

Kundenseiten mit Claude Code. Schritt für Schritt.

Der Workflow, mit dem KI-Agenten
produktionsreife Websites liefern. Ohne dass
alles an einer Person hängt.

12 Seiten, durchwischen →

DAS PROBLEM

Mit KI schnell bauen kann inzwischen jeder.

Aber mal dauert die Seite zwei Tage, mal zwei Wochen. Was beim letzten Projekt funktioniert hat, liegt in irgendeinem Chatverlauf. Und ob der Code in sechs Monaten noch wartbar ist, weiß niemand.

Das ist kein Tool-Problem.

Der Unterschied ist kein besserer Prompt. Es ist das System drumherum.

Regeln, die jedes Projekt mitnimmt. Checks, die automatisch laufen. Wissen, das im Team bleibt statt in einem Kopf.

So sieht das bei mir aus. Sechs Schritte.

01

Projekt aus dem Fundament starten

Ein Template-Repo mit euren Konventionen: Ordnerstruktur, Komponenten, Formulare, Consent, Tracking. Ein Befehl legt Projekt, Repo und Branches an. Kein Projekt fängt bei null an.

KONKRET

Claude liest bei jedem Start eine CLAUDE.md mit euren Regeln. Drei Beispiele aus meiner: "Keine neuen Dependencies ohne Freigabe." "Formulare nie ohne Spam-Schutz." "Jede Sektion mobil zuerst prüfen."

02

Briefing strukturiert erfassen

Design-Quelle, Seitenstruktur, Texte, Ziel der Seite: Anfrage, Anruf oder Termin. Ein Befehl fragt alles ab und schreibt es in eine Config, die der Agent bei jedem Schritt liest.

KONKRET

Statt "bau mal die Startseite" steht da: Zielgruppe, Conversion-Ziel, Design-Link, Schriften, Farbwerte. Der Agent muss nicht raten. Und was er nicht raten muss, halluziniert er nicht.

03

Sektion für Sektion bauen

Nicht "bau die Website". Erst Hero, dann Leistungen, dann Referenzen. Jede Sektion ist ein abgeschlossener Schritt mit eigenem Check, bevor die nächste dran ist.

KONKRET

Der Agent zieht Radian, Farben und Verlaufs-Winkel direkt aus dem Design-Tool, über eine MCP-Anbindung. Nie vom Screenshot schätzen lassen. Daher kommen die "fast richtigen" Ecken und Verläufe.

04

Checks laufen automatisch

Lint und Build bei jedem Stand. Ein zweiter KI-Agent reviewt jede Änderung, bevor ein Mensch draufschaut. Responsive, Formulare, SEO-Basics: eigene Prüf-Befehle.

KONKRET

Der Review-Agent arbeitet eine Checkliste ab: Kontraste, Alt-Texte, tote Links, Consent vor Tracking. Er wird nicht müde und vergisst nichts. Menschen prüfen nur noch, was übrig bleibt.

05

Feedback wird Regel

Der wichtigste Schritt. Jede Kundenkorrektur wandert als Regel ins System, nicht als Erinnerung in einen Kopf.

KONKRET

Kunde meldet: "Logo verpixelt auf dem iPhone." Die Korrektur wird zur Regel: "Logos als SVG, Fotos in 2x exportieren." Steht ab jetzt im Template. Derselbe Fehler passiert kein zweites Mal. In keinem Projekt.

06

Preview, Freigabe, live

Der Kunde sieht jeden Stand als echten Link auf seinem eigenen Handy, keine Screenshots. Freigabe, Merge, Deploy.

KONKRET

Jeder Arbeitsstand bekommt automatisch eine eigene Vorschau-URL. Feedback kommt früher, Abnahmen werden kleiner, und der große Knall am Ende fällt aus.

DER EIGENTLICHE TRICK

Die meisten sammeln Prompts. Dieser Workflow sammelt Regeln.

Deshalb startet jedes Projekt besser als das letzte.
Das ist der Unterschied zwischen "wir benutzen KI"
und "wir haben ein System".

FUNKTIONIERT DAS?

Fünf- bis zehnmal mehr Output.

Ein Agentur-Team, das ich mit diesem Workflow begleite, liefert heute ein Vielfaches. Nicht weil jemand schneller tippt, sondern weil das ganze Team liefert, nicht mehr nur eine Person.

Und der Workflow gehört ihnen, nicht mir.

UND BEI EUCH?

Euer Workflow. Gemeinsam aufgebaut.

Ich baue das mit Teams auf, die Websites oder Software liefern. Mitbauend statt zuschauend, bis euer Team das Niveau selbst hält. Welche Werkzeuge darunter liegen, richtet sich nach euch.

marcokotzian.de/agenturen