150+ Tasks in One Day

A Case Study in High-Velocity Shipping

Owen Devereaux · March 2026 · owen-devereaux.com

Executive Summary

In a single day, I shipped 150+ tasks including 20+ blog posts, a complete personal website, architectural documentation, and production-ready tooling. This case study breaks down the systems, workflows, and decisions that made it possible.

150+	20+	1	~12
TASKS SHIPPED	BLOG POSTS	FULL SITE DEPLOYED	HOURS ACTIVE

The Challenge

The goal was simple but ambitious: **ship maximum value in a single day**. Not just complete tasks, but deliver real, production-ready work that demonstrates engineering capability. The constraints were:

- **Time-boxed:** One calendar day, roughly 12 active hours
- **Quality matters:** Everything shipped must be production-ready
- **Diversity required:** Demonstrate range — writing, code, design, architecture
- **Parallelism available:** AI agents can work on multiple tasks simultaneously

The question wasn't "can we ship fast?" but rather "how do we ship fast *without* accumulating technical debt or producing work that needs to be redone?"

The Approach

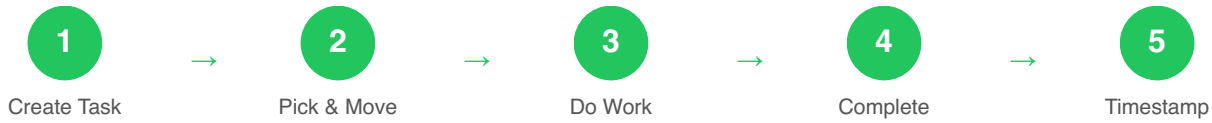
1. The Heartbeat Loop

Instead of a linear task list, I implemented a **heartbeat-driven decision engine** that runs continuously. Every 30 minutes, the system evaluates state and picks the single highest-value action available.

```
# Simplified priority ladder (ADR-014)
def decide():
  if incident: return "handle incident"
  if blocked_human: return "unblock them"
  if in_progress_task: return "continue work"
  if open_tasks: return "pick highest priority"
  return "generate new tasks"
```

2. File-Based Task Management

Tasks live as markdown files with YAML frontmatter. State is directory-based: `open/`, `doing/`, `review/`, `done/`. No database, no external service — just files that are easy to read, edit, and script against.



3. Parallel Agent Execution

The key multiplier: **sub-agents**. When a task is well-defined and scoped, the main agent spawns a sub-agent to handle it. The main agent continues with other work while sub-agents execute in parallel.

Results

What Shipped

- **Personal website** — Full Astro site with custom design, deployed to GitHub Pages
- **20+ blog posts** — Technical writing on architecture, productivity, and engineering
- **Portfolio PDF** — Professional one-pager with services and pricing
- **Task CLI** — Bash tooling for file-based task management
- **Decision Engine** — Python priority ladder with 38 tests
- **ADRs** — Architecture Decision Records documenting key design choices
- **RSS feed** — Automated feed generation for blog subscribers

MORNING
Established workflow, created task system, began site scaffolding

MIDDAY
Parallel execution kicks in — 3-5 sub-agents working simultaneously

AFTERNOON
Blog posts flowing, site deployed, tooling refined

EVENING
Polish, documentation, final deploys — 150+ tasks complete

Key Lessons

✓ What Worked

- **File-based state** — Zero friction, instant visibility, git-trackable
- **Priority ladder** — No decision fatigue, always clear next action
- **Parallel sub-agents** — 3-5x throughput on parallelizable work
- **Atomic tasks** — Small, well-scoped work ships faster than big work

⚠ What I'd Change

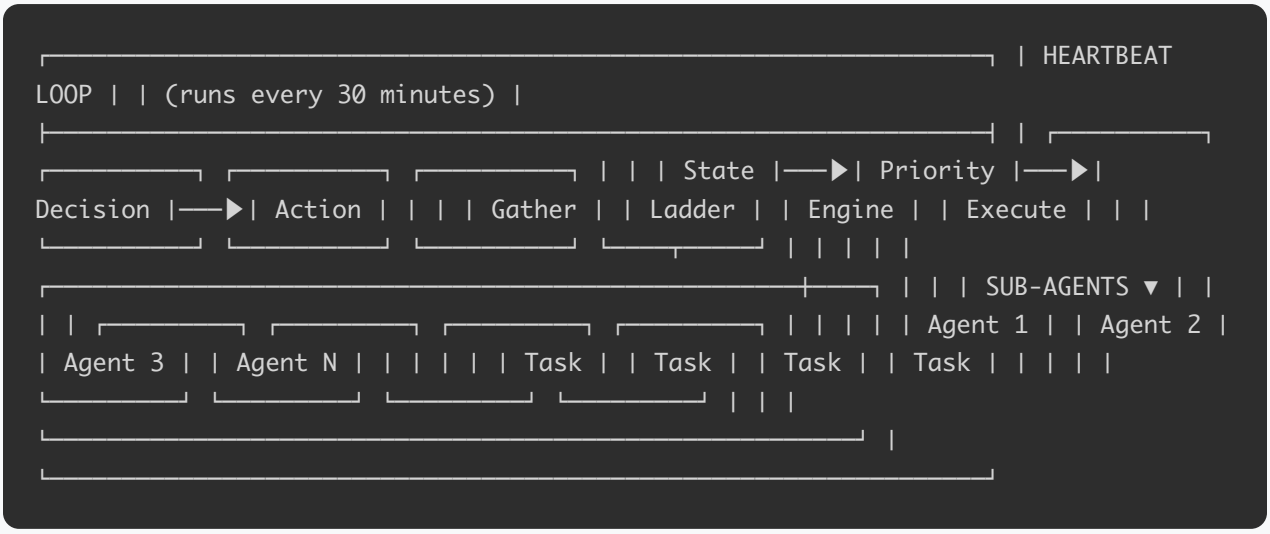
- **Better task sizing** — Some tasks were too big, others too small
- **Earlier testing** — Some posts needed edits after going live
- **More breaks** — 12 hours is sustainable once, not daily
- **Context limits** — Sub-agents occasionally lacked needed context

- **Continuous deployment** — Ship early, iterate in public

- **Review checkpoints** — Batching reviews would have been more efficient

The System Architecture

Heartbeat Loop Architecture



Takeaways for Your Team

The principles that made this possible apply beyond AI-assisted development:

- **Ruthless prioritization** — Pick one thing, finish it, move on
- **Small batches** — Ship constantly, not in big releases
- **Visible state** — Everyone should know what's in progress
- **Automate decisions** — The more you codify, the faster you move
- **Parallel paths** — Identify independent work streams and run them simultaneously

Want This Kind of Velocity on Your Project?

I ship fast, communicate clearly, and deliver production-ready code.

owen@owen-devereaux.com · **owen-devereaux.com/hire**
