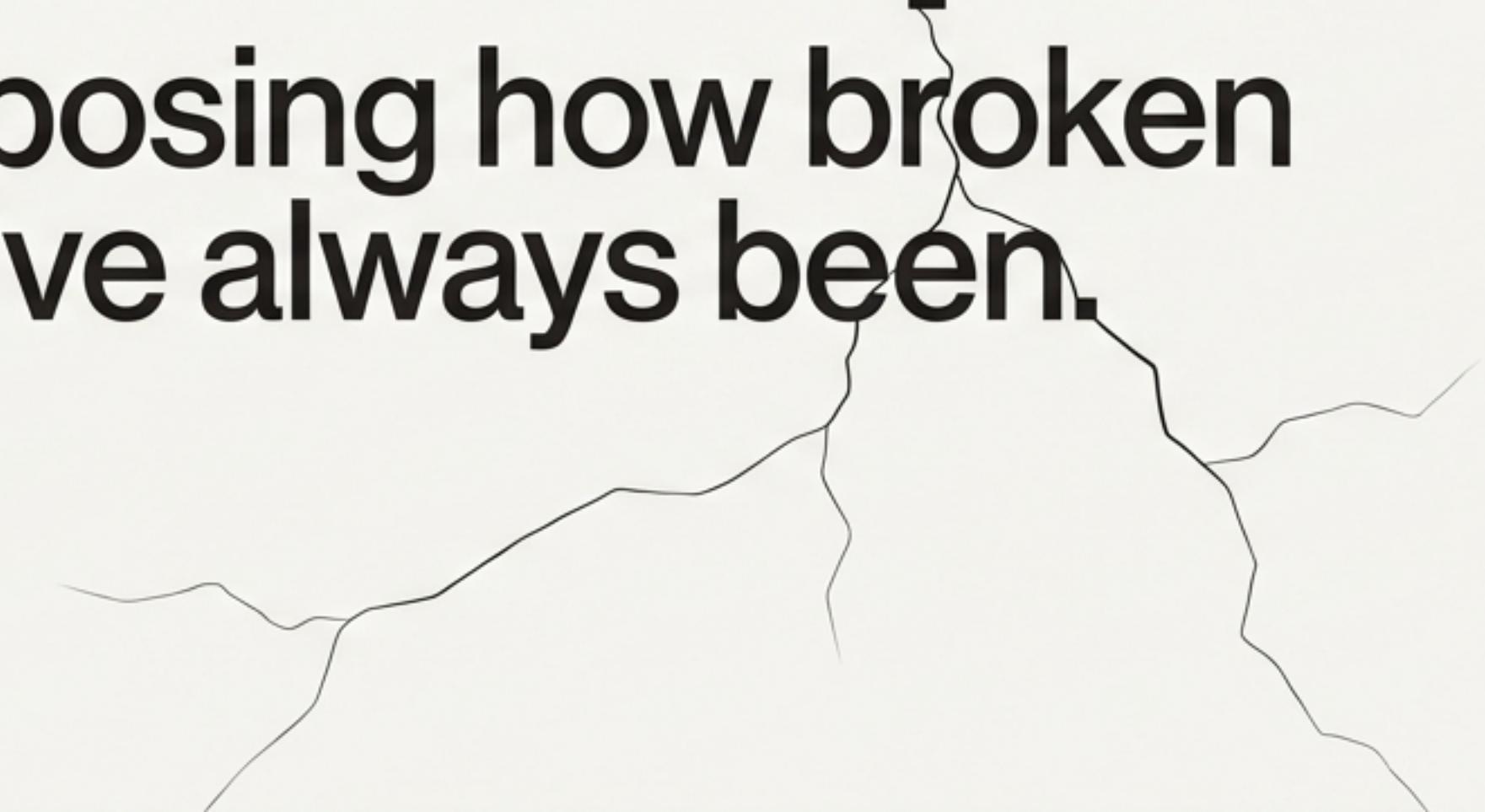


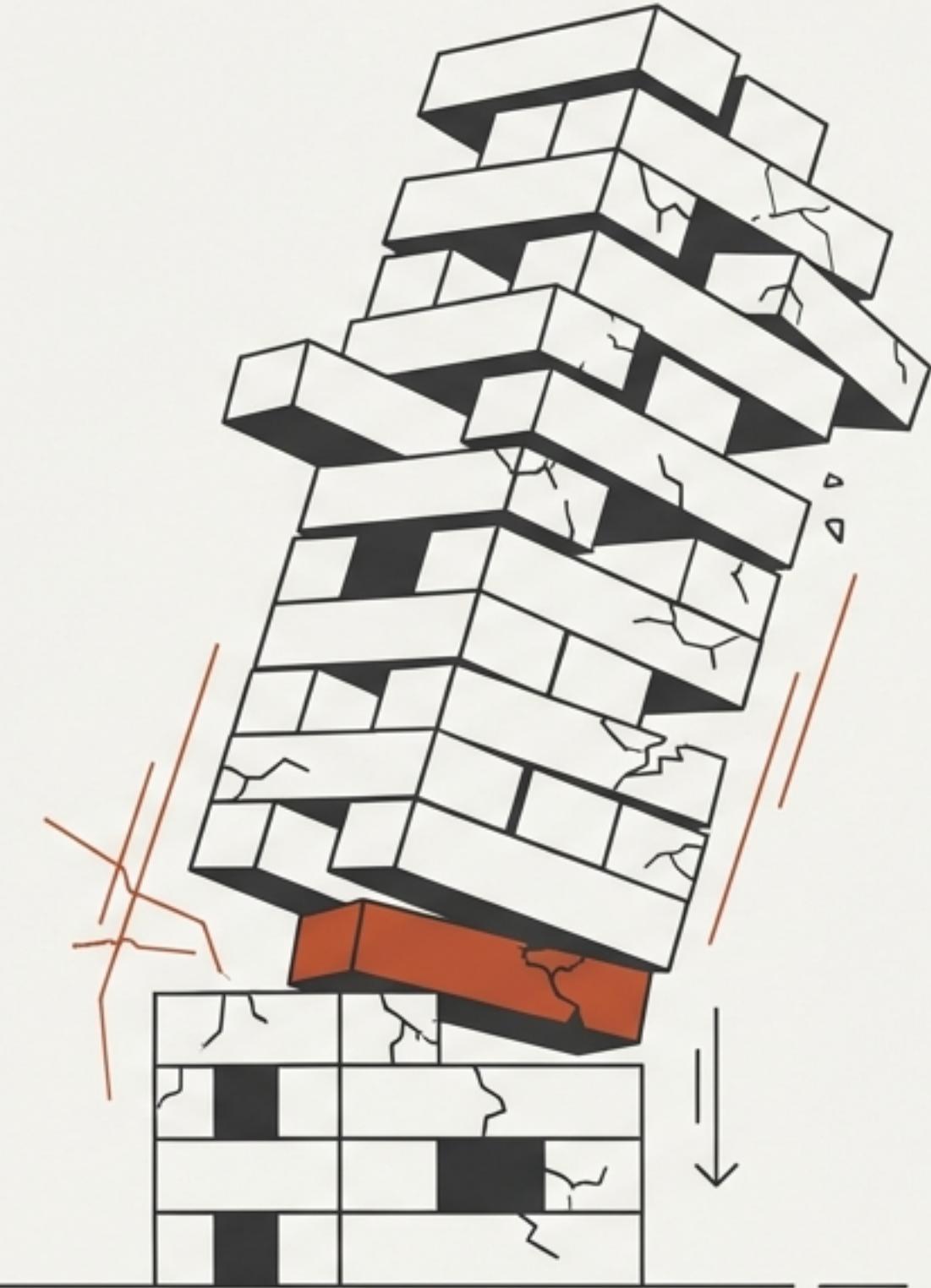
**GenAI isn't breaking
software development.**

**It's exposing how broken
we've always been.**



We've been hacking, not engineering.

- Reliant on individual heroics and tribal knowledge.
- Prioritizes shipping features over sustainable systems.
- Riddled with technical debt, minimal documentation, and flaky tests.
- Rewarding code quantity over architectural quality.



The trap of “more code.”

More code is not the same as better software. Unchecked AI-generated code that nobody understands is an accelerated journey into a maintenance nightmare.

This leads to “vibe coding”: slapping together AI suggestions without architectural understanding.



1 7X

More issues on average in AI-produced
pull requests compared to human code.

“The biggest weaknesses were in code quality and readability, which are exactly the issues that 'slow teams down and compound into long-term technical debt'.”

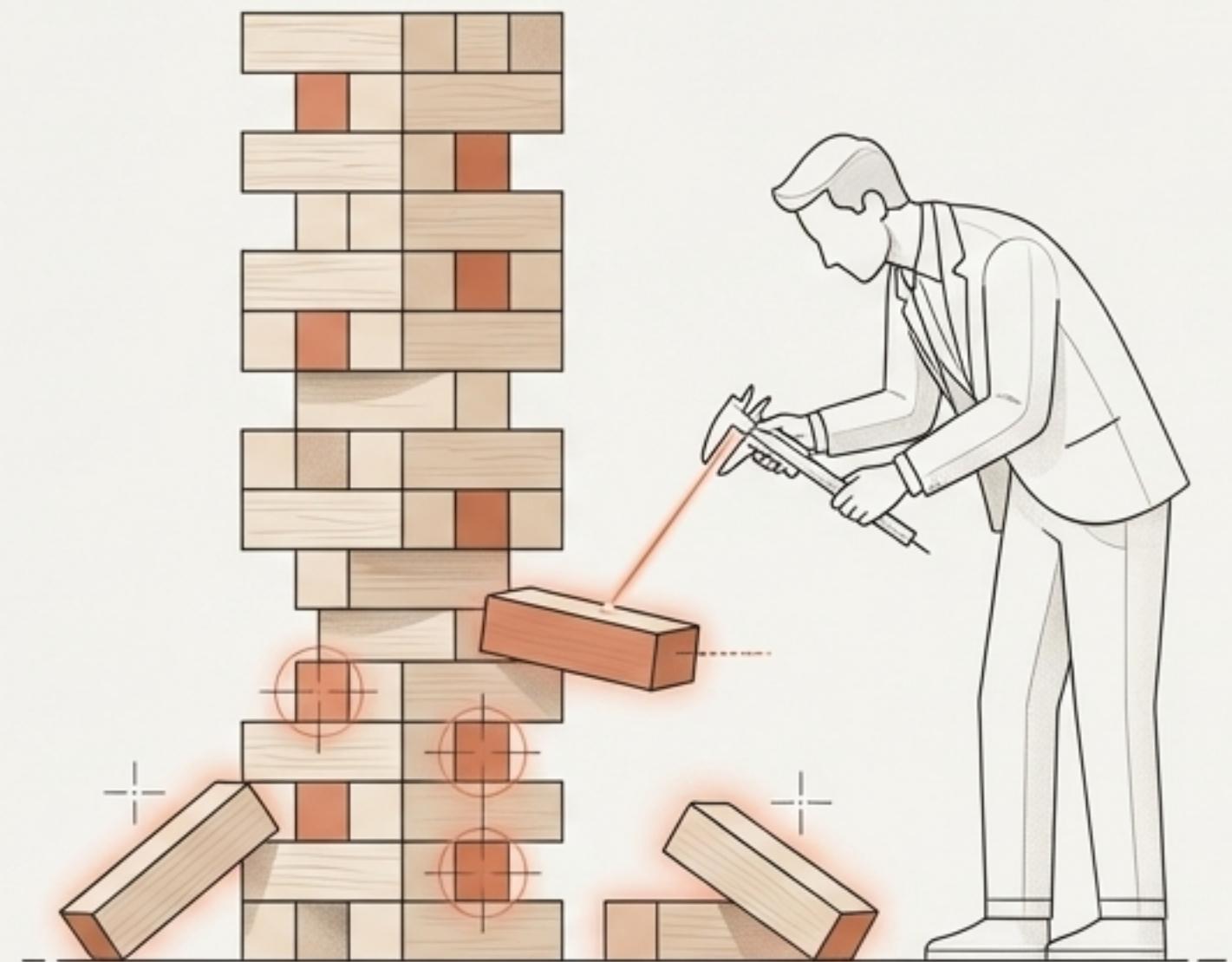
8X

The increase in duplicated code when
AI tools are used without supervision.

This is the price of playing “tech debt roulette.”

We must use AI to create better code, not just more code.

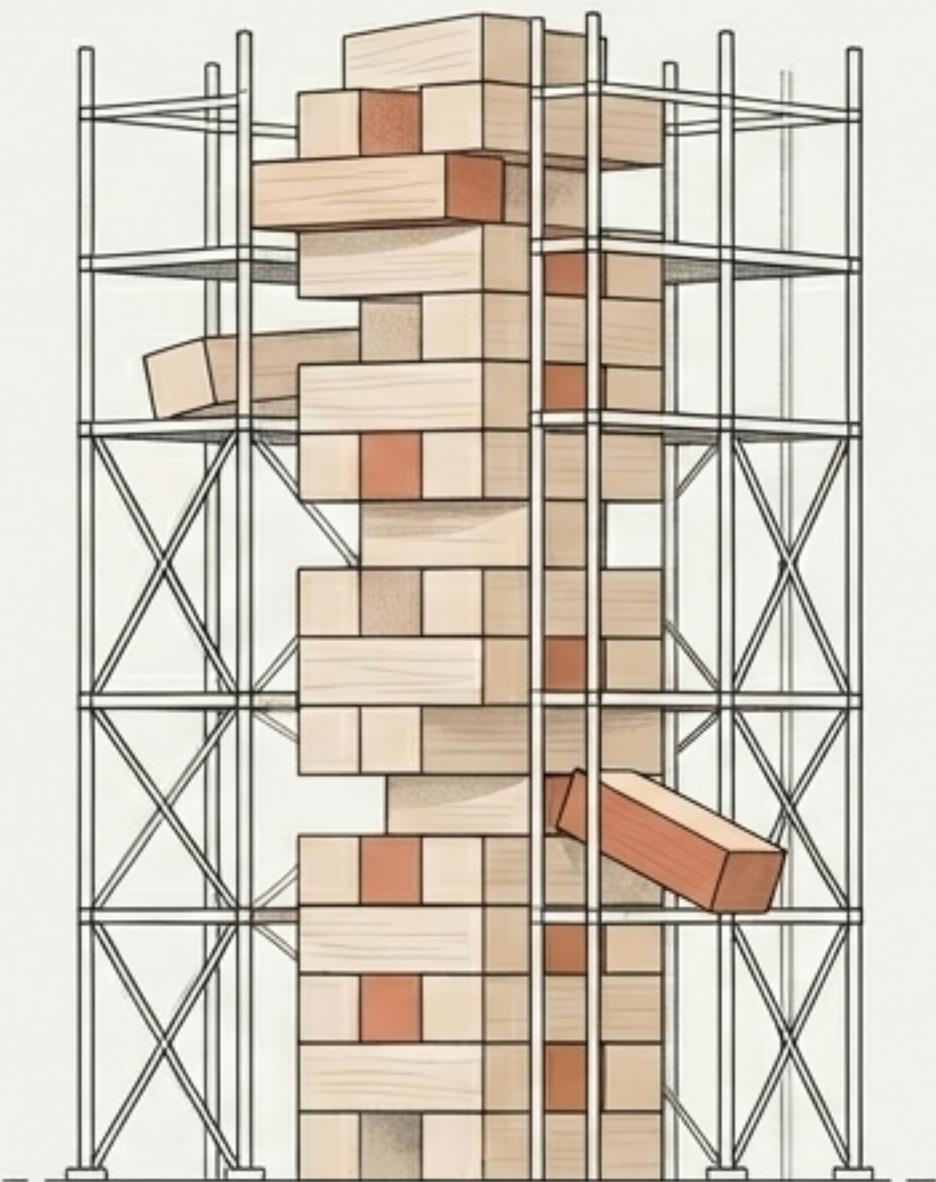
- Simplicity:** Enforce a strict rule: the team must understand every line of code. Use AI to *simplify aggressively* and refactor for clarity.
- Clarity:** Ask the question for every AI suggestion: "*Is this the simplest approach? Could this be clearer?*"



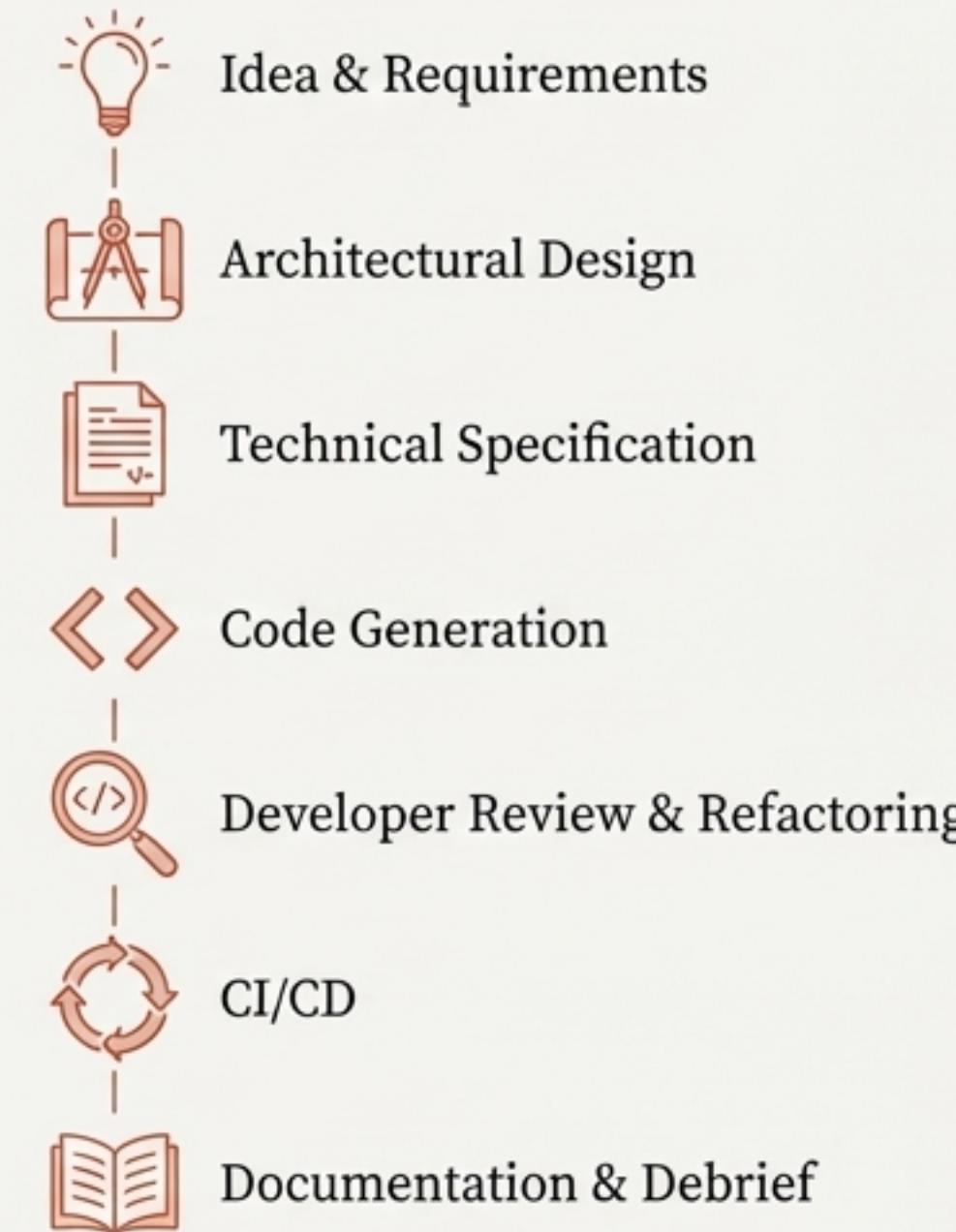
Testing is the backbone of AI-augmented development.

- Provides the safety net to incorporate AI code confidently.
- Grants the freedom to refactor relentlessly without fear.
- Acts as living documentation of intended behavior.
- A robust test suite + GenAI is the recipe for success.

“If the AI writes code, it also better help help write the tests for that code.”



The AI-Augmented Workflow: A Blueprint for Discipline



Harnessing AI for acceleration while maintaining human oversight at every step.

Stage 1-3: Design Before You Code.

- **Ideation:** Transcribe and structure ideas from voice memos or informal notes. AI as a brainstorming partner.
- **Architecture:** Use LLMs as a ‘rubber duck’ to discuss and refine system design—*explicitly generating no code yet*.
- **Specification:** Generate a detailed technical spec, including rationale and ASCII art diagrams, to minimize ambiguity. This becomes the implementation plan.



Stage 4-5: Generate, Then Scrutinize.

- **Code Generation:** Feed the detailed spec to a fresh LLM session. The AI produces a first draft implementation at high speed.
- **Developer Review:** This is the most critical step. Manually review and integrate all code. Run tests. Fix bugs.
- **Refactoring Loop:** If any code is complex or unclear, use the AI to refactor it. ‘Refactor this function into smaller ones with self-explanatory names.’ The goal: every line is understood and meets the quality bar.



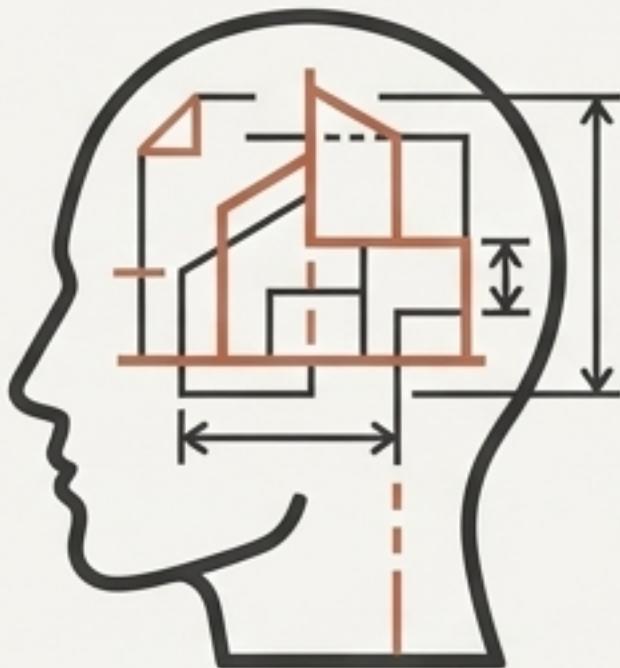
Stage 6-7: Deploy and Document.

- **CI/CD:** Merged code is continuously deployed to a staging environment, ensuring it works in a realistic setting. Deployment becomes a non-event.
- **Documentation:** Use the LLM to generate a full suite of documentation from the final code:
 - An updated design spec reflecting the final implementation.
 - Developer-oriented guides.
 - End-user manuals with examples.
- Key benefit: AI drastically reduces the drudgery that leads teams to skip writing documentation.



New Roles for a New Era

The GenAI Architect



The strategist. Focuses on system design, enforces quality, and steers the AI to produce desired outcomes.

The GenAI Developer



The implementer and reviewer. Treats AI output like the work of a junior dev—something to be critically reviewed, tested, and improved.

Success requires seasoned engineers who guide AI tools strategically, not blindly.

“AI amplifies your technical posture, good or bad.”



Poor practices + AI =
Digging a deeper hole, faster.



Solid engineering + AI =
Boosted productivity and quality.

This is how we turn a broken craft into an engineered discipline.



- GenAI doesn't replace human judgment; it elevates it. The role shifts from typist to architect.
- It frees us from grunt work (boilerplate, docs) to focus on creativity and design.
- By pairing GenAI's speed with human discipline, we achieve a new level of engineering excellence.

GenAI isn't breaking software development—it's finally giving us the tools to fix it.