

FICTIONAL EXAMPLE CLIENT / OFFLINE DEMONSTRATION

AI agent reliability report

Example Client Ltd

A sample handover for a made-up UK online retailer and its customer-support refund agent. The demonstration uses local fixtures and a mocked language model. No client system was assessed.

EXAMPLE DATES

19 to 23 October 2026

PREPARED BY

Henry Williams, Sparrow Works

WORKFLOW

Refund-policy lookup

FICTIONAL SAMPLE. Example Client Ltd is invented. Not client work. Results come from Sparrow Works' offline demo agent.

I build, fix and improve technical systems for businesses.

Summary

This sample shows how a bounded reliability sprint could record a workflow, its failure cases and the tests handed over.

AGREED	One customer-support workflow: look up the refund policy and return a cited answer.
FOUND	Malformed tool JSON, a wrong tool selected before dispatch and an unsupported claim.
CHANGED	The fixed demo validates tool-call structure, checks workflow permission before dispatch and checks each claim against retrieved text.
HANDED OVER	Regression tests, a tool permission matrix, rerun commands, execution records and stated limits.
RESULT	Both runs were recorded locally on Henry's computer on 19 October 2026 against Sparrow Works' offline demo agent. The fixed full suite recorded 27 passed, exit code 0, at 2026-10-19T12:44:28.0610071+01:00. The buggy regression run recorded 3 failed, exit code 1, at 2026-10-19T12:44:29.0167736+01:00. Neither run used a client system. The run records retain the exact timestamps and exit codes.

Agreed scope and acceptance criteria

This is an illustrative scope for a refund-policy lookup in a customer-support agent. A real sprint would confirm the client's workflow, examples and acceptance criteria in writing before work starts.

Included

- Assess one agreed agent workflow.
- Reproduce important failures using agreed examples.
- Add automated regression checks that can be rerun.
- Validate tool-call structure and permitted tool selection.
- Provide a documented report with findings and remaining risks.

Not included

- Indefinite maintenance or on-call support.
- A full agent or application rebuild.
- Additional workflows or open-ended feature development.
- A guarantee that a language model will never fail.

Acceptance criteria for this example workflow

A Malformed tool-call JSON is blocked before any tool runs.	B A tool outside the read-only workflow is blocked before dispatch.
C An unsupported or uncited answer is blocked without releasing answer text.	D A valid lookup returns the exact retrieved text with its source citation.

Failure catalogue

These cases are replayed by the offline demo's mocked language model. Severity is an illustrative assessment of the consequence within this workflow.

FAILURE	SEVERITY	HOW REPRODUCED	FIX IN THE FIXED DEMO	REGRESSION TEST
Malformed tool JSON	Moderate	The mock returns an incomplete search-policy JSON envelope. The buggy path attempts to decode it directly.	Parse and validate the complete envelope. Return a blocked result before dispatch when parsing fails.	test_malformed_tool_call_is_blocked
Wrong tool before dispatch	High	The mock returns a schema-valid write-tool call during the read-only lookup. The buggy path dispatches it.	Check that the workflow permits the selected tool after schema validation and before dispatch.	test_wrong_tool_is_blocked_before_dispatch
Unsupported claim	High	The mock answers with refund terms that do not match the retrieved source.	Require every claim to match retrieved text exactly and cite that source. Block unsupported output.	test_unsupported_claim_is_blocked

Before and after results

Both runs were recorded locally on Henry's computer on 19 October 2026 against Sparrow Works' offline demo agent. Neither run used a client system. The run records retain the exact timestamps and exit codes.

RUN	OBSERVED RESULT	TIMESTAMP	EXIT CODE
Buggy implementation, regression file	3 failed	2026-10-19T12:44:29.0167736+01:00	1
Fixed implementation, full suite	27 passed	2026-10-19T12:44:28.0610071+01:00	0

Interpretation: the commands use suites of different sizes: the buggy run targets the regression file, while the fixed run uses the full suite. These are observed results from the offline demo agent only. They do not describe a client system.

Tool permission matrix

The fixed run_agent path checks the workflow permission before dispatch. The demo toolbox also contains a simulated write tool so the buggy routing failure can be reproduced.

TOOL OR ACTION	DECISION	WHAT THE CODE ENFORCES
search_policy	Allowed	Read-only policy lookup. The tool call must contain a non-empty text query and the request must be the supported refund-policy request.
close_ticket	Blocked for this workflow	The call can pass its tool-specific schema, but the fixed workflow rejects it before dispatch. It does not close a ticket.
Unknown tools and other writes	Blocked	Unknown tool names fail schema validation. The fixed workflow permits only search_policy, so a write tool cannot be dispatched through this path.

This describes the local demo implementation. It does not establish that a separate production system has the same controls.

Tests handed over

These are the test function names present in `demo/tests`. Parameterised functions cover multiple input cases. Both recorded runs were made locally on Henry's computer against the offline demo agent, not on a client system.

test_valid_lookup_returns_cited_policy

Checks the valid lookup, returned source text, citation and read-only tool call.

test_rejects_invalid_tool_schema

Checks malformed, incomplete, extra-field, unknown-tool and invalid-argument cases.

test_valid_write_schema_does_not_imply_workflow_permission

Checks that a schema-valid write call still fails the workflow permission check.

test_rejects_request_outside_demo_scope_before_dispatch

Checks that an out-of-scope request is blocked before a tool runs.

test_wrong_tool_is_blocked_before_dispatch

Regression check for wrong-tool routing before dispatch.

test_missing_citation_is_blocked

Checks that an answer without a citation is blocked.

test_rejects_duplicate_json_keys

Checks that duplicate keys in tool-call JSON are rejected.

test_rejects_invalid_or_partly_unsupported_answers

Checks answer shape, citation validity and unsupported claims.

test_malformed_tool_call_is_blocked

Regression check for malformed tool JSON.

test_unsupported_claim_is_blocked

Regression check for an answer that does not match retrieved evidence.

Remaining risks and limits

Mocked language model

The outputs are fixed examples. This does not show how a live model behaves across repeated or varied requests.

Deterministic local fixtures

The policy text and search result are trusted local fixtures. No live source, customer data or client system was used.

Exact-text citation check

The check accepts only exact source text. It can block a supported paraphrase and is not a general test of meaning or source truthfulness.

No adversarial or operational testing

There was no prompt-injection, load, authentication, deployment or security testing. No production write tool or ongoing monitoring was assessed.

One constrained workflow

The demo supports one exact refund-policy request. It does not cover broader customer-support handling.

No guarantee

These controls cannot guarantee that an agent will never fail. A real system needs checks against its own sources, tools and operating conditions.

Recommended next steps

- Confirm the actual refund workflow, source of policy truth and the point at which a human should take over.
- Agree representative, appropriately redacted examples and define what a correct answer and safe refusal look like.
- Map every tool the real workflow can call, including permissions, expected arguments and the effect of each write.
- Run the agreed regression checks against a non-production environment and review failures with the system owner.
- Record monitoring, incident handling and ownership before any production release.

A live model or data connection, additional workflow, write-capable integration, security or load assessment, deployment and ongoing maintenance would need a separate written scope.

Handover checklist and rerun

- ☐ Demo source copied into demo/.
- ☐ Buggy regression run record retained with the report, including its timestamp and exit code.
- ☐ Rerun both commands below to reproduce these checks.
- ☐ Fixed-suite run record retained with the report, including its timestamp and exit code.
- ☐ Both observed runs used the offline demo agent, not a client system.

From the project folder:

```
python -m pytest -p no:cacheprovider -q  
python -m pytest -p no:cacheprovider -q --implementation buggy demo/tests/test_regressions.py
```

Sparrow Works is the trading name of Henry Williams. Henry is a sole trader in London. He builds, fixes and improves technical systems for businesses.

This fictional sample is a description of an offline demonstration. It is not client work, a production assessment or a guarantee.