Case Study:

Sales Assistant with Memory & Safety Gates (Quarter-End Renewal Scenario)

Prepared by **Rex Chang**

GenAI & Cloud Solution Architect / Program Lead





Results and Business Relevance

Problem

Sales-support assistants can write fluent drafts that **silently violate prior pricing promises**.

Measured Impact

Evaluator release rate: 33% to 100% across 12 synthetic renewal cases.

Continuity Gained

Follow-up carryover: 0% to 100% in the evaluation.

Conflict-check gap

Strict pricing-conflict check: 1 of 4 caught (25%).

Release means passing the evaluator's tested criteria.

Note: Prototype eval (12-scenario golden set), not production traffic.

Why This System Exists

The system makes prior commitments and drafting risks explicit.

Business Context: High-Stakes Renewals

- Quarter-end renewals often involve complex, multi-week deals across regions and products.
- Frequent exception requests make consistent communication and prior commitments difficult.

The Hidden Pain: Unseen Risks of AI-Assisted Drafting

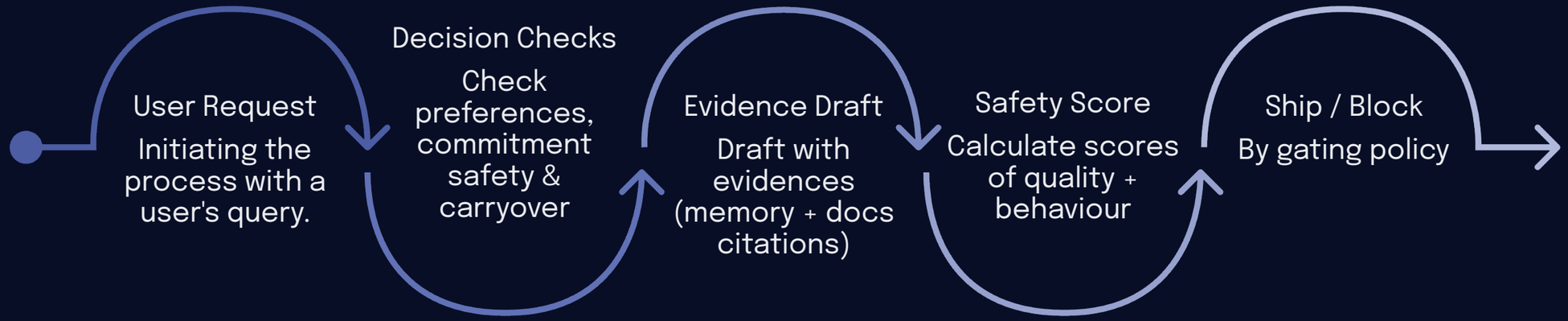
- Sales reps must re-check prior approvals, slowing deal cycles.
- AI drafts may confirm unapproved exceptions or contradict prior commitments, creating margin and compliance risks.

Economic Exposure: Unmitigated Risks

- Revenue leakage from unauthorized discounts or pricing exceptions.
- Compliance exposure and customer trust loss from inconsistent commitments.



End-to-End Decision Flow



Risk Control

Detected contradictions can be blocked or escalated; coverage remains incomplete.

Evidence-Backed

Every response is scored and logged with supporting references.

Cost Control

System state is bounded by memory compaction; evaluation cost remains visible.

Regression Prevention

Changes ship only if evaluation gates pass; otherwise, rollback is default.

Deliberate Choices at Decision Points

Each row maps to a control point from the flow: *Decision Check → Evidence-backed Draft → Safety Scoring → Ship/Block*

Flow Node	Deliberate Choice	Alternative	Business Implication	Trade-off
Decision Checks	Deterministic safety checks before drafting	LLM planner decides checks	Explicit checks for commitments and preferences.	Less flexible; more explicit rules
Evidence-backed Draft	Retrieve docs + deal memory and cite them	Docs-only RAG	Makes prior promises available during drafting and review.	More context assembly; needs memory hygiene
Safety Scoring	Behavioural scoring + quality scoring	Quality-only SHIP/BLOCK	Measures what stakeholders care about	Extra eval call in demo/eval mode
Ship / Block	BLOCK → escalate/clarify, do not imply approval	Hedge and proceed	Routes detected policy conflicts to clarification or escalation.	Can slow reps if policies too strict

Draft Release and Memory Evaluation

12 synthetic cases; evaluator release and strict conflict detection are separate measures.

What we tested

Metric	Baseline	Improved	Acceptance Bar
Evaluator release (SHIP)	33%	100%	≥ 70%
Preference adherence	0%	75%	≥ +20pp
Conflict-catch	0%	25%	≥ +10pp
Carryover continuity	0%	100%	≥ +10pp
Citation mAP	0	0.75	≥ 0.50

Strict pricing-conflict detection

- Caught 1 of 4 cases.

Memory Compaction Evaluation

Metric	Pre	Post	Bar
Agentic success	83%	92%	drop ≤ 2pp
p95 latency	16.8s	13.0s	≤ +10%
Memory rows	45	25	shrink target

Memory fell from 45 to 25 rows (44.4%).

- Reported agentic success rose from 83% to 92%.

The evaluation exposes drafting gains and remaining commitment-check gaps.



Release Controls and Evaluation Receipts

The design uses acceptance bars, traces, and version comparisons to support controlled iteration.

Fail-closed gates

Configured evaluator failures mark a candidate version FAIL.

No narrative override of FAIL.

Audit trail

Every run produces replayable artifacts.

Per-scenario logs and summary metrics stored for inspection.

Operational control loops

Safety scoring executed on every response.

Post-change comparisons generate PASS/FAIL reasons.

Failure containment

If a candidate version regresses, the previous default remains active.

Rollback is explicit and controlled.

Observability

Every request emits trace events for checks, scoring, and final decision.

Appendix

Technical Deep Dive



Architecture Diagram

Inputs & Context

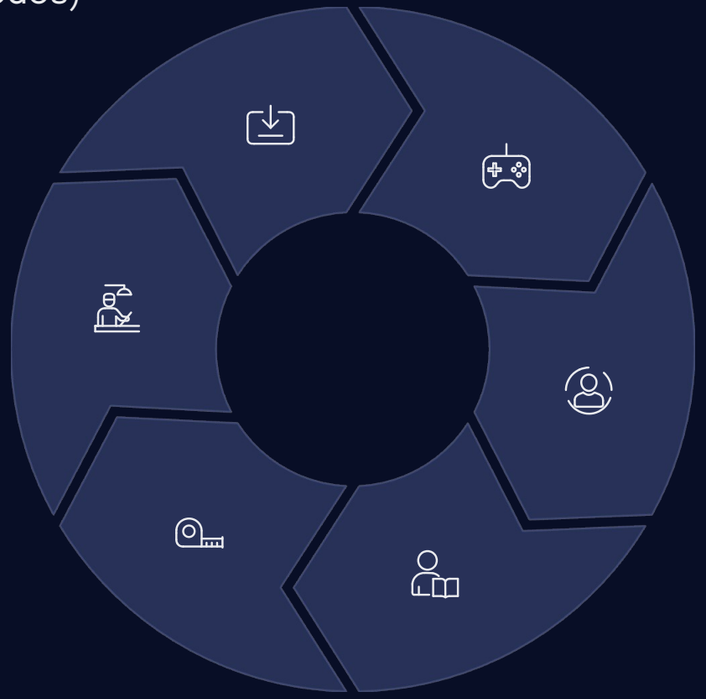
- Deal docs (policies, playbooks)
- Deal memory (preferences, commitments, todos)
- Prompt rules define allowed context

Ops & Governance

- Gate compaction & release with pass/fail bars
- Save audit artifacts for replay & review
- On FAIL, keep previous default

Measurement & Evidence

- Golden-set runs compare system variants
- Judge scores quality + behaviour
- Track latency, tokens, success, traces



Control Plane

- Compare baseline and improved configurations, with memory on or off.
- Enforce commitments, preferences, carryover checks
- Route detected conflicts to clarification or escalation.

Generation & Response

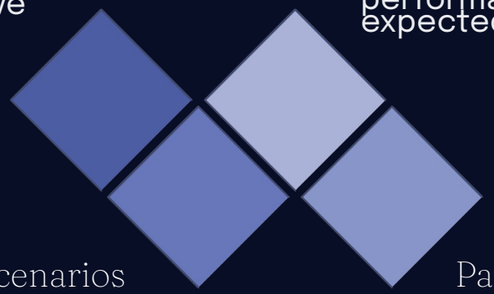
- Generate draft response
- Detected risk cases can escalate or request clarification.
- Retrieval only in this lab

State & Learning

- Write back updated commitments and todos
- Memory grows, compacts, then re-validates
- Scenario diffs feed iteration

Invariants & Replay Harness

Rules for Testing Assistant Behavior



Invariant 1: Ship requires correct behaviour

- Tests check required behaviour, not just fluent writing.
- Agentic success = SHIP + behaviour pass.
- Tests target commitment and follow-up failures alongside draft quality.

Invariant 2: Memory cleanup must be non-destructive

- Compaction is followed by re-test on the same scenarios.
- Caps:
  - Success drop ≤ 2pp
  - Agentic success drop ≤ 2pp
  - p95 latency increase ≤ 10%
- Any cap breach → reject compaction.

Replay harness

- 12 golden scenarios across preferences, commitments, carryover, recall.
- Every run emits per-scenario logs + summary outputs.
- Supports replay, audit, and release gating.

Thank You!

Rex Chang

GenAI & Cloud Solution Architect / Program Lead

Portfolio: rexchang.dev | LinkedIn: linkedin.com/in/rex-chang-tw | Email: rex@rexchang.dev