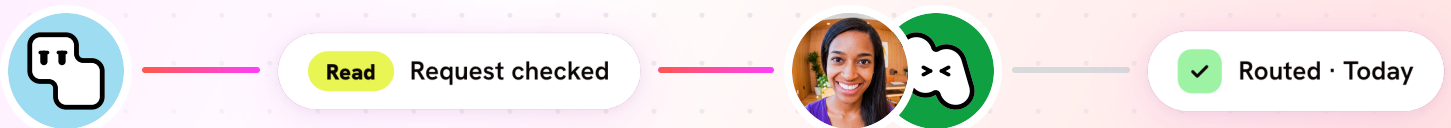


The practical guide to agentic workflows

How human-agent teams actually get work done.



What's inside

0	Introduction	Why we wrote this, and how to read it.	3
1	The 5% problem	75% of workers use AI daily. 5% of companies see the gains.	5
2	The four blockers	Why your AI tools don't help your team.	8
3	Agentified workflows	The actual unlock: humans + agents on a shared plan, in the open.	11
4	The taxonomy	Rules, AI Studio, AI Teammates, external assistants: which one fits.	14
5	The four jobs	Read, check, route, nudge, and the judgment layer above them.	18
6	Five workflows	Real builds walked through, from program management to content.	22
7	Before you build	What to get right before you add a single agent.	26
8	The checklist	Print it. Score your team honestly.	29
9	Where to go next	The summit, office hours, and a readiness consultation.	30

Before you start

Pick a single workflow your team runs every week, the one that quietly annoys everyone, and keep it in mind on every page. By the end, you will know exactly what to do with it.

Why we wrote this

There is no shortage of content about AI at work. You have seen it: the manifestos, the keynote clips, the threads promising that everything changes next quarter. If that content worked, you would not be here.

We wrote this guide because we sit in a different seat. Cirface is an Asana Solutions Partner, and building workflows is our actual job: intake systems for banks, campaign engines for marketing teams, service desks for IT, week in, week out, for years. We run our own business on the same workflows we sell. So when AI agents arrived, we did not get to theorize about them. Clients asked us to make them work on real processes, with real governance requirements, real messy inputs, and real people who would stop trusting the whole thing after one bad week. Some of what we tried worked immediately. A surprising amount did not. **This guide is the difference between those two piles.**

That seat shapes what this is and is not. It is not a manifesto, and it is not a product brochure. There are chapters in here about when a plain rule beats an agent, what to refuse to automate, and why “not ready” is sometimes the correct and respectable answer. We would rather you build one boring workflow that survives contact with your team than admire ten impressive ones that do not.

A few ways to read it. Front to back works; the chapters build an argument. If you are the skim-first type, go straight to the readiness checklist in Chapter 8, score your team honestly, and let the gaps send you to the chapters you need. And if you want the most from the next hour, do one thing before you start: pick a single workflow your team runs every week, the one that quietly annoys everyone, and keep it in mind on every page. By the end, you will know exactly what to do with it.

The people and agents described in these pages are working together right now, today, inside ordinary companies. Nothing here requires a lab, a moonshot budget, or a leap of faith. Just a workflow, a plan, and the willingness to start smaller than your ambition.

Let's get to work.



Marquis Murray

CEO, Cirface

cirface

The 5% problem

The personal AI story is settled. Copy written in seconds. A presentation drafted from bullet points. Code shipped overnight, meeting notes turned into tasks before the meeting ends. Nearly everyone on your team is measurably faster than they were eighteen months ago, and if you are reading this, you probably are too.

So here is the uncomfortable number.

~75%

of workers now use AI every day for their job

5%

of companies report measurable productivity gains from it

Sit with that gap for a second, because everything in this guide comes from taking it seriously. Seventy-five percent of individuals, five percent of organizations. If AI made people faster and companies barely noticed, then the thing standing between your team and real gains is not a better model.

It is not an AI problem

The models are extraordinary. Your people are faster. But your teams are not, and your company is not, and that gap does not close on its own.

The reason is unglamorous: individual speed does not compound. A marketer who drafts a campaign brief in ten minutes instead of two hours has saved themselves 110 minutes. The campaign still waits three days for the handoff, the review, the approval, the missing context from the call nobody wrote up.

Work does not move at the speed of its fastest task.
It moves at the speed of its **slowest handoff**.

Most teams are using AI like a faster intern: hand it a task, get a draft back, repeat. Useful, genuinely. But that is not where this is going, because the intern model keeps every gain locked inside one person's browser tab. The organization never feels it.

What a human-agent team actually is

The shift worth paying attention to is from AI as a personal tool to AI as a participant in the workflow. A human-agent team is exactly what it sounds like: people and AI agents working from the same plan, with the same context, under the same governance. The humans bring judgment, taste, and accountability. The agents bring tireless execution of the defined, repeatable work that sits between human decisions.

Agentic work management is the practice of running your workflows that way on purpose. Not an agent bolted onto the side of a process, but a process designed with agents inside it: **reading** what comes in, **checking** it against your standards, **routing** it to the right people, and **nudging** when something slips. If those four verbs sound oddly specific, they are. Chapter 5 is entirely about them.

What this guide will do

This is a practical guide, and we mean that word. We run these workflows for our own business and build them for clients every week, and the chapters ahead follow the order that work actually happens in:

First, why your current AI tools are not helping your team, and the four blockers behind that (Chapter 2). Then the reframe that changes what you build (Chapter 3), and a plain-language taxonomy of the tools so you stop guessing which one fits (Chapter 4). The four jobs agents do inside real workflows (Chapter 5), five worked examples (Chapter 6), and, just as important, what to get right before you add a single agent (Chapter 7), with a readiness checklist you can print and argue about as a team (Chapter 8).

You do not need to be in the 5% today. You need a first workflow, a clear job for the agent inside it, and a team that can see what it does. That is buildable, and by the end of this guide you will know how.

Why your AI tools don't help **your team**

If your people are faster but your company is not, something specific is in the way. It is not the models. The models are extraordinary. It is that almost everything blocking you sits between individual AI use and team execution. Four blockers show up again and again, and if you recognize your organization in more than one of them, you are in good company. Almost nobody has crossed this gap yet.

01

It's hard to get started

Agents do not arrive pre-configured. They show up capable of nearly anything and attached to nothing, and teams stall on the same two questions: which workflow do we put this in, and what exactly is its job? So the agent gets a vague mandate, produces vague output, and quietly gets abandoned. The fix is not a better agent. It is a defined place in a defined workflow, which is where the rest of this guide is headed.

02

Most agents are not team players

Most AI today is singleplayer by design. One person, one prompt, one chat window. The agent might be brilliant in that conversation, but your team never sees it. Nobody else can build on what it produced, correct what it got wrong, or even know it happened.

We see the practical version of this constantly, and our Head of Sales hears it first:

“Team are not using AI at the enterprise level, they're using it at the individual level. Some people are super effective, some are not, and there's no knowledge sharing internally. People get frustrated: how come are you using AI that way? I could have saved so much time.”



Maggie Falardeau · Head of Sales, Cirface

When each person connects an assistant to their own tools, each person authenticates their own email, their own drive, their own meeting notes, and each gets access only to what they can see. Two colleagues ask the same question about the same account and get two different answers.

Individual productivity compounds into organizational execution only when the whole team works from, and contributes to, the same picture.

03

Agents don't know your history or your context

An agent that does not know where it is in your business is a very confident new hire on day one. It has not read the plan. It does not know who owns what, which decisions were already made, or why the launch slipped last quarter. So it answers from general knowledge, and general knowledge is exactly what your team does not need more of.

Context is not something you can paste into a prompt every morning. It has to live where the work lives, so that every task completed and every piece of feedback carries forward, and every workflow starts smarter than the last.

04

Agents are hard to govern

CIOs have started calling it agent palooza: agents multiplying across the organization with full data access, no owner, no audit trail, and no cost oversight. And it is usually not malicious. It is a well-meaning team member connecting company tools to a personal AI plan because it made their afternoon easier.

That is shadow IT, and it is happening in most organizations right now, ahead of any policy. Without guardrails on what data enters these tools, who can enter it, and in what form, you can get yourself into real trouble, especially when your customers never agreed to have their information there. Governance is not the boring part of AI adoption. It is the part that decides whether legal and IT ever let the useful parts ship.

The pattern underneath all four

Look at the four blockers together and they are one problem wearing four costumes: **the agent is disconnected from the workflow**. It has no defined place (blocker 1), no shared audience (blocker 2), no accumulated context (blocker 3), and no oversight (blocker 4). Solving them one at a time, a policy here, a prompt library there, treats symptoms.

The next chapter is about the actual unlock: not adding more agents, but embedding the ones you have inside the workflows where your business already runs.

Agentified workflows: the actual unlock

The blockers in the last chapter share one root: the agent is disconnected from the workflow. So the fix is not more agents, better prompts, or a bigger model. The unlock is **agentified workflows: business processes where humans and agents collaborate on a shared plan, with shared context, in the open.**

That sentence is easy to nod at and hard to picture. So instead of defining it further, let us show you the moment it clicks.

What “in the workflow” actually looks like

On our channel, Marquis records a video where he builds a triage assistant, and he uses another AI teammate to help him do it. Watch what happens, because every beat is the reframe in action.

He creates a task on his video production board and assigns it to the teammate, the same way he would brief a colleague. The teammate does not answer in a chat bubble. It posts a plan, as subtasks, on the shared board: create the reference documents, write the behaviour instructions, draft the script. Then it works through them in order, in the open, while Marquis does other things. The documents land in Google Drive, where the rest of the team could comment on them. The behaviour instructions land in an Asana subtask. And before anything ships, the teammate assigns Marquis a formal review task: here are the deliverables, approve or push back.

Notice what never happened: nobody sat in a one-to-one chat window feeding prompts and copy-pasting outputs. The work has a home, a visible plan, an audit trail, and a human approval gate, because it lives where the team's work lives. If Marquis dislikes one deliverable, he comments on that one subtask, and the feedback applies exactly there. That is the difference between an assistant and a teammate: **an assistant answers you; a teammate works the plan with you.**

Joining the team, literally

The build itself makes the point a second time. When Marquis creates the triage assistant, he does not install it. He shares a project with it:

"We're basically adding a team member. Like you would share a project or a portfolio with someone, we're sharing this with the AI teammate."



Marquis Murray · CEO, Cirface

Then he gives it what you would give a new hire: the intake guidelines, the prioritization rubric, the governance rules. The teammate reads them and its memory visibly updates. From then on, every piece of feedback in a comment commits to that memory:

"They work on context. If you're not taking the time to prepare and give it good information, you're not going to receive the best output you possibly could."



Marquis Murray · CEO, Cirface

This is what "coaching a colleague, not coding a rule" means in practice. The teammate is **scoped** like a person (it sees only the projects it was invited to), **briefed** like a person (documents, not configuration), and **corrected** like a person (feedback, committed to memory, in front of the team).

The payoff, in one request

Then a request comes in through the intake form: new copy machines, every floor, \$50,000, urgent. The teammate validates the form is complete, scores the request against the weighted rubric it was given, 59 out of 100, medium priority, flags that engineering, a listed dependency, is sitting at 91% capacity, and recommends conditional approval, routing it to a senior analyst for the human call, exactly as the governance guidelines require.

Read that list again against Chapter 5's two layers: a check, an audit, a synthesis across projects, a route, and a human decision at the end. Work that used to be a meeting, a spreadsheet, and two weeks of back-and-forth became minutes, and the human enters at precisely the moment judgment is needed, with the assessment already on the table.

Agentified is a path, not a switch

One more thing the build quietly demonstrates: the layers stack. Marquis's last step is adding a plain rule, when a form submission arrives, assign it to the triage teammate. A rule hands work to an agent. That is the whole architecture in miniature, and it is why "agentified" is not a binary you flip but **a ladder you climb**: rules automate the fixed logic, AI Studio adds interpretation inside the process, and Teammates add judgment on top, each layer standing on the one below. You met that ladder in the case studies, and you will use it deliberately in the chapters ahead: first the tools (Chapter 4), then the jobs each layer owns (Chapter 5).



Watch this next The entire build, from blank teammate to the scored copier request: "How I Built an AI Agent to Triage Project Requests in Asana."

youtu.be/7UU5pGIayfY

The taxonomy: what's actually different

If you have ever sat in a meeting where someone said “let’s just use AI for that” and nobody could say which AI, you are not alone. Inside Asana there are rules, AI Studio, and AI Teammates. Outside Asana there are chat assistants that can now connect to your work through MCP. They sound similar. They are not made equally, and using the wrong one is how teams end up with automation they fight instead of automation that helps.

Here is the shortest version we can give you, and it comes from years of building these for clients:

“A **native rule** is best when the answer is fixed. **AI Studio** is best when the process is fixed, but the input needs a bit of interpretation. **AI Teammates** are best when the work needs context, judgment, or there are multiple variables involved.”



Marquis Murray · CEO, Cirface

That one distinction resolves most of the confusion. Let’s take each in turn.

⚡ Rules: when the answer is fixed

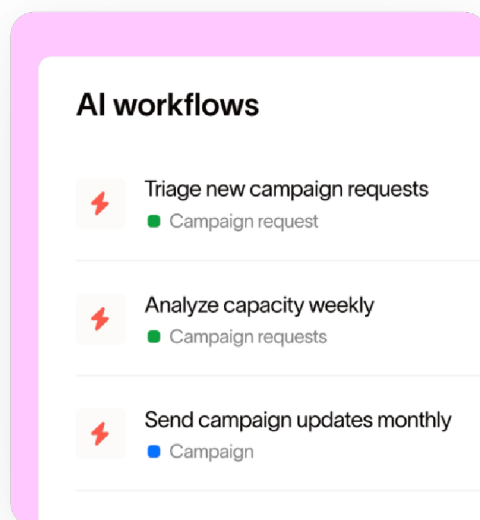
A rule is simple logic: when X happens, check that Y condition is met, produce Z result. When a task moves to the Declined section, complete it and rename it. When a request lands in Planning, assign it to triage and add a comment. You already know the logic. You just want it to happen without you.

Rules are best for moving or creating tasks, changing fields and statuses, and notifying the right people. And here is an unpopular opinion from our own channel: **you do not need AI for absolutely everything**. Asana will offer to build every rule with AI. If the logic is already worked out in your head, skip it. A plain rule is faster to build, easier to trust, and never surprises you.

✨ AI Studio: when the process is fixed but the input is messy

AI Studio still lives inside a defined workflow. What changes is that the workflow can now interpret what it receives. Not all submissions are created equally. Wording is off, information is missing, the request does not match your conventions. Instead of “when a task is created, move it to this section,” you are saying: when a task is created, read it, check it against the criteria I gave you, and take the next step based on what you found.

That read-and-check pattern is why AI Studio shines at intake: renaming requests to a standard format, catching duplicates, flagging missing information, routing work to the right team. It follows the guidelines you set, at volume, without a human standing over it. The process is yours. The interpretation is delegated.





AI Teammates: when the work needs judgment

A Teammate is not a smarter rule. It is closer to a co-worker. You can assign it tasks, mention it in comments, hand it documents, and give it feedback it learns from. It acts semi-autonomously: a status reporter that sweeps your portfolio every week, flags slipped due dates, and surfaces risk to the right people did not need you to prompt it that morning.

The deeper difference is where a Teammate lives. It sits directly on your work graph, so it has context on the work itself: who is doing what, by when, toward which goal. It is also multiplayer. Your whole team guides one agent, and everyone uses what it produces. It carries shared memory, so every workflow starts smarter than the last. And it is governed: named, scoped to the projects its builder can access, with an administrator deciding which apps and connections are allowed at all.

“AI Teammates have access to the work graph. They can access different projects, different portfolios. AI Studio is simpler to implement: you can have one project in Asana and set up an AI Studio rule. But if you only have one project, an AI Teammate won’t have the context of an already structured organization that has been using Asana for some time.”



Juli Arenzo · Senior Solutions Engineer & Community Manager, Cirface

What about external AI assistants?

Many teams also connect a general-purpose AI assistant to Asana and their other tools through MCP connectors, and there are genuinely good uses for that. Pulling context from your CRM, your meeting notes, and your inbox into one conversation is powerful for research and one-off investigation.

The trade-offs are structural. The conversation is one-to-one: you and the assistant. Each person authenticates their own tools and sees only what they have access to, so two colleagues asking the same question can get different pictures of the same account. Nothing the assistant learns is shared with the team. And when people connect company tools to personal AI plans, which is happening quietly in most organizations right now, you have a governance gap before anyone has written a policy.

None of that makes external assistants wrong. It makes them personal tools: excellent for one person's research, structurally unable to hold a team's shared picture. For work a team shares, you want agents that live where the work lives.



Watch this next The same job run both ways, Claude connected over MCP and then an AI Teammate, in "Claude-Asana MCP vs AI Teammates."

youtu.be/WbzW7L1tTtY

The one-line decision guide

The answer is fixed	→ build a rule
The process is fixed but the input needs interpretation	→ build it in AI Studio
The work needs context, judgment, or multiple variables	→ give it to an AI Teammate
It is personal research across your own tools	→ use your AI assistant, and know its limits

The **four jobs**, and where agents actually begin

Here is an honest sentence most AI content will not give you: the majority of what teams call “agentic” today is automation, and that is not a criticism. Automation is the foundation agents stand on. But the distinction matters, because the two layers are built differently, governed differently, and trusted differently, and knowing which one a task belongs to tells you what to build.

● **The workflow layer: read, check, route, nudge**

Across the workflows we build and run with clients, the work that keeps a process moving keeps reducing to four jobs. This is the territory of rules and AI Studio: a defined process, executed reliably, with AI interpretation where the input is messy.



Read turns unstructured input into structured understanding. A rambling request arrives; the workflow extracts what is being asked, by whom, for when, in what category, and renames it to match your conventions. Unglamorous, and the foundation everything else stands on.



Check compares what arrived against what is required. Complete? Within guidelines? Already requested by another team? The duplicate check is usually the first automation that makes a team say *oh, we needed this*. The instruction pattern Marquis teaches for read and check together: when a task is created, read it, check it against the criteria you were given, and take the next step based on what you found. That sentence is most of intake automation.



Route sends checked work to the right place with the right fields set.

Worth knowing: at scale, routing is often not even AI. Indeed's global creative team routes 1,500+ annual requests to 80 portfolios using 18+ plain rules keyed on region and asset type.¹ If the logic is fixed, a rule does it perfectly, forever, for free.



Nudge watches work in motion and speaks up: the slipped due date, the stalled dependency, the task sitting unassigned past its window, surfaced to the person who can act, before Friday's surprise.

These four jobs are where every team should start, and where a shocking amount of value already lives.

\$600K

Morningstar's teams saved over \$600,000 a year largely on this layer: AI Studio automating intake, request evaluation, and a content pipeline. No judgment calls delegated, just a defined process running without friction.²

The line between the layers

The four jobs execute a defined process. What comes next is different in kind: work where the process cannot fully specify the answer, where the agent has to exercise judgment inside your standards rather than follow steps through them.

Indeed's creative operations lead put the difference perfectly when the team moved from Studio automations to their first AI Teammate; setting one up is:

“Less like coding a rule and more like coaching a colleague.”

Creative operations lead, Indeed¹

You do not configure a judgment-layer agent with triggers and branches. You feed it your guidelines, your institutional memory, your examples of good and bad, and then you review its work the way you would a new hire’s.

● The judgment layer: audit, research, synthesize, advise

Look at what AI Teammates are actually doing in Asana’s published customer stories and the judgment work also reduces to a short list.



Audit is evaluating work against standards. Indeed runs a Content Strategist Teammate as an always-on brand auditor, matching localized assets to global brand guidelines across 28 languages, work that used to be manual review cycles by senior producers, now continuous and instant.¹



Research is deep, multi-source investigation. KW Automotive’s IT and support Teammates resolve multi-lingual technical tickets by researching across the company’s institutional knowledge, producing answers their COO describes as higher quality than busy human experts had time to give.³



Synthesize is turning scattered data into insight. KW’s Analyst Teammate builds campaign reports from raw data, saving hours per report, and, notably, proactively surfaces insights nobody asked for by correlating across projects.³ Proactivity is the tell: the workflow layer waits for triggers; the judgment layer notices things.



Advise is drafting the thinking, not just the document: creative briefs generated from market context, proposed next steps, a 70%-complete project scope that turns a senior person from author into editor.

What makes this layer work is everything Chapter 4 said about where Teammates live: on the work graph, with shared memory, multiplayer, and governed. An auditor is only useful if it knows your standards; a synthesizer is only trustworthy if it sees all the projects, not one person's slice.

Why you start at the workflow layer anyway

Two reasons, and they compound. First, trust: four small, observable jobs are easy to verify, and the confidence they build is what funds the judgment-layer conversation with your leadership. Second, and less obvious: the workflow layer creates the conditions the judgment layer needs. An audit agent can only audit work that flows through a consistent process. A synthesis agent can only correlate data that lives in structured fields. Read, check, route, and nudge are what put your work into the shape that judgment work runs on. Skip the foundation and the impressive agent has nothing solid to stand on.

The exercise, upgraded

Take one workflow your team runs every week and label each step. The first four labels are your automation roadmap: build those with rules and AI Studio, starting with the most boring. The judgment labels are your Teammate candidates, for after the foundation runs. And the human decisions are where your people should be spending their time anyway.

read

check

route

nudge

judgment work

human decision



Watch this next The intake examples above, auto-renaming and duplicate checking included, are demonstrated live in "Rules vs AI Studio vs AI Teammates."

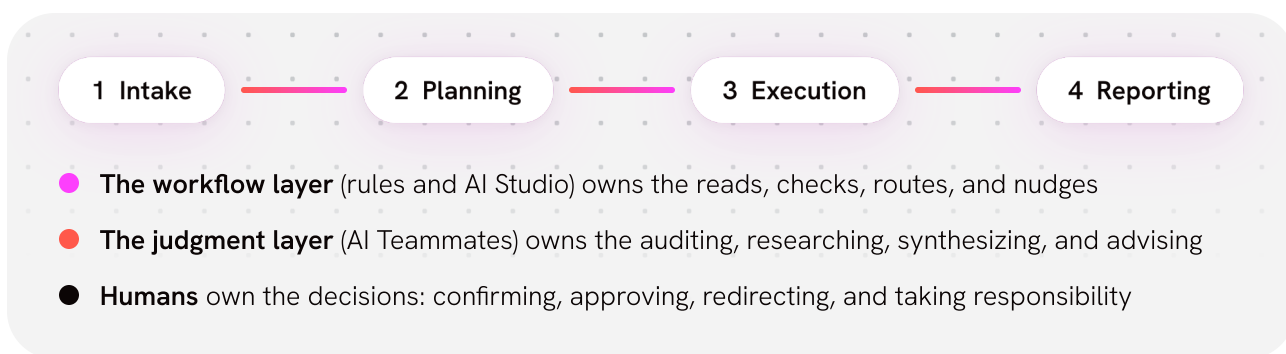
youtu.be/m6WS38eCuBY

Five workflows, **walked through**

Theory ends here. This chapter walks through real workflows the way they actually run, so you can see the pattern underneath and steal it.

The anatomy every workflow shares

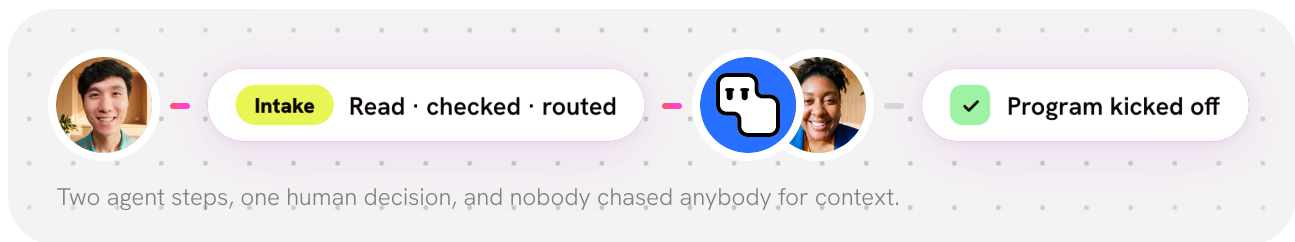
Strip away the department names and almost every business workflow has the same four phases: intake, planning, execution, reporting. Work arrives, gets shaped into a plan, gets done, and gets accounted for. And in an agentified workflow, every step in those phases has one of three owners:



Notice what this means: agentifying a workflow is not replacing the people in it. In every example below, a human sits at each moment that matters. What disappears is the connective drudgery between those moments.

Walkthrough: program management, end to end

Here is how the pattern plays out in a program management workflow, step by step.



Intake. A planning request arrives through a form or chat and lands in the workflow. The workflow layer reads it, checks for missing details and follows up with the requestor automatically, then classifies the program type and routes it to the right department. A program director receives a complete, classified request and confirms the initiative. Two agent steps, one human decision, and nobody chased anybody for context.

Planning. A business case builder drafts the program charter with ROI projections into a shared document. Executives leave feedback in comments. The builder incorporates the feedback and updates the charter. The program director reviews, approves, and kicks off the program. The judgment layer did the drafting and the redrafting; the humans did the judging.

Execution. A workflow optimizer watches the running program, surfaces process bottlenecks, and suggests changes. A decision tracker captures key pivots and decisions from meetings and notes, so the record keeps itself. The program director reviews and addresses what gets surfaced.

Reporting. A status reporter analyzes program health and performance across the connected projects. The workflow layer pushes a status update out to executives on schedule. Nobody spent Friday assembling a deck about work instead of doing it.

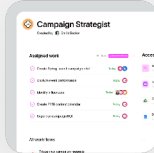
Count the human touches: **confirm, approve, review, address**. All judgment, no assembly. That is the shape you are building toward.

Four more, in brief



Marketing campaign intake and launch

The workflow layer reads incoming requests, renames them to convention, checks for duplicates and missing assets, and routes by campaign type: the intake pattern from Chapter 5, running at volume. The judgment layer drafts briefs from market context and builds the launch plan; marketers approve, create, and ship. Indeed runs a version of this at global scale: 1,500+ annual requests across 28 languages, routed to 80 portfolios, with a 60% faster path from raw request to active project and around \$300,000 a year saved.¹



Creative review and brand audit

The workflow layer moves assets through review stages and nudges approvers. The judgment layer does the part that never scaled before: auditing localized content against brand guidelines continuously, in every market, in every language: the job Indeed's Content Strategist Teammate does across 28 languages.¹ Humans stop checking translations and start directing creative.



Service and IT request triage

The workflow layer reads tickets, checks completeness, routes by type and urgency, and nudges on SLA risk. The judgment layer researches the hard ones across the company's institutional knowledge and drafts answers for review: the pattern behind KW Automotive's multi-lingual support Teammates, whose research their COO describes as beyond what busy human experts had time to produce.³ Humans handle the exceptions and the humans.



Content production pipeline

The workflow layer runs intake and evaluation of content requests; the judgment layer accelerates drafting and synthesis; editors keep taste and final say. Morningstar's research teams run an AI-powered content pipeline in this shape, part of how the company saves more than \$600,000 a year across its Asana workflows.²

The maturity ladder

Put the public case studies side by side and they form a ladder you can locate yourself on.

The first rung Morningstar · AI Studio at scale

Intake, request evaluation, and a content pipeline, no judgment work delegated, and over \$600,000 a year in value.² If you take one thing from this chapter, take this: the workflow layer alone is worth the climb.

The transition Indeed · Studio backbone + first Teammate

A mature Studio backbone (the rules-and-routing engine above) plus a first judgment-layer pilot, the brand auditor.¹ Foundation first, then one Teammate with a clearly scoped job.

Our Head of Delivery sees the same pattern in every build:

“Anytime you build a teammate, you’re solving a specific problem. The teammate has a specific role. Teammates are more successful when they’re dedicated and specialized, rather than a generalist you hope to pull into any situation.”



Brandon Llewellyn · Head of Delivery, Cirface

The destination KW Automotive · Teammates doing judgment work

Multi-lingual support research and proactive campaign analysis, on top of a company that first centralized everything (“Asana is our database,” as their head of organizational development puts it).³ The order was not an accident. The foundation came first.

You do not need to be on the top rung to win. You need to know which rung you are on, and what the next one asks of you, which is exactly what the readiness checklist in Chapter 8 measures.



Watch this next Marquis builds the intake pattern from the campaign workflow live in “Rules vs AI Studio vs AI Teammates.”

youtu.be/m6WS38eCuBY

What to get right before you add agents

This is the chapter most guides skip, because “you might not be ready” is a hard thing to sell. We are including it because we have watched what happens when teams skip it, and because the fastest way to sour an organization on agents is a first project that automates chaos.

Here is the principle everything below hangs on:
agents inherit your process problems.

An agent added to a workflow nobody has documented will faithfully execute the confusion. It will read inconsistent inputs inconsistently, check against criteria nobody agreed on, route work to owners who did not know they were owners, and nudge people about deadlines that were never real. The agent is not wrong. The process was, and now it runs at machine speed.

What ready looks like

Before an agent joins a workflow, the workflow needs five things. None of them requires AI, which is exactly the point.

01

The workflow is written down. Not in someone’s head, not “everyone kind of knows.” If you cannot draw the steps from intake to done, an agent cannot occupy one of them. The act of writing it down usually finds the problem an agent was about to inherit.

02

Every step has an owner. Including the agent’s steps, because someone has to own the agent: notice when its output drifts, update its instructions, answer for it in a retro. An agent without a human owner is an orphan process waiting to fail quietly.

03

The process is stable enough to trust. If the team changed how intake works three times this quarter, automate nothing yet. Agents amplify whatever exists. Stabilize first, then hand over.

04

The data is where the work is. An agent can only act on context it can reach. If half the decisions live in a chat thread and the other half in someone's inbox, the agent sees neither. Getting the work, the conversations about the work, and the context around the work into one place is not preparation for AI. It is the work AI runs on.

05

Governance exists before the first agent, not after the fifth. A short written policy answers: what data may enter AI tools, who may connect what, what each agent is scoped to, and who reviews it. Scope agents to the projects they need and nothing more. This is also the moment to bring IT and legal in as partners; a one-page policy they helped write clears the path for everything you build after.

“Do your research on what agents can and cannot do inside your tool before you build. And know how access works. It's complicated, but it's worth learning, because these things matter.”



Brandon Llewellyn · Head of Delivery, Cirface

The common first-30-days mistakes

We see the same handful on repeat.



Starting with the flashiest use case. The impressive demo, the agent that does everything, the workflow with the most exceptions. Start instead with the boring, high-volume, rule-shaped work: intake, renaming, duplicate checks, status collection. Small wins build the trust that funds bigger ones.



Letting agents multiply without a map. Three teams each quietly building their own version of the same assistant, on three sets of instructions, is how governance dies. Keep a simple registry: which agents exist, what each is for, who owns it.



Using AI where a rule would do. If the logic is fixed, build the rule. It is faster, cheaper, and never surprises you. Reaching for AI by default adds interpretation to steps that did not need any, and every unnecessary interpretation is a new way to be wrong.



No feedback loop. Teams launch an agent and stop looking. The teams that succeed treat the first month as training: reviewing output, correcting it in the open where the whole team can see, and tightening instructions weekly. Agents with shared memory get better exactly as fast as you feed them.

If you are not ready, that is a fine place to be

Not ready does not mean start over. It means your first project is process work, not agent work: document one workflow, name its owners, pull its context into one place, write the one-page policy. Teams can usually do this for a single workflow in a week or two, and it pays off whether or not an agent ever arrives.

The reverse is also true, and worth hearing from the person who runs our Office Hours:

“Sometimes customers are hesitant. They say, we are not ready, and maybe they are ready. With AI Studio and Teammates you need to explore: test the prompts, change the models, change the fields and actions. To be ready for that, you need to feel comfortable in the tool. It is not a matter of how many projects you have or whether your workflow is complex or simple.”



Juli Arenzo · Senior Solutions Engineer & Community Manager, Cirface

Then take the label exercise from Chapter 5, mark the reads, checks, routes, and nudges, and you have your first agent’s job description sitting in front of you.

The next chapter compresses all of this into a checklist you can put in front of your team.

The readiness checklist

Put it in front of your team, or hand it to an AI Teammate and argue with the results. A workflow is ready for its first agent when you can honestly tick every box in the first two sections.

The workflow

- ☐ The workflow is documented, end to end, and the team agrees the document is accurate
- ☐ Every step has a named human owner
- ☐ The process has been stable for at least a month (no structural changes mid-flight)
- ☐ The work, the conversations about the work, and the context around it live in one system the agent can reach
- ☐ Inputs arrive in a reasonably consistent shape (or the first agent job is making them consistent)

Governance

- ☐ A one-page AI policy exists: what data may enter AI tools, who may connect what, and who reviews it
- ☐ IT and legal have seen the policy and the plan
- ☐ The agent will be scoped only to the projects and data its job requires
- ☐ The agent has a named human owner who reviews its output and updates its instructions
- ☐ A simple registry exists of every agent: what it is for, who owns it

The agent's job

- ☐ The workflow's steps are labelled: read, check, route, nudge, or human decision
- ☐ The first agent job is boring, high-volume, and low-stakes (intake, renaming, duplicate checks, status collection)
- ☐ Fixed logic is built as plain rules, not AI
- ☐ Success is defined: what the team will measure after 30 days
- ☐ A weekly feedback rhythm is scheduled for the first month

Score yourself honestly

All boxes ticked: build. Start with the smallest read or check job on your list.

Workflow section incomplete: your first project is process work, not agent work. One to two weeks on a single workflow, then return here.

Governance section incomplete: stop before you start. Governance retrofitted after five agents exist is ten times the work of a one-pager written before the first.

Only the agent section incomplete: you are closer than most. Label the steps (Chapter 5) and the job writes itself.

Where to go next

You now have the argument, the taxonomy, the four jobs, and the checklist. Here is how to keep moving, in order of commitment.

See it live · Digital summit

Co-hosted with  asana

Asanathon 2026: Agentic Work in Action

Wednesday, September 30 · 3 hours, online · Free

Everything in this guide, built live. Watch a real workflow get agentified from intake to reporting, hear where agentic work management is heading next from the team at Asana, and then build one yourself: every participant gets their own demo space for the hands-on workshop, with our team guiding you step by step. You leave with a working workflow and the pattern to rebuild it on your own work.

Register now

cirface.com/agentic-work-in-action

Bring us a workflow

AI Studio Office Hours

Runs monthly and is exactly what it sounds like: bring a real workflow, real questions, or a half-built agent, and work through it live with our team. It is free, and the recurring questions from these sessions shaped half of this guide.

cirface.com/asana-webinars →

Talk to us about readiness

AI workflow consultation

If Chapter 8 told you your first project is process work, or you want a second set of eyes before your first agent ships, book a consultation. We will label the workflow with you, scope the first agent job, and leave you with a plan whether or not you build it with us.

cirface.com/contact →

If you are near New York

Work Innovation Summit New York · October 14 · Hilton Midtown

This guide shows you how agentic workflows work. WIS New York is where you build them. Asana's flagship event is a free full day of hands-on workshops, live demos, and sessions with the leaders putting AI Teammates to work. If you're in the greater New York area, we'll be there in person and would love to walk through your workflows with you.



[Register for WIS New York →](#)



Written by the team at Cirface, Asana Platinum Solutions Partner and Partner of the Year, Americas. Every workflow in this guide is one we run or have built for clients.

Case studies cited

- 1 **Indeed:** \$300,000/year saved, 1,500+ annual requests routed to 80 portfolios, 60% faster request-to-project lead time, 28-language brand audit.
asana.com/case-study/indeed
 - 2 **Morningstar:** \$600,000+/year in value from AI Studio intake, request evaluation, and an AI-powered content pipeline.
asana.com/case-study/morningstar
 - 3 **KW Automotive:** multi-lingual support Teammates, proactive Analyst Teammate; “Asana is our database,” says Mara Wohlfarth, head of organizational development.
asana.com/case-study/kw-automotive
-

Videos referenced throughout, on the Cirface YouTube channel: “Rules vs AI Studio vs AI Teammates” (youtu.be/m6WS38eCuBY), “Claude-Asana MCP vs AI Teammates” (youtu.be/WbzW7L1tTtY), and “How I Built an AI Agent to Triage Project Requests in Asana” (youtu.be/7UU5pGIYfY). Teammate role names (business case builder, status reporter, workflow optimizer, decision tracker) are written lowercase as roles, not product names.

© 2026 Cirface. The practical guide to agentic workflows, v1, July 2026. Asana, AI Studio, and AI Teammates are products of Asana, Inc.

Where humans + agents workflow together

Start smaller than your ambition. Start this week.



asana.com · cirface.com