Brik AI

AGENTIC SDLC PLATFORM

A process engine for teams of AI agents

Professional AI SDLC: stages, roles, checks

A person sets the goal. Brik AI moves it through stages: each has its own role, its own rules and a record of the step.

Claude Code | Codex | Cursor | ZCode

Who it is for

One product: for a developer and for a company

For a developer

Free

$0

- 1 project, up to 2 agents at a time.
- The same stages, roles and checks as teams have, without corporate extras.
- Connects over MCP to your Claude Code, Codex, Cursor or ZCode.

There is no self-registration yet: access is by invitation.

For a team and a company

Team, Business, Enterprise

from $29 per person per month

- One shared order for the team: stages, roles, a check before merge.
- A record of steps, token accounting, rights to knowledge.
- Parallel agents in a shared task queue.

Models on your keys, code in your repository.

What broke

There is more code than your team can check

Teams with high and low AI use are compared. Two years of data.

+441%

Time in review

Median time in review grew in teams with high AI use.

+51%

Size of a pull request

Requests got bigger and harder to read.

+28%

Bugs per pull request

More code means more bugs in every request.

+31%

Requests merged with no review at all

Their number grew by about a third with high AI use.



Sources. Faros AI, the AI Engineering Report 2026 and the Faros blog post of April 27, 2026: two years of data, 22,000 developers, 4,000 teams. Teams with low and high AI use are compared. Faros sells engineering analytics. DORA, 2025 report: 90%, nearly 5,000 respondents.

For companies

What a company gets at a professional level

In the code

A managed process

Each stage names a role, a model and rules. A task goes through the set of stages your team chose.

In the code

Quality control and review

A separate role checks the work before merge. A task does not move on until the conditions of the stage are met.

In the code

Audit and traceability

A task, a requirement with a revision number and a run of a stage are linked. A run has a journal: what was issued to the agent and what it read again. Steps and the result are recorded in the task.

Partly

Rights to knowledge

A role does not see the sections of knowledge that are not open to it. We will check isolation between different clients' projects on a shared instance before a pilot.

In the code

Token accounting

A run reports input, output and cache tokens. You see by task and by flow how many went to the brief and to re-reading. We do not count money: agent environments do not report cost.

In the code

Parallel agents

Several agents take tasks from the shared queue of a flow. One agent holds a task, a second cannot take it. The number of agents in a flow is limited.

Partly

Different agent environments

A plugin is built for each project: Claude Code, Codex, Cursor, ZCode. Protection against context loss on compaction exists only in Claude Code. We have not checked the setup on your team yet.

Term

Code and models stay yours

Models run on your keys, code stays in your repository. We do not offer self-hosted. Enterprise is a dedicated environment in our cloud.

“In the code” means: implemented and covered by tests in the platform repository on 2026-11-23. We did not run the tests while preparing the page.

What is better

How Brik AI differs from other classes of tools

An AI agent without a process · A task board with an AI plugin · Your own rules file · Manual review

What a team needs	Brik AI
A shared order: stages and roles	Stages from a set, a role for each stage
A check by a separate role before merge	A check stage with a separate role
Team rules without bloating the context	A role brief: its required rules, the rest on request
A requirement with a version the task is bound to	A requirement with a revision number, the task is bound to it
A record of steps and tokens per task	A run journal, steps in the task, tokens by task and by flow
Several agents without overlap	A flow queue: one agent holds a task

Modern approaches and how Brik AI covers them

Approach	How Brik AI covers it
Spec-driven development	A requirement is a record with a revision number. A task is bound to a revision. A stage does not start until the task is checked against the new revision (where it is switched on).
Context and harness engineering	A role brief: required rules and a list of optional ones. A context limit per role. Rules that must not be broken are led by a plugin with hooks.
A pipeline of roles and a separate reviewer	A role for each stage. A role records whether it changes code. The check before merge is a stage with a separate role.
Checks as gates: tests and automatic checks	Your team sets the order of checks in the rules of a role and a stage. The product makes them mandatory: the gates of a flow and a stage do not let work pass.

Modern approaches and how Brik AI covers them

Approach	How Brik AI covers it
Parallel agents in their own working copies	A flow queue with a hold on a task. The branch and the working copy are recorded in the task. The number of agents in a flow is limited.
Oversight by risk level	A task tier and gates by tier. A heavy stage only with a reason. Heavy runs and deletion of data only with a recorded owner word: the server checks it.
Measurement, not a feeling	Runs, attempts, time and tokens by task and by flow are recorded. Numbers replace the feeling of speedup.
Background agents and sandboxes	A dispatcher that starts agents and isolated sandboxes exist in the code. We have not checked them on a foreign team yet.

A hook is a script that the agent environment always runs at a set point. SSO is a single sign-in through the company account. A working copy is a separate copy of the repository for one task.

How a task goes

One path from a goal to checked code

Each project sets its own stages and their order. The agent works on the next stage and does not skip a required one.

01

Story

A person writes the requirement and the product analysis: who the task is for and what “done” means.

A person writes. A person checks.

02

Plan

The agent splits the task into levels: requirements, architecture, steps. The code is not touched yet.

The planner agent writes. A person approves.

03

Implementation

Code notes record what changed and why, linked to files.

The executor agent writes. The reviewer checks.

04

Review

A second agent or a person raises remarks with a severity.

The reviewer writes. The fixes stage checks.

05

Fixes

Each remark is fixed or closed with a reason.

Nothing is lost.

06

Knowledge

Conclusions go to the project knowledge base and help in the next task.

Any agent writes. A person checks.

Why trust us

We build our own platform in this order

160,000+

lines of code in the process module: stages, roles, runs, queues

≈ 3,000

tests in this module

Roles get only their own rules

Each stage names a role. The role reads the required rules and a list of optional ones. The context does not bloat.

Requirements live in records with a revision number

A task is bound to a revision of the requirement. If the requirement changes, everyone sees it.

What is found becomes a task

What an agent noticed and did not fix is not lost. It is a record with evidence.

A heavy check only with a reason

Every costly stage is switched on with a justification. The rest runs light and fast.

There are no public client cases yet. Brik AI is open by invitation for now.

Plans

You pay for a seat in the team

Free

$0

one developer

1 project, up to 2 agents at a time.

Team

$29

per person per month

Teams of 3–30 people. All projects, up to 5 agents per person, a rule base, roles and stages.

Business

$59

per person per month

For companies. An action journal exists already. Sign-in through SSO and your own agent dispatcher are planned.

Enterprise

from $1,500

per month

A dedicated environment in our cloud, a service-level agreement, support. Planned. We do not offer self-hosted.

These are launch prices, a hypothesis. There is no online payment yet: we issue an invoice by hand.

Request a demo or get Free

Tell us about yourself or your team. We will suggest a time for a demo or send an invitation.

Request a demo for your team