Measurement Workbook

What this is. The financial and operational instrument for the Governed Knowledge Loop: a 16-tab workbook whose center of gravity is **cost per answer** — the all-channel, issue-centric economics of your knowledge program — plus the KPI catalog, the gate and governance trackers, and a pre-written executive brief for the day your assisted-queue metrics start "worsening" on purpose.

When you need it. Week 1 (the Baseline Snapshot is your before-picture — unrecoverable later), and monthly forever after. Thirty minutes a month keeps it alive; the README tab has the ritual.

Which maturity level it serves. Basic answer economics from Level 3; the full governance telemetry belongs to Level 4. The KPI catalog marks each metric's level so you don't measure ahead of your maturity.

---

1. Why cost per answer leads

Deflection — the classic justification — counts tickets that *didn't* happen. It invites attribution fights, values only the ticket queue, and turns self-service success into a measurement problem. Cost per answer counts what *did* happen: every channel that delivers answers, each at its unit cost, rolled into one blended number that needs no attribution assumption at all.

The model underneath is **shift-left**: answers are delivered across a spectrum — specialist assist (1:1) → self-service (1:many) → community (many:many) → chatbot/AI assist → in-app/predictive → and the sixth channel, your customers' own AI agents consuming your knowledge via API. Each step left multiplies one captured solution's leverage and drops unit cost by one to three orders of magnitude. Knowledge capture is the mechanism that moves answers left; the workbook is how you *see* it moving — blended cost per answer trending down as the shift-left ratio trends up.

Deflection stays, as one clearly-labeled legacy tab, framed as the on-ramp for finance teams that still budget by it. Present its range, then show blended CPA, and let the contrast argue for you.

2. How to adopt it

1. **Day one**: fill the **Baseline Snapshot** — it doubles as the data-request list AnswerEcon would send you, which is not a coincidence.
2. **Week one**: build your **Channel Inventory** — every channel that answers, even at zero volume, even the ones you don't pay for directly. The loaded-cost build-up for the assisted channel is the only hard part; the tab walks it.
3. **Monthly**: update **Monthly Volumes** (the single input grid — everything else computes), log **Edit-Distance** entries and **Gap Log** clusters, refresh **Review Coverage**, send the **Exec Dashboard** per its distribution list.

4. **Before any expansion decision**: run the **Scenario Modeler** — move volume left on paper before you move it in production.

All illustrative data is replaceable and marked; the cell legend (README tab) tells you what to type and what never to touch. The workbook follows financial-model conventions: blue inputs, yellow key assumptions, black formulas, green cross-tab links.

3. The two metrics people get wrong

- **Cost per governed answer** (its own tab) adds curation minutes and AI drafting spend into the unit economics. Run it and you'll find governance costs *pennies* per answer against the dollars of leverage it buys. When someone calls the curation gate "overhead," this is the row you show them.
- **The context metrics** — assisted TTR, first-time-fix, escalation rate — are in the KPI catalog flagged as *expected to move "the wrong way"* as shift-left succeeds: the easy volume leaves the queue, so the queue's averages worsen by construction. The **Inversion Brief** tab is a pre-written one-pager for your executives on exactly this. Send it *before* the trend appears; the program that explains the inversion first survives it.

4. Rebuilding and adapting

The workbook ships ready to use, and its full source (`build_workbook.py`) ships beside it — every formula inspectable, the whole file regenerable. Adapters: map "answers" per channel to your stack's nearest honest proxy (resolved tickets; sessions ending without ticket creation; bot conversations with a resolution signal; API answer calls). Precision matters less than month-over-month consistency — pick a proxy, write it in the cell note, keep it.

What this module assumes from its siblings: reason codes and edit-distance from Module 4's gate; review-date and volatility data from Module 5's lifecycle; articles typed per Module 2. The Gap Log feeds Module 3's synthesis prompts with demand evidence. Nothing here breaks without them — the tabs simply wait for the data.
