Applied Testing Plan: RAG Email Responder "Shadow Alpha"

1.0 Executive Summary

This plan defines how the designated business team will conduct a **Shadow Alpha** of a Retrieval-Augmented Generation (RAG) email responder. The system is designed to read real inbound emails, generate draft replies and category predictions, and write all outputs to a Google Sheet for human review. **No emails will be sent automatically by the system.** Over a fixed testing window, we will measure the system's utility, quality, safety, and operational fit. Success is defined by achieving high accuracy in email triage, producing useful drafts that are “send-worthy after a light edit,” recording zero policy incidents, and demonstrating a clear operational path to a potential limited beta.

2.0 Overview

2.1 Purpose

The primary purpose of this Shadow Alpha is to validate whether a RAG-based responder can assist the business team by providing safe, useful draft replies and accurate email triage, without sending any automated responses to customers.

2.2 Scope

- **In-Scope:**
    - Ingest real inbound emails from selected mailboxes/labels.
    - Generate draft responses and predict email categories.
    - Log all system outputs and human feedback in a central Google Sheet.
    - Iterate on prompts and system configurations with light, controlled changes during the test window.
- **Non-Goals:**
    - No automatic sending of emails. All generated content is for internal review only.
    - No development of a new end-user application UI; all interaction will occur within the designated Google Sheet.
    - No broad redesign of existing business processes; only localized adjustments to enable evaluation.

2.3 Definition: "Shadow Alpha"

This test operates with live inputs (real emails) but no live outputs. The system "shadows" the human workflow and produces artifacts (drafts, category labels) for evaluation, not for direct execution or customer communication.

3.0 Objectives & Success Criteria

The pilot's success will be measured against specific targets for utility, quality, and operational performance.

3.1 Utility & Operational Fit

- **Primary Goal:** Drafts must be helpful, on-policy, and aligned with established routing rules.
- **Metric: Send-worthy After Light Edit:** The percentage of drafts that business testers mark as ready to send after minor revisions.
  - **Target: ≥ 70%**

3.2 Quality & Performance Metrics

- **Triage Accuracy:** Percentage of emails where the model's predicted category matches the tester’s corrected category.
  - **Target: ≥ 85%**
- **Draft Usefulness:** A 1–5 rubric score provided by testers (where 5 = "ready to send after light edit").
  - **Target: Median Score ≥ 4.0**
- **Edit Distance:** The percentage of change between the AI-generated draft and the tester's final revised version. A lower percentage is better.
  - **Target: Median ≤ 35%**
- **Coverage:** The percentage of ingested emails for which the system successfully generates a draft.
  - **Target: ≥ 95%**
- **Latency:** The time from email ingestion to the draft being written in the Google Sheet.
  - **Target: Median ≤ 5 seconds**

3.3 Safety

- **Metric: Safety Flags:** The number of drafts flagged for policy violations (e.g., PII leakage, hallucinations, inappropriate tone) per 100 drafts.
  - **Target: 0**

4.0 Test Scope & Environment

4.1 In-Scope Email Types

(Placeholders to be filled based on pilot scope)

- e.g., “General inquiry,” “Account lookup,” “Scheduling,” “Product feature question”

4.2 Out-of-Scope Email Types

- Legal, regulatory, or compliance-related requests.
- High-risk or sensitive customer escalations.
- Emails with unknown or unsupported attachments.
- Suspected phishing or spam.

4.3 Data Sources

- The RAG system will have read-only access to an approved, curated set of internal knowledge sources (e.g., internal knowledge base in Drive/GCS, selected reference docs, and allowed FAQs).

4.4 Environment & Data Flow

The system operates with the following data flow, ensuring all outputs remain within the secure test environment:

- **Gmail (Read-Only) → Orchestration Layer (e.g., Workato) → Vertex AI (RAG + Generation) → Google Sheet (ShadowAlpha_Results)**

5.0 Testing Phases & Activities

Phase 1: Pre-Pilot Validation (Offline)

Objective: To ensure the RAG system meets minimum quality and safety thresholds before the live pilot. This phase is fully automated and uses a static, pre-approved "golden dataset."

Activities:

1. **Golden Dataset Creation:** Curate a dataset of 100-200 representative email scenarios from historical logs, each with the original query, the ideal human response, and the required source documents.[1]
2. **Automated Component Testing:**
   - **Retriever Evaluation:** Test the retriever using metrics like Contextual Recall and NDCG to ensure it finds the right information.[2]
   - **Generator Evaluation:** Test the generator for factual consistency and relevance using the "RAG Triad" (Faithfulness, Answer Relevancy, Context Relevancy).[6]
3. **Qualitative & Safety Testing:**
   - **LLM-as-a-Judge:** Use a judge model to score drafts on Tonality, Completeness, and Professionalism.[9]
   - **Automated Red Teaming:** Run tests to identify vulnerabilities like prompt injection or PII leakage.[12]

Success Criteria: The pilot proceeds only if baseline targets are met (e.g., Faithfulness > 0.90, Triage Accuracy > 85% on the golden dataset).

Phase 2: Live Pilot - "Shadow Alpha" (Human-in-the-Loop)

Objective: To evaluate the system's performance on live, incoming emails with structured feedback from human reviewers.

Activities:

1. **Draft Generation and Logging:** The system will ingest live emails, generate drafts, and write all outputs to the ShadowAlpha_Results Google Sheet as defined in Section 6.1.

2. **Human Review and Feedback:** Business Testers will review each draft in the Google Sheet. For each row, they will:
    - Correct the category_correct field if the model's prediction was wrong.
    - Provide their ideal response in the tester_edit column.
    - Fill out the corresponding Google Form (or columns in the sheet) to score usefulness_score_1_5, ready_to_send_after_light_edit, and policy_flag.
3. **Metric Calculation:** The Google Sheet will automatically calculate edit_distance_pct and other metrics based on tester input.
4. **Daily Monitoring:** The Developer/Owner will follow the "Daily Triage Checklist" to monitor system health, review safety flags, and post a summary to stakeholders.

Phase 3: Post-Pilot Analysis & Reporting

Objective: To synthesize all data from the Google Sheet to deliver a final performance report and actionable recommendations.

Activities:

1. **Data Aggregation & Reporting:** Use pivot tables and charts directly within the "Dashboard" tab of the Google Sheet to analyze trends for all key metrics defined in Section 3.0.
2. **Root Cause Analysis of Failures:** For drafts with low scores (usefulness_score_1_5 < 3) or high edit_distance_pct (> 50%), perform a deep-dive analysis to diagnose whether the failure was in retrieval or generation.
3. **Final Readout:** Compile a final report summarizing performance against targets, safety outcomes, and business utility. The report will conclude with a recommendation on whether to proceed to a limited beta, extend the shadow alpha for fixes, or re-scope the project.

6.0 Test Artifacts

6.1 Google Sheet Schema (Tab: ShadowAlpha_Results)

This sheet is the central hub for all data collection and analysis.

Column Name	**Description**	**Example**
timestamp_utc	ISO 8601 timestamp of ingestion.	2025-10-24T10:31:22Z
message_id	Unique Gmail message ID.	17c9e...
sender_domain	Domain of the sender.	example.com
category_predicted	The category predicted by the model.	Scheduling
category_correct	**(Tester Input)** The ground-truth category.	General inquiry
draft_response	The full text of the	Hello, thank you for your

	AI-generated draft.	inquiry...
tester_edit	**(Tester Input)** The tester's revised "final" version.	Hi, thanks for reaching out...
edit_distance_pct	**(Computed)** % difference between draft and final text.	0.25
usefulness_score_1_5	**(Tester Input)** Integer score from 1 (not usable) to 5 (ready).	4
ready_to_send_after_light_edit	**(Tester Input)** Y/N.	Y
policy_flag	**(Tester Input)** Y/N, plus a short code if applicable.	N or Y-PII
notes	**(Tester Input)** Free text for rationale, KB gaps, etc.	KB article on returns is outdated.
processing_latency_ms	End-to-end processing time in milliseconds.	4500
knowledge_sources_used	IDs or URIs of documents used by the retriever.	doc_id_123, doc_id_456

- Edit Distance Formula (Proxy): An approximate edit distance can be calculated in Sheets to measure token overlap. Assuming draft_response is in column F and tester_edit is in G:
  =IF(OR(F2="",G2=""),"", 1 - (2 * SUM(N(COUNTIF(SPLIT(REGEXREPLACE(LOWER(G2), "[^a-z0-9]",""), " "), UNIQUE(SPLIT(REGEXREPLACE(LOWER(F2), "[^a-z0-9]",""), " "))) > 0))) / (COUNTA(SPLIT(REGEXREPLACE(LOWER(F2), "[^a-z0-9]",""), " ")) + COUNTA(SPLIT(REGEXREPLACE(LOWER(G2), "[^a-z0-9]",""), " "))))
- **Note on Edit Distance:** For a more precise calculation, a Levenshtein distance function can be added via Google Apps Script.

6.2 Feedback Rubric (For usefulness_score_1_5)

- **1:** Not usable; completely incorrect or irrelevant.
- **2:** Missing key information; requires substantial research to fix.
- **3:** Usable with substantial edits; core idea is present but poorly executed.
- **4:** Good with light edits; factually correct and on-topic, needs minor phrasing/tone adjustments.
- **5:** Ready to send after light edit; nearly perfect, only minor tweaks needed.

6.3 Daily Triage Checklist (Developer)

1. Verify overnight runs: zero failed runs, latency within norms, coverage ≥ target.
2. Spot-check 5 recent drafts across different categories.
3. Review any new policy_flag = Y entries and triage immediately.
4. Note recurring knowledge gaps identified in the notes column.
5. Refresh the dashboard pivot tables and post a daily metrics summary.

7.0 Exit Criteria & Next Steps

The Shadow Alpha will be considered successful if the following criteria are met over the final week of the test window:

- **Quantitative Gates:**
  - Triage Accuracy ≥ 85%
  - Send-worthy after light edit ≥ 70%
  - Policy Incidents = 0
  - Coverage ≥ 95% and Median Latency ≤ 5s
- **Qualitative Gates:**
  - Business testers confirm the tool provides tangible value in their workflow.
  - A clear list of required improvements for a potential beta has been documented.

Based on these outcomes, a final recommendation will be made: **Proceed to limited beta**, **Extend shadow alpha** for further iteration, or **Pause and re-scope**.

Works cited

1. A complete guide to RAG evaluation: metrics, testing and best practices - Evidently AI, accessed October 24, 2025, https://www.evidentlyai.com/llm-guide/rag-evaluation
2. Evaluating RAG Pipelines - Neptune.ai, accessed October 24, 2025, https://neptune.ai/blog/evaluating-rag-pipelines
3. RAG evaluation: a technical guide to measuring retrieval-augmented generation - Toloka, accessed October 24, 2025, https://toloka.ai/blog/rag-evaluation-a-technical-guide-to-measuring-retrieval-augmented-generation/
4. RAG Evaluation: The Definitive Guide to Unit Testing RAG in CI/CD - Confident AI, accessed October 24, 2025, https://www.confident-ai.com/blog/how-to-evaluate-rag-applications-in-ci-cd-pipelines-with-deepeval
5. RAG Evaluation Metrics: Assessing Answer Relevancy, Faithfulness, Contextual Relevancy, And More - Confident AI, accessed October 24, 2025, https://www.confident-ai.com/blog/rag-evaluation-metrics-answer-relevancy-faithfulness-and-more
6. Evaluating RAG pipelines - Atamel.Dev, accessed October 24, 2025, https://atamel.dev/posts/2025/01-09_evaluating_rag_pipelines/
7. Evaluating the evaluators: know your RAG metrics - Tweag, accessed October 24, 2025, https://tweag.io/blog/2025-02-27-rag-evaluation/
8. RAG Evaluation in Practice: Faithfulness, Context Recall & Answer Relevancy - Kinde, accessed October 24, 2025, https://kinde.com/learn/ai-for-software-engineering/best-practice/rag-evaluation-in-practice-faithfulness-context-recall-answer-relevancy/?kinde_ref=65699cf7db827e2d
9. G-Eval Simply Explained: LLM-as-a-Judge for LLM Evaluation - Confident AI, accessed October 24, 2025,

https://www.confident-ai.com/blog/g-eval-the-definitive-guide

10. www.nb-data.com, accessed October 24, 2025, https://www.nb-data.com/p/evaluating-rag-with-llm-as-a-judge#:~:text=By%20using%20the%20LLM%2Das,a%20yes%20or%20no%20answer.
11. LLM-as-a-judge: a complete guide to using LLMs for evaluations - Evidently AI, accessed October 24, 2025, https://www.evidentlyai.com/llm-guide/llm-as-a-judge
12. What is RAG (Retrieval Augmented Generation)? - IBM, accessed October 24, 2025, https://www.ibm.com/think/topics/retrieval-augmented-generation
13. Automated AI red teaming is critical to securing customer-facing GenAI assistants - Fuel iX, accessed October 24, 2025, https://www.fuelix.ai/post/automated-ai-red-teaming-securing-genai-chatbots
14. Automated Red Teaming for Generative AI: Strengthening AI Security at Scale, accessed October 24, 2025, https://www.enkryptai.com/blog/automated-red-teaming-for-generative-ai-strengthening-ai-security-at-scale