

Code Review Summarizer (Static-Analysis Aware)

Problem Statement:

- Pull request #347 has 2,000 lines of changes across 47 files. The junior developer who needs to review it has no idea where to start.
- The senior developer who should review it doesn't have 2 hours to dig through every line. Meanwhile, hidden in line 1,847 is a SQL injection vulnerability that nobody noticed.
- Code reviews either take forever (blocking deployment) or get rubber-stamped (letting bugs through). Half the team doesn't even read past the first file.
- Important architectural changes get buried in formatting fixes, and security issues hide in the noise of refactoring.

Solution Description:

- Build an intelligent code review assistant that reads pull requests like a senior developer would.
- It summarizes what actually changed ("Refactored authentication to use JWT tokens instead of sessions"), highlights risky areas ("Modified payment processing - requires careful review"), and catches common issues ("Potential N+1 query in getUserData()").
- It understands the difference between cosmetic changes and logic changes, knows your team's coding standards, and can spot security vulnerabilities.
- Instead of overwhelming reviewers with every detail, it provides an executive summary with deep-dive links for areas that need attention.

Input:

- Pull request diff with all file changes.
- Your team's style guide and coding standards.
- Security checklist and common vulnerability patterns.
- Static analysis results from tools like SonarQube or ESLint.
- Historical code review comments and patterns.

Desired Output:

- Executive summary of changes ("Authentication refactor: 70% of changes, Payment fix: 20%, Cleanup: 10%").
- Risk assessment with specific concerns ("High risk: Payment processing logic changed without tests").
- Security findings with severity ("SQL injection possible in line 234 of user_handler.py").
- Suggested review focus areas ("Pay special attention to the new caching logic in lines 145-289").
- Compliance check against team standards ("Missing error handling in 3 new functions").

Tech Stack:

- Open-source only.

Evaluation Criteria (recommended):

- Bug detection rate;
- Security issue identification.
- Review time reduction.
- Developer satisfaction.
- Post-release defect reduction.
- False positive rate