n'co studio

FEB.26

Repo OS

A multi-agent governance model for AI-Powered development with a built-in spec builder, prompt suite, quality gates, automated workflows, mechanical enforcement, and knowledge graph.

The problem

AI Agents Are Powerful, But Without Governance, They’re A Risk.

No Guardrails

AI agents can expose sensitive data, make sweeping changes, and skip quality checks. All without anyone reviewing their work.

Architecture Drift

Without clear boundaries, agents start making decisions that should be made by humans. Choosing technologies, inventing features, changing direction.

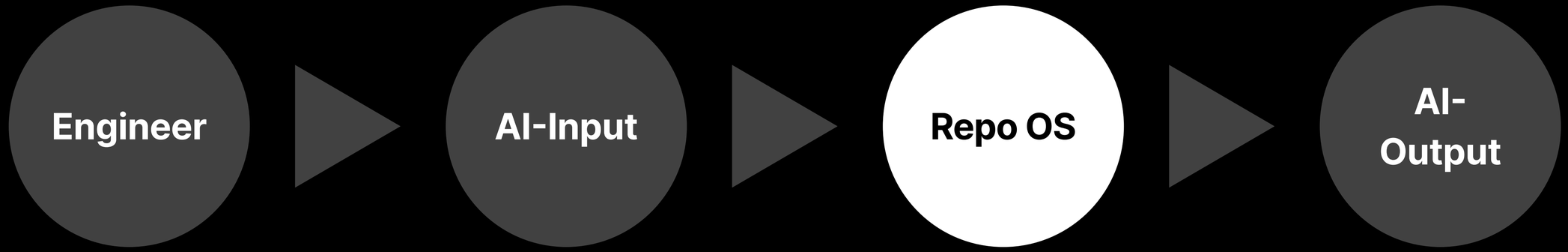
Audit Gaps

No record of what was decided, which checks ran, or why a change was approved. When something goes wrong, there’s no trail to follow.

The solution

Repo OS Embeds Governance Directly In The Development Workflow.

Making every AI action structured, verified, and traceable.



Designed with intention.

Every decision in Repo OS traces back to one of these four goals.

1. Quality Control

Every output is reviewed before it ships. Agents don’t self-approve — independent specialists verify each other’s work across every commit.

2. Token Efficiency

Structured sessions prevent drift, rework, and wasted context. The agent does the right work the first time — not three attempts to get there.

3. Automation

Mechanical enforcement replaces manual oversight. Gates run on every commit, hooks fire on every action. Nothing relies on the agent choosing to comply.

4. Traceability

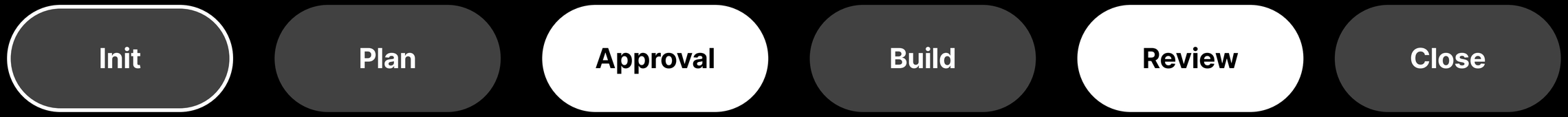
Every decision, override, and outcome is logged. The system produces a complete audit trail — not just code, but the reasoning behind it.

How it works

First We Structure The Work, Then Govern The Structure.

Different types of work follow different phase sequences.
With each phase following specific steps and triggering specific hooks.

Build sequence example:



Human input Autonomous Human review

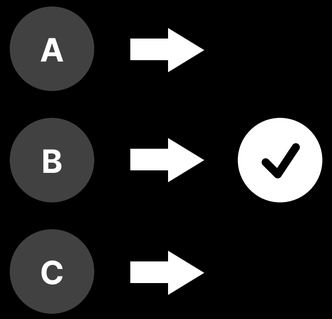
Agents Work Through Phases In 4 Different Ways.

1. Sequential

One agent finishes, the next begins.

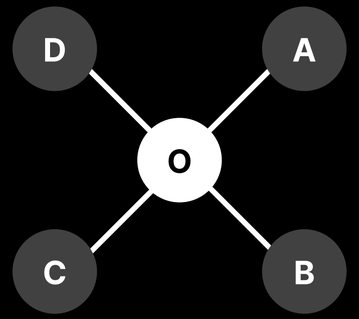
2. Simultaneous

Multiple agents work in parallel on the same task.



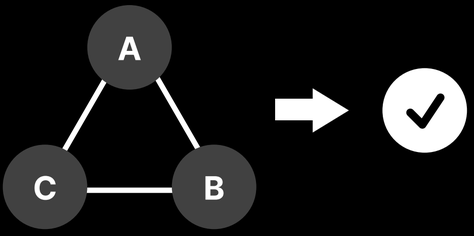
3. Orchestrator-Led

Central coordinator delegates and collects.



4. Peer Review & Consensus

Agents critique, debate, and converge.



Every Code Change Has Three Layers Of Governance.

Keeping agents on track, catching when they fall, and keeping humans in the loop.

1. Mechanical Enforcement

Git hooks block non-compliant changes before they're saved and verify structure after every commit. No agent can bypass them — they run unconditionally on every action.

2. Agent Peer Review

Independent specialist agents — Critic, QA, Security, Reviewer — verify work before the operator ever sees it. Each brings different expertise; none can override the others.

3. Human Checkpoints

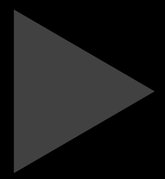
Deliberate human decision points built into the phase sequence. The operator approves the plan, reviews the output, and signs off — at defined moments, not ad hoc.

Contracts Prevent Agent Drift, And Keep Sessions Honest.

Every build session starts with a contract.
The agent declares its scope - then the system enforces it.

Contracts Declare Scope And Boundaries.

Contracts define session type, scope, and objectives are locked before the agent writes a line of code.



Phases Keep Sessions Aligned With contracts.

Phase transitions enforce contract alignment. Out-of-scope work is blocked before it reaches a commit.

Repo OS Is Designed to Make AI-Native Development:

Consistent

Decisions persist across sessions, enforced mechanically. The 10th session follows the same patterns as the 1st.

Cohesive

A Critic reviews every diff for internal coherence. An advisory board reviews specs before code begins. Nothing ships unchecked.

Auditable

Every commit carries session metadata, gate results, and decision references. Override usage is logged and summarized.

Scope Controlled

Task contracts define intent. Scope classification catches drift at commit time. The Pragmatist flags over-engineering before it starts.

Continuous

Session logs, decision records, and task group contracts carry context forward. Nothing is lost between sessions.

Reliable

12 quality gates across 3 tiers mechanically enforce standards before code reaches main. Zero secrets, zero shortcuts.

Interested in testing the model?

Send me a message on Linkedin or reach out via email.

Mark Preston
AI-Native Product Design Leader

n'co studio