

✦ PLAYBOOK · AI AGENTS

Human-in-the-Loop By Design

How to decide which agent actions need a person, build approval queues reviewers actually use, and expand autonomy safely as trust is earned.

FORMAT	LEVEL	READ TIME	UPDATED
Playbook	Intermediate	12 min	Oct 2026

OVERVIEW

What this playbook covers

Agents that act on real systems need a deliberate answer to one question: when does a human decide? Bolting on a generic “approve” button is not that answer. This playbook covers risk-tiering every action, designing approval queues and reviewer screens, escalation paths, sampling for quality assurance, capturing feedback that improves the system, audit trails, and a staged path from full review to earned autonomy.

KEY TAKEAWAYS

- Tier actions by impact and reversibility, not by how confident the model sounds.
- Reviewers need the evidence, the proposed action and a one-click decision on one screen — or approvals become rubber stamps.
- Escalation needs owners, deadlines and a safe default when nobody responds.
- Sample auto-approved work for QA so you can see drift before customers do.
- Expand autonomy one action type at a time, based on measured reviewer agreement, and keep a fast way back.

WHO IT’S FOR

Product owners of agentic features

Engineering leads building agents

Operations managers who will staff review queues

Risk, compliance and audit teams

HOW TO USE IT

Run the risk-tiering exercise in section 01 with operations and risk in the room. Use sections 02–04 to design the queue, and section 07 to agree the criteria for expanding autonomy before launch.

CONTENTS

01 Tier every action by risk
02 Designing the approval queue
03 A reviewer screen that prevents rubber-stamping
04 Escalation and the agent’s own limits
05 Sampling, feedback and audit
06 Expanding autonomy gradually

01 / RISK TIERS

Tier every action by risk

The unit of control is the action, not the agent. List every tool call the agent can make and place each one in a tier based on what happens if it is wrong.

TIER	TYPICAL ACTIONS	CONTROL	REVERSIBLE?
0 · Read	Search, look up records, summarize	Fully automatic, logged	Nothing to reverse
1 · Draft	Write a reply, prepare a form, propose a change	Human sends or applies	Yes — not yet executed
2 · Low-impact write	Tag a ticket, update an internal note	Automatic with QA sampling	Yes, easily
3 · External or financial	Email a customer, issue a refund, change an order	Approval before execution	Partly or costly
4 · Irreversible	Delete data, sign, pay a vendor, change access	Approval by a named role, often two people	No

01

Use thresholds, not just types

The same action can sit in different tiers by amount or audience — a small refund may be tier 2, a large one tier 4.

02

Confidence is a signal, not a tier

Low model confidence can push an action up a tier, but high confidence should never push an irreversible action down.

03

Enforce tiers in code

The tool layer checks the tier and blocks execution without an approval record. The prompt alone cannot be the guard.

04

Default to the higher tier

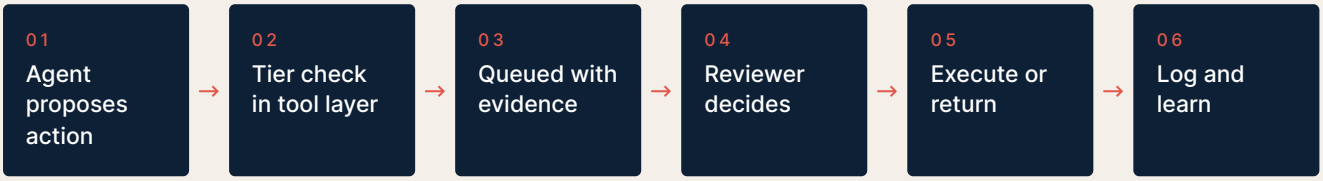
When an action is new or unclassified, treat it as requiring approval until someone decides otherwise.

> Autonomy is not a launch setting. It is something each action earns, one review at a time.
>
> SPLENDENSLABS ENGINEERING PRINCIPLE

✦ 02 / APPROVAL QUEUES

Designing the approval queue

An approval queue is a work system with throughput, deadlines and staffing. Design it like one.



01 Route by skill and authority

Send each item to the group allowed to approve it, using the same roles as your existing approval policy.

02 Set a service level per tier

Agree how long each tier may wait and show the remaining time on every item.

03 Make the pending state safe

While an action waits, the customer or process sees an honest status — never a silent stall.

04 Handle expiry explicitly

If an approval is not given in time, escalate or cancel. Never auto-approve because the timer ran out.

05 Re-validate before execution

Check that the underlying record has not changed between proposal and approval before executing.

✦ 03 / REVIEWER UX

A reviewer screen that prevents rubber-stamping

If reviewing takes longer than doing the task by hand, reviewers will either skip the tool or approve without reading. The screen decides which.

SCREENS THAT GET RUBBER-STAMPED	SCREENS THAT GET REAL REVIEW
Only shows the final action	Shows action, evidence and the agent's reasoning
Evidence is in another system	Highlights what is unusual about this item
Approve is the big default button	Approve, edit and reject are equal weight
No way to edit, only accept or reject	Rejection asks for a short reason

CONTEXT

Show the source

Link each claim in the proposal to the record or document it came from.

CONTEXT

Show the diff

For changes to records, show before and after, not just the new value.

SPEED

Batch the similar

Group near-identical low-risk items so they can be reviewed together.

SPEED

Keyboard first

High-volume reviewers should be able to work without a mouse.

QUALITY

Flag anomalies

Call out amounts, recipients or patterns that differ from the norm.

QUALITY

Watch approval speed

Approvals made faster than the content can be read are a signal to investigate.

✦ 04 / ESCALATION

Escalation and the agent's own limits

Good agents know when to stop. Define the triggers that hand work to a person, and who that person is.

Escalation triggers

- [ ] The request is outside the agent's defined scope
- [ ] Required information is missing or sources conflict
- [ ] The customer asks for a human or expresses distress or anger
- [ ] A policy exception or discretionary judgment is needed
- [ ] The same task has failed or been retried more than an agreed number of times
- [ ] Inputs look like an attempt to manipulate the agent

✦ DESIGN RULE

Escalate with a complete hand-off

The person who receives an escalation should get the conversation, what the agent tried, what it found and why it stopped — so the customer never has to repeat themselves.

✦ 05 / QUALITY

Sampling, feedback and audit

Approval covers what you chose to review. Sampling covers what you did not, and feedback turns every decision into training signal for prompts, rules and evals.

QA sampling plan

STREAM	WHAT TO SAMPLE	WHO REVIEWS	USE OF RESULTS
Auto-executed actions	A random slice of each action type	QA reviewers	Detect drift; decide tier changes
Approved actions	A slice of approvals, re-reviewed	Senior reviewer	Check reviewer consistency
Rejected actions	All, or a large slice	Product and engineering	Find prompt, tool or data gaps
Escalations	A slice per trigger type	Operations lead	Tune triggers and scope

01

Structured rejection reasons

Offer a short list such as wrong data, wrong policy, wrong tone or should not act, plus free text. Structured reasons can be counted.

02

Keep the edited version

When a reviewer edits before approving, store both versions. The diff is the most useful feedback you will get.

03

Feed failures into evals

Every confirmed mistake becomes a regression test the agent must pass before the next release.

04

Close the loop with reviewers

Tell reviewers what changed because of their feedback, or they will stop giving it.

Audit record for every action

- ☐ Who or what requested the action, and when
- ☐ Agent inputs, retrieved evidence and proposed action
- ☐ Prompt, model and tool versions in use

- [ ] Reviewer identity, decision, edits and reason
- [ ] Execution result and any downstream error

✦ 06 / AUTONOMY

Expanding autonomy gradually

Autonomy is granted per action type, on evidence, and can be taken back. Agree the criteria before launch so the decision is not made under pressure.

01 **Shadow mode**
The agent proposes; people do the work as before. Compare the two to establish a baseline.

02 **Review everything**
The agent's proposals go through approval. Track how often reviewers approve unchanged.

03 **Review by exception**
Routine cases within thresholds execute automatically; anything unusual still queues.

04 **Sampled autonomy**
The action type runs automatically with QA sampling and alerts on anomalies.

05 **Roll back on signal**
A rise in sampled errors or complaints moves the action type back a step automatically.

Before promoting an action type

- [ ] Agreed agreement rate between agent and reviewers sustained over an agreed period
- [ ] Enough volume reviewed to cover the common edge cases
- [ ] No unresolved severe errors in that action type
- [ ] Owner of the action type has signed off in writing
- [ ] Rollback switch tested and documented

✦ GET IN TOUCH

Let's Talk

We design and build agents with risk-tiered controls, approval queues and audit trails that operations and compliance teams can sign off.

- ✦ Action inventory and risk-tiering workshop
- ✦ Approval queue and reviewer UX design
- ✦ Audit, sampling and feedback pipelines
- ✦ A staged autonomy plan for your agent

BOOK AN AI STRATEGY SESSION →

EMAIL
hello@splendenslabs.com

WEB
splendenslabs.com

OFFICE
Sona Incubation Foundation, Salem