

FREE SAMPLE

The Agent Readiness **Kit**

Checklists, contracts and drills for putting AI agents into production.

Contents

01	How to use this kit	
02	Write the architecture contract	FULL KIT
03	Map tool permissions and blast radius	FULL KIT
04	Put approvals and dry runs where they matter	FULL KIT
05	Build an eval suite before you need one	FULL KIT
06	Run failure drills	FULL KIT
07	Observe without hoarding	FULL KIT
08	When it breaks: quarantine mode and incident response	FULL KIT
09	Prove it pays: the ROI and cost ledger	FULL KIT
10	The readiness review	FULL KIT

This free sample includes chapter 1 and the first 12 items of the 60-point readiness checklist. The full kit adds chapters 2 to 10 and 14 editable templates in Markdown, YAML and CSV.

Licensed for use inside your organization. Please don't resell or redistribute. Questions:
guide@alchemicttechnology.com

01

How to use this kit

Most agent prototypes stall in one of two ways. Someone hands them production credentials too early and they do something expensive, or nobody can say when they are safe enough, so they sit in a demo folder for months. This kit gives you a way to answer "is it ready, and for what?" with evidence instead of a feeling.

This guide explains the reasoning, one chapter per template. The templates are the work. Copy the `templates/` folder into the agent's repo and fill it in there, so a limit change lands in the same pull request as the code that enforces it.

The readiness ladder

Every chapter in this kit refers to four levels. They describe how much a human is involved in each action the agent takes.

LEVEL	WHO ACTS	WHAT HUMANS DO	TYPICAL USE
L0 Shadow	Nobody. The agent only drafts.	Do all the real work, compare against the drafts	New agents, new tools, after an incident
L1 Supervised	The agent, after approval	Approve or reject each action before it runs	Any write action that has no track record
L2 Bounded	The agent, alone, inside numeric limits	Handle everything outside the limits, review samples	Write actions with a clean L1 record
L3 Autonomous	The agent, alone, across its scope	Review samples and exceptions	Cheap, reversible actions with a long clean record

L0 Shadow

At L0 the agent runs on real inputs but nothing it produces reaches a customer or a system of record. Refund Desk spent its first weeks here, writing a proposed reply and a proposed action, such as "refund \$84.20 on order 10432", into an internal note on each ticket. Priya's team worked the ticket as usual and could see what the agent would have done.

L0 is cheap, and it is where you collect the material for everything else. The cases where the agent and the human disagreed become your first golden tasks in template 04. The token spend per ticket becomes the first line of your cost ledger in template 09.

Leave L0 when the drafts are good enough that a reviewer would rather approve them than write from scratch, and every tool the agent calls is either read-only or writes somewhere private.

L1 Supervised

At L1 the agent prepares a real action with real parameters, a human approves it, and the system executes exactly what the human saw. That last part matters. If the reviewer approves "refund \$84.20" and the agent can still change the amount after approval, your approval step is decoration. Chapter 04 covers how to bind an approval to its parameters.

The main risk at L1 is rubber-stamping. After the hundredth correct proposal, people stop reading. Track approvals, edits and rejections per action type. An edit tells you more than a rejection, because it shows exactly what the agent got wrong.

Leave L1 for a given action type when reviewers approve it without edits almost every time, and you can describe every rejection as a rule. "We rejected it because the customer had already been refunded" is a rule you can enforce in code. "It felt off" is not.

L2 Bounded

At L2 the agent acts alone inside limits written as numbers: amount per action, total per day, which tools, which hours, which order states. Anything outside those limits goes to a human. Refund Desk's L2 limits are refunds of \$250 or less, one refund per order, one automatic refund per customer per 30 days, \$1,500 in total per day, and return labels only for orders delivered in the last 30 days.

Enforce the limits outside the model, in the code that wraps each tool. A limit that lives in the system prompt is a suggestion. A cleverly written customer email can talk the model out of it, and you will not know until the payment processor report arrives.

Leave L2 only when you have weeks of clean sampled reviews, at least one failure drill passed, and a kill switch you have actually flipped.

L3 Autonomous

At L3 the agent acts alone across its whole scope. Humans review a sample and every exception. Hard ceilings still exist, but they are safety rails, not the line of trust.

Few agents that move customer money belong at L3 for anything but small amounts. L3 fits actions where the worst outcome is cheap and easy to undo, such as tagging tickets, drafting internal notes and looking up orders. If you cannot describe how you would reverse a bad L3 action in one sentence, it is not an L3 action.

Levels belong to actions, not agents

You do not promote an agent. You promote an action type. "Refund Desk is at L2" is shorthand for "the riskiest thing Refund Desk does alone is at L2". The real picture looks like this:

REFUND DESK ACTION	LEVEL TODAY	TARGET AFTER THIS KIT
Look up an order	L3	L3
Draft a reply	L3	L3
Send a reply from an approved template	L1	L2
Issue a refund of \$250 or less	L1	L2
Issue a refund over \$250	L1	L1, permanently
Create a return label	L1	L2
Change an order address	Never	Never

This stops you from holding back a whole agent because one action is risky, or promoting one because most of its actions are safe.

Deciding which level an action is ready for

Ask these five questions about each action. Answer them with logs and code, not opinions.

- 1. Can you undo it, and how fast?** A draft can be deleted. An email cannot be unsent. A refund can rarely be clawed back in practice. Irreversible actions need more evidence at every step.
- 2. What is the worst case before someone notices?** That is blast radius, covered in chapter 03. Multiply the worst single call by how many calls can happen before a human looks.
- 3. What does the level below show?** Promotion to L2 needs L1 approval logs. Promotion to L1 needs L0 drafts. No logs, no promotion.
- 4. Can you see every action?** Each action needs a trace with its input, the tool call, its parameters, and the result. Chapter 07 covers this.
- 5. Can you stop it in minutes, and have you tried?** A kill switch you have never flipped is a hope. Chapters 06 and 08 cover drills and quarantine.

Three rules follow. Promote one level and one action type at a time. Never skip L1 for a write action. Write the demotion triggers down before you promote, while everyone is calm.

Here is the bar Northwind set for moving refunds of \$250 or less from L1 to L2:

- 150 consecutive approved refund proposals under \$250 with no amount edits.

- Every rejection in the L1 period mapped to a rule now enforced in the refund tool wrapper.
- The \$250 cap and the \$1,500 daily cap enforced in code, with tests, and an alert to Priya when automatic refunds pass \$1,000 in a day.
- The payment partial-success drill, card D02 in template 06, run in staging and passed.
- The kill switch flipped in production during a quiet hour, with time to stop recorded.

And their demotion triggers: any refund that breaks a limit, any refund to the wrong order, or two customer complaints about refund amounts in one week. Any of these drops refunds back to L1 until the readiness review in template 10 clears them again.

Who fills what

A small team can run this kit with four roles. One person can hold more than one, except that the reviewer should not be the person who built the agent.

- The **agent owner** is accountable for what the agent does to customers and the business. At Northwind that is Priya, the support operations lead.
- The **engineer** builds and runs the agent. At Northwind that is Marcus.
- The **reviewer** reads everything cold and asks awkward questions. An ops lead, a security-minded engineer from another team, or a founder.
- The **finance contact** owns the numbers in the cost ledger. At a 40-person company this is often whoever approves software spend.

TEMPLATE	LEADS	CONTRIBUTES	SIGNS OFF
01 Architecture contract	Owner	Engineer	Reviewer
02 Permission matrix and risk tiers	Engineer	Owner	Reviewer
03 Approval and dry-run policy	Owner	Engineer	Reviewer
04 Golden task set	Owner	Engineer, support staff	Engineer
05 Eval scoring rubric	Engineer	Owner	Reviewer
06 Failure drill cards	Engineer	Reviewer	Owner
07 Telemetry and retention budget	Engineer	Reviewer	Owner
08 Incident and quarantine runbook	Engineer	Owner	Reviewer
09 ROI ledger	Owner	Finance contact, engineer	Finance contact
10 Readiness checklist	Reviewer	Everyone	Owner

If you are a team of one, add a time gap instead of a second person. Fill a template, wait a day, then review it with template 10 open. Better still, borrow someone for 30 minutes to attack your limits on paper.

The two-week plan

This plan assumes a working prototype at L0 or L1. Budget about half of the engineer's day and an hour or two of the owner's. Work the templates in order, since each one feeds the next.

Week 1: define and constrain

Day 1. Hold a 60-minute kickoff with the owner, engineer and reviewer. Everyone reads chapters 01 and 02 first. The owner drafts the purpose, scope, out-of-scope list and owners in template 01. The engineer fills in the model, provider and tool list.

Day 2. Finish template 01: data classes, readiness levels per action, escalation path, kill switch. The reviewer reads it and leaves questions in the file. Answer them the same day.

Day 3. Template 02. The engineer lists every tool action the code can call, straight from the tool registry, and fills the permission matrix. Score each new or disputed action with the worksheet in `02-tool-risk-tiers.md`. The owner checks the limits column against business rules.

Day 4. Template 03. Decide which actions need approval at each level, which support a dry run, and what the approver sees. Read chapter 04 first.

Day 5. Templates 04 and 05. Pull at least 25 golden tasks from your L0 or L1 logs. Make about a third of them cases where the right answer is to refuse or escalate. Write the scoring rubric, then run the suite once to get a baseline. Expect it to be humbling.

Week 2: prove and decide

Day 6. Template 06. Pick three drill cards that match your riskiest actions. Run one in staging today. Fix what it finds.

Day 7. Template 07. Decide what each trace records, what gets redacted, and how long each data class is kept. Check that your traces answer the questions yesterday's drill raised.

Day 8. Template 08. Write the quarantine and incident runbook. Then flip the kill switch for real during a quiet hour and time how long it takes for the agent to stop acting.

Day 9. Template 09. Fill in the ROI ledger with actual token spend, tool costs, and human review minutes from your logs. If the agent costs more than the work it replaces, you want to learn that now.

Day 10. Template 10. Hold a 60-minute readiness review. Score the 60-point checklist together and decide per action type: go (promote), not yet (hold), or roll back (demote). Write the decision and the date of the next review into template 01.

What done looks like after two weeks

Ten filled templates committed next to the code. A promotion decision per action type, with evidence linked. A kill switch someone other than the engineer has flipped. A date for the next review. "Not yet" is a fine outcome. You now know exactly what evidence is missing.

Before you move on

- ☐ The `templates/` folder is copied into the agent's repo and committed.
- ☐ Each of the four roles has a named person, and the reviewer did not build the agent.
- ☐ You have a current level for every action the agent can take, not one level for the whole agent.
- ☐ The two-week plan is on the calendar, including the day 10 review.
- ☐ Demotion triggers for your riskiest action are written down.

10

Readiness checklist (excerpt)

The first 12 of 60 items. In the kit, every item has the evidence you need to show and the readiness level it blocks.

ID	SECTION	ITEM	SCORE
AC-01	Architecture contract	The architecture contract (template 01) states the agent's purpose in one sentence and lists the actions it must never take.	
AC-02	Architecture contract	Every tool the agent can call is listed in the contract with its inputs, outputs and the system it touches, and the list matches the tool registry in code.	
AC-03	Architecture contract	The model, model version and system prompt version running in production match the versions recorded in the contract.	
AC-04	Architecture contract	The contract states the target readiness level and the explicit limits for it: amount, tools, time window and volume.	
AC-05	Architecture contract	Every hard limit in the contract is enforced in code outside the model, and a test tries to exceed each one.	
AC-06	Architecture contract	Someone other than the author compared the contract to the running code in the last 30 days.	
PC-01	Permissions and credentials	Every tool action has a row in the permission matrix (template 02) with a risk tier and a maximum blast radius.	
PC-02	Permissions and credentials	The agent runs under its own service identity, never a person's account or a shared admin login.	
PC-03	Permissions and credentials	Each credential grants only the scopes in the matrix's credential_scope column, and black-tier actions, such as changing an order address, are in no credential at all.	
PC-04	Permissions and credentials	Credentials for red-tier actions are issued per task or expire within the working session, as the credential_ttl column states.	
PC-05	Permissions and credentials	No credential appears in prompts, source code, tool outputs or the agent's context window.	
PC-06	Permissions and credentials	A person on the on-call list revoked the agent's credentials in a drill within the time the runbook promises.	

Get the full kit

Nine more chapters, 14 editable templates, and the complete Northwind worked example. \$19, instant download.

**[alchemicttechnology.com/guides/
agent-readiness-kit](https://alchemicttechnology.com/guides/agent-readiness-kit)**