# ****Code Review AI Agent with Security Focus - Process Overview****

**<https://www.sciencedirect.com/science/article/pii/S0164121224001055>**

**<https://arxiv.org/html/2507.02976v2>**

## ****The Complete Workflow (6 Key Steps)****

## ****1. Trigger & Code Capture****

* **GitHub webhook** fires when developer creates/updates a Pull Request
* Agent **clones the repository** and extracts the code diff
* **Parses code using AST** (Abstract Syntax Tree) for structural understanding
* Identifies changed files, functions, and dependencies

## ****2. Multi-Agent Analysis System****

* **Based on Endor Labs' approach, your system deploys 3 specialized AI agents:**
* **Developer Agent (GPT-5-Codex): Summarizes what actually changed in the code**
* **Architect Agent (Claude 4): Categorizes changes using security-focused taxonomy**
* **Security Agent (DeepSeek-R1): Evaluates impact on overall security posture**

## ****3. Multi-Layer Security Scanning****

* **Static Analysis: Catches syntax errors, code smells, formatting issues first**
* **Dependency Vulnerability Scan: Checks for known CVEs in packages/libraries.**
* **LLM Context Analysis: Evaluates business logic, authentication flows, data handling.**
* **Performance-Security Tradeoff Analysis: Suggests secure AND optimized solutions**

## ****Intelligent Risk Assessment** Categorizes vulnerabilities: SQL injection, XSS, hardcoded secrets, insecure configs Assigns priority levels: Critical/High/Medium/Low based on business context Provides rationale: Explains WHY each issue matters and potential impact Suggests specific fixes: Not just "fix this" but "change line 47 to use parameterized queries"**

## ****5. Automated Feedback Delivery Posts comments directly to PR**** in GitHub's "Files changed" tab

**Creates summary report** with overall security score and key recommendations

**Sends notifications** to Slack/Teams with critical issues flagged

**Updates security dashboard** showing trends across repositories

## ****6. Continuous Learning Loop****

**Tracks developer responses** to suggestions (accepted/rejected/modified)

**Learns from feedback** to reduce false positives over time

**Adapts to codebase patterns** and company-specific security policies

**Updates rules engine** based on new vulnerability discoveries

## ****Key Technical Components****

**Core Stack**: FastAPI backend, Redis for queuing, PostgreSQL for audit logs, Docker containers  
**AI Integration**: GPT-5-Codex API, Claude 4 API, DeepSeek-R1 for reasoning  
**Security Tools**: Snyk API, OWASP ZAP, custom rule engines  
**DevOps**: GitHub Actions, webhooks, CI/CD pipeline integration

## ****Real-World Example****

When a developer submits a PR with a new login endpoint:

**Developer Agent**: "Added authentication endpoint with password validation"

**Architect Agent**: "New security boundary - authentication logic modified"

**Security Agent**: "CRITICAL: Password stored in plain text, no rate limiting, SQL injection possible"

**Output**: Detailed PR comments with specific code fixes and security best practices

This creates a **24/7 security expert** that never misses details, learns constantly, and scales across unlimited repositories - solving the exact pain point where manual reviews fail.

1. <https://www.endorlabs.com/learn/introducing-ai-security-code-review>
2. <https://www.legitsecurity.com/aspm-knowledge-base/ai-code-review>
3. <https://www.reddit.com/r/AI_Agents/comments/1il8b1i/my_guide_on_what_tools_to_use_to_build_ai_agents/>
4. <https://www.augmentcode.com/guides/ai-agent-workflow-implementation-guide>
5. <https://www.qodo.ai/blog/3-steps-securing-your-ai-generated-code/>
6. <https://www.openxcell.com/blog/ai-code-review/>
7. <https://www.greptile.com/what-is-ai-code-review>
8. <https://dev.to/arkad23/automating-github-code-reviews-with-ai-agent-114p>