

Retention Agent – Phase 1–9 Verification & Acceptance Proof

This document verifies that the **Retention Agent** satisfies all required phases (Phase 1–9) of the Sales Intelligence & Coaching Agents framework. Each phase is mapped to **concrete implementation evidence**, including code artifacts, outputs, and design decisions.

The goal of this document is to demonstrate that the Retention Agent is **complete, safe, explainable, and evaluator-ready**.

PHASE 1 – Task Decomposition

Requirement

Decompose the business task into a repeatable, automatable workflow with clear inputs, decisions, and outputs.

Implementation Evidence

Business Task: Manual monitoring of customer usage dashboards to detect churn risk.

Decomposed Workflow:

- Trigger: Daily availability of usage telemetry
- Context: Historical usage trends per customer
- Decision: Churn risk scoring + action selection
- Action: Human-facing recommendation
- Feedback: CSM response logging

Inputs → Outputs:

- Input: usage.csv (mock telemetry)
- Output: Retention CSM cards (daily_output.json)

Acceptance

The task is fully decomposed into deterministic, repeatable steps.

PHASE 2 – Automation Classification

Requirement

Classify which parts of the task are automatable and which must remain human-controlled.

Implementation Evidence

- Automatable:
 - Signal extraction
 - Churn scoring
 - Risk categorization
- Human-controlled:
 - Customer outreach
 - Final retention decisions
 - Message approval or modification

Automation Level Chosen:

Level 1 – Shadow Mode

AI generates insights only, humans decide all actions.

Acceptance

Automation boundaries are explicit and conservative.

PHASE 3 – Agent Architecture

Requirement

Define a clear, modular agent architecture with separated responsibilities.

Implementation Evidence

Architecture Components:

- Trigger / Runner: run_daily.py
- Context & Signals: signal_extractor.py
- Decision Engine: churn_model.py
- Rule Router: action_router.py
- Output Formatting: output_formatter.py
- Feedback Logging: feedback_logger.py

Flow Pattern: Input → Understand → Decide → Suggest → Log

Each component is isolated and testable.

Acceptance

Architecture is modular, explainable, and aligned with the shared backbone.

PHASE 4 – Control Plane & Orchestration

Requirement

Define when and how the agent runs, including safeguards and batching.

Implementation Evidence

- Execution Model: Daily batch
- Entry Point: `python -m src.agents.retention_agent.run_daily`
- Batching: Per-customer processing
- Anti-spam: Threshold gating via rules
- Safety Gate: Kill switch check before execution

Failure Handling:

- Missing data → safe exit
- Kill switch ON → no output generated

Acceptance

The agent runs in a controlled, predictable, and safe manner.

PHASE 5 – Decision Engines

Requirement

Implement explainable decision logic for intelligence generation.

Implementation Evidence

Signal Extraction:

- Login drop
- Engagement drop
- Feature usage drop
- Inactivity patterns

Churn Scoring:

- ML-lite weighted risk points
- Sigmoid mapping to probability
- Calibrated using mock data

Action Routing:

- Rule-based thresholds
- Confidence gating

All components have unit tests and deterministic behavior.

Acceptance

Decision engines are transparent, calibrated, and safe.

PHASE 6 – Human-in-the-Loop

Requirement

Ensure humans remain in control of all consequential actions.

Implementation Evidence

- Outputs are recommendations only
- No automatic customer communication
- Explicit next-step guidance for CSMs
- Feedback actions: accept / ignore / edit

Code Evidence:

- output_formatter.py
- feedback_logger.py
- Manual (non-automatic) feedback logging

Acceptance

Human intervention is mandatory by design.

PHASE 7 – Observability & Feedback

Requirement

Enable monitoring, auditing, and learning from system behavior.

Implementation Evidence

- JSONL feedback logs
- Debug metadata in outputs (signals, thresholds, risk points)
- Deterministic outputs for replay and audit

Measurable Metrics:

- Acceptance rate
- Override rate
- False positives

Acceptance

The system is fully observable without behavioral changes.

PHASE 8 – Progressive Automation Ladder

Requirement

Clearly define the current automation level and prevent unsafe escalation.

Implementation Evidence

- Current Level: Level 1 – Shadow Mode
- No autonomous execution
- Rule + confidence gating
- Explicit separation between insight and action

Upgrade paths are intentionally out of scope.

Acceptance

Automation is constrained and intentional.

PHASE 9 – Governance & Safety

Requirement

Ensure system-wide control, auditability, and emergency shutdown.

Implementation Evidence

- Global kill switch
- Per-agent kill switch (retention_agent)
- Safe degradation when disabled
- No irreversible actions

Kill Switch Configuration:

- kill_switch.json explicitly includes retention_agent

Acceptance

Governance and safety mechanisms are enforced structurally.

Final Acceptance Statement

All nine phases for the **Retention Agent** have been implemented, verified, and evidenced through code, outputs, and documented behavior. The agent meets all evaluation criteria for correctness, safety, explainability, and human control.