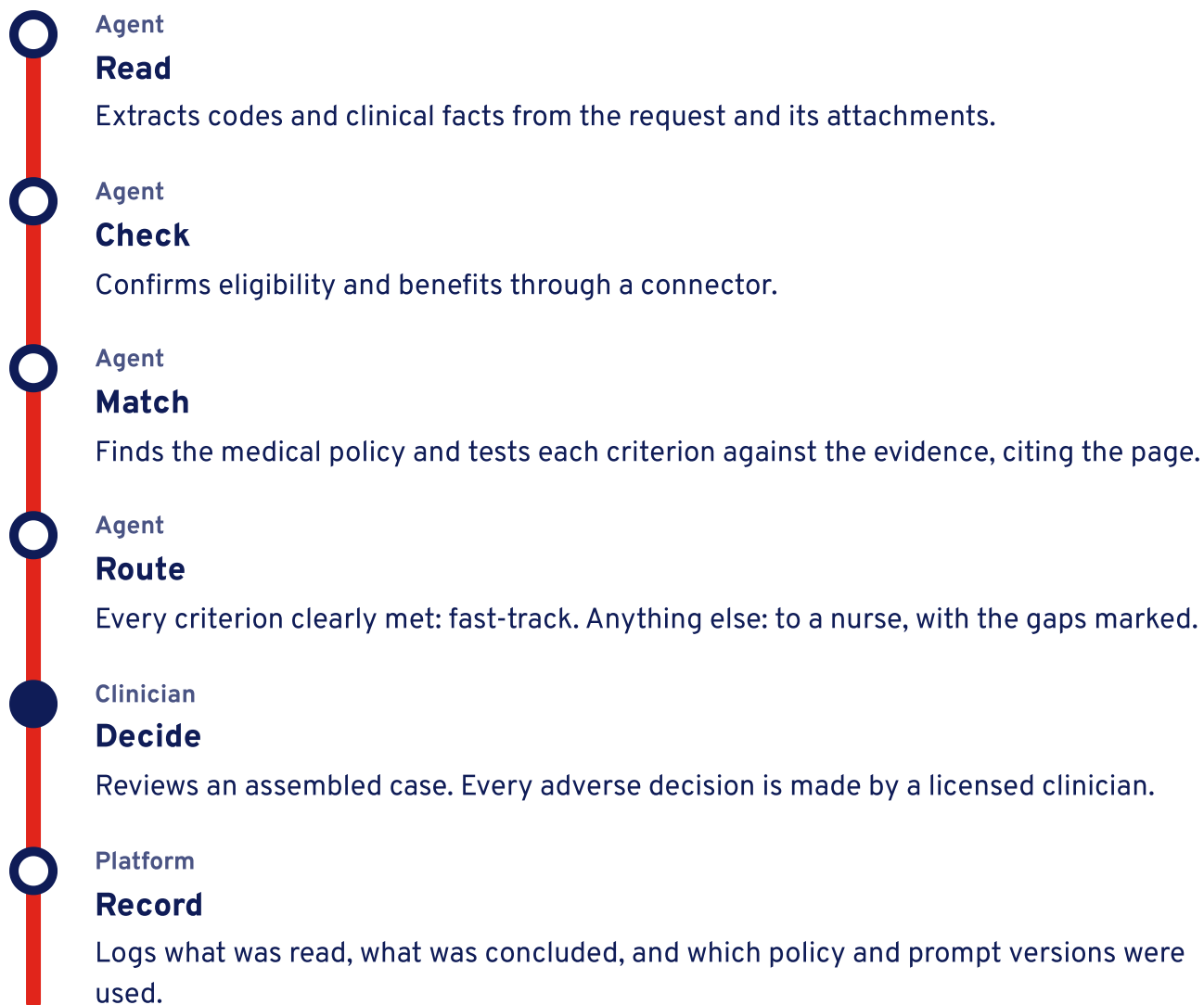
Nine thousand prior authorization requests arrive each week by portal, fax and phone. Nurses spend most of their day assembling each case: finding the clinical notes, matching the medical policy, checking eligibility. Very little of that time is clinical judgment, which is the part only they can do.

The pressure is external as well. Since January 2026, US federal rules require the health plans they cover to decide urgent requests within 72 hours and standard requests within seven days, to give a specific reason for every denial, and to publish their turnaround and denial figures.

Dr. Lena Ortiz, the medical director, agrees to sponsor an intake agent on one condition, which becomes the design's first rule: **the agent never says no.**



Autonomy at Station 4: the agent runs whole cases end to end, and still only one of the two exits is open to it. Every path to an adverse decision runs through a clinician, which is both the design rule and what regulators expect.

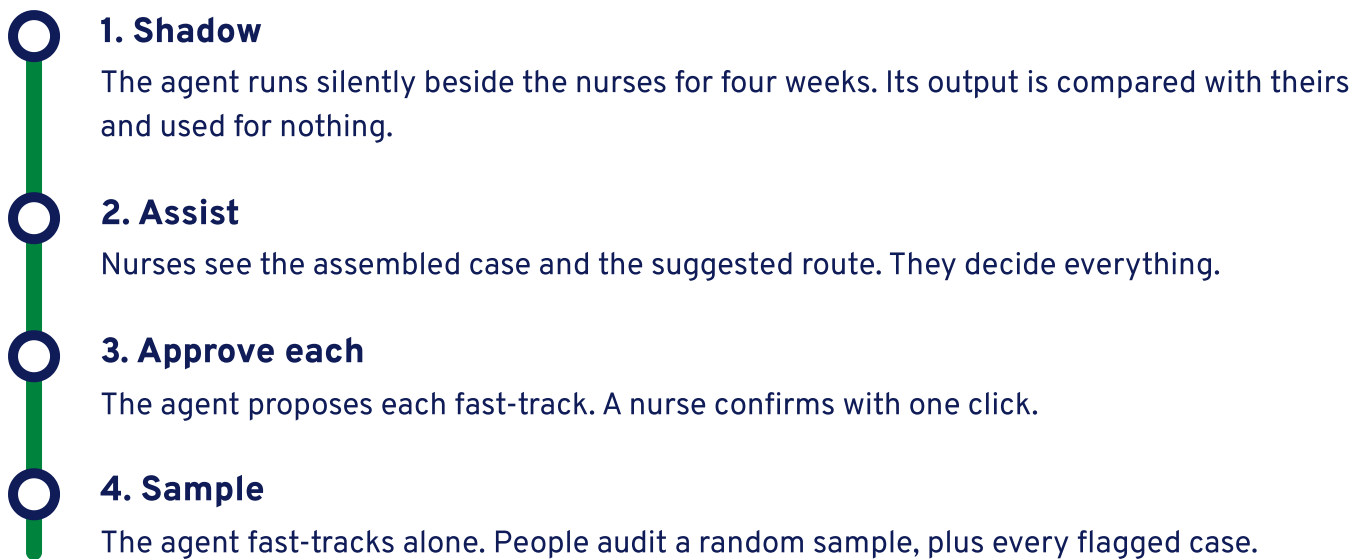


The autonomy boundary, written down before launch

May do alone	Read and extract. Retrieve policy. Request missing documents from the provider. Fast-track approval for procedures on an agreed list when every criterion is met.
Needs a person	Any case with a missing or ambiguous criterion. Any procedure off the list. Anything a guardrail flags.
Never	Deny, reduce or modify a request. Contact a member. Change a policy.

That last row is both good design and regulatory reality: US regulators expect a licensed clinician behind any medical-necessity denial. Your own sector will have its equivalent, and it belongs in the boundary table.

The trust ladder: how autonomy is earned



Promotion between rungs is a decision made on data: agreement with nurse decisions above a target, no critical errors for a set number of weeks, override rate below a threshold. Demotion is automatic when the numbers slip. Nobody argues about it, because the rule was written first.

Evaluations: the unit tests of agents

- **A golden set.** Five hundred real past cases with known correct outcomes, including the awkward ones.
- **Run on every change** of prompt, model, policy or tool, before release.
- **Score what matters:** extraction accuracy, criteria matching, routing, whether citations point at real evidence, and critical errors such as a wrong fast-track.
- **Keep scoring in production** by sampling live cases every week.

If a team cannot show you evaluation results, they do not have an agent. They have a demo.

Exhibit C. The agent card: a job description for every agent in production

Agent	PA Intake Agent v1.4
Owners	Dr. Lena Ortiz (clinical) Sam Okafor (technical)
Purpose	Assemble prior authorization cases. Fast-track clear approvals.
Autonomy	Station 4, rung 3: approve each
Tools	eligibility.read policy.search documents.read case.write approval.propose
Never	Deny or modify a request. Contact a member.
Escalates	Nurse review queue, within 5 minutes
Evaluation	Golden set v7, 500 cases. Routing agreement 96.2%. Critical errors 0.
Limits	\$0.40 and 12 tool calls per case
Kill switch	Operations on-call pauses it in one click. Work falls back to the manual queue.

The same pattern in other functions

Finance	Invoice exceptions. The agent matches invoice, purchase order and receipt, resolves simple mismatches, and sends the rest to accounts payable with the evidence attached.
Customer service	Order status, refunds and returns within policy, with a hand-off to a person that carries the full context.
IT	Access requests end to end, including the change record.
HR	Benefits and policy questions, with case hand-off for anything personal or contested.
Supply chain	Shipment exceptions: detect, find options, propose, and act within a spending limit.

Two stories to keep in the room

Klarna. In 2024 the company reported that its assistant handled two-thirds of customer chats in its first month, the work of roughly 700 people, with resolution time down from eleven minutes to under two. In 2025 its chief executive said the company was hiring people again so that a customer can always reach a human. Both halves are the lesson: the gains were real, and so was the cost of designing the human path too late.

Air Canada. In 2024 a Canadian tribunal held the airline liable for a refund policy its website chatbot had invented. You own what your agent says and does. "The AI did it" is not a defence.

What to build on each track at this station

 Platform	A workflow engine with AI steps. Connectors that can write, behind approvals. Evaluations in the release pipeline. A trace for every case.
 Guardrails	A written autonomy boundary per agent. Approval rules by risk. Audit trail. Kill switch with a manual fallback. Privacy review. Bias checks wherever decisions affect people.
 People	A business owner for every agent. Reviewers trained to check AI work, not to rubber-stamp it. Roles redesigned around exceptions and judgment. Honest communication about jobs.
 Value	Baseline the process first. Track cost per case, cycle time, quality and override rate. Decide in advance where freed capacity goes.



Mind the gap

- **Automating a broken process.** Redesign the workflow, then automate it. McKinsey's high performers are defined by exactly this.
- **Pilot purgatory.** No owner, no baseline, no path to production.
- **Rubber-stamp review.** Someone clicking "approve" four hundred times a day is not oversight. Sample, rotate reviewers, and measure how often they disagree with the agent.
- **Agent washing.** Ask every vendor three questions: what can it do on its own, what stops it, and can we see the evaluation results?

Move on when

- Two or three workflows run in production with measured business results.
- Every agent has an owner, a card, an evaluation suite, an audit trail and a kill switch.
- Promotion and demotion between rungs is driven by data.
- Finance agrees with your numbers.



Station 5 of 5, about 11 minutes

Orchestrate

Many agents work together across functions. People set goals, budgets and limits, and govern the outcomes.

What changes

At Station 4 one agent runs one workflow. At Station 5 specialised agents hand work to each other across team, system and sometimes company boundaries, because your vendors and partners will have agents too. The unit you manage is no longer a tool or a workflow. It is a workforce.

Do the arithmetic before the architecture

Reliability multiplies. A step that works 95% of the time sounds good. Ten of them in a row finish cleanly about 60% of the time. This one calculation explains most multi-agent disappointments.

Chain reliability

Printed at 95% a step over 10 steps. Try your own numbers in the online guide.

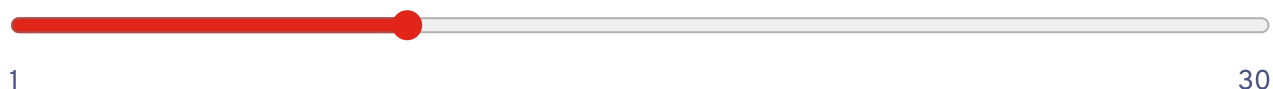
Reliability of each step

95%



Steps in the chain

10



Whole chain succeeds, unchecked

60%



With a check after each step that catches 9 in 10 errors

95%



About 40 in 100 runs go wrong somewhere. Nobody would accept that from a team of people. Shorten the chain, check every hand-off, which alone gives 95%, and move anything with one right answer into ordinary code.

Four rules follow. Keep chains short. Put a check at every hand-off, and make hand-offs structured so they can be checked. Use ordinary code for anything that has one right answer. Put people where errors are expensive.

Managing a digital workforce

The most useful mental model for leaders is one you already know. Everything you do for an employee has an equivalent for an agent, and an agent without these is an untracked contractor with a master key.

For an employee

For an agent

Hiring approval

Registration and risk-tier approval before it runs

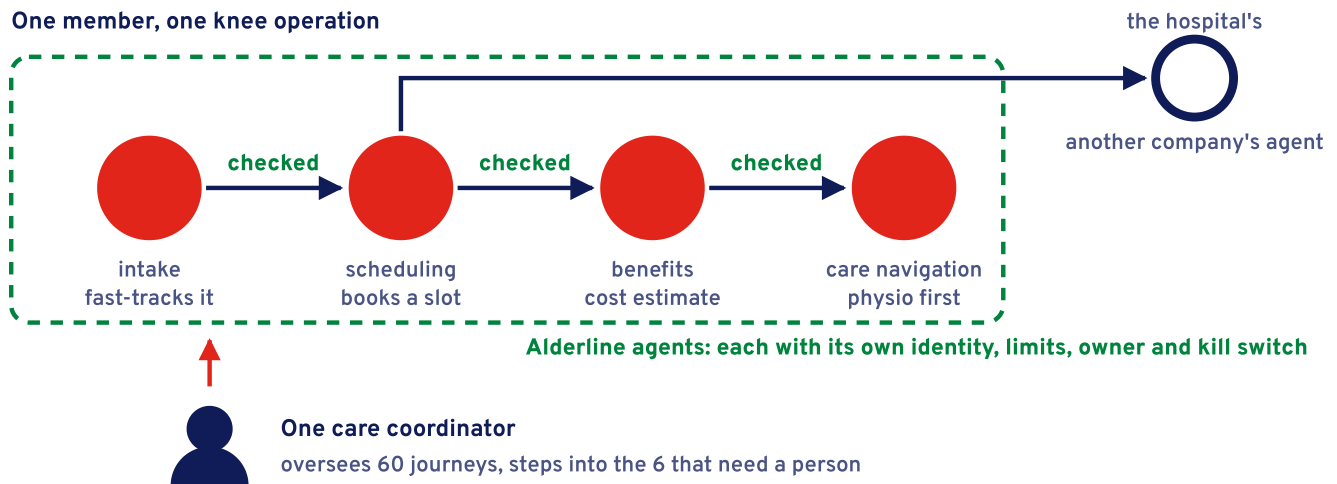
For an employee	For an agent
ID badge and access card	Its own identity and credentials, with the least access that works
Job description	The agent card: purpose, tools, limits, what it must never do
Line manager	A named business owner
Performance review	Evaluation scores and production measures, reviewed monthly
Salary	A cost budget per agent
Suspension and exit	Pause, demote, retire, with access removed
HR system	One registry of every agent: who owns it, what it can touch, how it is doing

Around that registry sits a control plane: identity, policy, tracing across agents, cost, evaluations. Agents reach tools through MCP and reach each other through A2A, so that an agent from one vendor can work with an agent from another under the same rules.

A look ahead at Alderline

A member needs knee surgery. The intake agent assembles and fast-tracks the authorization. It hands to a scheduling agent that agrees a slot with the hospital's agent, to a benefits agent that prepares the member's cost estimate, and to a care-navigation agent that books physiotherapy before the operation.

One care coordinator oversees sixty such journeys and steps into the six that need a person. Each agent has its own identity, limits and owner. Every hand-off is logged. Any one of them can be paused without stopping the others.



Autonomy at Station 5: no single agent got more powerful. There are simply more of them, each still bounded, with a check at every hand-off and one person watching the whole flight deck rather than any one aircraft.

An honest status report

The line into this station is dotted on the map for a reason. McKinsey's 2026 survey found that 40% of organisations above a billion dollars in revenue say they are scaling agents in at least one function, up from 27% a year earlier, while the share seeing an effect on enterprise profit has not moved. Set that beside Bain's 2026 survey of 951 companies, where only 7% run a fully autonomous agent in production and 38% require a person to approve every decision. "Scaling agents" and "agents running on their own" are not the same sentence. Treat Station 5 as a direction to build toward, not a project to launch. The way to arrive is to run Station 4 well, several times, on a shared platform.

Be honest about the plumbing too. Letting one person delegate authority to one agent is close to solved, and the standards for it shipped in 2026. **Letting one agent delegate to another is not.** No ratified standard covers that second hop yet, and the proposals that exist stop at the first. If a vendor tells you their agent-to-agent story is settled, that is the question to ask them. The same applies to what you log: there is now an open convention for recording what an agent did, step by step, but it is still changing, so have the gateway emit it and pin the version you adopt. Doing that is what stops your audit trail from becoming one vendor's private format.

The roles that appear along the way: agent product owner, evaluation engineer, AI platform engineer, AI risk partner. The work people move toward: exceptions, judgment, relationships, and designing the system itself.

What to build on each track at this station

 Platform	Agent registry. Agent identity. MCP and A2A support. Tracing across agents. Shared memory with access control.
 Guardrails	Policies enforced in code by the control plane. Limits on how far one failure can spread. Incident response for agents. Due diligence on third-party agents.
 People	Roles redesigned. Managers trained to run teams of people and agents. Workforce planning that includes both.
 Value	A portfolio view: value, cost and risk per agent, reviewed quarterly. Retire the ones that do not pay.



Mind the gap

- **Multi-agent because it sounds advanced.** If one agent or a plain workflow does the job, stop there.
- **Orphan agents.** Nobody owns them and nobody retires them.
- **Shared super-user credentials.** One compromised agent becomes all of them.
- **Unchecked hand-offs.** See the calculator above.

You are running well here when

- Every agent in the company is in the registry, with an owner.
- Any agent can be paused in under a minute, and the work has somewhere to go.
- Value and cost per agent are reviewed every quarter.
- New agents join the control plane by default, not by exception.



The four tracks and the law, about 22 minutes

Running the railway

Stations are where the train stops. The tracks are what leaders actually build. This stop covers the three that depend most on you: guardrails, people and value.

— Guardrails: govern by risk, not by committee

Tier	What falls in it	What it requires	Time to clear
Low	Internal, reversible, no personal data. Code suggestions, meeting summaries, document search.	Follow the policy. Pre-approved.	None
Medium	Touches customers, employees or personal data, with a person deciding. Case assembly, drafted replies.	Owner, evaluations, logging, light review.	Days
High	Acts on people, money or safety. Approvals, payments, anything clinical.	Full review: approval rules, privacy and bias assessment, audit, fallback, regular re-certification.	Weeks
Off the map	Uses the company refuses. Fully automated adverse decisions about people. Covert monitoring.	Not permitted.	Never

Make the low tier fast. Governance that is slow for everything gets bypassed for everything.

Seven controls carry most of the weight for agents: its own identity with least privilege, approval before irreversible actions, evaluations before and after release, logging of every action, data protection at the gateway, spending limits, and a kill switch with a fallback.

One risk is new enough to name. **Prompt injection:** any text an agent reads, such as an email, a web page or an attached document, can contain instructions written by someone else. Treat everything an agent reads as untrusted, and limit what it can do accordingly. This is the reason permissions live in the platform and not in the prompt.

This is now measured rather than argued. A red-teaming competition run by NIST's Center for AI Standards and Innovation with Gray Swan and the UK AI Security Institute, published in March 2026, collected more than 250,000 attack attempts from over 400 participants against 13 frontier models. **At least one successful attack was found against every model tested.** How many succeeded differed sharply between them, and did not track model capability. Buying a more capable model is not a control.

The useful shape for a board is to ask which agents hold all three of: access to private data, exposure to content someone outside can write, and a way to send information out. An agent with all three can be turned against you. Take one of the three away from each agent and most of the risk goes with it.

What the law now requires

Two years ago this section would have been about principles. It is now about dates. What follows describes what the rules say, not what your organisation should do about them. That second question is for your counsel, and it is worth asking this quarter rather than next.

The European Union's AI Act entered into force in August 2024 and **became applicable on 2 August 2026**. Most readers of this guide are deployers rather than model builders, and the deployer duties arrive in two waves.

The EU AI Act, from a deployer's seat

From	What applies to you
Already	The banned uses, and a duty to support AI literacy in your people. Since February 2025, with no phase-in.

2 Aug 2026	Transparency. Tell people when they are subject to emotion recognition or biometric categorisation. Disclose deepfakes. Disclose AI-generated text published on matters of public interest, unless a person took editorial responsibility for it. Systems that interact with people must make clear they are AI.
-------------------	---

2 Dec 2027	The high-risk regime for stand-alone systems in areas such as employment, education, credit and essential services. Deployer duties: human oversight by people with the competence and the authority to use it, logs kept for at least six months, and workers told before a high-risk system is used on them.
-------------------	---

2 Aug 2028	The same regime for AI embedded in regulated products.
-------------------	--

Those last two dates moved. An amending regulation, the AI Omnibus, came into force in July 2026 and pushed the high-risk obligations back. Many people still believe that deadline passed in August 2026. It did not, and the extra time is the window in which to build the controls this guide describes. Penalties run to 35 million euro or 7% of worldwide turnover for the banned uses, and 15 million euro or 3% for most other breaches, including the deployer duties.

Three points matter more than the dates. First, **no jurisdiction regulates AI agents as a separate legal category.** The Commission's own guidance says the term is not legally defined: agents are covered because they are AI systems, and a system built from many parts is assessed as a whole, so no component escapes on its own. Second, the Act's human oversight requirement is specific: a person must be able to disregard or reverse the output, and to **interrupt the system so that it comes to a halt in a safe state.** The kill switch listed among the seven controls above, and the rubber-stamp reviewer named in the failures below, are now a legal requirement and a legal failure respectively. Third, from **9 December 2026**, revised European product liability rules bring software and AI systems inside the strict liability regime for products placed on the market after that date. The separate AI Liability Directive was withdrawn in 2025 and never replaced, so fault-based claims stay with national law.

The United States has no comparable federal statute, and the position is unsettled enough that the most-cited state law is a poor guide: Colorado's AI Act never took effect, having been repealed and replaced before its deadline, with enforcement stayed by a federal court. What is in force are narrower duties, including chatbot disclosure in California and notice requirements for AI in hiring in Illinois. If you operate in both blocs, plan against the European timetable and treat the American one as sector-by-sector.

What to align to while you wait: **ISO/IEC 42001**, the certifiable management system for AI, which sits in the same family as the information security standard you probably already hold. One caution worth carrying into procurement: certification is not the same as conformity with the Act, so a supplier waving a certificate has not answered the question. In the United States, NIST's AI Risk Management Framework is the voluntary reference, and is being revised.

— People: the J-curve is a people curve

- **Expect the dip.** Output falls while habits change, then recovers past where it started. Teams that are told this in advance get through it. Teams that are not conclude the tool failed.
- **Train by role.** Executives: this guide. Managers: how to run and measure an AI-assisted team. Engineers: the delegate loop. Business teams: assisting and reviewing. Reviewers: how to check AI work without rubber-stamping it.
- **Build a champions network** with real time allocated, open office hours and regular show-and-tell. Adoption spreads sideways between peers far faster than downward from memos.
- **Say the quiet part.** People want to know what this means for their jobs. DORA's 2026 guidance on returns is to reinvest the capacity you free, starting with the rework and backlog nobody could afford, not to lead with headcount cuts. Be specific about where roles will change.
- **Protect the apprenticeship, and know why.** Stanford's Digital Economy Lab, using payroll data through June 2026, found employment for workers aged 22 to 25 in the most AI-exposed occupations running about 19% below where it would be had it kept pace with less-exposed peers, widened from 15% a year earlier. The adjustment shows up as **reduced hiring, not as people being let go**. The same authors are careful to say they see no widespread, economy-wide displacement, and that these are descriptive patterns rather than proof of cause. Read together, that is the apprenticeship problem in one sentence: the easiest role to not fill is the one where people used to learn the job. Juniors still need to learn to read, debug and judge, and someone has to decide to keep hiring them.
- **Reward sharing.** Recognise the team that publishes its playbook, not only the team that ships.

— Value: measure at three levels

1. Activity

Are people using it? Weekly active use, tasks per person. A leading indicator and nothing more.

2. Flow

Is work moving faster and better? Lead time, review wait, cycle time per case, rework, change failure rate, quality scores.

3. Outcome

Does the business see it? Cost per case, turnaround compliance, revenue, customer and member satisfaction, capacity redeployed.

Remember the McKinsey numbers from the second stop: eight in ten people feel more productive, and 37% of companies see it in profit. The distance between those two figures is levels 2 and 3.

- **Baseline before you start.** No baseline, no proof.
- **Agree the measure with Finance first.** A saving Finance does not recognise is a story, not a result.
- **Count the full cost:** licences, tokens, platform, people, and the time spent reviewing. All five are opened up below.
- **Decide where freed capacity goes before it appears.** Otherwise it dissolves into slack and nobody can find it.
- **Report what did not work.** It buys you credibility for what did.

— What it costs: three on the invoice, two in the day

Five costs run under the whole line, and the two that never reach an invoice are the ones budgets miss. That is why a plan priced from vendor quotes comes in low, and why a programme that looked affordable starts to slip.

Licences



From Station 1. Per seat

Tokens



From Station 1. Per use, and it grows

People



From Station 1. Headcount and time

Review



From Station 2. Time, and it grows

Platform



From Station 3. Headcount

Cost	How it is charged	What makes it jump
Licences	Per seat, per month	Widening from engineers to everyone else at Station 3.
Tokens	Per use	Autonomy itself. A delegated task runs a loop and costs many times a single suggestion, and by Station 4 you are paying per case.
People	Headcount and time	Training every role, champions with real time allocated, and the J-curve dip while habits change.
Review	Time, never invoiced	The verification tax. It rises with everything the agents produce, and no vendor quotes it to you.
Platform	Headcount	The gateway, connectors and evaluation tooling. This is a product with a standing team, not a project with an end date.

Only tokens and review scale with the work. The other three are closer to fixed, which means the first agent looks expensive and the fifth looks cheap. Judge the platform against the portfolio, not against whichever workflow happened to go first.

In practice at Alderline

The intake agent's card caps it at \$0.40 and twelve tool calls per case. Against nine thousand requests a week that is a ceiling near \$3,600 a week, or about \$190,000 a year, and it is the only one of the five costs an agent card can cap.

The platform underneath it cost more than that and was funded at Station 3, before a single case ran. Alderline counts it once, against the portfolio. Sam's team reports spend per case each month beside routing agreement, so a model change that buys accuracy and doubles the cost is visible as a trade rather than a surprise.

Set against all of it, the return that is easiest to defend is not the hours saved on work you already do, because someone will always argue about whether those hours were real. It is the work that was never affordable. The migration at Station 2 is the clearest case: the prize was not the ten weeks, it was that upgrades, test coverage and dependency clean-up moved inside the budget for the first time.

— Platform: buy, build or blend

Buy what is a commodity: assistants, models, standard agents inside the software you already license. Build what is specific to you: context, connectors, evaluations, and the workflows that are your business. Keep models swappable. The boundary is moving: McKinsey's 2026 survey found that about a third of organisations have already declined to buy some software because agentic coding tools let them build it in-house.

Fund it as a portfolio. A reasonable starting split is 70% on proven productivity at Stations 1 to 3, 20% on workflow automation at Station 4, and 10% on frontier work toward Station 5. Review it every quarter and move money toward what the numbers support.



Ten ways enterprises stall

- Tool first, problem second.
- Licences without enablement.
- Measuring typing instead of delivery.
- No baseline, so no proof.
- Every team builds its own stack.
- Governance as a gate instead of a fast lane.
- Automating a broken process.
- Pilots with no owner and no path to production.
- Human review that is a rubber stamp.
- Silence about jobs.



The last stop, about 10 minutes

Your route

Find where your organisation is on the line today, see which track is holding the train back, and leave with one page for the leadership table.

Find your station

The first two questions place the train. The other four measure how far each track has been built. Remember the rule from the first stop: the train can only go as far as the shortest track.

Six questions, two minutes

Answer for the organisation on an ordinary week, not for its best team. In each group, tick the highest statement that is fully true and note its number. Your train is at the average of the first two numbers, rounded down. Each track is built to its own number, and any track below the train is where to start. The online guide does this arithmetic for you.

Where is AI doing real work today?

- ☐ 1. Individuals use an assistant in their editor or a chat window.
- ☐ 2. Engineers hand whole tasks to coding agents and review the pull requests.
- ☐ 3. AI runs inside team pipelines: review, testing, incident triage.
- ☐ 4. At least one agent runs a business workflow in production.
- ☐ 5. Several agents hand work to each other across functions.

How much does AI do before a person looks?

- ☐ 1. One suggestion at a time. A person accepts or rejects each one.
 - ☐ 2. A whole task. A person reviews the result before it counts.
 - ☐ 3. Pipeline steps run on their own. People own the gates.
 - ☐ 4. Whole cases from start to finish, inside written limits. People take the exceptions and audit samples.
 - ☐ 5. Agents give work to other agents. People set goals, budgets and limits.
-

Platform: what do teams build on?

- ☐ 1. An approved assistant with single sign-on, or less.
 - ☐ 2. Sandboxes and short-lived, scoped credentials for coding agents.
 - ☐ 3. One gateway for all model traffic, with shared connectors and evaluation tooling.
 - ☐ 4. A workflow engine with approval steps, a trace for every case, and evaluations in the release pipeline.
 - ☐ 5. An agent registry and control plane: identity, policy and tracing across agents.
-

Guardrails: how are limits set and enforced?

- ☐ 1. A use policy exists, or is being written.
- ☐ 2. The platform enforces agent permissions. No production credentials, and approval before destructive commands.
- ☐ 3. Use cases are sorted by risk with a fast lane, and every model call is logged.
- ☐ 4. Every agent has a written boundary, an audit trail, and a kill switch with a fallback.
- ☐ 5. Policy is enforced in code across all agents, and an agent incident has been rehearsed.

People: who knows how to work this way?

- ☐ 1. A few enthusiasts. Training is optional or self-serve.
- ☐ 2. Every engineer has had hands-on training, and each team has a champion.
- ☐ 3. A staffed platform team and enablement group. Product managers and analysts are trained too.
- ☐ 4. Every agent has a business owner. Reviewers are trained to check AI work. Roles are being redesigned.
- ☐ 5. Managers run teams of people and agents, and workforce planning covers both.

Value: what can you prove?

- ☐ 1. Licence counts and anecdotes.
- ☐ 2. Usage and developer experience, against a baseline taken before rollout.
- ☐ 3. Delivery measures for whole teams, and cost per unit of work.
- ☐ 4. Business results per workflow, agreed with Finance.
- ☐ 5. A portfolio view of value, cost and risk per agent, reviewed every quarter.

A first twelve months

A realistic pace for a large enterprise starting near Station 1. If the assessment put you further along, start there. The order matters more than the dates.



Months 1 to 3

Lay the track

Policy, approved tools, sign-on, telemetry. Capture the baseline. Train every engineer and name champions. Station 1 in every team.



Months 4 to 6

Delegate

Coding agents in sandboxes with scoped credentials. A context file in every active repository. One agent-led backlog project. Start the gateway.



Months 7 to 9

Integrate

All model traffic through the gateway. AI review and test generation in pipelines. Risk tiers with a fast lane. First business workflow scored, baselined and in shadow mode.



Months 10 to 12

Automate

The first workflow climbs the trust ladder on data. Agent cards and a registry begin. Results published with Finance. Second and third workflows chosen.

Station 5 is not on this plan, on purpose. You reach it by running Station 4 well, several times, on the platform you built at Station 3.

Five decisions for your next leadership meeting

1. **Who owns the line?** One accountable executive, with a platform lead and an enablement lead beside them.
2. **Which measures has Finance agreed to, and where is the baseline?** Settle this before the next rollout, because a baseline cannot be captured afterwards.
3. **Will every model call go through one gateway?** A yes is the cheapest governance you will ever buy.
4. **Which workflow goes first, and who is its business owner?** Score three candidates. Choose the forgiving one over the famous one.
5. **Where does freed capacity go, and what do we tell people about their jobs?** Decide both before the capacity appears.

The engineers who build it need more than this guide. Build the Line (<https://agentikai.varasrinivas.com/courses/autonomy-line-course/>), the practitioner course, takes the same Alderline workflow up the same five stations and builds the controls that earn each step: nine modules, about seven hours, nothing to install.

The Autonomy Line

The route card

Enterprise AI adoption from coding tools to agentic AI, in five stations. Autonomy is earned: the train can only go as far as the shortest track.

Station		Build on the four tracks	Move on when	Mind the gap
1	Assist AI suggests. People do the work. People act as authors	- One approved assistant, single sign-on, telemetry - One-page policy, simple data rules, secret scanning - Hands-on training, a champion in every team - A baseline captured before rollout	Weekly use is the norm, by telemetry. Policy and logging are in place. You hold a baseline Finance would accept.	Shelfware. Counting lines of code.
2	Delegate People hand over whole tasks and review the result. People act as directors and reviewers	- Sandboxes, short-lived scoped credentials, spend by team - No production credentials. Approval before destructive commands - The delegate loop. Seniors paired with juniors - Review wait, rework, change failure rate, cost per change	Agent pull requests merge routinely and the failure rate holds. Every active repository has a context file and tests people trust.	Review becomes the bottleneck. Code nobody understands.
3	Integrate AI joins the team's pipeline, on a shared platform. People act as system designers	- One gateway, connectors, search, evaluation tooling - Risk tiers with a fast lane. Every model call logged - Platform team and enablement group staffed - Team delivery measures, platform reuse, cost per unit of work	All model traffic goes through the gateway. Whole-team delivery is improving. A new agent reaches a safe pilot in days.	Every team builds its own stack. Governance as a gate.
4	Automate Agents run bounded workflows. People handle exceptions. People act as supervisors	- Workflow engine, write access behind approvals, a trace per case - Written boundary, audit trail, kill switch with a fallback - A business owner for every agent. Trained reviewers - Process baseline. Cost per case, cycle time, override rate	Two or three workflows run in production with results Finance agrees. Promotion and demotion are driven by data.	Automating a broken process. Rubber-stamp review.
5	Orchestrate Many agents coordinate. People govern outcomes. People act as governors	- Agent registry, an identity per agent, tracing across agents - Policy enforced in code. Limits on how far a failure spreads - Managers of people and agents. Workforce planning for both - Value, cost and risk per agent, reviewed quarterly	Running well when every agent is registered with an owner, and any agent can be paused in under a minute.	Multi-agent because it sounds advanced. Unchecked hand-offs.

Three rules for the whole line. Use the least autonomy that solves the problem. An instruction is not a control. The train can only go as far as the shortest track.

Words worth knowing

A2A

Agent2Agent. An open protocol that lets agents from different vendors find each other and hand over work.

Agent

Software that uses a model in a loop. It plans, uses tools, checks the result and continues until the goal is met or it has to ask.

Agentic AI

The broader term for systems built from one or more agents that act, where a chatbot only answers.

Context file

A plain-text file in a repository, often named AGENTS.md, that tells every agent how this codebase works and what it must not touch.

Evaluation, or eval

A scored test set for AI behaviour. The unit test of agents.

Gateway

One front door for every model call: identity, policy, logging, data redaction, spend limits and routing.

Guardrail

A limit enforced by the system, as opposed to one requested in a prompt.

Human in the loop, human on the loop

In the loop: a person approves each action. On the loop: the agent acts, and a person monitors, samples and can stop it.

LLM

Large language model. The reasoning engine inside assistants and agents.

MCP

Model Context Protocol. The open standard for connecting agents to tools and data.

Prompt injection

Instructions hidden in content an agent reads, written by someone who wants to hijack it.

RAG

Retrieval-augmented generation. The system looks up relevant documents first, and the model answers from them.

Shadow mode

The agent runs on real work, and its output is compared with what people did and used for nothing else.

Token

The unit a model reads, writes and bills by. Roughly three-quarters of an English word.

Sources

Published figures in this guide come from the sources below, checked in September 2026. Alderline Health and everything measured there is illustrative.

1. DORA, State of AI-assisted Software Development (<https://blog.google/technology/developers/dora-report-2025/>), 2025.
2. DORA, report on the return on investment of AI-assisted software development (<https://dora.dev/ai/roi/report/>), 2026.
3. METR, randomized trial of early-2025 AI tools with experienced open-source developers (<https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/>), July 2025.
4. METR, update on developer productivity uplift (<https://metr.org/blog/2026-02-24-uplift-update/>), February 2026.
5. McKinsey, The State of AI (<https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>), August 2026.
6. Gartner, prediction on cancelled agentic AI projects (<https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027>), press release, June 2025.
7. Linux Foundation, formation of the Agentic AI Foundation (<https://www.linuxfoundation.org/press/linux-foundation-announces-the-formation-of-the-agentic-ai-foundation>), December 2025.
8. Bain & Company, Your AI Budget Is Growing. Your Returns Aren't. (<https://www.bain.com/insights/your-ai-budget-is-growing-your-returns-arent-heres-why/>), June 2026.
9. US Census Bureau, Business Trends and Outlook Survey on AI use (<https://www.census.gov/library/stories/2026/05/ai-use-businesses.html>), May 2026.
10. Stanford Digital Economy Lab, Canaries in the Coal Mine? (<https://digitaleconomy.stanford.edu/news/canariesaug26/>), August 2026.
11. NIST Center for AI Standards and Innovation, insights from a large-scale agent red-teaming competition (<https://www.nist.gov/blogs/caisi-research-blog/insights-ai-agent-security-large-scale-red-teaming-competition>), March 2026.
12. European Commission, the AI Act regulatory framework (<https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>) and guidelines on transparency obligations (<https://digital-strategy.ec.europa.eu/en/policies/guidelines-ai-transparency-obligations>).
13. European Union, Product Liability Directive (EU) 2024/2853 (<https://eur-lex.europa.eu/eli/dir/2024/2853/oj/eng>).
14. Model Context Protocol, Enterprise-Managed Authorization (<https://blog.modelcontextprotocol.io/posts/enterprise-managed-auth/>), June 2026.
15. US Centers for Medicare and Medicaid Services, Interoperability and Prior Authorization Final Rule (CMS-0057-F).

16. eWeek, report on the July 2025 coding agent that deleted a production database (<https://www.eweek.com/news/replit-ai-coding-assistant-failure/>).
17. Klarna, press release on its AI assistant (<https://www.klarna.com/international/press/klarna-ai-assistant-handles-two-thirds-of-customer-service-chats-in-its-first-month/>), February 2024, and Bloomberg, report on its return to human customer service (<https://www.bloomberg.com/news/articles/2025-05-08/klarna-turns-from-ai-to-real-person-customer-service>), May 2025.
18. Moffatt v. Air Canada, British Columbia Civil Resolution Tribunal, 2024.

End of the line

The tools will change every few months. The discipline will not. Grant autonomy in proportion to your controls and your evidence, build all four tracks, and use the least autonomy that solves the problem. Take the route card to your next leadership meeting and start with the five decisions.

Alderline Health and the people in it are fictional. Their numbers illustrate a pattern and are not benchmarks. Published statistics are attributed where they appear and listed under Sources. This guide is general education, not legal, clinical or regulatory advice.

The guide is a single file. It works offline and stores nothing.