

AI-Generated Code Audit Checklist

Review a defined change or module and record evidence that another engineer can check.

Understand the change

1. Record the scope, owner, affected workflows, dependencies, and intended business behavior.
2. Read changed code and relevant surrounding paths. Verify assumptions and identify areas nobody can explain.
3. Check dependency names, provenance, versions, and maintenance status. Investigate unfamiliar packages.

Check behavior and boundaries

1. Test normal inputs, invalid inputs, empty data, concurrent requests, retries, and partial failures where relevant.
2. Trace important data through storage and integrations. Check duplicate handling, transactions, migrations, and recovery.
3. Test authentication and object-level authorization. Inspect queries, uploads, secrets, logs, and error responses.
4. For model-connected features, test task correctness, prompt injection, output handling, tool permissions, and escalation.

Make a release decision

1. Record findings by impact and supporting evidence. Assign an owner and a verification step to each required repair.
2. Check monitoring, restore, rollback, and documentation. Have another qualified person review consequential changes.
3. Do not use the number of checked boxes as a substitute for judgment. One unresolved severe issue can be enough to stop release.

Module / change: _____. Findings and evidence: _____. Reviewer / decision: _____.