



AI-Based Code Generator and Code Reviewer

*R.P.R.T.N. Samaraweera¹, A.P.D. Malisha¹, G.U.T.N.C. Jayarathne¹,
U.G.R.D. Maithreepala¹, K.L.K. Sudheera^{1*}, L. Sandeepa¹, H. Wijayatilake¹*

¹ Department of Electrical and Information Engineering, Faculty of Engineering,
University of Ruhuna, Sri Lanka

**Corresponding author: kushan@eie.ruh.ac.lk*

Integrating artificial intelligence into large and legacy codebases remains challenging due to fragmented structure, outdated technologies, limited documentation, and frequent code changes. Existing AI-based code generators and reviewers rely mainly on surface-level context and treat files independently, resulting in a weak understanding of project structure and restricted input length constraints. This work presents a hybrid AI-based Code Generator and Code Reviewer that combines Retrieval-Augmented Generation, Abstract Syntax Tree (AST)-based semantic chunking, and a coordinated multi-agent framework. Large codebases are managed using AST-based chunking, which extracts functions and classes as structurally meaningful units for indexing and retrieval. File-level indexing and hashing are employed to detect modified files and incrementally update embeddings, enabling efficient adaptation to evolving repositories. Conversation history is selectively retained and pruned so that agents share only relevant context, preventing context overflow. The code generator operates in an agentic RAG mode for context-aware querying and in an autonomous agent mode capable of navigating files, applying edits, and executing commands. The code reviewer integrates static analysis tools with specialized AI agents for security, code quality, and code smell detection. Through an iterative refinement process supported by short-term and long-term memory, the system identifies high-impact issues and prioritizes fixes aligned with company coding standards and best practices. Experimental evaluation on the SWE-Bench Verified benchmark (500 real-world issues) demonstrates a pass@1 accuracy of 40.4% using Gemini 2.5 Flash, exceeding the official baseline of 28.73%. The reviewer's fine-tuned agents reached 98.1% (security), 99.8% (code smell), and 99.7% (quality) validation accuracy, and detected 33% more total issues than leading commercial tools. The results confirm that the proposed RAG-AST-multi-agent architecture provides an effective and scalable solution for AI-assisted development in large and legacy codebases.

Keywords: AI agents, AST, Code generator, Code reviewer, Large and legacy codebases