

Vom Prototyp in die Produktion: 25 Punkte vor dem Start

Checkliste für Anwendungen aus KI-Tools, von Freelancern oder Agenturen. Was fehlt, wie Sie es prüfen und wann Sie neu starten.

Für Gründer, die ihren Prototyp jetzt an echte Nutzer oder Kunden ausliefern. Sie gehen die Liste in einer Stunde durch, auch ohne Entwickler. Wenn Sie eine Antwort nicht wissen, fragen Sie den Entwickler der App.

Vibe Coding baut Ihnen einen Prototyp. Wir bauen Software, auf der ein Unternehmen läuft. Seit 2023 haben wir 15+ Produktionsprojekte ausgeliefert. Diese Liste nutzen wir bei jeder Übernahme.

Ein fehlender Punkt bedeutet keinen Neuanfang. Es ist einfach eine Arbeitsliste.

A. Login, Rechte und Zugangsdaten

- ☐ 1. **Zugriffskontrolle greift an jedem Eingang.** Auch in der API, nicht nur im Frontend. Testen Sie es: Als Standardnutzer anmelden und fremde Daten über die URL aufrufen.
- ☐ 2. **Passwörter und Schlüssel stehen nicht im Code.** Falls sie dort waren, wurden sie ausgetauscht.
- ☐ 3. **Admin-Funktionen sieht nur der Administrator.** Ein unsichtbarer Button reicht nicht aus.
- ☐ 4. **Passwort-Reset und Einladungen sind manipulationssicher.** Links gelten zeitlich begrenzt und verfallen nach einmaliger Nutzung.
- ☐ 5. **Formulare haben einen Bot-Schutz** und ein Limit für Versuche.

B. Daten

- ☐ 6. **Backups laufen automatisch.** Sie haben die Wiederherstellung mindestens einmal scharf getestet.
- ☐ 7. **Datenbank-Änderungen laufen über Migrationen.** Keine manuellen Eingriffe in der Live-Datenbank.
- ☐ 8. **Test- und Produktivdaten sind strikt getrennt.** Sie testen niemals an echten Kunden.
- ☐ 9. **Personenbezogene Daten liegen nur dort, wo nötig.** Sie wissen, wo sie liegen und wer Zugriff hat.

C. Deployment und Umgebungen

- ☐ 10. **Deployments erfolgen per Befehl oder Klick.** Nicht manuell von einem einzelnen Rechner aus.

- ☐ 11. **Es gibt einen Rollback-Pfad.** Die letzte Version lässt sich in wenigen Minuten wiederherstellen.
- ☐ 12. **Sie nutzen eine Testumgebung,** getrennt vom echten Produktivbetrieb.
- ☐ 13. **Dritte können die App von Grund auf starten.** Nach Anleitung, auf einem sauberen Rechner.

D. Qualität

- ☐ 14. **Kernprozesse sind mit Tests abgedeckt.** Registrierung, Bezahlung und alles, wofür Kunden bezahlen.
- ☐ 15. **Tests schlagen bei Fehlern tatsächlich fehl.** Ein grünes Häkchen ohne echte Tests ist nur Deko.
- ☐ 16. **Die CI blockiert den Merge,** wenn Tests fehlschlagen. Sie meldet Fehler nicht nur.
- ☐ 17. **Integrationen haben geprüfte APIs und Limits.** Payment-Gateways, E-Mails oder Drittanbieter-Dienste. Prüfen Sie Limits vor der weiteren Planung.

E. Betrieb

- ☐ 18. **Ein Alert meldet den Ausfall,** nicht der Kunde.
- ☐ 19. **Fehler werden mitgeloggt** und tatsächlich von jemandem gelesen.
- ☐ 20. **Hintergrundjobs werden überwacht.** Was ohne Aufsicht läuft, bleibt irgendwann unbemerkt stehen.
- ☐ 21. **Sie kennen die Betriebskosten** und die Entwicklung bei 10-mal mehr Nutzern.

F. Eigentum und Accounts

- ☐ 22. **Das Repository gehört Ihrer Firma,** inklusive der gesamten Historie.
- ☐ 23. **Domain, Cloud und Zahlungskonten laufen auf die Firma.** Nicht auf privaten Accounts der Entwickler.
- ☐ 24. **Mehr als eine Person hat Zugriff auf alle Accounts.**
- ☐ 25. **Der Vertrag regelt das Eigentum am Code** und die Übergabe beim Dienstleisterwechsel.

Warum Monitoring auf der Liste steht

Was ohne Aufsicht läuft, bleibt irgendwann unbemerkt stehen. Das ist auch uns passiert. Bei einem Projekt für einen Kunden mit Online-Marktplatz fiel im Dezember 2025 der Crawler aus. Das wurde erst spät bemerkt, an einem Diagramm. Die Seitenaufrufe fielen inzwischen von 250K auf 66K im Monat.

Ein einziger Alert am selben Tag hätte den Ausfall sofort gemeldet. Deshalb sind die Punkte 18 bis 20 Pflicht und kein Bonus.

Auswertung

Zählen Sie Ihre Häkchen. Prüfen Sie zuerst die Punkte 1, 2, 6 und 22. Fehlt auch nur ein Punkt, beheben Sie ihn vor dem Start. Die Gesamtpunktzahl spielt dabei keine Rolle.

Status	Urteil	Was das bedeutet
Die Basis steht, Tests laufen, Deployments sind wiederholbar	Weitermachen	Neue Features bauen und offene Punkte parallel schließen
Die App läuft, aber Security, Tests oder Monitoring fehlen	Stabilisieren	Zuerst diese Lücken schließen, danach neue Features entwickeln
Reparaturen dauern länger als ein kompletter Neubau	Neu anfangen	Der Prototyp dient als Spezifikation, der Code entsteht neu

Das häufigste Ergebnis ist kein Neubau. Meist reichen Sicherheits-Fixes, ein stabiles Deployment und danach reguläre Arbeit. Das lässt sich aber erst nach einem Blick in den Code ehrlich beurteilen.

Auch beim Neubau bleibt der Prototyp nützlich. Er zeigt die gewünschte Funktion besser als jede Spezifikation.

Vorbereitung auf das Gespräch

1. Zugriff auf das Repository oder Export aus dem KI-Tool,
2. wer die Anwendung heute wofür nutzt,
3. diese Checkliste mit allen Punkten, bei denen Sie unsicher waren.

Wollen Sie wissen, wie es mit Ihrem Prototyp weitergeht?

In 15 Minuten mit Filip besprechen wir Ihre offenen Punkte. Wir sagen Ihnen, ob Sie weitermachen, stabilisieren oder neu starten sollten.

Buchen Sie 15 Minuten mit Filip: digitaliko.com/de/contact

Digitaliko s.r.o. · www.digitaliko.com · Individualsoftware in Wochen, die Ihnen ab Tag eins gehört.