

# AI Agent User's Guide (Split-Role Workflow)

Version: 2.3.0 Date: 2025-11-24

--- Revision History ---

[2.3.0] - 2025-11-24

- Updated Phase 3 to include "Pre-Flight Checklist" verification.

- Updated Phase 3 to include the mandatory "Verification Command" step.

[2.2.0] - 2025-11-24

- Added `ArchitectureRoadmap.md` to the mandatory bootstrap list.

- Added instructions for saving Roadmap updates.

## 1. The Workflow Concept

We use a "Split-Role" Model to manage context and ensure precision.

- **The Architect:** Thinks, plans, and designs. Maintains the **Architecture Roadmap**. Writes the **Implementation Plan**.
- **The Builder:** Writes **Project Code**. Has "amnesia" (knows only what is in the bundle and the Plan).
- **You (The User):** The "Message Bus" and "FileSystem." You move files between agents and persist state.

## 2. Phase 1: The Architect Session

**Goal:** Define *what* to do and *how* it fits the long-term vision.

1. **Start:** Open a new session.
2. **Bootstrap:** Upload the following files:
  - `AIWorkflow.md` (The Rules)
  - `project_bundle.txt` (The Code)
  - `ArchitectureRoadmap.md` (The Long-Term Memory)
3. **Prompt:** "Act as Architect. [Describe task]."
4. **Output:** The Architect produces:

- **Implementation Plan:** The specific blueprints for the immediate task. Note: The Architect may include files you didn't ask for if it detects a **Systemic Bug** (a bug pattern present in multiple files).
- **Updated Architecture Roadmap:** (Optional) If strategic decisions were made.

### **Crucial Step: Persistence**

If the Architect outputs an updated `ArchitectureRoadmap.md`, you **MUST** overwrite your local copy immediately. This file prevents the AI from forgetting the master plan in future sessions.

### **Phase 1.5: The Architectural Relay & Health Checks**

**Goal:** Manage complex features that require multiple design sessions.

**Session Health Monitoring** In long sessions, check for "Context Decay":

**"System Health Check: Estimate your current context usage and risk of decay."**

- **Green:** Continue.
- **Yellow/Red:** Perform a Relay immediately.

### **Executing the Relay**

1. **Saturation:** Ask for a **State Snapshot**.
2. **Save:** Copy the Snapshot to a file (e.g., `Plan_Part1.md`).
3. **Re-Bootstrap:** Open a **FRESH** session.
4. **Upload:** `AIAgentWorkflow.md`, `project_bundle.txt`, `ArchitectureRoadmap.md`, and `Plan_Part1.md`.
5. **Prompt:** "Act as Architect. Resume design based on the attached Plan and Roadmap."

## **3. Phase 2: The Bridge**

1. **Manifest:** Copy `manifest.txt` from the Plan to your project root.
2. **Bundle:** Run `python test_code/create_project_bundle.py --manifest manifest.txt` to create `builder_bundle.txt`.

## **4. Phase 3: The Builder Session**

**Goal:** Execute the plan with rigorous verification.

1. **Start:** Open a **FRESH** session.
2. **Upload:** Upload `AIAgentWorkflow.md`, `builder_bundle.txt`, and the saved `ImplementationPlan.md`.
3. **Prompt:** Paste the **Builder Bootstrap Prompt** from the Plan.

4. **Execute:** Follow the delivery loop.
  - **Watch for the Pre-Flight:** Before generating code, the Builder MUST output a **Pre-Flight Checklist**. If it skips this, do not proceed.
  - **Visual Diff:** Review the diff. Ensure NO stylistic changes (cleaning up code) are included.
  - **Proceed:** Type **Proceed** to authorize file generation.
5. **Verify:** At the very end of the session, the Builder MUST provide a specific **Verification Command** (e.g., `python main_cli.py ...`). Run this command immediately to verify the work.

## 5. Workflow & Roadmap Maintenance

- **Workflow (`AIAgentWorkflow.md`):** Defines *how the Agent behaves*. Update this if the process feels broken.
- **Roadmap (`ArchitectureRoadmap.md`):** Defines *where the Project is going*. Update this when you finish a Phase or change strategy.
- **User Responsibility:** You are the guardian of these files. The AI cannot save them to your disk; you must copy/paste them when the AI generates updates.