Argus: Recovering code change rationale from software artifacts

The Incoming tide of New Code

- In 1985, Peter Naur talked about programming as theory building[1]
- Modern developers are inundated with code reviews, leading some to declare "The End of Code Review"[2], decry code reviews[3], and say code review is dead[4].
- One study[5] at Microsoft found a **24%** increase in PRs/engineer/day after introducing agentic tools like Claude Code and Copilot CLI at Microsoft in early 2026.
- However, a study[6] from 2025 found that LLMs introduced 185 security vulnerabilities, compared to 20 by developers.
- It's time to take back context.

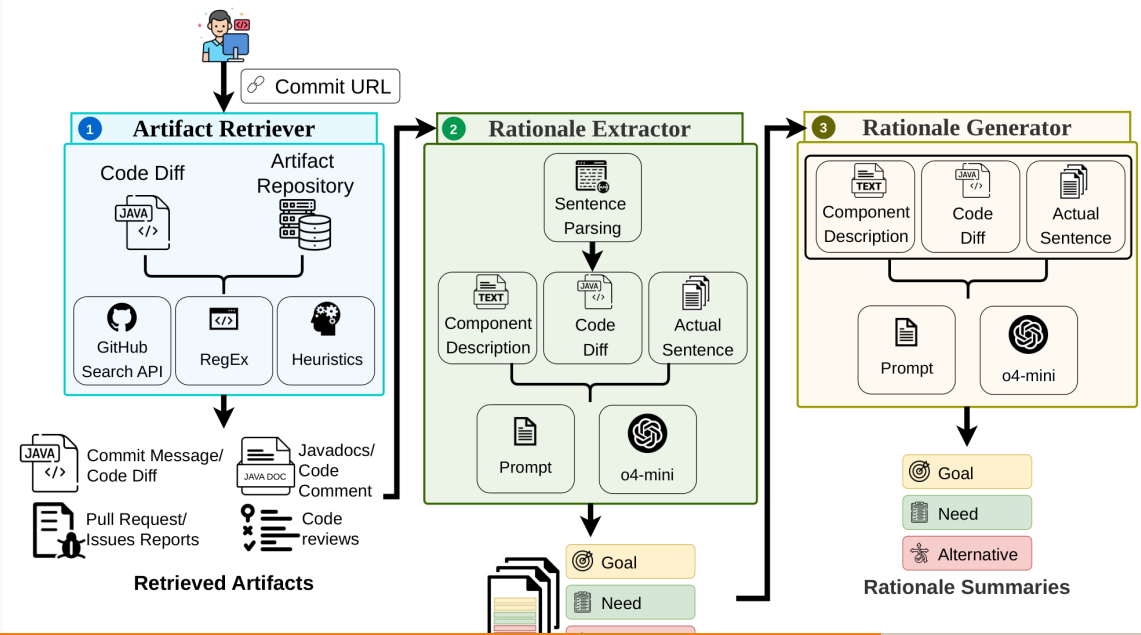
[1] Programming as Theory Building
[2] The end of code review
[3] What is happening with code reviews
[4] The code review is dead... long live the code review
[5] Adoption and Impact of Command-Line AI Agents
[6] Are AI-generated Fixes Secure?

Understanding Rationale

- Rationale provides context about the justification for a code change that can't be recovered from just the code diff[7].
- Our previous study[8] found that **Goal**, **Need**, and **Alternatives** are the most common of seven rationale components.
- So we built a tool called ARGUS to do collect, identify, and synthesize rationale information for developers.

[7] Information needs in collocated software development teams
[8] Recovering Fine-Grained Code Change Rationale from Multiple Software Artifacts

How ARGUS works



Let's build it

- A 2023 study[9] noted:
  "We conclude that in-IDE prompt-less interaction with LLMs is a promising future direction for tool builders."
- So that's what we did. Link to Extension

[9] Using an LLM to Help with Code Understanding