# **Technical Design Specification: Reflective Coding Agent**

Version: 7.0

Date: 18 June 2025

Status: Proposed (Based on TRR v23.0)

1. Architectural Overview

This specification details the implementation of the "Read-Think-Write" micro-cycle for the Coding Agent. The create\_coding\_plan\_prompt will be simplified to produce a high-level plan only (a list of files and modification goals), while the code\_execution\_node will be enhanced to perform the dynamic generation of file content.

2. Prompt Modifications (prompts\_v7.yaml)

* create\_coding\_plan\_prompt (Modified): This prompt will be simplified. It will now only instruct the LLM to create a list of steps, where each step contains a file\_path and a description of the change. It will no longer ask for tool calls or file content.
* generate\_file\_content\_prompt (New): A new prompt is required for the "Think" step of the execution cycle. It will take two inputs: instruction (the description from the plan) and original\_content (the text from the read\_file tool). Its output will be the new, complete file content.

3. Graph Modifications (agent\_core\_v7.py)

* coding\_planning\_node: The logic remains the same, but it will use the new, simpler create\_coding\_plan\_prompt.
* code\_execution\_node: This node will be completely rewritten to implement the Read-Think-Write loop. For each step in the coding\_plan from the state, it will:
  1. Invoke the read\_file tool with step['file\_path'].
  2. Invoke the LLM using the generate\_file\_content\_prompt, passing the step['description'] and the content returned from read\_file.
  3. Invoke the write\_file tool with the step['file\_path'] and the new content generated by the LLM.
  4. Append detailed logs of each Read, Think, and Write action to the coding\_log in the state.

4. Tool Modifications (supervisor\_v7.py)

* read\_file: This tool will be updated to have a real (though still safe and sandboxed) implementation. It will attempt to read the specified file from the project directory.
* write\_file: This tool will remain a "simulation" for safety. It will not write to disk but will log that the action was successful. This allows us to verify the content the agent *intended* to write before enabling destructive file operations.

This design creates a fully specified, testable Coding Agent that can generate code content dynamically based on a high-level plan.