

eBook

How to Innovate with AI: A Practical Guide for SMEs and Enterprises

Get Your Organization Ready for the Future

By Nicole Reineke

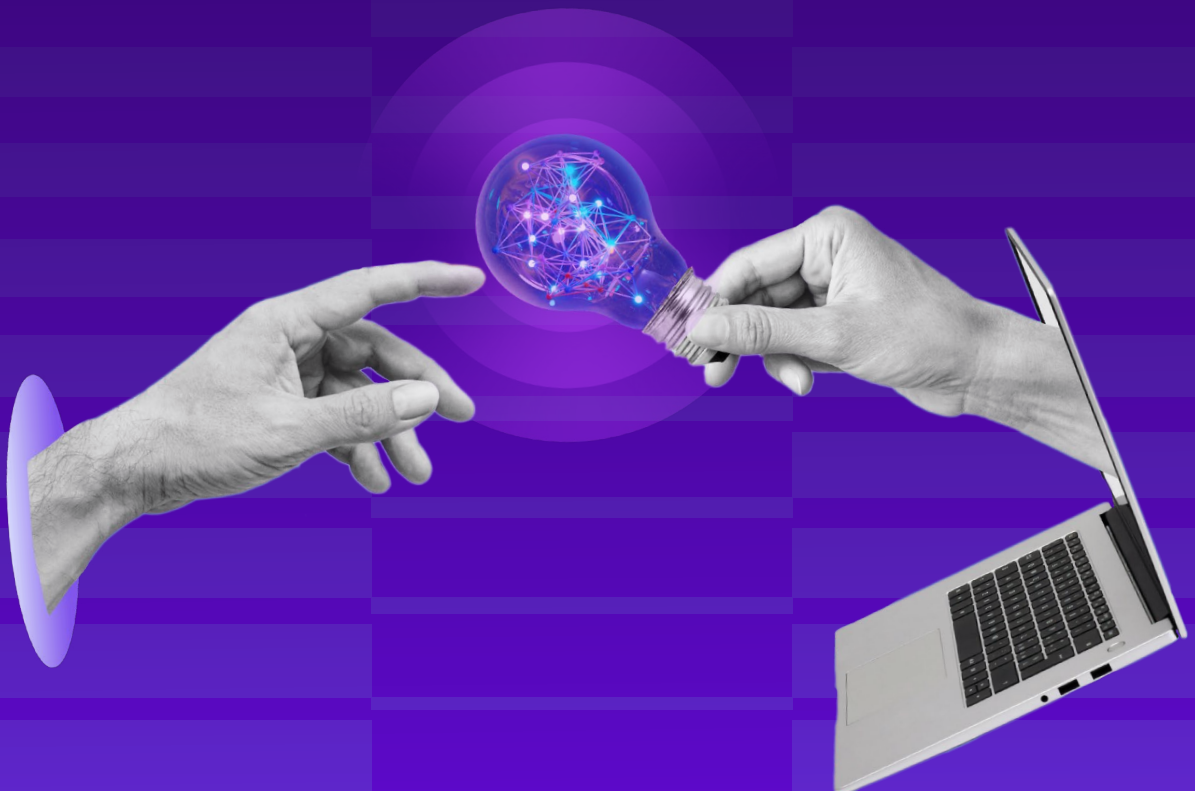


Table of Contents

Introduction: Why AI? Why now?	03	Quick wins (crawl): No-code adoption	22
Understanding the AI landscape	05	Intermediate projects (walk): Tuned models and integrated workflows	25
What are AI agents?	05	Advanced programs (run): Custom agents, automation, predictive models	27
The three experiences of AI: Choosing the right “face” for the job	06	Aligning near-term value with long-term vision	29
Collaborative AI: Assisting your people	06	Measuring success	30
Personalized AI: Adapting to context	07	AI Risk	32
Proactive AI: Acting before humans intervene	07	Establishing AI pillars: Consent, transparency, and context	32
Garbage in, garabage out	08	Setting organizational ownership and governance	33
Do you use, tune, train, or build? Knowing where to start	09	From project to practice: Making AI part of how you work	34
Approaching AI innovation	11	Step 1: Define the AI system lifecycle	34
Innovation	12	Step 2: Create AI-ready workflows	35
Problems	13	Step 3: Train for AI fluency	36
Finding the \$10,000+ use case	14	Step 4: Track and evolve usage over time	36
Example use cases for mid-sized organizations	15	Conclusion	37
Fast iteration innovation process	19		
Journey maps and points of friction identification	19		
Crawl, walk, run – Your roadmap	22		

How to Innovate with AI: A Practical Guide for SMEs and Enterprises

Get Your Organization Ready for the Future

By Nicole Reineke

Introduction

Why AI? Why now?

If you're human, you're probably feeling mildly overwhelmed by all the AI news filling your feeds. New products. Claims that AI will replace all coding in the next few months. Major vendors running models on edge devices without internet access. Headlines about AI-driven medical breakthroughs. It's enough for even someone who studies this space daily to feel like there's no way to catch up.

If you're overwhelmed, it's understandable. AI is moving fast, but it hasn't passed you by. What we need now is to start learning and actively looking for ways to apply AI to our companies, teams, and our own work. Specifically, to identify points of friction and automate pieces of our business.

For example, let's say you work at a large enterprise that hasn't adopted AI at an org-wide level. You might think the AI solution you need is to build an AI agent that scrapes all team chat conversations from one group, correlates Confluence pages and Jira tickets from a specific time period, adds in relevant email threads, and automatically outputs an executive-ready monthly report. And anything less feels like failure.

In many cases, that kind of automation isn't achievable today. The data isn't centralized or clean. The internal tooling doesn't exist yet. But what is achievable is this: you can manually copy some of the data, use an agent to pull other pieces, and combine them to get a solid first draft. Then you review it using your expertise and tailor it to the team who needs it.

A two-hour task becomes a twenty-minute one. And that's a high-value use of both your time and the tools. This is the difference between setting your North Star (a necessary goal) and finding your Crawl Path (the first stage of adoption).

Extend this to your department. Perhaps you are leading a support team, managing endpoint security, or running internal operations for dozens of business units. You are likely facing increasing complexity with the same or fewer resources.



AI is not going to replace your people, but it can extend their reach. When implemented well, AI can reduce task time and protect your team's energy and focus.

For organizations struggling with IT technician scarcity, this matters. Every unnecessary ticket, every redundant task, every alert that could have been filtered is another drain on limited capacity. A successful AI rollout will make your best people more effective and, in the long term, it will help with retention. When technicians can spend more time solving real problems and less time on triage or repetition, their job becomes more rewarding, not more replaceable.

This book will walk you through a new way to think about innovation and AI adoption. A way that's strategically innovative and wildly practical. Built on a proven innovation process designed to help you succeed with AI.

Let's start with the basics so we are all using the same terms.



Understanding the AI landscape

Artificial Intelligence is the umbrella term for a field of research related to technology that mimics human intelligence— things like learning, reasoning, and perception. Imagine it as the broad concept of machines being able to “think”.

Machine learning (ML) is a subset of AI. It's how machines learn patterns from data. If you've ever seen your email automatically tag a message as “promotions” or “spam,” that's ML at work. It learns from large volumes of labeled or unlabeled examples to identify patterns. In some cases, it correlates patterns with specific outcomes. Once trained, models can classify new, unseen data by comparing it to learned patterns, making what we often call “smart guesses” based on probability, not certainty.

Generative AI (GenAI) is where things start to feel futuristic. I would venture to guess most of the readers have already been using it. GenAI creates new content (text, code, images, music) based on patterns it's learned. It doesn't just label or sort; it makes. ChatGPT, Midjourney, and GitHub Copilot are all GenAI. They don't have true understanding, but they use math and plotting to predict what might come next, and they're shockingly good at it when given the right prompts.

What are AI agents?

AI agents take things a step further. Instead of just generating a response, they interpret what you want, decide what actions to take, may call tools to take actions, and execute workflows using AI as part of the engine. Think of them as little task-completing co-workers. And they can do this without requiring that you stand up a monolithic architecture.

For example, a GenAI chatbot might write a paragraph. An orchestrator AI agent might listen to your request, send a request to a research agent to pull the relevant data, and summarize it. A visualization agent can create a visual in your brand aesthetic and call an email tool to send the result without you needing to say “Please generate a chart and format it for marketing.” It senses, reasons, and acts.

Agents can work alongside you, run quietly in the background, or be triggered by specific events in your systems. And right now, the most effective ones are the ones you configure carefully with the data and tasks they’re responsible for.

The three experiences of AI: Choosing the right “face” for the job

When you start thinking about applying AI in your business, you’ll quickly realize that not all AI feels the same. That’s because it’s not just what the AI does, it’s how it shows up.

There are three main “faces” of AI you’ll see in real-world use. Each has a different experience, different level of effort, and a different role to play. The key is to match the right face to the right problem.

Let’s break them down.

1 Collaborative AI: Assisting your people

This is where most people start. Collaborative AI works with you. It helps you do what you were already going to do; just faster, smarter, or with less friction.

Examples:

- ▲ Copilots that draft emails, reports, or ticket responses
- ▲ Chatbots that suggest answers while you’re typing
- ▲ Agents that summarize meetings or suggest tags based on content

This is your “AI-powered assistant.” It still needs you. It enhances productivity, reduces context switching, and can be inserted into daily workflows with little disruption.

When to use it: You’ve got skilled people doing repetitive, time-consuming work that still needs their judgment. The task doesn’t need to be taken over, it just needs to be lightened.

2 Personalized AI: Adapting to context

This is the next level. Personalized AI adapts to the user, the moment, or the surrounding data. It tailors outputs based on who is asking, what they've done before, what's happening now, or what matters most to them.

Examples:

- ▲ Dashboards that show different KPIs depending on the user's role
- ▲ Recommendation engines that get better the more you use them
- ▲ Support tools that adjust suggestions based on your customer segment, risk score, or account history

This is not just "smart", it's situationally aware. It makes users feel like the AI knows them.

When to use it: You want AI to drive better decisions or actions based on real-time relevance. The same answer won't work for everyone or every situation.

3 Proactive AI: Acting before humans

This is the holy grail for most organizations; AI that acts. It senses patterns, predicts what's next, and takes action before a person even notices a problem.

Examples:

- ▲ An agent that automatically detects drift in configuration before it causes an outage
- ▲ Security AI that flags and quarantines a threat before an alert is triggered
- ▲ Scheduling tools that proactively shift resources based on forecasted workload

This AI removes friction entirely. It doesn't wait for input. It's anticipatory.

When to use it: The cost of delay is high. The risk of a bad decision is low. Or the problem is invisible until it's already done damage. You need eyes on the future and actions taken early.

You don't need to pick just one. A great AI experience may include all three, layered together. But know which one leads and build for that.

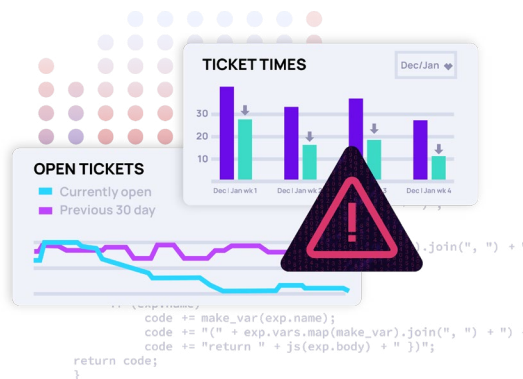
If your AI shows up in the wrong face, it creates confusion. Too proactive, and it feels invasive. Not proactive enough, and it feels like old tech with a new label.

Get the face right, and the rest falls into place.

Garbage in, garbage out

Let's clear something up: AI is not magic. It's math. And while the outputs can sometimes feel magical, predictive, even eerie in how well they match what we're thinking, they are still just the result of models trained on data. Which brings us to one of the oldest, truest sayings in computing: garbage in, garbage out.

If the training data is biased, incomplete, outdated, or just wrong, then the AI's output will reflect that. It doesn't fix bad information. It amplifies it, polishes it, and confidently hands it back to you.



Let's say you feed your model inconsistent ticket notes, vague status fields, and knowledge base articles written in 2014. Then you ask it to generate root cause summaries or recommend next steps. You'll get outputs that will appear helpful, but they will be wrong.

This is especially important for Generative AI. These models are trained to predict the next most likely word, phrase, or pattern, not the "right" one. They can only reflect what they've seen before. If what they've seen is skewed, the results will be skewed. If what they've seen is brilliant and structured and annotated with care, then yes you can get powerful, context-aware outputs. There are approaches where you can have models and agents check each other for accuracy, but in the absence of good data, it's an exercise in insanity.

So, before you dive into building the next brilliant AI solution, ask yourself a few honest questions:

- ▲ How much data do I have?
- ▲ Is my data consistent?
- ▲ Is it accurate?
- ▲ Is it recent?
- ▲ Is it labeled or structured in a way a model can understand?
- ▲ Would I feel confident making a business decision based on the trends in this data?

If not, you've got a data problem, not an AI problem.

This book isn't going to pretend that data prep is glamorous. It isn't. But it is the foundation of every successful AI initiative. Treat your data with the same care you treat your strategy. Because if the foundation is weak, nothing you build on top of it will stand.

The good news is: you don't need perfect data to start. You just need to know what your data is, where it lives, what it represents, and whether it tells the truth. Then make a decision on whether you need to buy someone else's product instead of building one yourself. Because no matter how smart your model is, it can't overcome broken inputs.

Do you use, tune, train, or build? Knowing where to start

Once you've done the honest work of assessing your data, the next question comes up fast: Do I just use AI off the shelf? Do I tune it? Do I need to train my own model? Do I have to build this thing from scratch?

Here's the good news: you almost never need to build from scratch.

Let's break it down. Think of these four options like levels of customization and complexity. Where you fall depends on your use case, your data, and your available skills.



This is the crawl stage. You use existing tools like ChatGPT, Microsoft Copilot, Google Gemini, or AWS Bedrock-backed applications. These come pretrained and ready to go. You enter a prompt or feed them simple context, and they do the work.

Perfect for:

- ▲ Summarizing meetings
- ▲ Generating first drafts
- ▲ Answering customer questions with existing content
- ▲ Exploring ideas
- ▲ Enhancing ticket notes

Start here unless you have a reason not to. Seriously.



Tune

This means you're still using a pretrained model but now you're nudging it to perform better for your domain. Think of it like customizing a template. You're not rebuilding the engine. You're feeding it your documentation, sample conversations, or labeled examples to make the results more relevant.

Perfect for:

- ▲ Custom chatbots
- ▲ Industry-specific recommendations
- ▲ Internal copilots that know your policies or product names
- ▲ Teams who don't write code but have strong internal data and structure

Most GenAI tools today support tuning via file upload, URL scraping, or embedding context. Many major cloud providers have AI studios that will do the same for many ML models—no ML engineer required.



Train

Now you're entering deeper water. Training means you're taking a base model and teaching it new capabilities or improving performance with large, high-quality data sets. This is time and compute-intensive and usually requires expert help.

Perfect for:

- ▲ Organizations with a lot of proprietary data
- ▲ Highly specialized use cases (like identifying anomalies in radiology images or financial fraud in a specific vertical)
- ▲ Scenarios where existing models perform poorly due to gaps in public training data

If you don't already have 10,000+ good records, look before you leap. This is not a weekend project.



Build

This is the moonshot. You're designing the architecture, selecting the algorithms, sourcing and labeling the data, and handling the training pipeline. It's rare, expensive, and risky but sometimes necessary if you're solving a novel problem no existing model can touch.

Perfect for:

- ▲ Research institutions
- ▲ Tech companies building core AI products
- ▲ Situations where compliance or IP ownership rules out third-party models entirely

If you're even considering this, you already have a team of data scientists, ML engineers, and lawyers in the room. Otherwise, don't.

When in doubt, default to Use or Tune. You can deliver 80% of the value at 20% of the cost. **Save Train and Build for the cases where there's no other path.**

Next up, we'll talk about how to identify where to apply AI inside your business and how to spot the difference between "cool" and worth doing.

Approaching AI innovation

At this point, you've got the basics down. You know what AI is, how machine learning and Generative AI fit into the picture, and what it looks like when AI shows up in different "faces": collaborative, personalized, or proactive.

Now comes the shift. The real work of innovation doesn't start with technology. It starts with intention.

Innovation

Let's be clear about what innovation really means when we talk about AI.

It's not about chasing trends or building something flashy for the sake of showing progress. Innovation, especially in the context of AI, solves meaningful problems in better ways. It creates lasting change in how work gets done to improve outcomes and enhance experiences.

AI is a tool. Innovation is how you use it.

To bring structure to that idea, we use the Innovation Matrix. This framework helps you think about the kind of change you're driving and where it sits in your strategy.

Here's how it breaks down:

- ▲ **Basic Research is foundational.** You're learning. You're exploring. You're scanning for signals and subscribing to the right sources so you don't fall behind.
- ▲ **Sustaining Innovation improves what already exists.** You're making your core offerings stronger, faster, or easier. This is where AI shines in internal automation, ticket deflection, or speeding up customer responses.
- ▲ **Disruptive Innovation creates new paths.** You reimagine workflows, pricing models, or service delivery. AI might open up new segments or automate tasks that were never scalable before.
- ▲ **Breakthrough Innovation solves hard, cross-functional challenges.** It's where collaboration across departments or industries leads to solutions no one could have built alone.

In practice, most successful AI innovation happens through repeatable improvements on a point of friction.

A **point of friction** is any moment in a workflow, customer experience, or internal process where momentum slows, effort increases, or results fall short. It is the place where things break down, fall through, get delayed, or demand more energy than they should.



Friction is what makes work feel harder than it needs to be. It's what causes your team to open a second browser tab to copy data from one system to another. It's the reason a technician rewrites the same response for the tenth time. It's the cause of delays when a customer waits for help because the right person didn't get notified.

Friction doesn't always look like failure. In fact, most of the time, the job still gets done. But it gets done inefficiently, with frustration, repetition, or risk.

**Friction is your entry point
for innovation.**

**You find a friction point. You test a smarter
way to deal with the problem. You show
that it works. Then you build on it.**

Innovation doesn't always require big ideas. It requires useful ones. It must drive value. If your application of AI improves performance, removes a barrier, or reveals something your team couldn't see before, that's innovation.

So, before you build anything, pause and ask: Where does this idea fit?

Is it just interesting, or is it fixing something that slows you down every day?

Are you using AI to keep up or are you using it to level up?

The strongest AI innovation starts by being brutally honest about the problem. What's the pain? Who feels it? What's the cost of doing nothing?

You are probably wondering: How do we find problems?

Problems

You want to start by identifying problems that are significant enough to warrant the time, energy, and change that comes with implementing an AI-powered solution. You need to understand what you're solving for, and why, before you even consider how AI might help. Otherwise, you risk building something that looks impressive and solves nothing.

This book is not about building AI for the sake of AI. It's about applying intelligence, human and machine, to the right places.

So, before you open a tool or type a prompt, ask:

- ▲ Where are we wasting time?
- ▲ Where are we losing momentum?
- ▲ Where are customers stuck, confused, or underserved?
- ▲ Where are our people doing repetitive tasks that make them feel less human?
- ▲ Where do we drown in data, but lack actionable insight?

- ▲ Where do we have multiple data sets that could enrich each other through correlation?
- ▲ Where do we need early warning?
- ▲ Where can we personalize our end users' experiences?

This is how you start. Not by imagining what's possible, but by facing what's painful.

Once you've got a real problem in your hands, then (and only then) do we walk through how to apply AI tools, techniques, and frameworks to explore solutions. We'll show you how to map the friction, test ideas quickly, build only what's needed, and measure what matters.

Finding the \$10,000+ use case

Innovation often fails because it solves an imaginary problem or solves a real problem no one actually cares about.

So how do you find the ones worth doing?

You start by asking a brutal question: What's the cost of not solving this?

If the answer is "Nothing," walk away. If the answer is "We waste hours, lose revenue, frustrate customers, or burn out employees," then you might be onto something.



Here's a fast way to gut-check whether an AI idea is worth your time:

- ▲ **Volume:** Is the task repeated hundreds or thousands of times a week? High repetition is low-hanging fruit.
- ▲ **Pain:** Is the task annoying, confusing, or disliked by everyone who does it? Even small wins here create real value.

- ▲ **Risk:** Does getting it wrong have low impact? AI is probabilistic. It's great for triage, not judgment calls.
- ▲ **Friction:** Is the process slow, manual, or filled with copy/paste? That's a flag.
- ▲ **We can't without It:** Are there tasks you literally can't do today because there's too much data or not enough people?

If you can check two or more of these boxes, you may have a real candidate.

Now flip the lens. Watch out for:

- ▲ **Vanity projects:** If it sounds good in a keynote but no one would use it on a Tuesday, skip it.
- ▲ **One-time tasks:** Don't build automation for something that only happens once.
- ▲ **Over-engineering:** If a rule or script can do it just fine, do not layer in AI.
- ▲ **Unclear ownership:** If no one is excited to use or maintain it, it will die.

One simple trick: walk through your current workflow and ask, "What is annoying, boring, or slow?" Then ask, "Could a co-pilot help me do this faster, or could an agent do it without me at all?"

You don't need to start with the moonshot. Start with the friction you already know and solve that. The best AI projects don't feel like AI. They feel like relief.

Example use cases for mid-sized organizations

Let's run some common use cases through the \$10,000 test.

You don't need spreadsheets and forecasts to start. You need five brutally honest questions.

Use these to filter the real opportunities from the "cool but not critical" noise.

The \$10,000 idea test

- 1 Is the problem painful enough? Does it drain time, money, morale, or customer trust? Minor annoyances don't justify major innovation.
- 2 Is it repeated often enough? One-time issues rarely deliver scalable ROI. You want something that happens weekly, daily, or constantly.
- 3 Is it specific enough to solve? You need clear inputs, outcomes, and constraints. Vague ideas like "make reporting better" don't pass.

- 4 Can we actually affect the outcome? If solving this requires restructuring the whole company or changing user behavior you can't control, it's a no.
- 5 Would someone pay \$10K to make this go away? If the answer is yes, either in direct savings or unlocked growth, then you're onto something.

Streamlining internal operations

Example: AI agent that pre-tags incoming support tickets and drafts responses for tier-one issues.

- ▲ Painful? Yes. Manual triage eats up valuable technician time.
- ▲ Repeated? Constant. Every support queue gets flooded.
- ▲ Specific? Very. Structured data, clear workflows.
- ▲ Controllable? Absolutely. You own the tools and the team.
- ▲ Worth \$10K? Easily. You'll get there in labor savings alone.



This passes the \$10K test.

Automating customer service and support

Example: Chatbot that answers common billing and login questions.

- ▲ Painful? Yes, for both customers and staff.
- ▲ Repeated? Daily. These are your most frequent inquiries.
- ▲ Specific? Yes. FAQs make for perfect training content.
- ▲ Controllable? Yes. You own the service scripts.
- ▲ Worth \$10K? Depends on your volume, but in most orgs, yes.



This is a solid candidate, especially if you scale support.

Enhancing sales and marketing

Example: AI-driven personalization engine that tailors outbound emails based on customer profile and product usage.

- ▲ Painful? Depends. Is your pipeline leaking due to bad targeting?
- ▲ Repeated? Yes. Campaigns run monthly or continuously.
- ▲ Specific? It can be, if your data is clean.
- ▲ Controllable? Partly. You'll need to integrate with your CRM and usage data.
- ▲ Worth \$10K? If conversion improves even slightly, probably yes.



This passes, but only if your data hygiene and segmentation are strong.

Generating insights from unstructured data

Example: AI-driven personalization engine that tailors outbound emails based on customer profile and product usage.

- ▲ Painful? Yes. Insights are buried and slow to surface.
- ▲ Repeated? Yes. New logs and transcripts appear every day.
- ▲ Specific? Sort of. Depends on how messy the data is.
- ▲ Controllable? Mixed. You may need cleanup first.
- ▲ Worth \$10K? If it improves root cause detection or roadmap planning then yes.



This is a high-potential project, but only if you're ready to invest in some data prep.

Phishing email triage and prioritization

Example: AI that reviews suspected phishing emails reported by users, compares them to known attack patterns, checks embedded links or attachments, and prioritizes which incidents need human investigation.

- ▲ **Painful?** Yes. IT and security teams waste time manually reviewing phishing emails, many of which are spam or low-risk. Meanwhile, high-risk messages can sit unreviewed for hours. Every minute matters in an active phishing attack.
- ▲ **Repeated?** Absolutely. Users report suspicious emails every day. And threat actors keep changing tactics, making the manual process harder over time.
- ▲ **Specific?** Yes. You have structured email metadata, known malicious indicators, and access to blocklists or threat intelligence feeds. The AI can learn which traits matter most.
- ▲ **Controllable?** Yes. You control the inbox, the intake process, the classification thresholds, and the rules for escalating real threats. AI decisions can be checked, scored, and improved over time.
- ▲ **Worth \$10K?** Without question. Even a single high-fidelity phishing campaign that slips through can cost far more in downtime, remediation, or exposure. Automating the triage process protects people, data, and reputation and saves hours per week.



This passes the \$10K test. It improves both defense and productivity.

The Bottom Line:

**Run every AI idea, no matter how shiny, through this test.
If it doesn't hurt, repeat, clarify, or deliver, you're probably
looking at a distraction, not a breakthrough.**



Fast iteration innovation process

Pick one of your \$10,000 problems. Before you build anything, you need to understand the experience you're trying to improve. That means mapping it.

Journey maps and points of friction identification

Start by building a journey map. List the real people involved in the process. And the steps that are happening. Not just titles. Think like a documentary filmmaker. Who's doing the work? Where are they stuck?

Draw the circle, not the line

When people first hear "journey map," they picture a left-to-right timeline. A linear path from start to finish. But in the real world, most business processes don't move like that. They loop. They reset. They recycle. They evolve.

That's why we use a proto journey map, a quick, low-fidelity way to visualize the real flow of work, and we draw it as a circle.

Why a circle?

By mapping in a circle, you give yourself permission to revisit and re-enter the process from any point, and you immediately see where AI can make a difference—not just once, but every time the loop spins.

How to build a proto journey map

This isn't about perfect visuals or formal workshops. This is fast, messy, and powerful.

- 1 Pick one core, specific process**
 Something like onboarding, triaging a ticket, running a QBR, or deploying a patch.
- 2 Identify the key roles involved**
 Think about who participates. Not just their title, but what they actually do. Technician. Customer. Sales lead. Account manager. Automation script. These are your actors.
- 3 Draw the steps (roughly) in a circle**
 Use sticky notes or a whiteboard. Ask:
 - ▲ What kicks this off?
 - ▲ What happens next?
 - ▲ Where does it loop?
 - ▲ Where does it stall?
 - ▲ How does it end or restart?

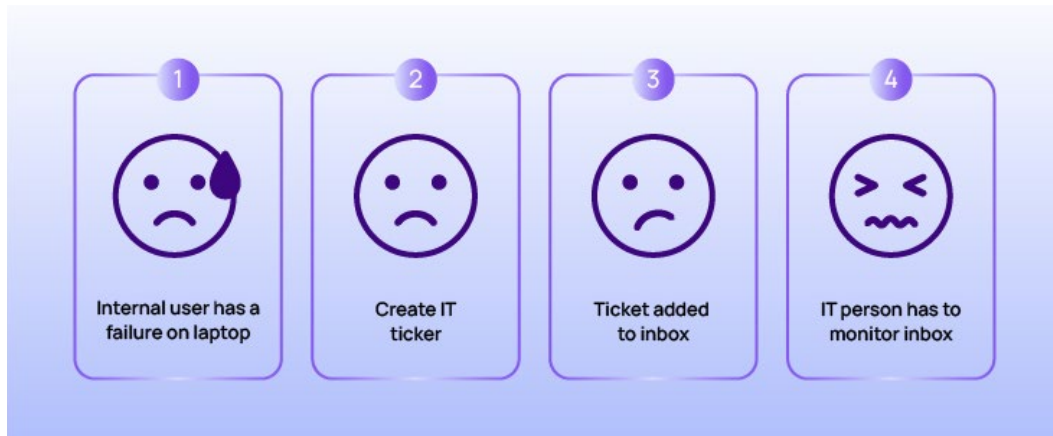


4

Mark the friction points

Look for:

- ▲ Delays
- ▲ Copy-paste work
- ▲ Rework
- ▲ Handoffs that fail
- ▲ Systems that don't talk to each other



5

Tag with AI opportunity types

Ask:

- ▲ Could this be automated?
- ▲ Could this be predicted?
- ▲ Could this be summarized or clarified with AI?
- ▲ Could this be personalized based on who's in the loop?

Why this works

You're not trying to capture every detail. You're trying to surface where the pain is and where intelligence (human or machine) can step in to improve the flow.

This approach does three things:

1. **It makes invisible work visible.**
2. **It helps teams align on what's broken.**
3. **It shows you where recurring value can be created.**

A one-time fix is helpful, but fixing something that happens hundreds of times? That's innovation. And when it sits on a loop, you see the return every time the cycle completes.

Crawl, walk, run – Your roadmap

Not every idea should start at full scale. The smartest AI teams know how to prove value fast, then scale when it works.

Quick wins (crawl): No-code adoption

Start here. This is your fast lane to proof of value. You don't need a data scientist. You don't need custom models. You just need a real workflow, a real pain point, and an off-the-shelf tool.

These are projects you can implement in days or weeks, not months. They work best when the process is repetitive, predictable, and easy to isolate. The value is immediate, and the learning starts on day one.

Use off-the-shelf tools like ChatGPT, Microsoft Copilot, or Google Gemini. Focus on tasks that are easy to isolate and measure:

- ▲ Drafting customer emails
- ▲ Generating meeting recaps
- ▲ Summarizing ticket history
- ▲ Auto-tagging support issues

These projects require little or no integration and give you fast feedback.



In the above example, we can use AI to do things like:

▲ **Drafting customer emails**

Use GenAI to generate responses to common customer questions. Have your team review, tweak, and send. Save minutes per ticket and improves tone consistency.

▲ **Summarizing ticket history**

Copy ticket comments into a GenAI tool to generate a clean summary for handoff. This eliminates redundant review and improves team alignment.

▲ **Auto-tagging support issues**

Feed past cases into an LLM prompt and ask it to generate standard tags or categorize based on your taxonomy. Even a copy/paste workflow can improve visibility.

Low-code examples

Low-code tools give you slightly more power. like adding buttons, simple automations, or connecting systems with logic. These still avoid complex integrations but allow you to embed intelligence inside real workflows.

Using the use case above, we could change our story to be aligned to:

Use Case: Laptop Failure Self-Service and Ticket Triage

Persona: Internal user and IT technician

Pain Point: Every ticket starts manually. Users wait. IT spends time just figuring out what the issue even is.

Low-Code AI Journey:

1

User sees an issue on their laptop

- ▲ The screen provides a **prompt**: "Need help? Click here."
- ▲ Low coding required, this is a basic modal or system tray message.

2

AI prompt opens Claude or another LLM

- ▲ User describes the issue or pastes an error message.
- ▲ Claude looks up the error. This can be connected via API or just linked through an in-browser window.

3

AI suggests remediation steps

- ▲ Claude responds with 2–3 suggestions.
- ▲ If the user succeeds, the ticket is avoided altogether.

4

If it fails, AI helps draft the ticket

- ▲ Claude asks: “Do you want to create a ticket? Here’s what to include.”
- ▲ The system then opens a ticket with pre-filled fields (device name, error, steps tried).

5

Technician receives contextual ticket

- ▲ AI auto-tags severity or issue type.
- ▲ Techs use a second AI agent to sort by highest priority.

This entire loop requires no model training. The only development is connecting existing prompts and actions using a low-code platform like Power Automate, Zapier, or a workflow engine in your RMM/ITSM system.

Why quick wins matter

- ▲ They de-risk AI. You learn what works without heavy investment.
- ▲ They drive adoption. People see immediate benefits in their daily work.
- ▲ They build momentum. Each success story fuels the next idea.

If you’re just getting started, this is where you begin. Crawl with confidence. Because if your quick wins don’t deliver value, your advanced AI plans won’t either.

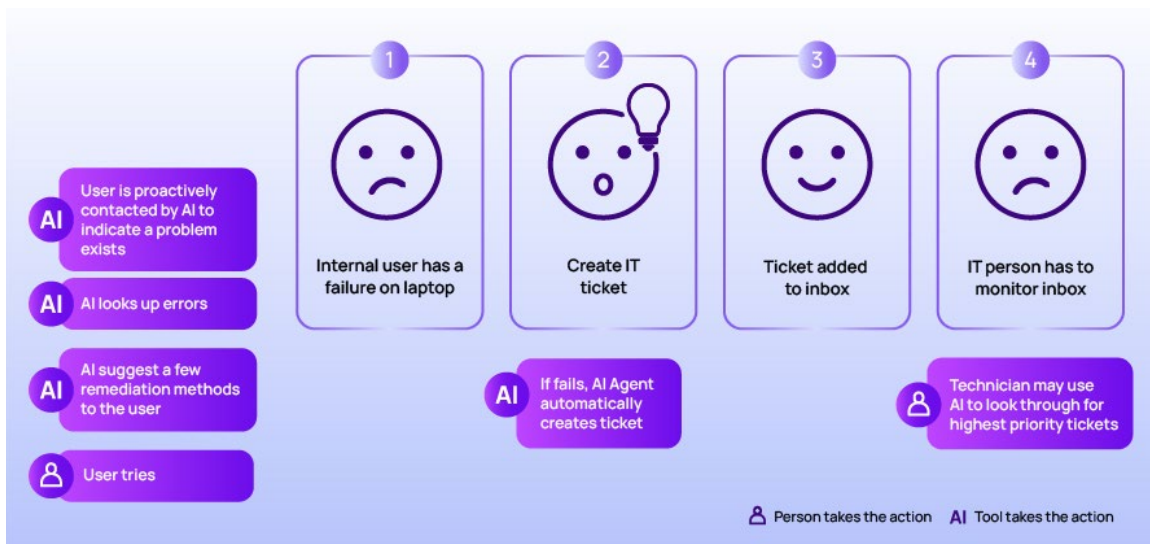


Intermediate projects (walk): Tuned models and integrated workflows

The Walk phase is where AI starts to stretch its legs. You've proven the concept in Crawl. You know the value is there. Now you integrate AI further into the workflow.

This phase is all about semi-automation. AI is acting, but it's still supervised. Humans can review, approve, or override. It's the right blend of speed and safety. Let's revisit our journey with this in mind.

Real Example: Proactive Laptop Failure Workflow (Low-Code + AI Agent Integration)



Problem:

Internal users often don't notice system issues early. By the time they open a ticket, performance is degraded. The process is reactive. Support is flooded with vague requests. Response is slow.

Walk-stage solution:

1 AI proactively contacts the user

Instead of waiting for an issue report, an agent monitors for early warning signs (e.g., CPU spikes, disk errors, failed patching). When a threshold is hit, the AI notifies the user directly.

2 AI investigates and diagnoses

The agent looks up relevant error data or logs in real time. It narrows down the likely cause, just like a technician would, but in seconds.

3 AI suggests remediation

The user receives 2–3 suggestions based on their device state and known fixes. This is powered by a tuned LLM (like Claude or ChatGPT) pre-fed with internal documentation and historical fixes.

4 User tries fix

The user attempts the recommendation. They may even be given one-click scripts or guided steps.

5 If fix fails, AI creates the ticket automatically

Instead of asking the user to start over, the agent generates a ticket, pre-filled with device info, error logs, what the user already tried, and any relevant screenshots or telemetry.

6 Technician uses AI to prioritize

On the backend, another AI agent ranks incoming tickets based on urgency, user role, or device criticality. Ensuring the right eyes are on the right issue first.

Why this is walk-stage

- ▲ AI is Embedded: It's part of the process.
- ▲ The Human Is Still in Control: Users make choices. Technicians review escalations.
- ▲ You're Using Multiple Tools: Monitoring system + LLM + ITSM platform + workflow logic.
- ▲ You're Getting Smarter Over Time: The more this runs, the better it gets at suggesting the right fix and avoiding noise.

Other walk-stage projects you can try

- ▲ AI-Powered Internal Copilots: Pull from product documentation, ticket history, and chat logs to support customer service or tech teams.
- ▲ Augmented Dashboards: Use AI to interpret metrics and highlight risks or anomalies to department leaders in plain language.

- ▲ Enhanced QBR Builders: Connect AI to your reporting tools to suggest key themes, metrics, and customer-specific recommendations based on log and ticket data.

These are all about blending human and machine intelligence. Let the AI do the heavy lifting while your team focuses on the decisions that matter.

Advanced programs (run): Custom agents, automation, predictive models

This is where AI is the system. Not something you interact with, but something that operates continuously in the background. It detects issues before they happen. It acts without asking. It closes the loop, end to end.

The Run phase is for programs that are ready to scale. The logic is mature. The data is clean. The team trusts the system. What started as a quick win is now embedded into how the organization works.

You are not guessing anymore. You are executing.

Real example: Autonomous device management

Take the Walk-stage process from earlier. You already have a technician triage assistant and proactive remediation suggestions. Now imagine that logic running on its own.

- 1 AI monitors the environment continuously**
It checks for performance degradation, drift, or signature failures.
- 2 When it sees a problem, it investigates**
It pulls the logs, looks at the pattern, compares to past fixes.
- 3 If it's confident, it resolves the issue**
It restarts the service, removes the blocked update, or rolls back the patch. No prompt, no approval.
- 4 Every action is logged**
The audit trail is attached to the ticket. The system state is recorded before and after. It's all there.

5

If confidence is low or remediation fails, it escalates

The ticket includes all the context. The technician doesn't start at zero. They pick up where the agent left off.

6

Each case teaches the model what to do next time

This isn't static automation. It's a learning loop.

This isn't theory. This is running now in high-performing organizations.

Other run-stage ideas

▲ **Predictive support**

Use AI to identify which customers are likely to open tickets next quarter. Address the issue before it lands in the queue.

▲ **Customer health monitoring**

Combine support history, usage data, and feedback scores to create real-time health indicators. Adjust outreach based on risk or growth signals.

▲ **Closed-loop compliance**

Detect misconfigurations, fix them automatically, confirm success, and document the cycle without human input.

▲ **Revenue intelligence**

Agents scan CRM, usage, and support data to find expansion or churn indicators. Feed them into the sales pipeline without a request.

When are you ready to run

Run is not where you start. It's where you arrive when you've proven the value, tuned the logic, and earned the trust of your team.

You move into run when:

- ▲ Your team is no longer reviewing every action.
- ▲ Your data is structured and stable.
- ▲ Your quick wins are delivering visible outcomes.
- ▲ Your guardrails and governance are already in place.

These programs make your operations more efficient. They change what you're capable of delivering.

Aligning near-term value with long-term vision

Once the quick wins are flowing and the larger systems are in motion, it's easy to lose focus. You start chasing ideas. Requests come in from every direction. Everyone wants AI everywhere.

That's when you pause and ask:
What are we building toward?



AI innovation isn't a collection of disconnected use cases. It's a capability. A system. A foundation for how your business operates in the future. The work you do today should unlock what's possible tomorrow.

Here's how to keep that alignment tight:

Use crawl projects to unblock future steps

- ▲ Quick wins are not side projects. They are strategic enablers.
- ▲ When you summarize tickets with AI, you're also building a dataset of structured issue types.
- ▲ When you use a chatbot for internal support, you're testing your knowledge base and seeing where gaps exist.
- ▲ When you run a pilot for proactive remediation, you're identifying which devices and configurations are agent-ready.
- ▲ Crawl is preparing the environment.

Use walk projects to train behavior and culture

- ▲ As you move into Walk, the tech becomes more integrated but so does the human. These projects introduce new expectations.
- ▲ AI is now expected to participate.

- ▲ The team starts trusting automated suggestions.
- ▲ Workflows shift from reactive to augmented.
- ▲ You're not just building systems. You're building belief.
- ▲ That belief is what enables your Run programs to succeed. If you skip Walk, your Run will stall.

Run projects drive competitive advantage

Once your agents are acting on their own, resolving issues, predicting churn, or adapting pricing strategies in real time, you're not just improving operations. You're creating distance. You are now delivering a level of service and speed your competitors cannot match unless they've taken the same path. And most haven't yet.

But this only works if your Run strategy connects back to a clear vision.

Ask:

- ▲ What will our operations look like when AI is embedded end-to-end?
- ▲ What decisions will we make differently?
- ▲ What roles will shift?
- ▲ Where will human attention move once AI takes the load?
- ▲ This is where long-term vision becomes a design principle. Every AI project is either pulling you toward that state or distracting you from it.

Measuring success

If you cannot measure it, you cannot improve it. That's true for AI just like anything else. But when it comes to AI innovation, many teams get distracted by surface metrics. They track the number of prompts, the size of a model, or how many users logged into a chatbot.

Those are not measures of success. They are activity.

What you care about is impact.

You are not here to prove that AI works. You are here to prove that AI made something better.

What to Measure

Start with this question: what changed? Then track it.

Experience metrics

- ▲ User satisfaction
- ▲ Technician feedback
- ▲ Sentiment scores from customers or employees
- ▲ AI-assisted workflows adopted over manual ones

These tell you if people trust the system and prefer to use it.

Business metrics

- ▲ Cost per ticket
- ▲ Revenue per agent
- ▲ Expansion pipeline lift from AI insights
- ▲ Decrease in customer churn
- ▲ Increase in cross-sell or upsell conversions

Do not focus on model size, response speed, or novelty. Do not reward AI for participation. Reward it for results. Most users will not care how it works. They will care that it saves them time, reduces friction, and makes them more effective.

Measure what they care about.

How to use metrics to guide your plan

Every AI rollout should have a baseline. Take a snapshot before you launch. Measure again at regular checkpoints. Use the data to decide what to expand, what to refine, and what to stop.

You are not looking for perfection. You are looking for progress that is meaningful and repeatable.

When a quick win delivers value, make it standard. When a Walk-stage project increases confidence, use that momentum to plan your Run. When a Run program proves its worth, build on it.

AI innovation is not a one-time rollout. It is a living system. Your metrics are how you listen to it.



AI risk

If you're serious about adopting AI, you must be just as serious about managing its risks. That doesn't mean overengineering everything. It means being intentional. Building the foundation that lets you scale with confidence instead of fear.

You must be aware of the legal requirements in your operating location and in the operating location of your users and the data sources. For example, if you are using customer data from the EU, you will be required to comply with the EU AI Act and any other local laws where the processing is taking place.

Additionally, you need to consider that whatever can go wrong will go wrong. So, when your AI agents are hijacked, what are the possible negative outcomes?

The question is not should you care, it's how will you lead?

Establishing AI pillars: Consent, transparency, and context

At N-able, we built our AI strategy around three core principles. They are simple, non-negotiable, and applicable across every use case.

These are our pillars:



Consent

Every user should have control over whether and how AI is used. This applies to technicians, customers, and partners. AI should never be applied by default, silently, or without choice. Users must be able to opt in, understand their options, and turn it off if needed.



Transparency

People should know when AI is being used and how. This includes what data it sees, how it makes decisions, and what happens next. Transparency creates trust. Hiding complexity behind a magical interface does not.



Context

AI should be applied in ways that respect the reality of the moment. The same action may not be right for every user, every customer, or every device. Context ensures that AI solutions are appropriate, relevant, and safe. It prevents automation from becoming reckless.

If you use these pillars to guide your decisions, you reduce risk automatically. You also increase adoption. People don't fear what they understand and control.

Setting organizational ownership and governance

One of the most common failure points in adoption of AI is that no one owns it. The model gets built. The feature goes live, but no one is watching.

To manage AI risk, you must define ownership from day one.

Start with a clear question: Who is responsible for ensuring that every AI system in production meets your company's standards for safety, accuracy, privacy, and ethics?

That person or team needs the authority to make decisions and the visibility to see across departments. They need to be involved early, not brought in to approve something that's already done.

You don't need a hundred rules. You need a small set of standards, enforced consistently. Make sure AI systems are:

- ▲ Trained on approved data
- ▲ Used within agreed risk thresholds
- ▲ Monitored for drift, abuse, or degradation
- ▲ Auditable at any time

When the AI makes a mistake, and it will, this structure ensures that someone knows, someone cares, and someone can fix it.



From project to practice: Making AI part of how you work

By now, you've done the foundational work. You've mapped the friction points. You've identified the opportunities. You've built and tested a prototype, and you've shown that it delivers value. You've put guardrails in place. You've established your AI pillars and you've created a structure to manage risk.

It's time to shift from experimentation to integration. You are no longer proving whether AI can help. You are building the environment where it becomes part of how your organization runs.

Most organizations stop at pilot projects. They launch tools that people don't use. Or they deploy systems that technically work but feel disconnected from the way teams get things done.

Operationalizing AI means embedding it into your workflows, systems, teams, and culture. It means treating AI as a capability that is continuously improved and expanded not as a single use case or short-term initiative.

Here is how you make that shift.

STEP 1

Define the AI system lifecycle

AI systems are not one-time builds. They require care, feedback, and improvement. If you do not create a structure for managing them, they will either become irrelevant or create problems you cannot see until they break something important.

Every AI solution should follow a defined lifecycle. That lifecycle should include:

- ▲ **Initiation:** Who requested this? What is the problem we are solving? What is the data source? Why now?
- ▲ **Development:** What logic or model are we using? How will we validate accuracy, fairness, or risk? What are the known limitations?
- ▲ **Deployment:** Who owns the outcome? Where will the user experience live? Who is responsible for support?
- ▲ **Monitoring:** What metrics will tell us this is working? Who is reviewing them? What happens if performance drops?
- ▲ **Iteration:** How are we collecting feedback? What is the schedule for reviewing and improving the system? Are we getting smarter over time?
- ▲ **Sunset:** When does this solution become obsolete? What is the signal to deprecate it or rebuild?

This lifecycle becomes your shared operating model. It brings consistency to how AI is developed and maintained across the organization. And it ensures that every solution is treated like a product, not an experiment.

STEP 2

Create AI-ready workflows

Before you embed AI into a process, you need to understand how that process actually works. Where are the inputs? Where are the handoffs? What decisions are being made?

What is manual that should be automated? What is complex and still requires human oversight?

Build the AI into the workflow as if it was always supposed to be there. They should experience better outcomes without needing to change how they work.

Here is your test: if the AI system went offline tomorrow, would people notice? Would they miss it? Would they ask for it back?

If the answer is yes, you've operationalized it. If the answer is no, you've built a prototype that didn't land.



STEP 3

Train for AI fluency

You do not need every employee to understand how AI models work. You do need them to understand how to work with AI.

AI fluency includes:

- ▲ Knowing when to ask AI for help
- ▲ Understanding what the system is doing and what data it relies on
- ▲ Recognizing when the output feels wrong or off
- ▲ Knowing how to correct or escalate when something doesn't make sense
- ▲ Being confident enough to use it and curious enough to learn from it

If your team does not trust or understand the AI, they will not use it. Or worse, they will use it blindly. Neither is acceptable.

You need to train for thoughtful adoption. Provide examples. Set expectations. Show the limits and strengths. This training should not be optional or one-time. It should be embedded in onboarding, reinforced in practice, and refreshed regularly.

STEP 4

Track and evolve usage over time

Once AI becomes part of your daily operations, it needs to be measured and managed like any other key capability.

Track where it is being used. Measure how often it is relied on. Watch for where it is ignored or bypassed. If people are consistently skipping the AI step, something is wrong with the experience. If they are overriding the results, the output might not be useful. If they are misusing it, the training is incomplete.

Use this data to adjust. Improve your prompts. Refine your models. Add new context. Remove friction.

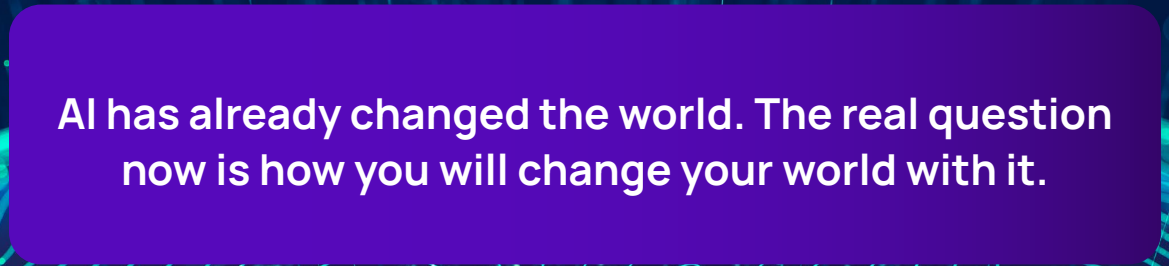
AI systems reflect the environments they live in. If your environment changes, the system must adapt.

As your business evolves, your AI should evolve with it. It should grow from solving tactical problems to creating strategic advantage.

You should stop talking about “the AI project” and start talking about how you work.

One team using prompts becomes five teams using embedded agents. Quick wins evolve into integrated systems. Experimentation becomes standard practice. AI becomes expected.

This is how you scale innovation. This is how you build capability. This is how AI becomes infrastructure.



AI has already changed the world. The real question now is how you will change your world with it.

The path ahead will not be linear. AI will continue to evolve. New models will emerge. Regulations will shift. Expectations will rise. But the principles you have now (friction-first design, measurable outcomes, ethical clarity, and adaptive systems) will hold.

Innovation is not a single breakthrough. It is a pattern of action. A habit of identifying friction, testing ideas quickly, learning from the results, and scaling what works. AI does not replace that process. It accelerates it.

But only if you lead with intention.

You now have the tools to do that. You know how to map opportunity, define the right kind of AI experience, and use the Crawl-Walk-Run approach to deliver results that grow over time. You understand the importance of risk management, governance, and clear ownership. You know how to train your teams not just to use AI, but to work with it responsibly. And you know how to scale without losing control.

That is what separates organizations who try AI from those who transform through it.



At N-able, our mission is to protect businesses against evolving cyberthreats with a unified cyber resiliency platform to manage, secure, and recover. Our scalable technology infrastructure includes AI-powered capabilities, market-leading third-party integrations, and the flexibility to employ technologies of choice—to transform workflows and deliver critical security outcomes. Our partner-first approach combines our products with experts, training, and peer-led events that empower our customers to be secure, resilient, and successful. n-able.com

This document is provided for informational purposes only and should not be relied upon as legal advice. N-able makes no warranty, express or implied, or assumes any legal liability or responsibility for the information contained herein, including for the accuracy, completeness, or usefulness of any information contained herein.

The N-able trademarks, service marks, and logos are the exclusive property of N-able Solutions ULC and N-able Technologies Ltd. All other trademarks are the property of their respective owners.

© 2025 N-able Solutions ULC and N-able Technologies Ltd. All rights reserved.