AI ENABLEMENT KIT · 1 OF 3

Rollout Playbook

How I take an AI tool from working prototype to a team actually using it. Written against two real builds: the n8n support-email triage workflow (Gmail, Slack, Jira, human approval) and the SupportOps MCP server.

PRINCIPLE

A tool nobody uses is a failed project with good screenshots. Rollout is the product. The plan below is how I ran change in a 40-agent contact center, applied to AI tools.

PHASE 1 — PROVE IT WITH ONE WORKFLOW (WEEK 1)

- Pick one painful, repetitive workflow with a visible owner. For triage: routine support emails. Narrow beats broad.
- Run the tool in shadow mode: it drafts, humans still do the work. Compare outputs side by side. No customer sees anything yet.
- Exit gate: the owner says the drafts are usable more than half the time. (Triage V1 test set: ~40% usable drafts on routine emails; that clears the bar for shadow-to-pilot.)

PHASE 2 — PILOT WITH CHAMPIONS (WEEKS 2-3)

- Choose 2-3 champions: respected, curious, not necessarily technical. Skeptics convert when a peer says it works, not when a manager says it will.
- Train by sitting in their real work, not in a slide deck: one 30-minute working session per person, using their actual queue. Then structured practice: five real items a day through the tool.
- Keep the human gate absolute during pilot: every AI action needs an approval (the Slack Approve/Reject card in triage; the tool-permission prompt for MCP access).
- Collect friction daily in one channel. Fix the top complaint each week; announce the fix. Visible responsiveness drives adoption more than features.

PHASE 3 — EXPAND AND MAKE IT STICK (WEEKS 4-6)

- Champions demo to the wider team; each new user gets the same working-session-plus-practice pattern.
- Write the one-page SOP: when to trust the tool, when to override it, who to ask. Put it where work happens (Slack pin, KB article), not in a shared drive graveyard.
- Address the fear directly: the tool removes the repetitive slice so people spend time on judgment calls. Say it, then prove it with the time data.
- Exit gate: adoption metrics (doc 2 of 3) hit their week-6 thresholds, or run a retro on why not before scaling further.

ROLES

Owner (me or the enablement lead): builds, trains, fixes, reports weekly. **Champions**: first users, feedback source, internal marketing. **Manager of the team**: sets the expectation that the tool is part of the workflow, protects practice time. **IT/Security**: reviews the usage policy (doc 3 of 3) before pilot, not after launch.

Adoption Metrics

What I measure to know whether an AI tool is working, and the thresholds where I call it. Instrumentation comes free if you design for it: the triage workflow's audit log (every email: classification, route, approver action, latency) and the MCP server's tool-call log are the data source.

THE FOUR METRICS THAT MATTER

METRIC	DEFINITION	SOURCE	WEEK-6 THRESHOLD
Active use	Share of eligible items going through the tool (emails hitting the triage flow; KB questions asked via chatbot or MCP instead of shoulder taps)	Audit log row count vs total volume	60% of eligible volume
Acceptance rate	Share of AI outputs approved without edits (Slack Approve vs Reject; escalations judged correctly routed)	Approver-action column	50% approved as-is, trending up
Time to done	Median minutes from item arrival to resolved, before vs after	Timestamp deltas in the log	25% faster on routine items
Week-4 retention	Users still putting items through the tool a month after training, without prompting	Distinct users per week in the log	80% of trained users

COUNTER-METRICS (WATCH FOR HARM)

COUNTER-METRIC	WHY	RED LINE
Error escape rate	An approved AI output that later proved wrong. Speed means nothing if quality slips.	Above the human-only baseline = pause and retrain the prompt
Rubber-stamping	Approval decisions faster than a human can read the draft signal the gate is theater.	Median approval under 10 seconds = redesign the review step
Silent abandonment	Usage that spikes at launch and decays says the tool lost to the old habit.	Two straight weeks of declining active use = interview users, fix or kill

CADENCE

Weekly during rollout: one short post in the team channel: the four numbers, one fix shipped, one ask. **Monthly after:** the same numbers to leadership plus hours saved (items through the tool × minutes saved per item, from the time-to-done delta). Hours saved is estimated and labeled as such; the audit log numbers are exact.

THE HONEST RULE

Decide the thresholds before launch, in writing. If the tool misses them, say so and either fix it or kill it. A killed tool with a clear retro builds more trust for the next rollout than a zombie tool nobody admits is dead.

Usage Policy

The rules that make an AI tool safe enough for IT and Security to say yes. This is the working policy behind my two prototypes; adapt the specifics, keep the shape.

1. HUMANS OWN DECISIONS

- No AI output reaches a customer, a ticket queue, or a system of record without a named human approving it. In the triage workflow that is the Slack Approve/Reject card; approval sends, rejection logs a reason.
- The approver is accountable for what they approve. The tool drafts; the person decides. Training covers when to override, not just how to click.
- AI findings are advisory. Escalations created by the tool (the Jira path) are labeled as machine-generated and carry the model output plus a link to the source so a human can check the work.

2. LEAST ACCESS BY DEFAULT

- Tools get the narrowest scope that does the job. The MCP server exposes four tools: three read-only (search, fetch article, fetch spec) and one write that only appends to an escalation log. Anything outside the allowlist is rejected by schema.
- Credentials live in the platform's credential store (n8n credentials, MCP client config), never in code or prompts.
- New scopes require a stated reason and a policy update, not a quiet config change.

3. EVERYTHING IS LOGGED

- Every AI action writes an audit row: timestamp, input reference, model output, route taken, human decision, latency. The triage workflow logs every email plus a separate tab for every outbound side effect; the MCP server logs every tool call with arguments.
- Logs are for accountability and improvement, not surveillance: they answer "what did the tool do and who approved it," and they feed the adoption metrics (doc 2 of 3).
- Retention and access to logs follow the same rules as the underlying data.

4. DATA BOUNDARIES

- Know what leaves the building: classify what the model sees (customer emails, policy text) and confirm the model provider's data handling terms fit before launch.
- No secrets, credentials, or regulated data in prompts. Synthetic or sanitized data for all demos, tests, and training examples.
- Personal identifiers get minimized where the workflow allows it.

5. WHEN THE TOOL IS WRONG

- Anyone can flag a bad output in one step (the same channel used for approvals). Flags get reviewed weekly; repeated misses trigger a prompt or scope fix.
- An approved output that later proves wrong is an error escape: logged, counted, and reviewed against the pre-agreed red line before the tool keeps running.
- There is always an off switch: one person, one action, tool paused, work continues the manual way.