

How we rebuilt our website with Claude Code - what *actually* happened

HOSTS

Zoran Arsovski · Nikol Anestieva

THU · JUL 23

Community Green · Campus X

Agenda

01 Phase 1 - Main website

Every page of the live site, rebuilt

02 Phase 2 - Content hub

Reusing the pipeline at scale

03 What worked / what didn't

The honest, softened version

04 Practical takeaways

What to do if you're considering this

— LET'S GET TO KNOW EACH OTHER

We'd love to hear a little about *you*

Whenever you're comfortable - your name, and what brought you to this VertoTalks.

→ Tried any coding? We'd love to hear about it :)

— WHY A REBUILD

The old way of shipping a page was slow, and it wasn't ours

CMS COMPLEXITY

Simple updates needed specialist help, and staying secure meant endless plugins and patches.

EXTERNAL PARTNERS

Every new page went out to an agency. Their framework, their process, their queue.

WEEKS PER PAGE

Rounds of back-and-forth to launch one page. A full refresh meant months.

A little groundwork let us move fast

Three foundations had to exist first - and they turned out to be reusable far beyond the website.

01

Brand guidelines

How we speak and show up - the rules everything else is checked against.

02

Positioning document

Who we're for, what we stand for, and the pipeline-first story we tell.

03

Visual identity

Colors, type, spacing, components - the system the site is built from.



The payoff compounded: the same guidelines and visual identity now generate on-brand assets - copy and images - for LinkedIn, VertoTalks, Instagram, Facebook and every other channel.

— THE REAL WORKFLOW

One page template at a time, through three tools

STEP 01

Claude Chat

Wrote the brief for each page - goals, sections, copy direction.



STEP 02

Claude (design)

Turned the brief into a page template - layout and structure.



STEP 03

Claude Code

Built the page in the repo - real, production code.

— ZOOM-IN

What one page actually looked like

1 · BRIEF.md · Claude Chat

/services

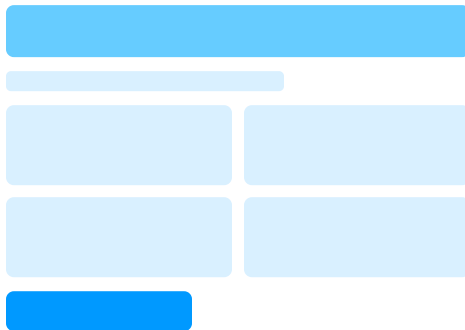
Goal: capture demand,
route to a call.

Sections:

- Hero + eyebrow
- 5 pipeline services
- Proof strip
- CTA

Voice: peer, pipeline-
first, no fluff.

2 · TEMPLATE · Claude
Design



3 · BUILT PAGE · Claude
Code

PIPELINE SERVICES

Turn every channel into a pipeline engine

Book a call →

— INFRASTRUCTURE DECISIONS

Three choices, one stack we fully own

The stack that made us fast, secure, and optimization-ready - page speed, agent visibility, and near-instant staging-to-production deploys.

THE BUILDER

Claude Code

Writes production code directly in the repo, with our brand loaded as skills. No handoff, no translation layer.

THE SOURCE OF TRUTH

GitHub

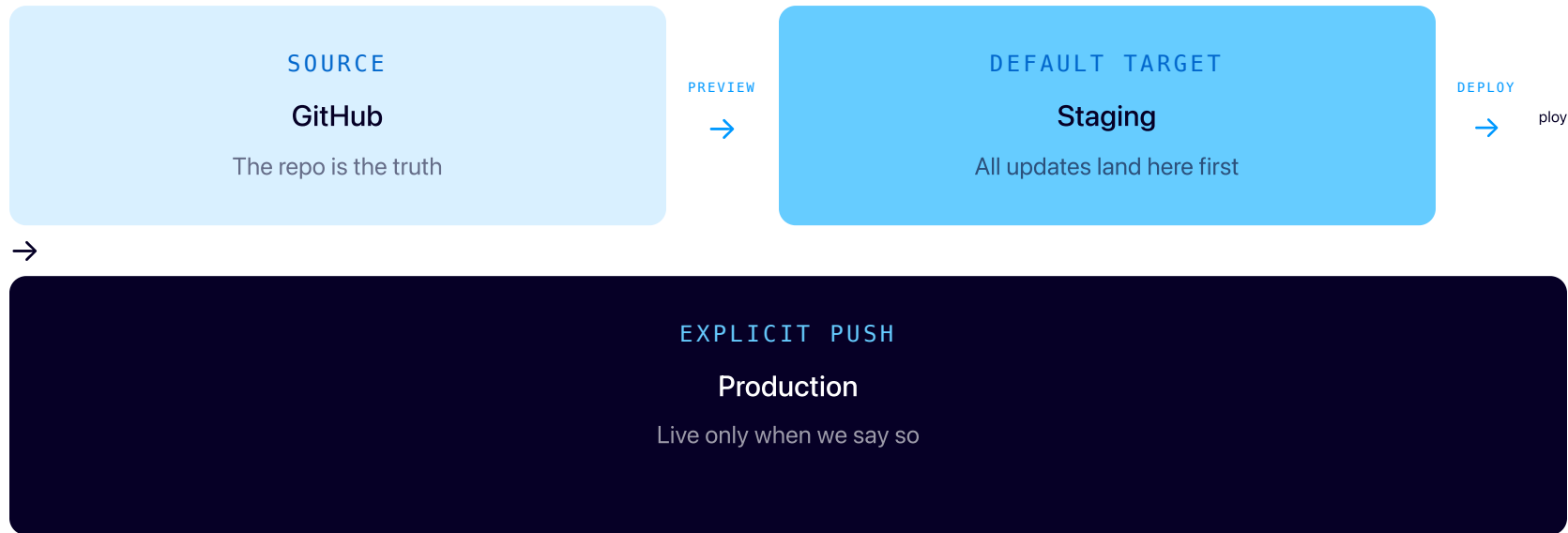
Every change versioned and reviewable. The repo is the site - not a database we can't see into.

THE DEPLOY

Cloudflare Pages

Builds straight from git, fast global edge, staging and production from the same repo.

GitHub + Cloudflare, with a safe promotion step



Cloudflare Pages builds both environments straight from the synced repo.

— PHASE 1

The core website - every page but the hub

SCOPE

Homepage

Services

Industry pages

Case studies

About

Contact

Everything on vertodigital.com except the content hub.

WHAT WENT WELL

- ✓ Brand held page to page - voice and visuals stayed consistent
- ✓ Fast - the 3-tool pipeline repeated cleanly for each page
- ✓ Production code in our repo, not a locked builder

— PHASE 1 FRICTION

Where it needed us in the loop

It accelerated the work - it didn't replace judgment.

Brand rules had to be explicit

When we assumed instead of documenting, it filled gaps its own way.

Human QA every time

Nothing reached production without a review pass from us.

Long pages ate context

Exactly why the /compact ritual became non-negotiable.



Most of our feedback - 80-90% - was on copy. Claude is a great storyteller, but it doesn't know our business. Watch for AI-ish tone and factually wrong claims about what we do and how we help clients.

— PHASE 2

The content hub - same pipeline, different problem

WHAT IT COVERED

The content hub - article templates, listing and taxonomy pages, the reading experience.

WHAT CHANGED

Phase 1 was unique pages. Phase 2 was one repeatable system that had to hold across many.

— PHASE 2 FRICTION

The hard part wasn't building - it was deciding

01

Information architecture

When you can build this fast, the wish-list balloons. The real question: what can we actually maintain - we still rely on people to produce content.

02

Draft first, then ask

Shifted from "what do you want?" to shipping a first draft and asking "what do you think?"

03

Templatize or build first?

Deciding when to templatize up front versus build one, then extract the template.

— SEO & AEO

We optimized every page for three audiences

Findability was built into each page brief - not bolted on afterward.

01 · HUMANS

The open web

Clean semantic structure, fast pages, and content people actually want to read.

02 · SOCIAL CHANNELS

OG tags

Open Graph metadata so every shared link renders a clean, on-brand preview.

03 · AI AGENTS

Train, search, act

Structured, answer-ready, agent-accessible - so AI can find, cite, and act on us.

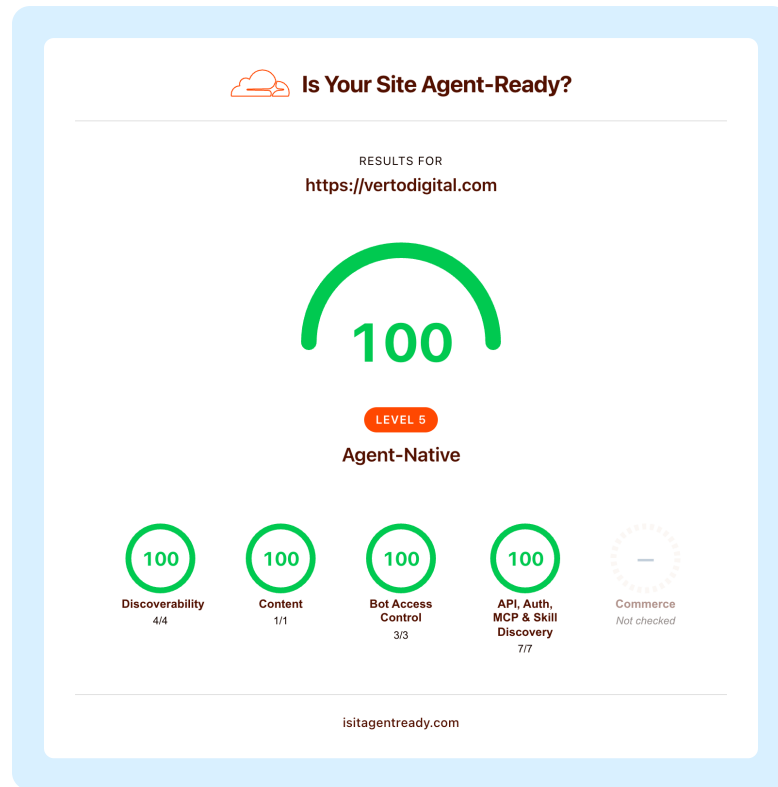
— THE AI-AGENT SCORECARD

Our website is now 100% Agent-Ready

We scored the site on the factors agents care about - discoverability, content, bot access, and API / Auth / MCP / skill discovery.

- 4/4 Discoverability
- 1/1 Content
- 3/3 Bot access control
- 7/7 API, Auth, MCP & skill discovery

Scored via isitagentyready.com ·
<https://vertodigital.com>



We taught Claude our brand as documentation

DESIGN.md

A single brand reference Claude read on every page - the source of truth for look and feel.

Deep Navy · #050126

TWO SEPARATE DOCUMENTED SKILLS

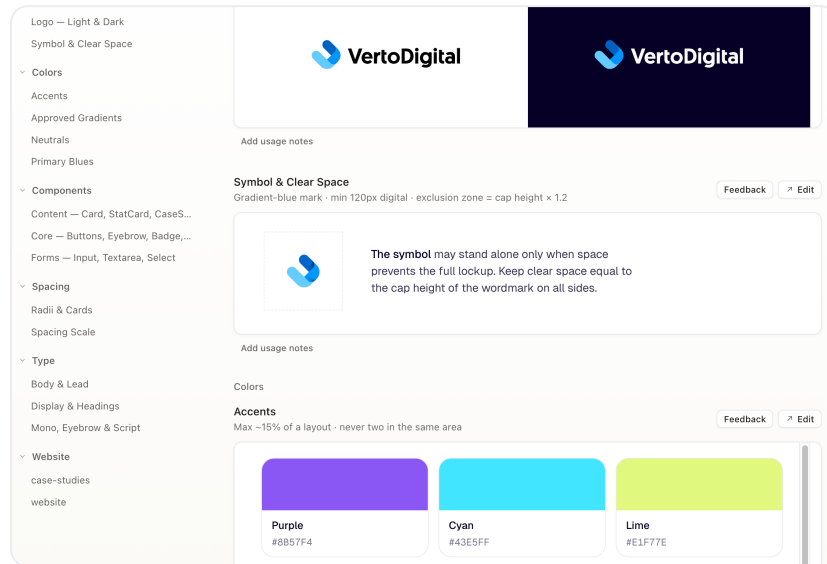
Visual identity

Colors, type, spacing, components.

Brand voice

Pipeline-first, peer-to-peer.

OUR DESIGN SYSTEM, LIVING IN CLAUDE



— THE PAYOFF

Months of work, done in *weeks*

~4

WEEK FOR THE FULL SITE

months

THE OLD TIMELINE

1

PAGE COULD TAKE PARTNERS WEEKS

Before, the work was spread across external partners - each on their own framework and process. A single new page could take weeks. We did the whole site in about four.

— DISCIPLINE

A repeatable ritual for QA-ing and shipping each page

Not just managing context - a consistent way to review, ship, and reset on every single page.

`/core-review`

QA the page

A structured pass to catch brand and layout drift before it moves on.

`staging → prod`

Ship the page

Review on staging, then an explicit deploy - the page is done and live.

`/compact`

Reset for next

Clear the working context so the next page starts fresh.

→ The payoff: maximum clean context available for the *next* task - fewer mistakes, steadier quality across dozens of pages.

— GOING FORWARD

Maintenance is just the workflow, repeated

STAGING → PRODUCTION

Changes land on staging, get reviewed, and go live on an explicit deploy. Updates stay small and safe.

NO CMS LAYER

Edits happen in code, versioned in git. No CMS bloat, no plugins to babysit - with Claude Code making the edits approachable.

— COMPOUNDING TAKEAWAY

The repo got smarter as we went

Every task left something reusable behind - so the next one started further ahead.

A LIBRARY OF SKILLS IN THE REPO

We built up reusable skills as we worked. Now a new page, a redesign, or an optimization pass just references the right one.

agency-credentials

brand-voice

case-studies

linkedin-ads

schema-markup

visual-identity

CLAUDE.md

The starting point for every prompt - read before anything gets coded, so each task begins with the same shared context.

→ Read CLAUDE.md before coding anything.

— WHAT'S NEXT

Tailored landing pages, same pipeline

Next up: campaign-specific LPs. We reuse the exact process – build the template once, reference it forever.

BUILD ONCE

A Claude Design LP template

One layout/template for landing pages – structure, sections, and brand baked in.



REUSE EVERY TIME

One LP per campaign, fast

Every new campaign references the template – on-brand pages in a fraction of the time.

— THE BIGGER SHIFT

Marketers need a *GitHub* account now

This isn't a dev-only tool anymore. Getting hands-on is the real unlock - and there's a natural on-ramp.

START HERE

Build prototypes

Get an account, get hands-on, get comfortable with how these systems actually work - low stakes.



THEN AIM FOR

Production-ready apps

Once the workflow feels familiar, build things you actually ship - like this website.

— KEY TAKEAWAYS

If you're thinking about doing this

01 Document your brand first.

Visual identity and voice as separate, written references.

03 Manage context deliberately.

Review each page, compact before the next.

05 Bake in SEO/AEO.

Put findability in the brief, not a cleanup phase.

02 Work page by page.

One brief, one template, one build - then move on.

04 Own your infra.

Git + a deploy pipeline with a staging→production step.

06 Keep a human in the loop.

It accelerates the work; you still own the judgment.

— THANKS, OVER TO YOU

Questions & *stories*

Tell us your questions - or your own AI coding stories.
What have you shipped?