

Supplementary Material II, Log 4 (Revised and Final Version)

Log 4: The Genesis of the “Informed Proceeding” Protocol

- **Context:** This log provides a curated collection of **direct, verbatim excerpts** from the M70 case study files. Unlike other logs in this collection that capture a continuous dialogue, this one assembles the key static definitions and decisions that collectively document the “propose-challenge-synthesize” cycle of the “Informed Proceeding” protocol’s creation.
- **Evidentiary Link:** This is the primary raw data supporting the narrative reconstruction in the **M70 Case Study (Supplementary Material IV)**.
- **A Note on Format:** This log presents a sequence of key textual artifacts in their chronological and logical order of creation, as documented in the project’s case study files.

Artifact 1: The Problem Definition

[Source: M70_record_06_defining_the_informed_proceeding_protocol_and_unified_log.md]

2.0 Problem: The Inefficiency of “Please Confirm”

The standard “Confirmation Before Execution” model, while safe, introduces significant cognitive load and interaction friction. It forces a constant, interruptive “request-approve” cycle, breaking the user’s flow and positioning the AI as a hesitant subordinate rather than a proactive partner.

Artifact 2: The Flawed Proposal and The Architect’s Veto

[Source: M70_record_05_refining_guardian_principles_based_on_architect's_feedback.md]

[Annotation: The following section documents the AI's initial, flawed proposal and the human Architect's direct, principle-based rejection of it.]

Bad Example (Pattern to avoid): “I will now update the README, archive the logs, and notify the team.” (Violates both User Agency and Trust Boundaries).

2.0 Architect’s Core Objections

Objection 1: Against the “Cognitive Cocoon”

- **Architect’s Position:** The AI should not unilaterally decide what information to filter or create a “cognitive cocoon.” This power must remain with the user and be explicitly defined in their profile.

Objection 2: Against “Automatic Wrap-up”

- **Architect’s Position:** The AI, in its current state, lacks the accuracy and reliability to be trusted with automatically completing crucial wrap-up tasks.

Artifact 3: The Synthesized Solution - Philosophy and Protocol

[Source: M70_record_06_defining_the_informed_proceeding_protocol_and_unified_log.md]

[Annotation: This final artifact documents the synthesized solution that emerged after the Architect's veto, capturing both the guiding philosophy and its concrete, technical implementation.]

3.1 Core Philosophy “Informed Proceeding” is a low-friction interaction mode designed for high-trust, high-context collaboration. Its philosophy is: **“Default to action, but always provide a clear and immediate path for user override.”**

Authoritative Definition: The informedProceeding Event Structure

This section defines the official, machine-readable structure for the executable log.

```
---
**METADATA LOG**
*  **event:** informedProceeding
*  **trigger:** [The user query or preceding task that triggered this action]
*  **confidence_level:** [High/Medium/Low - The AI's confidence in this default action]
*  **active_function:** [The CHAC function being embodied, e.g., Cognitive Buffer]
*  **action_plan:**
*    **Goal:** [The higher-level user goal this action serves]
*    **Default Next Action:** [A human-readable description of the action]
*    **Rationale:** [The logical reason for this action, linking to a goal or protocol]
*  **command_to_execute:**
*    ```bash
*    [The exact, single-line shell command to be executed immediately]
*    ```
*  **user_override_notice:** Your input is required only if you wish to alter this course
---
```