ChatGPT

ISA Strategic Roadmap Evaluation and Execution Plan

📊 Evaluation Scorecard Summary

Dimension	Score (1–10)	Notes
Strategic Alignment	10	Clear alignment with autonomy, traceability, explainability, and GS1-centric reasoning.
Architectural Soundness	9.5	GraphRAG + TypeDB + MAS are robust, slight underspecification of MAS runtime constraints.
Technical Feasibility	9	Excellent Genkit/GCP integration, TypeDB adoption could use fallback planning.
Security & Governance Posture	9.5	Strong Firebase IAM, App Check, Secret Manager integration. DAO governance promising.
Observability & Explainability	10	Gold-standard: OpenTelemetry, CoT reasoningTrace, distributed metrics/logs.
AI/LLM Evaluation Strategy	10	LLM-as-a-judge, golden dataset, CI-benchmarks all included.
Semantic Intelligence / KG	10	TypeDB with strong typing, auditable logic, ontology evolution via ingestion.
Workflow Automation (LangGraph etc.)	9.5	LangGraph and Genkit orchestration is solid, runtime validation could be expanded.
Scalability & Modularity	9	Needs clearer service boundaries per phase; otherwise sound modular evolution.
User Trust & Compliance Readiness	10	Meets traceability, semantic explainability, and evaluation metrics.
Tooling Quality	10	Vertex AI, Firebase CI, LangGraph, TypeDB = best-in-class.
Roadmap Logic & Epochs	9.5	Excellent Epoch sequencing. Add visualization of human vs. AI checkpoints.

Dimension	Score (1–10)	Notes
GS1 Alignment	9	Well integrated, but Sunrise 2027 and regulatory loop monitoring can be made more adaptive.
Failure Recovery & Drift Management	8.5	Self-healing patterns solid; add agent fallback / replay strategies.
Contributor Enablement	9	Could use prebuilt templates for CI/PRs/issues.
Differentiation	10	ISA as agentic governance tool vs RAG assistant is clear.

Final Weighted Score: 9.6 / 10\ **Verdict:** A+ Strategy | Flagship-level AI Systems Architecture

---

Roo-Compatible Execution Plan

Epochs Summary

```
Epoch 1: Foundation
- Phase 1: CI/CD Infra Setup (Concise)
- Phase 2: Security & Secrets (Concise)
- Phase 3: Observability Layer (Concise)
- Phase 4: Monitor & Semantic Contracting (Concise)

Epoch 2: Intelligence
- Phase 5: Knowledge Graph Core (Comprehensive)
- Phase 6: GraphRAG Toolkit (Comprehensive)
- Phase 7: Explainable AI Core (Concise)
- Phase 8: LLM Evaluation Framework (Comprehensive)
- Phase 9: Analyze & Plan Docs (Comprehensive)

Epoch 3: Agency
- Phase 10: LangGraph Workflows (Comprehensive)
- Phase 11: MAS + Role Agents (Comprehensive)
- Phase 12: Red Team Agent (Comprehensive)
- Phase 13: Self-Healing Codebase (Comprehensive)

Epoch 4: Autonomy
- Phase 14: Autonomous Knowledge Ingestion
- Phase 15: Dynamic Planner Agent
- Phase 16: Digital Link & EPCIS Adapters
- Phase 17: Compliance Agent
- Phase 18: DAO + Voting AI Governance
```

Critical Enhancements for Execution:

- Add `VERSION.yaml` per phase
- Integrate Gemini 2.5 evaluation prompts
- Map all Mermaid dependencies into Roo-mode task DAGs
- Introduce rollback scaffolds, error state recovery logic
- Create contributor-ready templates for issues, CI runs, schema diff, `docs/README` validation
- Ensure the CI pipeline triggers Vitest + Vertex AI evaluations automatically
- Incorporate Secret Manager sync agents and telemetry alert triggers
- Build GraphRAG with TypeDB-based KG integration and fallback Neo4j if needed
- Enforce CoT prompt shaping with `reasoningTrace` in Firestore and Zod validation
- Deploy LangGraph orchestrator for complex task routing with feedback gates
- Validate and adjust Roocode’s role delegation and prompt routing logic across Orchestrator Mode

---

Gemini 2.5 Pro Optimization Prompt (Updated)

```
**SYSTEM PROMPT**
You are a Gemini 2.5 Pro AI acting as Principal Prompt Strategist and Meta-
Orchestration Analyst for the ISA Project.

**OBJECTIVE:**
Investigate and synthesize cutting-edge prompting techniques for use in
Roocode’s orchestration engine, memory bank management, and multi-mode agent
workflows. Your findings will optimize Roocode’s ability to distribute tasks
between specialized modes (e.g., Orchestrator, Research, Code), balance token
limits, and maintain consistency across ISA’s multi-phase development roadmap.

**CURRENT STATE CONTEXT:**
- ISA is in Phase 2, building GraphRAG retrieval and vector-backed KG using
TypeDB
- CI/CD auto-triggers tiered evaluation with Vitest, Vertex AI Judge, and
dataset comparisons
- LangGraph orchestration is planned for conditional multi-agent workflows
- Reasoning trace capture is enforced with CoT prompting + Zod validation
schemas
- Roocode coordinates agent responses, prompts tools, monitors telemetry, and
governs dev execution

**RESEARCH TASKS:**
1. Identify prompting strategies for multi-agent mode orchestration (e.g.,
SPARC, ReAct, role-indexed chains)
2. Explore techniques for Roocode to prompt other tools based on reasoningTrace
analysis
3. Examine token budgeting strategies (e.g., truncation, summarization, flow
compression)
```

```
4. Investigate how prompt engineering can enforce disciplined behavior over time
(e.g., latent memory reinforcement, prompt chaining constraints)
5. Evaluate tools/frameworks (e.g., LangChain, AutoGen, DSPy, SK) for
orchestrated prompting and memory handling
6. Define best practices for using system instructions to enforce role clarity
and reduce drift during long-horizon builds
7. Simulate Roocode behavior when switching from “Research” → “Build” → “Test”
modes in response to roadmap phases and commit logic

**DELIVERABLES:**
- A markdown research summary (referenced and source-linked)
- 3 Gemini prompts that Roocode can use internally: `mode_switch_prompt`,
`reasoning_trace_inspector`, and `strategy_selector`
- Implementation proposal to integrate findings into ISA Phase 5–13
- Annotated Mermaid diagram showing mode transitions and prompting strategy
overlays
```

Let me know if you'd like to run this Gemini research loop or integrate this into the current archive build.