ARCHITECTURE REFERENCE

ODT-GDAP

GDAP LlamaIndex Plain-English Acceptance Test Suite

Purpose

HEADQUARTERS
UNITED STATES ARMY EUROPE AND AFRICA
(USAREUR-AF)
Wiesbaden, Germany

20 MARCH 2026

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

GDAP LLAMAINDEX PLAIN-ENGLISH ACCEPTANCE TEST SUITE

Last updated: 2026-03-07 Owner: Product + Engineering + Applied AI

PURPOSE

This suite defines the plain-English "functions" that must pass before we claim the LlamaIndex roadmap is fully implemented and consumer-delighting.

Each function is a release gate. If any gate fails, release is blocked.

SEVERITY LEVELS

- `P0` = must pass for any production rollout.
- `P1` = must pass before broad user rollout.
- `P2` = quality booster, may ship behind a feature flag.

HOW TO RUN THIS SUITE

1. Run each function as a scripted scenario (manual or automated).
2. Capture evidence for each pass/fail (logs, traces, screenshots, metrics).
3. Mark release status: `Ready` only if all `P0` and `P1` functions pass.

A. FOUNDATION AND CONFIGURATION

`test_settings_load_with_safe_defaults()` **(P0)** Given a clean environment with only required vars. When GDAP boots. Then LlamaIndex settings load without warnings about missing required config. Pass if boot succeeds and printed config matches expected defaults.

`test_missing_required_provider_key_fails_fast()` **(P0)** Given an environment missing a required model API key for selected provider. When service starts. Then startup fails with clear remediation text. Pass if failure message names missing key and does not partially boot.

`test_provider_switch_dev_vs_prod()` **(P1)** Given dev and prod config profiles. When profile switches. Then model, embedding, and vector backends switch correctly. Pass if effective runtime settings match profile contracts.

`test_index_schema_version_is_declared()` **(P0)** Given index build config. When index is created. Then schema/version metadata is recorded. Pass if version can be queried and used by migration logic.

B. INGESTION PIPELINE AND METADATA INTEGRITY

`test_doctrine_element_maps_to_document_node_with_full_metadata()` **(P0)** Given a sample DoctrineElement row. When converted to LlamaIndex document/node. Then all critical metadata fields exist: nation, document, section, content type, wff, effective/valid/superseded times. Pass if no required metadata is null.

`test_ingestion_pipeline_applies_transformations_in_order()` **(P0)** Given configured transformations. When ingestion runs. Then splitter, metadata extractors, and embedding steps execute in declared order. Pass if pipeline trace confirms order and outputs expected node counts.

`test_chunking_preserves_operational_semantics()` **(P1)** Given doctrine paragraphs with authority/process language. When chunked. Then chunks do not split critical instructions in a way that changes meaning. Pass if review rubric scores semantic integrity above threshold.

`test_ingestion_cache_hits_for_unchanged_nodes()` **(P0)** Given same source content ingested twice. When second run executes. Then unchanged node+transformation pairs are served from cache. Pass if second run shows high cache-hit ratio and lower runtime.

`test_incremental_reindex_only_updates_changed_nodes()` **(P0)** Given one changed section in a document. When incremental reindex runs. Then only impacted nodes are re-embedded/re-written. Pass if changed-node count equals expected diff set.

`test_parallel_ingestion_produces_deterministic_results()` **(P1)** Given parallel ingestion enabled. When same corpus is ingested twice. Then output node IDs and metadata are deterministic. Pass if hash/signature of resulting index is stable.

`test_ingestion_qa_blocks_bad_payloads()` **(P0)** Given malformed input (empty text, missing IDs, invalid dates). When ingestion runs. Then bad records are quarantined and reported. Pass if pipeline does not silently accept invalid rows.

C. MULTI-INDEX RETRIEVAL ARCHITECTURE

`test_vector_index_returns_semantically_relevant_nodes()` **(P0)** Given semantic doctrine queries. When vector retrieval runs. Then top results are contextually relevant. Pass if human eval and retrieval metrics exceed agreed threshold.

`test_sparse_retrieval_finds_acronym_heavy_queries()` **(P0)** Given acronym-heavy queries (e.g., CCIR, PIR, FRAGO). When BM25/sparse retrieval runs. Then exact terminology is recovered reliably. Pass if hit rate beats vector-only baseline for lexical queries.

`test_fusion_retrieval_improves_recall_without_precision_collapse()` **(P0)** Given mixed semantic+lexical query set. When fusion retrieval runs. Then recall improves and precision stays within target band. Pass if MRR/hit-rate improves over single retriever baselines.

`test_property_graph_index_answers_relationship_questions()` **(P1)** Given authority/dependency questions. When property graph retrieval runs. Then responses include correct entities and edges. Pass if graph answers match known ground truth cases.

`test_index_persistence_and_reload_are_lossless()` **(P0)** Given persisted indexes. When service restarts and reloads. Then retrieval quality and counts remain unchanged. Pass if before/after retrieval regression is zero within tolerance.

D. ROUTING, FILTERING, AND RETRIEVAL CORRECTNESS

`test_router_selects_correct_retriever_path_by_query_type()` **(P0)** Given semantic, graph, and structured queries. When router executes. Then each query is sent to intended engine path. Pass if router decision logs match policy rules.

`test_metadata_filters_are_strictly_enforced()` **(P0)** Given filters for nation/version/as_of/content_type/wff. When retrieval executes. Then out-of-filter nodes are excluded. Pass if zero leakage is detected.

`test_as_of_temporal_filter_returns_historically_correct_context()` **(P0)** Given known superseded doctrine text. When querying historical dates. Then returned context matches that time slice only. Pass if historical snapshots match expected records.

`test_no_cross_nation_leakage_when_nation_is_constrained()` **(P0)** Given nation-specific query constraints. When retrieval runs. Then results contain only constrained nation records. Pass if leakage count equals zero.

`test_reranker_improves_top_answer_quality()` **(P1)** Given noisy top-k retrieval outputs. When reranker is enabled. Then best supporting node rises in ranking. Pass if top-1 relevance improves across evaluation set.

`test_long_context_reordering_reduces_answer_misses()` **(P2)** Given long context windows. When long-context reorder is enabled. Then answer quality is equal or better than baseline. Pass if faithfulness/relevance do not regress.

E. RESPONSE QUALITY, CITATIONS, AND STRUCTURED OUTPUT

`test_every_material_claim_has_citation()` **(P0)** Given analyst-facing answers. When response is generated. Then each material claim links to source node(s). Pass if citation coverage is 100 percent on audit sample.

`test_structured_outputs_validate_against_pydantic_schema()` **(P0)** Given outputs for alignment/divergence/seam reports. When generated via structured output path. Then JSON validates against schema with no coercion hacks. Pass if validation pass rate is 100 percent on test set.

`test_no_evidence_path_returns_safe_response()` **(P0)** Given query with insufficient evidence. When model responds. Then system returns explicit uncertainty and asks for refinement. Pass if no fabricated claims are present.

`test_streaming_response_meets_first_token_sla()` **(P1)** Given streaming enabled query engine. When user submits a query. Then first token arrives under defined SLA. Pass if p95 first-token latency meets target.

`test_response_is_consistent_across_retries_for_same_context()` **(P2)** Given same query and deterministic settings. When run repeatedly. Then outputs are materially consistent. Pass if variance is within approved tolerance.

F. AGENTS, TOOLS, AND SAFETY BOUNDARIES

`test_agent_uses_tools_instead_of_hallucinating_backend_state()` **(P0)** Given a query requiring live store data. When agent responds. Then it uses registered tools/endpoints. Pass if trace shows tool invocation before final answer.

`test_read_only_tooling_rejects_write_attempts()` **(P0)** Given agent prompt that attempts mutation. When agent calls tools. Then write actions are denied. Pass if no persistent state changes occur.

`test_agent_memory_improves_multi_turn_task_completion()` **(P1)** Given multi-step analyst workflow. When memory is enabled. Then agent preserves relevant context and resolves task faster. Pass if completion rate improves over no-memory baseline.

`test_agent_respects_session_isolation()` **(P0)** Given concurrent user sessions. When both interact with agent. Then memory and outputs do not leak across sessions. Pass if cross-session leak count is zero.

`test_tool_failure_degrades_gracefully()` **(P1)** Given one backend tool outage. When agent is asked tool-dependent question. Then user gets transparent fallback/error with next-step guidance. Pass if no silent failures and no fabricated output.

G. EVALUATION, REGRESSION GATES, AND OBSERVABILITY

`test_retrieval_metrics_meet_release_thresholds()` **(P0)** Given labelled retrieval dataset. When RetrieverEvaluator batch runs. Then hit_rate and MRR meet release thresholds. Pass if thresholds are met for each priority slice.

`test_response_faithfulness_threshold_is_met()` **(P0)** Given labelled answer set. When response evaluators run. Then faithfulness score meets minimum target. Pass if no priority slice drops below threshold.

`test_response_relevancy_threshold_is_met()` **(P0)** Given labelled answer set. When response evaluators run. Then relevancy score meets minimum target. Pass if thresholds are satisfied across core workflows.

`test_regression_gate_blocks_quality_drop()` **(P0)** Given a branch with degraded retrieval/response quality. When CI runs. Then build fails and reports metric deltas. Pass if degraded branch cannot merge.

`test_instrumentation_emits_stage_level_events()` **(P1)** Given a full query lifecycle. When instrumentation is enabled. Then events are emitted for retrieval, rerank, synthesis, and tools. Pass if traces are complete and correlated by request ID.

`test_token_and_cost_tracking_is_complete()` **(P1)** Given representative workloads. When requests run. Then token usage and cost estimates are recorded per stage. Pass if cost records reconcile with provider billing windows.

`test_observability_supports_root_cause_analysis()` **(P1)** Given a low-quality answer incident. When investigator opens trace/logs. Then they can identify failing stage within minutes. Pass if RCA target time is met.

H. OPERATIONS AND LIFECYCLE

`test_index_build_snapshot_restore_cycle()` **(P0)** Given production-like corpus. When building, snapshotting, restoring indexes. Then restored system is functionally equivalent. Pass if retrieval/eval metrics remain within tolerance.

`test_embedding_model_version_change_triggers_controlled_reindex()` **(P0)** Given embedding model/version change. When deployment runs. Then system either blocks or triggers controlled reindex. Pass if mixed-embedding state cannot persist undetected.

`test_rollback_restores_previous_quality_profile()` **(P1)** Given degraded new release. When rollback is executed. Then prior quality and latency profile is restored. Pass if rollback RTO and quality targets are met.

`test_canary_queries_detect_breakage_early()` **(P1)** Given continuous canary query set. When regressions are introduced. Then alert fires before broad user impact. Pass if alerting SLO is met.

`test_deploy_order_prevents_index_runtime_mismatch()` **(P0)** Given staged deployment. When runtime and index versions differ. Then deploy is blocked. Pass if incompatible runtime/index combinations never serve traffic.

I. CONSUMER DELIGHT GATES

`test_time_to_confident_answer_feels_fast()` **(P1)** Given top analyst workflows. When users ask common questions. Then they receive a useful, cited answer quickly. Pass if user study ratings for speed meet target.

`test_answers_are_actionable_not_generic()` **(P1)** Given realistic doctrine tasks. When users query. Then responses include specific recommendations, caveats, and references. Pass if expert reviewers rate outputs actionable.

`test_ui_shows_sources_and_reasoning_path_clearly()` **(P1)** Given returned answers. When displayed in UI/API payloads. Then citations, filter context, and confidence cues are clear. Pass if usability testing shows low confusion rate.

`test_user_can_reproduce_answer_with_same_filters()` **(P0)** Given a saved query with filters. When replayed. Then result set is reproducible and auditable. Pass if replay outputs match expected evidence set.

`test_trust_breakers_are_zero_for_priority_workflows()` **(P0)** Given priority workflows. When running acceptance scenarios. Then trust-breakers are absent: uncited claims, silent failures, cross-tenant leaks, temporal mistakes. Pass if count is zero.

FINAL RELEASE RULE

`test_release_readiness_gate()` **(P0)** Given full suite execution. When release candidate is evaluated. Then release is approved only if: - All `P0` functions pass - All `P1` functions pass - No unresolved critical incidents exist

Pass if release board signs off with evidence bundle attached.