

# Forecast, Measure, Optimize: A Workflow-Level Model for the Value and Consumption of Agentic AI

Working paper, version 0.1

Korey Dillon, LatentAxis

September 2026. This version: v0.1, 1 September 2026. License: CC BY 4.0.  
Correspondence via latentaxis.io. DOI: <https://doi.org/10.5281/zenodo.23197852>

## Abstract

Enterprise AI business cases usually produce one number, the value of deploying, and leave the second number, what it will cost to run, to be discovered in the first invoice. This paper presents a workflow-level model that produces both from the same inputs. A workflow is decomposed into steps with measured human effort; each step is assigned a disposition describing what it becomes with AI; each disposition removes a calibrated fraction of effort along a bounded ramp; and each AI-touched step carries a consumption profile priced in the vendor’s billing unit. The model yields month-by-month hours returned, value, run cost with an asymmetric confidence band, net value, and payback, and it is designed to be re-estimated from telemetry once the workflow is live. I describe the model, the reasoning behind its structural choices, a calibration procedure, a regime map showing where run cost is and is not material, a worked example, and a candid separation between what the framework settles and what remains to be learned.

## 1 Introduction

Most organizations approve AI workflow automation on a single number. Someone estimates the hours a workflow consumes today, guesses what fraction an AI system removes, multiplies by a loaded rate, and presents the result as annual savings. The number is not wrong so much as incomplete. It says nothing about the shape of the ramp, nothing about what the system will cost to operate, and nothing about how confident anyone should be in either.

The consequence shows up a quarter or two after launch. Consumption bills arrive that nobody forecast. Adoption lags the assumed instant switch-over. Finance asks for the business case to be redone, and the redo is no better grounded than the original because nothing was instrumented.

The claim of this paper is that the same decomposition that makes value estimable also makes consumption estimable, and that a business case which produces only one of the two is discarding information it already has. I present a model that produces both, with an explicit treatment of ramp and uncertainty, and that is structured so that measured telemetry replaces assumptions step by step after deployment. I call the overall discipline the forecast-measure-optimize loop, or FMO.

The model is deliberately simple. Every quantity in it is something an operations manager already tracks or can measure in a week. That is a design constraint, not a limitation: a model whose inputs cannot be gathered by the people who own the workflow will be filled in with guesses and will not survive contact with the CFO.

## 2 The forecast-measure-optimize loop

Three activities recur in every AI deployment I have been close to, usually performed by three different vendors who each describe their piece as the whole.

**Forecast.** Before anything is built, estimate the value the change creates and the consumption it will incur, over time, with a stated range.

**Measure.** After deployment, instrument the workflow so that hours, volumes, and consumption per step are observed rather than assumed.

**Optimize.** Use the gap between forecast and measurement to reduce consumption (caching, model right-sizing, loop guards, prompt hygiene) and to increase realized value (adoption, disposition changes, step redesign).

Treating these as one loop has a practical consequence: the forecast model must be the same object the measurement fills in. If the forecast is a spreadsheet and the measurement is a dashboard, there is no learning, only two artifacts that disagree. The model below is written so that every parameter has an observable counterpart.

## 3 The model

### 3.1 Decomposition

A workflow  $W$  is an ordered set of steps  $i = 1, \dots, n$ . Each step has a measured human effort  $h_i$  in hours per unit of work, where a unit is whatever the workflow processes: a ticket, a claim, a hire, a dispute. The workflow processes  $U$  units per month and the organization values an hour of the relevant labor at a loaded rate  $\rho$ .

Total human effort per unit today is

$$H_0 = \sum_{i=1}^n h_i.$$

Decision points and branches are recorded for the diagram and for the export, but they do not change the arithmetic: a branch that is taken for a fraction of units simply has its hours per unit already averaged over that fraction when  $h_i$  is measured. This is a simplification and I return to it in Section 8.

### 3.2 Disposition

Each step is assigned exactly one disposition  $d_i$  from a small set describing what the step becomes when AI is applied:

| Disposition | Meaning                                                         | Factor $\delta$ |
|-------------|-----------------------------------------------------------------|-----------------|
| Automate    | An agent performs the step end to end; humans handle exceptions | high, near 0.85 |
| Assist      | A human performs the step with a copilot                        | mid, near 0.45  |

| Disposition | Meaning                                                                    | Factor $\delta$ |
|-------------|----------------------------------------------------------------------------|-----------------|
| Human       | A human performs and decides;<br>AI supplies a brief                       | low, near 0.15  |
| Eliminate   | The step is removed because<br>upstream automation makes it<br>unnecessary | 1.0             |

The factor  $\delta(d)$  is the fraction of the step’s human effort removed once the change is fully mature. The values shown are illustrative starting points, not findings; Section 7 addresses their calibration.

Two remarks on the structure. First, the disposition set is small on purpose. Finer gradations invite false precision, and in practice the argument in the room is always about which steps are *human* and which are *automate*; the *assist* bucket absorbs the rest. Second, choosing dispositions is the substantive expertise in the entire exercise. Which steps regulators allow you to automate, which need a named signature, and which are actually two steps in disguise is knowledge that comes from having run the workflow, and no model supplies it.

### 3.3 Ramp

No change arrives on day one. Each step carries a maturity profile: a start month  $s_i$ , a ramp length  $k_i$  in months, and a curve shape. The realized fraction of the full effect in month  $m$  (zero-indexed) is

$$r_i(m) = c\left(\frac{m - s_i + 1}{k_i}\right), \quad m \geq s_i,$$

and  $r_i(m) = 0$  for  $m < s_i$ , where  $c(\cdot)$  is a curve function clamped to  $[0, 1]$ . Three shapes are sufficient in practice:

- **Linear:**  $c(t) = t$ .
- **Step:**  $c(t) = 1$  if  $t \geq 1$ , else 0. Appropriate for a step that is switched off, i.e. eliminated.
- **S-curve:** a logistic normalized so that  $c(0) = 0$  and  $c(1) = 1$ :

$$c(t) = \frac{L(t) - L(0)}{L(1) - L(0)}, \quad L(t) = \frac{1}{1 + e^{-10(t-0.5)}}.$$

The S-curve is the default. Adoption of a new tool by a team is slow at first, accelerates as the early users become references, and saturates. The steepness constant (10) is a choice that makes the curve reach roughly 95% of its effect at  $t = 0.8$ ; it is not calibrated and any value between 6 and 12 produces indistinguishable business-case results.

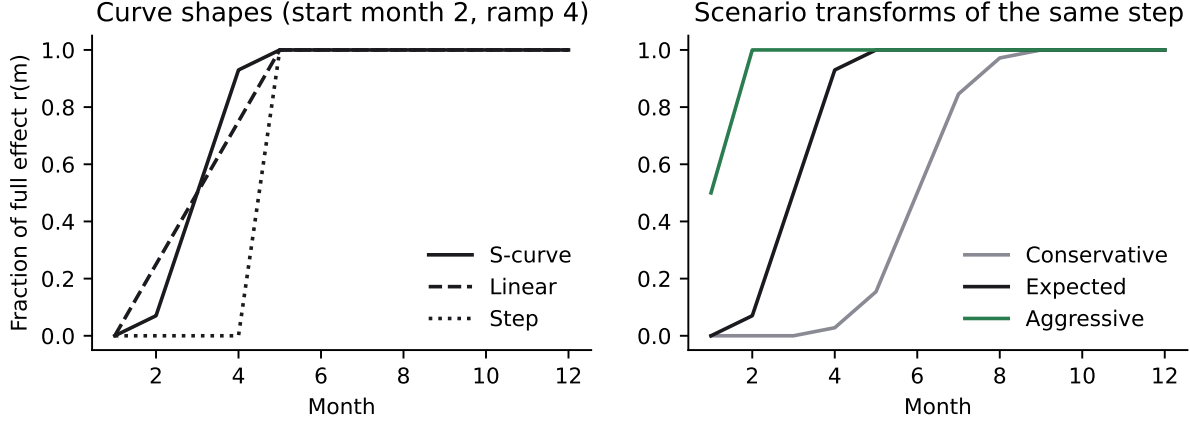


Figure 1: Ramp curve shapes and scenario transforms.

### 3.3.1 Scenarios

Rather than asking a stakeholder to set start and ramp for every step three times, the model applies a workflow-level scenario transform to every step's profile: a shift  $\sigma$  added to each start month and a multiplier  $\kappa$  applied to each ramp length.

| Scenario     | $\sigma$ | $\kappa$ |
|--------------|----------|----------|
| Conservative | +2       | 1.5      |
| Expected     | 0        | 1.0      |
| Aggressive   | -1       | 0.6      |

The effective profile is  $s'_i = \max(0, s_i + \sigma)$  and  $k'_i = \max(1, \text{round}(k_i \kappa))$ . This keeps the per-step judgment (which steps are early, which are late) intact while letting the audience see the whole case under three rollout postures.

## 3.4 Value

Hours removed per unit in month  $m$ :

$$\Delta h(m) = \sum_i h_i \delta(d_i) r_i(m).$$

Hours returned, FTE equivalent (with  $F$  productive hours per FTE-month, typically 160), and gross value:

$$\text{Hours}(m) = U \Delta h(m), \quad \text{FTE}(m) = \frac{\text{Hours}(m)}{F}, \quad V(m) = \rho U \Delta h(m).$$

The value line is called *returned hours*, not *savings*, deliberately. Whether returned hours become productive capacity, headcount reduction, or absorbed slack is a management decision the model cannot make. Section 8 treats this as an open parameter.

### 3.5 Consumption

Each step with disposition *automate* or *assist* carries a consumption profile: runs per unit  $g_i$  (how many times an AI system is invoked per unit of work), billable units per run  $q_i$ , and a price function. The billable unit is deliberately abstract. Token-billed vendors have a price per million input and output tokens with an input share  $\alpha_i$ ; other vendors bill per assist, per credit, or per seat. The price of  $x$  units for step  $i$  is

$$p_i(x) = \begin{cases} \frac{x}{10^6} (\alpha_i \pi^{\text{in}} + (1 - \alpha_i) \pi^{\text{out}}) & \text{token billing} \\ x \pi^{\text{unit}} & \text{otherwise.} \end{cases}$$

Run cost in month  $m$ , at the point estimate:

$$C_{50}(m) = \sum_i p_i(U g_i q_i r_i(m)).$$

The ramp appears in the cost line as well as the value line. This is the single most common omission in business cases I have reviewed: value is ramped and cost is assumed at full volume, or the reverse.

#### 3.5.1 Asymmetric confidence band

Per-run consumption in agentic systems is right-skewed. Loops, retries, tool-call chains, and long contexts push consumption above the typical run; nothing pushes it below by a comparable amount. Figure 2 shows the shape from a lognormal illustration; observed distributions from production systems are typically at least this skewed.

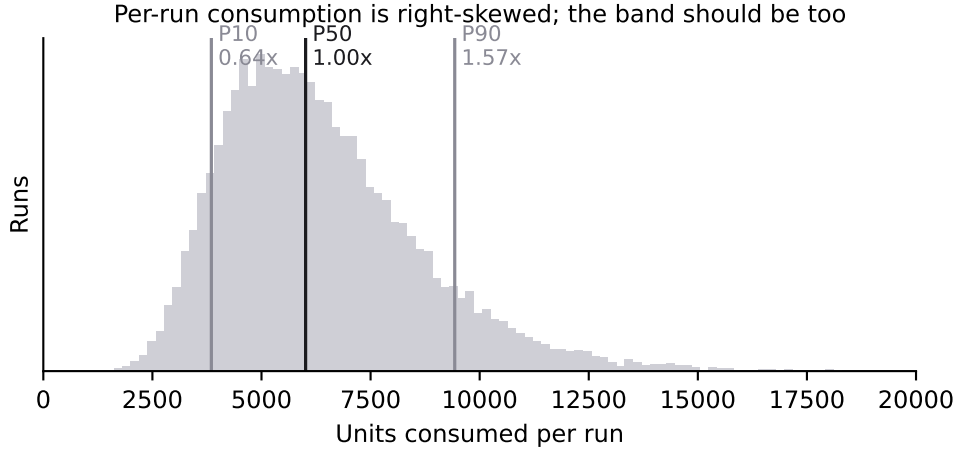


Figure 2: Per-run consumption is right-skewed, so the band around the point estimate should be too.

The model therefore reports a low and high run cost by scaling units per run:

$$C_{\text{low}}(m) = \sum_i p_i(U g_i q_i r_i(m) \beta_{\text{low}}), \quad C_{\text{high}}(m) = \sum_i p_i(U g_i q_i r_i(m) \beta_{\text{high}}),$$

with default multipliers  $\beta_{\text{low}} = 0.75$  and  $\beta_{\text{high}} = 1.6$ . A symmetric band of plus or minus 25% is, in my experience, a reliable sign that the estimate was made by someone who has not operated these systems. The multipliers are placeholders until telemetry provides P10 and P90 of units per run for the actual step, at which point  $\beta_{\text{low}} = P_{10}/P_{50}$  and  $\beta_{\text{high}} = P_{90}/P_{50}$ .

### 3.6 Net, cumulative, payback

$$N(m) = V(m) - C_{50}(m), \quad N_{\text{low}}(m) = V(m) - C_{\text{high}}(m), \quad N_{\text{high}}(m) = V(m) - C_{\text{low}}(m).$$

Cumulative net over the horizon  $M$  (12 months by default) and the payback month:

$$\bar{N} = \sum_{m=0}^{M-1} N(m), \quad m^* = \min\{m : \sum_{j \leq m} N(j) > 0\}.$$

Steady state is defined as the last month of the horizon, which is correct when every ramp completes within the horizon and is reported as such when it does not.

### 3.7 Invariants

Any implementation of this model should satisfy the following, and I recommend they be encoded as tests rather than assumed:

1. Steps with disposition *human* or *eliminate* contribute zero run cost.
2. If every step starts at month one or later, month zero equals the current state exactly.
3. If every ramp completes within the horizon, the final month equals the closed-form full-maturity value  $\sum_i h_i \delta(d_i)$ .
4.  $C_{\text{low}} \leq C_{50} \leq C_{\text{high}}$  for every step and month.
5. Removing a step never increases hours returned.
6. Conservative  $\leq$  Expected  $\leq$  Aggressive on cumulative hours returned.
7. Payback is null, not zero, when cumulative net never crosses zero.

## 4 Where run cost matters: a regime map

A useful thing the model makes visible is that run cost is second-order for some workflows and first-order for others, and the boundary is predictable. At full maturity, the ratio of value to run cost for an automated step is

$$R = \frac{h \delta \rho}{g p(q)}.$$

Figure 3 plots  $R$  across human hours per unit and tokens per run at frontier list pricing, a \$60 loaded rate, one run per unit, and  $\delta = 0.85$ .

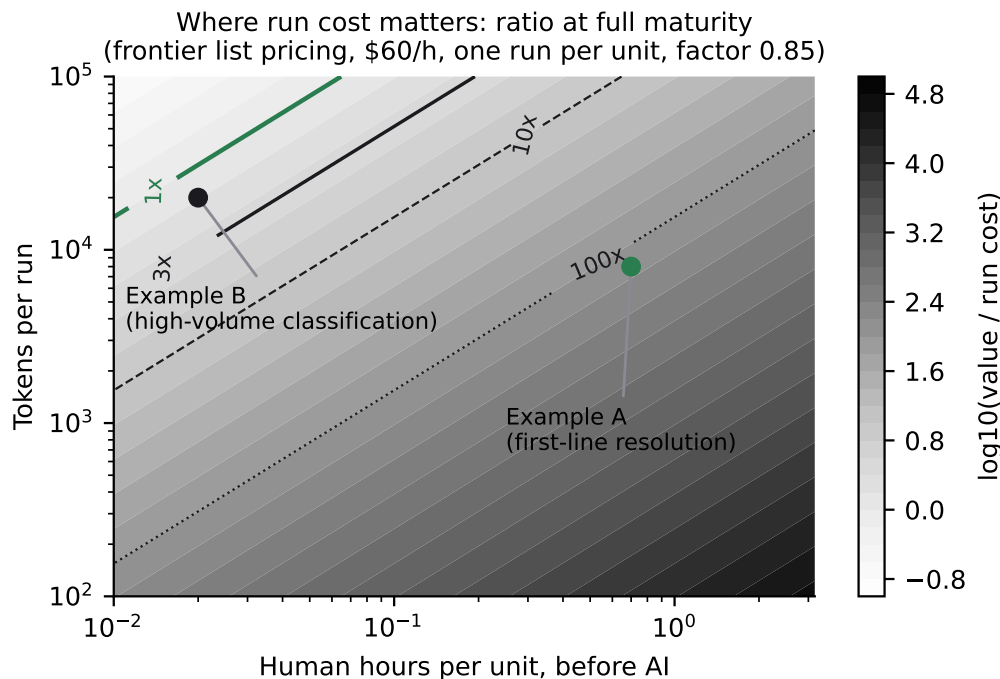


Figure 3: Value-to-run-cost ratio at full maturity across the space of workflows.

Two regimes emerge.

**Labor-heavy workflows** (right side of the map) sit at ratios above 100x. A step that took a person 40 minutes and now consumes eight thousand tokens returns two orders of magnitude more than it costs. For these, the consumption band matters for budget forecastability, not for the go/no-go decision. The decisive risks are adoption and disposition, not run cost.

**High-volume, low-effort workflows** (upper left) sit near or below 3x. Classifying half a million short items a month that each took a person a minute, using a frontier model with a large context, returns roughly 1.5 dollars per dollar of run cost. Here the consumption band is the business case. Model tier, context length, and runs per unit are the levers, and a 1.6x overrun turns a positive case negative.

The 3x contour is a working threshold, not a law. Below it, I have found that unmodeled costs (review burden, error handling, platform fees) reliably consume the remaining margin.

## 5 Worked example

**Example A: a support request workflow.** Six steps, 4,000 requests per month, \$60 loaded rate, mid-tier token pricing at \$3 and \$15 per million. Dispositions: intake, triage, and closure automated; first-line resolution and specialist investigation assisted; approval kept human.

| Step   | $h_i$ | $d_i$    | $s_i$ | $k_i$ | $q_i$ (tokens) |
|--------|-------|----------|-------|-------|----------------|
| Intake | 0.30  | Automate | 1     | 3     | 3,000          |

| Step                     | $h_i$ | $d_i$    | $s_i$ | $k_i$ | $q_i$ (tokens) |
|--------------------------|-------|----------|-------|-------|----------------|
| Triage                   | 0.20  | Automate | 1     | 3     | 2,500          |
| First-line resolution    | 0.70  | Assist   | 0     | 2     | 8,000          |
| Specialist investigation | 1.40  | Assist   | 2     | 4     | 15,000         |
| Approval                 | 0.30  | Human    | 0     | 1     | 0              |
| Closure and notes        | 0.30  | Automate | 1     | 2     | 4,000          |

$H_0 = 3.2$  hours per unit. At steady state,  $\Delta h = 1.67$  hours per unit, hours returned 6,680 per month (about 42 FTE), value 401K per month, run cost 858 per month with a band of 644 to 1,373. Cumulative net over twelve months: 3.9M expected, 2.9M conservative, 4.5M aggressive. Payback in month one, because assisted steps start immediately.

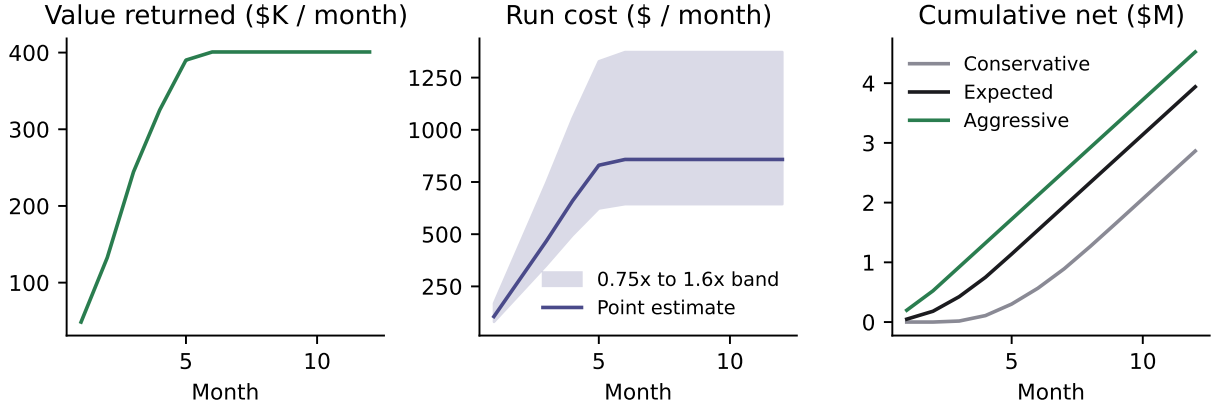


Figure 4: Example A under the model: value, run cost with band, and cumulative net by scenario.

The run cost is a rounding error against the value, which is the point of showing it: for this class of workflow, the band is a budgeting tool and the decision turns entirely on whether the disposition choices survive contact with the compliance team and whether the assisted steps actually get adopted.

**Example B: high-volume classification.** 500,000 items per month, 0.02 hours per item today, automated with a frontier model at 20,000 tokens per run. Value 510K per month, run cost 330K per month, ratio 1.55. The high band takes run cost to 528K, above the value. This workflow is only viable if the model tier or context can be cut, and the model tells you that before anything is built.

## 6 Measurement and calibration

The forecast model has seven kinds of parameter. Each has an observable counterpart, listed here in the order I would instrument them.

| Parameter | Forecast source               | Measured counterpart                                 |
|-----------|-------------------------------|------------------------------------------------------|
| $h_i$     | Time study, ticket timestamps | Same, post-deployment, on the human-handled residual |

| Parameter     | Forecast source                 | Measured counterpart                                  |
|---------------|---------------------------------|-------------------------------------------------------|
| $U$           | Volume reports                  | Same; watch for induced volume (Section 8)            |
| $\delta(d_i)$ | Prior by disposition and domain | Realized $\Delta h_i/h_i$ at maturity                 |
| $s_i, k_i$    | Rollout plan                    | Fitted from the adoption curve of the step            |
| $g_i, q_i$    | Design estimate                 | Telemetry: invocations per unit, units per invocation |
| $\beta$       | Default 0.75 / 1.6              | P10/P50 and P90/P50 of $q_i$ from telemetry           |
| $\pi$         | Vendor list price               | Invoiced effective price                              |

A minimal telemetry design attributes every model invocation to a step identifier and a unit identifier. With that, runs per unit, units per run, and the full per-run distribution fall out directly, and the band becomes empirical within the first month.

For the disposition factor, the update I use is a shrinkage toward the prior:

$$\hat{\delta} = w \delta_{\text{observed}} + (1 - w) \delta_{\text{prior}}, \quad w = \frac{n}{n + n_0},$$

where  $n$  is the number of units observed at maturity and  $n_0$  is a prior weight (I use 500). This avoids over-fitting to the first noisy weeks while letting a step with real volume overwrite its prior quickly.

The optimize phase then acts on the measured parameters with known levers. Caching and prompt hygiene reduce  $q_i$ . Model right-sizing reduces  $\pi$ . Loop guards truncate the right tail and narrow  $\beta_{\text{high}}$ . Step redesign changes  $d_i$ . Each intervention has a predicted effect in the model, which makes it testable.

## 7 What is settled and what is still being learned

I want to be explicit about which parts of this framework I am confident in and which are open.

### 7.1 Settled, in the sense that I would defend them to a skeptical CFO

- **Decomposition to steps with measured hours** is the right unit of analysis. Whole-workflow percentages hide the disposition argument, and that argument is the entire substance.
- **A small disposition set with per-disposition factors.** Four classes are enough; more invites false precision.
- **Ramp on both the value and the cost line.** The asymmetry of ramping one and not the other is the most common error in the business cases I review.
- **Asymmetric consumption bands.** The right tail is real, and any model that reports a symmetric band is not describing agentic workloads.
- **Billing in abstract units** so that the model is vendor-neutral before vendor selection.
- **The regime map.** Whether run cost matters is predictable from hours per unit and units per run, and this should be the first thing a team looks at.

- **Invariants as tests.** A business-case engine without tests will silently drift as the spreadsheet is edited.

## 7.2 Still being learned

- **Disposition factor values.** The illustrative 0.85 / 0.45 / 0.15 are priors. They almost certainly vary by domain, by step type, and by how the assisted tool is integrated. Publishing calibrated values by domain is the most valuable thing this line of work could produce and I do not yet have enough post-deployment data to do it responsibly.
- **Ramp shapes and durations.** The S-curve default is a prior from technology adoption generally. Whether individual step ramps are actually sigmoidal, and what typical  $k_i$  values are for assisted versus automated steps, is an empirical question I am collecting data on.
- **Band widths.** 0.75x and 1.6x are placeholders. Early telemetry suggests the high multiplier is, if anything, conservative for multi-step agents.
- **The capacity conversion coefficient.** The model reports returned hours at face value. In reality some fraction converts to productive capacity, some to headcount change, and some evaporates. A coefficient  $\gamma \in [0, 1]$  multiplying  $V(m)$  is the honest fix, and I do not have a defensible prior for it beyond “less than one.”
- **Induced volume.** Making a workflow cheaper often increases its use. If  $U$  rises after deployment, both value and cost rise, and the ratio  $R$  is unchanged but the budget is not. The model treats  $U$  as fixed; it should not.
- **Step interactions.** Effort removed at one step can change effort at the next (better triage shortens investigation). The model is additive. A first-order fix is to allow  $h_i$  to depend on upstream dispositions; I have not needed it yet but expect to.
- **Quality and error cost.** An automated step that is wrong some fraction of the time creates rework downstream. The model has no term for this. For regulated workflows this may be the dominant unmodeled cost.
- **Review burden on human-kept steps.** Steps kept human with an AI brief sometimes get slower, not faster, when the brief must be verified. The 0.15 factor may need to be allowed to go negative.
- **Price drift.** List prices for tokens have fallen steadily. A forecast at today’s prices will overstate run cost over a multi-year horizon; a forecast at extrapolated prices invites optimism. I currently forecast at list and note the direction of the error.
- **Attribution in telemetry.** Assigning invocations to steps and units is straightforward for an agent that owns a step and messy for a copilot a human uses across steps. This affects how quickly the assisted factors can be measured.

## 8 Limitations

This is a planning model, not a simulation. It does not model queues, staffing constraints, or the second-order organizational effects of removing work. It treats months as the unit of time, which is coarse for a fast rollout and fine for a business case. It ignores fixed costs of building and platform licensing, which belong in the same case but in a separate line, and it says nothing about the value of speed (shorter cycle time) as distinct from the value of effort removed. All of these are additive extensions; none of them, in my view, changes the core structure.

## 9 Conclusion

The two numbers a business case needs, value and run cost, come from the same decomposition. Producing only one of them is a choice to discard information. A model that carries a disposition per step, a ramp on both lines, and an honest band on consumption is small enough to explain in a meeting and rigorous enough to survive a finance review. Its parameters all have measured counterparts, which is what turns a forecast into a learning loop rather than a document that ages. The regime map gives a first-pass answer to whether consumption is even the question. Everything beyond that is calibration, and calibration is the work I expect to be publishing next.

## 10 Appendix: notation

| Symbol                                    | Meaning                                                     |
|-------------------------------------------|-------------------------------------------------------------|
| $h_i$                                     | Human hours per unit at step $i$ , today                    |
| $H_0$                                     | Total human hours per unit, today                           |
| $U$                                       | Units processed per month                                   |
| $\rho$                                    | Loaded hourly rate                                          |
| $F$                                       | Productive hours per FTE-month                              |
| $d_i, \delta(d_i)$                        | Disposition of step $i$ and its effort-removal factor       |
| $s_i, k_i$                                | Start month and ramp length of step $i$                     |
| $r_i(m)$                                  | Realized fraction of full effect in month $m$               |
| $\sigma, \kappa$                          | Scenario shift and ramp multiplier                          |
| $g_i, q_i$                                | Runs per unit and billable units per run                    |
| $\alpha_i$                                | Input share of tokens                                       |
| $\pi$                                     | Price per billable unit (or per million tokens, in and out) |
| $\beta_{\text{low}}, \beta_{\text{high}}$ | Band multipliers on units per run                           |
| $V, C, N$                                 | Value, run cost, net per month                              |
| $\bar{N}, m^*$                            | Cumulative net over horizon; payback month                  |
| $R$                                       | Value-to-run-cost ratio at full maturity                    |
| $\gamma$                                  | Capacity conversion coefficient (open)                      |