

More Than Coding: The /mtc Workflow

The Dream: Building a complete game in 2 hours with AI. Zero handwritten code.

The Inspiration: Trae IDE's full-stack agent mode, where an AI developer built a game through rigorous pre-planning.

The Adaptation: Replicating this discipline as a custom /mtc skill within Claude Code.

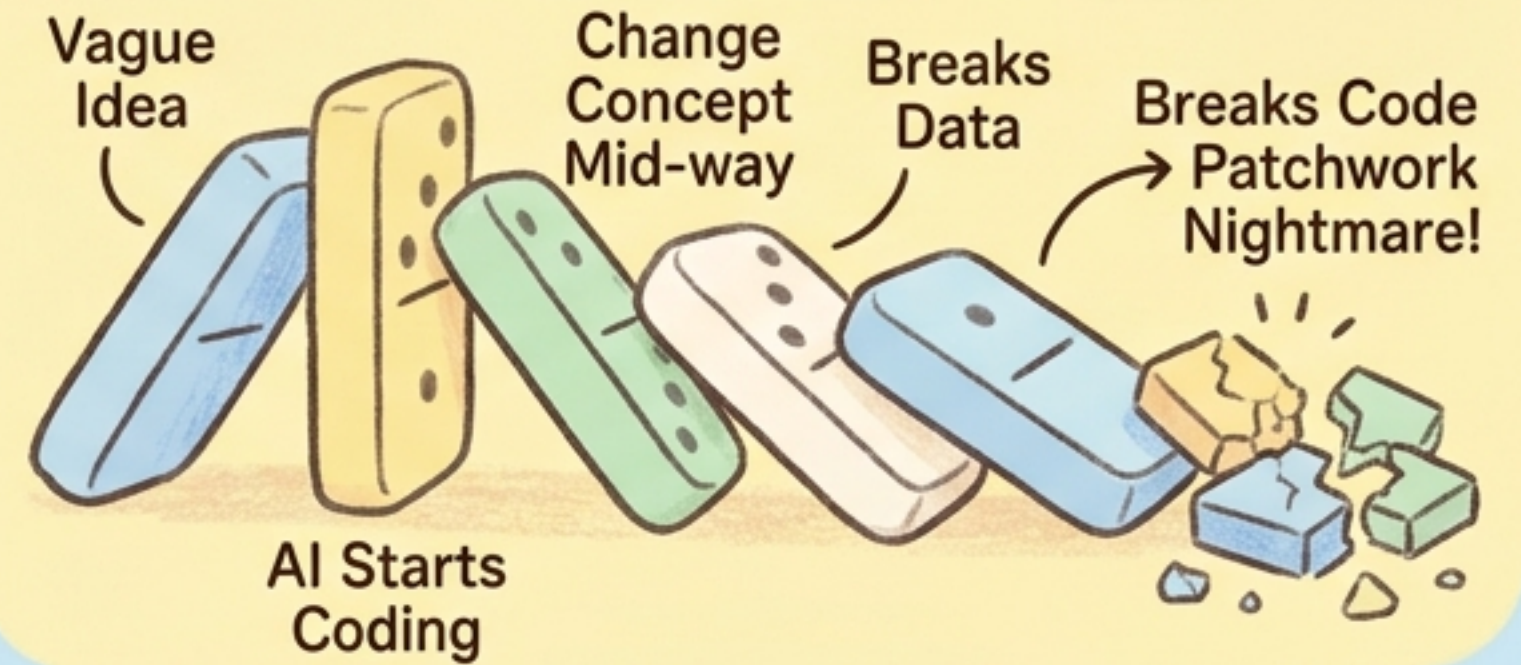
The Philosophy: "Everything before the code." Handling design, analysis, and docs completely before typing a single line.



Why "Vibe Coding" Fails



The Cascading Failure

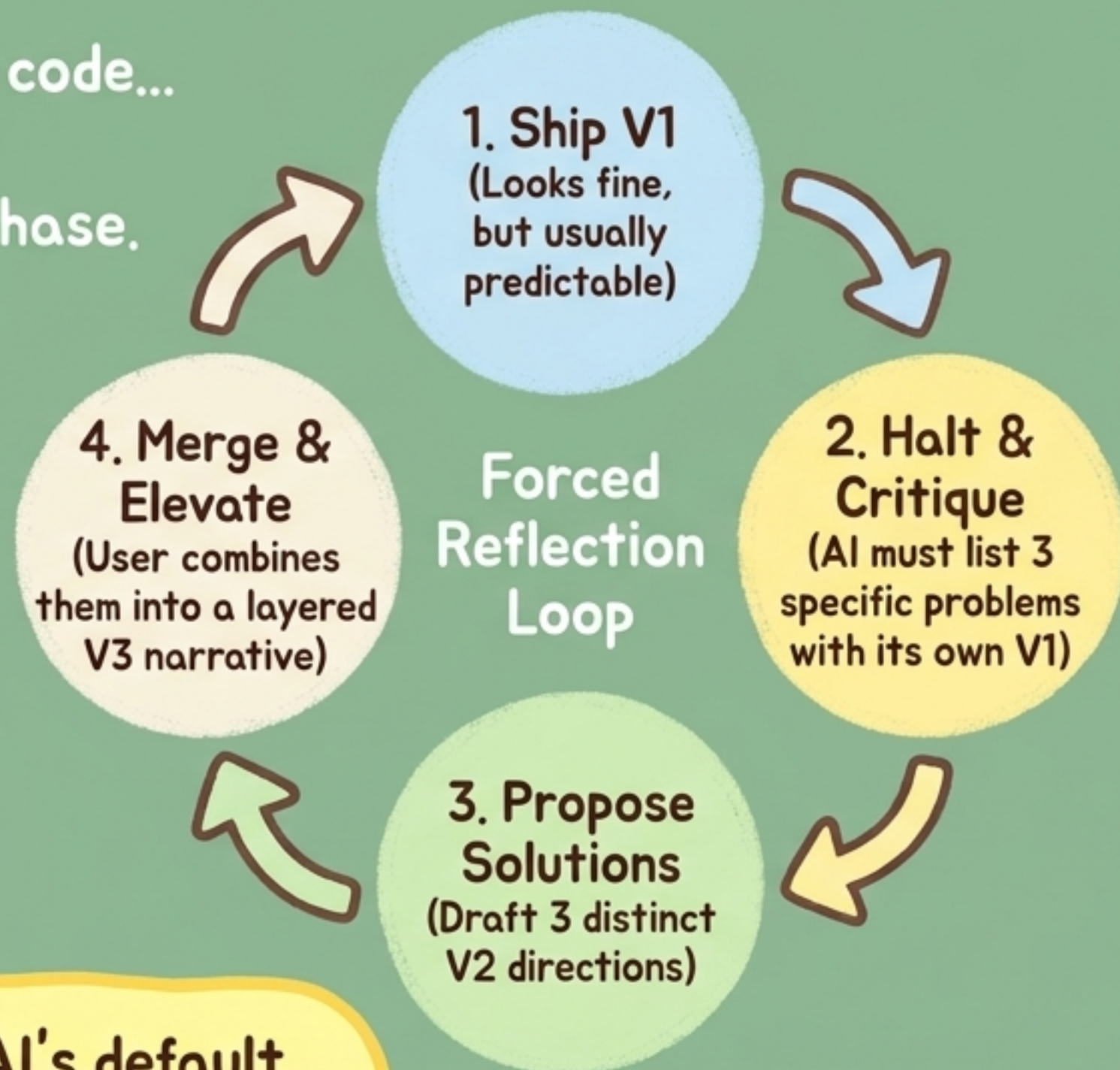


Diagnostic Matrix

	Vibe Coding vs.		MTC Workflow
Pacing:	Start typing immediately	vs.	6 Strict Phases (No skipping).
Process:	Make it up as you go	vs.	Each phase outputs a feeder document.
Memory:	You forget original decisions	vs.	Docs preserve the exact 'why'.
Rule of Thumb:	1 hour of debugging rework	vs.	10 minutes of concept planning.

Phase 1: The 'Critique First' Rule

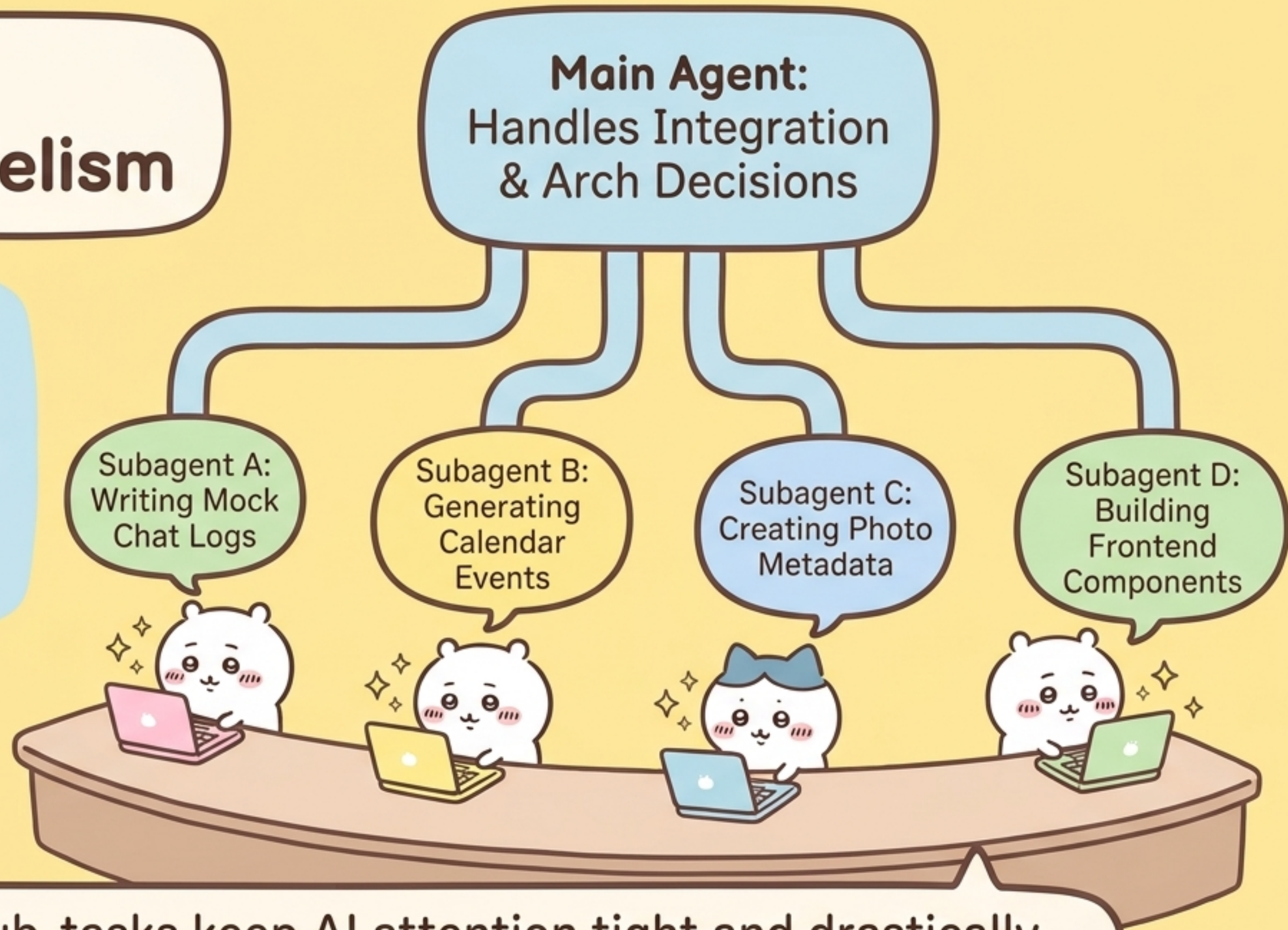
If you let AI reflect on concepts while writing code...
it won't reflect. It will just keep typing.
Concept needs a dedicated, single-threaded phase.



Self-critique isn't AI's default behavior. The workflow forces it.

Phases 3 & 4: Subagent Parallelism

While Phase 1 (Concept) requires a single-threaded global view, Phase 3 (Data) and Phase 4 (Build) thrive on parallel speed.



Independent sub-tasks keep AI attention tight and drastically speed up generation without losing context.

Phase 5: Polish & The Final ROI

Phase 5 Constraints:

- ☒ Add Background Music & SFX
- ☒ Add UI Animations
- ☒ Hide Easter Eggs

Strict Rule: NO new features at this stage! Only texture.



Synthesis Matrix

The Cost: 30–60 minutes of strict pre-work (Docs, Concept, System Design).

The Savings: 2–3 hours of debugging and structural rework entirely bypassed.

The Quality Shift: Transitioning from writing code in a 'local context' -> Writing code guided by a holistic, documented design.

Not a silver bullet for tiny tweaks, but an absolute game-changer for full projects.