# Replication Protocol for AIRR v1.4 Coding

## 1. Prerequisites

- Dataset must include at minimum:  
 - Post\_ID (unique identifier for each post).  
 - Post\_Text (full text of each post).  
- Optional but recommended: Resolution\_Annotation (if prior human codes exist).  
- Data should be formatted in a spreadsheet (.xlsx or .csv).  
- Inclusion filter: Only include posts with at least 25 words unless otherwise specified.

## 2. Coding Framework (AIRR v1.4)

Each post is coded for a primary theme, with optional subtypes and notes:  
  
1. Working/TAP (Think-Aloud Problem-solving)  
 - Baseline exploratory activity.  
 - Includes calculations, reasoning sketches, incremental progress.  
 - Precedence rule: If a post mixes TAP with another move but the functional center is exploratory/problem-solving, code as TAP.  
  
2. Idea Generation (with subtypes):  
 - Problem Reformulation — recasting the entire problem in a new framework.  
 - Special Case Proposal — proposing to handle a subset separately.  
 - Refinement — adjustment or improvement to an existing step.  
 - Alternate Technique — replacing one method with another.  
 - Case Unification — merging multiple subcases under a general framework.  
  
3. Evaluation/Testing  
 - Probing validity, checking assumptions, counterexamples, boundary conditions.  
 - Precedence rule: If the dominant function is evaluating an existing idea, code as Evaluation.  
  
4. Meta/Coordination (with subtypes):  
 - Project Evaluation — reflecting on overall progress, successes, or direction.  
 - Workflow/Task Updates — coordinating tasks, assigning work, noting completion.  
 - Process Proposals — suggesting new organizational strategies or structures.  
 - Shared Resource Updates — announcing or updating external resources (wiki, documents, etc.).  
  
5. Clarification/Communication  
 - Ensuring shared understanding, providing/asking for definitions, acknowledgments, thanks.

## 3. Coding Procedure

1. Initialize columns in the dataset:  
 - AI\_v1.4\_primary\_theme  
 - AI\_v1.4\_idea\_subtype  
 - AI\_v1.4\_secondary\_code  
 - AI\_v1.4\_notes  
 - Meta\_Subtype  
 - Source (“Resolved” if from existing annotation, otherwise “AI\_v1.4”).  
  
2. Apply Resolution Annotations:  
 - If a post already has a Resolution\_Annotation, copy it into AI\_v1.4\_primary\_theme.  
  
3. Code remaining posts:  
 - Read the full Post\_Text.  
 - Identify the functional center of gravity (main purpose of the post).  
 - Assign primary theme using AIRR v1.4 rules.  
 - If theme = Idea Generation, assign appropriate subtype.  
 - If theme = Meta/Coordination, assign subtype.  
 - Add notes to capture coding rationale if needed.  
  
4. Handle overlaps/divergence:  
 - Posts containing multiple signals (e.g., TAP + Evaluation) should be coded to the dominant theme.  
 - Record the post in the Divergence Log if ≥2 strong signals are present.  
 - Use precedence hierarchy: TAP → Evaluation → Clarification → Meta → Idea, unless a stronger signal clearly dominates.  
  
5. Assign secondary codes when appropriate:  
 - If a secondary signal is strong but not dominant, add it to AI\_v1.4\_secondary\_code.

## 4. Divergence Log

For each ambiguous case, log:  
- Post\_ID  
- Short snippet of text  
- Initial Suggestion (e.g., “Multiple possible (Idea/Eval/TAP)”)   
- Final v1.4 Code  
- Notes  
  
Standardize divergence categories where possible (e.g., 'Boundary TAP vs Eval', 'Mixed Idea Subtypes').

## 5. Output Structure

For each coding batch:  
- FullDataset sheet → all coded posts.  
- Summary sheet → theme counts and subtype breakdowns.  
- Divergence Log sheet → ambiguous cases.  
  
When all blocks are complete, merge into a master workbook with:  
- FullDataset\_All (all posts)  
- Summaries\_All (all block summaries)  
- Divergences\_All (all logs).

## 6. Prompting Guidelines for AI Agent

When coding with an AI agent, prompts should emphasize:  
- “Read the full post and identify the functional center of gravity.”  
- “Default to TAP unless there is a clearly framed proposal or evaluation.”  
- “If multiple signals exist, select the dominant one and log it as a divergence.”  
- “Use v1.4 subtypes when coding Idea Generation or Meta/Coordination.”  
- “Assign secondary codes where strong secondary signals exist.”  
- “Maintain consistency with precedent examples in the codebook.”

## 7. Replication Notes

- Always proceed in ascending Post\_ID order.  
- Save work in incremental blocks to avoid data loss.  
- Store both block-level files and a master consolidated file.  
- Divergence logs are essential for AIRR calibration (human–AI agreement).