

The Marketing OS Blueprint

This is the shape of the system that wrote, designed, and shipped the website you found this on, with our specifics swapped for yours. It runs on Claude Code; the structure below is what turns a coding tool into a marketing team.

The structure

```
marketing-os/
├─ CLAUDE.md           how the system works, read by every session
├─ GOVERNANCE.md       what lives where, and who decides
├─ brands/
│   └─ yourbrand/brand.yaml  positioning, voice, audiences, in one file
├─ knowledge-base/
│   └─ brands/yourbrand/    the deep truth: positioning, buyers, proof
├─ .claude/
│   ├── skills/           the playbooks: workflows the system executes
│   ├── agents/           the team: researcher, strategist, creator, QA
│   └─ rules/             guardrails loaded into every working session
├─ content/             what the system produces, findable again
├─ web-assets/          the website, built and shipped from inside
├─ tokens/              the design system: color and type from one file
└─ tasks/               briefs in, work tracked, lessons kept
```

Why this shape works

knowledge first, in order

Positioning, then brand identity, then buyers, then voice, then the competitive frame. The order is load-bearing. Build it out of sequence and every asset downstream inherits the mistake, which is most of what people are calling AI slop.

playbooks, not prompts

A prompt is a one-time ask. A skill is a written playbook the system executes the same way every time, and improves when you correct it once.

agents with lanes

Research, strategy, creation, and review are different jobs. Separate agents hold separate standards, and the QA agent never grades its own homework.

change once

Positioning lives in one place, referenced everywhere. Update the file and every playbook that touches it works from the new truth.

Standing yours up

Ten steps, in the order that works. Each one is small; the sequence is the system.

- 01 **Write the positioning document.** One page: who you serve, what you replace, why you win. Plain sentences a machine can quote.
- 02 **Collect the voice corpus.** Five to ten pieces you actually wrote or said, the way you said them. Transcripts beat polished posts.
- 03 **Create brand.yaml.** Positioning, audiences, voice rules, and banned words, structured. This is the file everything else reads.
- 04 **Seed the knowledge base.** Buyers, proof points, competitors, offers. Deep truth in folders, summaries in the yaml.
- 05 **Write your first playbook.** Pick the asset you make most often. Write the workflow: inputs, steps, output format, quality bar.
- 06 **Define the review gate.** What does the QA check reject? Voice drift, unsupported claims, banned words. Write it down; the system enforces what is written.
- 07 **Run one real brief end to end.** Brief in, draft out, review, revise, approve. Fix the playbook wherever it stumbled.
- 08 **Add the design tokens.** Colors, type, spacing in one file, so every visual asset agrees with every other one.
- 09 **Wire the channels.** The website, the newsletter, the social surfaces, shipping from the system instead of alongside it.
- 10 **Keep the lessons.** Every correction becomes a rule the system reads next session. This is the compounding part; do not skip it.

Three places the tuning actually bites. The order of the knowledge docs, because everything downstream inherits it. The voice corpus, where ten transcripts beat fifty polished posts, because polish is the part that is not you. And the review gate, because a system only enforces what someone bothered to write down. Get those three right and the rest is patient assembly. If you would rather walk a finished one than build from scratch, mine is at themultiplier.io/install