ÆNIX

Platform Engineering Maturity Assessment

Bewerten Sie Ihre Organisation entlang von 8 Dimensionen und 5 Reifegraden und erkennen Sie, wo sich Plattforminvestitionen am schnellsten auszahlen.

Für Leiterinnen und Leiter von Plattform, Infrastruktur & Engineering · aenix.io

So nutzen Sie dieses Assessment

Dieses Self-Assessment bewertet Ihre Platform-Engineering-Praxis anhand einer **5-stufigen Reifegradskala**: Pre-Platform → Gemeinsame Infrastruktur → Self-Service-Primitive → Internal Developer Platform → Ausgereift. Es ist dasselbe Basisframework, das Aenix in seinen Platform-Readiness-Assessments verwendet, bevor eine Phase-2-Investition empfohlen wird.

Sie können es auf drei Arten einsetzen:

- [ ] **Self-Assessment (empfohlen für den ersten Durchlauf).** Eine Person — typischerweise eine Plattformleitung oder VP Engineering — bewertet in 30–45 Minuten.
- [ ] **Workshop-Assessment (empfohlen für Investitionsentscheidungen).** Bringen Sie 4–8 Personen aus Platform Engineering, Produkt-Engineering, Security und Ops zusammen und bewerten Sie jede Dimension gemeinsam. Uneinigkeit über den Reifegrad ist selbst diagnostisch wertvoll.
- [ ] **Baseline vor der Zusammenarbeit.** Bewerten Sie, bevor Sie Aenix oder eine andere Beratung hinzuziehen, und bringen Sie das Assessment zum ersten Gespräch mit.

Die 5 Reifegrade

Dieselben fünf Reifegrade gelten einheitlich für alle acht Dimensionen.

Stufe	Bezeichnung	Kurzbeschreibung
Stufe 1	Pre-Platform	Jedes Team für sich.
Stufe 2	Gemeinsame Infrastruktur	Ein zentrales Team verwaltet die gemeinsame Infrastruktur.
Stufe 3	Self-Service-Primitive	Teams bedienen gängige Abläufe selbst.
Stufe 4	Internal Developer Platform	Platform-as-Product, vollständige IDP.
Stufe 5	Ausgereift	Kumulierende Erträge.

Bewertungsregeln

- [ ] Wählen Sie für jede Dimension die Stufe, die Ihren **aktuellen Zustand** am besten beschreibt (nicht Ihr Ziel).
- [ ] Liegen Sie **zwischen** zwei Stufen, wählen Sie die niedrigere Zahl — es sei denn, der Übergang ist bereits weit fortgeschritten (dann die höhere).
- [ ] Halbe Werte sind nur für den Durchschnitt zulässig — einzelne Dimensionen werden ganzzahlig von 1 bis 5 bewertet.
- [ ] Bewerten Sie **ehrlich** — der Wert liegt darin zu erkennen, wo investiert werden sollte, nicht darin, ein bequemes Bild zu bestätigen.

> Bewerten Sie zunächst mit Bleistift und bringen Sie den Bogen dann in einen Workshop ein. Das Gespräch darüber, warum jemand eine Dimension mit 2 statt 3 bewertet, ist wertvoller als die Zahl selbst.

---

1. Workload-Portabilität

Wie leicht lässt sich ein Workload zwischen Substraten (Cloud-Regionen, Rechenzentren, Anbietern, Kubernetes-Clustern) verschieben, ohne ihn neu schreiben zu müssen?

Stufe	Bezeichnung	Beschreibung
Stufe 1	Pre-Platform	Jedes Team konfiguriert seine eigene Infrastruktur. Kein IaC. Bare-Metal-Skripte, Click-Ops. Workloads verankern Substratannahmen tief im Code. Vendor-Lock-in überall.
Stufe 2	Gemeinsame Infrastruktur	Ein zentrales Team verwaltet die gemeinsame Infrastruktur. Teilweises IaC (Terraform / CloudFormation stückweise). Einige Workloads sind teamübergreifend portabel; Substratwechsel erfordern weiterhin ein Redesign.
Stufe 3	Self-Service-Primitive	IaC für die meisten Bereitstellungen. Teams bedienen gängige Abläufe selbst. Workloads sind meist als Kubernetes-Manifeste plus Helm ausdrückbar. Substratspezifische Annahmen sind isoliert.
Stufe 4	Internal Developer Platform	Workload-Portabilität ist strukturell verankert. Multi-Substrat-fähig (Region-Failover, Cluster-Migration). Anbietergebundene Dienste werden explizit gekennzeichnet und isoliert.
Stufe 5	Ausgereift	Workloads nutzen ausschließlich Plattformabstraktionen. Mit vorhersehbarem Aufwand über Substrate hinweg portabel. Drift-Erkennung automatisiert. Neue Substrate werden als austauschbare Backends angebunden.

Ihre Bewertung (1–5): _____ Warum? Belege?

_____

2. GitOps-Adoption

In welchem Ausmaß sind Infrastruktur- und Plattformänderungen deklarativ, versioniert, per PR geprüft und über kontinuierliche Reconciliation ausgerollt?

Stufe	Bezeichnung	Beschreibung
Stufe 1	Pre-Platform	Manuelle Änderungen. SSH, Click-Ops, kubectl von Hand. Keine Source of Truth für den Umgebungszustand.
Stufe 2	Gemeinsame Infrastruktur	Etwas IaC ist in Git eingecheckt, wird aber manuell angewendet. PRs sind ein Wunschbild. Drift sammelt sich an.
Stufe 3	Self-Service-Primitive	IaC plus GitOps für einige Workloads (Argo CD / Flux für App-Deployments). Infrastrukturänderungen weiterhin gemischt manuell und automatisiert. PR-Review bei Plattformänderungen.
Stufe 4	Internal Developer Platform	GitOps für App und Infrastruktur. Kontinuierliche Reconciliation. Drift wird sichtbar gemacht und behoben. PRs sind der Kanal für das Change-Management. Vollständiger Audit-Trail.
Stufe 5	Ausgereift	Vollständig deklaratives Modell — Cluster, Netzwerke, IAM, Observability werden alle aus Git rekonziliiert. Drift ohne Alarm unmöglich. PRs durch automatisierte

Stufe	Bezeichnung	Beschreibung
		Policies (OPA / Gatekeeper, Conftest) abgesichert.

Ihre Bewertung (1–5): _____ Warum? Belege?

_____

3. Vereinheitlichung der Observability

Verfügen Ihre Engineers über einen einzigen, verlässlichen Weg, um Metriken, Logs, Traces und SLOs über alle Umgebungen und Mandanten hinweg zu sehen?

Stufe	Bezeichnung	Beschreibung
Stufe 1	Pre-Platform	Jedes Team wählt eigene Werkzeuge. Wissen darüber, wo man nachschaut, ist informell. Post-Mortems nach Ausfällen offenbaren fragmentierte Daten.
Stufe 2	Gemeinsame Infrastruktur	Zentralisierte Metriken und Logs (z. B. Prometheus + Loki oder VictoriaMetrics + VictoriaLogs). Abdeckung unvollständig. SLO-Disziplin lückenhaft.
Stufe 3	Self-Service-Primitive	Ein zentrales Dashboard (Single Pane of Glass) für die meisten Workloads. SLOs für Tier-1-Dienste definiert. Alert-Hygiene verbessert sich. Tracing entsteht.
Stufe 4	Internal Developer Platform	Vereinheitlichte Metriken, Logs und Traces. SLO-Disziplin durchgängig für Tier 1 und Tier 2. Starke Alert-Hygiene (wenig Alarmrauschen, verlinkte Runbooks). On-Call-Dashboards standardisiert.
Stufe 5	Ausgereift	Vollständiges Observability-Mesh. SLOs werden automatisch aus Plattformverträgen instrumentiert. Anomalieerkennung speist das Alerting. Observability-Kosten werden erfasst und optimiert. Mandanten erhalten eingegrenzte Sichten.

Ihre Bewertung (1–5): _____ Warum? Belege?

_____

4. Umgang mit Secrets

Wie werden Secrets erzeugt, gespeichert, verteilt, rotiert und auditiert?

Stufe	Bezeichnung	Beschreibung
Stufe 1	Pre-Platform	Secrets in Env-Dateien, im Code, in Chat-Nachrichten. Keine zentrale Verwaltung. Rotation manuell oder nie.
Stufe 2	Gemeinsame Infrastruktur	Vault oder Ähnliches für einige Workloads im Einsatz. Rotation ein Wunschbild. Audit-Trail lückenhaft.
Stufe 3	Self-Service-Primitive	Zentraler Secret-Store (Vault / External Secrets Operator). Die meisten Workloads beziehen daraus. Rotationsrichtlinie definiert. Teilweise Auditierbarkeit.
Stufe 4	Internal Developer Platform	Alle Secrets zentralisiert. Rotation erzwungen (richtlinienbasiert). Workload-Identität föderiert (keine eingebetteten Zugangsdaten). Vollständiger Audit-Trail.

Stufe	Bezeichnung	Beschreibung
Stufe 5	Ausgereift	Ausschließlich Workload-Identität — keine statischen Secrets in Workloads. Rotation kontinuierlich und automatisiert. Kundenkontrollierte Schlüssel für souveräne Workloads. Regelmäßiger Chaos-Test der Rotation.

Ihre Bewertung (1–5): _____ Warum? Belege?

_____

5. Identitätsmodell

Sind Mitarbeiteridentität (Engineers) und Workload-Identität (Dienste) zu einer einzigen vertrauenswürdigen Quelle föderiert oder über Systeme hinweg fragmentiert?

Stufe	Bezeichnung	Beschreibung
Stufe 1	Pre-Platform	Überall lokale Konten. Geteilte Zugangsdaten. Joiner-Mover-Leaver (JML) manuell.
Stufe 2	Gemeinsame Infrastruktur	SSO / OIDC für einige Systeme. Workload-Identität ad hoc (statische API-Schlüssel). JML überwiegend manuell.
Stufe 3	Self-Service-Primitive	SSO für den Zugang der Engineers. Workload-Identität im Entstehen (z. B. Service Accounts mit OIDC, IAM-Rollen für Service Accounts). JML teilautomatisiert.
Stufe 4	Internal Developer Platform	Durchgängiges SSO plus MFA für Engineers. Föderation zwischen Mitarbeiter- und Workload-Identität (z. B. SPIFFE / SPIRE, OIDC-basiert). JML vollständig automatisiert. RBAC vierteljährlich überprüft.
Stufe 5	Ausgereift	Zero-Trust-Identität. Autorisierung pro Aktion (nicht nur pro System). Kundenseitige Identitätsföderation unterstützt (BYOidp). Kontinuierliche Attestierung.

Ihre Bewertung (1–5): _____ Warum? Belege?

_____

6. Mandantenfähigkeit

Wie wird die Isolation zwischen Teams, Produkten, Kunden oder Umgebungen erreicht? Auf welcher Ebene? Wie robust ist die Grenze?

Stufe	Bezeichnung	Beschreibung
Stufe 1	Pre-Platform	Keine formale Isolation. „Test" und „Prod" teilen sich Cluster, VLANs oder Datenbanken. Noisy-Neighbour-Probleme zwischen Mandanten sind häufig.
Stufe 2	Gemeinsame Infrastruktur	Namespace pro Team in Kubernetes. Netzwerkisolation lückenhaft (keine NetworkPolicy / Cilium). Ressourcenquoten lückenhaft.
Stufe 3	Self-Service-Primitive	Namespace plus Quota plus NetworkPolicy. Manche Mandanten erhalten einen Cluster pro Team für strengere Isolation. Identität pro Mandant eingegrenzt.

Stufe	Bezeichnung	Beschreibung
Stufe 4	Internal Developer Platform	Tenant-CRD oder Äquivalent — der Mandant ist ein erstklassiges Plattformkonzept. Isolation auf Netzwerk-, Identitäts-, Storage- und Observability-Ebene. Quoten und SLOs pro Mandant.
Stufe 5	Ausgereift	Mehrstufige Mandantenfähigkeit (Organisation → Projekt → Workload). Zusicherungen zwischen Mandanten sind nachweisbar. Kundenisolationsgrad (geeignet für souveräne / regulierte Workloads). Mandanten-Onboarding vollständig automatisiert.

Ihre Bewertung (1–5): _____ Warum? Belege?

_____

7. Disaster-Recovery-Reife

Werden Backup, Restore und Disaster Recovery routinemäßig getestet und nachgewiesen, oder ist es ein Plan auf dem Papier?

Stufe	Bezeichnung	Beschreibung
Stufe 1	Pre-Platform	Backup teilweise und ungetestet. Kein DR-Plan oder ein seit Jahren nicht getesteter Plan. RTO / RPO nicht dokumentiert.
Stufe 2	Gemeinsame Infrastruktur	Backup für die meisten Workloads automatisiert. Restore ad hoc getestet. DR-Plan vorhanden, aber in den letzten 12 Monaten nicht geübt.
Stufe 3	Self-Service-Primitive	Backup und Restore jährlich für Tier-1-Workloads getestet. DR-Standort vorhanden; Failover teilweise getestet. RTO / RPO dokumentiert.
Stufe 4	Internal Developer Platform	Backup, Restore und Failover vierteljährlich für Tier 1 und jährlich für Tier 2 getestet. RTO / RPO gemessen (nicht nur dokumentiert). DR-Testergebnisse für Audits aufbewahrt.
Stufe 5	Ausgereift	Kontinuierliche DR (Game-Day-Rhythmus). Tier-1-Workloads Multi-Region Active-Active oder Active-Warm. Kundenisolation während der DR gewahrt. Compliance-Nachweise (DORA Art. 25, NIS2-BCP) automatisch erzeugt.

Ihre Bewertung (1–5): _____ Warum? Belege?

_____

8. Tiefe des Self-Service

Wie schnell gelangt eine Produkt-Engineerin von „Ich brauche eine Umgebung" zu „Ich habe eine, die meinen Code ausführt" — ohne ticketgetriebene Übergaben?

Stufe	Bezeichnung	Beschreibung
Stufe 1	Pre-Platform	Ticketgetrieben. Tage bis Wochen. Engineers warten auf Ops. Kein Standardweg.

Stufe	Bezeichnung	Beschreibung
Stufe 2	Gemeinsame Infrastruktur	Etwas Self-Service für Dev-Umgebungen. Produktion weiterhin ticketgetrieben. Standardwege existieren informell; Dokumentation lückenhaft.
Stufe 3	Self-Service-Primitive	Self-Service für Non-Prod und viele Prod-Abläufe. Golden Path für gängige Workload-Typen dokumentiert. Ticket-Warteschlange schrumpft.
Stufe 4	Internal Developer Platform	Durchgängiger Self-Service (Umgebung, Deploy, Observability, Skalierung, Rollback). Time-to-Environment in Minuten. Golden Paths sind erstklassig — Abweichungen erfordern einen PR plus Review.
Stufe 5	Ausgereift	Platform-as-Product. Engineers beschreiben Ergebnisse (Workload plus SLO plus Mandanten); die Plattform realisiert sie. Interner NPS gemessen. Kontinuierlicher Verbesserungskreislauf.

Ihre Bewertung (1–5): _____ Warum? Belege?

_____

Ihre Bewertungen

Übertragen Sie jede Dimensionsbewertung unten und berechnen Sie den Durchschnitt.

Dimension	Bewertung (1–5)	Stufe
1. Workload-Portabilität	_____	_____
2. GitOps-Adoption	_____	_____
3. Vereinheitlichung der Observability	_____	_____
4. Umgang mit Secrets	_____	_____
5. Identitätsmodell	_____	_____
6. Mandantenfähigkeit	_____	_____
7. Disaster-Recovery-Reife	_____	_____
8. Tiefe des Self-Service	_____	_____
Durchschnitt	**_____**	**_____**

Niedrigste 2 Dimensionen: _____

Höchste 2 Dimensionen: _____

> Ein gezacktes Profil — in manchen Dimensionen sehr hoch, in anderen sehr niedrig — deutet meist auf ungleichmäßige Investitionen hin. Die niedrigsten Dimensionen anzuheben hat in der Regel eine größere Hebelwirkung als die bereits starken weiter zu verbessern.

Was bedeutet mein Durchschnittswert?

Wertebereich	Stufe	Interpretation	Nächste Schritte
1,0 – 1,5	Stufe 1 — Pre-Platform	Überwiegend fragmentierte Infrastruktur. Heldentaten halten den Betrieb am Laufen. Jede Investition in ein Plattformfundament zahlt sich sofort aus.	Ein Plattformteam aufbauen (3–5 Engineers). IaC plus GitOps für neue Workloads einführen. Secrets zentralisieren.
1,5 – 2,5	Stufe 2 — Gemeinsame Infrastruktur	Ein Kern-Stack existiert, ist aber ticketgetrieben. Engineers warten auf Ops. Ein sichtbarer Engpass.	Golden Paths für die drei häufigsten Workload-Typen definieren. Bereitstellung auf Self-Service umstellen. GitOps durchsetzen.
2,5 – 3,5	Stufe 3 — Self-Service	Auf dem Weg zur IDP. Self-Service teilweise. Golden Paths im Entstehen.	In ein IDP-Produktdenken investieren. Entwickler als Kunden behandeln. Time-to-Environment und internen NPS messen.
3,5 – 4,5	Stufe 4 — IDP	Internal Developer Platform in Betrieb. Die meisten Teams bedienen sich selbst.	Mandantenfähigkeit, DR und Observability vertiefen. Die Plattform intern produktisieren. Auf unterversorgte Workload-Typen ausweiten (Daten, KI, Edge).
4,5 – 5,0	Stufe 5 — Ausgereift	Kumulierende Erträge. Das Plattformteam wirkt hebelnd über die gesamte Organisation.	Kontinuierliche Verbesserung. Extern benchmarken. Erwägen, Werkzeuge extern zu teilen (Open-Source-Beitrag, Community-Präsenz).

Worin sollten wir zuerst investieren?

Wenn Ihre schwächste Dimension ... ist

Schwächste Dimension	Investition mit der größten Hebelwirkung
1. Workload-Portabilität	IaC durchgängig einführen; anbietergebundene Dienste isolieren; ein Kubernetes-basiertes Substrat pilotieren.
2. GitOps-Adoption	Argo CD oder Flux ausrollen; App-Deployments auf einen PR-getriebenen Ablauf umstellen; anschließend Infrastruktur-GitOps ergänzen.
3. Vereinheitlichung der Observability	Vereinheitlichte Metriken plus Logs aufbauen (z. B. VictoriaMetrics + VictoriaLogs); SLOs für Tier 1 definieren; Alerts bereinigen.
4. Umgang mit Secrets	Zentralen Secret-Store plus External Secrets Operator ausrollen; statische Secrets in Workloads eliminieren.
5. Identitätsmodell	Durchgängiges SSO plus MFA; Workload-Identität über OIDC / SPIFFE; JML automatisieren.
6. Mandantenfähigkeit	Ein Tenant-CRD-Muster einführen (z. B. Cozystack Tenant CRD); NetworkPolicy / Cilium durchsetzen; Quoten pro Mandant.

Schwächste Dimension	Investition mit der größten Hebelwirkung
7. Disaster-Recovery-Reife	Backup-Restore plus Failover für Tier 1 innerhalb von 30 Tagen testen; RTO / RPO dokumentieren; danach vierteljährlicher Rhythmus.
8. Tiefe des Self-Service	Die drei wichtigsten Golden Paths definieren; ein Portal oder einen CLI-Wrapper bauen; Time-to-Environment messen.

Häufige Muster

> **Muster „Stark bei Werkzeugen, schwach bei Mandantenfähigkeit / DR".** Häufig bei Organisationen zu beobachten, die seit 2–3 Jahren in ihre Plattform investieren. Die Werkzeugkompetenz ist hoch (GitOps, Observability, Secrets), doch teamübergreifende Isolation und getestete DR hinken hinterher. Investitionen in Mandantenfähigkeit als Produkt erschließen regulierte Workloads und kundenorientierte Plattformangebote.

> **Muster „Viel Self-Service, geringe Reife".** Engineers können vieles selbst bedienen, doch die zugrunde liegenden Primitive sind fragil. Das Risiko ist ein plötzliches Plateau, wenn diese Schwäche zutage tritt. Investieren Sie in die niedrigsten 2 Dimensionen, bevor Sie den Self-Service weiter vertiefen.

Nächste Schritte

Option 1 — Selbst nachbessern mit diesem Assessment. Teilen Sie das Assessment mit der Engineering-Leitung. Bearbeiten Sie die niedrigsten 2 Dimensionen in einem Workshop. Definieren Sie einen 90-Tage-Plan.

Option 2 — Platform Readiness Assessment mit Aenix (14 Tage). Wir bewerten gemeinsam mit Ihnen, prüfen die Ergebnisse auf Plausibilität und erstellen eine Roadmap für die Phase-2-Plattformarbeit. Zum Festpreis.

Option 3 — Managed Engagement für Platform Engineering. Ein mehrmonatiger Aufbau mit Architektur, IDP-Design, Golden Paths, Observability, DR und Mandantenfähigkeit. Der Umfang wird nach Phase 1 festgelegt.

> **Vereinbaren Sie ein Erstgespräch** → aenix.io/services/platform-readiness-assessment/

Aenix ist das Team hinter Cozystack (CNCF Project) und bietet Ænix Platform — unser kommerzielles, produktisiertes Angebot auf Basis von Cozystack. aenix.io