— PORTFOLIO CASE STUDY

SCM

Make vs. Buy Decision Framework

Strategic sourcing decision support for aerospace and defense
Kenyon Woodley | Western Washington University | 2026 | Stack: Python + Streamlit + Claude API

01 — PROBLEM

Make vs. buy gets decided on a spreadsheet that optimizes unit cost. The expensive mistakes happen in the factors the spreadsheet leaves out.

The default make-vs-buy analysis is a cost comparison: model the in-house cost, get a supplier quote, pick the cheaper one. It is fast, it is defensible in a budget meeting, and it is wrong often enough to be dangerous. The decisions that come back to hurt a program are rarely the ones where the cheaper option was also the safe one. They are the ones where cost pointed one way and something the spreadsheet never captured pointed the other.

A part that is cheaper to buy but carries proprietary design IP. A flight-critical component with one qualified supplier in the market. An ITAR-controlled item where an unrestricted outsource is not viable without a qualified, compliant sourcing path. A part that looks attractive to insource until you notice the program is being retired in three years and the tooling will never amortize. None of these show up in a unit-cost delta, and all of them change the right answer.

> A make-vs-buy decision that ignores IP control, criticality, single-source exposure, and export status is not a sourcing decision. It is an accounting exercise.

02 — INSIGHT

Cost is one input on one axis. The decision lives in two dimensions, and the interesting cases are off the diagonal.

The core architectural decision was to refuse to collapse the decision into a single number. The engine scores a part on two independent layers: an Economic layer that captures whether making is cheaper and whether you have the capacity and volume to justify it, and a Strategic layer that captures criticality, IP control, supply-market risk, export status, and where the program sits in its lifecycle. Each layer is scored 0 to 100, where higher favors Make and 50 is neutral.

Plotting the two scores against each other produces a 2×2 quadrant. The two diagonal cases are the easy ones: both layers favor Make, or both favor Buy. The value of the tool is in the two off-diagonal cases, where the layers disagree. A part can be more expensive to make and still be the right thing to make. A part can be cheaper to make and still be the wrong thing to commit to. Those are the decisions a cost-only model gets backwards.

The signature output is "Strategic Make": a part the economics say to buy, but criticality, IP, single-source exposure, or export control say to keep in-house. It is the recommendation a spreadsheet would never produce, and it is the one most worth getting right.

03 — ENGINE ARCHITECTURE

A deterministic two-layer engine that places a part on a quadrant, then resolves it with explicit overrides

The engine is built in Python with strict separation between the deterministic core and the explanatory layer. Scoring, quadrant assignment, mode logic, and overrides are fully deterministic and reproducible: the same inputs always produce the same recommendation. Claude generates the plain-language rationale from the computed result. It never generates or modifies a score or a recommendation.

TWO-LAYER BLEND

Blended = (Economic Score × Economic Weight) + (Strategic Score × Strategic Weight)

Both layers scored 0–100, higher favors Make, 50 neutral.
Economic layer: make-vs-buy cost (45%), capital/NRE burden (25%), capacity (15%), volume stability (15%)
Strategic layer: criticality (23%), IP control (20%), supply-market risk (20%), regulatory/export (16%), roadmap (11%), lifecycle (10%)

Quadrant assignment uses the two layer scores directly. The blended score drives mode-based tipping on close calls only.
NRE amortized over 5 years to match defense program horizons. All weights and lookup tables centralized as named constants — single source of truth, rendered live in the "How scoring works" panel.

The recommendation is derived from the quadrant, then modified by an explicit precedence chain: scoring mode can tip genuine close calls, a sunsetting-program override enforces capital discipline, and an ITAR override sits at the top as a hard constraint. Every override surfaces as a visible flag. Nothing is folded silently into a number.

04 — SCORING LAYERS

Ten factors across two layers, each tied to a specific input and a fixed scoring function

Economic Layer: does making this part make financial sense right now?

E1
MAKE VS BUY COST
Loaded make cost (including amortized NRE) against the buy quote, as a bounded ratio so the score normalizes

E2
CAPITAL / NRE BURDEN
Amortized tooling and qualification investment as a share of annual buy spend. Heavier burden favors Buy.

E3
INTERNAL CAPACITY
Idle / Available / Tight / Over-capacity. Idle capacity lowers the marginal cost of

E4
VOLUME STABILITY
High and stable volume justifies absorbing fixed make costs. Low or erratic

across part values rather than swinging on raw dollar magnitude. Highest economic weight at 45%.

Rests at neutral when no investment is required, so a zero-NRE part carries no false bias toward Make.

making and favors Make; over-capacity favors Buy.

volume favors Buy to avoid fixed commitment.

Strategic Layer: should this part be controlled in-house regardless of cost?

S1

PART CRITICALITY

Flight-critical / Mission-critical / Important / Non-critical. Flight-critical work favors in-house control of quality, traceability, and configuration. Highest strategic weight at 23%.

S2

IP / DESIGN CONTROL

Core proprietary / Sensitive / Standard / Commodity. Differentiating IP favors keeping production in-house to protect it; commodity parts favor Buy.

S3

SUPPLY MARKET RISK

Single-source / Limited / Competitive / Commodity. A fragile or concentrated market favors insourcing to de-risk; a deep competitive market favors Buy.

S4

REGULATORY / EXPORT

ITAR-controlled / Export-sensitive / Regulated-process / Unrestricted. Controlled work favors domestic make. ITAR also triggers a hard override (Section 06).

S5

TECHNOLOGY ROADMAP

Strategic-to-own / Useful / Neutral / Divesting. A capability worth owning for the future favors Make; a declining or non-core technology favors Buy.

S6

PROGRAM LIFECYCLE

Growing / Stable / Declining / Sunsetting. A growing program justifies building capability; a sunsetting program argues against capital investment and can trigger a hard override.

05 — QUADRANT & RECOMMENDATIONS

The two layer scores place the part on a quadrant. The off-diagonal cases are where the tool earns its keep.

ECONOMIC LAYER — FAVORS MAKE →

The two diagonal quadrants are agreement cases. The two off-diagonal quadrants are where the layers disagree and where a cost-only analysis fails. When both layers land inside a neutral band (45 to 55 on each axis), the tool returns **Marginal — Gather More Data** rather than forcing a call it cannot defend. That is the human-in-the-loop principle made literal: the tool declines to fake precision on a genuine coin flip.

STRATEGIC MAKE

low economic · high strategic

More expensive to make, but criticality, IP, single-source exposure, or export control justify keeping it in-house. The signature output.

MAKE

high economic · high strategic

Both layers favor insourcing. A clean make case.

BUY / QUALIFY SECOND SOURCE

low economic · hedge on risk

Buy, but single-source exposure on a critical part means hedging the buy by qualifying a second source rather than relying on one supplier.

OPPORTUNISTIC MAKE / DUAL-SOURCE

high economic · low strategic

Cheaper to make and capacity allows it, but with no strategic lock-in. Make while favorable, treat it as reversible, and hold dual-source as a hedge if capacity tightens.

MARGINAL — GATHER MORE DATA

both layers inside the 45–55 neutral band

Not a quadrant outcome but a deliberate refusal to decide. When neither layer points clearly, the tool declines to fake precision on a genuine coin flip and asks for more data instead. The human-in-the-loop principle made literal.

06 — OVERRIDES & PRECEDENCE

Two cases where a score is not enough, resolved by explicit rules in a fixed order

Some factors do not belong on a weighted axis because they are not tradeoffs. They are constraints. The engine handles these as overrides that fire after the quadrant assignment, in a fixed precedence order, each surfacing as a visible flag rather than being absorbed into a number.

LIFECYCLE — OVERRIDE

Sunsetting program with unfavorable make economics is forced off Make. Capital should not fund tooling for a program being retired, regardless of how strategically attractive the part looks. The override reads the full economic layer score, so it inherently weighs tooling, unit-cost delta, and volume together rather than a single arbitrary threshold. It lands on Buy, or Buy / Qualify Second Source when single-source risk exists, except in a commodity market where a second source is automatic.

ITAR — TOP PRECEDENCE

An ITAR-controlled part never returns an unrestricted Buy. It is floored to a domestic, qualified source. This override outranks the lifecycle override: a compliance constraint beats an economic-rationality rule every time. ITAR status also suppresses cost-driven mode tipping, because export control is precisely the reason a part is held in-house and a cost posture should not override it.

07 — SCORING MODE

A decision lens that tips genuine close calls without overriding clear-cut ones

Three scoring modes re-weight the blend between the two layers, matching how a sourcing organization actually frames a decision cycle: are we under cost pressure, or are we prioritizing supply security? The mode does not change the individual factor scores, and it cannot move a decisive case. It only tips a recommendation when the case is genuinely close: when the mode-weighted blend falls within a defined margin of neutral and the balanced read was already near the line.

LAYER BLEND BY MODE

Balanced 50/50 · Cost-Driven 70/30 · Strategic-Driven 30/70

Balanced gives the structural read.
Cost-Driven can step a borderline Strategic Make down to a hedged Buy (never an unrestricted one, and never for an ITAR part).
Strategic-Driven can step a borderline Opportunistic Make down to Buy when there is no strategic lock-in to justify tying up capacity.
Decisive cases and the Marginal band never flex with mode.

This was a deliberate correction during development. An earlier version computed a blended score that moved with the mode but never actually changed the recommendation, which made the control misleading. The fix tied mode to the recommendation in close cases only, so the lens means something without letting a posture override a clear or compliance-bound decision.

08 — WORKED EXAMPLES

Two off-diagonal cases that a unit-cost comparison gets backwards

PART	WHAT THE COST SAYS	WHAT THE TOOL RECOMMENDS, AND WHY
Titanium structural fitting (flight-critical, ITAR, single-source, growing program)	Buying is roughly $134K/year cheaper all-in. A cost model says outsource.	**Strategic Make.** Flight-criticality, ITAR control, and a single-source market justify the cost premium and rule out an unrestricted buy. The economics lean Buy at 31/100; strategy leans Make at 83/100. This is the case a spreadsheet gets wrong.
Sunsetting actuator housing (flight-critical, single-source, program winding down)	Same high-risk criticality profile as the fitting, but on a winding-down program. A strategic checklist says make it.	**Buy / Qualify Second Source.** The program is being retired and the economics do not favor making, so the lifecycle override forbids sinking tooling capital into a dying program. But single-source exposure on a flight-critical part still needs hedging, so the buy is paired with qualifying a second source. The case a strategic checklist gets wrong.

The two examples are deliberate mirror images. The first shows strategy correctly overriding cost. The second shows lifecycle discipline correctly overriding strategy. Neither answer falls out of a single number, and that is the entire point of the tool.

09 — AI NARRATIVE LAYER

Deterministic first. AI explains the result — it does not generate it.

The AI layer translates a computed decision into a short, leadership-ready rationale. Every score, the quadrant, the recommendation, and any override are computed before the model is called. The prompt passes the final recommendation as a fixed fact and instructs the model that implying a different one is a defect. The engine recommendation is also displayed directly above the narrative, so the deterministic result and its explanation can never be confused.

If no API key is present, the layer falls back to a deterministic template, so the live tool never breaks on the AI path. The narrative is honest about mode-tipped outcomes: when a part lands on Buy despite favorable economics, the explanation says so rather than claiming both layers favored buying.

WHAT AI GENERATES

A two-to-four sentence rationale naming the dominant drivers, the cost picture, and any active override. Plain, leadership-ready language with no hedging on compliance constraints.

WHAT AI NEVER TOUCHES

Layer scores, quadrant assignment, recommendation selection, override logic, break-even math. All computed deterministically before the call. The model string and token limit are fixed in code: the AI layer is boxed, not open-ended.

10 — TECHNICAL DESIGN

A deterministic core, an explanatory shell, and 121 passing tests

MODULE STRUCTURE

decision_engine.py → ai_narrative.py → streamlit_app.py

```
decision_engine.py: all constants, lookup tables, validated input schema, two-layer scoring, quadrant logic, mode tipping, overrides, break-even, sensitivity — single source of truth
ai_narrative.py: Anthropic SDK call, prompt construction, deterministic fallback
streamlit_app.py: full UI layer, quadrant chart, comparison table, live "how scoring works" reference
```

The test suite enforces 121 assertions including: cost-delta direction and break-even math, every category-to-score lookup, layer-weight integrity, mode-blend correctness, quadrant assignment across all four regions plus the neutral band, the sunsetting and ITAR overrides and their precedence, mode-driven tipping on close calls, input validation bounds, determinism (identical inputs yield identical output), and a regression lock asserting each of the six sample parts produces its intended recommendation.

The test suite earned its place during development: the regression lock caught a capital-burden factor that was defaulting to a Make-favoring value when no investment was required, biasing the engine toward Make on zero-NRE parts. The fix reset it to neutral. The bug was invisible without the tests.

11 — LIMITATIONS & SCOPE

What this tool does not do — and why those are explicit decisions

KNOWN **It does not compute the make cost.** The loaded make unit cost is an input, not an output. Computing it is the job of the Should-Cost Model upstream in the portfolio. This tool consumes a make cost and a buy quote and asks the higher-order question: should we make it at all.

KNOWN **It does not evaluate specific suppliers.** "Buy" is treated as a sourcing strategy, not a vendor. Supply-market risk is about the structure of the market, not the performance of one supplier. Evaluating a named supplier is the job of the Supplier Scorecard, a separate tool in the portfolio.

KNOWN **Strategic factors are categorical, not continuous.** Criticality, IP, market structure, and lifecycle are entered as named tiers, not free numeric values. This keeps the tool usable in a short working session and the scoring fully reproducible, at the cost of fine-grained nuance within a tier.

KNOWN **ITAR is modeled as domestic-only.** The tool treats ITAR control as requiring a domestic qualified source. Licensed international production of controlled articles exists in reality but is deliberately out of scope: modeling it would add licensing nuance with no portfolio value.

KNOWN **No export, no persistence.** The live tool is the artifact. The session comparison table accumulates up to six parts and resets on refresh. There is no saved state and no document export, consistent with the rest of the portfolio.

12 — BROADER CONTEXT

The decision layer that sits above should-cost and beside supplier scoring

The Make vs. Buy Decision Framework is the highest-altitude tool in the portfolio. It operates at the level a sourcing leader works at: not "what should this part cost" or "how good is this supplier," but "should we make this at all, given that cost is only one of the factors that matter." It consumes a make-cost figure that the Should-Cost Model can produce, and it treats the buy path as a strategy that the Supplier Scorecard can then evaluate vendor by vendor. Together the tools form a buy-side decision system from part prioritization through award.

SUITE PROGRESS

→ Part Prioritization Framework: Complete

→ RFQ Lifecycle Tracker: Complete

→ Parametric Should-Cost Model: Complete

→ Supplier Performance & Risk Scorecard: Complete

→ Make vs. Buy Decision Framework: Complete

DESIGN PHILOSOPHY

Deterministic scoring. AI explains outputs, never generates them. Cost is one weighted input, never the answer. Constraints like export control are hard overrides, not weighted variables. The tool produces the recommendation and shows its work; the sourcing professional makes the call, and the tool is honest enough to say when the call is too close to make.

13 — CLOSING NOTE

Built to close the gap between "the supplier quote is lower, so we outsource" and "we have a structured, defensible view of whether a part belongs in-house across cost, control, risk, and program life." Deterministic by design. Explainable by requirement. Aerospace-specific by intent. **The cheapest option is**

often the wrong one, and the most strategic option is not always worth the capital. The tool surfaces both. The sourcing professional decides.

Python engine + Streamlit UI + Claude API. | Two-layer scoring: 4 economic factors, 6 strategic factors. | 2x2 decision quadrant with off-diagonal Strategic Make and Opportunistic Make outputs. | Three scoring modes that tip genuine close calls. | Sunsetting and ITAR overrides with fixed precedence. | Break-even volume and sensitivity flag. | Marginal band for genuine coin-flips. | Six engineered sample scenarios. | 121 passing tests with a sample-recommendation regression lock.