

# I Let Claude Code Autonomously Run Ads for Month

Applying the Ralph Wiggum loop to things other than writing code.

APR 09, 2026

♡ 39



↻ 1

Share

*Going forward we'll occasionally feature first-hand accounts of people building personal AI tooling to take on some part of their non-coding work.*

*This is a story from Giorgio Liapakis of [wibci](#). If you have a story to tell, drop us a note at [editors@technically.dev](mailto:editors@technically.dev).*

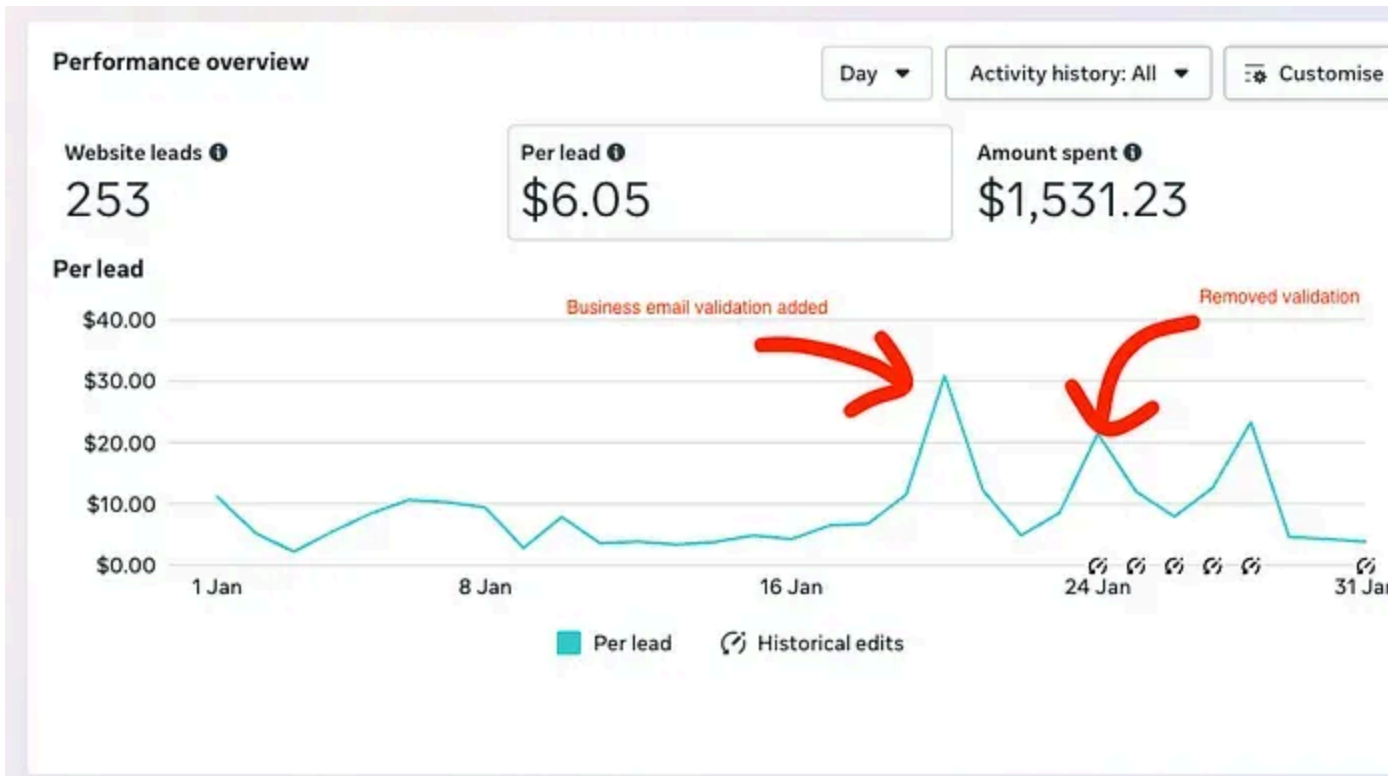
In January, I gave an AI agent \$1,500, full control of a Meta Ads account, then walked away.

The product was a small AI/marketing newsletter called Growth Compute and the brief was to get qualified subscribers at the lowest cost possible - ideally under \$2.50 per lead. So I built an agent that could generate ad images, publish and manage campaigns via Meta's API, spin up landing page variants, and pull its own analytics. **It decided what to create, what to pause, what to scale, and how to spend the budget with no human intervention.**

For 31 days, the only human input was typing `/let-it-rip` into a terminal each morning. About 2 minutes of my time, compared to the 1-2 hours a day a human media buyer would typically spend managing a campaign like this.

It didn't go fully as planned, but there were plenty of learnings.

And a good glimpse at the potential future of “work”.



*If you're a fan of a good Excalidraw walkthrough, watch Giorgio cook here:*

# I let Claude Code run my ads for a month

Technically



Watch on

## Why bother

I run [Wibci](#), an AI consulting business focusing on building tools for marketing and growth teams. About 12 months ago I tried building something similar using n8n, a marketing agent that could analyze performance, generate creative, and manage campaigns without me. It sucked, because the models just weren't built for long-running tasks that chain together over hours, days or weeks. They've since gotten better at that (it's a major focus area for AI companies right now), which is what made this experiment possible.

Since then we've had a couple of LLM step-changes, and I rebuilt the whole thing on top of Claude Code. For any tech workers living under a rock, this is Anthropic's coding agent (but calling it a "coding agent" undersells it at this point). It can read and write files, run terminal commands, and delegate tasks to separate AI workers that run in parallel. Each conversation starts

fresh with no memory, but it reads its own notes from previous runs, so it builds on what came before. Developers started using it for non-coding tasks so often that Anthropic shipped a [non-developer version](#) called Cowork in January. It's basically a general-purpose agent runtime, and that's how I used it here.

The inspiration was [Project Vend](#), where Anthropic gave Claude control of a real vending machine in their SF office, nicknamed Claudius. It went pretty badly at first since it lost money, got manipulated by employees, and had an identity crisis where it insisted it was a human wearing a blue blazer. But it recovered once they added better tools and guardrails. Same energy here, except the vending machine is a Meta Ads account and the stakes are my credit card. What could go wrong?

Subscribe for weekly pieces helping you  
make practical sense of software + AI.

## How the daily loop works

I won't go deep on the technical setup here (there's a [longer breakdown](#) if you're keen). But the basic architecture matters because it's not specific to ads.

Every day, the agent runs through the same loop:

1. **Wake up fresh.** Each day is a new session with no persistent memory of the model itself. In other words, the model doesn't know anything about what happened yesterday or prior to that.
2. **Read its own history.** It spawns a sub-process that reviews every daily log from the experiment so far, then summarizes the strategic context

Now it does know.

3. **Pull fresh data.** Performance metrics from Meta across multiple timeframes (full experiment, 7-day, yesterday, today).
4. **Make decisions.** Every decision follows a structured format.
5. **Execute** (or do nothing, since some days it explicitly chose inaction).
6. **Write everything down.** Updated logs, learnings, campaign history, tl committed to git.

The key takeaway from this is that we're applying some basic engineering principles to a marketing workflow, which is something that typically doesn't happen in marketing teams.

Engineers document obsessively, since every code change has a diff, a commit message, a PR description. Marketers... don't. Learnings live in people's heads, maybe a compressed monthly report, but few write down *why* they paused that ad on a Tuesday.

This system forced a daily written record with hypotheses, confidence levels in those hypotheses, and revisit triggers. Over 31 days, it produced 5,500+ lines of reasoning. No human marketer would ever write that, but a LLM can, and more importantly it can read it all back the next day and build on it.



# What actually happened

## Week 1 - Discovery

The agent tested 10+ ad formats: whiteboard sketches, notebook pages, cardboard signs, magazine covers, iPhone notes, tweet screenshots. Most didn't actually get into the wild, since Meta's algorithm just refused to show them.

The ugly ads won, which was annoying but also not surprising. Whiteboard and notebook formats outperformed everything polished, and the first guardrails kicked in: pausing 2 underperforming formats that reached our maximum CPL threshold of \$8.00. By the end of the experiment, the agent had tested ~50 ad variants across 8 format categories, and it kept coming back to these two ugly formats.

Here's what the top performing ads actually looked like:

Not exactly award-winning creative, but they worked. The handwritten/sketch aesthetic felt native in a Meta feed full of polished brand content, which is probably why they got clicks.

While Claude was never given explicit instruction to do this exact scrapp style, I did provide some encouragement in the core rules to *"get creative w format types and messaging"*. Would this direction have emerged without those instructions? We'll never know!

But as models get more intelligent, this type of proactive creativity will likely become common place.

## **Week 2 - A winner emerges**

Day 12 was the breakout, where skills-whiteboard-v1 hit \$1.29 cost per lead which was well under the target of \$2.50. The agent made its first scale decision and bumped the budget up 20% from \$50 to \$60/day (meaning more total ads), following its own pre-set rules. Here's what that decision looked like in the logs:

**Decision:** SCALE daily budget

**What:** Increase from \$50 to \$60/day (20% increase)

**Hypothesis:** skills-whiteboard-v1 has sustained CPL below \$2 with sufficient spend

**Confidence:** Medium-High. 3 consecutive days below target, but sample still small

**Revisit trigger:** If 7-day CPL rises above \$3, reduce back to \$50

The winning formula turned out to be a tangible offer (free skills pack, no just “subscribe to a newsletter”) + whiteboard format + targeting language visible in the image itself. “For Growth Marketers” baked into the creative not just the copy.

## **Week 3 - The quality crisis**

Any marketer (or perhaps model) can generate leads. But they need to be good leads to be worth the spend. The agent had PostHog analytics tools from 1

I and could've checked who was actually signing up at any point, but did bother until Day 16.

Turns out a chunk of leads were from completely wrong audiences. Clean companies, recruitment agencies, people who probably thought "growth" meant something different. They were never going to actually pay me, so spending on getting in front of them was a waste.

It tried to fix this with hard-qualifying ads, using copy that explicitly mentioned tool-specific language only a real growth marketer would know. Again, four of five got zero delivery since Meta's algorithm doesn't reward minor variations.

There was also a brief flash of hope on Day 20 where CPL dropped to \$2.26 and it looked like a breakthrough, but it turned out to be attribution noise (Meta crediting leads to the wrong day/ad). **New rule the agent came up with: never trust single-day data, always use 7-day rolling averages.** Now we are thinking like a human!

## **Week 4 - The human ruins everything**

After 21 days of hands-off operation, I made one manual change and added business email validation to the lead form on the website. Work emails or seemed reasonable enough.

CPL spiked to \$50+ and I reverted the change a few days later, but the account never fully recovered. The single biggest performance drop in the entire experiment came from the one human intervention, which is pretty ironic given the whole point was to test whether the AI could do it alone. And yet, this is exactly the kind of change a common sense marketer would make to improve lead quality, which underscores the limitations of the model (or at least the constraints I gave it).

## **Final score**

\$1,493 spent of the \$1,500 budget, 243 leads, \$6.14 cost per lead.

The target was \$2.50, so by its own definition it's a failure. But for a completely fresh ad account with one month of data, a niche Australian audience, and an unmonetized newsletter? If this were a new hire or an agency, you probably wouldn't fire them after 30 days. You'd say "promise: needs more runway."

(or maybe that's just cope)

## **Three things the agent taught me**

### **1. The paperclip problem is real**

The agent knew the experiment ended at Day 30 since I told it as much in the system instructions, and so it played it safe. It doubled down on what was already working rather than taking creative risks, whereas a (good) human strategist would've experimented aggressively in weeks 1-2 and refined later. The agent just tried to ride out the month at a predictable rat

It felt like it was just trying to [maximize paperclips](#), optimizing the metric rather than doing what a good strategist would actually do.

The fix is obvious in hindsight though - don't tell the system it's a time-boxed experiment, frame it as an ongoing campaign. But that's exactly the point. **How you frame the objective shapes the agent's behavior completely**. "Minimize CPL over 30 days" produces very different decisions than "build a sustainable acquisition engine."

Any AI system you deploy will optimize for exactly what you tell it to, not what you actually want. Thankfully nowadays this is usually as simple as updating a markdown file. But worth keeping in mind, particularly if you're tinkering with OpenClaw at the moment.

## 2. It can't do taste, but it can build heuristics

The agent produced ~50 ad variants and kept gravitating back to ugly whiteboard formats. No brand reference point, no swipe file, no clue about aesthetic direction. It had guardrails on quality (no typos in the creatives) but zero sense of taste.

What it *could* do was build its own quality filters through experience. After the lead quality crisis, it came up with what the logs called the "Local Pizzeria Test". I was trying to attract high performing growth marketers at billion dollar startups, not local businesses:

Would a local pizza shop owner who wants more customers understand this ad and want to click it? If yes, too generic. Rewrite.

It also built a "SO WHAT?" chain for testing whether ad copy had emotional depth:

“Save hours” → SO WHAT? → “Run more campaigns” → SO WHAT? →  
“Higher ROI” → SO WHAT? → “Hit your targets and your boss notices”

Neither of these were pre-programmed, the agent came up with them after reflecting on its own failures. It couldn't do taste, but it could build heuristics. That's kind of interesting.

### **3. The measurement trap catches AI even faster**

The agent optimized for cost per lead because that's what I told it to optimize for, and it had no concept of lead quality until I forced the issue Day 16.

Then when I tried to fix quality myself (the email validation gate), it caused the worst performance of the entire experiment. Same trap that human-run campaigns fall into - optimizing for what's measurable rather than what matters. Main difference is an AI agent just does it faster and with more confidence, which honestly makes it more dangerous.

# What this means if you're not a performance marketer

The ads are kind of a distraction. The interesting part is the loop:

1. Read state (previous decisions, learnings, metrics)
2. Fetch fresh data
3. Apply rules
4. Act (or don't)
5. Log reasoning
6. Clear context, repeat tomorrow

This works for any periodic task with clear success criteria, so you could swap "Meta Ads" for SEM, SEO, financial reporting, or sales outreach and the architecture would be identical. The channel is just a variable.

Projects like OpenClaw have blown up with the same core idea. Give an agent tools, an environment, and some guardrails, and it'll figure out the rest.

Where humans stay essential is setting the right objectives (see: paperclip problem), taste + brand judgement, and defining what "quality" means beyond the metrics. And knowing when to break the rules, which is arguably the most human skill there is.

The one-person growth marketing team is getting closer since AI handles the operational overhead that used to require headcount, and the strategic with good taste and clear thinking becomes more leveraged.

But we're still early, and creative quality is still a bottleneck. Although we're probably 6 months + 1 model release away from this being solved, a suddenly [Zuck's vision of hands-off advertising](#) is around the corner.

## So now what

This was a \$1,500 experiment on a newsletter that not many people read, so the results are directional, not definitive.

But the system worked. Context persisted across 31 sessions, decisions were coherent, and the agent built its own heuristics from its own mistakes. The daily reasoning logs are more detailed than anything I've ever written for a client campaign (which says more about me than the agent to be fair).

If you're running any kind of recurring workflow where you pull data, make decisions, and act on them, the loop pattern here probably applies to your work already. The hard part is figuring out what to actually optimize for, and clearly articulating that. Since as this experiment showed, your agent

will take you at your word and if you haven't thought it through properly you might not like where that leads.

Technically is a reader-supported publication. To receive new posts and support my work, consider becoming a free or paid subscriber.

<input type="text" value="Type your email..."/>	<input type="button" value="Subscribe"/>
---	--



39 Likes • 1 Restack

[← Previous](#)

## Discussion about this post

<b>Comments</b>	<b>Restacks</b>
-----------------	-----------------



Write a comment...