Dillic

Planning and change management, in one product

Define the objective, measure where you stand, find out why, choose among the solutions on the table, and run the change through to release - on one set of figures.

PRODUCT DEMO

THE PROBLEM

Planning and delivery are measured in different systems

A strategy deck says one thing, the board says another, and nobody can show the line between them. The result is familiar: initiatives that ship without moving anything, and no way to prove it either way.

The plan is an opinion

Current state is asserted in a workshop rather than measured, so nobody can say whether it changed.

The work is disconnected

The backlog knows nothing about the objective it serves, so priority is argued rather than derived.

The report is a retelling

Progress is written by hand from whichever tool was open, and the audit trail is assembled afterwards.

WHAT IT IS

Two halves, one set of figures

Both halves ship in this release, and both are reported on - with AI analysis over the same numbers rather than a separate narrative.

PLANNING

Measure, explain, choose

A configurable entity hierarchy, people and per-entity roles, factors and objectives, the current state assessment and its 2×2 matrix, weighted composite roll-ups, root cause analysis, and desired future state comparison.

CHANGE MANAGEMENT

Govern, deliver, evidence

A configurable workflow engine with approval gates, Scrum and Kanban boards, backlog and sprints with a burndown, ITIL 4 change enablement with a calendar and the DORA figures, version-control webhooks, and sync with fifty-three trackers.

Beside the two halves: a service desk, adoption, a portfolio, SAFe for those who run it, Pages, enterprise identity, and a phone app.

> The join is the point: a root cause links to the initiatives that address it, an initiative answers for whether the factor it targeted moved, and the service desk reports back. The plan, the delivery, and the desk name the same services, so each half checks the other.

PRODUCT GOALS

What the product is for

- Align strategic objectives with **measurable** outcomes.
- Support evaluation of **current and target states** across organizational entities.
- Enable **root cause analysis** and solution comparison.
- Support change execution using **ITIL-aligned, agile-ready** workflows.
- Provide **governance, traceability, reporting**, and AI-assisted analysis.

TARGET CUSTOMERS

Who it is for

Multi-entity organisations

Companies with hierarchical structures, where a division's picture has to roll up into the company's without being flattened.

Teams that need structure

Groups that want planning, prioritisation, and change management to be one process instead of three tools.

Compliance-facing businesses

Organisations requiring enterprise identity integration and audit-ready evidence of who decided what, and when.

Licensing. A free 10-user starter plan - same email domain, 2GB, no AI, and up to 100 survey takers from outside per survey. Paid tiers add storage and AI credits per user, and up to 1,000 survey takers from outside per survey plus 100 per user at no extra charge. The launch price is $7, $8, or $9 per user per month, kept by every organisation that subscribes at it; later it is $12, $13, or $14. The beta is free.

THE DEMO

Northwind Logistics

Everything that follows is one seeded organisation. Responses and history are generated from a **fixed random seed**, so the matrix and the burndown look the same on every machine.

3
divisions over twelve teams, weighted 45 / 35 / 20

62
people across four roles

50
accepted factors, assessed at a baseline and again mid-year

13
initiatives in three programmes, and 165 work items on 15 boards

Planning

Twenty-four assessments: a baseline and a mid-year current-state assessment (a CSA) for each division, weighted composites at the company, eight competing desired future state assessments (DFSAs), and three root cause analyses.

Delivery

Customer Operations runs Scrum - three sprints closed, one active with a burndown. Technology runs Kanban with work-in-progress limits. Nine ITIL change requests over eighteen services, and a service desk with three queues.

PART ONE

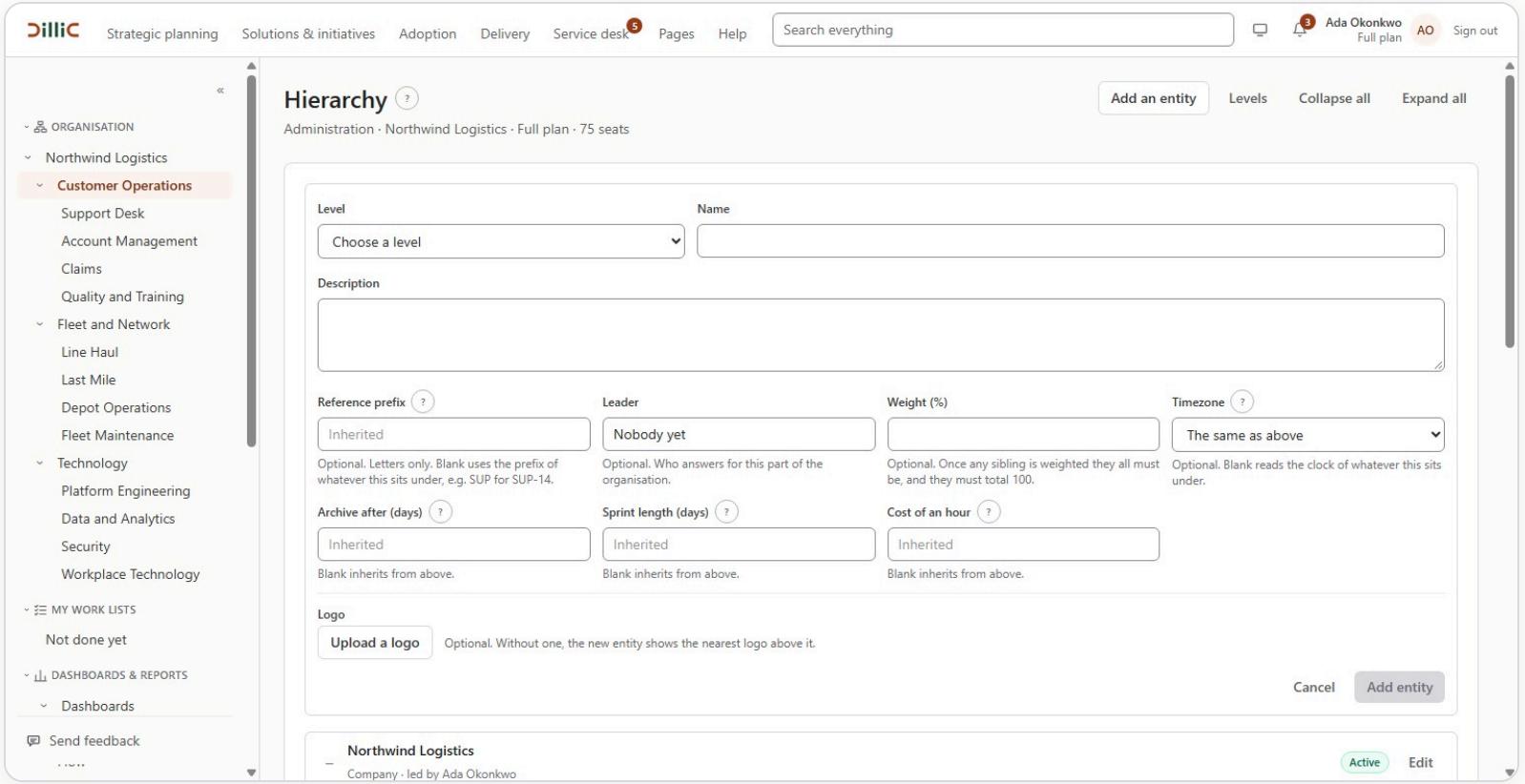
Planning

The product in the order it is meant to be used: hierarchy, factors, assessment, results, composite, root cause, and future state.

Levels, then the entities in them

Company → Division → Team is configuration, not a fixed shape. An account defines its own ladder.

- A level has **at most one parent** level.
- An entity may sit only beneath an entity of its level's parent.
- Sibling weights must total **100%**.



ORGANISATIONAL MODEL

Weight is what makes a roll-up honest

A division that is 45% of the company should move the company's score by 45% of its own change - not by a third because there happen to be three divisions.

Declared, not inferred

Sibling weights are set by an administrator and validated to total 100%. An entity with no weight declared contributes equally, and says so.

Access inherits downward only

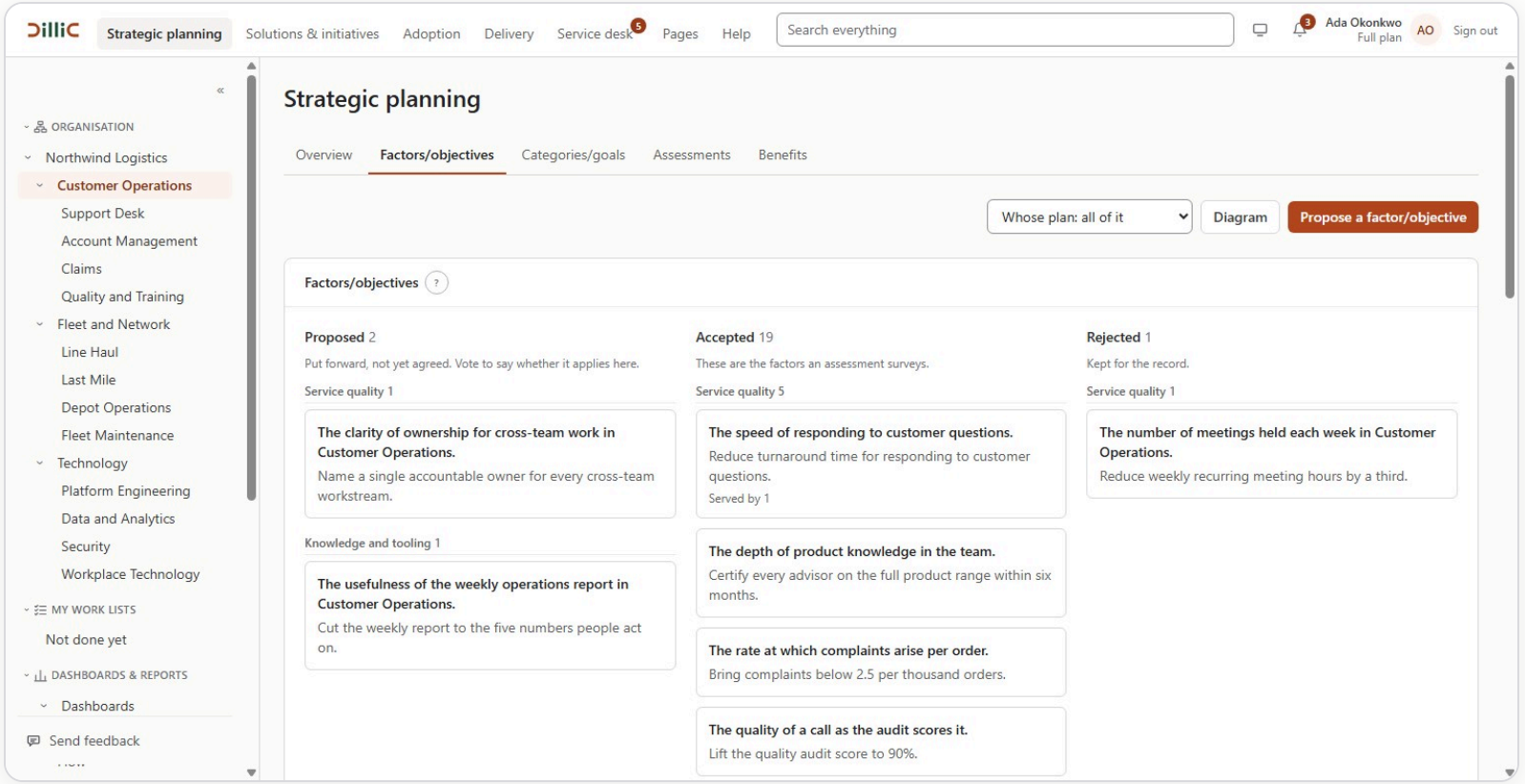
A role held on a parent entity applies to every descendant. Resolution walks upward from the target and unions what it finds - so a role on a team confers nothing on its division.

Terminology is account configuration. Organisations name things differently; renaming a term in Admin changes what the whole product calls it.

A factor is worded so it can be scored

The factor is neutral. Its objective states the SMART target. They are one record, because one without the other is not actionable.

- Propose, vote, refine, accept, or reject.
- Grouped and statused, so a proposal is visibly not yet a measurement.



CHANGE HISTORY

Rewording a factor asks you what kind of change it is

Because that is the question that decides whether past assessments stay comparable - and it is a judgement the product cannot make for you.

COSMETIC

History continues

A typo or a clearer phrasing. Respondents were effectively asked the same question, so the series is unbroken.

PARTIAL RESET

History is marked

The meaning shifted. Earlier results remain readable but are flagged as measuring something slightly different.

FULL RESET

The chain breaks

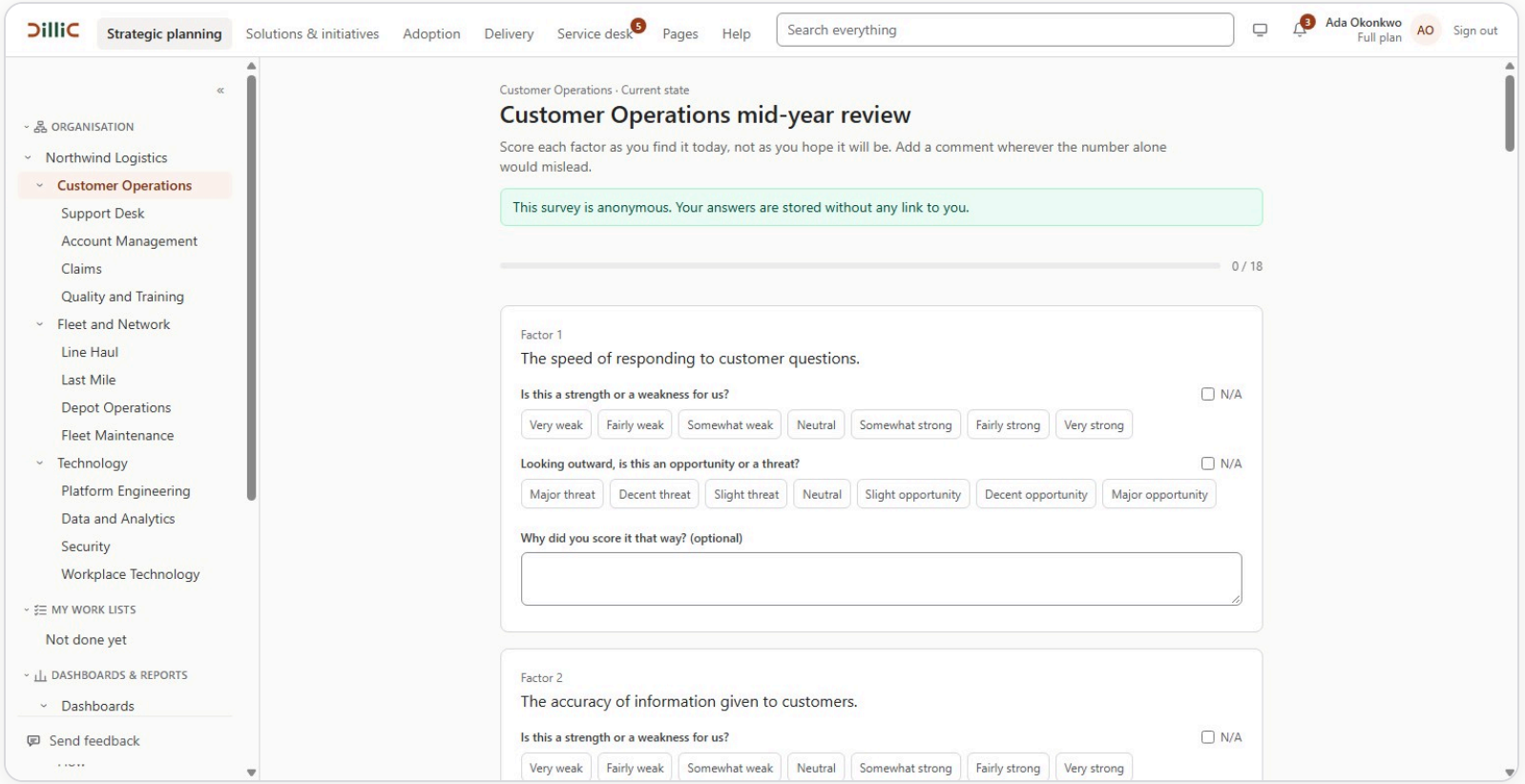
A new question. Prior assessments are no longer comparable, and the product refuses to pretend otherwise.

Statements are snapshotted into an assessment. A later reword cannot retroactively change what respondents were asked.

Accepted factors freeze into a survey

Two –3…3 questions per factor - one for strength, one for opportunity - plus a comment.

- The words on each scale are **editable**.
- Neutral can be switched off.
- Anonymity is chosen up front and **shown to respondents**.



A GUARANTEE, NOT A SETTING

Anonymity is structural

On an anonymous assessment the participant id is **never written** to the response row. There is no column to subpoena and no join to reconstruct - no one, not even an administrator, can recover it later.

Completion is still tracked

On the participant record, not the response. Response rate is not attribution, so an organiser can chase a survey without being able to read an individual's answers.

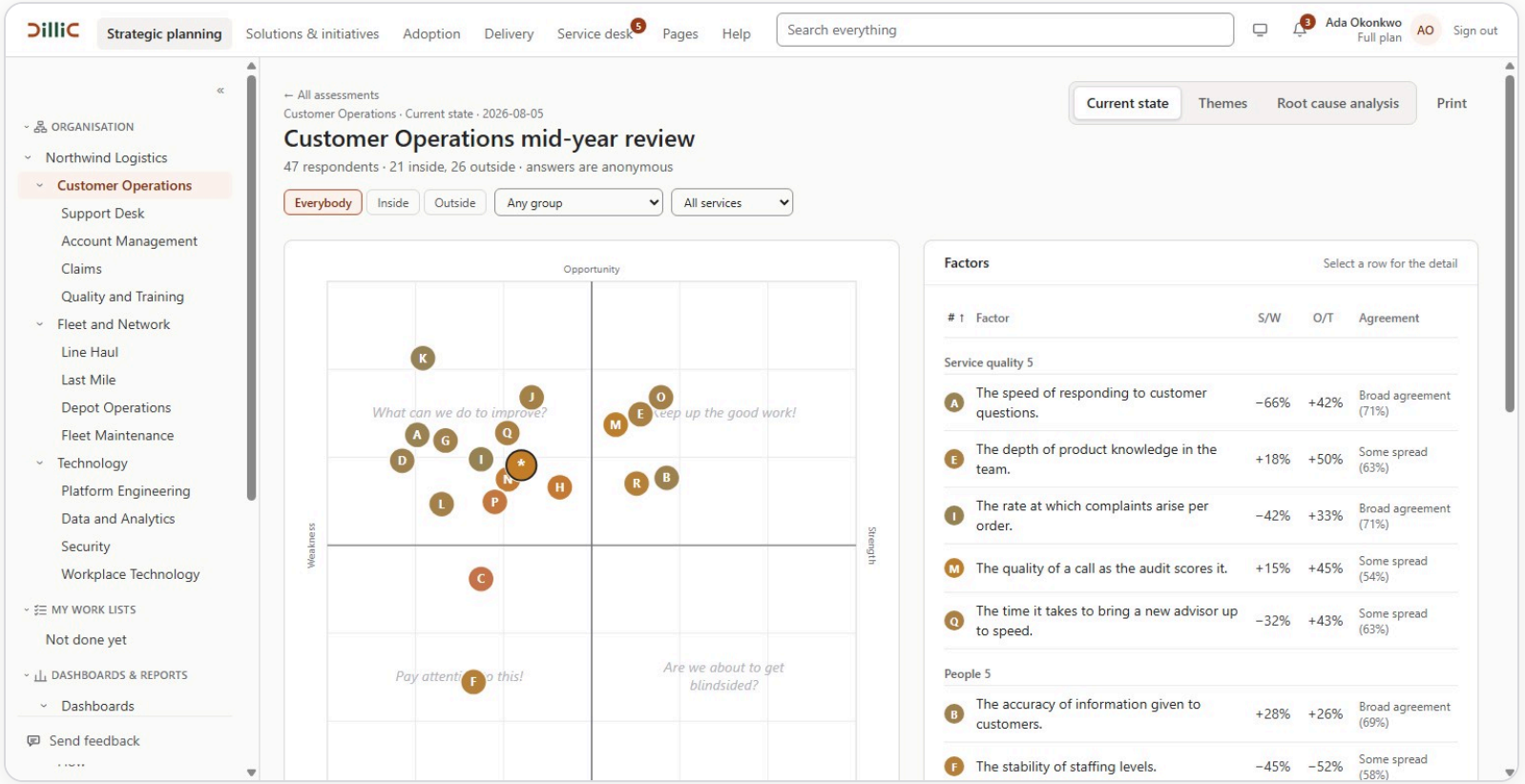
There is a test for this

Its own test in the suite pins the guarantee, so a later schema change cannot quietly regress it.

The matrix

Weakness left, strength right, threat below, opportunity above. Every factor is a lettered circle; the combined score is marked ∗.

- Colour is how much the room **agreed**.
- Every value the matrix encodes as position or colour is also readable as a number in the table beside it.



HOW TO READ IT

Four quadrants, one question each

UPPER LEFT · WEAK, BUT RISING

A weakness sitting in opportunity. The cheapest ground to take - the room already believes it can move.

UPPER RIGHT · STRONG AND RISING

Working, and expected to keep working. Protect it; do not spend the next quarter here.

LOWER LEFT · WEAK AND THREATENED

The urgent quadrant. Weak today and getting worse - the case a root cause analysis is for.

LOWER RIGHT · STRONG BUT THREATENED

A strength being eroded. Easy to miss, because today's number still looks healthy.

COLOUR CODING

The colour is disagreement, and it is never the only clue

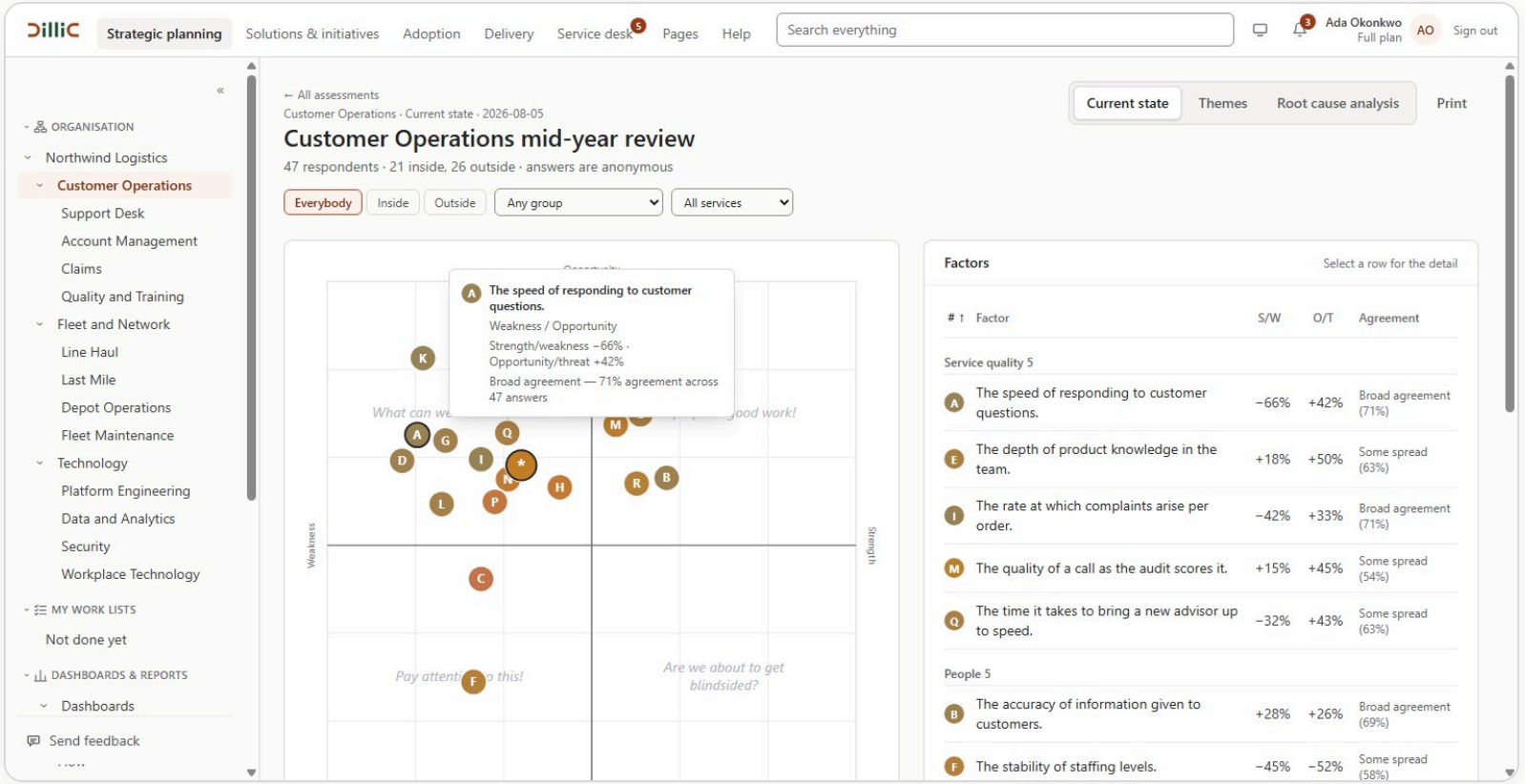


- **No red/green pair.** The ramp runs blue-green → amber → pink, so it survives the common colour vision deficiencies.
- **Amber, not yellow.** Pure yellow measures 1.6:1 against the page - a mid-dispersion circle would all but vanish. All three anchors clear 3:1.
- **Read by hue, not brightness.** The anchors sit at roughly equal lightness, so no step drops out. Validator: adjacent CVD ΔE 11.5 protan / 11.7 tritan, normal-vision ΔE 21.6.
- **Redundantly encoded.** Every mark carries its letter, and the table prints the dispersion as a number and as words.

Click a mark for what is under it

Its two scores, then **More** for the distributions and the comments.

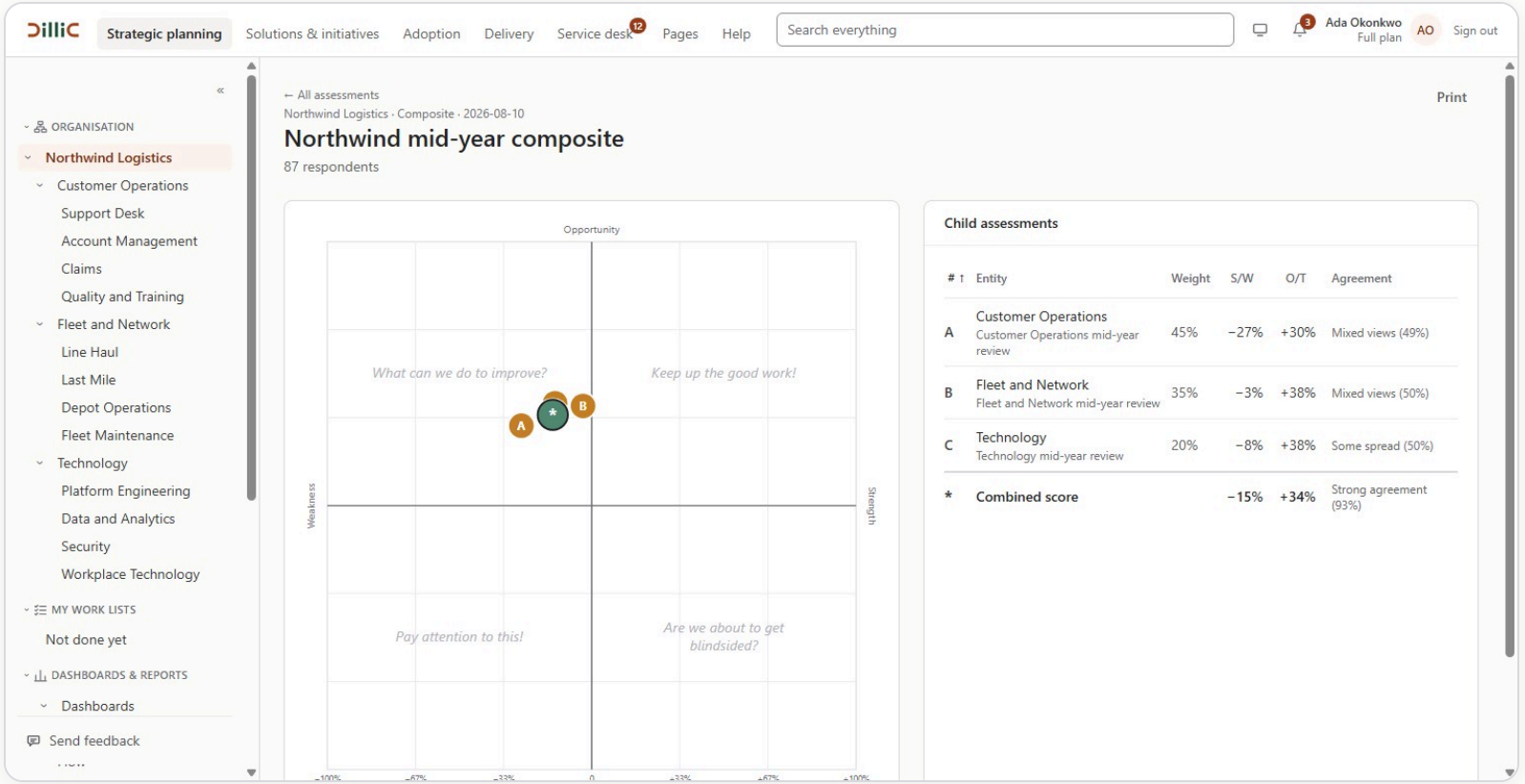
- A mean of zero can be a room that agrees on neutral, or a room split in half. The distribution tells them apart.
- Organisers group comments by theme.



Roll divisions up into the company

At a parent entity, combine child assessments taken around the same time.

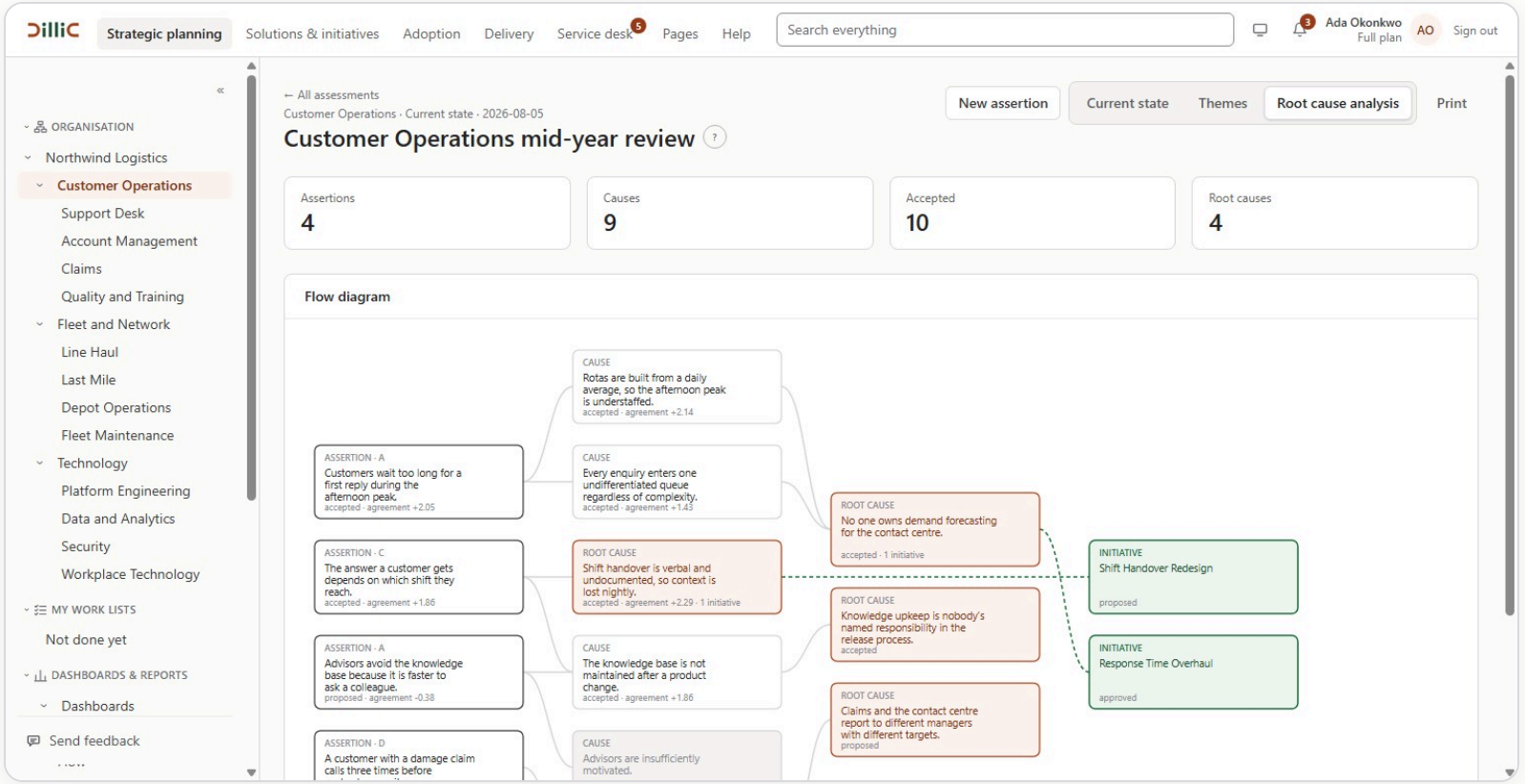
- Each child contributes its own ∗ as a single mark.
- Weighted by that child entity's share.
- Recurses to parent-of-parent.



Ask why until nothing is left underneath

Assert something about a factor, survey agreement on it, accept or reject it - then ask why beneath each accepted node.

- A node with nothing under it is a **root cause**.
- Root causes link to the initiatives that address them.



RELATIONSHIP RULES

Only an assertion may attach to a factor

A neutrally worded factor has nothing to explain. Something has to be **claimed** about it first - and that claim is what starts the analysis.

ASSERTION

A claim about a factor

Surveyed for agreement, then accepted or rejected. Accepting it earns the right to ask why.

CAUSE

An answer to a why

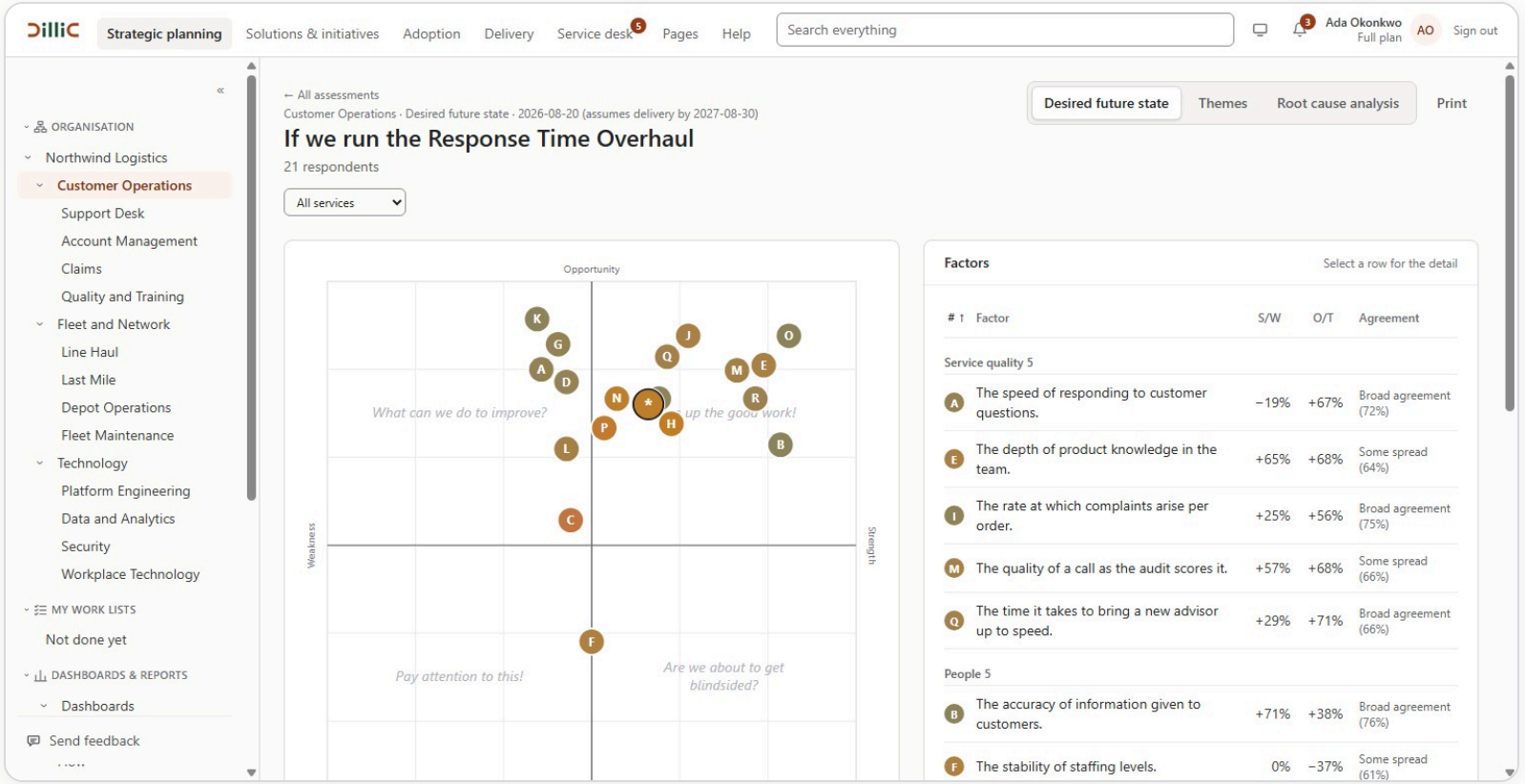
Sits beneath an accepted node. Never attaches to a factor directly, because it explains the assertion, not the wording.

> Both live in **one table**, separated by a type field - as the spec defines them. The separation exists for the rule above.

The same factors, answered about a future

The same matrix, answered as though the chosen initiatives had been delivered.

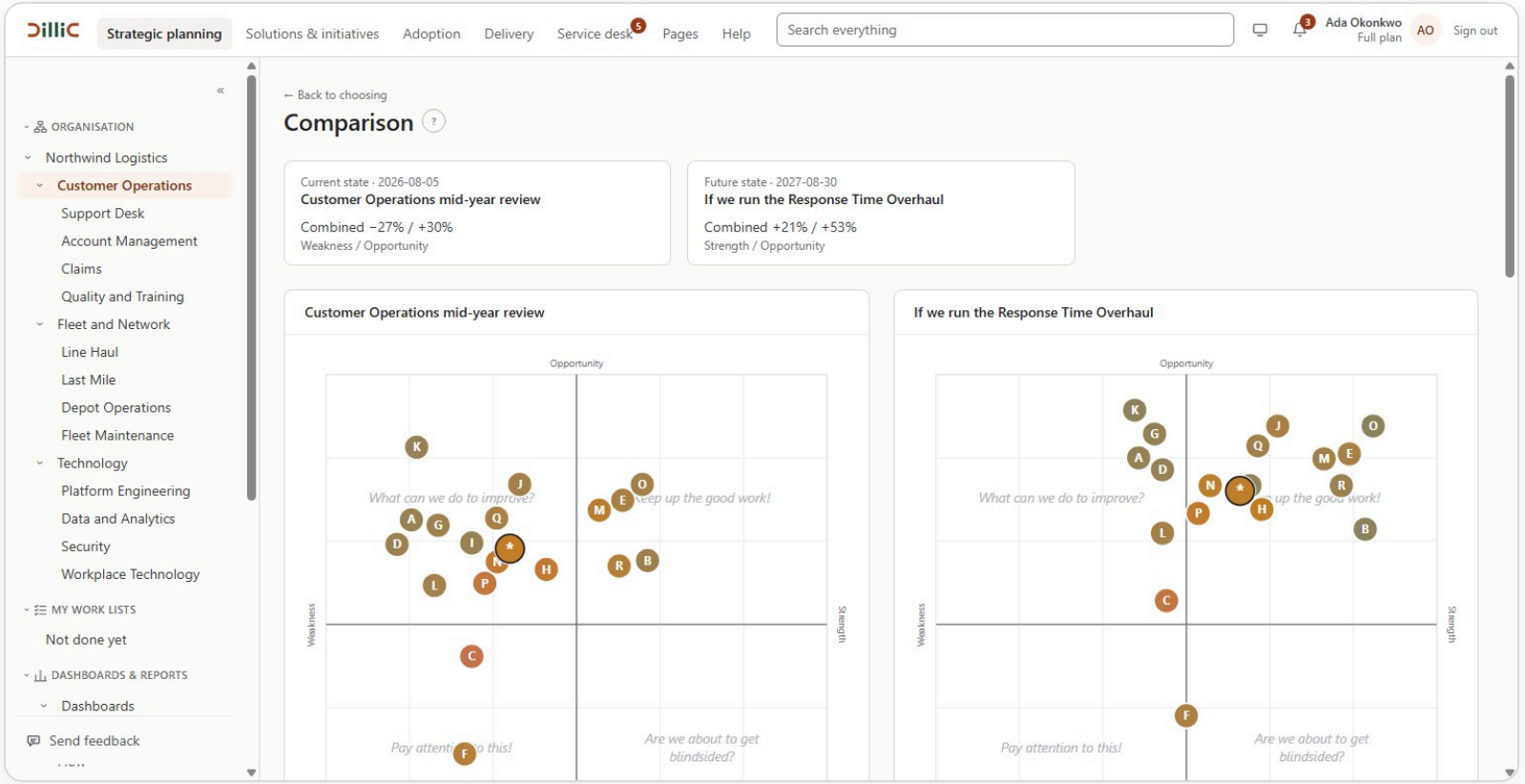
- A desired future state assessment and the solution it assesses are **one record** - the way a factor and its objective are.
- One initiative and the solution *is* that initiative. Two or more and it is a grouping with a name of its own.



Two questions, one screen

Compare **one desired future against another** to choose between solutions on the table.

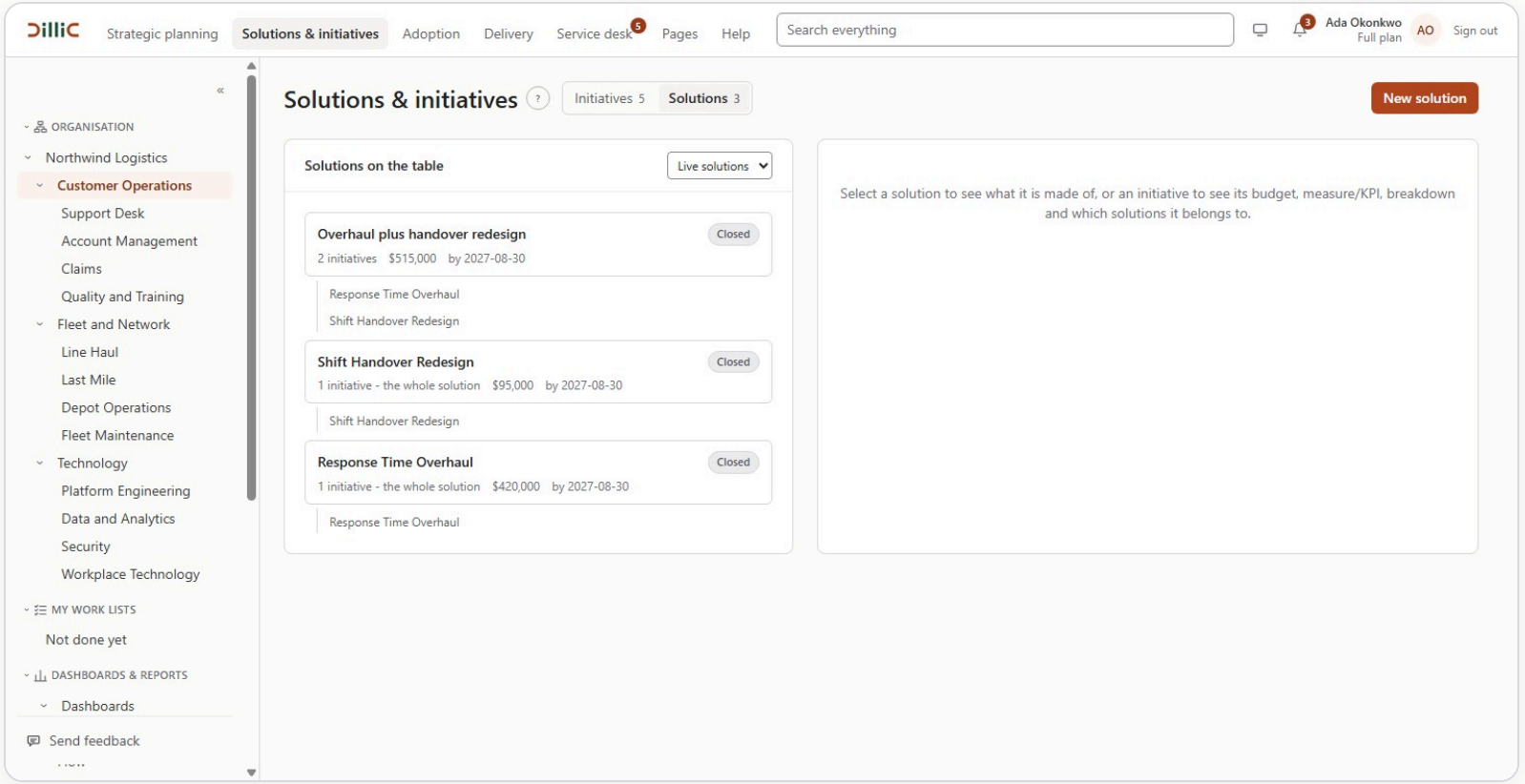
- Or compare the current state against a **later current state** to see whether anything moved.
- Same factors, same axes - so the delta is a measurement, not a retelling.



What is on the table, and what it is made of

Each solution opens to the initiatives that make it up.

- Each initiative shows which solutions have **claimed** it - or that none has yet.
- An unclaimed initiative is the interesting one: work with no solution answering for it.



PART TWO

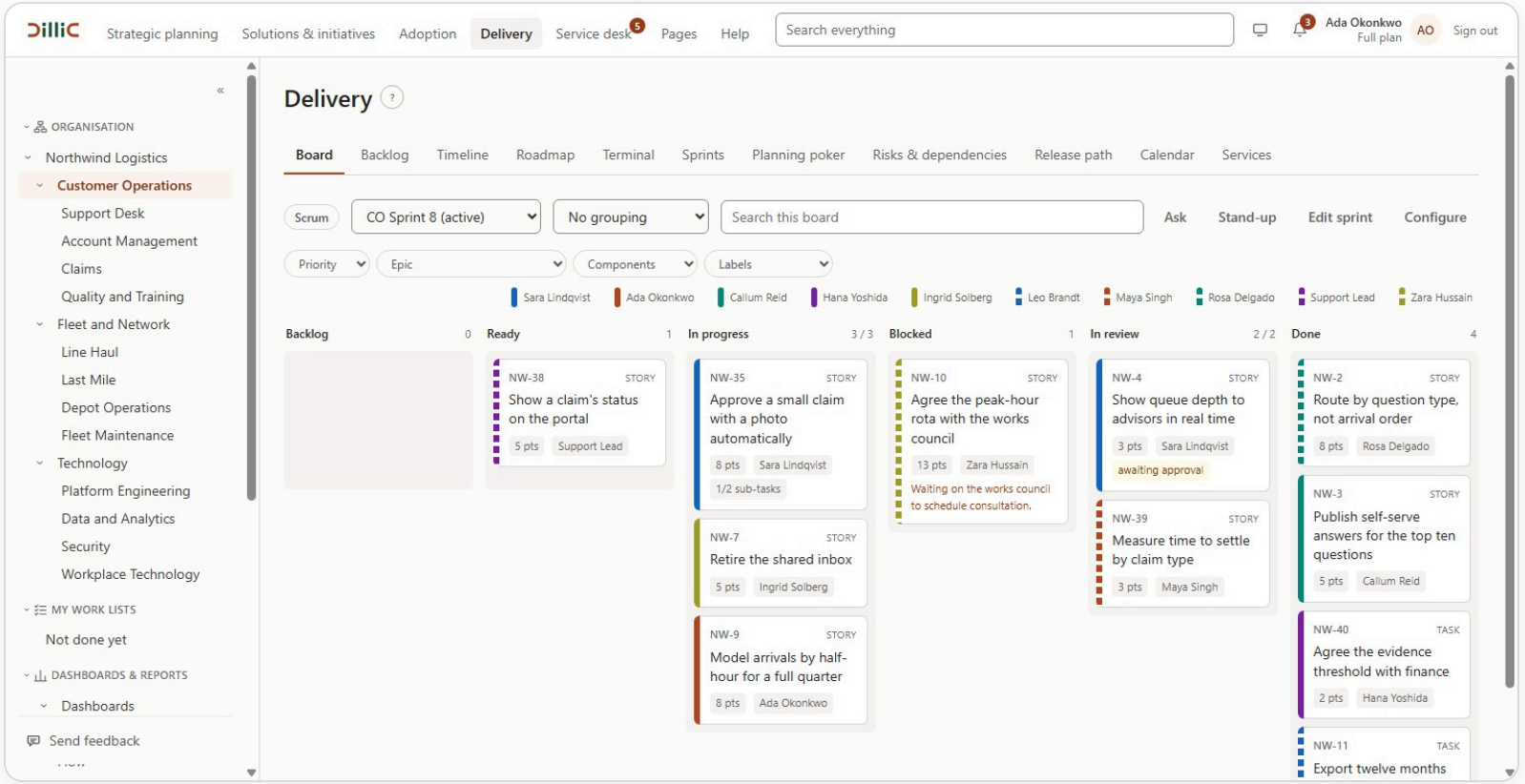
Change management

One workflow engine, configured three ways: Scrum on one entity, Kanban on the next, and ITIL 4 change enablement over both.

The board

Customer Operations runs Scrum: three sprints closed, one active.

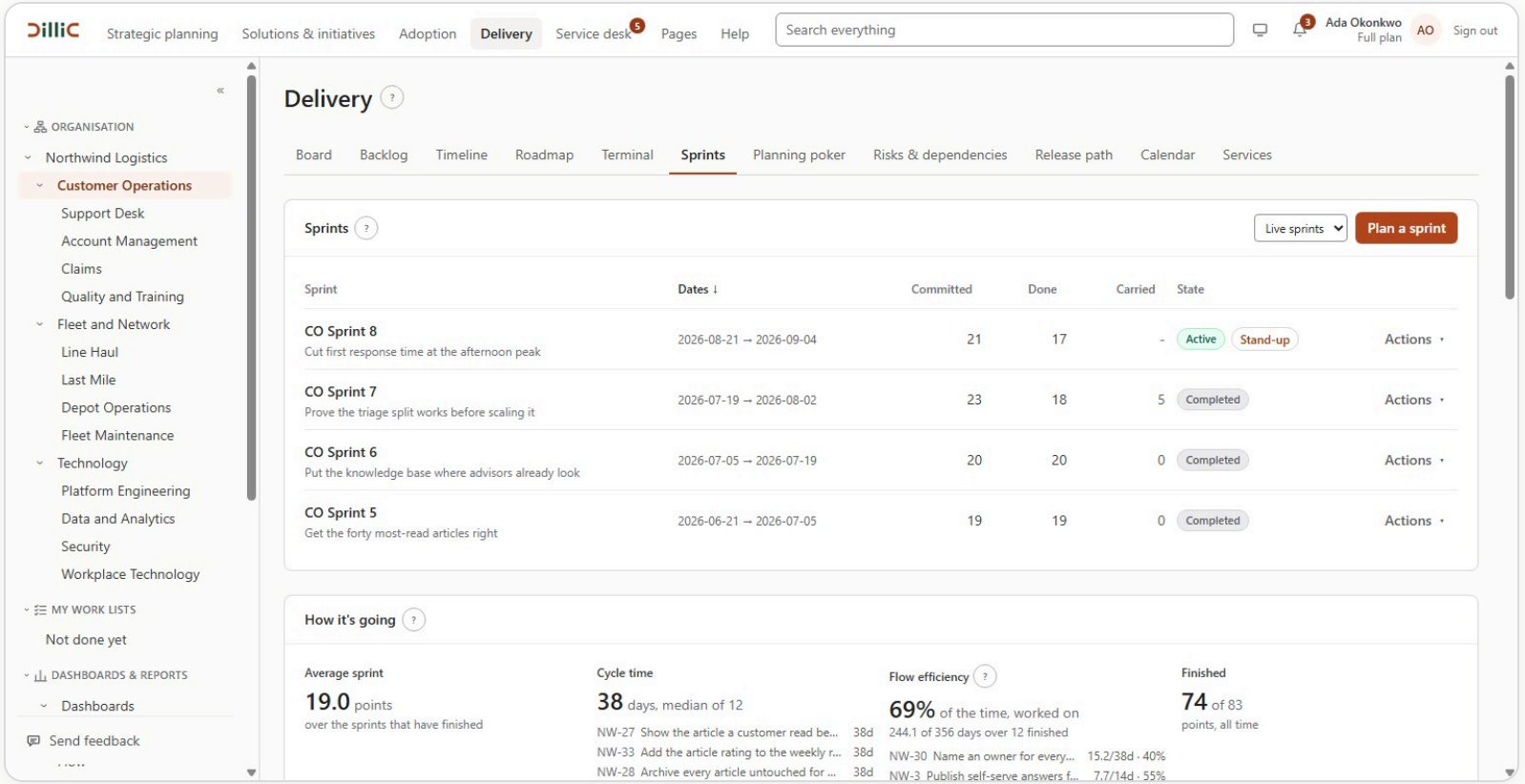
- Moving a story to Done needs a sign-off from **somebody other than its author**.
- Moves go through the transition menu, so a gated move can explain itself before it is attempted.



Sprints, and how it is going

Every transition writes an event carrying the estimate **at that moment**.

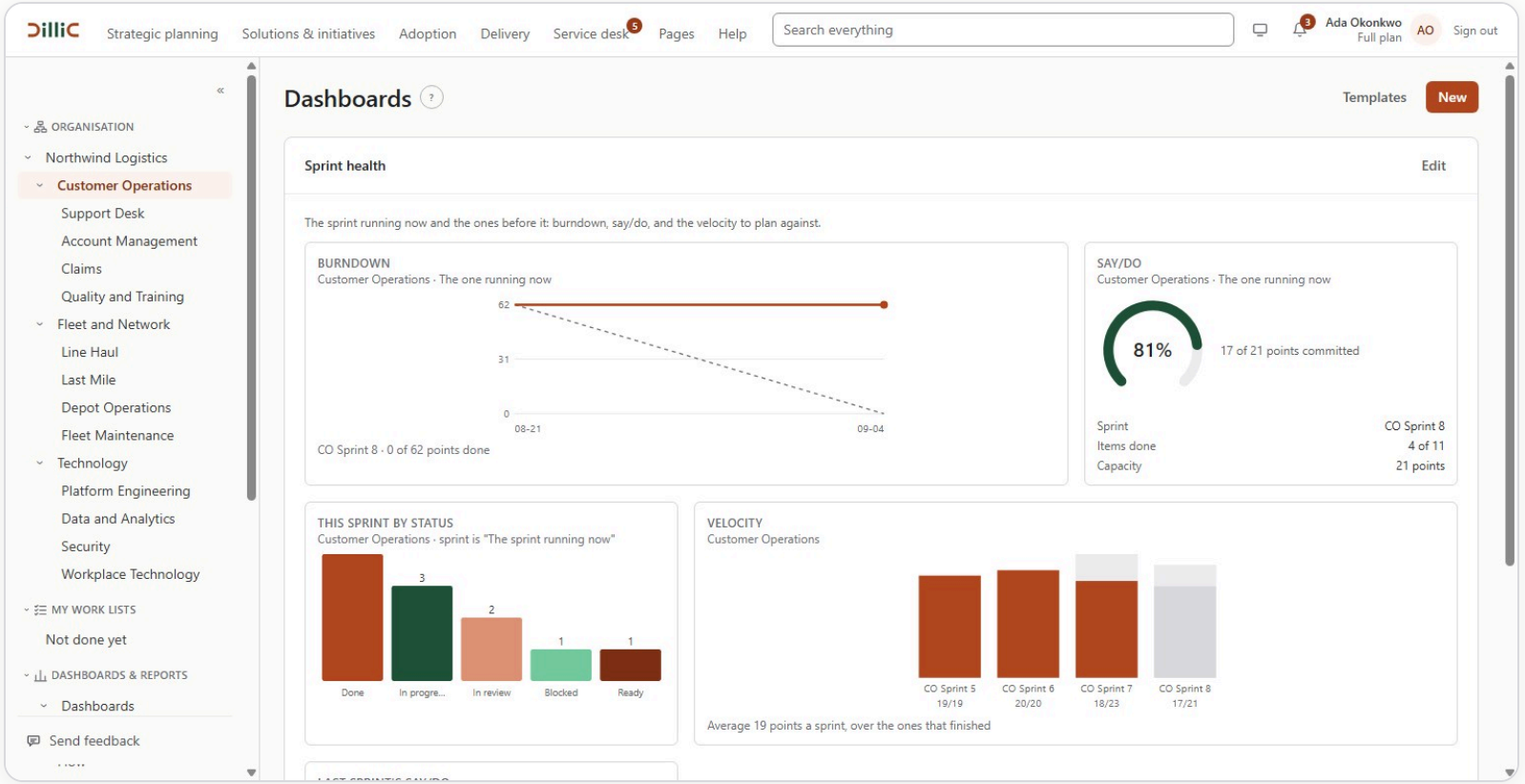
- So re-estimating a story today cannot rewrite last sprint's chart.
- Completing a sprint carries unfinished work back and records what was done.



A page of pictures

Half a dozen questions at once, across as many teams as you like, each one a query - and **most of them drawn**, because a figure has to be read and a shape is seen.

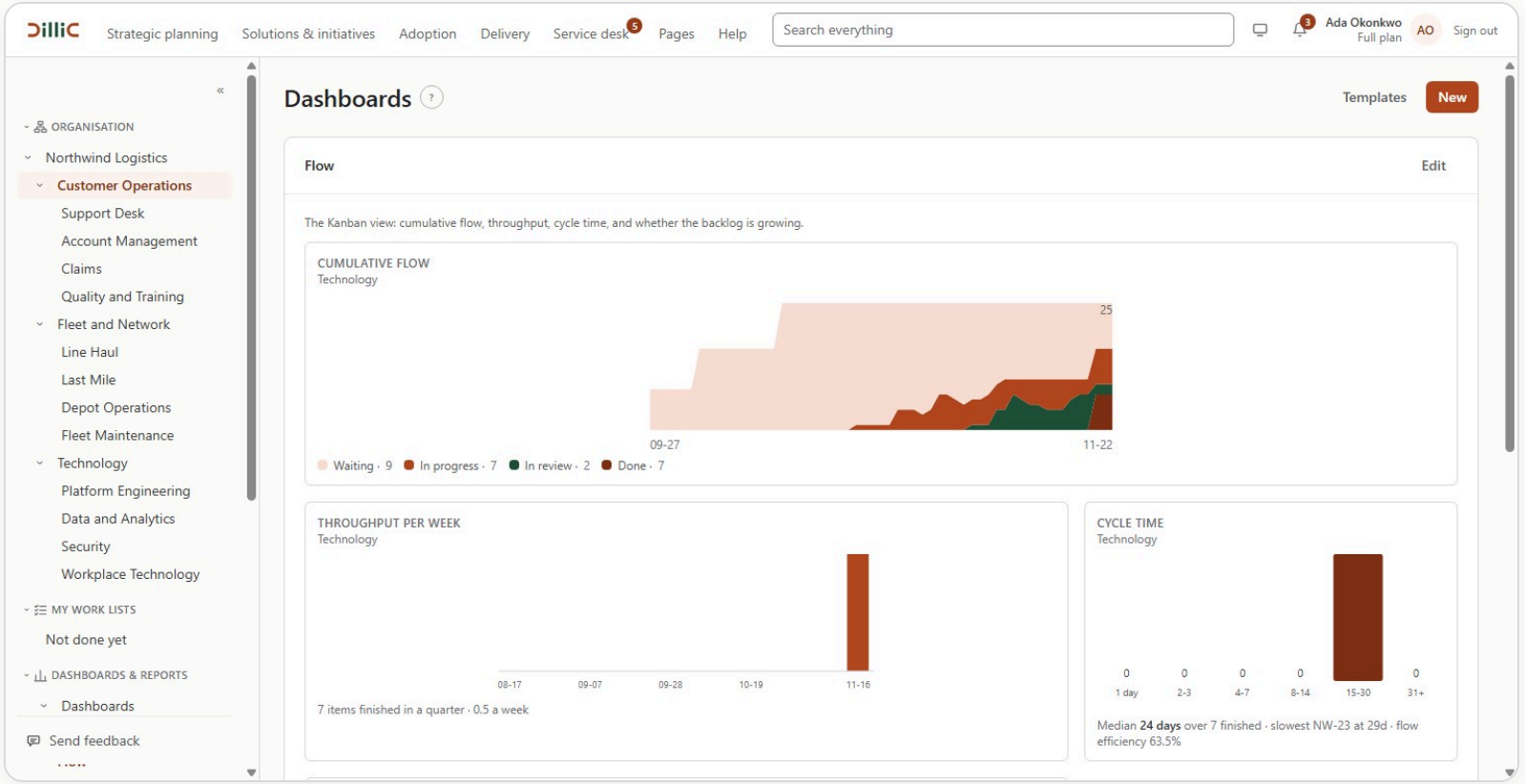
- A breakdown as bars, columns, a donut, or one stacked bar; a say/do as a gauge; a burndown and a velocity run that are the sprint's own.
- Every gadget is answered against **whoever is reading**, so a shared page shows two people two correct sets of figures.



Flow, without a framework

Five graphs of one question each about how work moves, read from the event log over a window that **slides with today**.

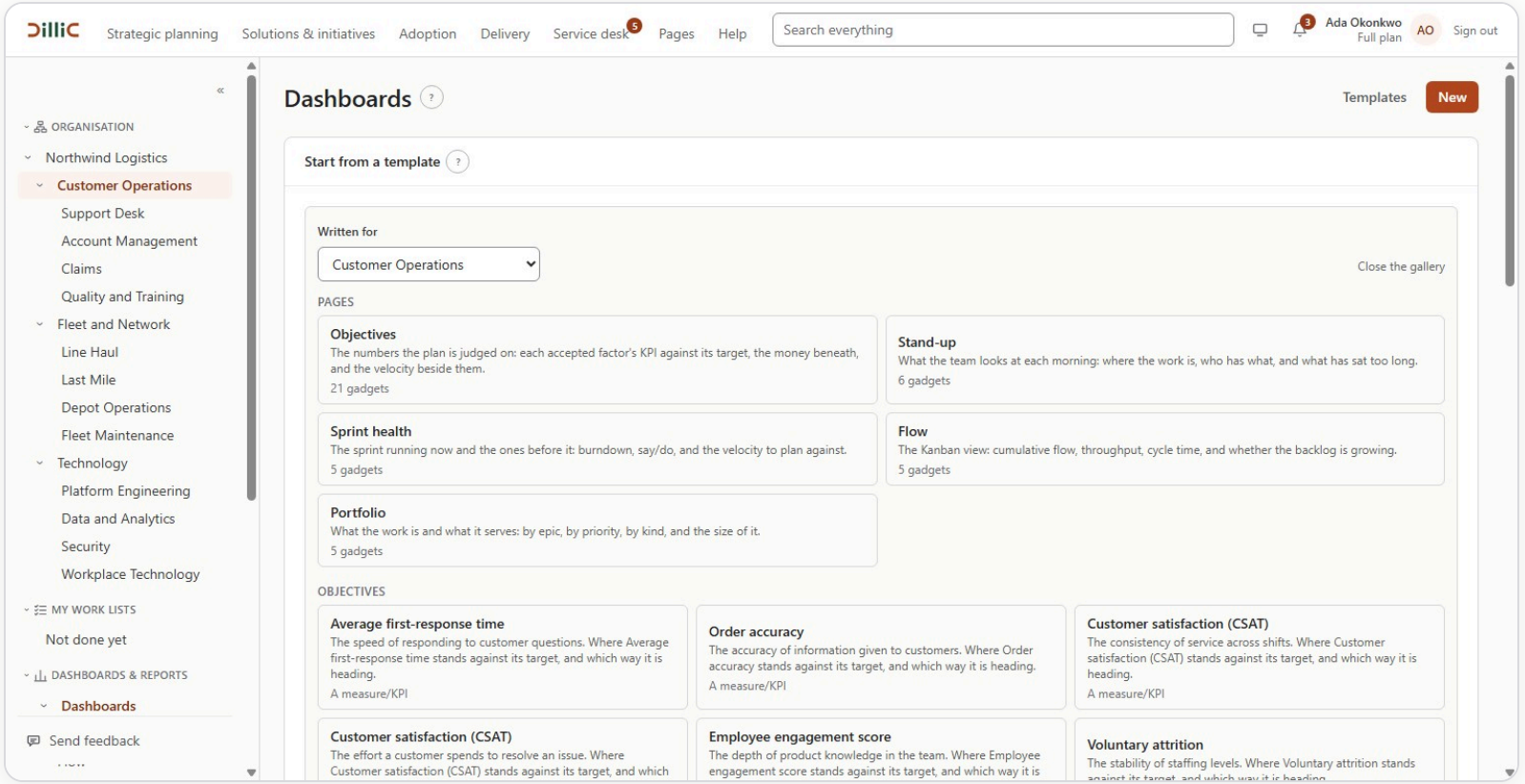
- Cumulative flow, throughput per week, a cycle-time histogram with its median, the aging of what is in flight, and arrivals against departures.
- Each still takes a question, so **"throughput of defects"** is a gadget rather than a report.



Start from a template

Twenty-one gadgets and one for each KPI, on five shelves, and five pages made of them - objectives, stand-up, sprint health, flow, and portfolio - written in **the board's own status names**.

- Resolved per board on the server, since only it knows what a board calls Done; a template the board cannot be asked is left off its shelf.
- A page from a template opens in the editor rather than saving, so you can review it first.



THE CORE DECISION

The workflow is data, not code

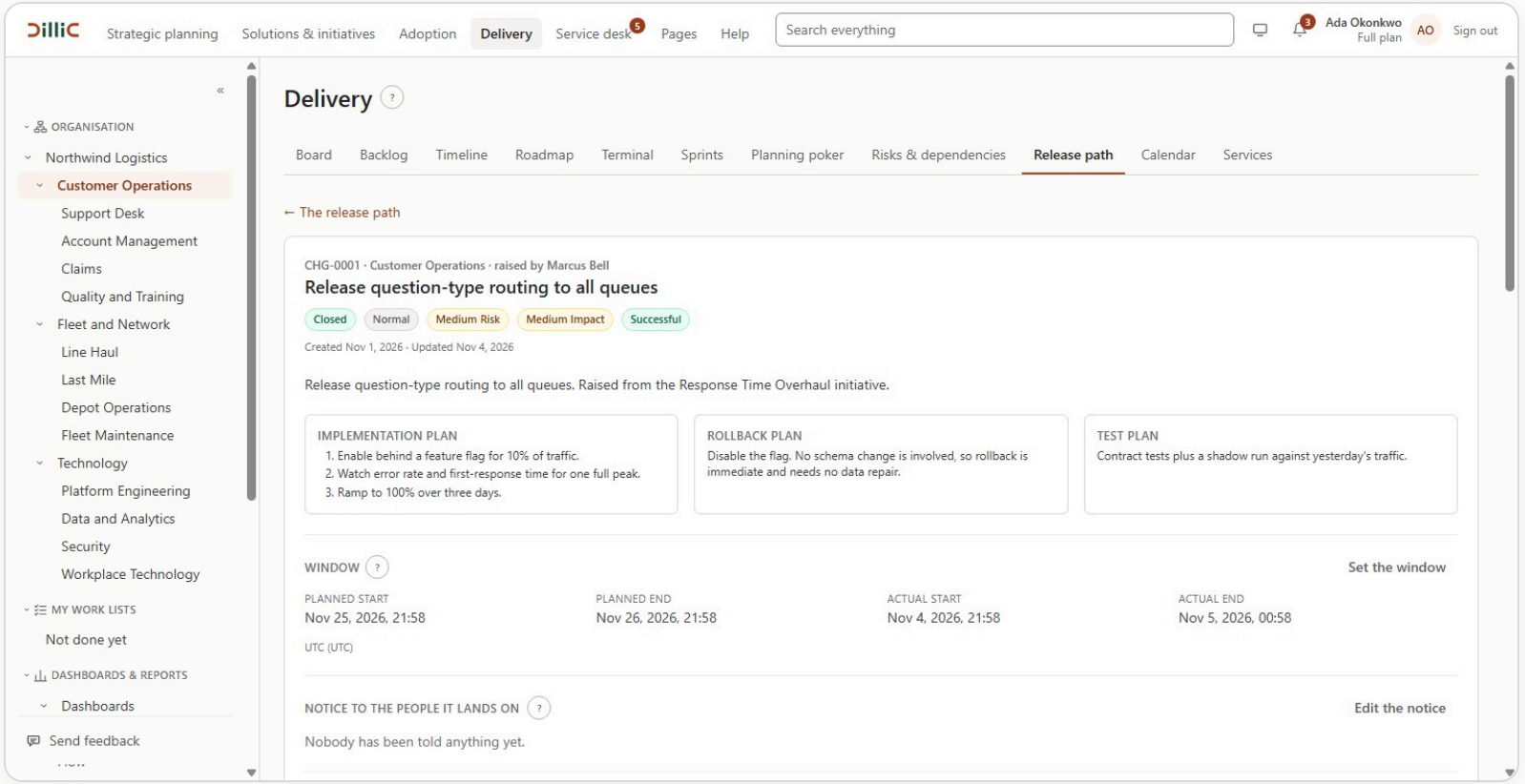
Board columns, the permitted moves between them, and the approvals those moves require are all **account rows**. Scrum, Kanban, and ITIL change enablement are three configurations of one engine, not three code paths.

- There is no `if methodology == "kanban"` anywhere in the codebase.
- There is **no "set the state" operation**. The only way across is a declared transition - so a gate cannot be routed around.
- **One engine, two subjects.** A work item and a change request are both a thing in a state with moves out of it, so one service drives both. An approval gate written once serves a sprint story and a production release.
- **Every status is one of seven states** - Backlog, Ready, In progress, Waiting, Being reviewed, Complete, and Cancelled - so an organisation can run **several workflows**, each board picks one, and reports still count across all of them by state.

Change enablement, end to end

Assess risk and impact, write the implementation and rollback plans, take it to the Change Advisory Board, schedule, implement, and record what happened.

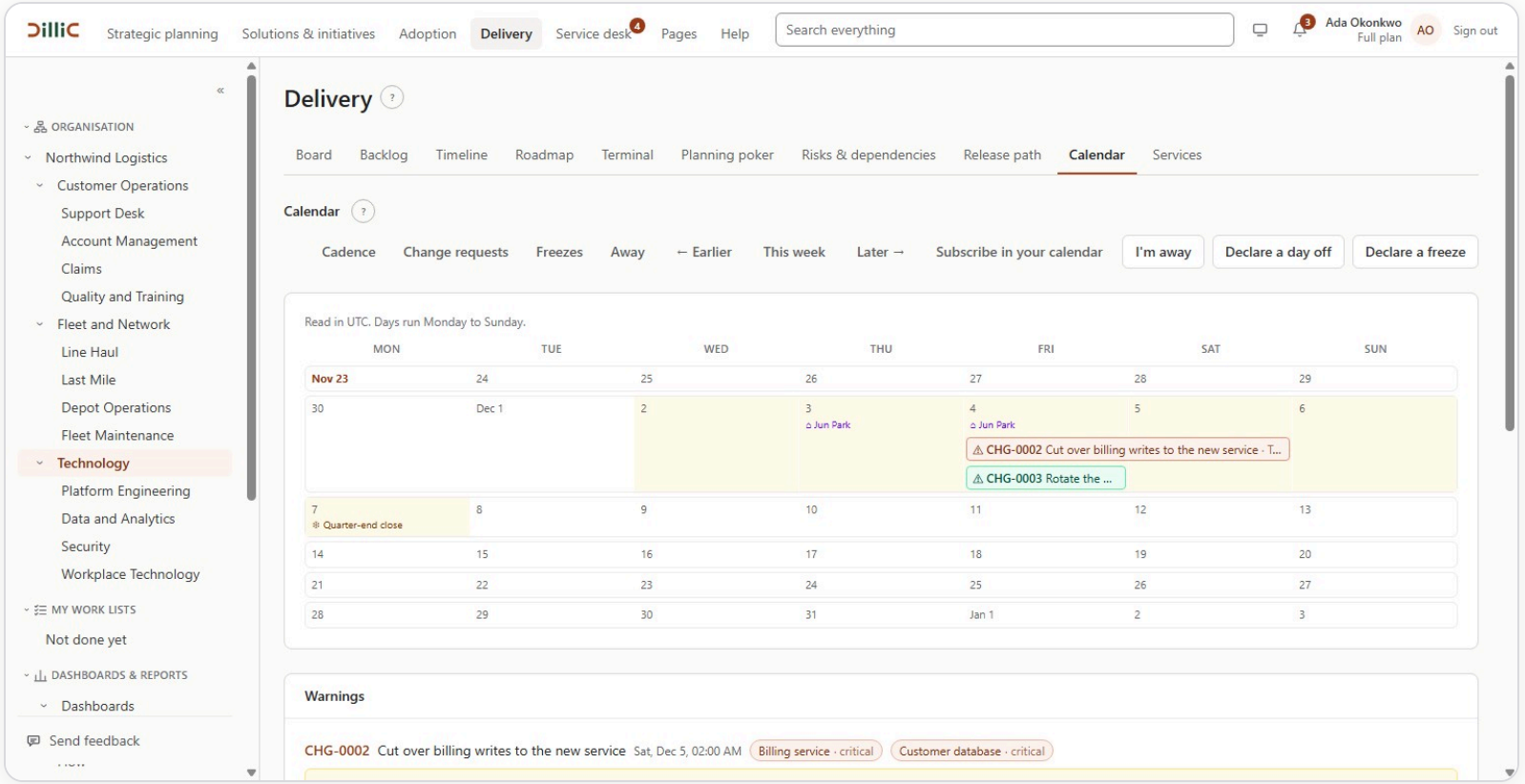
- A **standard** change is pre-authorised and skips the board.
- A **normal** one cannot - and authorisation without plans is refused.



A calendar that warns, and tells

Changes sit on a calendar with the services they touch. A collision on a service, or a window inside a freeze, is a warning **before it is a problem**.

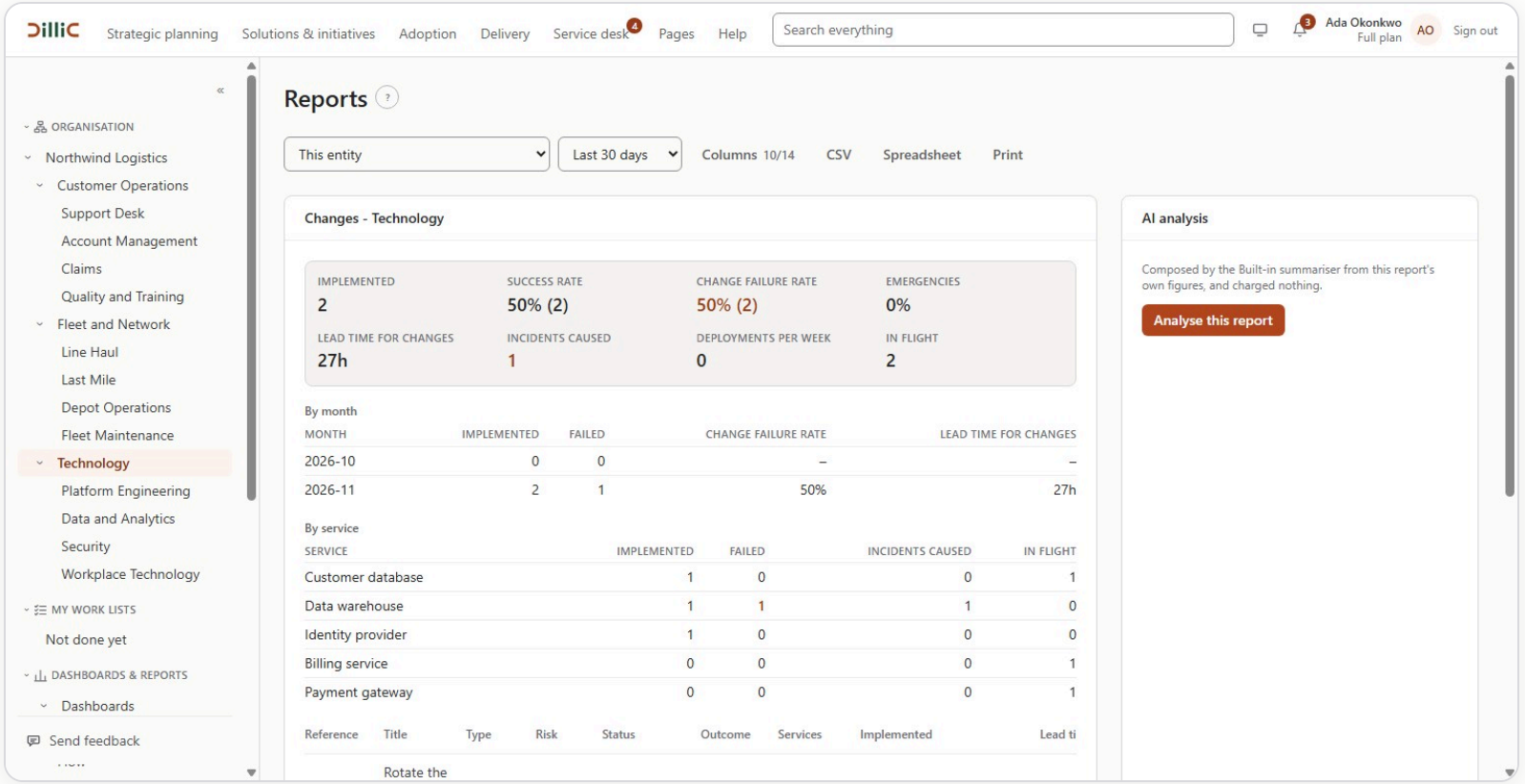
- A change model holds the plans, services, and risk answers for a kind of change you repeat; a standard model is the pre-authorisation.
- Each person can subscribe from Outlook or Google Calendar, and an announced change appears on the portal and on the service's page.



Deployments and the DORA figures

A release path names versions and the change requests that cover them, and deployments arrive from your pipeline.

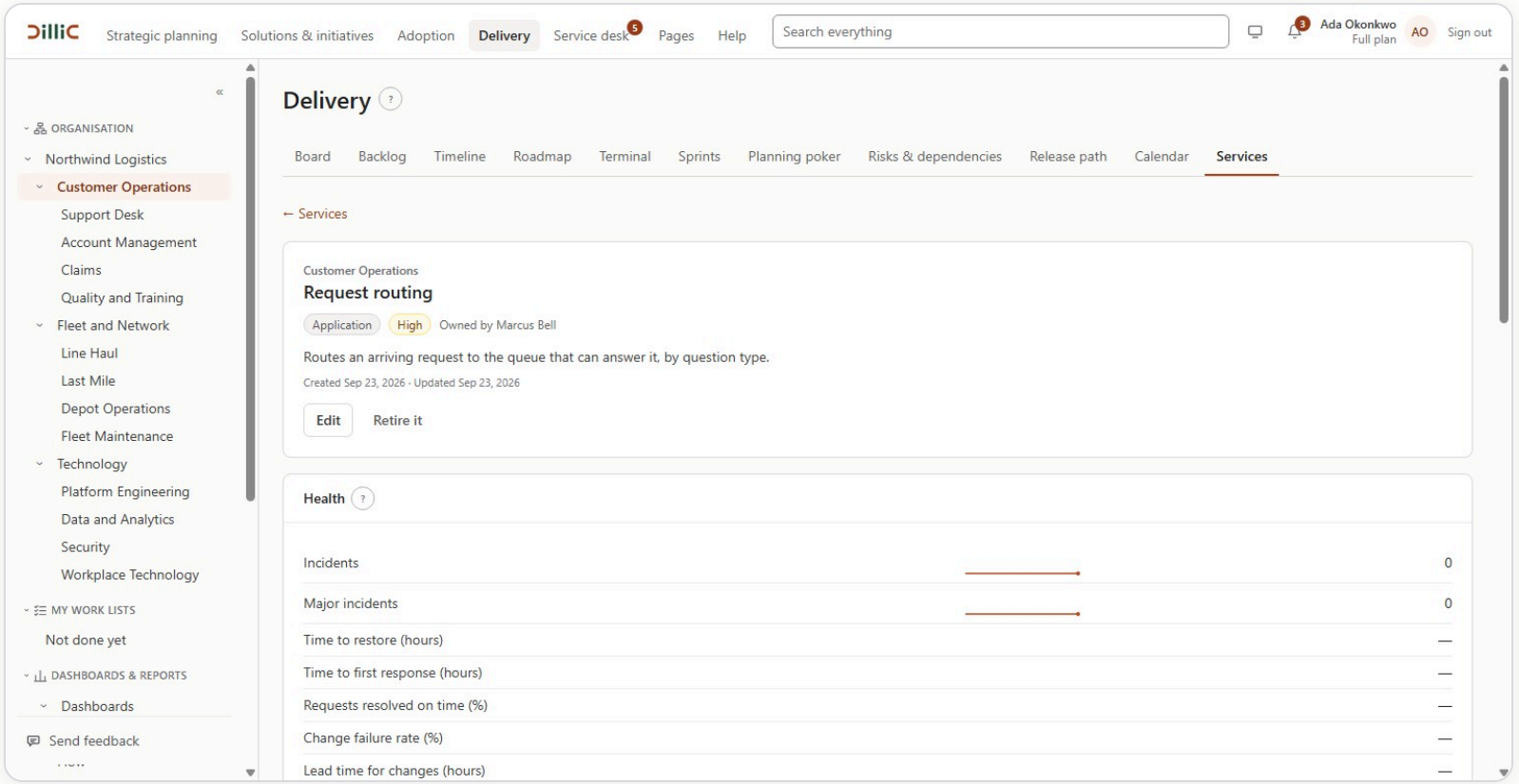
- Lead time, change failure rate, deployments per week, and incidents caused, **by month and by service**.
- The incidents caused are the desk's own, each naming the change behind it.



The loop closes on a service

A factor names the services it evaluates, an initiative names the services it establishes, enhances, or retires, a change names the services it touches, and an incident names the services it was about. One name, so the plan, the delivery, and the desk check each other. A part that runs several services may plan each on its own.

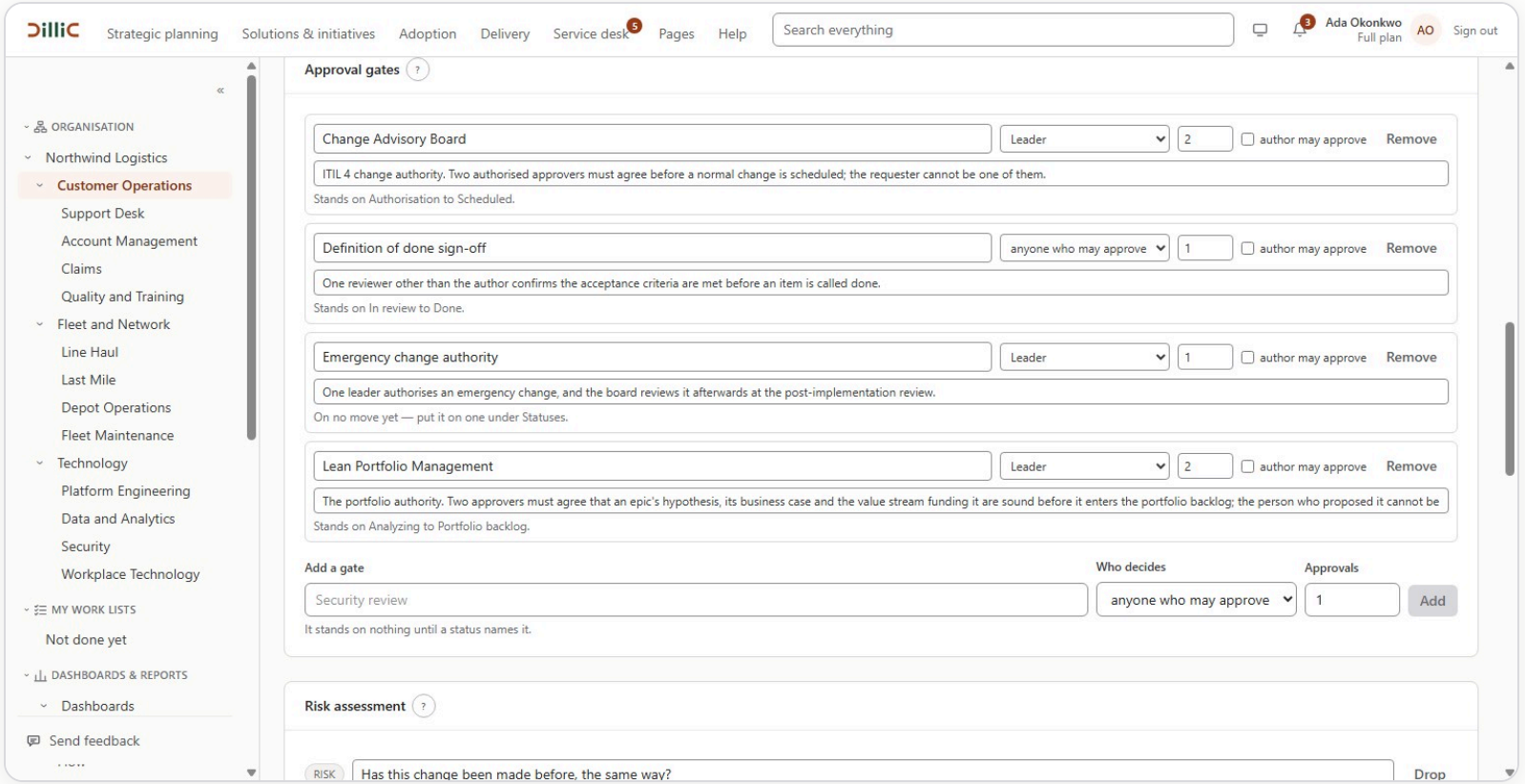
- A measure **reads its figures from the desk** - incidents, hours to restore, changes on time - and realigning a factor rewrites its readings.
- An initiative whose releases touched none of its services **says so**; a service taking changes that serve no initiative is unplanned.
- A root cause **cites the desk's incidents** as evidence, and the citation says what to assess next time.



A gate is configuration

Who decides, how many of them, and whether the author may be one - set once, and stood on a move between statuses.

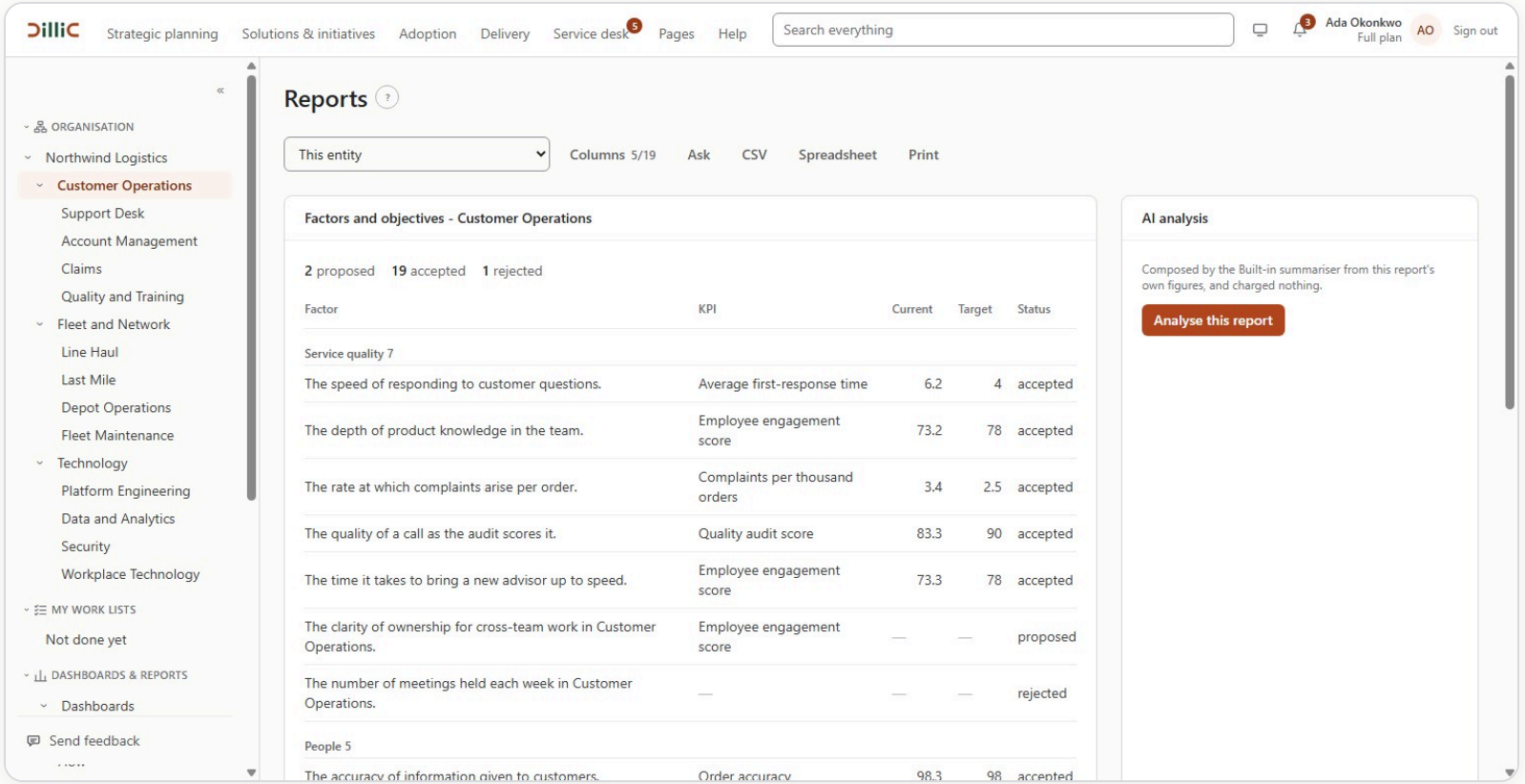
- An author cannot approve their own work, and a rejection stops the move.
- The item stays put when a gate blocks - it does not half-move.



One per step, printable

Factors, assessment, root cause analysis, sprint, and initiative - each with AI analysis beside it.

- The analysis reads the report's own figures. It is a garnish on a report, **never the report itself**.



PART TWO AND A HALF

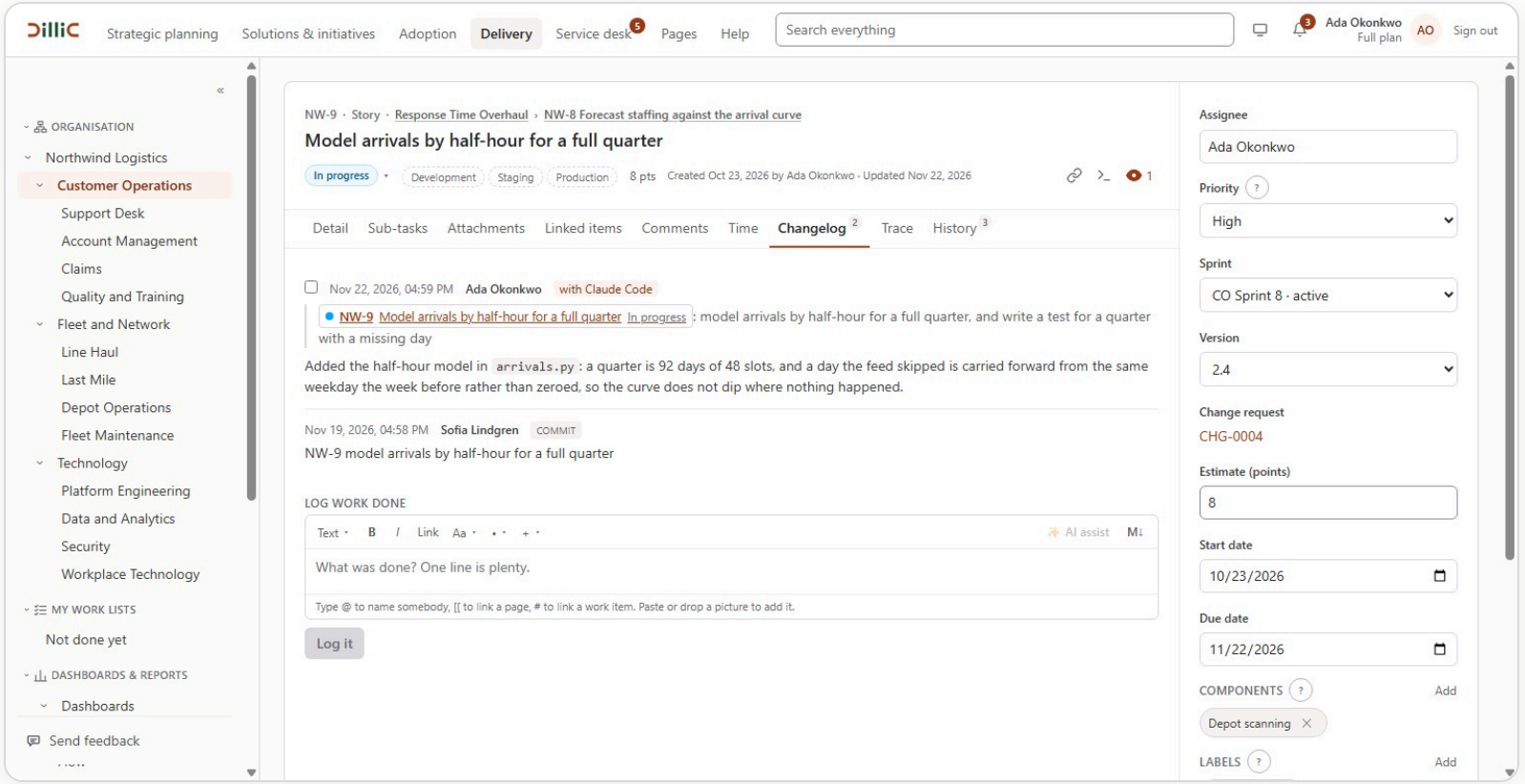
Work done with an assistant

What you asked an AI coding assistant, and what it answered, written on the item it was about - from any assistant that can run a hook, and every one that matters can.

The assistant's words, on the item

One hook script serves Claude Code, Cursor, Codex, Gemini, Copilot, Junie, Kiro, and Devin. Each turn lands as the question, quoted, then the answer.

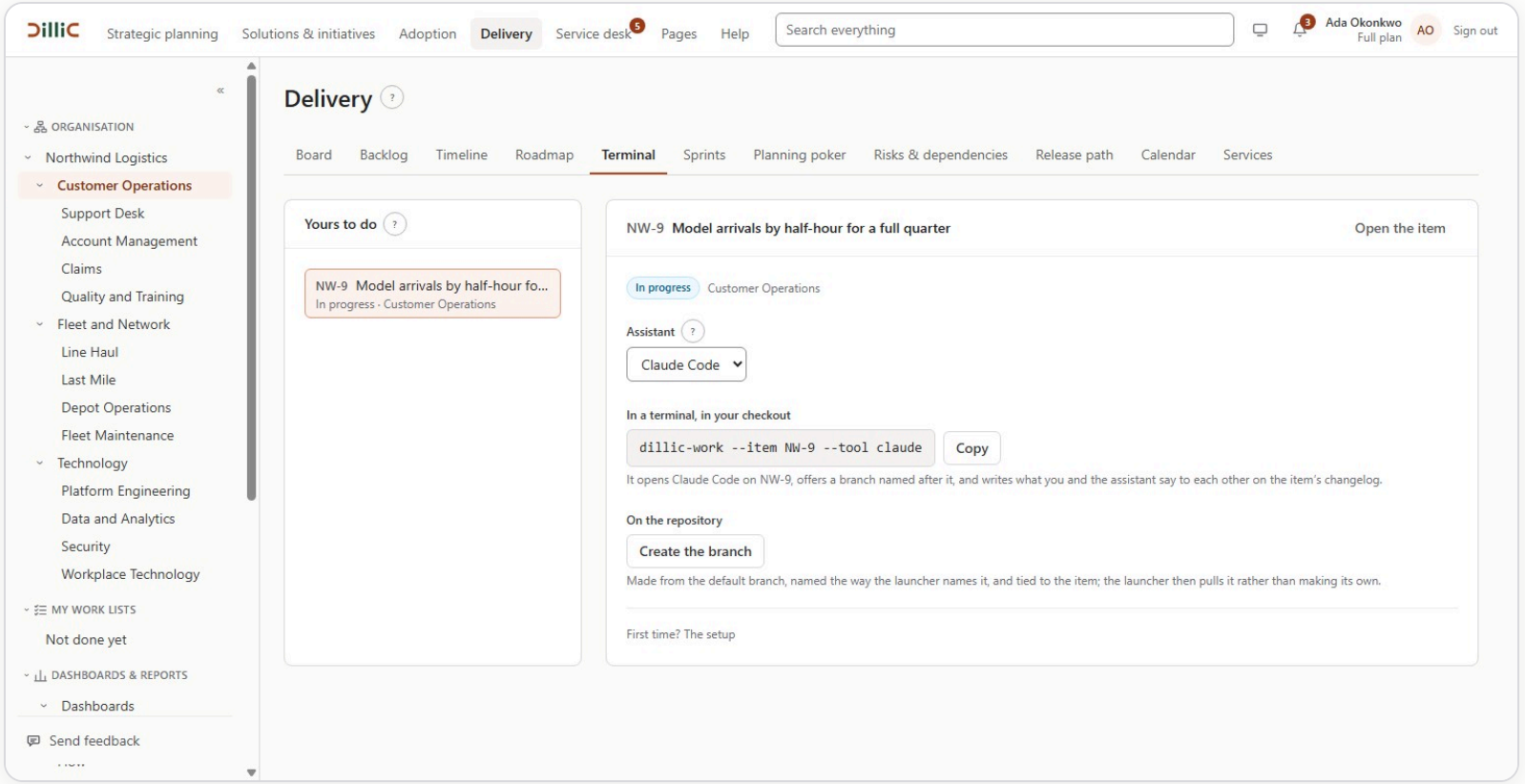
- Still **the person's line**, marked with the tool's name - a record of what was said, moved if it landed on the wrong item, never reworded.
- Commits and pull requests read in the same list, where their time puts them.
- A commit is not logged twice: the repository webhook already records it, so the prompt that asked for it is skipped.



Start from a terminal

Pick one of your items, pick the assistant you prefer, and paste one command. `dillic-work` opens the assistant with the hook wired in, on a branch named for the item.

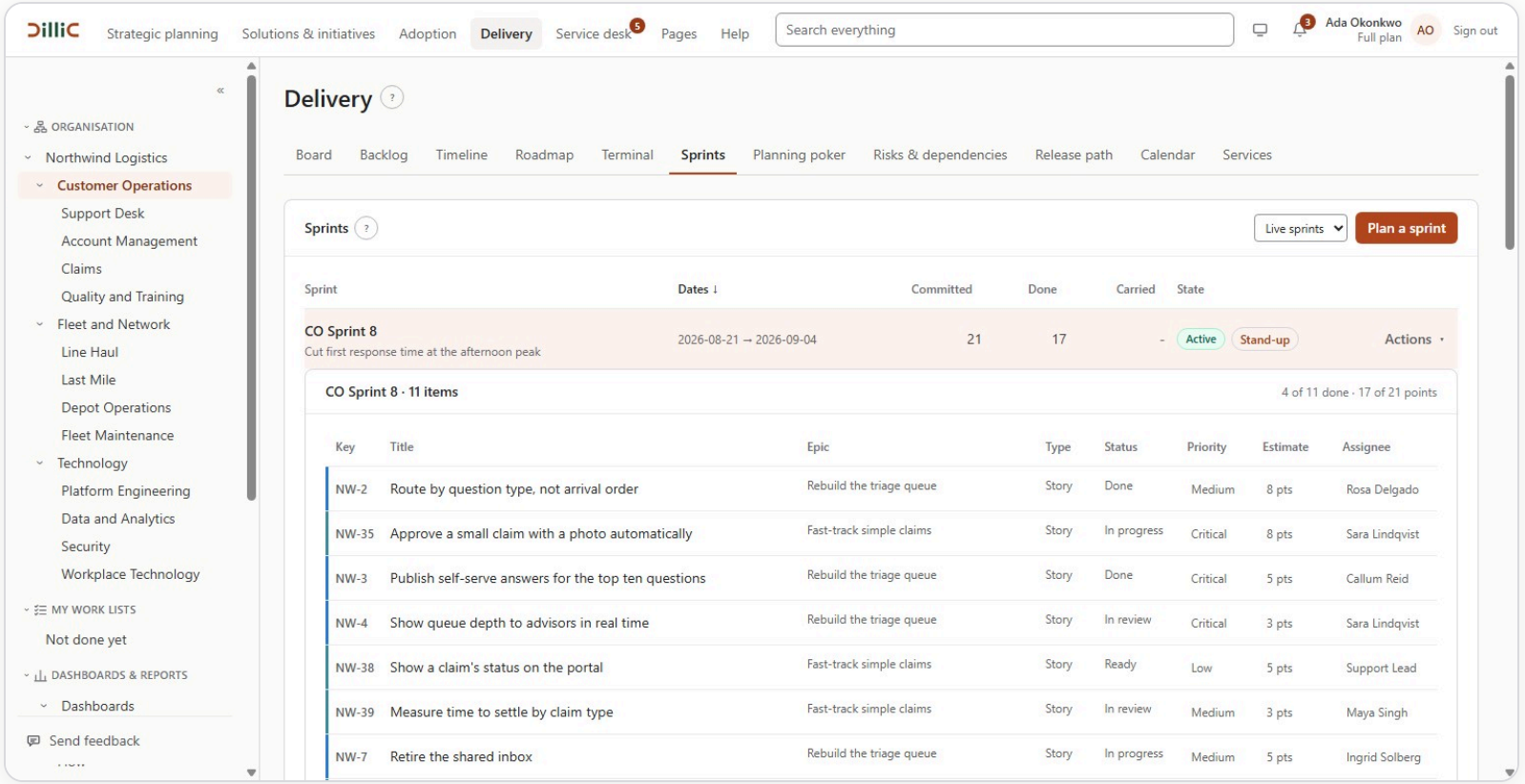
- An item nobody has started is offered its next step first: nothing is logged on work that has not begun.
- A browser cannot open an assistant on your machine, so the tab hands you the command rather than pretending to.



Epics in colour, work in groups, questions anywhere

Every row wears its epic's colour, the same colour wherever the epic appears. Backlog, lists, and board gather under assignee, epic, priority, type, or initiative.

- **Ask** is a question in the list language - *assignee is me and priority is Critical* - and it narrows the board, filters the backlog on the server, and answers a report without saving one.
- The same language the list builder writes, so you learn it once.



PART THREE

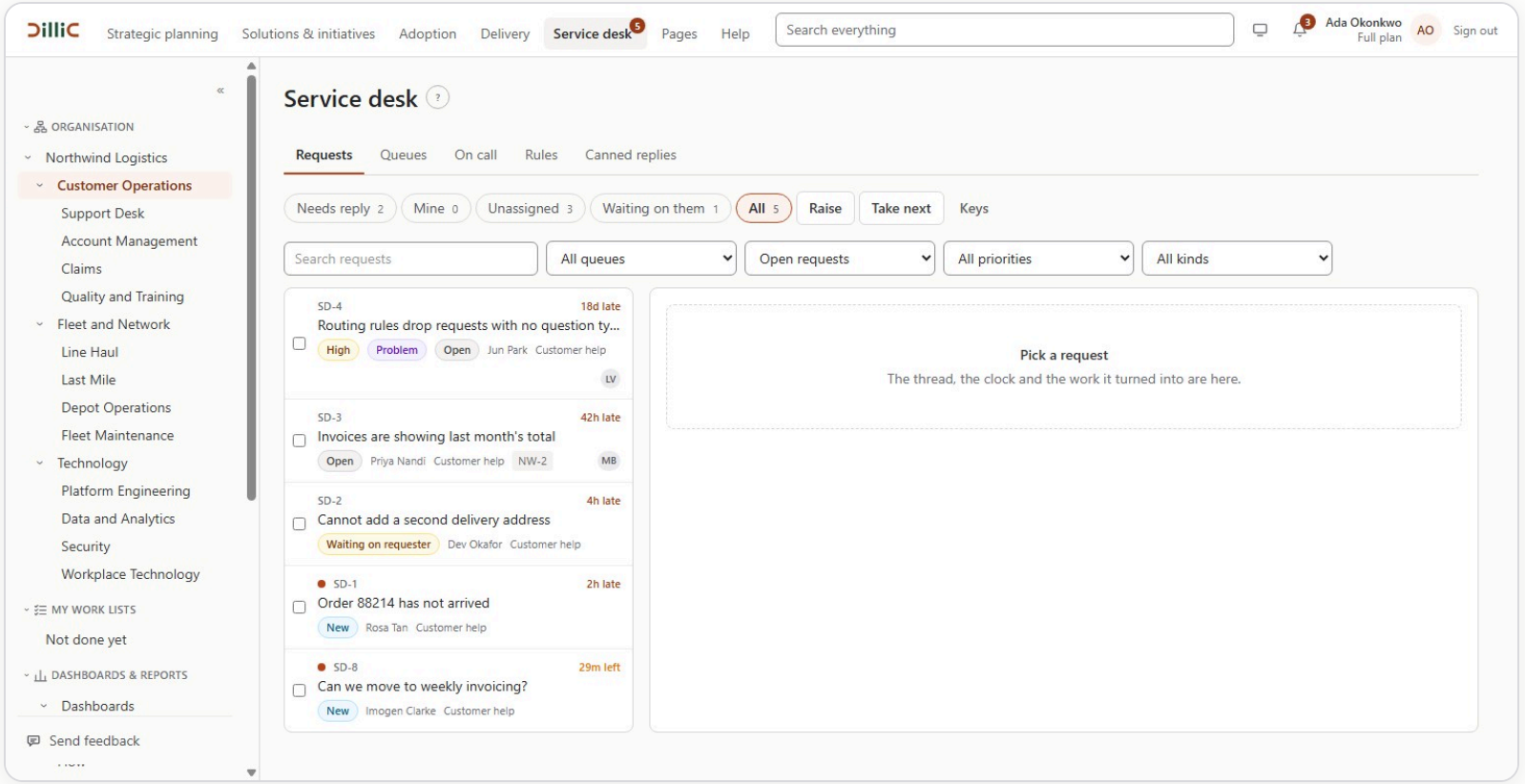
Beside the two halves

The desk that reports back, the people a change happens to, the funding behind it, and the tools an organisation already has.

A desk that needs no seat to ask

Requesters raise and follow requests through a portal of their own, or by mail. An agent takes the next one and sees the thread, the clock, and the work it turned into.

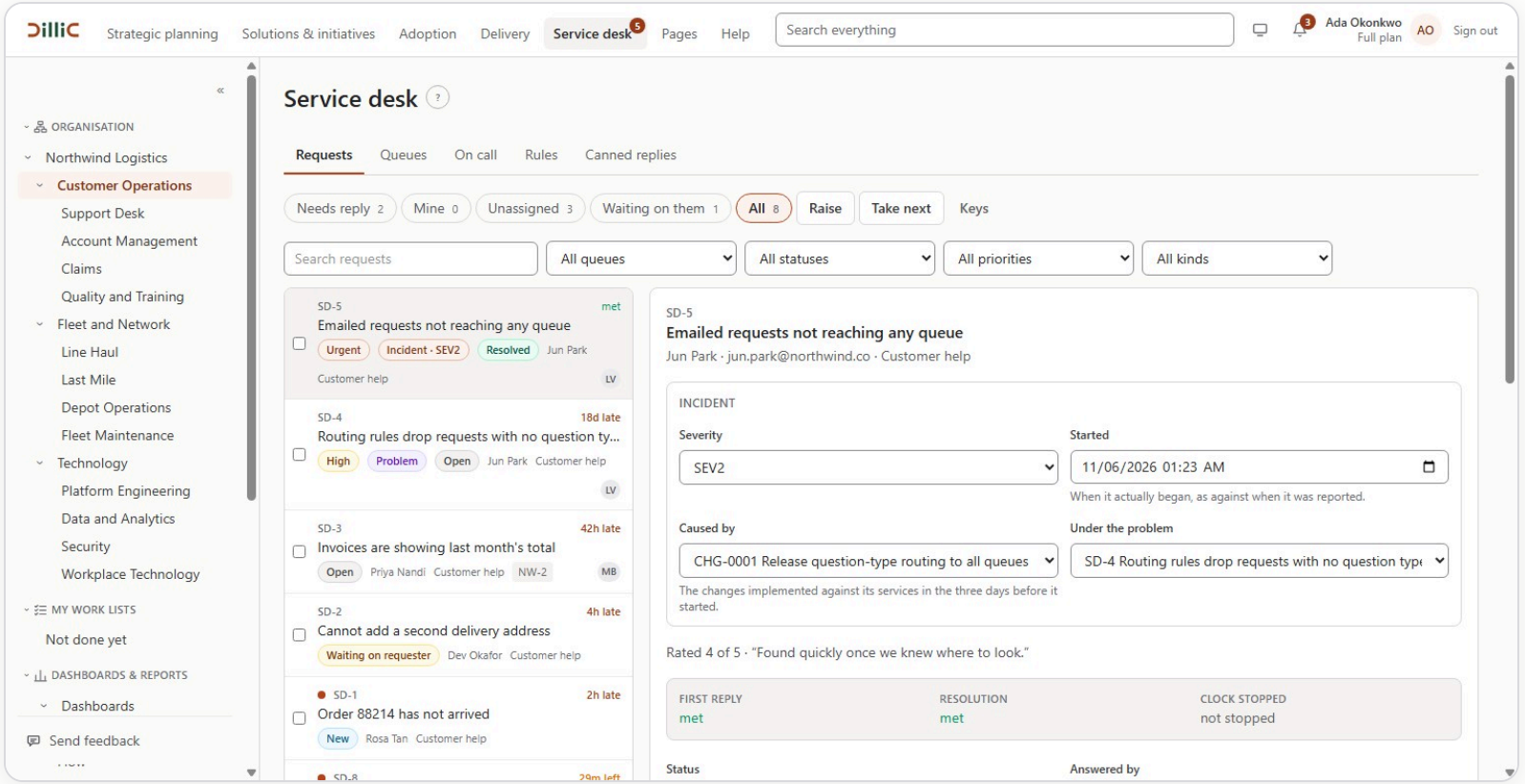
- Queues, routing rules, canned replies, and **clocks for the promises you make**.
- Knowledge articles come from Pages, so the wiki answers the desk, and satisfaction is rated on what was resolved.



An incident names its change

An incident names the services it was about and the change that caused it; a problem gathers several and names the change that fixes it.

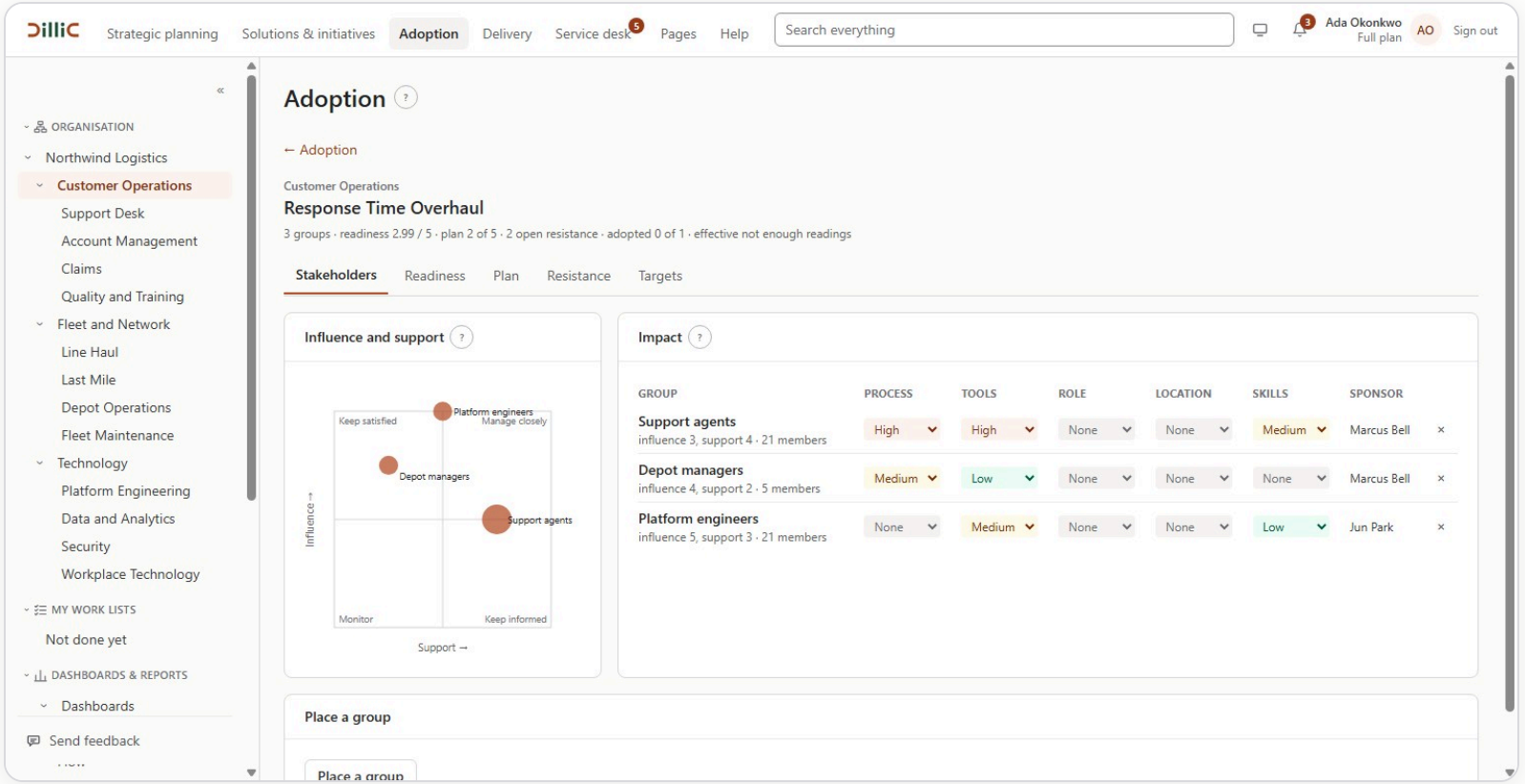
- **A monitor can raise one.** Grafana, PagerDuty, or anything that posts a signed webhook raises an incident on the service it names, adds to it when the alert fires again, and resolves it when the alert clears.
- An organisation that keeps its own desk sends its incidents the same way, so the loop closes without a migration.



The people a change happens to

Adoption answers the question delivery cannot: **did it land?** Stakeholder groups sit on the influence-and-support grid, with the impact on each by process, tools, role, location, and skills.

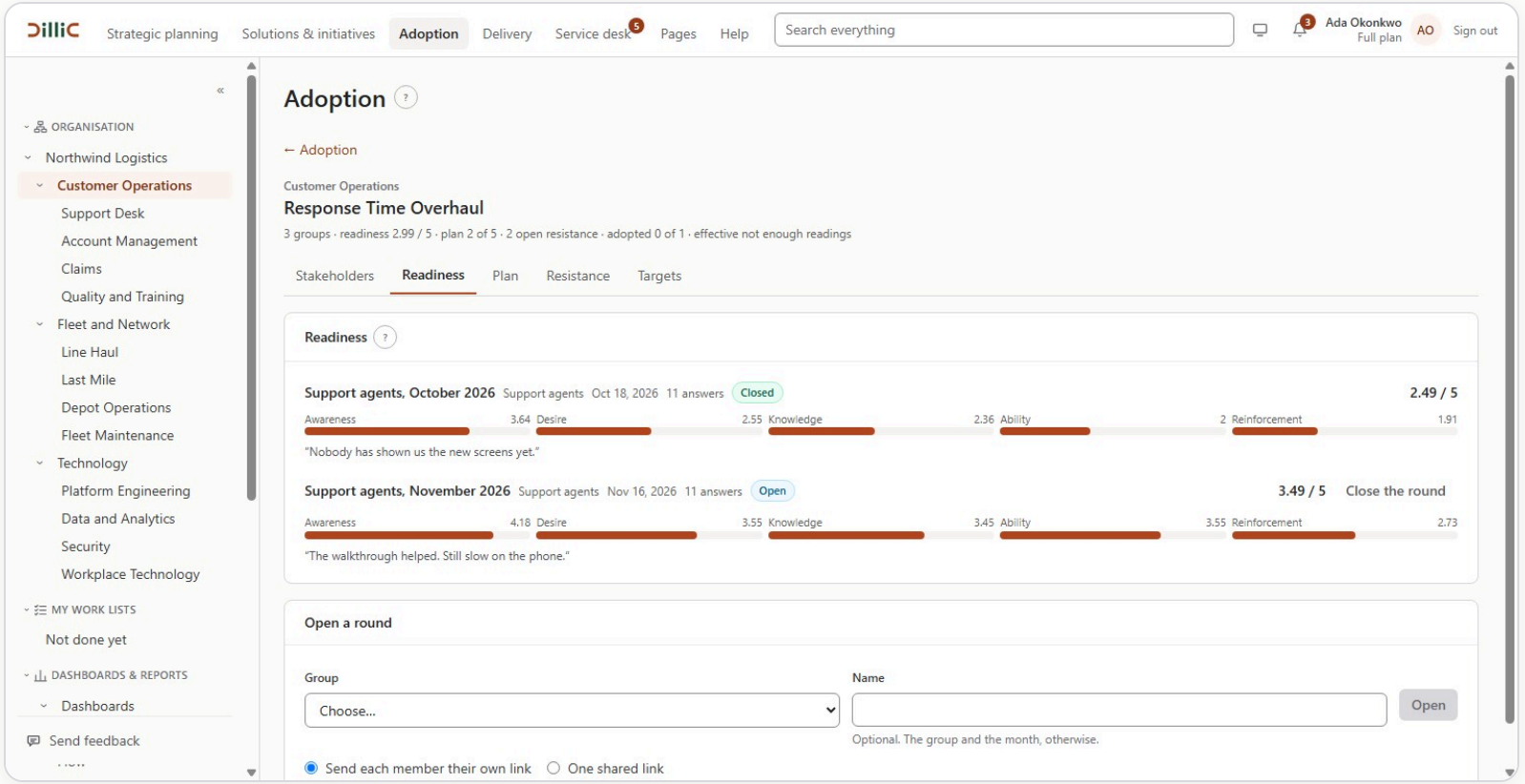
- A communications and training plan, a resistance log, and adoption targets drawn from the measure library.
- They sit beside whether the initiative moved its measure, so shipped and adopted are two answers.



Ask them, and ask again

A readiness round is answered anonymously through a shared link and scored on awareness, desire, knowledge, ability, and reinforcement.

- Rounds a month apart show **what moved**, and what the people said beside it.
- A saturation heatmap shows change landing on each part of the organisation, month by month, before people are overloaded.



PORTFOLIO

Funding, behind an authority

A kanban an initiative is funded through, behind a lean portfolio authority, with the hypothesis and the lean business case it carries.

Value streams

Each with a budget and guardrails, so what is spent beneath an initiative reads against what was meant for it.

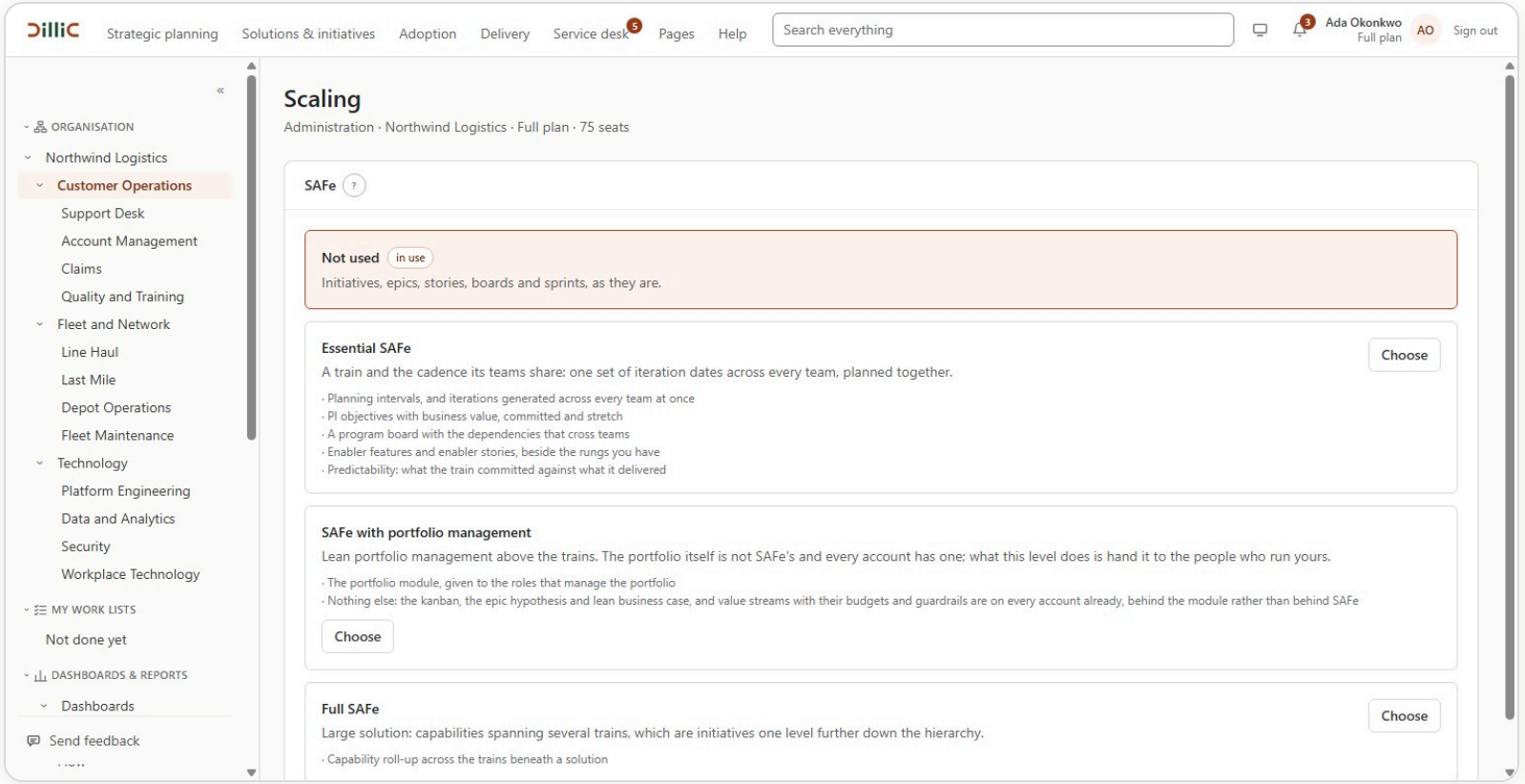
A module, not a framework

Lean portfolio management predates the framework that packaged it. Whose job it is to look at funding is a question for the organisation, so an account switches the portfolio on by itself.

SAFe, as much of it as you run

An account chooses none, essential, portfolio, or full. Essential adds planning intervals that put every team on a train onto one cadence.

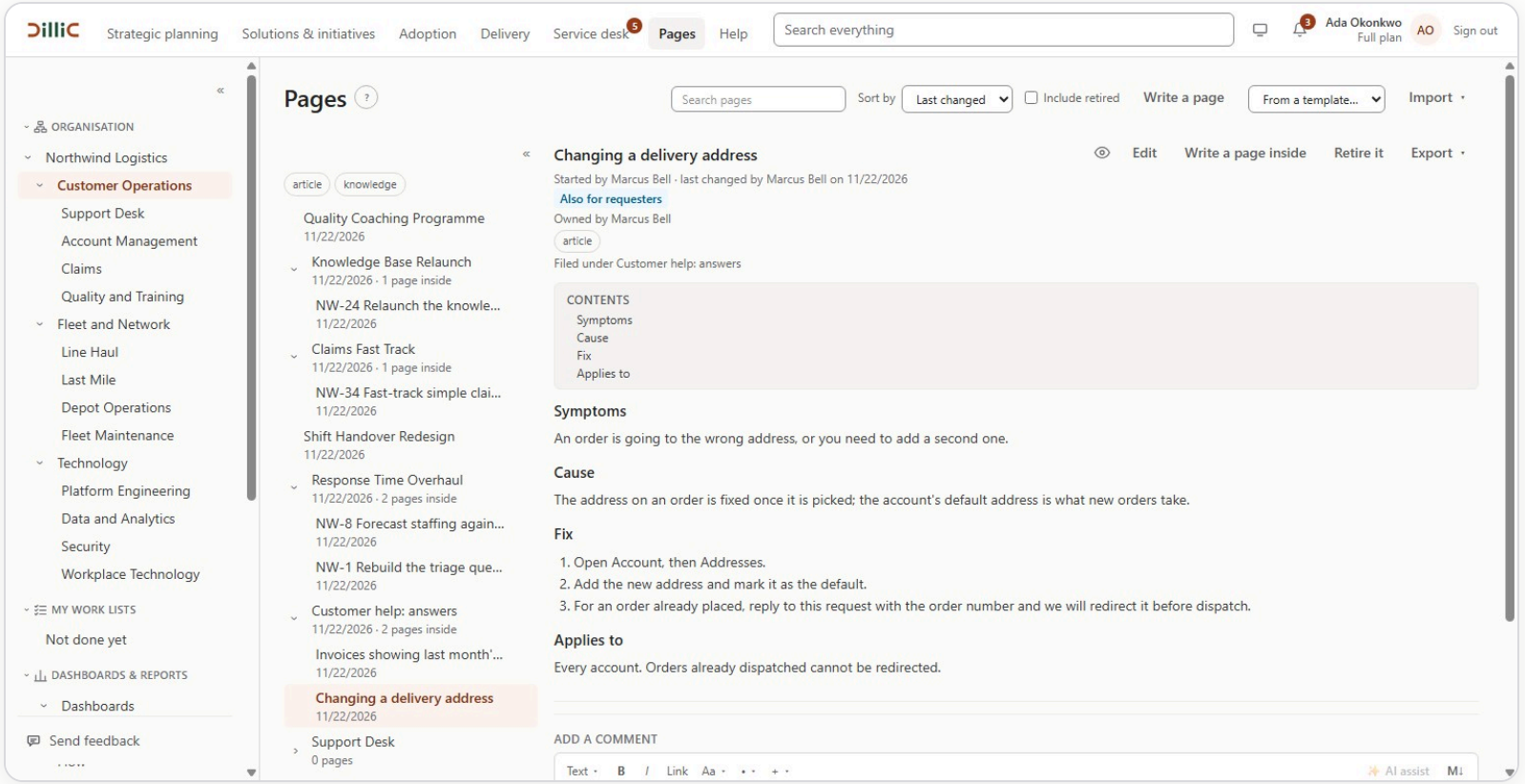
- PI objectives with business value, a program board drawn from the dependencies the work already carries, and predictability.
- WSJF, risks, velocity, flow, and the portfolio are offered to every account; **nobody reaches them through a framework**.



The wiki in the product

Pages are versioned and linked to work. Every initiative has one, and a page answers the desk as a knowledge article.

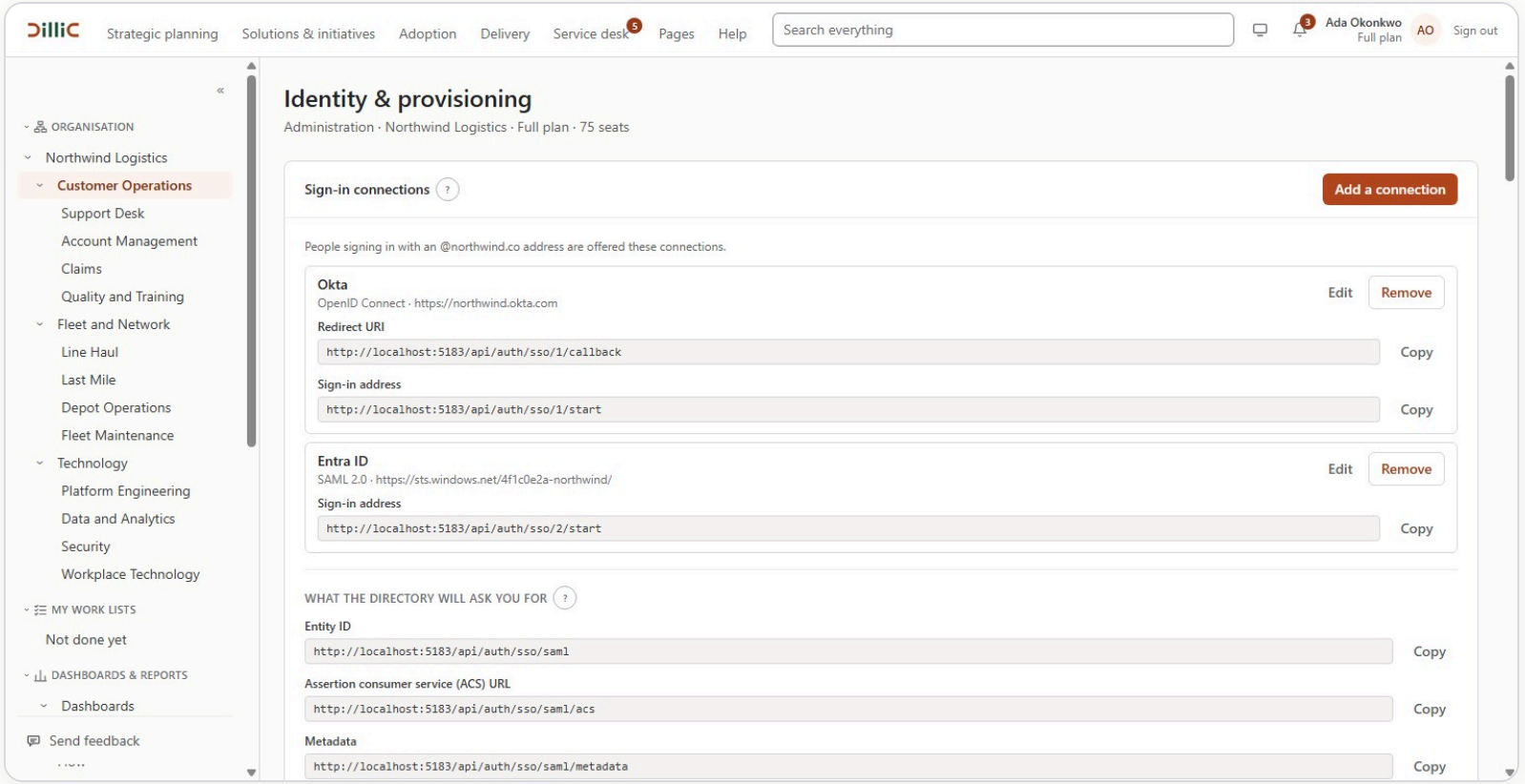
- A page is read by people who never open a board, so Pages is a module of its own.
- A request on the desk is shown the pages that sound like it, before anybody types a reply.



Signing in through your directory

SAML 2.0 and OpenID Connect single sign-on, and SCIM 2.0 provisioning, so a joiner or a leaver is told to DilliC by the directory that already keeps them.

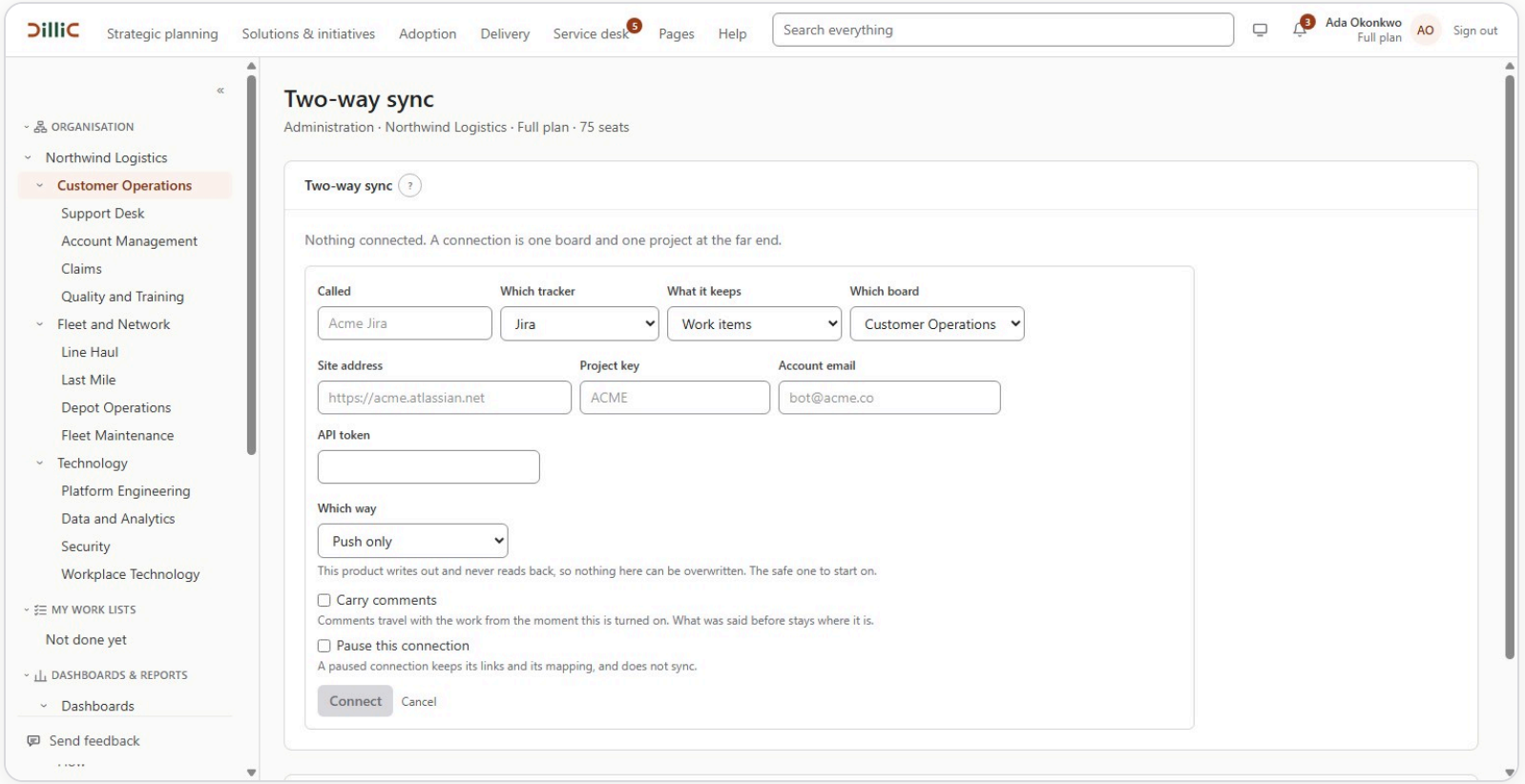
- Multi-factor by passkey or authenticator app, with a policy the account sets.
- Roles scoped to parts of the organisation, and personal access tokens narrowed to some of a person's rights and given an end date.
- **In every plan**, the free one included.



Fifty-three trackers, and a phone

A team's board can stay in its current tool while planning, change control, and adoption move into DilliC. Forty-eight of the fifty-three take a status as well as give one.

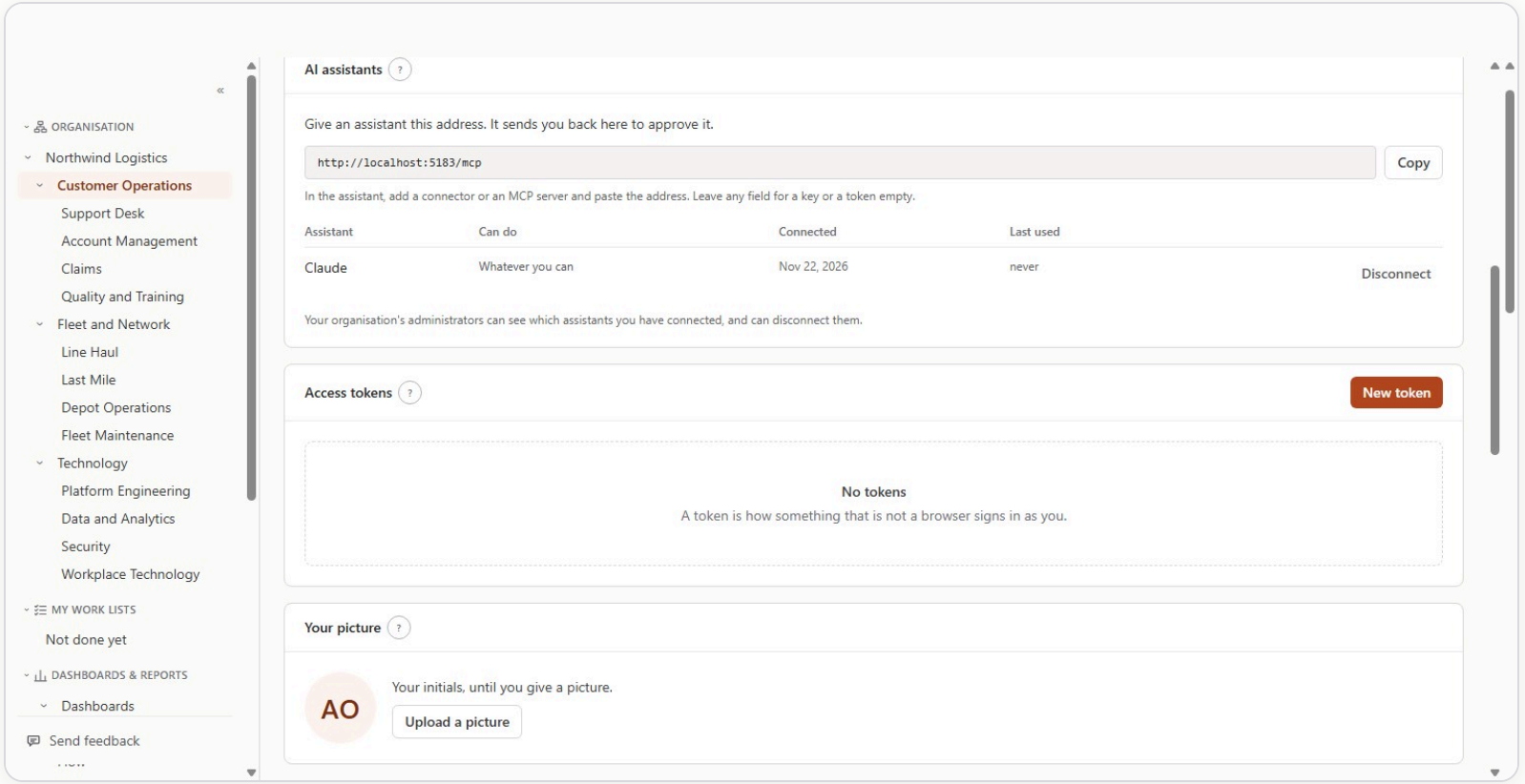
- Statuses travel **by their meaning**, so Done in one tool and Closed in another both arrive complete.
- A connection can bring in what is new, carry comments both ways, and take a call from the far end when something changes.
- The same product on Android and iOS, signed in with a device token the Security screen can revoke.



An assistant signs in as its owner

Give an AI assistant DilliC's address and it sends its owner to DilliC to approve it. It then works over the Model Context Protocol **as that person**, and no further.

- Five tools reach every operation of the API, so roles, scopes, limits, and the audit trail apply unchanged.
- The owner may keep it to the permissions they tick, and the Security page lists each connected assistant with Disconnect.
- A write carrying an idempotency key runs once, so a repeat is safe.



PART FOUR

How it is built

The decisions that are hard to reverse, and the reasoning behind the ones the specification left open.

STACK

Boring where it should be

BACKEND

FastAPI + SQLModel

- SQLite today, with the move to Postgres planned
- Nothing in the schema is SQLite-specific
- Argon2 password hashing, JWT in an httpOnly cookie
- Routers per domain; pure scoring maths in its own module

FRONTEND

React 19 + TypeScript

- Tailwind v4 and TanStack Query
- One file per screen
- The matrix and the burndown are hand-built SVG
- Vite proxies `/api`, so the cookie is first-party

Multi-tenancy is one function. Every tenant-scoped table carries `account_id`, and routers fetch through one helper that returns **404, never 403**, for another tenant's row - so no one can probe an id for existence.

MODELLING

Three things that look different are the same thing

ONE TABLE	WHY
Current state, desired future state and composite	A desired future state assessment is a current-state one answered against an assumed future; a composite is one whose marks are other assessments. Only survey construction and reading differ.
Assertions and causes	Separated by a type field. Only assertions may attach to a factor; that rule makes the distinction worth keeping.
Work items and change requests	Both are a thing in a state with declared moves out of it, so one workflow engine drives both and one gate serves both.

> **History is a record, not a projection.** Every consequential mutation writes an audit event; every work item carries its own immutable history with actor and timestamp.

WHERE THE SPEC LEFT A CHOICE

Two decisions, made and written down

Which standard deviation colours a circle

The spec says "the standard deviation" without saying which axis. DilliC takes the mean of the two per-axis population standard deviations, so disagreement on either axis shows and neither can mask the other.

- Maximum attainable on a −3…3 scale is 3.0
- Reported normalised onto 0…1 against that
- One function, so the definition changes in one place

The colour ramp

The spec asks for blue-green → yellow → pink. The midpoint became an amber for contrast, and the anchors were levelled so the ramp reads by hue.

- Checked with a palette validator
- All three anchors above 3:1 contrast
- Colour is never the only encoding

What "integrate" means here

BOTH DIRECTIONS

GitHub, Bitbucket, GitLab, SourceForge, RhodeCode, Azure DevOps Server

Inbound, a webhook arrives, the account's secret is checked against it, and a work-item reference in the commit message, the branch name, or the pull request gets a link. Outbound, on your say-so and each its own switch: a comment back on what was tied, a branch made for an item, and the item's next step when its pull request merges.

- **Signed:** GitHub (HMAC-SHA256) and SourceForge (HMAC-SHA1, because that is what Allura sends)
- **Secret in a header:** GitLab in its own, Bitbucket and Azure DevOps in one you add - or, for Azure, as the basic-auth password
- **Secret in the body:** RhodeCode, beside the event
- Six services, three ways of proving a sender - each said plainly

INTERCHANGE

Twenty-seven trackers

Import and export in each tool's own columns, Jira and ServiceNow among them. Fifty-three trackers sync, forty-eight of them with statuses written both ways.

- A connection to a service desk can keep change requests in step instead, for thirteen desks; one to Aha! or Jira Align can keep initiatives
- Status translated through a state category, not by name, so a renamed column still exports correctly
- Import defaults to a dry run and reports every row it could not read

COMPLIANCE

Erasure that does not destroy the evidence base

CCPA is answered in two endpoints: export everything held about the signed-in person, and anonymise - which destroys the identity and severs attribution but leaves the content.

Why that reading

A survey response and an approval decision are facts about the organisation as much as about the person. "Delete" must not quietly destroy an organisation's compliance evidence. The last administrator cannot be erased.

SOC 1/2/3

A certification, not a feature - but the prerequisite is built. Every consequential mutation writes an audit event, and every work item and change request carries immutable history with actor and timestamp.

One interface, two implementations

A built-in summariser composes its answer from the report's own figures - which quadrant the entity sits in, which factors scored lowest, and which drew divided answers. It spends no credits, and the layout is always exercised.

- On a paid plan the same calls go to **Claude** - Haiku, Sonnet, or Opus, by tier - falling back to the built-in summariser if the API is unreachable.
- It reads eight report shapes: factors, assessment, root cause analysis, comparison, sprint, planning interval, initiative, and a report you build yourself.
- The **sprint** reader answers both questions the spec asks - a plan while the sprint is ahead, an outcome once it has closed.
- The **initiative** reader compares delivery progress against the movement of its key performance indicator, and that is where an initiative that ships without changing anything shows up.
- AI is also a licensing feature: on the starter plan the panel explains what the upgrade adds, rather than offering a button that fails.

TESTS

6,594 tests, aimed where a bug would be invisible

PLANNING

The scoring maths at both extremes. Weighted and recursive composite roll-ups. The hierarchy and weighting rules. Inherited access and the tenant boundary. The 52-mark cap. An end-to-end propose → vote → accept → survey → matrix flow.

- The anonymity guarantee has its own test
- So does the full-reset chain break

CHANGE MANAGEMENT

That an undeclared move is refused. That a gate blocks and the item stays put. That an author cannot approve their own work. That a Kanban work-in-progress limit stops the fourth card.

- That a normal change cannot skip ITIL authorisation and a standard one can
- That `ANW-14` is not a reference to `NW-14`
- That every tracker survives an export/import round trip
- That each of six repositories is let in by its own proof and no other, and that a line written with an assistant is moved, never reworded

4,092 on the API, 1,833 on the web screens, 569 on the phone's, and 100 on the code they share, including the hook every assistant runs and the launcher that opens one.

SCOPE

Not in this release, and where it plugs in

DEFERRED	WHERE IT PLUGS IN
The commit's own time on a code link	A link carries the moment the webhook arrived; the author's timestamp is in every payload and is one field away
Assistants with no hooks: Aider, JetBrains AI Assistant	The changelog endpoint takes a reference, a prompt, and an answer from anything that can make an HTTP request
Folding a session's many turns into one changelog line	The hook posts a line per turn; a window would coalesce them
A live run of the newest tracker adapters	A smoke tool tries one adapter against a real account; those added in November have not yet run against one
An agent inside the browser	The Terminal tab hands you the command; running the assistant on a server with a checkout is a different product decision

ACCEPTANCE THEMES

What "done" was measured against

- Model complex organisational hierarchies.
- Support planning and change management as **separate or combined** workflows.
- Measure current state and compare future states using **validated factors**.
- Support governance, auditability, and enterprise compliance requirements.
- Provide AI-enhanced reporting and planning support.
- Integrate with common enterprise and engineering tools.

One line from objective to release, and back from the desk

Open it yourself: Northwind Logistics, the organisation in this deck, opens from the sign-in page.