

Platform-Bottleneck-Checkliste

Diese Checkliste stellt vierundzwanzig Aussagen über euren Weg in die Produktion auf, verteilt auf sechs Bereiche. Ihr hakt ab, was heute stimmt. Was offen bleibt, ist die Liste, die eure Auslieferung ausbremst — und die Grundlage für die Entscheidung, ob sich daran überhaupt etwas mit einer Plattform lösen lässt.

Stand: 2026-08-09

<https://appetizers.io/de/platform-bottleneck-checkliste/>

Für wen sie gedacht ist

Für Engineering-Leitung, Plattform-Teams und alle, die erklären müssen, warum Releases länger dauern, als sie sollten. Geht sie mit zwei oder drei Personen durch, die den Weg in die Produktion aus unterschiedlichen Richtungen kennen: Entwicklung, Betrieb und wer für das Budget geradesteht.

So benutzt ihr sie

Hakt nur ab, was heute schon so ist — nicht, was geplant, beantragt oder in einem Ticket beschrieben ist. Wenn ihr euch bei einer Aussage nicht einig seid, gilt sie als offen: Uneinigkeit im Raum ist selbst schon ein Befund. Am Ende zählt ihr die offenen Punkte, nicht die abgehakten.

01

Weg in die Produktion

- ☐ Eine kleine Änderung geht am selben Arbeitstag live, ohne dass jemand einen Freigabetermin suchen muss.
- ☐ Zwischen Merge und Produktion liegt kein manueller Schritt, den nur eine bestimmte Person ausführen darf.
- ☐ Ein fehlgeschlagenes Deployment lässt sich zurückrollen, ohne dass dabei jemand improvisiert.
- ☐ Jedes Team sieht selbst, ob seine Änderung in Produktion angekommen ist, ohne nachzufragen.

02

Umgebungen und Testdaten

- ☐ Ein Team bekommt eine neue Umgebung, ohne ein Ticket an ein anderes Team zu stellen.
- ☐ Umgebungen entstehen aus Code und nicht aus einer über Jahre gewachsenen Handkonfiguration.
- ☐ Brauchbare Testdaten sind verfügbar, ohne echte Personendaten zu kopieren.
- ☐ Die Unterschiede zwischen Staging und Produktion sind bekannt und aufgeschrieben.

03

Betrieb und Rufbereitschaft

- ☐ Jeder Dienst hat ein Team, das ihn im Störfall übernimmt — benannt, nicht „die IT“.
- ☐ Alarme, die nachts wecken, sind so gebaut, dass die geweckte Person daraufhin auch handeln kann.
- ☐ Für die häufigsten Störungen gibt es Runbooks, die auch jemand ohne Vorwissen abarbeiten kann.
- ☐ Nach einem Vorfall wird die Ursache behoben und nicht nur das Ticket geschlossen.

04

Plattform als Produkt

- ☐ Entwicklerinnen und Entwickler erledigen die häufigsten Aufgaben selbst, statt sie zu beantragen.
- ☐ Es gibt eine Dokumentation, mit der jemand am ersten Arbeitstag etwas ausliefern kann.
- ☐ Es ist entschieden und aufgeschrieben, was die Plattform anbietet und was ausdrücklich nicht.
- ☐ Das Plattform-Team weiß, wer seine Nutzer sind, und ob diese freiwillig kommen.

05

Kosten und Kapazität

- ☐ Jedes Team sieht die Cloud-Kosten, die es selbst verursacht.
- ☐ Ungenutzte Umgebungen und Ressourcen verschwinden automatisch wieder.
- ☐ Kapazitätsentscheidungen stützen sich auf Messwerte statt auf Schätzungen aus der Anfangszeit.
- ☐ Bei einer Kostenüberraschung ist noch am selben Tag klar, woher sie kommt.

06

Wissen und Abhängigkeiten

- ☐ Kein kritischer Teil des Systems hängt an einer einzigen Person.
- ☐ Architekturentscheidungen sind nachlesbar, samt der Gründe, die dagegen sprachen.
- ☐ Versions- und Sicherheitsupdates laufen als Routine und nicht als Projekt.
- ☐ Ein Anbieterwechsel wäre teuer, aber der Preis ist bekannt und benannt.

Wie ihr das Ergebnis lest

Gezählt werden die offenen Punkte, also alles, was ihr nicht abhaken konntet. Die Verteilung sagt mehr als die Summe: Vier offene Punkte in einem einzigen Bereich sind ein anderes Problem als vier über alle sechs verteilt.

0 bis 4 offen

Einzelne Lücken. Das ist normale Instandhaltung und kein Plattform-Thema. Nehmt euch die offenen Punkte einzeln vor.

5 bis 12 offen

Ein erkennbarer Engpass, meist konzentriert in ein oder zwei Bereichen. Hier lohnt es sich, die Ursache zu suchen, bevor ihr Werkzeuge einkauft.

13 und mehr offen

Der Engpass ist strukturell. Einzelmaßnahmen verpuffen hier meist; die Reihenfolge der Maßnahmen entscheidet über das Ergebnis.

Was sie nicht leistet

Die Checkliste misst nichts. Sie ersetzt keine Auswertung eurer Delivery-Daten und keinen Blick in Cluster, Pipelines und Rechnungen. Sie sortiert eine Diskussion, die sonst bei Werkzeugen anfängt, und stellt sie auf den Weg in die Produktion um. Für belastbare Zahlen braucht es die Daten selbst.

Was ihr mit den offenen Punkten macht

Nehmt die Bereiche mit den meisten leeren Kästchen und klärt zuerst, ob dahinter ein gemeinsamer Grund steckt. Wenn ihr eine zweite Meinung wollt: schreibt an info@appetizers.io, mit den Bereichen und ein paar Sätzen zur Lage. Ihr bekommt eine Einschätzung, ob sich daran mit einer Plattform etwas ändern lässt — und die Antwort darf auch nein sein.