DECISION INTELLIGENCE SUITE

Project Development Report

A Personal Prototype in Decision Robustness and Scenario Analysis

Prepared by

Shubham Gambhire

Prototype status. Not affiliated with, and containing no data from, the originating case competition sponsor.

Contents

1. Executive Summary
2. Background and Origin
3. Objectives
4. Scope and Design Philosophy
5. Decision Modeling Methodology
6. System Architecture and Technology Stack
7. Feature Overview
8. Design and Visual Identity Evolution
9. Naming Evolution
10. Issue Log and Resolutions
11. Known Platform Limitations
12. Testing and Validation Approach
13. Data Privacy and Fictional Data Note
14. Known Limitations and Future Work
15. Conclusion

1. Executive Summary

Decision Intelligence Suite is a self-contained, browser-based decision analysis tool that tests whether a mathematically optimal choice between two competing options remains the right choice once uncertainty, hard constraints, and factors that cannot reasonably be priced in currency are taken into account. It was built as an interactive prototype following an observation made during a live case competition: an optimization model recommended one course of action, but that recommendation was silent on several considerations - such as retailer trust and already placed pre-orders - that a real decision maker would not be able to ignore.

The tool does not run its own optimizer. It accepts the output of an existing model - a value for each option under each scenario - and layers a transparent, editable robustness analysis on top of that output: expected value, scenario win rate (with correct handling of ties), worst-case value, worst-case regret, a continuous capacity-allocation control backed by a real formula, and a qualitative factor register that distinguishes

between factors that should disqualify an option outright, factors that should nudge a recommendation without overriding it, and factors that exist purely to inform the reader's own judgment.

Functional work includes full decision-state persistence via localStorage, correct win-rate handling under ties, Reversibility wired into soft-factor scoring, input validation and edge-case guards, pure metrics functions, and JSON export/import. The Overview is structured around a clear primary recommendation block, a stronger Position Line, and comparison cards with explicit recommended-state treatment. This document is written as a project record rather than promotional material.

2. Background and Origin

The idea behind this tool originates from a supply-chain case competition, Maestros 2026, Supply Chain Track, in which the presenting team built a mixed-integer linear program to decide between two competing uses of limited festive-season production capacity: protecting committed volume for long-standing wholesale and retail partners, or redirecting capacity toward a large, unplanned surge in demand. Under the case's own numbers, the optimization model recommended the surge option by a wide margin.

During the development of that case solution, a specific limitation became apparent. The optimization model correctly captured revenue, cost, and capacity, but it had no mechanism to represent retailer relationship risk, already placed pre-orders, or the strategic importance of protecting a flagship selling season. The team's own presentation materials acknowledged this directly, stating plainly that the model could not calculate loss of goodwill or loss of trust from retailers and the resulting channel erosion.

That acknowledged gap prompted a broader question outside the scope of the competition itself: what would it take to build a tool whose entire purpose is testing whether a financially optimal recommendation is still the right one once those unquantified factors are taken seriously, without simply inventing a currency value for them. This project is the resulting personal prototype. It was built independently of the competition and its sponsor, using the case's numeric structure for realism while fictionalizing every company name, division, and scenario involved.

3. Objectives

- Demonstrate, using a real optimization result as a starting point, that a financially optimal decision and a robust decision are not always the same decision.
- Build a decision-support layer that consumes an optimizer's output rather than replacing it, keeping the boundary between what the model says and what a person should do visible rather than blended into a single figure.
- Give qualitative, non-financial considerations a defined and limited role in the recommendation - one that can be inspected, challenged, and adjusted - rather than either ignoring them entirely or silently converting them into a currency value.
- Produce a portfolio piece with its own distinct visual identity, in-house methodology documentation, working persistence (including full-state export/import), and a recommendation surface that a decision maker can act on in seconds.

4. Scope and Design Philosophy

4.1 Why the Tool Does Not Optimize

A natural temptation when building a tool of this kind is to have it run its own mixed-integer program. That was deliberately avoided. The Scenario Log tab is an input surface: a table of values, one row per scenario and one column per option, that is assumed to already be the output of an optimization exercise performed

elsewhere. Every other tab recalculates live from that table. This keeps the tool honest about what it actually contributes, which is robustness analysis and structured judgment capture, not optimization.

4.2 Why Qualitative Factors Are Typed, Not Priced

Rather than asking the user to estimate a currency value for something like retailer trust, the tool asks for three things instead: how much does this matter (Impact), how sure is this really true (Confidence), and how easily could this be reversed later (Reversibility). All three now scale Soft Preference influence. It also asks the user to classify each factor into exactly one of three roles.

- Hard Constraint. Can disqualify an option outright when marked active. This is a categorical veto rather than a valuation, and is implemented as a boolean override rather than a number added into a sum.
- Soft Preference. Nudges the recommendation by a bounded amount, scaled by Impact × Confidence × Reversibility, and capped so that no combination of soft preferences can outweigh the underlying financial numbers on its own.
- Contextual. Displayed for the reader's judgment only. Never enters any calculation.

This distinction, and its limits, are explained inside the tool's own Methodology tab, including a dedicated section that directly addresses the objection that assigning any number to a qualitative factor is itself a form of quantification. The tool does not dispute that objection. It argues instead that the numbers involved are ordinal rather than cardinal, structurally capped, never merged into the same figure as the financial output, and, in the case of hard constraints, a veto rather than a price.

4.3 Two Numbers, Never One

The tool always shows two separate figures rather than blending them into one score: the Financial Optimum, which is the pure expected value across scenarios and is unaffected by any qualitative input, and the Robustness-Adjusted Position, which blends expected value, worst-case performance, and qualitative nudging under two user-controlled settings. Keeping these visibly apart, rather than merging them into a single number, is treated as the tool's actual audit trail. The Overview now leads with an explicit primary recommendation statement so the gap between the two numbers is immediately actionable.

5. Decision Modeling Methodology

5.1 Core Metrics

For each option, the following are computed directly and only from the Scenario Log.

- Expected Value. The probability-weighted average of that option's value across all scenarios.
- Scenario Win Rate. The share of total scenario probability in which that option scores highest. When two options tie for best in a scenario, the probability mass is split equally so that win rates never sum to more than 100%.
- Worst-Case Value. The option's lowest score across all scenarios (a maximin measure).
- Worst-Case Regret. The largest gap, in any single scenario, between what the option scored and what the best available option would have scored in that same scenario (a minimax-regret measure).

Worst-case value and worst-case regret are deliberately kept as two separate figures rather than collapsed into one, because they can and do disagree with each other in the tool's own example data. This is a known property of decision theory - specifically the difference between Wald's maximin criterion and Savage's minimax-regret criterion - rather than a defect in the tool, and the Tripwires tab explains the disagreement directly using the seeded data as a worked example.

5.2 Blending Financial and Qualitative Signals

The Robustness-Adjusted Position is computed as a weighted blend of the normalized difference in expected value and the normalized difference in worst-case value between the two options, controlled by a single Risk Posture setting (0 = trust the financial optimum fully; 100 = trust the safer worst-case outcome fully). A second setting, Strategic Weight, controls how much influence Soft Preference factors are allowed to add on top of that blend, bounded to a fixed maximum share of the total so that qualitative input can never fully override the underlying financial numbers. Soft Preference influence is now scaled by Impact × Confidence × Reversibility (Irreversible = 1.0, Partially reversible = 0.7, Reversible = 0.4).

If any active Hard Constraint disqualifies an option, that constraint overrides the blended position entirely for that option, and the interface displays this state as an explicit veto rather than as an extreme value on the same numeric scale used everywhere else. When both options are disqualified, the Overview shows a distinct “No viable option” state and the Position Line does not claim a recommended side.

5.3 Fragility Heuristic

Fragility (LOW / MEDIUM / HIGH) is no longer driven by win probability alone. It incorporates: (a) whether the financial leader is disqualified, (b) the financial leader’s scenario win rate, (c) the size of the expected-value gap relative to average EV, and (d) how close the robustness-adjusted position is to the 50% midpoint. This produces a more diagnostic signal for how brittle the current recommendation is.

5.4 The Continuous and Discrete Tripwires

One relationship in the underlying case data was expressed as a genuine mathematical formula linking a single capacity-allocation decision to a resulting value. That relationship is preserved as a live, draggable control - the only continuous tripwire in the tool - with a real formula behind it derived directly from the case’s own objective function and verified to reproduce both of its named endpoints exactly. Every other scenario in the Scenario Log is a discrete, independently specified data point rather than a point on that same continuous line, and the tool is explicit about this distinction rather than implying a precision that does not exist.

6. System Architecture and Technology Stack

The application is distributed as a single self-contained HTML file with no external backend, consistent with the rest of this portfolio. React and ReactDOM are loaded from pinned CDN builds; JSX is transformed in-browser with a pinned Babel standalone build. Styling uses a minimal self-contained utility CSS set (no Tailwind CDN) plus the project’s design tokens.

Layer	Approach
UI framework	React 18 (functional components and hooks), loaded via UMD CDN
Styling	Self-contained utility CSS (Tailwind-compatible class names) plus hand-written design tokens: bracket-frame panels, bracket text stamps, horizontal ticker readouts
Iconography	Inline SVG icons matching the Lucide set, selected per section for semantic fit
Typography	Three-typeface system: serif display face for the brand headline, geometric sans for labels and body, monospace for data and figures
Persistence	Standard localStorage with versioned full-state payload (scenarios, factors, settings, options, capacity allocation). JSON export and import for platform-independent portability
Metrics engine	Pure functions (computeMetrics, computeDisqualified, computeDerived) extracted outside the React component tree for testability and freedom from side effects
Animation	CSS keyframe and React key-based remount techniques for save confirmation;

	lightweight flash effect on numeric displays when underlying values change

7. Feature Overview

7.1 Overview

The landing tab. Leads with a Primary Recommendation block: the recommended option name, a one-sentence rationale, and four compact key numbers (financial optimum, expected advantage, robustness lean, win rate of the financial leader). Below that sits the Position Line (financial optimum vs robustness-adjusted position with clearer markers and gap readout), side-by-side option comparison cards with a RECOMMENDED badge and a consistent 2×2 metric grid, and a Key Factors list with a link to the full Qualitative Factors register.

7.2 Scenario Log

An editable table representing the optimizer’s output: one row per scenario, a probability (clamped 0-100 with a non-blocking warning when the sum is outside 99-101), a short description of the triggering event, and a value for each option. Fully persisted as part of the decision package. Every other tab recalculates from this table. Destructive deletes require confirmation.

7.3 Qualitative Factors

A register of judgment-based considerations, each typed as a Hard Constraint, a Soft Preference, or Contextual information, with Impact, Confidence, and Reversibility recorded for every entry. Soft Preference influence is scaled by all three. Includes a save control that persists the full decision state without forcing navigation away from the tab.

7.4 Tripwires

Shows the core robustness metrics side by side, the one continuous capacity-allocation control described in Section 5.4, and a set of discrete indicators showing which scenarios, if they occurred, would flip the recommendation from one option to the other.

7.5 Settings

Houses the Risk Posture and Strategic Weight controls, the Core Service Floor constraint that caps the capacity-allocation control, and an editor that allows both compared options to be renamed throughout the entire tool. Also provides full-state JSON Export and Import (file-based, validated on import). Includes a save control covering all settings. Full Reset requires confirmation.

7.6 Methodology

A plain-language explanation of what the tool does, what is genuinely computed versus what is illustrative, what the tool deliberately does not do, a direct response to the objection that scoring a qualitative factor is itself a form of quantification, and a provenance note describing which figures in the tool are real and which are fictionalized. Updated to reflect persistence behavior, tie-splitting, and reversibility weighting.

8. Design and Visual Identity Evolution

The visual identity of this tool went through several distinct, deliberate iterations rather than settling on a first pass, and each iteration was driven by direct feedback rather than by the author’s own preference alone.

Initial direction. The first build used a dark, angular interface described internally as an Ops Board: bracket-cornered panels in place of rounded cards, a horizontal ticker of key figures in place of a grid of stat cards, and bracket-style text stamps in place of pill-shaped badges. This was a deliberate attempt to distinguish the tool from the dark, nautical control-tower aesthetic already used by a sibling project in the same portfolio.

First correction. Despite the new component language, the tool still read as visually similar to the sibling project on inspection, because the underlying structural patterns - rounded bordered panels, a dot-and-border status badge, and a blurred glass header - had been reused wholesale and only recolored. The interface was rebuilt using the angular bracket-frame system, replacing every one of those reused patterns rather than simply adjusting their color.

Second correction. Even after the structural rebuild, the color palette itself had not changed, and the tool still read as a variation on the same dark interface family. Three genuinely different color directions were proposed; the light palette, internally named Field Drab, was selected and implemented.

Accessibility correction. The Field Drab palette's muted secondary text colors were later found to fall below standard contrast guidelines. The palette was revised to verified contrast ratios, and the number of distinct accent colors was reduced from five closely related warm hues to four clearly distinguishable ones (rust, amber, slate, moss).

Terminology correction. Several tab and section names (Sitrep, Field Intel, Command Settings, Field Manual) were identified as needlessly military in register. Most labels were renamed to plain business language. The single exception was Tripwires, which was judged to be genuinely clearer than any plain alternative and was kept.

Full rebrand. Following a supplied design reference describing a specific palette and typographic direction under the name Decision Intelligence Suite, the tool adopted that exact color palette, replaced its display typeface with a serif face reserved for the single true headline moment, replaced its label typeface with a cleaner geometric sans, and introduced a custom rounded logo mark distinct from the angular bracket-frame language used everywhere else.

Overview UX redesign. A critical review identified that the Overview optimized for visual identity more than decision speed. The tab was restructured around a Primary Recommendation block (option name + one-sentence rationale + four key numbers), a stronger Position Line with higher-contrast markers and clearer legend, option comparison cards with an explicit RECOMMENDED badge and consistent 2×2 metric grid, and reduced visual competition between Fragility and the recommendation itself.

9. Naming Evolution

The product name changed several times over the course of development, each time for a specific, stated reason rather than as an aesthetic preference alone.

Name	Reason for the change
Decision Robustness Engine	Working name carried over from the original scoping discussion. Descriptive but generic.
Decision Intelligence Sitrep	Adopted to fit a military situation-report theme. Later identified as jargon-heavy and difficult to parse for a non-specialist reader.
Decision Robustness	A plain, interim simplification after the jargon in the previous name was flagged as a readability problem.
Decision Intelligence Statement (DIS)	Adopted to preserve a three-letter acronym pattern while sounding more like a formal deliverable.
Decision Intelligence Suite	Final name. Judged to sound less like an early draft, and accurately describes the

	tool as several connected modules rather than a single document.

10. Issue Log and Resolutions

The table below is a complete record of the functional and usability issues identified during development - most through direct testing and user feedback - and how each one was resolved. Issues are listed in the approximate order they were discovered.

#	Issue	Root Cause	Resolution
1	Interface read as visually derivative of a sibling portfolio project.	Component structures reused wholesale and only recolored.	Rebuilt with angular bracket-frame system, ticker readout, and bracket stamps.
2	After structural rebuild, interface still read as unchanged.	Dark color palette itself was untouched.	Moved to light, high-contrast Field Drab palette.
3	Secondary text strained readability; palette felt unpolished.	Muted text failed contrast guidelines.	Recalculated palette to verified contrast; reduced accents to four clear hues.
4	Tab names difficult for non-specialist audience.	Military situation-report terminology.	Renamed to plain business language; kept Tripwires only.
5	Both-options-disqualified state reused single-option message.	Single code path for disqualification banners.	Separate, visually distinct banner for dual disqualification.
6	Capacity slider could exceed active constraint limit.	Slider max bound to fixed upper limit.	Bound slider range and display to constraint-adjusted limit.
7	Empty Scenario Log produced unbounded worst-case values.	No floor for empty scenario list.	Guard defaults worst-case to zero; empty-state messages added.
8	Confidence field appeared to have no effect.	Only Impact was used in soft-factor scoring.	Confidence now scales Soft Preference influence alongside Impact.
9	Confidence fix silently changed default recommendation.	Default factor had low Confidence.	Recalibrated default Confidence so original figures preserved.
10	Edits immediately after open were occasionally lost.	Async load could overwrite in-progress edits.	User-edited guard skips applying loaded data after first edit.
11	Save confirmation animation became unnoticeable.	Animation and tab navigation fired in same render.	Delayed navigation; timestamp-based animation key.
12	Two divergent save-button implementations.	Built independently per tab.	Consolidated into one shared SaveButton component.
13	Position markers too close; overridden state unclear.	Limited vertical separation; color-only signal.	Increased separation, gap readout, explicit VETO label.
14	Options could not be renamed.	Names hardcoded as constants.	Editable application state with Settings editor.
15	Save failed on mobile client.	Hosting platform storage limited to certain tiers/clients.	Documented platform limitation; export/import fallback scoped.
16	Scenario Log not persisted across reloads.	Only factors, Qs, and settings were saved.	Full decision state (including scenarios) now persisted via localStorage.
17	Non-standard window.storage API.	Custom host API; failed outside host environment.	Replaced with standard localStorage and versioned JSON payload.
18	Win rates could sum above 100% under ties.	Tied options each received full scenario probability.	Probability mass split equally among tied options.
19	Reversibility field was dead code.	Stored and displayed but never used in scoring.	Wired into soft-factor scoring (1.0 / 0.7 / 0.4 multipliers).
20	No input validation on	Raw Number() with no clamps	Clamps, safeNum guards, and non-blocking

	probabilities or numeric fields.	or warnings.	probability-sum warning.
21	Fragility based only on win probability.	Single-threshold heuristic.	Now incorporates disqualification, EV gap ratio, and distance to 50%.
22	Save forced navigation to Overview.	Tab switch bundled with save confirmation.	Save stays on current tab; animation retained.
23	No platform-independent portability.	Storage depended on host environment.	Implemented full-state JSON export and validated import.
24	Destructive actions had no confirmation.	Delete and Reset executed immediately.	window.confirm before scenario/factor delete and full Reset.
25	Metrics and derived state mixed into App component.	Harder to test and reason about.	Extracted pure computeMetrics, computeDisqualified, computeDerived.
26	Overview lacked a decisive primary recommendation.	User had to assemble the answer from multiple equal-weight panels.	Primary Recommendation block with name, rationale, and key numbers.
27	Position Line markers weak; legend unclear.	Low-contrast geometry; cramped vertical separation.	Higher-contrast markers, clearer gap callout, improved legend.
28	Option cards lacked recommended-state treatment.	Both cards looked equal regardless of recommendation.	RECOMMENDED badge, accent border, consistent 2×2 metric grid.

11. Known Platform Limitations

Earlier revisions depended on a host-provided browser-scoped storage API that was limited to published artifacts on certain subscription tiers and clients. That dependency has been removed. The application now uses standard localStorage for full-state persistence and provides JSON export/import so a decision package can be moved between browsers, machines, or offline contexts without relying on any host-specific capability.

The remaining distribution constraint is that the single-file HTML build uses in-browser Babel transformation. This produces a console advisory recommending precompilation for production. For a portfolio prototype the trade-off (zero build step, open-and-run) is intentional; a precompiled bundle would be the natural next step if the tool were productized.

12. Testing and Validation Approach

- Static syntax verification. Revisions were checked in-browser and against the rendered UI after each functional and visual change.
- Manual calculation audits. Core figures (expected value, win rates under ties, robustness-adjusted position, fragility bands) were recalculated by hand from the underlying formulas and cross-checked against rendered values.
- Reference sweeps after each revision. Source searched for leftover references to old storage APIs, old variable names, or incomplete guards.
- Direct user testing. The majority of issues in Section 10 - especially the Overview hierarchy and Position Line clarity problems - were identified through direct use of the working tool rather than code review alone.
- Edge-case exercise. Empty scenario lists, dual disqualification, zero core commitment, and probability sums far from 100% were exercised to confirm defined UI states.

13. Data Privacy and Fictional Data Note

- The company name, division name, and every named scenario used as example data in this prototype are invented. None correspond to any real organization.
- The two base-case financial figures and the capacity-allocation formula preserve the numeric structure of the originating case competition exercise, retained for realism, but are not attributed to any real company and do not reproduce any confidential materials from that competition.
- The application has no backend server. Saved data lives in the browser's localStorage and is not transmitted to any third party by the application itself. Export produces a local JSON file under the user's control.

14. Known Limitations and Future Work

Gap	Why It Was Left Out	Possible Next Step
Limited to exactly two compared options	Position Line and blending formula are built around a two-way comparison. Generalizing would change the core visual metaphor.	A separate multi-option mode with a different visual treatment, built as an additional view.
In-browser Babel transformation	Zero-build distribution preferred for a portfolio prototype.	Precompiled production bundle if the tool is productized.
No automated test suite	Validation relied on manual audits and direct use; appropriate to single-file prototype scale.	Unit tests around the pure metrics engine (now extracted and side-effect free).
Only one continuous, formula-backed relationship	Only the capacity-allocation relationship in the source case was expressed as a genuine formula.	None planned. The tool is intentionally explicit about this distinction.
No multi-user or account support	Out of scope for a personal portfolio prototype with no backend.	Would require a hosted backend and authentication layer.

15. Conclusion

Decision Intelligence Suite demonstrates a specific, defensible position: that a financially optimal recommendation and a robust one are not automatically the same thing, and that the gap between them can be made visible, measured, and discussed without either ignoring qualitative judgment or dressing it up as false precision.

Functionally, the tool persists the full decision package, handles ties and edge cases correctly, gives Reversibility a real role, validates inputs, and can be exported and imported as ordinary JSON. Presentationally, the Overview leads with a primary recommendation, a clearer Position Line, and comparison cards that show which option the system actually endorses.

The project also continues to serve as a working record of iterative development in the ordinary sense - including visual identity decisions that did not work on the first attempt, a naming process that took several passes to settle, and a series of concrete functional and UX defects that were found, diagnosed, and fixed rather than glossed over. That combination - a stated methodology the tool is willing to defend and qualify in its own words, together with an honest account of what went wrong along the way and how it was corrected - remains the intended contribution of this document and of the prototype it describes.