DECISIONFLOW

AI Engineering & Evaluation Report

Implementation, guardrails and benchmark plan

Document code	DEV-02
Version	2.0
Status	Final Public Portfolio Edition — Implementation Baseline
Baseline date	14 July 2026
Author	Min Thu Kyaw
Role	AI Product Manager

Evidence transparency: Actual implementation evidence is separated from targets, proposed release actions, and the illustrative DEV-06 continuity bridge. Later commercial outcomes are not presented as actual results.

Document Control

Purpose	Document the Develop phase for AI Engineering & Evaluation Report.
Primary audience	Hiring managers, product leaders, engineering stakeholders, and pilot reviewers.
Evidence classification	Verified implementation fact / implementation-derived assessment / target / proposed next action.
Source limitation	Implementation status is grounded in the authored implementation baseline and public repository references supplied for the case study.
Change control	Publication rule: retain evidence labels unless content is replaced by documented real-world observations.

Evidence Links

- Low-Fidelity Prototype
- High-Fidelity Prototype
- MVP Frontend Repository
- MVP Backend Repository

Contents

AI Product Objective

DecisionFlow uses generative AI to convert unstructured notes or transcripts into a structured decision model and to produce an explainable recommendation. The product principle is explicit: AI assists; the user decides. AI output must be reviewable, editable, attributable, versioned and accepted by a human before it becomes a final decision record.

Evidence status — Structured extraction, recommendation generation, JSON validation and recommendation history are implementation facts from the code-derived baseline. Precision, recall, hallucination and user-trust values remain targets until a benchmark is executed.

Current AI Pipeline

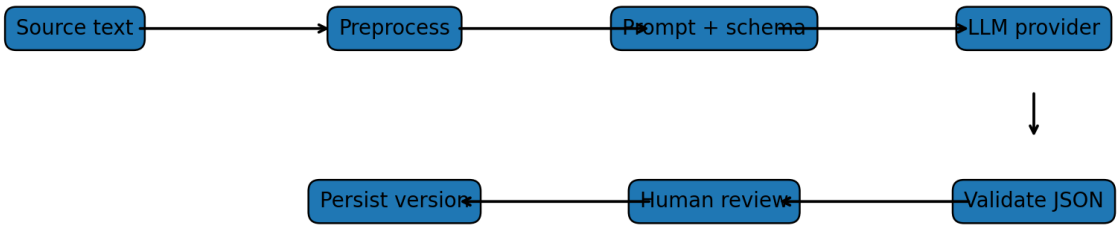

Figure 1. Current and target AI processing pipeline.

Stage	Current behavior	Release requirement
Source intake	Paste or UTF-8 TXT.	Explain privacy, size and provider processing.
Preprocessing	Cleaning, limits and validation.	Version preprocessing rules and test edge cases.
Prompt construction	System instruction, task, context and JSON schema.	Store prompt version and exact model configuration.
Model call	OpenRouter-compatible chat completion.	Fixed pilot model, budget and provider error taxonomy.
Schema validation	Pydantic validates structured response.	Reject invalid content consistently and log safe metadata.
Human review	Editable fields in UI.	Persist reviewed version and make it recommendation input.
Recommendation	Reasoning, confidence, risks, alternatives and next steps.	Expose provenance and immutable versions.

Model and Provider Strategy

- Use a provider abstraction so model changes do not require product-logic rewrites.

- Fix one model/configuration for the pilot to make evaluation and regression results comparable.
- Store provider, model, prompt version, generation time, token usage and fallback state.
- Route to smaller or self-hosted models only after quality/cost evidence exists.

Prompt and Output Contract

Contract element	Requirement
Instruction boundary	Treat transcript content as untrusted data and ignore embedded role-changing instructions.
Output mode	Return a JSON object only; prose must not bypass schema validation.
Extraction fields	Topic, options, criteria, stakeholders, risks and action items.
Recommendation fields	Recommended option/path, confidence, rationale, risks, alternatives and next steps.
Limits	No more than 20 cleaned list items and 500 characters per item in the current target contract.
Versioning	Prompt, schema and preprocessing versions must be saved with every generation.

Human Review and Provenance

1. Present AI extraction as a draft, not a fact.

2. Allow edits to every structured field.

3. Persist the reviewed representation separately from the original model output.

4. Use the reviewed representation for recommendation generation.

5. Record acceptance of a specific recommendation version.

6. Persist the final selected option even when it differs from AI advice.

Trust requirement — Confidence alone is not explainability. Users need source evidence, model/fallback provenance, editable fields and an explicit distinction between AI advice and the human decision.

Fallback Behavior

Condition	Current/target behavior	Required UX
No API key	Heuristic fallback may be produced.	Prominent “fallback” label; do not imply model reasoning.
Invalid JSON/schema	Selected failures can fall back.	Show reason category and require full review.
Invalid key/model	Provider error.	Actionable configuration message.
Quota exhausted	Provider error.	Explain temporary unavailability and avoid duplicate charges.
Timeout/unavailable	Retry policy then fail safely.	Allow retry without losing the decision state.
No viable options	Conservative output.	Warn that recommendation quality is limited.

Evaluation Dataset

Create a versioned benchmark set before pilot. The set should contain synthetic or approved, de-identified examples and manually annotated ground truth. A useful initial target is 80–120 source items across the following strata:

Stratum	Coverage
Well-structured decision notes	Clear topic, options and criteria.
Messy meeting transcript	Multiple speakers, irrelevant discussion and implicit references.
Sparse notes	Missing criteria or options.
Conflicting evidence	Participants disagree or reverse direction.
Prompt injection	Transcript attempts to override instructions or extract secrets.
Sensitive content	Names, customer references and confidential context.
No-decision content	The correct result is no valid decision.
Multilingual content	Only for languages declared in pilot scope.

Metrics and Rubrics

Metric	Definition	Pilot target	Status
Topic correctness	Human-rated match to source intent.	≥ 90% acceptable	Target
Option precision	Extracted options that are supported by source.	≥ 90%	Target
Option recall	Ground-truth options captured.	≥ 85%	Target
Criteria field accuracy	Correct names, descriptions and weights.	≥ 85%	Target
Unsupported-claim rate	Generated claims without source support.	≤ 5%	Target
Recommendation groundedness	Reasoning uses reviewed options, criteria and notes.	≥ 4.2/5 human rating	Target
Edit rate by field	Share of model fields modified by users.	Baseline in pilot	Measure
Fallback rate	Generations completed with heuristic fallback.	< 5% under normal config	Target
p95 generation latency	End-to-end AI request latency.	< 35 seconds	Target
Cost per finalized decision	AI usage divided by completed decisions.	Budget-defined	Target

Human Evaluation Rubric

Dimension	1	3	5
Groundedness	Mostly unsupported.	Mixed support.	Every material claim traceable to reviewed data.
Completeness	Misses core options/criteria.	Captures main structure with gaps.	Captures all decision-critical information.

Dimension	1	3	5
Clarity	Ambiguous or verbose.	Understandable.	Concise and decision-ready.
Calibration	Confidence contradicts quality.	Partly aligned.	Confidence tracks evidence strength and uncertainty.
Actionability	No useful next step.	Some usable guidance.	Clear risks, alternatives and next steps.

Failure Modes

Failure	Risk	Control
Hallucinated option or rationale	Misleads decision owner.	Source-grounded schema, review, unsupported-claim metric.
Omitted critical option	Narrows decision incorrectly.	Recall benchmark and “add option” review affordance.
Wrong criterion weight	Recommendation contradicts user priorities.	Persist reviewed weights and test exact prompt payload.
Fallback presented as AI	False trust signal.	Explicit provenance label and analytics event.
Prompt injection	Instruction bypass or data exfiltration.	Untrusted-data system prompt, test corpus and secret isolation.
Model regression	Output quality changes silently.	Pinned pilot model, prompt versioning and regression suite.

Privacy and Security

- Source text may be sent to a third-party model provider; disclosure and consent are required.
- Do not log transcript content, decision text or model prompts in general application analytics.
- Store model credentials only in backend-managed secrets.
- Define raw source retention and user deletion behavior.
- Use de-identified or approved data for benchmark and staging.

Latency and Cost

Signal	Instrumentation
Request latency	p50, p95 and timeout by provider/model.
Token usage	Input/output tokens per extraction and recommendation.
Estimated cost	Per generation and per finalized decision.
Retry duplication	Duplicate request IDs and user resubmission rate.
Fallback usage	Reason and workflow stage.
Regeneration	Count and cost before acceptance/finalization.

AI Release Criteria

- Fixed benchmark and annotation guide approved.
- Prompt/schema/model versions stored for every generation.
- Evaluation targets met or explicitly waived with risk owner.
- Fallback is visible and measurable.
- Recommendation uses persisted human-reviewed values.
- No critical prompt-injection or cross-user data leak in tests.
- Provider failure has actionable, recoverable UX.

AI Improvement Backlog

Priority	Item	Evidence produced
P0	Persist reviewed extraction and criteria payload.	Request/response integration tests.
P0	Add prompt/model/provenance fields.	Traceable generation record.
P0	Build benchmark and regression runner.	Versioned evaluation report.
P0	Define comparison methodology.	Reproducible score snapshot.
P1	Model routing experiment.	Quality/cost comparison.
P1	Multilingual benchmark.	Language-specific quality evidence.
Post-MVP	Outcome tracking.	Recommendation vs actual-result learning dataset.

Evidence and References

Source	Use in this document
Implementation-derived baseline	DECISIONFLOW_MVP_DEVELOPMENT_PHASE.md, baseline date 14 July 2026.
Low-fidelity prototype	https://minthukyawdev-droid.github.io/decision-flow/
High-fidelity prototype	https://minthukyawdev-droid.github.io/decisionflow-hifi-prototype/
Frontend repository reference	https://github.com/minthukyawdev-droid/decisionflow-webapp
Backend repository reference	https://github.com/minthukyawdev-droid/decisionflow-api
Plan-phase package	DecisionFlow Plan Phase Document Set, version 1.0.

Interpretation rule — Targets, proposed metrics, future releases, and pilot thresholds are not reported as achieved results. Actual results must be added only after executing the relevant test or study.