AI agents for the work around the code and the change

Agents that read your issues, your proposed changes, your test runs and your build logs, prepare the work, and stop for a named engineer. On infrastructure you control.

THE PROBLEM

Of 14 public engineering agent templates and example configurations we opened, none requires a named person or a branch rule before a change reaches the main branch or production; 8 do so in part and 6 not at all. Of 135 public interface descriptions, 115 import and 20 are over the import limit; the 115 give 24,103 operations, 12,931 sent as changes, so tools are chosen one by one.

HOW IT WORKS, IN THREE STEPS

STEP 01

Connect what the agent may read.

The coding standard, the review checklist, design documents, decision records and runbooks become knowledge sources, with each reader's permissions kept. Source control, the issue tracker and the build system are reached through the description of an interface you already run; diffs, test reports and build logs are uploaded by the engineer who holds them.

STEP 02

Decide where it stops.

Every tool call passes your policy. A tool that would merge, deploy or change a system waits for a named person.

STEP 03

Run it where the source code lives.

In your data centre or your private cloud, on a model you serve.

THE RULE EVERY AGENT FOLLOWS

It prepares. A named engineer decides.

Thirty-five agents in nine areas

Backlog and issues	Issue Triage. Duplicate Issue Check. Bug Report Check. Backlog Summary
Specifications and design	Specification Draft. Design Review Preparation. Decision Record Draft. Requirement Trace Check
Code review	Change Summary. Code Review Preparation. Change Impact Note. New Dependency Check
Tests	Test Case Draft. Test Failure Summary. Unstable Test Report. Test Gap List
Documentation	Knowledge. Code Documentation Draft. Interface Change Note. Runbook Draft
Dependencies	Dependency Update List. Licence Check. Advisory Impact
Build and release	Build Failure Summary. Release Note Draft. Production Change Record Draft. Release Readiness Check
Incidents and reliability	Incident Response. Post-Incident Review Draft. Error Report Grouping. Service Level Report
Running the department	Technical Debt Register. Engineer Onboarding Guide. Delivery Report. Change Control Evidence

Also in the pack: the full list of what each agent reads and where it stops, and the white paper “AI agents inside the engineering department”, with sources.

Seven questions to ask before an agent touches a repository, a build or a production system

01 Where does it run, and where is the model?

02 Does it act as the person using it, with that person’s permissions?

03 Which tools merge, deploy or change a system, and does each wait for a person?

04 Are the read operations and the change operations separate tools?

05 Does it approve a change or stand as the second reviewer, or does an engineer?

06 What does it do with instructions hidden in an issue, a file or a package?

07 What is recorded, and can we stop one agent or all of them at once?

WHERE THE PLATFORM RUNS TODAY

In production at Taiwan Cooperative Bank, air-gapped, inside the bank’s own perimeter.

MODON built its AI Centre of Excellence on Katonic inside the Kingdom of Saudi Arabia.

ePLDT, the ICT arm of the Philippines’ largest telco, selected Katonic to power Pilipinas AI, a sovereign AI cloud hosted in the Philippines.

Where to start

One team, one repository, invented data first. A knowledge agent on your engineering documents can run on the first day, because no repository is changed.

KATONIC.AI/SOLUTIONS/ENGINEERING