Founders AI
founders-ai.dev

PRACTICAL AI FIELD GUIDE

AI Agent System

How to Delegate Work Without Losing Control

A practical delegation guide for founders, business owners, creators, consultants, and professionals who want AI to handle repeatable work without losing control, approvals, or verification.

- Pick the right task
- Define the job
- Set permissions
- Add approval gates
- Verify outputs
- Increase autonomy slowly

Most people hear “AI agent” and imagine a digital employee that can run entire workflows by itself.

That is the wrong starting point.

An AI agent is better understood as a scoped worker. It has a job, instructions, context, tools, limits, approval gates, and a review process. If you remove the limits, you do not get a better employee. You get a faster mistake machine.

The goal of this guide is to help you delegate repeatable work to AI without handing over control too early.

The rule

Do not start by asking, “What can AI do for me?”

Start by asking, “What work can I safely delegate, verify, and improve?”

That one question keeps you out of the two traps most beginners fall into:

- They use AI like a chat box forever and never build leverage.
- They give AI too much autonomy too early and stop trusting it after one bad result.

The middle path is the agent system.

What an AI agent actually is

For practical use, an AI agent is a system that can:

- Understand a goal.
- Break that goal into steps.
- Use tools or information.
- Make progress across multiple actions.
- Report what happened.
- Ask for help or approval when needed.

That does not mean it should be allowed to do everything. Research-backed agent design emphasizes clear goals, task decomposition, tool use, logs, monitoring, and human approval for high-impact actions (IBM).

Think of an agent as an assistant with a checklist, not an owner with a blank check.

The three levels of AI help

Before you build an agent, decide whether you actually need one.

Level 1: Prompt

Use a prompt when the task is short, low-risk, and does not require tools.

Good examples:

- Rewrite this email.
- Summarize this document.
- Give me five title ideas.
- Explain this concept in plain English.

If the task can be completed in one answer, you probably do not need an agent.

Level 2: Workflow

Use a workflow when the task has a predictable sequence.

Good examples:

- Take a call transcript, extract action items, draft a follow-up email, and create a CRM note.
- Take a blog post, create five LinkedIn posts, three X posts, and one email.
- Take a lead list, classify the accounts, and draft research notes.

Anthropic describes workflows as predictable systems where LLMs and tools follow predefined code paths, while agents make dynamic decisions about their own process (Anthropic).

If the steps are mostly the same every time, build a workflow before you build an agent.

Level 3: Agent

Use an agent when the task is open-ended, multi-step, and requires judgment about what to do next.

Good examples:

- Research this market and keep looking until the answer is complete.
- Review these files, find the issue, propose a fix, and ask before editing.
- Monitor a process and escalate when something looks wrong.
- Compare multiple options, identify gaps, and request missing information.

Agents are useful when you cannot fully predict the number of steps in advance, but that flexibility also creates more chances for compounding errors (Anthropic).

The Agent System Loop

Use this loop before giving AI more autonomy.

Choose one job

Bad agent instruction:

“Help me run marketing.”

Good agent instruction:

“Review one video transcript, extract five short-form clip ideas, write hooks for each clip, and flag anything that needs human judgment.”

An agent should own a task, not a department.

Define the finish line

An agent needs to know what “done” means.

Define:

- The output format.
- The quality bar.
- The required sources.
- The review checklist.
- The handoff point.

If you cannot describe the finish line, the agent will invent one.

Set the context

Agents perform better when they are not forced to guess.

Give them:

- Business context.
- Audience.
- Offer.
- Brand voice.
- Examples of good output.
- Examples of bad output.
- Current constraints.
- Known risks.

Context is the difference between a generic assistant and a useful one.

Set permissions

This is where most people make the biggest mistake.

Do not give an agent every tool just because the tool exists. Give it the minimum access needed to do the job.

Use this permission ladder:

Level	Permission	What it can do	Human role
1	Read-only	Read, summarize, classify, compare	Review output
2	Draft-only	Create drafts, plans, checklists, recommendations	Approve or edit
3	Internal edit	Update low-risk internal docs or systems	Review change log
4	External action	Send, publish, post, buy, delete, or change live data	Approve before action
5	Autonomous execution	Act within preapproved limits	Audit and monitor

Most beginner agents should live at Levels 1 and 2.

Add approval gates

An approval gate is a rule that forces the agent to stop before a risky step.

Use approval gates before the agent:

- Sends a message.
- Publishes content.
- Deletes anything.
- Changes customer-facing copy.
- Modifies money, billing, legal, hiring, medical, or financial information.
- Contacts a customer, lead, partner, or employee.
- Uses private or sensitive data.
- Makes a claim that needs verification.

Human approval is especially important for high-impact actions, including mass emails and financial actions (IBM).

Add verification

Verification is not optional.

NIST flags confabulation, information integrity, privacy leakage, prompt injection, and automation bias as important generative AI risks (NIST). That means your agent needs a check before you trust the result.

Verification can include:

- Source checks.
- Math checks.
- Duplicate checks.
- Policy checks.
- Brand voice checks.
- Sensitive data checks.
- “Would I send this under my name?” checks.

The agent can help with verification, but it should not be the only reviewer for important work.

Keep a run log

Every useful agent should leave a trail.

The run log should capture:

- The task.
- The inputs.
- The tools used.
- The decisions made.
- The output created.
- Any uncertainty.
- Any approval requested.
- Any error or blocker.
- The final human decision.

Logs make the system improvable. Without logs, you only have vibes.

Review the failure

When the agent messes up, do not just blame the model.

Ask:

- Was the task too broad?
- Was the context missing?
- Were the tools unclear?
- Were the permissions too loose?
- Was the output format vague?
- Was the approval gate missing?
- Was the verification checklist weak?

Most agent failures are system failures.

Increase autonomy slowly

Autonomy should be earned.

Use this progression:

1. Watch the agent do the task.
2. Let the agent draft the output.
3. Let the agent complete the task with approval.
4. Let the agent complete low-risk tasks within limits.
5. Let the agent run automatically only after repeated successful runs.

Microsoft's agent maturity model makes the same point at the organizational level: teams should move from experimentation to repeatable, governed, measured adoption before pushing into deeper autonomy (Microsoft Learn).

Agent readiness scorecard

Before turning a task into an agent, score it.

Question	Yes	No
Is the task repeatable?		
Can the goal be clearly defined?		
Can the output be reviewed quickly?		
Are the inputs available and clean?		
Can the agent work read-only at first?		
Can mistakes be reversed?		
Can risky steps require approval?		
Is there a clear log of what happened?		
Is there a human owner for the final decision?		
Is the upside worth the setup time?		

If you have fewer than seven "yes" answers, do not build an autonomous agent yet. Use a prompt or workflow first.

Good first agents

Research collector

Job:

Find sources, extract useful notes, organize them, and identify missing information.

Why it works:

The agent can do heavy collection work while the human still judges the final answer.

Permission level:

Read-only.

Approval gate:

Human approves any published claims.

Meeting prep assistant

Job:

Review notes, prior conversations, website pages, CRM data, and recent activity. Create a meeting brief.

Why it works:

The output is useful even if it is imperfect because the human can review it before the meeting.

Permission level:

Read-only or draft-only.

Approval gate:

Human reviews before sharing externally.

Follow-up drafter

Job:

Turn meeting notes into a follow-up email, action list, and CRM note.

Why it works:

The structure is repeatable and the human can approve before sending.

Permission level:

Draft-only.

Approval gate:

Human sends the email.

Content repurposing assistant

Job:

Turn one long video, call, article, or guide into shorts ideas, social posts, email copy, and newsletter sections.

Why it works:

The agent saves time without needing permission to publish.

Permission level:

Draft-only.

Approval gate:

Human approves anything public.

Operations checklist assistant

Job:

Turn messy process notes into SOPs, checklists, and handoff templates.

Why it works:

The agent helps standardize work while the human confirms the real process.

Permission level:

Draft-only or internal edit.

Approval gate:

Human approves before team adoption.

Bad first agents

Do not start with agents that:

- Send cold email automatically.
- Publish directly to social media.
- Reply to customers without review.
- Handle billing or refunds.
- Make hiring decisions.
- Give legal, financial, medical, or compliance recommendations without expert review.
- Delete or overwrite important data.
- Operate across too many tools at once.

These can become future systems, but they should not be your first systems.

The Agent Charter

Use this before creating any agent.

```
Agent name:

Primary job:

Who it serves:

What a successful run produces:

Inputs it can use:

Tools it can use:

Tools it cannot use:

Data it must not access:

Actions it can take without approval:

Actions that require approval:

Actions it must never take:

Output format:

Quality checklist:

Verification steps:

Escalation rules:

Run log requirements:

Human owner:
```

If you cannot fill this out, the agent is not ready.

Prompt: decide if this should be a prompt, workflow, or agent

```
You are helping me decide how to use AI safely and effectively.

Task I want help with:
[describe task]

Context:
[describe business, audience, tools, risk, deadline]

Classify this task as one of three options:
1. Prompt
2. Workflow
3. Agent

For your answer, include:
- Best classification
- Why
- Risk level
- What the AI should be allowed to do
- What should require human approval
- What verification is needed
- The simplest version I should start with
```

Prompt: create an agent charter

```
You are an AI operations architect.

Help me design a safe agent charter for this task:
[describe task]

Create a practical charter with:
- Agent name
- Primary job
- Inputs
- Tools
- Permissions
- Forbidden actions
- Approval gates
- Output format
- Verification checklist
- Escalation rules
- Run log format
- First three test runs

Assume I am starting with low trust and want to increase autonomy only after repeated successful runs.
```

Prompt: permissions matrix

```
Build a permissions matrix for this AI agent:
[describe agent]

Use these permission levels:
1. Read-only
2. Draft-only
3. Internal edit
4. External action with approval
5. Autonomous within limits

For each tool or action, tell me:
- Current permission level
- Why
- Risk if it fails
- Required approval gate
- Verification method
- What evidence would justify increasing autonomy later
```

Prompt: approval gate

```
Before continuing, stop and ask for approval if any of these apply:
- The action sends, posts, publishes, deletes, buys, changes live data, or contacts another person.
- The action involves money, legal, medical, hiring, compliance, customer data, or private information.
- The answer depends on facts that need verification.
- The task has unclear instructions.
- A mistake would be difficult to reverse.

When you stop, provide:
- What you plan to do
- Why approval is needed
- The exact content or action to approve
- The risk if wrong
- A safer alternative
```

Prompt: verification checklist

```
Review this agent output before I use it:
[paste output]

Check for:
- Unsupported claims
- Missing sources
- Math errors
- Overconfident wording
- Privacy or sensitive data issues
- Brand voice mismatch
- Missing steps
- Risky recommendations
- Anything that should require human approval

Return:
1. Pass/fail
2. Required fixes
3. Optional improvements
4. Final safe-to-use version
```

Prompt: run log

```
Create a run log for this AI task.

Include:
- Task
- Start time
- Inputs used
- Tools used
- Steps taken
- Decisions made
- Output created
- Uncertainties
- Approval requests
- Errors or blockers
- Final result
- What to improve next time
```

Prompt: post-run review

```
Review this completed agent run:
[paste run log and output]

Tell me:
- What worked
- What failed
- What was unclear
- What the agent should do differently next time
- What instructions should be added
- What permissions should stay the same
- Whether autonomy should increase, decrease, or stay unchanged
```

The 30-minute setup

If you want to apply this today, do not try to build ten agents.

Build one.

Choose a task you already do every week.

Good options:

- Meeting prep.
- Follow-up drafts.
- Content repurposing.
- Research collection.
- SOP creation.
- Weekly planning.

Then do this:

1. Write the agent charter.
2. Keep it read-only or draft-only.
3. Run it on one real task.
4. Use the verification checklist.
5. Save the run log.
6. Review what failed.
7. Improve the instructions.
8. Run it again.

That is how you build useful AI systems. Not by trusting AI more, but by making the work easier to inspect.

The one-page version

Use this when you are about to delegate work to AI.

Step	Question
Task	What exact job should the agent do?
Finish line	What does “done” look like?
Context	What does the agent need to know?
Tools	What can it access?
Limits	What can it not do?
Approval	What requires human permission?
Verification	How will the output be checked?
Log	What record will it leave behind?
Review	What will be improved after the run?
Autonomy	Has it earned more freedom?

Final reminder

The goal is not to replace your judgment.

The goal is to stop using your judgment on work that can be prepared, organized, drafted, checked, and handed to you in a better state.

That is the real value of an AI agent system.

You do not need to trust AI blindly. You need to build a system where trust is earned.