AI Coding Agent Plan Review Gate Checklist

Brand: OperatorForge AI
Updated: 2026-06-15
Format: free checklist + plan-risk rubric + approval template
Use when: a founder, engineering lead, consultant, or agency asks Claude Code, Codex, Cursor, Copilot, OpenCode, Devin-style tools, or background coding agents to write an implementation plan before touching code.

Why this matters now

The 2026-06-15 scan found a fresh cluster around plan-phase review and sandboxed execution for AI coding agents:

- Hacker News surfaced "Batta - plan-phase security reviews for AI coding agents (OSS)" on 2026-06-15.
- HN also surfaced "What are you looking for when reviewing LLM generated code?", "Sandbox AI coding agents with microVMs on Fedora Linux", "Kster.ai - Structured product context your coding agent reads over MCP", "Agents as Code", and ongoing Claude/Codex workflow discussion.
- GitHub search found fresh repositories updated on 2026-06-14/15 for per-project microVM sandboxes, agent kernels, sandboxed AWS query runtimes, and governance layers for AI-native projects.
- Reddit JSON checks remain blocked or fragile for this cron profile, so public HN/GitHub sources drove selection.

The monetizable gap: teams are letting agents execute plans that have never been reviewed as plans. The cheapest place to prevent security, scope, cost, and rollback mistakes is before the first file edit.

The 10-minute plan review rule

Before an AI coding agent executes, its plan must be reviewed against explicit risk gates by a human owner. A plan that cannot name scope, files, tests, secrets, permissions, rollback, and evidence is not ready to run.

AI Coding Agent Plan Review Gate Checklist

1. Force the agent to produce a reviewable plan

- Require a one-paragraph objective tied to a ticket, bug, or business outcome.
- Require explicit non-goals: what the agent must not change.
- Require expected files, directories, services, APIs, and config surfaces.
- Require the exact commands the agent expects to run.
- Require acceptance checks before implementation starts.

2. Score plan risk before execution

Assign one risk tier before approving the plan:

- Tier 0 - read-only: inspection, docs, local analysis, no writes.
- Tier 1 - local reversible: small code/docs changes, no credentials, no deploy.
- Tier 2 - production-adjacent: CI, database migrations, auth, billing, emails, prompts, MCP config, customer data, or workflow automation.
- Tier 3 - live authority: deployment, payments, credentials, destructive commands, outbound messaging, or irreversible data changes.

Approval rule: Tier 2 requires named human approval; Tier 3 requires human approval plus rollback proof and bounded execution.

3. Gate sensitive paths and permissions

- Does the plan touch auth, authorization, billing, payments, refunds, email delivery, production deploy, database migrations, secrets, logging, telemetry, CI/CD, prompts, AGENTS.md, CLAUDE.md, Cursor rules, MCP config, or automation workflows?
- If yes, name the reviewer and approval criteria before running.
- Confirm the agent does not need production credentials for the task.
- Confirm the agent is not allowed to deploy, merge, purchase, message customers, or alter payment settings unless explicitly approved.

4. Require a command and tool policy