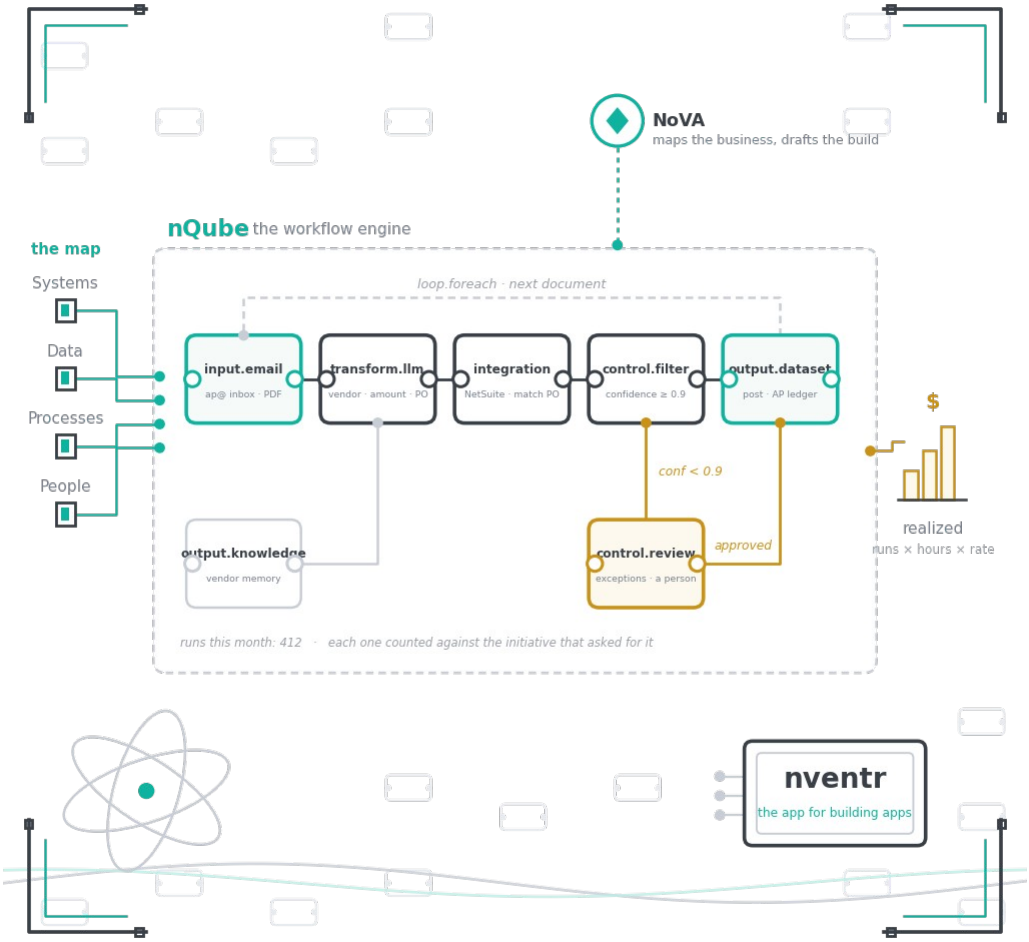
NVENTR — THE ONBOARDING GUIDEBOOK: MAP, THEN OPTIMIZE

Version 1.0 · October 2026 · How a new customer starts with nventr: map the business first — systems, data, processes, people — then let the platform propose the optimizations worth doing, build them on nQube, the App Kit, Integrations, the Training Center and Agents, and track every initiative against a savings forecast that the platform proves from what actually ran. Written for the people who run the business and the people who will build with them. Web edition at nventr.io/guide; NoVA, the platform's agent, teaches the same path in your console.



The path this book walks — the four maps feed nQube, the workflow engine, which NoVA drafts from the map; what runs is measured as realized savings

© 2026 Nventr. All rights reserved. — Map, then optimize.

Contents

PART I — WHY MAP FIRST

The one idea in this book 3
What you get at the end 3
How NoVA runs this with you 4

PART II — THE FOUR MAPS

Systems: what you pay for and where people live 6
Data: where the records live 6
Processes: where the hours go 7
People: who does the work 8

PART III — THE SIX LEVERS

From a map to a proposal 8
The levers 9
Choosing the first one 9

PART IV — BUILDING IT

The platform, one initiative at a time 10
Your first workflow on nQube 10
Your first app on the App Kit 11
Connecting what you keep 12
Your first model in the Training Center 12
Your first agent 12

PART V — TRACKING AND FORECASTING

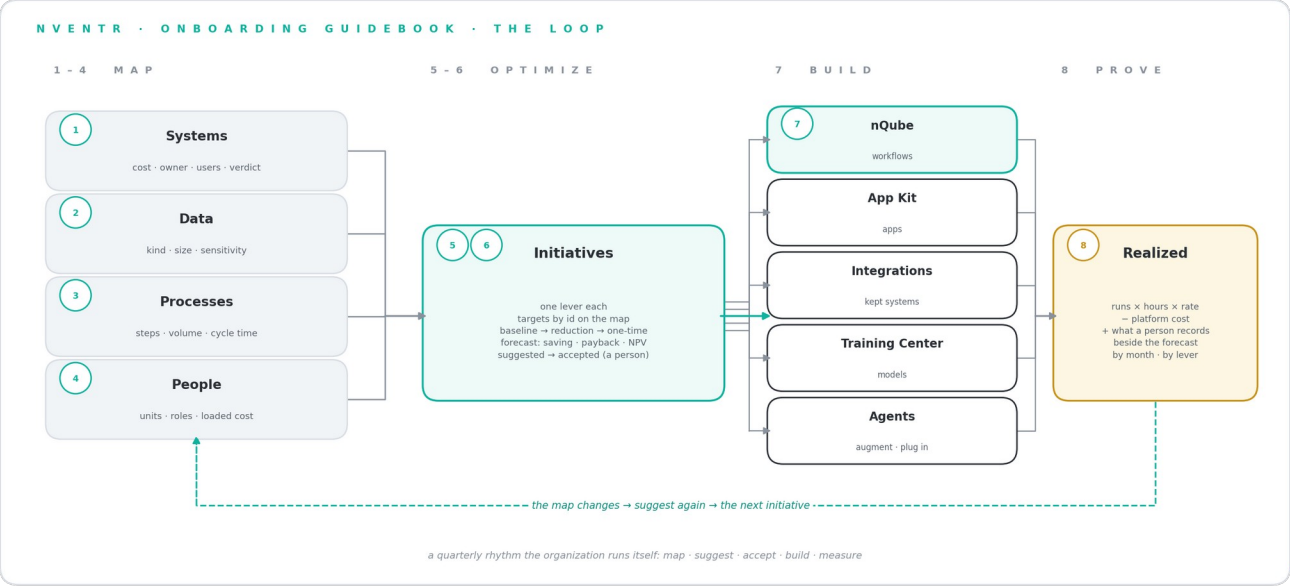
The initiative over time 12
The forecast 13
The realized column 14

PART VI — THE 30/60/90 PLAN

Days 1–30: map 15
Days 31–60: build the first win 15
Days 61–90: prove, then widen 16
Where to click 16

© 2026 Nventr. All rights reserved. — Map, then optimize.

The one idea in this book



The loop — four maps feed the initiatives, the initiatives are built on the platform and measured, and the map changes again; the numbers are the Blueprint's step rail

Most software onboarding starts with the software: here is the product, here are the features, here is how to log in. nventr starts with your business. Before anything is built, the platform asks you to put down, in plain structured form, what you already have — the systems people live in, the places your data sits, the processes that eat the hours, and the people who do the work. That picture is the **Blueprint**. Everything else in this book follows from it.

The reason is practical rather than philosophical. nventr can build a great many things: a workflow on nQube that takes a six-step manual process down to one review, an application on the App Kit that replaces a system you are paying for, a model trained on your own records, an agent that augments a role that is drowning. Each of those is only worth building if it pays for itself against something real — a cost you carry today, in licenses, in hours, in headcount, in delay. The map is where those costs are written down. Without it, every proposal is a guess and every saving is a story. With it, the platform can point at the specific process, the specific system, the specific role, say what it would change, and put a number on it that you can argue with.

So the method is simple, and it is the whole of Part I: **map first, optimize second, build third, prove always.** The map takes an afternoon to start and a few weeks to finish. The optimizations fall out of it. The building happens on the platform you already have. And the proof — the realized savings beside the forecast — accrues month by month from what the platform actually ran.

↑ Contents

What you get at the end

A customer who follows this guidebook finishes the first quarter with five things, and every one of them lives in the console at Blueprint:

© 2026 Nventr. All rights reserved. — Map, then optimize.

- **A business map** — systems with their yearly cost and a verdict (keep, augment, replace, retire), data stores with their sensitivity, processes with their steps, volumes and cycle times, and an org chart of units and roles with loaded cost. It is yours; export it, print it, hand it to a new hire.
- **A portfolio of initiatives** — each one a named optimization against named records on the map, with a lever (process, model, augment, replace, new software, agent), a rationale, a forecast and a status. nventr proposes; a person accepts.
- **Deliverables on the platform** — the workflows, apps, agents, models and integrations each initiative produced, linked back to the initiative that justified them.
- **A forecast you can defend** — baseline cost, reduction, one-time cost and platform cost per initiative, rolled up into annual savings, payback months and a three-year net present value for the whole plan.
- **Realized savings, measured** — from the month an initiative goes live, the platform counts what ran (workflow runs, app usage), prices the hours it displaced at your own loaded rates, subtracts what the platform cost, and writes the result beside the forecast. You add anything it cannot see. The gap between the two columns is the honest conversation.

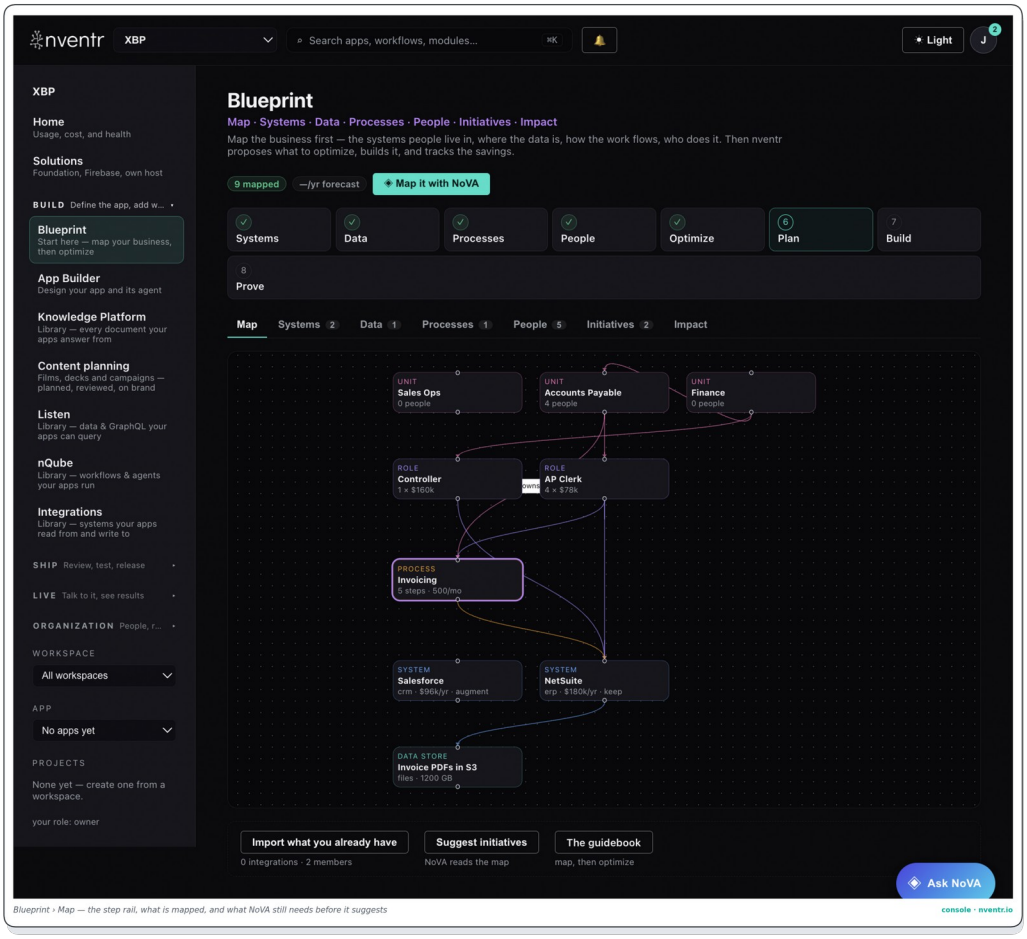
↑ Contents

How NoVA runs this with you

NoVA is the agent in every nventr console. For onboarding it does three things. It **interviews**: open the Blueprint and say "map my business" and it asks one question at a time — what system does finance live in, roughly what does it cost, who owns invoicing, how many a month, how long does one take — and records each answer onto the map as you give it. It **proposes**: once the map has systems and a process with steps, "what should we automate first" produces initiatives, each tied to the records that justify it, each with a draft forecast. And it **tracks**: "how are we doing" reads the plan back — what is live, what it has saved, what is waiting on a decision.

Two things NoVA never does. It never accepts an initiative — that is a person's click, and the console marks it so. And it never invents a saving — every forecast number traces to a field on the map, and every realized number traces to a run, a usage count or a figure a person typed in with their name on it.

© 2026 Nventr. All rights reserved. — Map, then optimize.

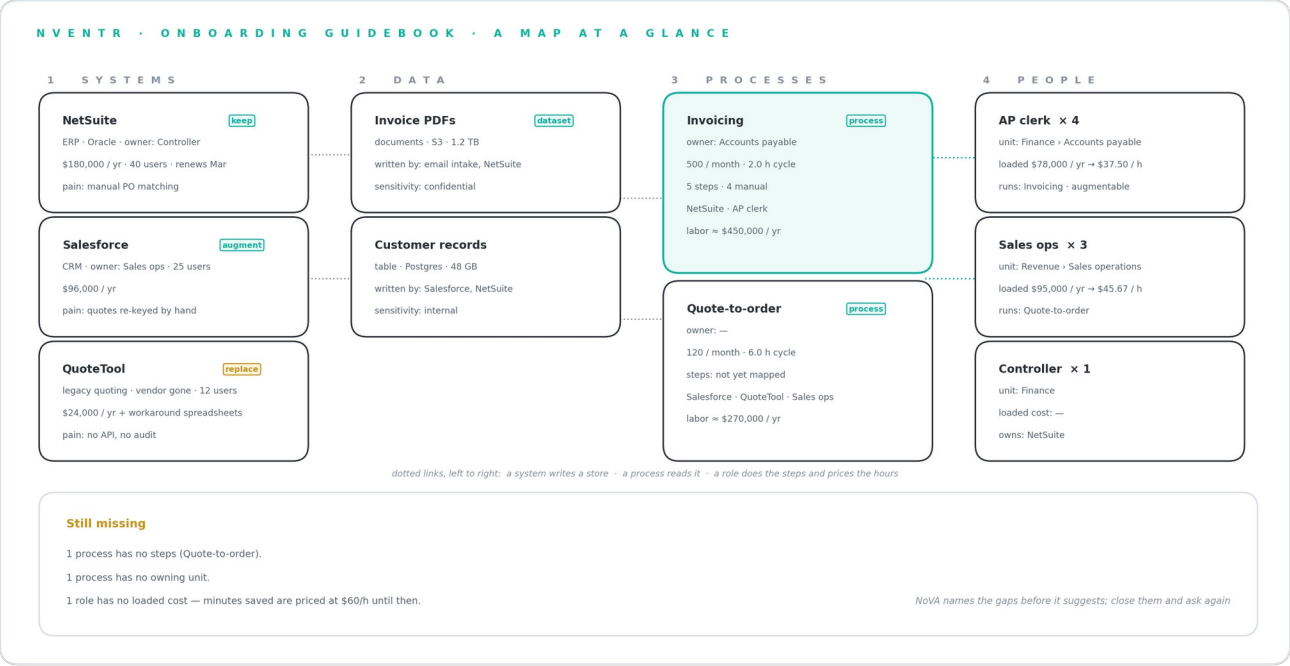


The Blueprint in the console — the step rail across the top, the records NoVA has mapped so far on the canvas, and the verbs along the bottom

↑ Contents

© 2026 Nventr. All rights reserved. — Map, then optimize.

Systems: what you pay for and where people live



A map at a glance — three systems with verdicts, two data stores, two processes, three roles, the links between them, and what NoVA says is still missing

Start with systems because they are the easiest to list and they anchor everything else: a process runs *in* systems, data lives *in* systems, roles spend their day *in* systems. For each one record a name, a category (ERP, CRM, HR, ticketing, email, BI, spreadsheet, custom, other), the vendor, the yearly cost, the renewal date, the role that owns it, how many people use it and the pain points people name without prompting. Then give it a verdict — **keep**, **augment**, **replace**, **retire** — or leave it undecided for now. The verdict is the first optimization decision you make, and you make it before the platform suggests anything.

If your organization already connects systems through nventr Integrations, the Blueprint imports them in one click; you fill in cost and verdict. Spreadsheets are systems too — the ones that hide the most manual work usually are.

Worksheet, one line per system: name · category · vendor · yearly cost · renewal · owner · users · pain points · verdict.

↑ Contents

Data: where the records live

A data store is any place records accumulate: a database behind a system, a warehouse or lake, a file share, a spreadsheet, a SaaS API you pull from. Record its kind, the engine if you know it, the rough size, which systems write to it, and its sensitivity — public, internal, confidential, regulated. Sensitivity matters early because it decides what a model may be trained on and what an agent may read; nventr enforces that at the data layer, not in a prompt.

The data map is what makes the **model** lever possible. A process with ten thousand historical decisions in a store you control is a model waiting to be trained; the same process with no record of past decisions is not. Knowing which is which before you plan saves a quarter.

© 2026 Nventr. All rights reserved. — Map, then optimize.

Worksheet: name · kind · engine · size · sensitivity · systems that write it · dataset (once imported).

↑ Contents

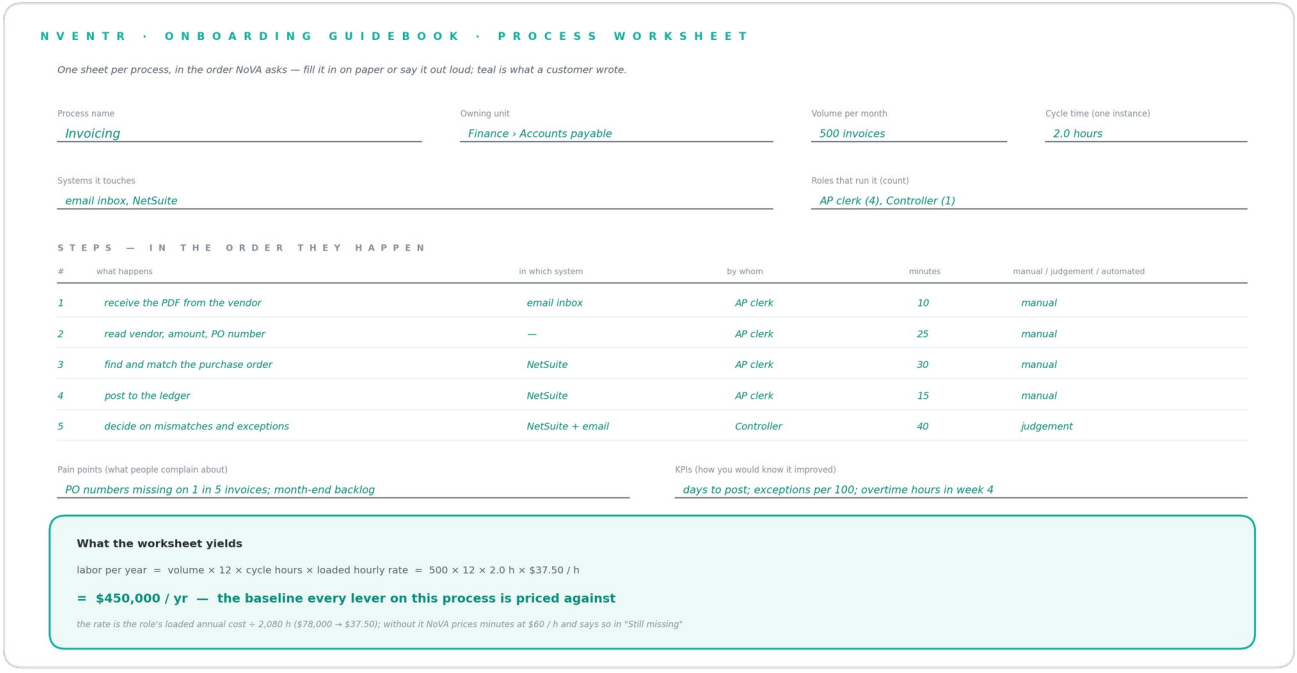
Processes: where the hours go

This is the map that pays. A process is a repeatable piece of work with a start and an end: invoicing, onboarding a hire, triaging a support ticket, reconciling a statement, approving a purchase. For each one record the owning unit, the **steps** in order — each step names the system it happens in and the role that does it — the **volume per month**, and the **cycle time** of one instance in hours. Add the pain points and the KPIs the owner already watches.

Volume and cycle time are what the forecast is built from. A process that runs five hundred times a month and takes two hours each is a thousand hours a month; at the loaded rate of the role that does it, that is a yearly labor cost the platform can now reason about. Half of those hours removed by a workflow is a number, not an adjective. Be honest about cycle time — count the waiting, the chasing, the re-keying — because the saving is only ever as real as the baseline.

Pick the three processes that eat the most hours and map them fully before mapping twenty shallowly. NoVA will tell you which processes have no steps yet; a process without steps cannot be priced.

Worksheet: name · owning unit · steps (name · system · role · minutes) · volume/month · cycle hours · pain points · KPIs.



NVENTR · ONBOARDING GUIDEBOOK · PROCESS WORKSHEET

One sheet per process, in the order NoVA asks — fill it in on paper or say it out loud; teal is what a customer wrote.

Process name	Owning unit	Volume per month	Cycle time (one instance)
Invoicing	Finance › Accounts payable	500 invoices	2.0 hours

Systems it touches	Roles that run it (count)
email inbox, NetSuite	AP clerk (4), Controller (1)

STEPS — IN THE ORDER THEY HAPPEN

#	what happens	in which system	by whom	minutes	manual / judgement / automated
1	receive the PDF from the vendor	email inbox	AP clerk	10	manual
2	read vendor, amount, PO number	—	AP clerk	25	manual
3	find and match the purchase order	NetSuite	AP clerk	30	manual
4	post to the ledger	NetSuite	AP clerk	15	manual
5	decide on mismatches and exceptions	NetSuite + email	Controller	40	judgement

Pain points (what people complain about)	KPIs (how you would know it improved)
PO numbers missing on 1 in 5 invoices; month-end backlog	days to post; exceptions per 100; overtime hours in week 4

What the worksheet yields

labor per year = volume × 12 × cycle hours × loaded hourly rate = 500 × 12 × 2.0 h × $37.50 / h

= $450,000 / yr — the baseline every lever on this process is priced against

the rate is the role's loaded annual cost ÷ 2,080 h ($78,000 → $37.50); without it NoVA prices minutes at $60 / h and says so in "Still missing"

The process worksheet, filled in for invoicing — the questions in the order NoVA asks them, and the yearly labor figure the answers yield

↑ Contents

© 2026 Nventr. All rights reserved. — Map, then optimize.

People: who does the work

The org chart on the Blueprint is units and roles, not names. A **unit** has a parent, a head role and a headcount. A **role** belongs to a unit, has a headcount and a **loaded annual cost** per person (salary, benefits, overhead — the number finance uses), lists the processes it runs and the systems it uses, and carries one flag: **augmentable** — would this role be better with an agent beside it than with another hire? Notes hold what the org chart cannot.

Loaded cost is what prices every hour saved on the process map; without it the platform uses a default rate and says so. The augmentable flag is what makes the **augment** lever concrete: an agent that drafts, retrieves, summarizes and prepares for a role that is capacity-bound is often the fastest payback on the whole plan, and the least disruptive.

If your team is already on nventr, the Blueprint imports your members as roles by their seat; you adjust from there.

Worksheet: unit · parent · head role · headcount; role · unit · headcount · loaded cost · processes · systems · augmentable.

↑ Contents

From a map to a proposal



The six levers — what each one changes, what it is built from, what it is priced against, and the typical share of the baseline it removes

An optimization on nventr is called an **initiative**, and every initiative pulls exactly one of six levers. The levers are deliberately few so that a plan reads at a glance and so that every proposal is priced the same way: a baseline the map already holds, a reduction the lever is known to deliver, a one-time cost to build, a yearly platform cost to run.

When you ask NoVA what to optimize, it reads the whole map, finds the heaviest manual process, the costliest system with a replace or retire verdict, the most augmentable role, the data store rich enough to train on, and proposes one

© 2026 Nventr. All rights reserved. — Map, then optimize.

initiative per lever the map can support — never one the map cannot justify, never a duplicate of one you already have. Each proposal names its targets by id, so you can click through from the initiative to the process it would change. Suggested initiatives sit in the plan as **suggested** until a person accepts or rejects them.

↑ Contents

The levers

Process optimization — a workflow on nQube takes a manual, multi-step process down to one review or none. Baseline: the labor cost of the targeted processes (volume × cycle hours × loaded rate). Typical reduction 40 percent. Deliverable: a workflow. The fastest to build and the easiest to measure, because every run is counted.

Train a model — a model trained on your own records replaces a judgment step that today needs a person: classification, extraction, scoring, forecasting. Baseline: the labor in the judgment steps plus the cost of the errors. Typical reduction 25 percent. Deliverable: a training run in the Training Center and a model block in a workflow. Needs a data store with history.

Augment a role — an agent beside a capacity-bound role: it drafts, retrieves, prepares, follows up; the person decides. Baseline: the loaded cost of the role's headcount. Typical reduction 30 percent of that capacity, usually realized as deferred hiring. Deliverable: an agent with a knowledge scope and allowlisted tools.

Replace a system — an application built on the App Kit replaces a system with a replace or retire verdict, on your own infrastructure, with your data in it. Baseline: the system's yearly license plus the labor of working around it. Typical reduction 60 percent of the license. Deliverable: an app, usually with an integration to migrate from the old system. The biggest saving and the longest build; sequence it after a workflow win.

New software — an application for something no system does today, where the process lives in email and spreadsheets. Baseline: the labor of the manual process. Typical reduction 50 percent. Deliverable: an app.

Plug in an agent — an agent on the website, in the app or inside an existing system, answering and acting under review: triage, routing, first-line response. Baseline: the labor of the process. Typical reduction 35 percent. Deliverable: an agent, and the workflow it reaches for when it needs a side effect.

↑ Contents

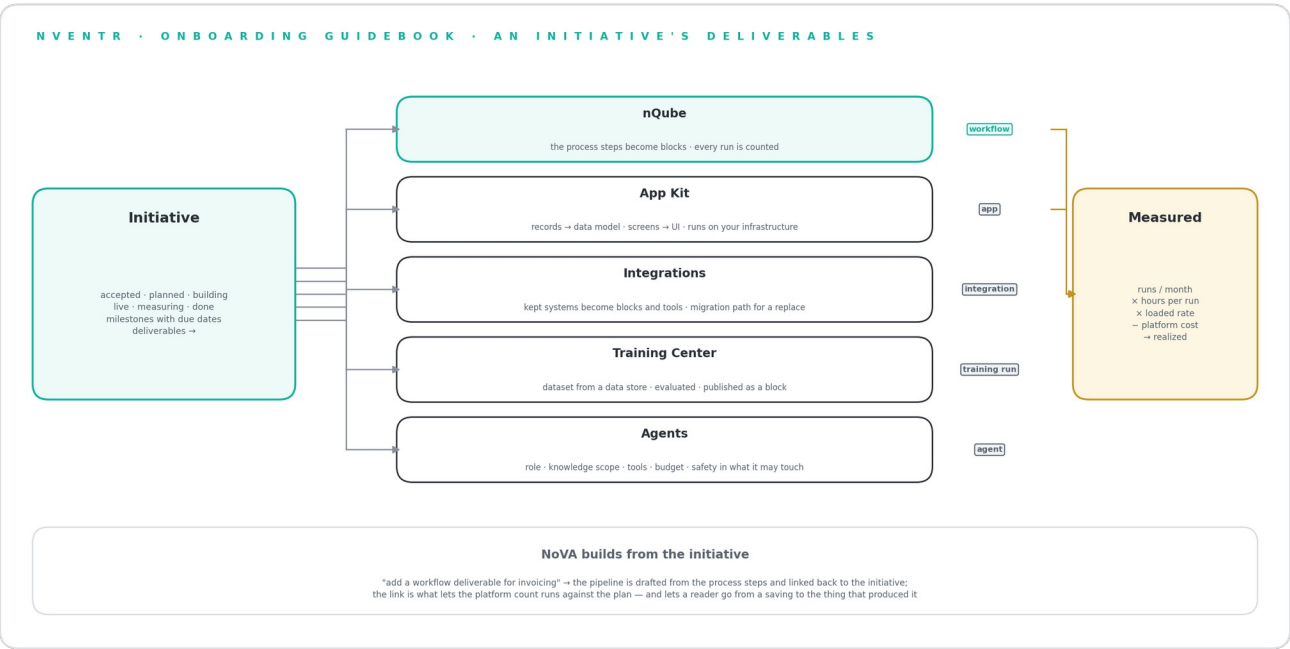
Choosing the first one

The right first initiative has four properties: it targets a process that is fully mapped (steps, volume, cycle time, roles with loaded cost); it pulls the process lever or the augment lever, because those ship in weeks; its payback is under six months on the forecast; and its owner is in the room. A plan that starts with a system replacement spends its first quarter on migration and shows no savings; a plan that starts with the heaviest manual process shows a realized number by month two, and that number is what wins the room for the bigger levers.

↑ Contents

© 2026 Nventr. All rights reserved. — Map, then optimize.

The platform, one initiative at a time



What gets built — an initiative's deliverables across nQube, the App Kit, Integrations, the Training Center and Agents, each linked back to the plan

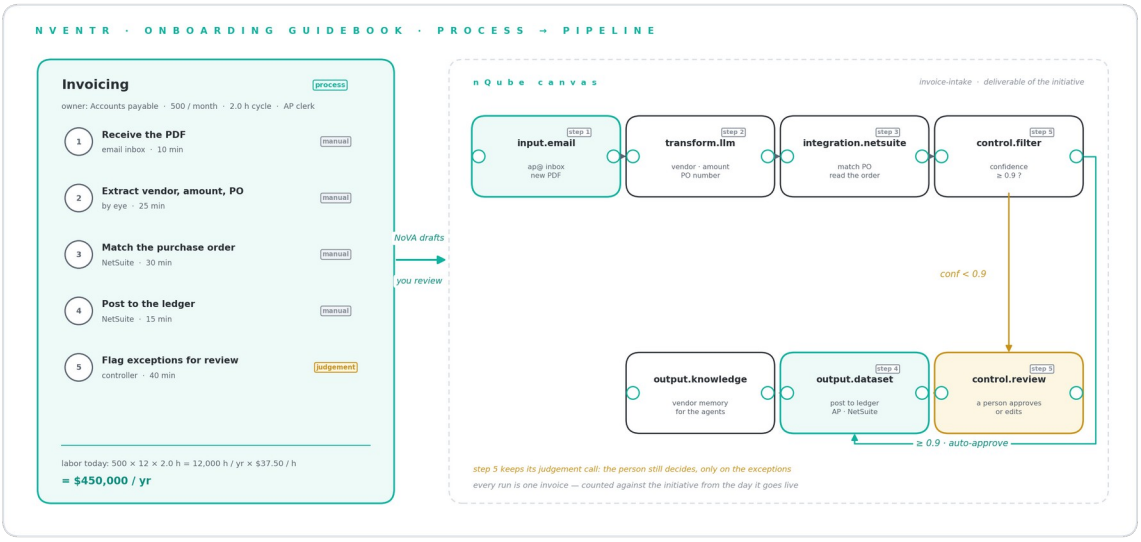
An accepted initiative is a list of deliverables, and every deliverable is built with one of five parts of the platform. The Blueprint links them: open an initiative, add a deliverable, and NoVA builds it in the right place and records the link, so a workflow always knows which initiative justified it and an initiative always knows what it shipped.

↑ Contents

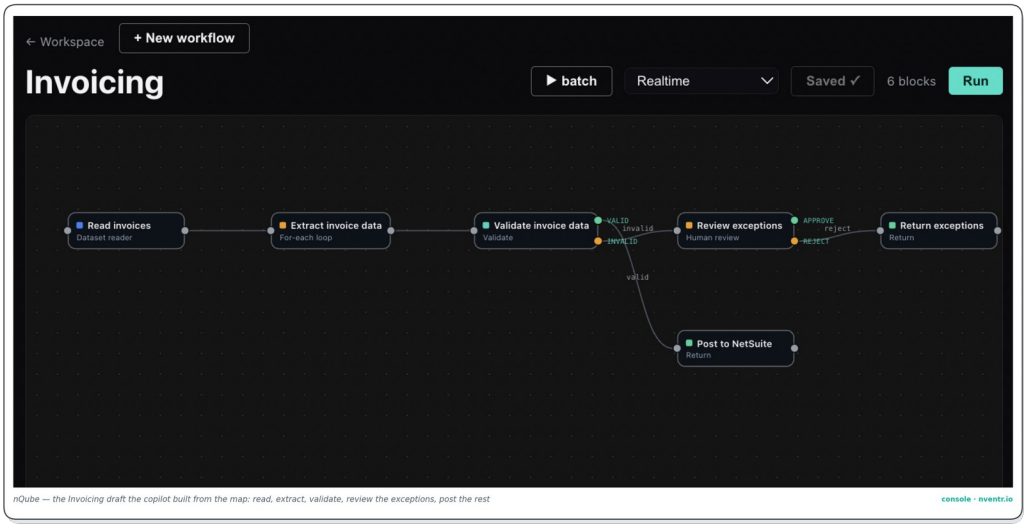
Your first workflow on nQube

nQube is the workflow engine: a canvas of blocks — inputs, logic, model calls, integrations, approvals, outputs — connected into a pipeline that runs on a schedule, on an event, or when someone asks. For a process-lever initiative, the walkthrough is: open the initiative, add a deliverable of kind *workflow*, and tell NoVA what the process is in the words from the map ("invoicing: receive the PDF, extract the fields, match the purchase order, post to NetSuite, flag exceptions for review"). NoVA drafts the pipeline from the process steps, you review it on the canvas, and you run it against a handful of real cases before switching it on. Every run is counted against the initiative from that day.

© 2026 Nventr. All rights reserved. — Map, then optimize.



From a process on the map to a pipeline on nQube — the five invoicing steps become blocks; the judgement step keeps its person, now only for the exceptions



The Invoicing pipeline on nQube — drafted by the copilot from the initiative on the map: read the invoices, extract each, validate, send the exceptions to a person, post the rest

↑ Contents

Your first app on the App Kit

The App Kit builds complete applications — database, API, interface, agent, hosting — from a design you shape with NoVA in the App Builder. For a replace or new-software initiative, the walkthrough is: add a deliverable of kind *app*, open the Builder, and start from the system you are replacing — its records become the data model, its screens become the first version of the interface, the integration to the old system becomes the migration path. The app runs on your own infrastructure or on nventr's, with your data in it either way. Its running cost shows up in the initiative's realized column automatically, so the saving is net.

© 2026 Nventr. All rights reserved. — Map, then optimize.

↑ Contents

Connecting what you keep

Not every system gets replaced; most get kept and connected. Integrations give workflows, apps and agents authenticated access to the systems on your map — to read, to write, to listen for events. A kept system with an integration becomes a block on the canvas and a tool an agent can use. Map the integration back to the system record so the Blueprint knows which systems are reachable.

↑ Contents

Your first model in the Training Center

For a model-lever initiative, the Training Center takes a dataset from one of your data stores, trains a model on it — classification, extraction, regression, a fine-tuned language model — evaluates it against held-out records, and publishes it as a block for workflows and a tool for agents. The walkthrough is: add a deliverable of kind *training run*, point it at the dataset the initiative targets, and let NoVA propose the task and the evaluation. Nothing trains on a store marked confidential or regulated without the permission the data map requires.

↑ Contents

Your first agent

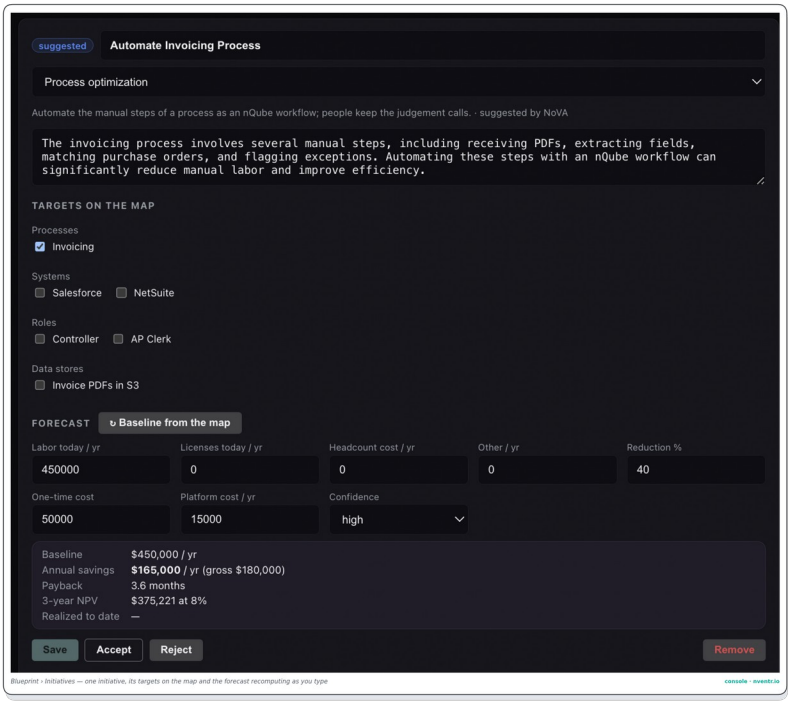
An agent on nventr is a scoped worker: a role it plays, the knowledge it may read, the tools it may use, the budget it may spend, and the person it reports to. For an augment initiative the agent sits beside the role — it prepares and the person decides; for a plug-in-agent initiative it runs the process within its scope and escalates what it cannot settle. Build it from the initiative: add a deliverable of kind *agent*, name the role and the processes from the map, and NoVA drafts the scope. Safety lives in what the agent is allowed to touch, never in the wording of its instructions.

↑ Contents

The initiative over time

Every initiative moves through the same statuses: **suggested** (NoVA proposed it) → **accepted** (a person said yes) → **planned** (milestones set) → **building** → **live** → **measuring** → **done**, or **rejected** at any point with a reason. Milestones carry due dates and completion; deliverables carry links to what was built. The Initiatives tab is the plan; filter by status, sort by payback, and see at a glance what is waiting on a decision.

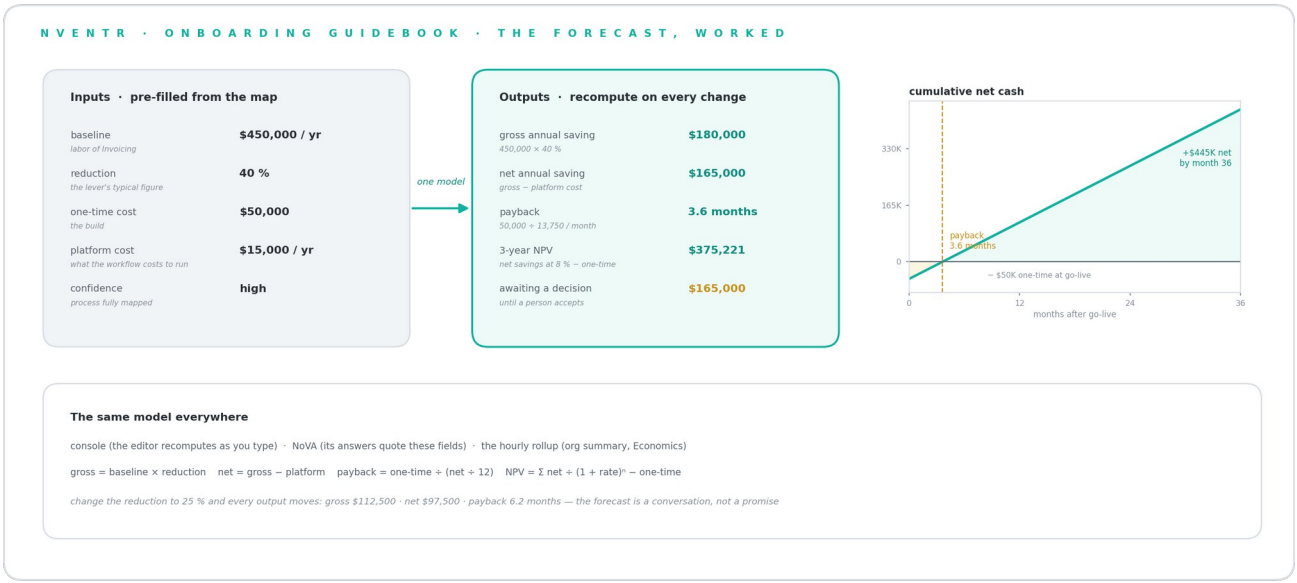
© 2026 Nventr. All rights reserved. — Map, then optimize.



One initiative in the console — its lever and rationale, its targets on the map, the forecast inputs and the outputs that recompute as you type, and the decision buttons only a person may press

↑ Contents

The forecast



The forecast, worked — five inputs pre-filled from the map, six outputs from one model, and the cumulative cash curve crossing zero at payback

© 2026 Nventr. All rights reserved. — Map, then optimize.

Every initiative's forecast has the same five inputs and the same six outputs, and the inputs are pre-filled from the map the moment the initiative is created.

Inputs: **baseline** (labor, licenses, headcount and other yearly cost the initiative targets — summed from the targeted processes, systems and roles), **reduction** (the percentage of the baseline the lever removes; the lever's typical figure until you change it), **one-time cost** (the build), **platform cost per year** (what the deliverables cost to run), and a **confidence** (low, medium, high).

Outputs: **gross annual saving** (baseline × reduction), **net annual saving** (gross minus platform cost), **payback** in months (one-time cost divided by monthly net saving), **three-year NPV** (net savings discounted at the rate your organization sets, minus the one-time cost), and the rollups across the plan — total forecast annual savings, total one-time investment, and the savings still **awaiting a decision** in suggested initiatives.

Change any input and every output recomputes on the spot; the model is the same in the console, in NoVA's answers and in the hourly rollup, so the numbers never disagree with each other.

↑ Contents

The realized column



Forecast against realized, month by month — measured savings from what ran, what a person recorded on top, and the ledger the Impact tab keeps

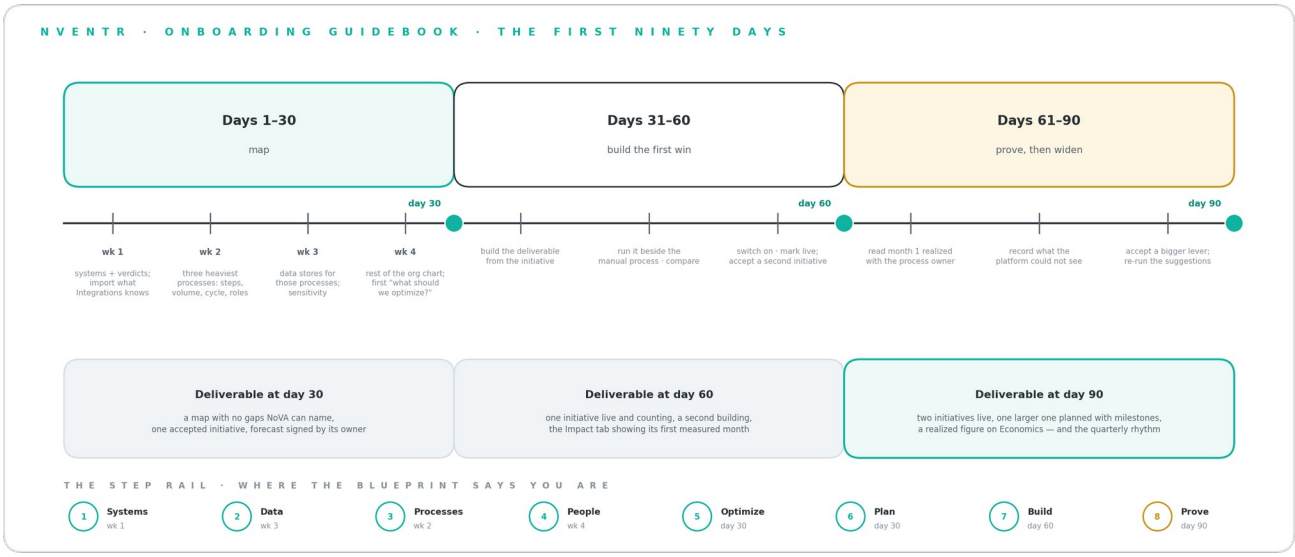
From the month an initiative goes live, the platform measures it. For each workflow deliverable it counts the runs that month; one run is one instance of the targeted process, and the hours it displaced are the process's cycle time times the initiative's reduction, priced at the loaded rate of the roles that used to do it. For each app deliverable it takes the app's running cost from the month's cost rollup. Gross saving minus platform cost is the month's **measured** realized saving, written beside the forecast with its basis stated. Anything the platform cannot see — a license cancelled, a hire deferred — a person records for the month with a note and their name.

© 2026 Nventr. All rights reserved. — Map, then optimize.

The Impact tab shows forecast against realized by month, by lever and by initiative, and the Economics module carries the plan's savings beside the month's margin. The gap between the columns is not a failure; it is the next conversation — a reduction that was optimistic, a process that had more volume than mapped, a saving that needs a line a person has to add.

↑ Contents

Days 1–30: map



The first ninety days — three bands, what happens each week, the deliverable at each milestone, and where the step rail says you are

Week one: systems, with verdicts; import what Integrations already knows. Week two: the three heaviest processes, fully — steps, volume, cycle time — and the roles that run them with loaded cost. Week three: data stores for those processes, with sensitivity. Week four: the rest of the org chart, and the first "what should we optimize first" — expect four to six suggested initiatives. Accept one process-lever or augment-lever initiative with payback under six months. Deliverable at day 30: a map with no gaps NoVA can name, and one accepted initiative with a forecast the owner has signed.

↑ Contents

Days 31–60: build the first win

Build the first initiative's deliverable — a workflow or an agent — with NoVA from the initiative itself. Run it beside the manual process for a week; compare. Switch it on; mark the initiative live. Meanwhile accept a second initiative from the suggestions, ideally one that reuses the integration or the dataset the first one created. Deliverable at day 60: one initiative live and counting, a second in building, and the Impact tab showing its first measured month.

↑ Contents

© 2026 Nventr. All rights reserved. — Map, then optimize.

Days 61–90: prove, then widen

Read the first realized month against the forecast with the process owner; adjust the reduction if the measurement says so. Record anything the platform could not see. Use that number to accept the first bigger lever — a model, or a system replacement whose renewal is coming. Re-run the suggestions; the map has changed. Deliverable at day 90: two initiatives live, one larger initiative planned with milestones, a realized figure on the Economics module, and a quarterly rhythm — map, suggest, accept, build, measure — that the organization now runs itself.

↑ Contents

Where to click

Everything in this book is one place in the console: **Blueprint**, in the Build section of your organization. Its tabs are the chapters — Map, Systems, Data, Processes, People, Initiatives, Impact — and its step rail shows where you are on the path. The first time you open it NoVA offers to interview you. Say yes.

↑ Contents

© 2026 Nventr. All rights reserved. — Map, then optimize.