

Ihr KI-Prototyp ist gebaut. 12 Dinge zwischen ihm und Produktion.

Von Thomas Faß · 22. September 2026 · aktualisiert 28. September 2026

Ein Prototyp beweist die Idee. Ob er in Betrieb trägt, entscheidet sich an Stellen, die in der Demo unsichtbar sind. Diese Liste ist die ehrliche Version — kein Verkaufsgespräch, sondern das, was wir bei jedem Prototyp prüfen, bevor er produktionsreif ist.

Das Wichtigste in Kürze

- Ein Prototyp beantwortet die Frage „geht das?“ — nicht die Frage „hält das?“. Die zweite entscheidet sich an zwölf Stellen, die auf einem Screenshot nicht zu sehen sind.
- Die ersten Punkte der Liste prüft man an einem Nachmittag: Secret-Hygiene, Eingabevalidierung, Fehlerbehandlung.
- Die späteren entscheiden, ob das System trägt: Skalierung, Monitoring, Datenpfad, Wartbarkeit, Übergabe.
- Was trägt, bleibt. Neu gebaut wird nur, was in Produktion nicht hält — ein Prototyp ist kein Wegwerfprodukt, aber auch kein fertiges System.

Was steht zwischen Prototyp und Produktionsreife?

Zwischen einem funktionierenden KI-Prototyp und einem System, das im Betrieb trägt, liegen zwölf Punkte, die in der Demo unsichtbar bleiben: Auth und Rechte, Secret-Hygiene, Datenmodell und Migrationen, Eingabevalidierung und Injection-Schutz, Fehlerbehandlung, Tests, Deploy-Pfad, Skalierung und Last, Monitoring und Logging, DSGVO und Datenpfad, Wartbarkeit sowie Übergabe und Eigentum. Ein Prototyp beweist die Idee. Ob er im Betrieb trägt, entscheidet sich an diesen Stellen.

Die zwölf Punkte

1. **Auth & Rechte.** Ein Prototyp kennt oft nur „eingeloggt oder nicht“. Produktion braucht Rollen, Rechte, sauberes Session-Handling und sichere Passwort-/Token-Flows.
2. **Secret-Hygiene.** Keys, Zugangsdaten und API-Tokens gehören nicht in den Code und nicht ins Frontend. Im Prototyp sind sie oft hartcodiert oder im Client sichtbar.
3. **Datenmodell & Migrationen.** Hält das Schema echte Datenmengen aus — und lässt es sich ändern, ohne Daten zu verlieren? Ein Prototyp-Schema ist selten für beides gebaut.
4. **Eingabevalidierung & Injection-Schutz.** Jede Eingabe ist ein Angriffsvektor. SQL-Injection, XSS, unsichere Datei-Uploads — im Prototyp meist ungeprüft.
5. **Fehlerbehandlung.** Was passiert, wenn etwas schiefgeht? Ein Prototyp zeigt den Stacktrace. Produktion fängt Fehler ab, protokolliert sie und zeigt dem Nutzer etwas Sinnvolles.

6. **Tests.** Ohne automatisierte Tests ist jede Änderung ein Blindflug. Die Kern-Logik braucht Abdeckung, sonst bricht bei jedem Update unbemerkt etwas.
7. **Deploy-Pfad.** „Läuft auf meinem Rechner“ ist kein Deployment. Nötig sind ein reproduzierbarer Build, eine definierte Umgebung und ein Rollback-Weg.
8. **Skalierung & Last.** Hält es 10 Nutzer? 1.000? Prototypen brechen an Stellen, die bei drei Testnutzern nie auffallen – N+1-Queries, fehlende Indizes, kein Caching.
9. **Monitoring & Logging.** Merkt jemand, dass es kaputt ist, bevor der Kunde anruft? Produktion braucht Health-Checks, Fehler-Alerting und nachvollziehbare Logs.
10. **DSGVO & Datenpfad.** Wo liegen die Daten, wer verarbeitet sie, gibt es eine Rechtsgrundlage? Ein Prototyp auf US-Cloud ist im regulierten Umfeld ein Problem, kein Feature.
11. **Wartbarkeit.** Versteht ein anderer Entwickler den Code in sechs Monaten – und kann ihn ändern? KI-generierter Code ohne Struktur wird schnell zur Blackbox.
12. **Übergabe & Eigentum.** Gehört Ihnen der Code? Gibt es Dokumentation, Betriebshandbuch und rotierte Zugänge – oder eine Abhängigkeit durch die Hintertür?

Was die Liste Ihnen sagt

Die ersten Punkte prüft man an einem Nachmittag. Die späteren entscheiden, ob das System trägt – und genau die sind auf dem Screenshot unsichtbar.

Warum diese Punkte beim schnellen Bauen liegen bleiben und wie eine Überführung abläuft, beschreibt der Artikel [Vom KI-Prototyp zur Produktion](#).

Häufige Fragen

Was kostet es, einen Prototyp produktionsreif zu machen?

Das hängt davon ab, wie weit Ihr Prototyp von den zwölf Punkten oben entfernt ist – ein sauber gebautes MVP braucht weniger als ein zusammengeclickter Wegwerf-Prototyp. Deshalb nennen wir keinen Pauschalpreis ins Blaue. Der Weg zum Preis: Im Erstgespräch erhalten Sie einen unverbindlichen Investitionsrahmen (von-bis). Danach nehmen wir Ihren Prototyp in einem kostenlosen Scoping-Paket auf, entscheiden gemeinsam, was bleibt und was neu muss, und legen ein verbindliches Festpreis-Angebot als Werkvertrag vor. Kein Stundenzettel, keine Überraschung hinterher.

Wie lange dauert das?

Auch das hängt vom Abstand zwischen Ihrem Prototyp und der Produktionsreife ab. Häufig ist die Prototyp-Rettung schneller als ein Neubau, weil die Idee bereits validiert ist und wir gezielt die Produktionslücke schließen – oft in Wochen, nicht Monaten. Einen verbindlichen Termin nennen wir nach dem Scoping, mit Termingarantie im Werkvertrag. Was wir nicht tun: ein Tempo versprechen, bevor wir Ihren Code gesehen haben.

Bleibt mein bestehender Code erhalten?

Was trägt, bleibt. Wir starten mit einer Bestandsaufnahme entlang der zwölf Punkte und sagen Ihnen ehrlich, welche Teile produktionsreif sind und welche in Produktion nicht halten würden. Neu gebaut wird nur, was nicht trägt – kein pauschaler Neubau, der die Rechnung aufbläht, und kein „lassen wir so“, das später teuer wird. Die Einordnung bekommen Sie transparent, bevor gebaut wird.

Spielt es eine Rolle, mit welchem Tool der Prototyp gebaut wurde?

Nein. Ob mit Lovable, v0, Bolt, Claude oder klassisch von Hand – entscheidend ist der Zustand des Codes, nicht das Werkzeug. Die zwölf Punkte gelten für jeden Prototyp gleich. Wir übernehmen ihn, wie er ist, und ziehen ihn durch dieselben Senior-Gates.

WENN SIE DEN WEG NICHT ALLEIN GEHEN WOLLEN

Erstellt von InnoShore

Wir bauen Individualsoftware KI-nativ und bringen bestehende Prototypen produktionsreif in Betrieb – senior-gesteuert in Deutschland, souverän gehostet.

[Zur KI-nativen Softwareentwicklung](#)

Thematische Vertiefungen

[KI-native Softwareentwicklung für den Mittelstand](#)

[KI-Produktivität in der Softwareentwicklung](#)

[Individuelle Softwareentwicklung](#)