OpenAI

A business leader's guide to working with agents

Preparing your teams for the shift to agents

Contents

Introduction

79%

of senior executives report that AI agents are already in use in their companies

PwC’s AI Agent Survey, 2025

86%

expect to be operational by 2027

PagerDuty

2/3

of executives believe agents will reshape the workplace more than the internet did

PwC’s AI Agent Survey, 2025

A year ago, the term “AI agent” was largely abstract, used more to describe potential than practice. Today, businesses are shifting from curiosity to integration: startups and enterprises alike are building agents, organizations are embedding them in workflows, and employees are already expanding the scope of what they can achieve.

Yet the term is still fuzzy. We’re regularly asked what an agent is, how to work with them, and how to plan for them.

In this guide, we will

01	Define AI agents and their core components
02	Show how they work and where they add value
03	Outline steps for delegating, supervising, and optimizing agents
04	Demonstrate how to organize agents and tips to avoid unnecessary complexity

This shift is happening quickly. When used well, agents free people to think deeply, work creatively, and solve problems together. Now is the time to experiment, understand how agents work, and create guardrails that ensure safe, valuable use.

There are many types of agents. This guide focuses on using pre-built agents like ChatGPT in agent mode, but you can also create custom agents - whether for internal productivity or to power new revenue-generating features in your product. For guidance on building custom agents, see our Practical Guide to Building Agents, or checkout AgentKit, our full-stack agent platform. We'll share more guidance on how to scope, build, and measure the success of agents in a future guide.

What is an AI agent?

An AI agent is a system that can plan, decide, and **act independently to achieve a goal** while operating within guardrails set by humans. Agents connect reasoning with tools, adapt mid-task based on information they encounter, and can carry complex tasks end-to-end.

Agents are built from three key elements:

01 **A model** Interprets instructions, plans steps, and decides what to do next.

02 **Tools** Connectors, APIs, or functions the agent can use to gather information, analyze data, or take action.

03 **Guardrails** Govern action, keep decisions aligned with human intent, policy, and ethics.

Together, these give agents the capacity for autonomous decision-making and action. We expand on how agents plan, act, and adapt, and how this differs from workflows in the sections ahead.

What AI agents are not

Many AI systems can answer questions or automate tasks, but they don’t plan, adapt, or carry work forward independently, that’s where agents come in.

For example, a Q&A chatbot can return answers and even make tool calls to gather the most accurate, up-to-date information—but it can’t plan or adapt. Imagine asking, “What’s our travel policy?” It could pull the latest version from your HR system and provide a clear answer, but it wouldn’t be able to plan a full offsite. It won’t find venues, compare costs, or draft an agenda unless every step is explicitly programmed.

Similarly, traditional workflow automations follow fixed rules. They can complete a checklist but struggle when conditions change. For instance, an automated expense-approval system might freeze if a new vendor field is blank while an AI agent can reason through the change and adapt instantly.

In contrast, agents combine a model, tools, and guardrails with the ability to decide and act at each step toward a goal.

Deep research as an agent

One way to see these elements in action is through deep research—an agent that combines models, tools, and guardrails to conduct research.

01 **Model** Reasoning model plans a research path, reasons through sources, and adapts if blocked.

02 **Tools** Browse the web, search with APIs, analyze documents, and source citations.

03 **Guardrails** Resist malicious prompts and enforce accurate results and citations.

How models shape an agent’s abilities

The underlying model is what gives the agent its ability to interpret goals, break them down into steps, and adjust its approach as new information appears. It selects the right tools and revises its plan as it works.

In other words, the model gives the agent judgment, the ability to choose and adapt its path rather than simply execute set rules.

For example, with the right model, an agent can take a high-level objective like “build a market research report” and break it into smaller actions: gathering data, summarizing findings, and generating insights.

Where a traditional workflow might stop when a data source is missing, an agent recalculates — finding alternate data or trying a new approach. Like a GPS adjusting to a blocked road, its reasoning ensures progress even when conditions shift.

Example: If asked to report on the TAM for AI in Australia, an agent could map a research path, pull data, and produce a summary. If blocked by a paywall or missing numbers, it could draw on alternate sources or shift to qualitative signals. A workflow would stop, an agent keeps working towards an answer.

Memory

Memory is becoming a critical part of how agents become successful. Agents that recall past steps can refine their approach, much like an employee who learns from experience. As memory improves and can factor in different tools and data sources agents will become even smarter, and users will be able to teach agents directly, or train them on their day to day tasks.

When do agents start and end?

Today, agents run when prompted or scheduled. In the future, they may self-initiate based on goals, context or even new events or learnings, preparing materials ahead of meetings on your calendar or analyzing updates continuously in the background.

Tools

While the model provides reasoning and judgment, tools let an agent act. They turn plans into actions, helping the agent gather data, analyze information, and perform tasks.

What makes agents distinct is not just their access to tools, but the model's ability to decide which tool to use, and when. This combination is what allows an agent to operate independently, reroute when conditions change, and complete work that would otherwise stall.

Tools can include:

- Access or retrieval of data from internal systems
- Editing or changing information in internal systems
- Function calling or API access
- Image generation, search, or data analysis
- Executing code in a secure terminal
- Using a computer or interacting with websites
- Or even *other* agents (e.g., Deep Research, Operator) via orchestration

This isn't an exhaustive list. The tools available to agents are expanding quickly and will continue to broaden in the future.

Today, agents range from those using a single tool, like search, to advanced agents chaining multiple tools to complete complex tasks.

How agents differ from other AI systems

Type of agent	Number of tools	Types of actions
Single-tool agent	Runs one tool	• Web search + analysis • Deep research
Multi-tool agent	Chains tools	• Search + analysis + website interaction • ChatGPT agent

As models improve, agents will become more capable with whatever tools they have. A smaller toolkit won't limit performance. What matters is how effectively the model uses the tools available.

In practice, an agent with just computer access and API connections might accomplish more than another with a larger set of specific tools. In the future, agents may also run multiple tools in parallel, widening the range and pace of what they can achieve.

How it works in practice

ChatGPT agent when performing market research might combine web search, data analysis, and image generation—pivoting between tools to gather data, analyze trends, and create a presentation-ready summary.

Tools give an agent the ability to act, but these actions need boundaries. In the next section, we'll explore guardrails, explaining how agents use tools safely and with the right oversight.

Guardrails

Because agents can plan, adapt, and act across multiple tools, they require guardrails — clear boundaries that ensure safe use and human oversight, paired with system-level guidance that helps them make sound decisions.

Guardrails are the rules and safety mechanisms that define how an agent behaves. They operate at the model and application levels, shaping both what the agent can do and how it interacts with its environment. Instructions complement these boundaries by guiding how agents approach decisions, helping them act in line with business goals as well as safety requirements.

Guardrails ensure safe use of agents, allow for human supervision, and provide an audit trail.

For example, ChatGPT agent has been trained on the following guidelines:

- Seek confirmation before many real-world actions.
- Ask for oversight on sensitive tasks (e.g., sending emails).
- Refuse high-risk actions (e.g., financial transfers).
- Resist malicious prompts.
- Limit data sources to approved sets.
- Restrict where/how it can act.

Reliable guardrails provide clear limits and oversight, while still giving the agent room to carry out meaningful work independently.

For leaders, balancing guardrails with agent independence protects compliance, quality, and reputation. As agents grow more capable and their scope of tasks expands, guardrails and instructions will become even more essential to safe operation.

Customized guardrails and instructions

Today, customization of guardrails and instructions in pre-built agents is limited, so choosing a well-designed option matters. Over time, more customization will move into the application layer. Some organizations will build agents for full control, trading speed for effort. Our newest agent builder tool, AgentKit, gives more granular control and customization over guardrails for any agent you build.

Workflow automation vs. agents

Workflow automations, agents, and even LLM powered steps are often confused. They each can play a role in getting work done and can be used together.

Workflow automations follow predefined steps and rules, sometimes with simple “if X, then Y” logic. They are reliable for repetitive, stable tasks like routing support emails or auto closing a ticket when a condition is met. The tradeoff is rigidity. Building and maintaining rules takes time, and workflows often break when conditions change. Their strength is auditability since every rule and outcome is predetermined.

LLM-powered steps within a workflow add flexibility without changing the overall structure. The workflow remains rule based, but an LLM can handle a single step that requires interpretation, such as classifying a request, summarizing a document, or extracting fields from an attachment. The LLM does not plan or take multiple steps on its own. It performs one judgment step and then hands control back to the workflow. This works well when the process is mostly predictable but benefits from occasional reasoning.

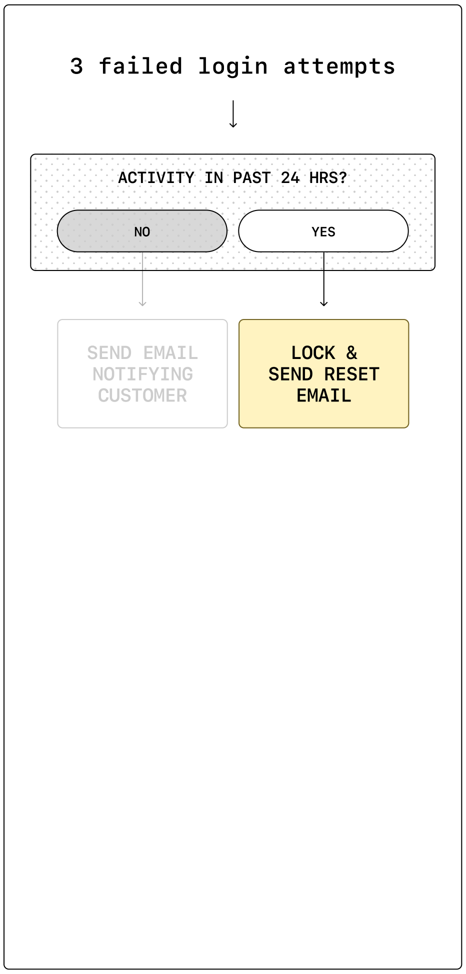
Agents start with a goal, then plan their work, select the right tools, and use data and take action. They can adjust their approach when conditions change or when new information is received, and they can pause to request clarification before continuing. This allows them to reroute their plan and keep work moving without every step being defined in advance, reducing rework and allowing teams to focus on higher value work.

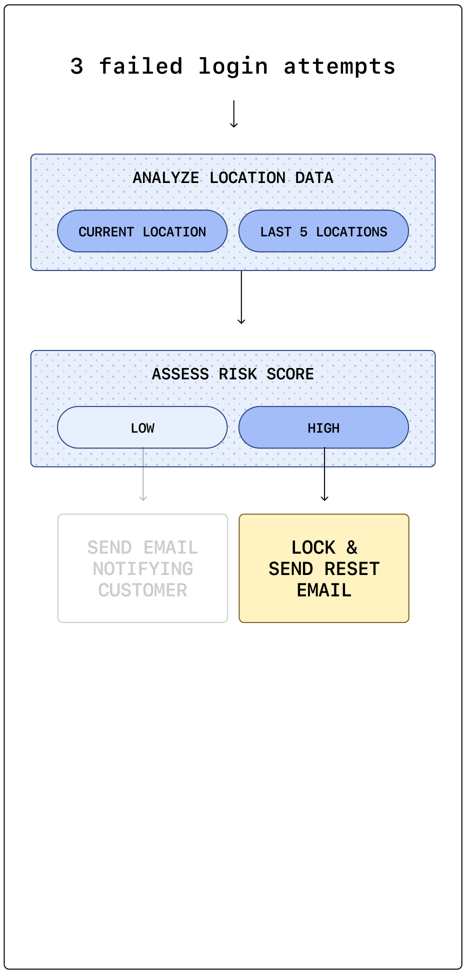
When to use each

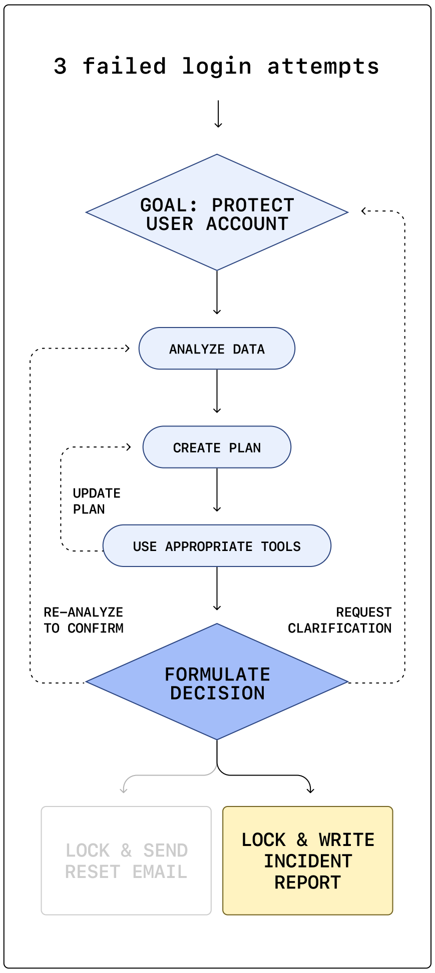
Workflow automations suit predictable, repetitive tasks but require setup and ongoing maintenance.

LLM-powered steps suit workflows that need occasional interpretation or judgment without redesigning the full process.

Agents suit work that requires adaptation, context, or multi-step reasoning.







When to use each, cont.

Continuing the examples from the previous page:

Workflow automation	After three failed logins and no activity in 24 hours, the system follows a fixed rule: lock the account and send a reset email.
LLM-powered steps	After three failed logins, an LLM handles one step in the workflow: interpreting recent location data and risk level to guide whether the system should lock the account
Agent	After three failed logins, an agent works toward a goal, analyzing data, using tools, and updating its plan before deciding and documenting what to do.

The three can also complement each other, working together to take on more complicated workflows.

As agents evolve, they will take on more structured, workflow-like behaviors while still retaining the ability to adjust when conditions change. For leaders, this means more consistent execution alongside the flexibility to handle the unexpected.

Moving from definition to practice

Agents combine planning, tools, guardrails, and instructions to operate independently and adapt to your goals as conditions change. For employees, this means less repetitive work and more time for judgment, creativity, and strategy. For the business, it means higher-quality outputs delivered consistently and safely, while ensuring compliance and protecting reputation.

Agents are advancing quickly, and starting now makes it easier to adapt as they evolve. With the basics in place, we can look at how teams can best work alongside them.

Preparing your teams to work with agents

As agents enter daily work, the focus shifts from what they can do to how people and agents work together. Teams learn best by doing. Trying new tasks, giving direction, reviewing the outputs, and then adjusting their prompts and instructions. This practice builds new instincts around judgment and delegation and builds AI literacy.

Building momentum with employee agents

Early progress often comes from putting agents directly into daily work. Employees know the steps, decisions, and data behind their tasks better than anyone, so they are best placed to find where agents help.

For example, a finance team might use an agent to:

01 Size a new market (TAM, SAM, SOM)

02 Factor in cultural, competitive, and logistical context

03 Pull relevant data from internal systems

04 Draft a first-cut report, dataset, or slide deck

The employee then reviews, adds context, and refines the output. Each cycle builds more AI literacy, and highlights repeatable patterns in the work. As these patterns emerge, teams can capture them, share their work, and start forming a library of agent-supported workflows that can be scoped and scaled.

To speed up this experimentation, it helps to guide teams on how to delegate effectively to agents.

Teach teams how to delegate to agents

Clear instructions lead to better results. Teams get more from agents when they share the context, expectations, and structure behind a task, just as they would with a colleague.

For example:

- Ask the agent to consider only peer-reviewed sources
- Specify which internal systems to read for context
- When requesting a spreadsheet, define tabs, columns, and data

While agents can produce useful results with open-ended prompts, this level of guidance improves both the accuracy and consistency of their work.

Here’s how adding structure can change the output. Compare:

“Plan my team’s quarterly offsite.”	**“Plan my team’s quarterly offsite** • **Start** by finding 3–5 venues within 50 miles that can host 25 people and provide catering. • **Create** a table comparing location, cost, and amenities. • Then **draft a sample agenda** with 3 team-building activities and 2 workshop sessions based on our H2 priorities doc. • Finally, **draft an email invitation** that includes the date, venue options, and a short agenda summary."

With agents, employees can now front-load the steps and tasks required, review the output, and focus on higher-value tasks while the agent handles the busywork.

Action item

Ask employees to sketch out broader workflows into the following elements:

1. The key use cases or tasks they might string together.
2. The data sources they’d need to use.
3. The actions that need to occur.
4. The output style and format most valuable to them.

Capture these use cases in a shared database. This makes it easier to:

- Spot opportunities to reuse use cases across teams.
- Identify where use cases can be scaled into operational processes.
- Assess which patterns might evolve into IT-owned agent projects.

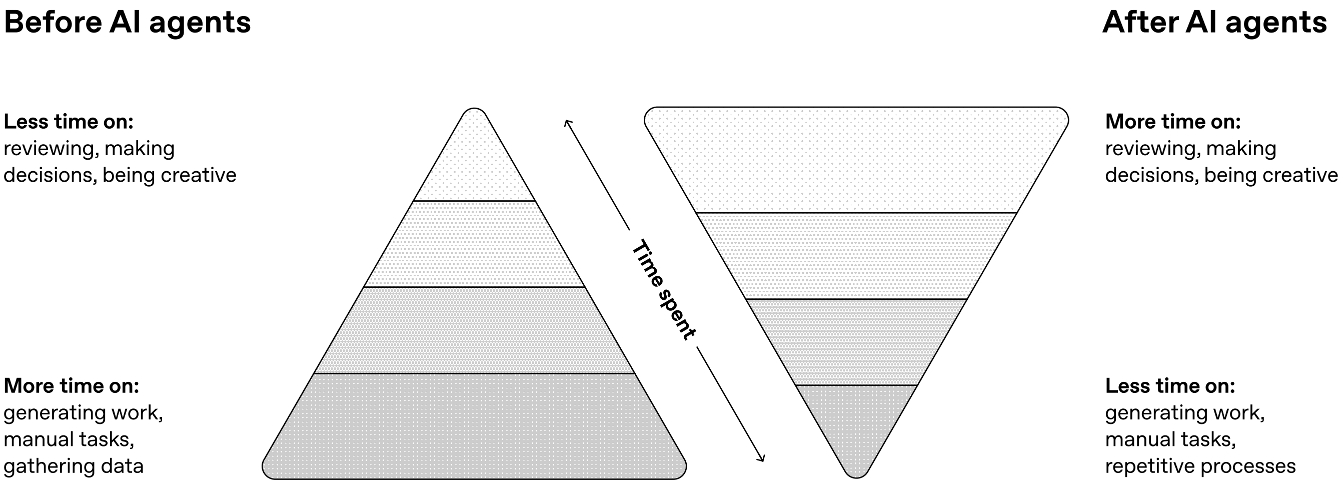
By building this habit, employees shift their role from creating work to reviewing and refining it, while organizations build a growing library of reusable, scalable use cases.

Shifting from generating work to reviewing and acting

The shift from creating work to reviewing it reduces busywork and creates more room for judgment, strategy, and creativity. It also calls for new skills and a different mindset.

As agents take on more execution, employees will spend more time shaping work than producing it. They will set clear briefs, assess output, and guide revisions. The same qualities that define strong managers today, such as clear communication, sound judgment, and the ability to coach quality, will matter even more in an agent-supported environment.

This shift enables faster decisions and gives teams space to explore more ideas. When time moves from manual work to thoughtful review and action, organizations respond more quickly to new opportunities and challenges.


Before AI agents
After AI agents
Less time on:
reviewing, making decisions, being creative
More time on:
generating work, manual tasks, gathering data
Time spent
More time on:
reviewing, making decisions, being creative
Less time on:
generating work, manual tasks, repetitive processes

> Our sales team now spends more time refining talking points and supporting customers directly, because agents prepare them for key meetings in advance. With pre-meeting briefs, account insights, and recaps handled automatically, reps can focus on strategic conversations instead of manual research.

Encourage a culture of supervision and auditing

As teams shift from generating work to reviewing it, auditing and refining outputs should be a standard part of every workflow.

After an agent completes a task, have employees:

Review the chain of thought	Understand the steps taken and tools used.
Check the output	Confirm quality, completeness, and that citations or references are correct.
Request changes	Specify what needs revision, expansion, or refinement.

Pair newer employees with experienced reviewers to walk through outputs together. This develops taste, standards, and confidence — and improves the work at the same time.

Bonus tip: Build craft while reviewing

Pair newer employees with senior staff to review an agent's output together. Have the senior walk through the reasoning, highlight strengths, weaknesses, explain changes, and discuss why they matter. This critique will improve your next prompt and workflow, while helping newer employees develop judgment and a sense of quality.

Start thinking about how you'll organize different agents

As agents become more capable, some businesses will deploy many of them to handle a wide range of work. When adding new agents, it helps to think about how they are structured and coordinated across the organization.

Adding or orchestrating multiple agents makes the most sense when you want to:

- Increase speed by running work in parallel
- Build agents with deep domain expertise
- Add safeguards such as permissions and verification for sensitive tasks

As agents expand in scope, they will interact with different teams and support different types of work. Establishing a simple structure early makes it easier to grow responsibly and avoid fragmentation as capabilities mature.

Some broader agent categories may emerge:

Category	Example
Employee agents that help teams get work done	ChatGPT agent
Domain agents for high-privilege or deep-expertise tasks	Deep research, Operator, or a homegrown agent
Operational agents that run behind the scene tasks	Compliance monitoring tools, IT automatons, or internal builds
Customer-facing agents	AgentKit: Support agents, sales assistants

If you're building in-house agents, especially those that will support customers or handle more sensitive work, our Practical Guide to Building Agents offers deeper guidance on design, safety, and deployment.

As you introduce new agents, a few simple questions help keep things clear and manageable:

Purpose	What job is this agent responsible for?
Scope	Which data, tools, and actions should it access?
Ownership	Who maintains and improves it over time?
Coordination	How will it interact with people, workflows, or other agents?
Lifecycle	When should it evolve, consolidate, or retire?

Over time, some agents will take on broader tasks while others stay specialized. A thoughtful structure makes it easier to grow responsibly and ensures choices made today remain workable as capabilities advance.

Set clear guidance on how to measure agent performance

Measuring ROI starts the same way it does for any new tool: establish a baseline for how the work happens today. Ask teams to capture typical time spent, cost, and accuracy so you have a clear point of comparison. This will help you understand where to expand use.

Agents often also improve iteratively. Instructions become clearer, tasks can expand, and more data sources and tools can be added. Because of this, check results across multiple cycles rather than only after the first attempt. Look for steady improvements in speed, quality, and reliability, and consider scaling agents that consistently deliver value across teams.

Conclusion

Agents are quickly reshaping how work happens. This is a good moment to look closely at where your team's time goes today, the routines, handoffs, and moments of friction, and start imagining what could shift. With thoughtful experimentation, teams can uncover where agents remove effort, create space for deeper thinking, and support better decisions.

As adoption grows, agents will sit alongside everyday tools, helping ideas move faster and giving people more room for creativity, curiosity, and judgment.

The opportunity ahead is to make work not just more efficient, but more thoughtful and imaginative. Beginning now gives your teams space to learn, adapt, and shape this new way of working today.