CHECKLIST

Claude Code Review Checklist

Ten checks to run before you approve a change written with Claude Code: scope, evidence, tests, root cause, dependencies, permissions and configuration, security, the real diff, and whether you can explain it.

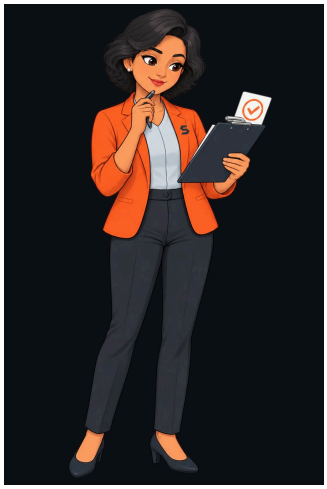


#	✓	Check	What to look for
1	☐	**Scope**	Every changed file belongs to the task you asked for. Read the file list before any code.
2	☐	**No drive-by changes**	No unrelated refactors, renames or formatting churn that make the change harder to review and revert.
3	☐	**Evidence, not reassurance**	You've seen the actual test, build or lint output, not just a statement that it passed.
4	☐	**Tests weren't weakened**	No tests deleted, skipped, loosened or rewritten to expect the wrong value.
5	☐	**Root cause**	The fix addresses the cause, not just the symptom or the error message.
6	☐	**Dependencies**	Any new package, version change or lockfile change was a deliberate choice: maintained, licensed acceptably, not already covered by an existing library.
7	☐	**Permissions and configuration**	No unexpected changes to `.claude/settings.json`, `CLAUDE.md`, `.mcp.json`, CI workflows, deployment config or anything handling secrets.
8	☐	**Security**	Untrusted input is traced to where it's used; no string-built queries or commands; no keys or tokens in the diff.
9	☐	**The real diff**	You reviewed `git diff`, including changes made by shell commands, scripts or generators, not only the agent's summary.
10	☐	**You can explain it**	You could walk a colleague through every change, and why it's safe, without asking the agent.

How to use it

Run checks 1 and 2 before reading any code, 3 to 5 while reading the tests and logic, 6 to 8 on the files listed, and 9 and 10 last. Paste the pull-request template on page 3 into your repository so it appears on every agent-written pull request.

Why each check matters



Scope comes first (checks 1–2)

Agents often tidy nearby code, rename things or reformat files they opened. Anthropic's own suggested review prompt asks a reviewer to confirm that "nothing outside the task's scope changed."[1] For multi-file work, plan mode lets Claude explore and propose a plan before it edits anything, so you can compare the diff against the plan.[1]

Evidence beats reassurance (checks 3–5)

Anthropic's guidance is to give Claude a check it can run, such as tests, a build or a linter, and to have it "show evidence rather than asserting success."[1] Its example prompts also ask Claude to "address the root cause, don't suppress the error."[1]

Dependencies are decisions (check 6)

If you don't want new packages, say so in the task. Anthropic's own example prompt reads: "build from scratch without libraries other than the ones already used in the codebase."[1]

Configuration changes what happens next time (check 7)

Anthropic's security guidance says you're responsible for reviewing proposed code and commands before approval, and recommends that you "verify proposed changes to critical files."[2] It also notes that reviewing `.mcp.json` doesn't show every MCP server a session can load, because servers can also be configured outside the repository.[2]

Security still needs a person (check 8)

45%
of AI-generated code samples failed security tests and introduced OWASP Top 10 vulnerabilities[4]

72%
security failure rate for Java, the worst of the four languages tested (Python 38%, JavaScript 43%, C# 45%)[4]

86%
of relevant samples failed to defend against cross-site scripting (CWE-80)[4]

Veracode, 2025 GenAI Code Security Report (30 July 2025), testing over 100 large language models across Java, Python, C# and JavaScript.

The summary isn't the change (check 9)

Claude Code's checkpoints "only track changes made through Claude's file editing tools," so edits made by Bash commands or external processes aren't captured, and Anthropic notes checkpoints aren't a replacement for git.[1] Review the version-controlled diff.

Tools that help

- **A fresh-context reviewer.** Anthropic recommends having a subagent review the diff in a fresh context, or running the bundled `/code-review` skill. It also warns that a reviewer asked to find gaps will usually report some, so ask it to flag only issues that affect correctness or the stated requirements.[1]
- `/security-review` runs an on-demand security pass over the changes on your current branch.[2]
- **Hooks for non-negotiables.** Anthropic describes hooks as deterministic, unlike CLAUDE.md instructions, which are advisory, so use them for checks that must run every time.[1]
- **Permissions decided in advance.** Allowlists, permission modes and sandboxing limit what the agent can do before review is needed.[3]

Copy into your pull request template

Save as `.github/pull_request_template.md` (GitHub) or your platform's equivalent. An editable copy ships with this checklist.

```
## Agent-written change review
- [ ] Every changed file is in scope for this task
- [ ] No unrelated refactors, renames or formatting churn
- [ ] I saw test/build/lint output (not just "tests pass")
- [ ] No tests were deleted, skipped or loosened
- [ ] The fix addresses the root cause
- [ ] New or changed dependencies were intended
- [ ] No unexpected changes to .claude/, CLAUDE.md, .mcp.json, CI or secrets
- [ ] Untrusted input is handled; no keys or tokens in the diff
- [ ] I reviewed git diff, including shell-made changes
- [ ] I can explain every change without asking the agent
```

STHUTHI TECHNOLOGIES

Know who on your team actually does this.

A checklist only works if people apply it under time pressure. Our Claude Code certification ends in a proctored task in a live repository: the engineer directs Claude Code through a scoped change and explains their review of the diff to a human assessor before it merges. Every pass carries a verifiable credential ID.

sthuthi-tech.com/certifications/claude-code

Book a scoping call → Read the checklist online → 90-day rollout playbook →

Client case studies →

SCAN TO BOOK A CALL

Sources

1. Anthropic, "Best practices for Claude Code." https://code.claude.com/docs/en/best-practices
2. Anthropic, "Security" (Claude Code documentation). https://code.claude.com/docs/en/security
3. Anthropic, "Configure permissions" (Claude Code documentation). https://code.claude.com/docs/en/permissions
4. Veracode, "We Asked 100+ AI Models to Write Code. Here's How Many Failed Security Tests" (2025 GenAI Code Security Report, 30 July 2025). https://www.veracode.com/blog/genai-code-security-report/

Sources checked 6 October 2026; Claude Code changes often, so confirm against Anthropic's current docs. Sthuthi Technologies is independent and not affiliated with Anthropic. Claude and Claude Code are trademarks of Anthropic. © 2026 Sthuthi Technologies.