



Rollout Playbook

What this is. The 90-day implementation plan plus the artifacts that keep a rollout alive in a real organization: a pilot charter with contracted success criteria, an exec pitch kit with objection handlers, a curator training session, comms templates, and the catalog of the eight ways rollouts die.

When you need it. The moment this stops being one person's project and needs other people's time, money, or belief.

Which maturity level it serves. The Level 1→3 climb is where this playbook earns its price; its structures keep serving through Level 4.

1. The shape of a rollout that survives

One queue or product domain. Ninety days. Rung 1 of the structure gradient — no enterprise platform, no seven-figure program. Contracted success criteria signed *before* week one, so nobody gets to move the goalposts in either direction. Everything below assumes you've run the Module 1 assessment and are climbing from your true level, not your flattering one.

Weeks 1–2 — Baseline (nothing changes for agents yet)

- Maturity placement signed (Module 1) and Baseline Snapshot captured (Module 6) — the before-picture is unrecoverable later; it comes first.
- Channel inventory with unit costs (Module 6) — finance will ask; have the loaded-cost build-up ready.
- Corpus quick-audit of the pilot domain: run the metadata-extraction prompt (Module 3) over existing articles; expect a fifth (20%) to be retire-candidates.
- **Pilot charter signed** (`pilot-charter.md`) — scope, curators named, success bar, kill criteria.
- Announce to the pilot team (`comms-templates.md`) — the message is "stop writing articles," which is the easiest change-management sell in the industry when you actually mean it.

Weeks 3–6 — Pilot (the loop starts breathing)

- Templates and metadata schema live in the system of record (Module 2).
- Drafting prompts wired to resolved-case exports; **evaluation protocol passed before production** (Module 3 §4 — do not skip; one hour here prevents the "AI writes garbage" verdict that kills programs).
- Gate running: 2–3 curators trained (`curator-training.md`), dispositions logged with reason codes and edit-distance from candidate one (Module 4).
- Lifecycle and ownership policies ratified (Module 5) — an afternoon, not a committee.
- Weekly 15-minute pulse: queue age, acceptance rate, edit-distance. Tune the capture threshold before adding curator hours.

Weeks 7–12 — Prove and expand

- Governed articles compounding; capture-to-publish holding under a day.
- Gap clustering running (Module 3 + Module 6's Gap Log); top gaps firing capture requests — show the loop closing *itself* in the readout; it's the single most persuasive demo the program has.
- Blended cost-per-answer baseline and shift-left ratio on the dashboard; **exec readout delivered** (`exec-pitch-kit.md`), Inversion Brief pre-sent.
- Expansion decision against the charter's criteria: second queue, or fix-and-extend. Resist expanding a pilot that missed its bar — diagnose with `failure-modes.md` first.

Design targets throughout: solver effort per capture ≤ 30 seconds · clean approve ≤ 3 minutes · capture-to-publish ≤ 1 business day · curator SLA 2 business days.

2. Roles (hats, not hires)

- **Sponsor** (support VP/director): signs the charter, receives the dashboard, unblocks. ~1 hr/week.
- **Knowledge lead** (often the person reading this): owns the rollout, the policies, and the workbook. 4–8 hrs/week during the 90 days.
- **Curators** (2–3 senior agents): the gate. Per Module 4's sizing table.
- **Everyone else:** flags KB-worthy cases in ≤ 30 seconds and otherwise does their job. That's the entire ask — say so, loudly.

3. The artifacts in this module

File	Use it when
<code>pilot-charter.md</code>	Week 0 — sign before anything ships
<code>exec-pitch-kit.md</code>	Selling the program up, weeks 0 and 12 (and every budget cycle after)
<code>curator-training.md</code>	Week 3, and for every curator added later
<code>comms-templates.md</code>	Announcing to agents (week 2) and reporting to leadership (monthly)
<code>failure-modes.md</code>	Weekly pulse checks, and any time momentum stalls

4. When to bring in help

Honest criteria, because the playbook is designed for self-serve: get outside help if (a) your Module 1 assessment shows Delivery ≥ 3 while Governance ≤ 1 — you're already answering customers from an ungoverned corpus, and the remediation order matters more than speed; (b) the pilot missed its bar twice with `failure-modes.md` applied — you have an organizational blocker, not a methodology gap; or (c) you're consolidating multiple KBs/languages/brands, where sequencing mistakes are expensive. Otherwise: this playbook, run honestly, is enough.

What this module assumes from its siblings: all of them — it's the glue. Every step above names its module.

Contextkeeping is a KCS-aligned operating model. KCS® is a service mark of the Consortium for Service Innovation; this material is not affiliated with or endorsed by the Consortium.