



HOW TO BUILD **AGENTIC WORKFLOWS** FOR BUSINESS

DESIGNED FOR BUSINESS PROFESSIONALS.
INCLUDES WORKBOOK WITH PRACTICAL EXERCISES.

RECORDLY

CONTENTS

03

INTRODUCTION

04

FROM ANSWERS TO ACTION

- How agentic AI changes your work from one-off answers to continuous execution
- Chat-based AI vs. agentic workflows in your daily tasks
- Exercise

08

COWORK INTERFACE

- How to work with AI as a coworker
- Using roles, inputs, and iterative review loops
- Exercise

12

SYSTEM PROMPTS

- How system prompts define the agent's role, behavior, and decision-making across tasks
- How to structure a strong system prompt using role, style, rules, and output format
- Exercise

16

SKILLS

- Turn instructions into reusable standards
- How skills help teams scale expertise and produce more consistent outputs
- Exercise

20

COLLABORATE EFFECTIVE WITH IT

- Translate customer feedback into development tasks
- Go smoothly from the client's initial request to the technical solution
- Exercise

23

HOW TO START WITH AGENTIC WORKFLOWS TODAY

- A five-step model to help you succeed

24

TOP 5 TASKS YOU CAN HAND OVER TO AN AGENT TODAY

26

HOW TO KEEP YOUR OWN SKILLS FRESH IN THE FAST-MOVING AI WORLD

- Build a library of your team's best practices
- Learning by doing: test, refine, and turn small experiments into repeatable practices
- Exercise

Many people use AI as a chat interface. They open a window, ask a question, get an answer, and move on. It's a manual way of working, often involving copy-pasting text or code between the chat and other systems. That already saves time, but it isn't yet a new way of working. The promise of AI agents is that they can act with a level of autonomy and handle tasks directly, instead of humans manually moving information around.



Tuomas Leidén, ML Engineer,
author of this guide

This guide exists to help you start treating AI like a virtual coworker that can follow instructions, work through your material, use tools, and help move your work forward across multiple steps.

We created the guide for business professionals who want to take advantage of the agentic shift in their work without having to become technical specialists.

In our guide, you will find tips to:

- Understand the difference between chat-based AI and agentic workflows
- Identify which parts of your work you can hand over to an AI agent
- Write effective instructions using roles, structure, and constraints (system prompts)
- Use skills in a practical way
- Set guardrails that reduce risk and improve consistency
- Collaborate more effectively with technical teams
- Get started with your first agentic workflow as a business professional.

We've shaped the guide around Anthropic's Claude, but similar methods would be applicable to tools like Google Gemini and OpenAI's ChatGPT.

This guide includes a workbook. Print it out to do the writing exercises by hand. You can choose to print out only the exercises (pages 7, 11, 15, 19, 22 and 28) to save paper and ink.



1. Moving from answers to action

Why agentic AI changes how work gets done

In this chapter, you'll learn:

- **How agentic AI changes your work from one-off answers to continuous execution**
- **Chat-based AI vs. agentic workflows in your daily tasks**
- **Assignment: Identify areas in your own work that an agent can take over.**

Moving from interaction to execution

Chat-based AI supports thinking, helps structure ideas, and makes it easier to process information. However, it remains an interaction model where the human is still responsible for driving the work forward step by step.

Agentic AI changes this dynamic by allowing AI to participate in the execution of work, operate within a defined role, and follow instructions across multiple steps. It can work with materials provided to it on a whole other level than just uploading a document to the chat interface. It executes on commands, performs tasks and uses tools on a computer, reading files, interpreting structured inputs, and applying consistent logic to produce outputs that align with a specific goal.

Continuous workflows

This difference becomes visible in how work progresses. In a chat-based setup, each session is isolated. The user needs to reframe the context, restate expectations, and guide the output each time for the output to be consistent. In an agentic setup, the AI can maintain a working logic across tasks and (crucially) take actions: calling tools, executing steps, and adjusting based on results. It understands how to approach a type of problem and can repeat that approach reliably. This enables the type of continuity that is difficult to achieve in standard chat usage.



Handling complex and structured work

In practice, this means that agentic AI can handle more complex and structured work. When a task involves multiple steps, dependencies, or defined processes, the AI can follow a sequence and move the work forward without requiring constant re-instruction. It can analyze material, structure findings, and produce outputs in a predefined format. It can also decide how to approach a task based on the instructions it has been given, which introduces a level of autonomy within controlled boundaries.

Working inside real environments

Agentic workflows also make it possible to connect AI to the actual working environment, such as folders, documents, or systems. This allows it to work with the same materials that a human would use, which improves relevance and reduces the need for constant context switching. The AI becomes part of the workflow rather than an external tool that requires translation of the problem into prompts.

Designing the outcomes

Instead of focusing only on how to phrase a prompt, the focus moves to setting a goal that the agent needs to achieve or a problem it needs to solve. This can be improved by defining how a task should be carried out, what inputs are needed, what a good output looks like, and where human judgment is required. This creates a more structured collaboration between human and AI, where the human sets direction and the AI supports execution.

Connecting to tools and systems

Model Context Protocol (MCP) is an open standard created by Anthropic. MCP is a secure, standardized way for AI applications to connect to external tools, data sources, and services. When working in agentic workflows, the agent is the one executing the task, calling tools, and making the MCP connections.



TABLE 1. CHAT-BASED AI VS. AGENTIC AI USAGE.

TASK	CHAT-BASED	AGENTIC
Brainstorming	x	x
Rewriting	x	x
Summarizing	x	x
Ideating	x	x
Quick questions & answers	x	x
One-off analysis	x	x
Continuously creating outputs in a required format	x	x
Working around multi-step tasks		x
Using instructions repeatedly in the same way		x
Reading from defined files, folders, or systems		x
Choosing the right process based on the task		x
Using tools, templates, and context consistently		x



Apply this to your work

Exercise 1: Identify agent opportunities

- What are three tasks you repeat weekly?
- Which of these involve:
 - documents
 - multiple steps
 - structured outputs
- Circle the best candidate for an agent.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

2. Cowork interface

Claude Cowork as a tool

In this chapter, you'll learn:

- **How to work with AI as a coworker by delegating tasks instead of asking questions**
- **How to structure effective collaboration using roles, inputs, and iterative review loops**
- **How to use Cowork for real work tasks and choose the right AI tool for different needs.**

Every organization runs on shared work: documents that need to exist, tasks that need tracking, messages that need sending, and files that need to end up in the right place. Cowork is where Claude shows up for that work as a colleague you hand things off to.

Think of the Cowork interface as your shared desk with Claude. You bring a task, such as turning meeting notes into a client summary, organizing last quarter's reports by project, or drafting a follow-up to the proposal you sent on Friday. Claude picks it up, works through it step by step using the tools connected to your workspace, and delivers a result you can immediately use or send.

The key shift Cowork asks of you is simple: stop describing what you want and start delegating it. Instead of asking Claude to explain how to format a budget spreadsheet, you hand Claude the spreadsheet and say what it needs to become. Cowork is built for pure outcomes.



Give Claude a role before giving it a task

The most effective way to work with Claude in Cowork is to establish context before diving into the task. It can be very concise, such as: “You are helping me prepare for a client review.” That one sentence calibrates tone, vocabulary, and judgment for everything that follows.

Let Claude review your inputs

One of the most underused patterns in Cowork is handing Claude a folder, a document, or a set of notes and asking it to orient itself before acting. “Read these three files and tell me what is missing before we start” – this review step prevents wasted effort downstream and surfaces gaps you may not have noticed yourself.

Work in loops

The most productive Cowork rhythm is this loop: assign, review, refine, approve, hand off. You assign the task, Claude produces a first version, you review and give feedback, Claude refines, and once it meets the bar, you approve and hand it off. Expecting a strong draft that improves through iteration leads to consistently good work.

Other ways to work with Claude

Cowork is one of several ways to access Claude, and choosing the right interface matters. For quick questions, research, or open-ended conversation, the Claude web, desktop, and mobile apps offer a familiar chat experience with no setup required. Developers and teams building custom workflows can access Claude directly through the Claude API, integrating it into their own products and pipelines.

For those working in a terminal or codebase, Claude Code lets developers delegate coding tasks to Claude directly from the command line. Cowork sits alongside these options as the interface designed specifically for non-developers who need to get real work done with files, tasks, and connected tools.



Other AI providers

Claude isn't the only AI tool in this space, and different tools suit different needs. OpenAI's ChatGPT is the most widely recognized, available via web, mobile, and the API, with strong general-purpose capabilities and a large ecosystem of integrations. Google Gemini is deeply embedded in Google Workspace, making it a natural fit for teams already working in Docs, Sheets, and Gmail. Microsoft Copilot is integrated into the Microsoft 365 suite, bringing AI assistance directly into Word, Excel, Outlook, and Teams.

For coding specifically, GitHub Copilot has become a popular choice among developers. While these tools overlap significantly in capability, they differ in where they live, what data they can access, and how they handle privacy and enterprise controls. Choosing between them often comes down to your existing toolstack and where your team already spends most of its time.



Apply this to your work

Exercise 2: Map your first Cowork session

- What task will you bring to Claude this week?

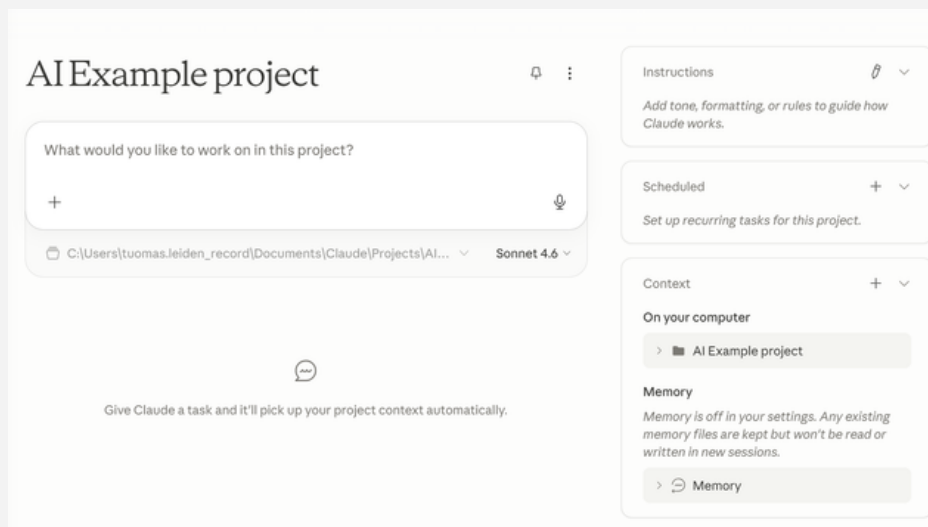
- In one sentence, what role should Claude play?

- What does a good output look like?

- When will you review the result and what would make you approve it?

3. The system prompt is the soul of the agent

IMAGE 1. CLAUDE INTERFACE WITH WINDOW FOR SYSTEM PROMPT (INSTRUCTIONS)



In this chapter, you'll learn:

- How system prompts define the agent's role, behavior, and decision-making across tasks
- How to structure a strong system prompt using role, style, rules, and output format.

A system prompt defines how an AI agent should think, behave, and work before any task-specific instructions are given. It acts as the underlying logic that shapes every response and action the agent takes. While regular user prompts tell the AI what to do in a specific moment, the system prompt defines how it should approach all work in general. That's why we describe it as the "soul of the agent." Once it is in place, everything that follows is filtered through it.



In practice, this means that even when you give new tasks later, the agent will prioritize the rules and structure defined in the system prompt. If the system prompt says to be concise, structured, and cautious with assumptions, the agent will apply those principles regardless of how loosely the next user prompt is written. A useful way to think about a system prompt is as a combination of role description and operating manual.

It answers questions such as:

- Who is the AI in this context?
- What is its goal?
- What is it allowed to do?
- What is it not allowed to do?
- How should it approach the work step by step?
- What should the final output look like?

TABLE 2. EXAMPLE OF A SYSTEM PROMPT.

Role	You are a senior marketing analyst responsible for producing structured, insight-driven quarterly marketing reports for a B2B company.
Style	• Professional, neutral, and analytical • Translate marketing data into actionable insights for leadership • Direct, brief and clear • Write for leadership audience • Act as a data-driven marketing strategist.
Constraints	• Do not fabricate metrics, benchmarks, sources, or outcomes • Do not use generic statements such as “performance was strong” without evidence • Do not over-explain obvious numbers • Do not bury bad news • Do not use cliché language such as “game-changing,” “world-class,” “revolutionary,” or “cutting-edge.” • Use bullets and tables where they improve readability • Separate facts, interpretations, and recommendations clearly • State uncertainty plainly when the data does not support a firm conclusion • Prefer fewer, sharper insights over broad summaries • Cite credible sources or references to support your answers with links • If a mistake is made in a previous response, recognize and correct it.



With a well-defined system prompt, all expectations are already built into the agent. This allows the user to focus on the task itself rather than on continuously managing the AI.

System prompts also make collaboration easier. When multiple people use the same agent or workflow, a shared system prompt ensures that the outputs follow the same logic and quality standards. Learning to define system prompts is one of the most impactful ways to improve how AI is used.

In agentic workflows, the quality of the system prompt determines the quality of the outcome. It is the layer that connects intention with execution. When it is clear, structured, and aligned with the task, the agent becomes easier to trust and more useful in everyday work.



Apply this to your work

Exercise 3: Define your first agent

What are your agent's:

a) Role:

b) Goal:

c) Constraints (what it must and must not do):

d) Steps to follow:

e) Output format:

4. Skills

In this chapter, you'll learn:

- **How skills turn instructions into reusable standards for recurring work**
- **How to define a skill by clarifying when it should be used, what it should do, what the output should look like, and what it should avoid**
- **How skills help teams scale expertise and produce more consistent outputs across people, projects, and time.**

Every experienced professional has their own way of approaching recurring work. A consultant who writes proposals every week has a rhythm; the questions they ask first, the structure they default to, and the tone they calibrate for the client's industry. A skill in Claude works the same way, encoding that rhythm so the agent can follow it without being re-explained every time.

A skill is a reusable instruction block for a recurring type of work. Instead of briefing Claude from scratch each time you need a proposal drafted, a meeting summarized, or an RFP analyzed, you save that briefing as a skill. Claude picks it up automatically when the situation calls for it.

Without a skill, you spend time re-explaining context, format, and quality bar before every task. The agent produces something reasonable, but it reflects a fresh interpretation of your need rather than your established standard. With a skill, the agent already knows how to approach the work: what to look for, how to structure the output, and what to leave out. Your standards become the agent's default behavior.



This matters especially for teams. When a skill exists for proposal drafting or customer research, any team member gets the same quality of output regardless of how much they know about prompting. Organizational expertise stops living only in the heads of the people who've been doing the work longest.

How to use skills

A well-written skill answers four questions:

1. When should this be used?
2. What should the agent do?
3. What should the output look like?
4. And what should it avoid?

These four guardrails are enough to make a skill reliable, but you need to be precise about the parts that matter most.

A simple example

Take a “meeting notes to executive summary” skill. It might read “use this when the input is workshop notes, meeting transcripts, or rough written thoughts. Produce a concise summary for senior leaders that includes context, decisions made, open questions, and next actions. Keep it under one page. Do not include personal opinions or attribution unless explicitly asked”.

That is all you need. The skill is now a repeatable standard, consistent across team members, projects, and time.

Skills worth building first

The best candidates for skills are tasks you do repeatedly, tasks where quality varies too much across people or attempts, and tasks where you've already figured out what good looks like. In most business teams, that means proposal drafting, meeting summarization, RFP analysis, customer research synthesis, account planning, workshop preparation, and stakeholder briefings. Start with whichever one would save the most re-explaining.

IMAGE 2. EXAMPLE OF A SKILL.

Description ⓘ

Convert meeting notes into a polished executive summary PDF. Use this skill whenever the user uploads a .txt or .docx file containing meeting notes and wants a summary, executive brief, or condensed overview. Also trigger when the user says things like "summarize this meeting", "make an exec summary", "turn these notes into a summary", or "create a summary from my meeting notes". Always use this skill when a meeting notes file is present and any kind of summary output is requested — even if the user just says "summarize this" or "what happened in the meeting".

Meeting Notes → Executive Summary PDF

Overview

Read a `.txt` or `.docx` file of meeting notes, extract the key information, and produce a clean, professionally formatted PDF executive summary.

Output Sections (always include all four)

1. **Key Decisions Made** — Concrete decisions that were reached
2. **Action Items & Owners** — Tasks assigned, with owner names if mentioned
3. **Next Steps / Follow-Ups** — What happens after this meeting
4. **Risks or Blockers** — Anything flagged as a concern, risk, or impediment

Also include a header with:

- Meeting title (infer from content if not explicit)
- Date (extract if present, otherwise leave blank)
- Attendees (list if mentioned)

Step-by-Step Instructions

Step 1 — Read the input file

For `.docx` files:

You may have noticed many similarities between system prompts and skills, and a system prompt can also tell the agent how to use skills. For example, it can instruct the agent to check whether any reusable skill applies before starting the task. In Claude, skills are then used when they're relevant to the user's request. This creates a useful hierarchy: the system prompt defines the agent's general way of working, while skills provide task-specific methods the agent can apply when the situation calls for them.



Apply this to your work

Exercise 4: Design your first skill

Pick one task or thought process you repeat at least once a week and answer the four skill questions.

1. When should this skill be used?

2. What should the agent do?

3. What should the output look like?

4. What should the agent avoid?

5. How to collaborate more effectively with IT

With the help of agentic AI

In this chapter, you'll learn to:

- **Translate customer feedback into a development task**
- **Move from reactive hand-offs to proactive collaboration**
- **Get a smoother path from the client's initial request to the actual technical solution.**

Across organizations there is a persistent gap between people gathering customer feedback and the technical teams charged with building solutions. Business professionals may sit in the same meetings as developers, but they use a completely different vocabulary and set of incentives. When vocabularies don't connect, miscommunication leads to rework, missed opportunities and delays. The bridge between these functions is a clear translation of needs into shared understanding and value that both are able to measure in the same way.

In previous chapters, we've established that an agent can hold a role, follow a system prompt, use skills, and operate across many steps. This design approach is especially valuable when business professionals work with data and IT teams. Instead of handing over raw feedback or loosely phrased requests, the AI agent can help translate the voice of the customer into structured requirements in a language that aligns well with developers.



A well-crafted system prompt instructs the agent to extract the “why” behind each piece of feedback, identify the underlying problem, and frame it as a measurable outcome. By emphasizing what success looks like and which files or templates to use, the agent can automatically produce user stories, tickets or briefs that include the business context and acceptance criteria, reducing ambiguity and aligning all sides.

This translation process requires more than just rephrasing, however. It involves understanding the customer’s problem, clarifying the desired outcome and the metrics that matter, and then explaining it in terms of the system’s capabilities and constraints. Business professionals should attach every request to a measurable business goal. Rather than asking for “better security,” they might ask for “measures to reduce data breach risk by a defined percentage”. The agent can then incorporate this goal into a technical specification that describes the scope, risks, assumptions and acceptance tests. Including a one-sentence explanation of “why this matters” for each ticket helps developers appreciate the business impact and invites them to suggest better alternatives if the proposed solution clashes with technical realities.

An agent can maintain a record of the agreed vocabulary, templates and metrics, ensuring that every new ticket or requirement follows the same structure. It can also act as a translator in meetings, summarizing technical discussions in business terms and vice versa. This continuity means that the data team receives well-formed, prioritized requests with clear value statements, and the business professionals get progress updates in a language that’s familiar to them.

Based on a thorough brief from the business, developers can create prototypes already before the first client kick-off meeting. Moreover, business stakeholders can be invited to sprint reviews, roadmap discussions and demos so they have a chance see trade-offs and ask informed questions.



Apply this to your work

Exercise 5: Translate customer or internal feedback into a development task (or ticket).

First, define:

Customer or internal feedback:

What is the underlying business problem?

Who is this development task delegated to?

Give the above answers to the AI agent and let it create a development task based on it.

Finally, are you satisfied with the result and agent's interpretation of the problem? Ask the person or team you delegated the task to for their interpretation as well.

6. How to get started with agentic workflows today on Claude

Getting started does not require a big rollout or IT approval. It only requires one task, one clear output, and the willingness to run a small experiment. The five steps below will take most business teams from zero to a working agentic workflow within a week.

You can make it a simple five-step model:

1. Pick one repeated task. Choose something that happens every week.
2. Define the ideal output. What does “good” look like?
3. Give the agent a role. What kind of coworker is it acting as?
4. Add rules and review points. What must it not do? When do you review?
5. Run small, then repeat. Don't automate a huge process first. Start with one narrow workflow.



6.1 Top 5 tasks to hand over to an AI agent today

1. Turning messy and scattered material into decision-ready summaries

Meeting notes, workshop outputs, customer calls, and internal documents often contain useful information buried in various locations. An agent can process that material and return a clean, structured summary ready for a decision maker to read.

- Meeting and workshop notes
- Customer call transcripts
- Internal documentation reviews.

2. First-draft proposal and response preparation

RFP responses, follow-up emails, partnership summaries, and account plans follow repeatable structures. An agent can produce a strong first draft from your briefing notes, saving hours on the version that actually goes to the client.

- RFP responses
- Customer follow-up drafts
- Account plans and partnership summaries.



3. Research and synthesis across provided material

When you need to understand a competitor, scan a market, or review public procurement documents, an agent can read across a large body of material and return a structured synthesis. You review the conclusions, not every source.

- Competitor summaries
- Market and sector scans
- Public procurement and tender reviews.

4. Structured follow-up and action tracking

After meetings and workshops, someone has to produce the action list, draft the tickets, and chase the next steps. An agent can take the raw notes and return a clean, assigned action list every time.

- Meeting action lists
- Stakeholder next steps
- Ticket and issue drafting.

5. Content repurposing across formats

Material created for one format rarely gets reused in others. A workshop becomes a memo, an interview becomes an article outline, a webinar becomes a campaign. An agent can do the format conversion so your original thinking reaches more audiences without more writing time.

- Workshop notes to memo or report
- Interview or conversation to article outline
- Webinar or talk to campaign angles
- Notes to presentation structure.



7. How to keep your own skills fresh in the fast-moving AI world

Now with agentic capabilities

In this chapter, we'll talk about:

- **Why build a library of your team's best system prompts and skills**
- **Learning by doing: test, refine, and turn small experiments into repeatable practices.**

The pace of change in AI isn't slowing down. New models, tools, and capabilities appear constantly, and it is easy to feel that keeping up requires continuous effort and some technical understanding, too. In practice, the most valuable skill is developing a way of working that can evolve as the tools do.

Throughout this guide, the focus has been on moving from individual prompts to structured workflows. That same principle applies to learning. Instead of chasing every new feature, it is more effective to focus on how your own work is changing. Which tasks can now be handled by an agent? Which workflows are becoming reliable enough to reuse? Where do you still need human judgment, and where can AI take over more responsibility?

One practical way to stay current is to build a personal or team playbook. When something works well, it should not remain a one-off success. Save the system prompts that produced good results, document the roles that made the agent useful, and keep examples of outputs that met your quality expectations. Over time, you can create a library of effective system prompts that can be reused and improved, or used as skills. This also supports a shared way of working where agents are part of how the organization operates.



Regular reflection is equally important. The question is not what is new, but what has become possible. Something that was unreliable three months ago may now work consistently. A task that required manual effort may now be handled by an agent with clear instructions and review points. Taking time to revisit your workflows helps identify where to standardize and where to invest further.

Learning also happens fastest through doing. Courses, certifications, and online material can provide useful structure, but the real shift comes from applying these ideas in your own work. Testing an agent on real materials, refining outputs, and adjusting system prompts creates a much deeper understanding than passive learning. This is why hands-on formats such as workshops and hackathons are so effective. They compress learning into a short time and make the impact immediately visible.

You don't need to master everything to get started. What matters is building a habit of testing, documenting, and refining how you work with AI. The teams that move forward aren't the ones that know the most, but the ones that turn small experiments into scalable practices.



Apply this to your work

Exercise 6: Your first week plan

Day 1: What workflow will you test?

Day 2: What will your system prompt look like?

Day 3: What material will you use?

Day 4: What needs improvement?

Day 5: What will you standardize?

THANKS FOR READING

Congratulations! If you filled even one of our exercises, you've already started designing agentic work.

If you want to accelerate the agentic shift in your organization, the most effective way is to do it together. **Recordly works with teams to move from individual AI use to structured agentic workflows in real business contexts. Through hands-on, custom hackathons, we help you identify the right use cases, design system prompts, build reusable skills, and test workflows on your own materials.** A focused hackathon is often the most effective starting point, as it combines learning, experimentation, and real outcomes in a short time.

CONTACT US

 recordlydata.com

 sales@recordlydata.com

 Kaivokatu 10B, 00100 Helsinki

RECORDLY INSIGHTS



RECORDLY INSIGHTS

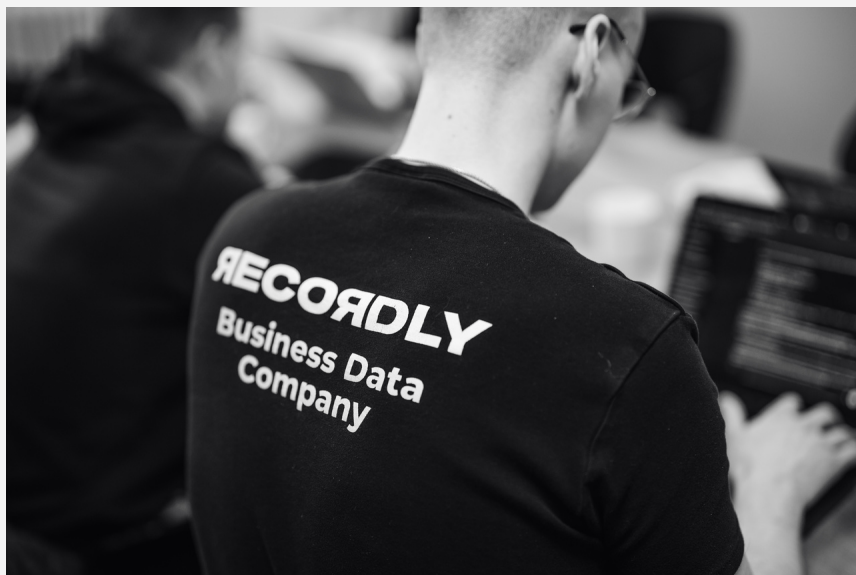
ABOUT RECORDLY

Recordly is a Finnish data & AI consultancy and strategic partner in AI transformation with the mission to define how humans and data cooperate. We build scalable data systems, advise on data & AI strategy, and deliver production-grade GenAI and agentic solutions that solve your business challenges and drive measurable ROI and growth. Our NPS score of 85 proves our clients trust us to deliver measurable business impact every day.

CREDITS

All core learning material in this guide has been developed by Tuomas Leiden, ML Engineer at Recordly, based on hands-on work with agentic AI and experimentation. The concept, layout, and editorial work of this guide have been designed by Michaela Holmström, Marketing Lead at Recordly. This guide is a result of close collaboration between technical and business perspectives, reflecting the same way of working it promotes. All photography in the guide is by Jami Ivanoff.

© Recordly Oy, 2026. All rights reserved.





recordlydata.com

RECORDLY