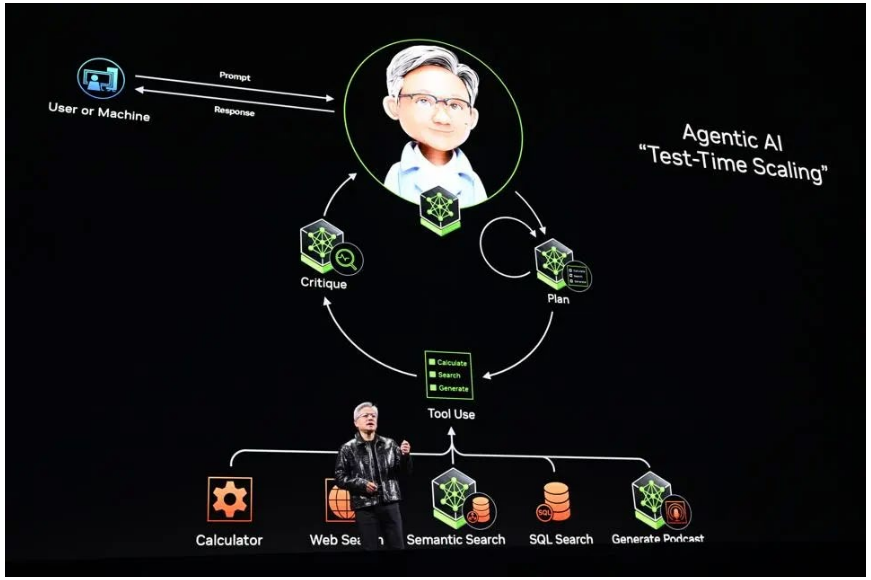
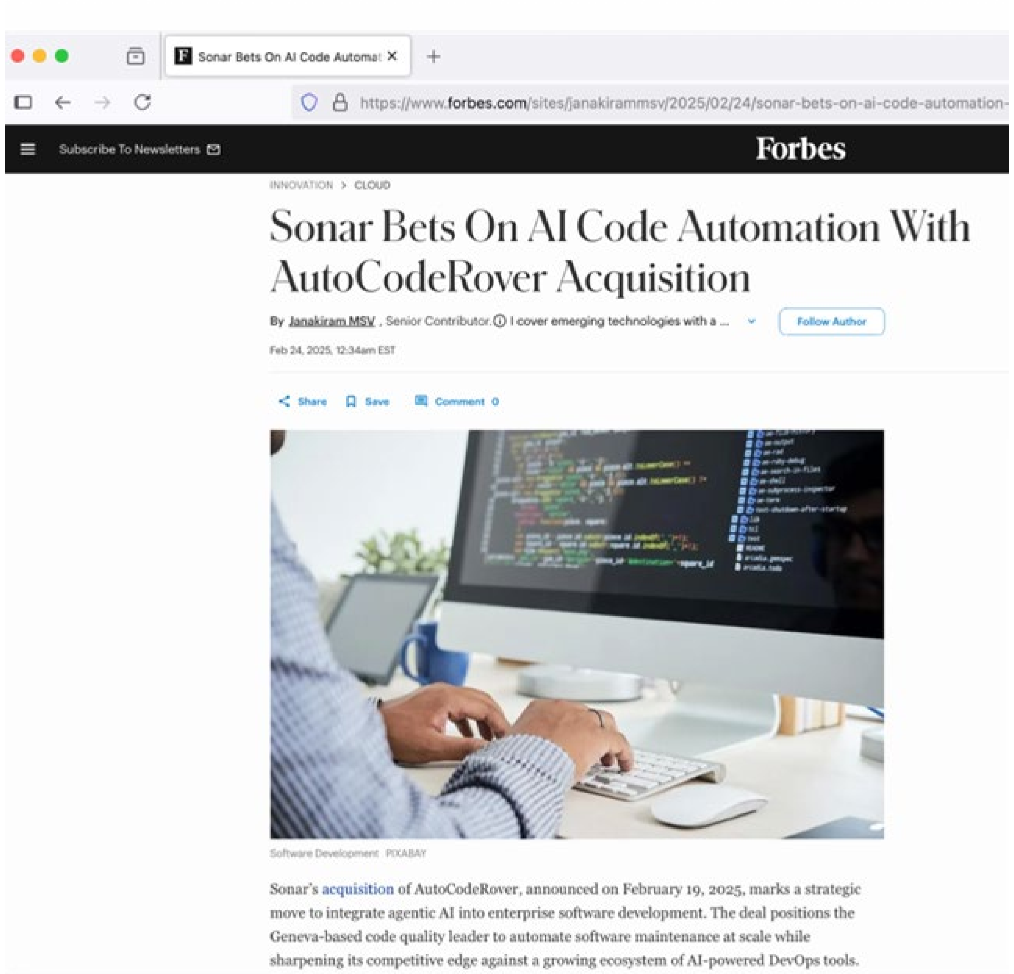
Past work on Agents: AutoCodeRover

Nvidia CEO Jensen Huang Consumer Electronics Show (CES) 2025 unveiled advanced AI for training agents, robots and cars. (Photo by Artur Widak/Anadolu via Getty Images)
Anadolu via Getty Images

2025: “AI agents represent a multi-trillion $ opportunity”



Integrated inside SonarQube Code Analysis tool
In-use by > 100,000 enterprise customers for enhancing code quality and security.


Sonar Bets On AI Code Automat ×
https://www.forbes.com/sites/janakirammsv/2025/02/24/sonar-bets-on-ai-code-automation-
Subscribe To Newsletters
Forbes
INNOVATION > CLOUD

Sonar Bets On AI Code Automation With AutoCodeRover Acquisition

By Janakiram MSV, Senior Contributor. ⓘ I cover emerging technologies with a ...

Follow Author

Feb 24, 2025, 12:34am EST

Software Development PIXABAY

Sonar's acquisition of AutoCodeRover, announced on February 19, 2025, marks a strategic move to integrate agentic AI into enterprise software development. The deal positions the Geneva-based code quality leader to automate software maintenance at scale while sharpening its competitive edge against a growing ecosystem of AI-powered DevOps tools.

Existing work: Agent for Regulatory Compliance

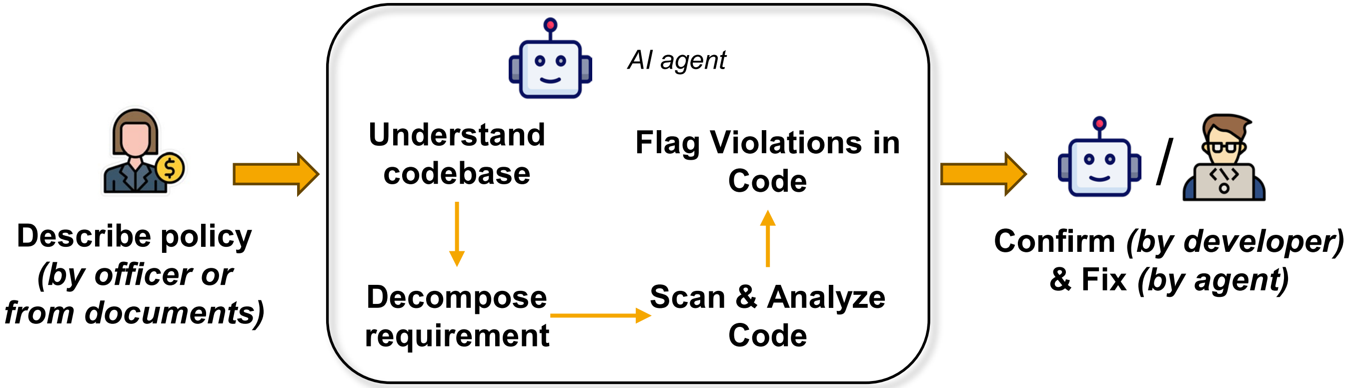
Core capability of the agent

- LLM agent clarifying high-level requirements in natural language
- Transform requirements into actionable sub-requirements on codebase
- Program analysis to determine whether sub-requirements are met
- Coding agent to fix requirement violations after human confirmation
- Work with both closed- and open-source LLMs, can be deployed fully on-prem

Examples of high-level policies

- All personal data must be encrypted before being stored in database. *(PDPA)*
- All transactions over $10,000 must trigger a special approval workflow. *(Transaction Integrity)*
- All APIs handling transactions must use TLS 1.3 protocol or higher. *(Cybersecurity)*

Workflow



Example workflow by the agent

Policy: Without authorization, subscription pricing should not be manipulated.

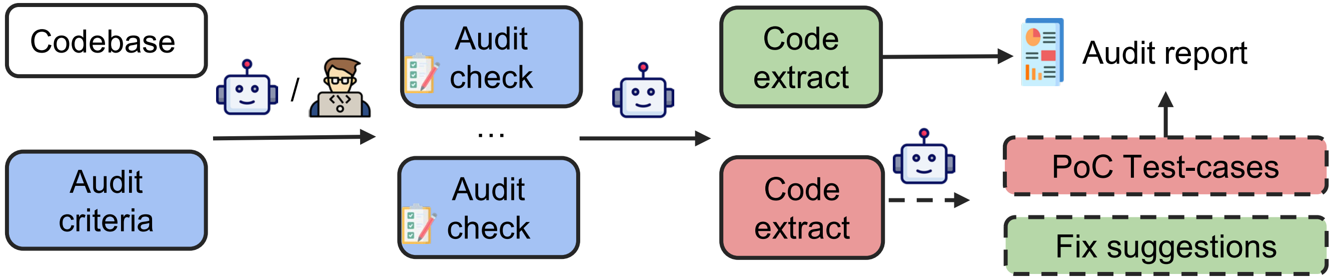
- Explore codebases to find relevant code components: subscription database, payment gateway, etc.
- Within components, identify relevant code location: e.g. *functions/handlers/addToSubscription.js: line 12*
- At code locations, generate sub-requirement: confirm the subscription price matches that in the DB
- Invoke program analysis to check this requirement
- Generate a test-case if there is a potential violation
- (After human confirmation) Propose a code fix to the violation

Trust layer on AI-generated Code

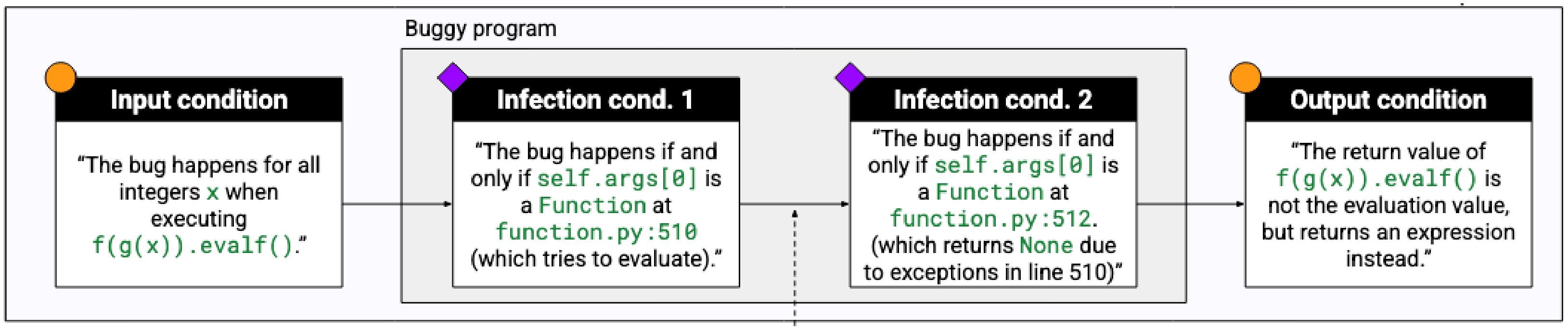
Core capability

- Provide artifacts to enhance trust on AI-generated code (from LLM / agent).
- Lightweight verification on generated code.
- Proof-of-Concept test-cases.
- Text explanation on code changes. Explanation generated based on symbolic properties instead of purely LLM.

Use Case: Security Audit *(test-cases to enhance trust)*



Example of property-based explanation



Input condition
"The bug happens for all integers x when executing f(g(x)).evalf()."

Infection cond. 1
"The bug happens if and only if self.args[0] is a Function at function.py:510 (which tries to evaluate)."

Infection cond. 2
"The bug happens if and only if self.args[0] is a Function at function.py:512. (which returns None due to exceptions in line 510)"

Output condition
"The return value of f(g(x)).evalf() is not the evaluation value, but returns an expression instead."

"The auto-generated code prevents this error propagation, which is why it is correct"

Property-based reasoning is provided to developers as an explanation, instead of relying on an LLM for explanations.