YOUR AGENTS LACK CONTEXT

# Here's how to fix "You're absolutely right!"

Dennis Pilarinos

Unblocked

AI-generated code should feel like it was written by someone who's been on your team for years.

For years, **you** have been the context layer.

Unblocked

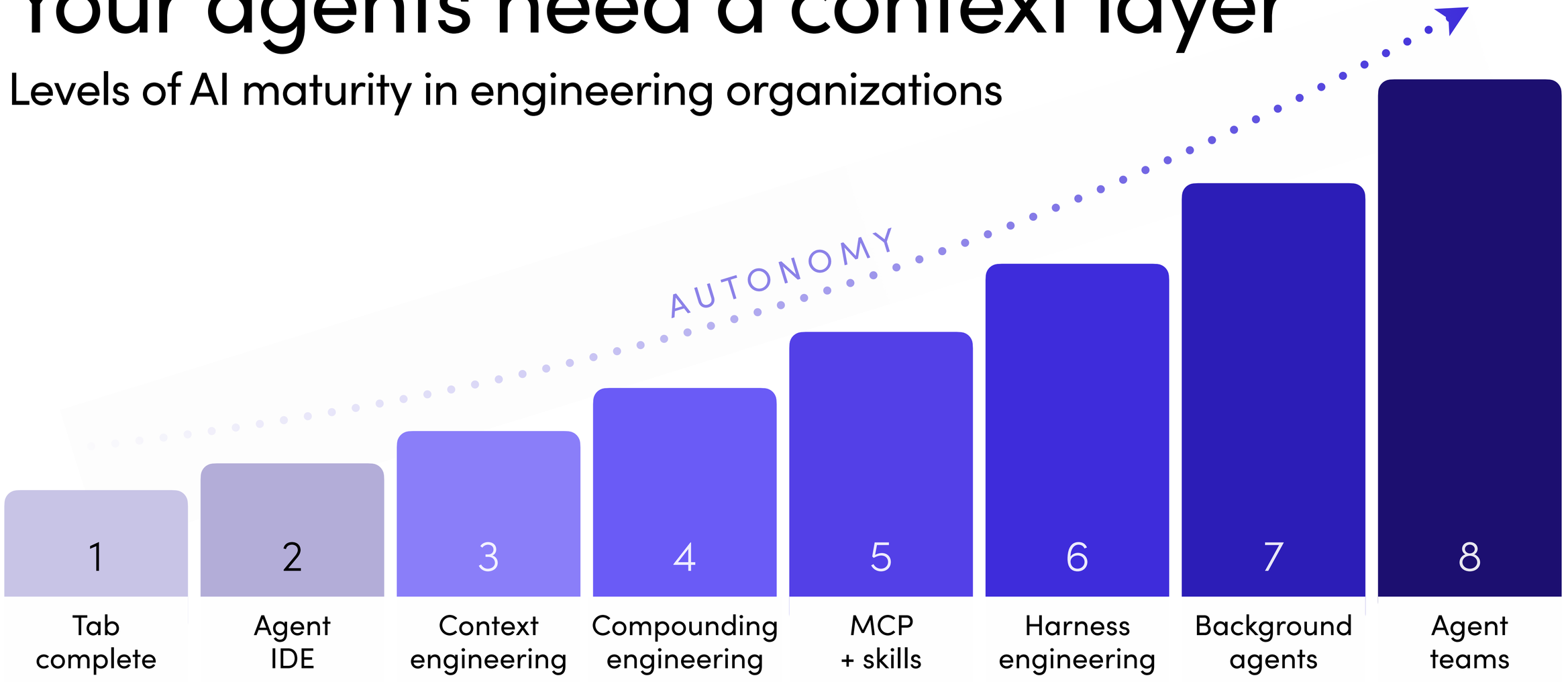
# Remember how you built context?

# Your agents have the same problem.

Unblocked

# Your agents need a context layer

## Levels of AI maturity in engineering organizations




Unblocked

Stages adapted from Basssim Eledath

# The cost of bad context compounds with every step toward autonomy

TAB COMPLETE

## You catch it

One wrong completion. You hit escape, move on.

AGENT IDE

## Doom loops

You re-prompt 3–5 times. Tokens burned, time lost.

PARALLEL AGENTS

## Review tax

Five PRs land. Your senior engineers spend the day reviewing instead of building.

WITHOUT YOU IN THE LOOP

## Silent failure

Wrong code ships. Wrong ticket enriched. Nobody saw it happen.

# Common approaches that don’t work

- **Curated-context trap**
  The shared rules repo rots faster than you can update it
- **MCP plateau**
  More servers, same output quality — agents don’t know when or why to call them

# The problem: access ≠ understanding

# Four myths about building a context layer

01 More rules and CLAUDE.md files will fix it

02 If I just connect enough MCPs, I'm done

03 A bigger context window will solve this

04 Naive RAG over my docs is a context layer

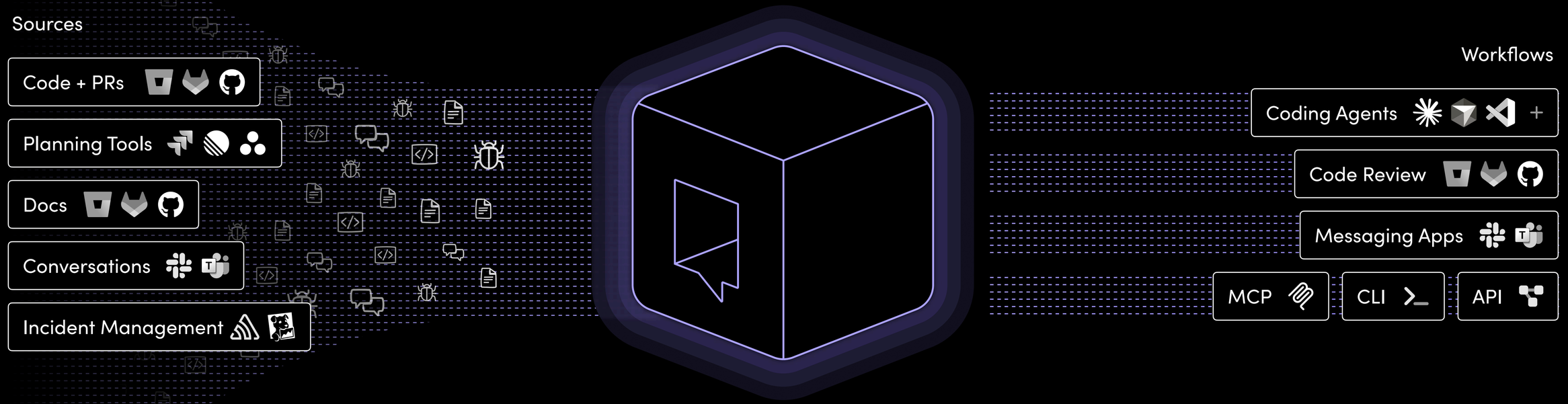

*What your agent can't see*

Unblocked

# Minimum requirements of a context layer

- Understands who you are and which information actually matters
- Resolves conflicts between different sources (code, docs, Slack, tickets)
- Respects permissions and governance
- Delivers a token optimized response depending on the “client”

# How a context layer works




### Unified system context
Merges signals from all sources before delivery to the agent.

### Targeted retrieval
Retrieve only the information agents need from the graph.

### Conflict resolution
Recency + authority signals resolve contradictions automatically.

### Personalized relevance
Scoped to your repos, teammates, and work history.

### Token optimization
Ranked, compressed context — no waste in the prompt window.

### Permission enforcement
Permissions and policies enforced automatically across systems.

Unblocked

# Six requirements for a real context layer

### Unified system context
Merges signals from all sources before delivery to the agent.

### Targeted retrieval
Retrieve only the information agents need from the graph.

### Conflict resolution
Recency + authority signals resolve contradictions automatically.

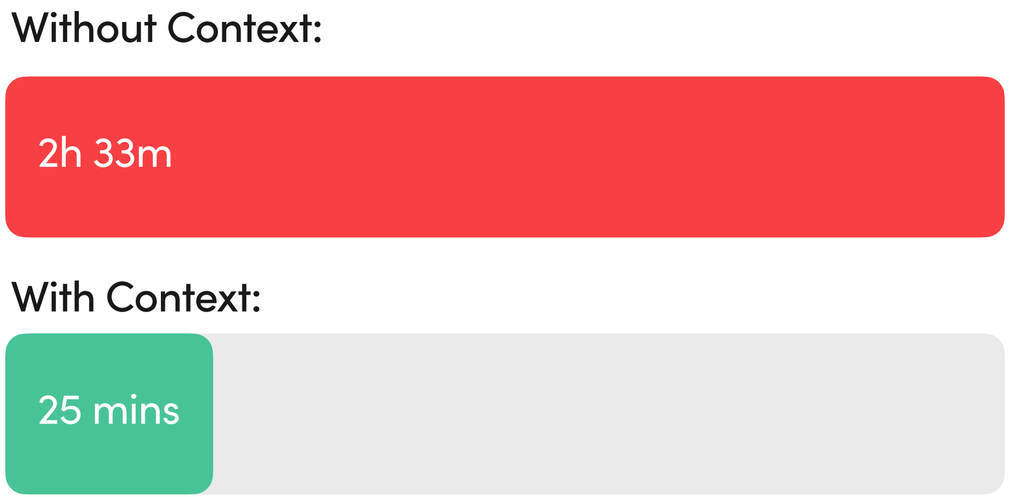
### Personalized relevance
Scoped to your repos, teammates, and work history.

### Token optimization
Ranked, compressed context — no waste in the prompt window.

### Permission enforcement
Permissions and policies enforced automatically across systems.

# Same prompt. Same model.

## Time Savings




**2 hours** saved

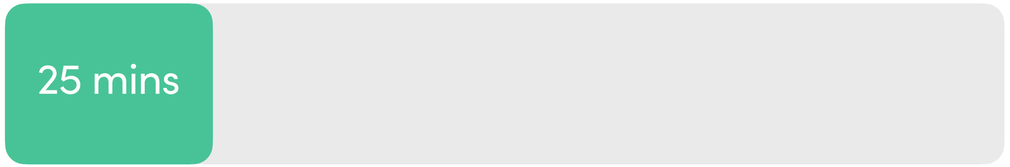
Unblocked

# Same prompt. Same model.

## Time Savings

Without Context:


With Context:




**2 hours** saved

## Token Savings

Without Context:


With Context:


**48%** fewer tokens

Unblocked

“ ~50% fewer tokens, faster triage, and the answer was actually better because it wasn't buried in noise."

- Charles, Cloudbeds

# Open Source Software to help

(GET YOUR CAMERAS READY)

Unblocked

# Context Engine Simulator

Experiment Results — June 2, 2026 at 08:18 AM **Powered by Unblocked**

| | | | | | |
|---|---|---|---|---|---|
| Repository | unblocked | Branch | | claude-opus-4-8 | Duration 12m 18s |

QUALITY IMPROVEMENT

**+7**

77 → 84 out of 100

TASK COST

**-33%**

$4.1130 → $2.7601

## Quality Score

BASELINE

**77**/100

WITH CONTEXT

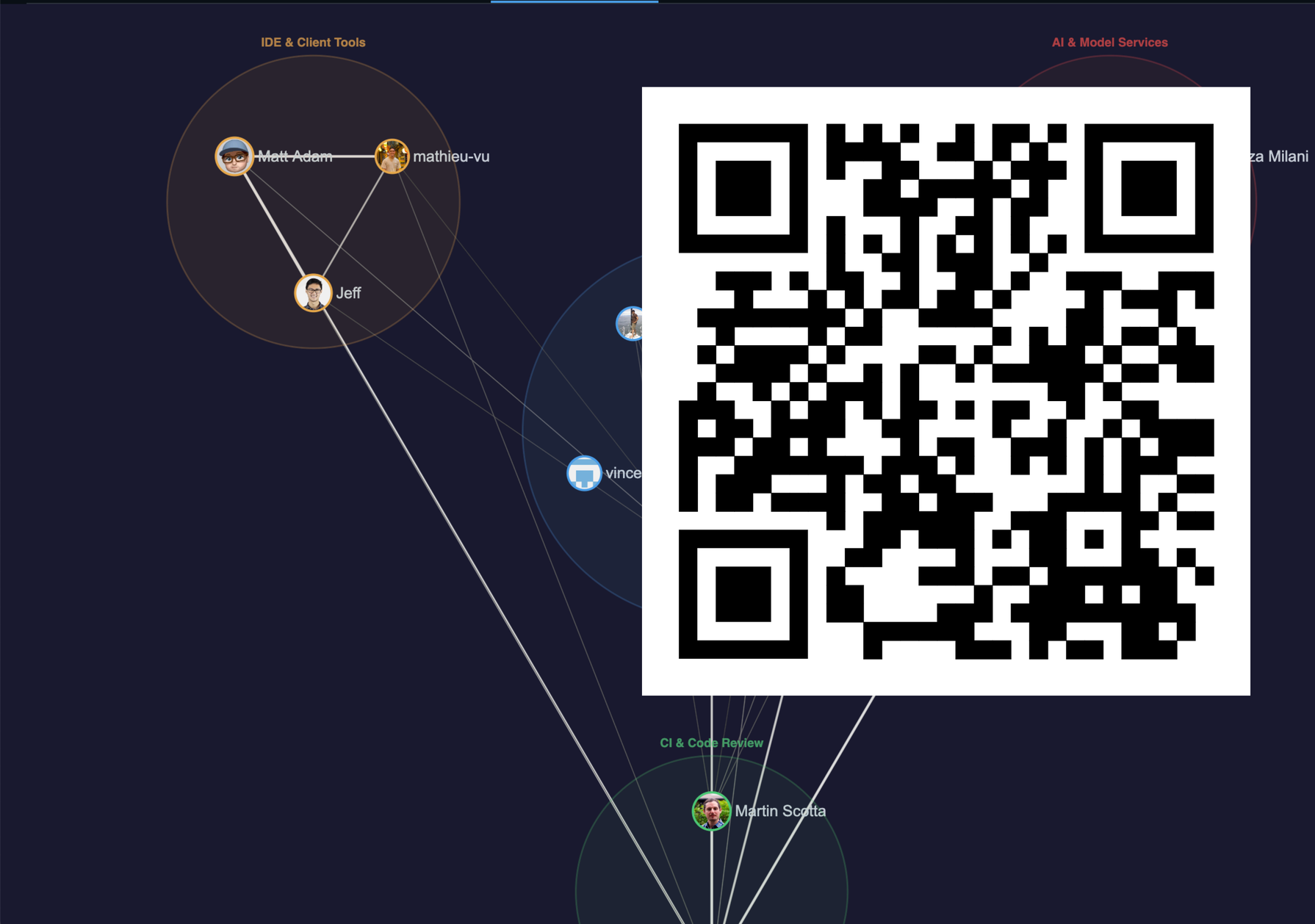
**84**/100

Unblocked

# Social Comment Network

Determines how interested team members are in each other's discussions, based on historical activity in their pull request comments.

Heat Map Grid | Peer Tables | Teams | Experts | **Interactive Graph**




Hover to highlight connections. Click a node for details.

**Rashin Arab**
@rasharab

Community 1 · Weighted degree: 6.33

### RASHIN ARAB REVIEWS PRS FROM

| | |
|---|---|
| andrey-unblocked | 1.000 |
| Narayan Sainaney | 1.000 |
| Richie Bresnan | 0.884 |
| David Lam | 0.860 |
| Peter Werry | 0.729 |
| Martin Scotta | 0.362 |
| tristansturgess | 0.291 |
| Morteza Milani | 0.259 |
| vincent-unblocked | 0.174 |

### WHO REVIEWS RASHIN ARAB'S PRS

| | |
|---|---|
| andrey-unblocked | 0.460 |
| Martin Scotta | 0.157 |
| Richie Bresnan | 0.155 |

● Here's a 13-rule sample (5 must / 5 should / 3 can) from unblocked:

| # | Sev | Title | Source |
|---|---|---|---|
| 1 | **must** | Never use non-null assertion operator !! | S.md |
| 2 | **must** | Understand LoadableState pattern and useStream | end/AGENTS.md |
| 3 | **must** | Build CLI binary with OS/arch-specific npm comm | end/cli/AGENTS.md |
| 4 | **must** | Deploy landing pages via separate static Astro | end/web-landing/AGENTS.md |
| 5 | **must** | Project structure organization for Python code | n/AGENTS.md |
| 6 | should | Use exhaustive when expressions for enums and s | S.md |
| 7 | should | Prioritize consistency with existing patterns o | end/AGENTS.md |
| 8 | should | Verify context-* tool calls via trace logging | end/cli/AGENTS.md |
| 9 | should | Astro component structure: page → island → Reac | end/web-landing/AGENTS.md |
| 10 | should | Use snake_case, PascalCase, UPPER_SNAKE_CASE na | n/AGENTS.md |
| 11 | *can* | Rarely specify font-family; use inherited root | end/AGENTS.md |
| 12 | *can* | Run individual linting tools: oxlint, oxfmt, st | end/AGENTS.md |
| 13 | *can* | Use unblocked-git-utils for blob SHA computatio | n/apps/repo-rules-agent/AGENTS.md |

Sampled diverse source files where possible. The full 124-rule index is at ~/Library/Caches/repo-rules/unblocked-95f13827/index.json if you want to slice it differently (e.g. query --task co
--severity must).

Unblocked

# Use cases of a context layer beyond code generation

### AGENT PLANS & CODE

Hydrate context before planning to improve codegen quality.

### TICKET ENRICHMENT

Gathering context across systems to improve tickets.

### TRIAGE OPERATIONS

Routing issues, enriching tickets, validating bug reports.

### INCIDENT MANAGEMENT

Improving response and identifying root cause.

### CODE REVIEW

Grounded findings that cite team conventions and prior PRs.

### CUSTOMER SUCCESS & SALES

Surfacing relevant knowledge in Slack / Teams.

# Where are you? Here's what's next.

01

## Find yourself on the staircase.

Most teams sit between Agent IDE and Parallel agents — stages 2–4.

02

## Pick the next step.

One stage forward in a quarter is doable. Two stages requires a real context layer.

03

## We help teams skip ahead.

Come talk to us at our booth.

Unblocked

> The gap is not intelligence.
> It's context. – Andrej Karpathy

# Thanks!

Dennis Pilarinos

dennispi

@dennispilarinos

Unblocked