

Wer fährt eigentlich?

Nicht (nur) die KI.

Wo Vibecoding unsere UX-Arbeit schneller gemacht hat und wo es an seine Grenzen stösst

Patrick Federi, Head of UX, Ergon Informatik AG · Swico LAT «KI in Action», 22.09.2026

ergon

Patrick Federi

- **Rolle:** Head of UX bei Ergon: Kundenprojekte, KI-Adoption, internes Tooling
- **Ergon:** Individualsoftware für diverse Branchen.
- **Ausbildung:** MAS HCID (2015), CAS Digital Healthcare (2026).
- **Community:** Aktiv in Ergons Community of Practice AI for Engineers.
- **Schreibt:** Auf Medium über UX, Vibecoding und KI in der Praxis: medium.com/@federi

Agenda

-
- 1 Wo KI heute im UX-Alltag steckt
 - 2 Wie wir bei Ergon arbeiten – Fokus Prototyping
 - 3 Grenzen aus der Praxis
 - 4 Human in the Lead: wozu es uns noch braucht?
-

Wo KI heute schon im UX-Prozess steckt

Research & Strategie

- Interviews transkribieren & auswerten
- Personas & Insights synthetisieren
- Konkurrenzanalysen beschleunigen

Design & Prototyping

- Vibecoding-Prototypen statt Wireframes
- Varianten in Minuten statt Tagen
- Figma-MCP verbindet Design und Code

Testing & Delivery

- Nutzertests protokollieren & clustern
- Content & Microcopy generieren
- Deployment

Wie wir bei Ergon arbeiten

Vibecoding-Prototypen

- Nutzertests
- Stakeholder-Workshops
- Kundenpitches & Offerten

ProtoShip

- Projekt-Setup (inkl. Skills/CLAUDE.md)
- Deployment auf Knopfdruck
- Kein Infrastruktur-Wissen nötig

figma-console-mcp

- Direkt an Claude Code angebunden
- Design fließt direkt in Code und vice versa

„Vibecoding“ ≠ Vibecoding

Vibecoding

- Schnelle, freie Exploration
- Erste Idee sichtbar machen, eine Fragestellung testen
- Bewusst wegwerfbar

Spec-driven agentic development

- Gemeinsames Template, klare Vorgaben
- Für alles, was mehrfach genutzt oder länger betrieben wird

Faustregel

**Was bleiben soll, braucht mehr
Spezifikation und weniger Vibe.**

Wo es uns tatsächlich schneller gemacht hat

Nutzertests

- Prototyp in Stunden statt Wochen
- Teilnehmende reagieren auf echtes Verhalten, nicht auf eine Vorstellung davon

Kundenpitches & Offerten

- Ein klickbarer Prototyp überzeugt schneller als 40 Folien
- Gute Grundlage für die weitere Arbeit

Infrastruktur

- Coolify statt Vercel
- Supabase
- Mit ProtoShip mit einem Klick deploybar

Vier Grenzen aus der Praxis

Arbeiten mit Agenten ist auf der grünen Wiese einfach.

Schwierig wird es bei echten Systemen, ungepflegtem Design und Kunden ohne Gespür für Prototyp vs. Produkt.

Grenze 1: Grüne Wiese vs. Bestandssystem

Grüne Wiese

Neues, unbelastetes Projekt: erstaunlich einfach.

Bestandssystem

Neues Feature in bestehende App bringen: braucht Auseinandersetzung mit der bestehenden Lösung.

Stolperfalle: Nutzertests

Sieht der Prototyp nicht aus wie die echte Applikation, sind auch die Erkenntnisse nicht mehr belastbar.

Grenze 2: Nur so gut wie das Design-System

Sauberes Design-System

Claude Code baut Screens direkt aus Figma: schnell, grösstenteils korrekt.

Unsauberes Design-System

Deutlich schlechteres Ergebnis, denn die KI ist nur so gut wie die Struktur, die sie vorfindet.

Grenze 3: Ein UX-Team bringt kein DevOps mit

Vorher: Vercel & Supabase

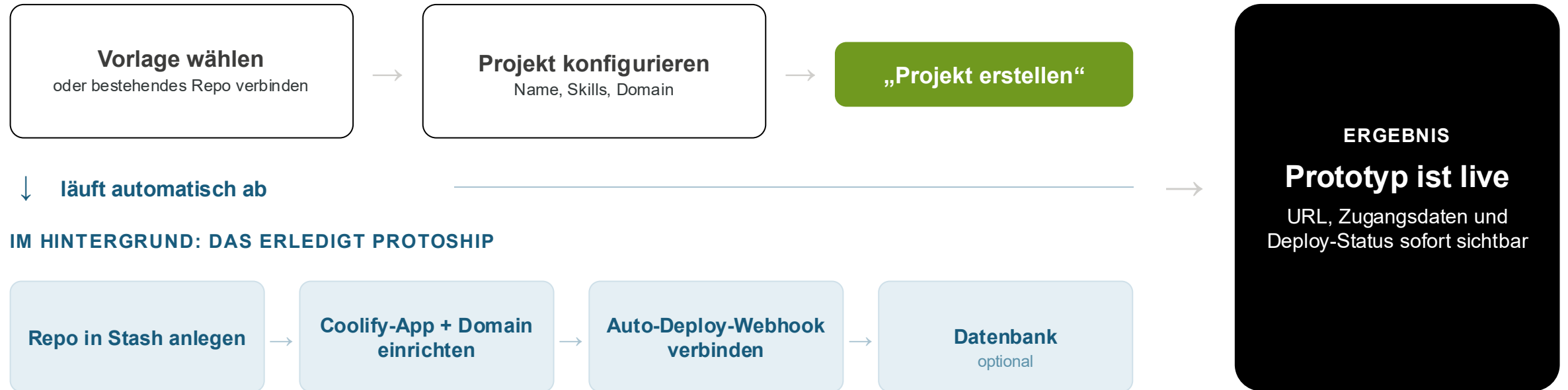
- Kein gemeinsames Repo, kein einfaches Teilen im Team
- Deployment aufwändig
- Alles irgendwo verteilt

Jetzt: ProtoShip

- 5 Felder im Formular → 1 Klick → Live-URL
- Niemand im UX-Team braucht dafür Git-Hosting, Coolify- oder Server-Wissen.

ProtoShip im Detail: der Ablauf

IM VORDERGRUND



ProtoShip im Detail: die Eingabe

Eingabe

- Projektname
- Verantwortliche Person
- Branch
- Subdomain
- Start-Template

 **ProtoShip**
Prototypen schnell deployen

[← Zurück zur Übersicht](#)

Neues Projekt

Neu aus Template

Bestehendes Repo

Skills

Light Full

2 von 15 ausgewählt

Nicht ausgewählte Skills (.md-Dateien) werden aus dem neuen Repo entfernt.

Projektname

Mitarbeiter

Wird in Coolify als Beschreibung hinterlegt (aus deinem Stash-Login vorausgefüllt).

Branch

main

Domain

proto-ship

.coolify.ergon.ch

☐ Mit Datenbank (Supabase)

☐ Auto-Deploy beschränken
Nur Pushes der gelisteten Stash-Nutzer auf den gewählten Branch lösen

 **ProtoShip**
Prototypen schnell deployen

[← Zurück zur Übersicht](#)

Neues Projekt

Neu aus Template

Bestehendes Repo

Skills

Light Full

2 von 15 ausgewählt

CODE & ARCHITEKTUR

CODING_STANDARDS.md

COMPONENT_STANDARDS.md

DATA_FORMATTING.md

PERFORMANCE.md

QUALITÄT & SICHERHEIT

DEFINITION_OF_DONE.md

EDGE_CASE_CHECKLIST.md

SECURITY_BASELINE.md

Branch

main

Domain

proto-ship

.coolify.ergon.ch

☐ Mit Datenbank (Supabase)

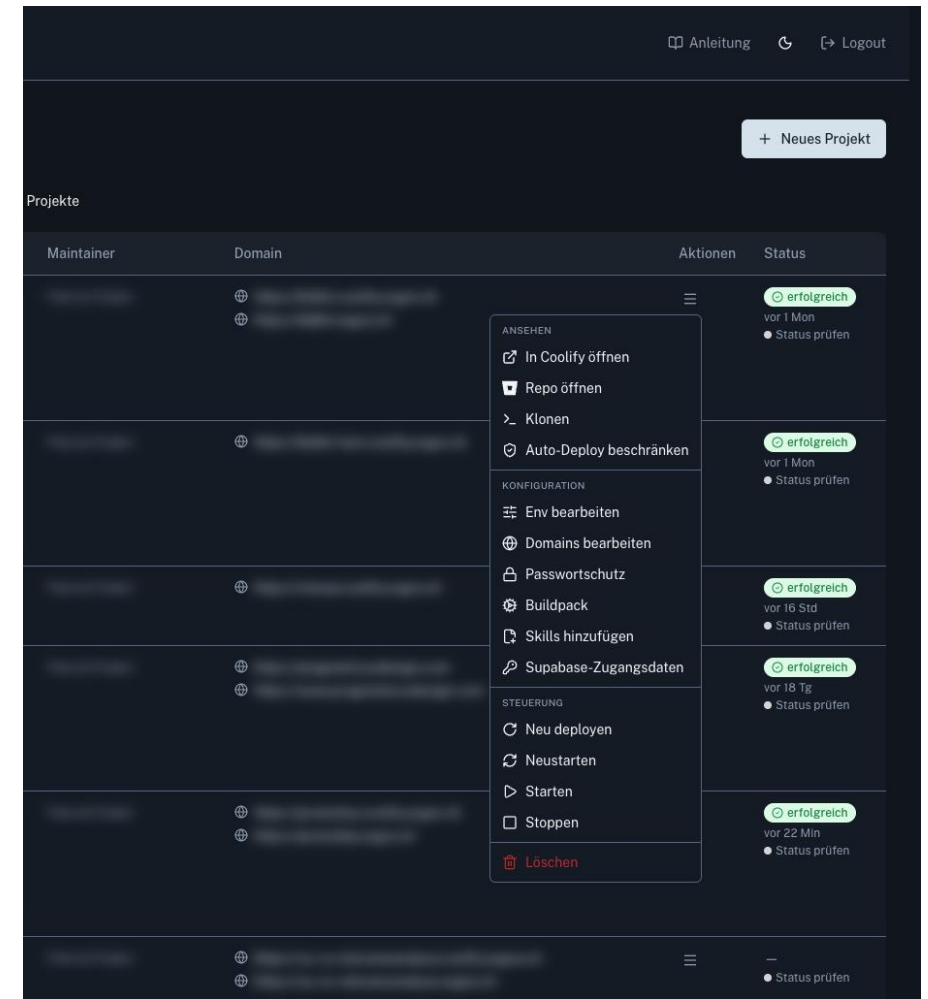
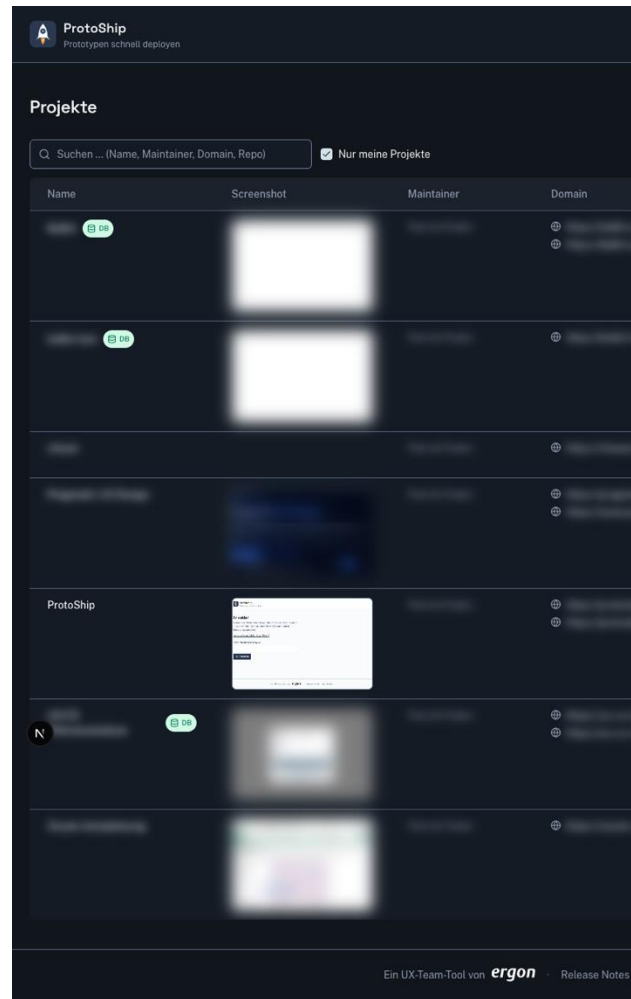
☐ Auto-Deploy beschränken
Nur Pushes der gelisteten Stash-Nutzer auf den gewählten Branch lösen

ProtoShip im Detail: das Ergebnis

Ergebnis

- Repository-Link
- Live-Prototyp-URL
- Coolify-Projekt
- Supabase-DB

Publizieren wird Teil des Prototyping, kein separater Schritt danach.



Grenze 4 – Kundenerwartung & Preismodell

«Das läuft doch schon. Wozu noch Entwickler?»

Ein überzeugender Prototyp löst diese Frage ungefragt aus. Wer den Unterschied zwischen Prototyp und Produkt nicht vorgängig aktiv vermittelt, verliert die Preis- und Vertrauensbasis für die eigentliche Arbeit, die danach noch kommt.

Die 4 Grenzen auf einen Blick

1

Grüne Wiese vs.
Bestandssystem

2

Design-System-Qualität

3

Fehlendes DevOps-
Grundwissen

4

Kundenerwartung &
Preismodell

Wozu braucht es uns dann noch?

- Design ist nicht der Screen, sondern die begründete Entscheidung dahinter: wie etwas funktionieren, wirken und verstanden werden soll.
- Eine KI liefert einen überzeugenden ersten Entwurf. Sie weiss nicht, welcher Edge-Case im Betrieb kritisch wird oder welcher Zielkonflikt hinter einem Feature steckt.
- Ohne klare Kriterien lässt sich eine Entscheidung nicht verteidigen. Zeigt sich der Fehler erst in Produktion, kostet die Korrektur mehr, als ihn vorher zu vermeiden.

**KI ersetzt kein Design. Sie ersetzt
Arbeit, die fälschlich als Design galt.**

Der Unterschied ist nicht, wer den Prototyp baut. Es ist die Methodik, mit der Entscheidungen getroffen werden, und wer dafür geradesteht.

Human in the Lead

Jede KI-generierte Entscheidung durchläuft eine Person, die dafür geradesteht, nicht das Tool.

Wer ein Ergebnis durchwinkt, übernimmt die Verantwortung dafür. Ob er will oder nicht.

Nur: Diese Haltung setzt voraus, dass es auch in Zukunft genug Menschen gibt, die diese Führung übernehmen können. Woher sollen sie kommen, wenn wir ihnen heute die Übung dazu nehmen?

**Arbeiten mit Agenten beschleunigt
das Auto. Es entscheidet nicht, wohin
es fährt.**

**Danke.
Merci.
Grazie.
Grazia fitg.**



Wer fährt eigentlich bei Euch? –

Patrick Federi,
Head of UX,
Ergon Informatik AG

ergon