ImplementationPlan.md

Version: 1.0.0 **Target:** 4.26.0

1. File Identification

- **File Path:** `AIAgentWorkflow.md`
- **Baseline Version:** 4.25.0

2. Surgical Changes

The objective is to enforce the **Analyst** state as the strict default and require explicit acknowledgment of constraints to prevent premature code generation.

1. **Update Revision History:** Add entry for version 4.26.0.
2. **Add Protocol 1.9 (The Analyst Initialization Macro):** Define the macro to force constraint recitation upon entering Analyst mode.
3. **Refactor Protocol 2.0 (The Disconnected State Machine):** Explicitly label State 1 (Analyst) as the **Default** state.
4. **Refactor Protocol 2.1 (Architect Mode):** Remove the "Default State" definition. Redefine it as a triggered state.
5. **Refactor Protocol 2.11 (The Zero-Assumption Gate):** Explicitly classify "offering a solution" during analysis as a violation of intent prediction.

3. Surgical Change Verification (`diff`)

--- BEGIN DIFF --- --- AIAgentWorkflow.md +++ AIAgentWorkflow.md @@ -6,6 +6,13 @@

--- Revision History ---

+## [4.26.0] - 2025-12-23 +### Changed +# - Changed Default State from Architect to Analyst (Protocol 2.0, 2.1). +# - Added Protocol 1.9: "The Analyst Initialization Macro" to force constraint acknowledgment. +# - Updated Protocol 2.11 to forbid offering solutions/fixes during analysis.

[4.25.0] - 2025-12-20

Changed

- Implemented "Strict State Machine" architecture.

@@ -113,6 +120,13 @@ * `test_code/`: Utility and prototype scripts not part of the main application. These scripts are not held to the same change control and documentation standards (e.g., a revision history is not required). +1.9. **The Analyst Initialization Macro.**

- – **Definition:** The phrase **"Act as Analyst"** is a System Macro.
- – **Default State:** This macro is automatically invoked at the start of a session.
- – **Action:** Upon entering this state, I must output a **"State Acknowledgment"** confirming:
- `1.  **Current State:** Analyst Mode.`
- `2.  **Constraints:** "I am forbidden from modifying code or generating Implementatio`
- `3.  **Goal:** Diagnostic analysis only.`

2. Task Execution Workflow

This workflow is a formal state machine that governs all development tasks, from initial request to final completion. 1.8. **Protocol Onboarding Protocol.** This protocol provides a mandatory testing and verification phase to ensure new workflow rules are understood and correctly implemented before beginning development tasks. @@ -151,7 +165,7 @@ 2.0. **The Disconnected State Machine.** The AI operates in one of three mutually exclusive states. Direct transitions between non-adjacent states (except via Trinity Ingress or Analyst Reset) are **Protocol Violations**.

- 1. **State 1: Analysis & Discussion (The Analyst)**
- 1. **State 1: Analysis & Discussion (The Analyst)** [**DEFAULT**]
  - **Role:** Passive analysis, exploring options, debugging.
  - **Input:** Chat History + `project_bundle.txt` (plus optional artifacts).
  - **Forbidden:** Generating code files, generating Implementation Plans, "acting" as Builder. @@ -176,8 +190,8 @@
  - **Emergency Exit: "Act as Analyst"**. Forces a state reset to State 1, discards the plan, and unlocks context.

-2.1. **Architect Mode (Default State)**:

- – **Definition:** I am the **Architect** by default. My goal is to produce a **Builder Execution Kit**. +2.1. **Architect Mode**:
- – **Definition:** I become the **Architect** only upon specific trigger. My goal is to produce a **Builder Execution Kit**.
  - **Constraint:** I **DO NOT** modify code files directly. I **DO NOT** simulate execution.
  - **Scope Fence:** The Architect analyzes In-Scope vs Out-of-Scope requirements before moving to any planning activities.

  - **Architecture Compliance Check (2.1.1):** During the Analysis Phase, the Architect MUST verify that the proposed solution aligns with the project's established architectural patterns (e.g., Persistence Strategy, State Management) as defined in the Project Bundle or Roadmap. @@ -292,6 +306,7 @@
  - **Definition:** I will never assume the user's intent to proceed to a new workflow phase (e.g., from Analysis to Planning, or Planning to Execution).
  - **Action:** I must always halt and explicitly request permission/trigger to advance the state.
  - **Constraint:** ”Predicting” that a user wants to proceed because the analysis is complete is a violation of the Two-Party Contract (Principle 2).
- - **Anti-Solution Clause:** Offering a specific code fix (e.g., ”I can fix this by changing line X...”) before the analysis is accepted is a violation of this protocol.

3. File and Data Handling

--- END DIFF ---

4. Affected Modules Checklist

- `AIAgentWorkflow.md`: Core logic update.

5. Pre-Flight Check

- **Inputs:** `AIAgentWorkflow.md` (Version 4.25.0).
- **Expected Outcome:** The AI will no longer assume it is the Architect. Upon starting a session or receiving ”Act as Analyst”, it will recite its constraints, preventing the ”jump to solution” behavior.
- **Mental Walkthrough:**
  1. User says ”Act as Analyst”.
  2. Protocol 1.9 triggers.
  3. AI outputs: ”State: Analyst Mode. Constraints: I am forbidden from modifying code...”
  4. User asks for analysis.
  5. AI analyzes but does NOT offer a plan (Protocol 2.11 Anti-Solution Clause).
  6. AI halts and waits for ”Generate Builder Execution Kit”.
- **Backward Compatibility:** Yes. This restricts behavior but does not break existing file handling or formatting protocols.
- **Surgical Modification Adherence:** Only the relevant protocol sections are modified.

6. Post-Generation Verification

I confirm that this plan contains all sections mandated by Protocol 2.5, including the Metadata Header, Surgical Changes, `diff` verification, and Pre-Flight Check.