ScottishPower

Software Engineer

Take Home Assignment

1.The Main Task

Your task is to design and implement an intelligent, automated code review system that integrates with GitHub's pull request workflow. This agent should autonomously analyse code changes, provide meaningful feedback, and when necessary, delegate to a refactoring agent to apply improvements.

The main objective is to propose and implement an end-to-end agentic code review pipeline, and to demonstrate its effectiveness through working examples and clear documentation.

Build a Baseline Code Review Agent (Required)

 Set up GitHub integration and webhook infrastructure: Configure automatic triggers when a pull request is opened or updated. Authenticate securely with the GitHub API, handle rate limits, fetch PR metadata, diffs, file contents, and commit history, and post line-level review comments directly to the PR.

 Implement the core code analysis pipeline: Parse diffs to identify changed files, code chunks, and modification types. Retrieve full file contents for context. Load coding-standards rules from YAML/JSON and apply 5–7 rules across Style & Formatting, Code Quality, Security, and Best Practices.

 Generate intelligent review comments: Use static analysis (AST parsing, pattern matching) and LLM-based semantic analysis to produce review comments with clear problem descriptions, file/line references, severity levels, actionable fix suggestions, code examples, and documentation citations. Post comments with proper GitHub formatting.

 Ensure reproducibility and observability: Output all analysis results in machine-readable formats (JSONL/CSV/JSON). Log triggered rules, confidence scores, and reasoning. Add a summary comment with overall statistics and ensure the review process is deterministic and auditable.

Build Multi-Agent Workflow and Advanced Capabilities (Multi-Agent Delegation System)

- Implement an agent-to-agent handoff mechanism where the code review agent can delegate to a specialised refactoring agent.
- Define clear criteria for when delegation should occur (e.g., high complexity scores, multiple violations in same file, low test coverage).
- Build a refactoring agent that can apply automated fixes (extract methods, rename variables, simplify conditionals).
- Have the refactoring agent commit changes back to the PR with explanatory comments
- Implement safety mechanisms (verification agent, test execution, rollback capability) - Optional
- Design the agent communication protocol and state management across handoffs.

1.1. Required Components

Your submission must include:

- Source Code (Repository/Zip) – Well-structured implementation with clear separation of concerns
- README.md (Markdown) – Setup instructions, architecture overview, usage examples
- Configuration (YAML/JSON) – Sample coding standards configuration
- Examples (Directory) – Sample PR review outputs, logs, or demo
- Presentation (PDF/PPTX) – Slide deck covering baseline, enhancements, and results
- Logs/Output (JSON/JSONL) – Machine-readable analysis results showing reproducibility

1.2 Recommended Technology Stack

- Programming Language: Python – Core implementation
- GitHub Integration: PyGithub, GitHub REST API, Webhooks, GitHub Actions – PR automation and API interaction
- LLM / AI: Ollama (recommended), OpenAI API, Anthropic API – Code analysis and review generation
- Agent Framework: LangGraph, CrewAI, AutoGen, or custom – Multi-agent orchestration
- Code Analysis: Tree-sitter, AST parsers, linters – Static analysis and parsing
- Hosting: GitHub Actions, local execution – Workflow automation (optional)
- Checklist: A GitHub account with API access (personal access token with appropriate scopes). Your chosen development environment set up and tested. A test repository to use for PR reviews (can be a simple test repo you create)

1.3 Submission Guidelines

Your solution should be:

- Easy to set up – Clear installation instructions, under 5 minutes to run
- Well-documented – Architecture diagrams, code comments, comprehensive README
- Configurable – Customisable rules, standards, and agent behaviour
- Testable – At least one example PR or test case included
- Secure – Proper secrets management via environment variables

1.3 Timelines

The candidate has 7 days to complete the task. Please spend approximately ~4 hours on this assignment.

1.4 Scope Management

If you don't complete all features within 4 hours,

Please include:

- Summary of what you accomplished
- Planned next steps if you had more time
- Architectural decisions and trade-offs made
- Known limitations or areas for improvement