

AI code without *tech debt* — the checklist

Vibe-coding gets you to 80%. This checklist covers the last 20% — so AI code becomes a **product**, not technical debt. Run it **before** you merge AI-generated code. The **prompt boxes** are copy-paste ready.

— Nicolas Schmid, AI Systems Architect · halvic.ch

01 Before you prompt

- ☐ I **specified precisely** what to build — not just "build X".
- ☐ Inputs, edge cases & expected behaviour are named (null, empty, huge, invalid).
- ☐ I know **how the result fits the system** — which layer, which pattern.

Prompt

"Before writing any code: list the spec, inputs, edge cases (null/empty/huge/invalid) and expected behaviour. Then ask me 3 questions about anything unclear."

03 Tests as a contract

- ☐ A **test that proves the behaviour** — not just "it runs".
- ☐ Critical tests written **myself or separately** — not the same agent, same run.
- ☐ **Failure cases and limits** tested, not just the happy path.

Prompt

"Write a test that proves the required behaviour — including failure cases and limits, not just the happy path. Don't mock the core logic."

05 Security & data

- ☐ **No secrets** in the code/repo; inputs validated.
- ☐ No **silent security hole** that never shows up in the demo.
- ☐ **Data flow & storage** are clear — where it lives, who has access.

Prompt

"Audit this code for hardcoded secrets, missing input validation and silent security holes. Cite specific lines and a fix for each."

02 Interrogate the agent

- ☐ What **assumptions** did the agent make? Asked explicitly.
- ☐ What happens on the **edge case** nobody prompted for?
- ☐ Do I understand every non-trivial line — even in 6 months?
- ☐ No **invented or outdated dependencies**.

Prompt

"List every assumption you made and flag the riskiest. Then name 5 edge cases that would break this code, and what happens in each."

04 Architecture & system fit

- ☐ No **logic duplicated** that already exists.
- ☐ No **layer boundary broken**, no third pattern for something with two already.
- ☐ Stays **maintainable and extensible** — not just "done".

Prompt

"Is there already code or a pattern for this logic in the project? Show me where, and reuse it instead of building something new."

06 Before merge / ship

- ☐ **Reviewed like human code** — not "it was just AI".
- ☐ Runs under **real load**, not just the demo dataset.
- ☐ **Logging/monitoring** for when it breaks in production.

Prompt

"Review this diff like a strict senior: name the 3 most likely bugs and one question that exposes each of them."

★ Does it hold — and does it sell?

- ☐ Does it solve a **real customer problem** — or is it just technically impressive?
- ☐ Does **anyone but me** know it exists? (Marketing is planned.)

The first 80% is free. The last 20% is what you get paid for.

More: halvic.ch/blog · Tickets → shipped, tested code: anvil-coder.tech · Audiobooks: nicolasschmid.ch