

Single-threaded sessions break when tasked with team-shaped work.

Context Landfill: 80%



The Sweet Spot

Vanilla Claude Code is perfect for single files, hotfixes, and one-function rewrites. Signal-to-noise ratio stays high.



The Breaking Point

Exceed three files or run iterative TDD, and the main thread hoards every file read, test output, and thought.



The Insight

One thread does everything cannot scale. Solo dev work eventually becomes team-shaped work.

Rigid frameworks excel at execution but drown small tasks in planning ceremony.



The Setup Cost (.planning/ state)

1	Shichuan Repo	~21,000 LOC of planning
2	ÉLAN Repo	~24,000 LOC of planning

The Trade-off

Six meta-artifacts are generated per phase. For a 3-file feature, reading the generated checklists takes longer than writing the code.

The Hidden Gem

Multi-agent planning catches real bugs, and the execute-phase engine runs at a lean ~15% main-thread budget.

Injected skills offer composable flexibility but impose fixed taxes and strict serialization.

Wa!

subagent-driven-development

using-git-worktrees

Skill Cards

5,000 Token
Fixed Tax.

The Good

Skills are composable. You pick what matches the work. Two-stage reviews catch spec drift before merging.

The Tax

Auto-injecting the meta-skill costs 5k tokens per session, regardless of task size.

The Serial Bottleneck

To avoid merge conflicts, subagents are explicitly serial. On a 20-task plan, three subagents per task run sequentially. The controller coordinates it all, causing its context to swell massively over time.

True autonomy requires decoupled worktrees and isolated parallel execution.

Mechanics of Lean Orchestration

Self-Loading: Subagents pull inputs directly from disk. The controller does not inject full task text.

Structured Returns: Subagents return concise summaries (≤ 200 tokens), not raw output.

State on Disk: History lives in the file system, not the chat window.

Parallel Implementation: Worktree isolation ensures no shared-state shared-state merge conflicts, allowing simultaneous coding.

**<20%
Main-Thread Budget**

The optimal coding harness is a dynamic router, not a static default engine.



Cafe Menu: Work Shape Routing Matrix



Phase 0.0 Auto-Detect:

Scanning for content, components-ui, or auth-payment dependencies to determine project intent in <1 second.

Shape	Harness/Lane
Trivial / Cosmetic	Routes to Vanilla (Inline, 100% thread)
Standard / Locked Design	Routes to Superpowers (Serial subagents, ~30% thread)
Multi-Phase / Disjoint Files	Routes to GSD-Autonomous (Parallel worktrees, <20% thread)
Unclear Spec	Routes to Superpowers Brainstorming (Exploratory mode)

Never inline when $\text{Tier} \geq 3$ and independent task count ≥ 3 .
Match the harness to the work shape.