Engineer

Service Level Objective

◎ Messbares Zuverlässigkeitsziel, das ein Dienst erfüllen muss

Was es ist

Ein Service Level Objective ist ein Ziel für das Verhalten eines Dienstes, so genau formuliert, dass jeder seine Einhaltung prüfen kann: ein Anteil von Anfragen, der innerhalb eines benannten Zeitraums erfolgreich oder schnell genug beantwortet werden muss. Fast jeder kürzt es zu SLO ab. Googles Buch Site Reliability Engineering definiert es als Zielwert oder Wertebereich für ein bestimmtes Serviceniveau, gemessen durch einen Indikator.

Dieser Indikator heißt Service Level Indicator, kurz SLI: eine sorgfältig definierte Größe, die einen Aspekt des tatsächlich bei den Nutzern ankommenden Dienstes beschreibt. Anfragelatenz ist die übliche erste Wahl, dicht gefolgt von Fehlerrate und Durchsatz. Das SRE Workbook empfiehlt, jeden SLI als Verhältnis guter Ereignisse zu allen Ereignissen zu schreiben. Damit reicht er von 0 %, wo nichts funktioniert, bis 100 %, wo nichts fehlschlägt. Das Objective zieht dann eine Linie auf dieser Skala: Der Indikator muss auf der richtigen Seite eines Zielwerts oder zwischen einer unteren und einer oberen Grenze bleiben.

Ein vollständiges Ziel nennt alles, was zu seiner Prüfung nötig ist. Das Beispiel des Buchs lautet: 99 % der Get-RPC-Aufrufe werden in weniger als 100 ms abgeschlossen; dabei wird der Anteil über eine Minute gemittelt und über alle Backendserver gemessen. Ein Indikator kann mehrere Ziele zugleich tragen, etwa 90 % der Aufrufe unter 1 ms, 99 % unter 10 ms und 99,9 % unter 100 ms. Ein einzelnes Perzentil beschreibt selten eine ganze Verteilung. Das Workbook trennt außerdem die Spezifikation, also das für Nutzer relevante Ergebnis, von der Implementierung, also dem Messort: einem Serverprotokoll, einer Sonde mit Browser oder Code in der Seite selbst. Jede Variante sieht andere Fehler, und die Wahl gehört in die Definition.

Ein SLO ist kein SLA. Ein Service Level Agreement ist ein ausdrücklicher oder stillschweigender Vertrag mit Nutzern, der Folgen an das Einhalten oder Verfehlen seiner Ziele knüpft, am sichtbarsten eine Rückvergütung oder Vertragsstrafe. Der Test des Buchs fragt, was beim Verfehlen des Ziels passiert: Folgt nichts ausdrücklich Vereinbartes, ist es ein SLO. Eine Fußnote ergänzt, dass mit einem verletzten SLA fast immer ein verfehltes SLO gemeint ist. Ein wirklicher SLA-Verstoß kann wegen Vertragsverletzung vor Gericht enden. Google Search hat laut Buch kein SLA für die Öffentlichkeit, doch ein Ausfall kostet dennoch Ansehen und Werbeeinnahmen.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Service Level Objective?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Monitoring und Alarme melden sich, solange ein Fehler noch klein ist; Kennzahlen zeigen, wie oft das gelingt.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Das erste Observability-Dashboard aufgebaut — der erste Einblick des Unternehmens in die eigenen Systeme

Das erste Observability-Dashboard aufgebaut Kontakt aufnehmen →

Warum es wichtig ist

Ohne erklärtes Ziel erfinden Nutzer ihr eigenes. Das SRE-Buch beschreibt beide Fehlentwicklungen: Menschen vertrauen einem Dienst mehr, als sie sollten, und mögliche Nutzer halten ihn für unzuverlässiger, als er ist, und bleiben fern. Das Beispiel ist Chubby, Googles Sperrdienst für verteilte Systeme. Globale Ausfälle waren so selten, dass andere Dienste unbemerkt davon abhängig wurden, dass er nie ausfiel. Bei den seltenen tatsächlichen Ausfällen brachen diese Dienste zusammen. Die Antwort war, die globale Instanz ihr SLO erfüllen zu lassen, ohne es wesentlich zu übertreffen: In jedem Quartal, in dem echte Ausfälle die Verfügbarkeit nicht unter das Ziel gedrückt hatten, legte SRE das System absichtlich still.

Das Ziel klärt auch, wie viel Zuverlässigkeit genügt. „100 % Zuverlässigkeit ist das falsche Ziel“, schreibt das SRE Workbook, und seine Gründe sind praktisch. Komponenten können unabhängig von ihrer Bauweise gleichzeitig ausfallen. Auch die Kette aus Netzwerken und Geräten zwischen einem Dienst und seinen Kunden fällt aus. Die Kunden würden die 100 % deshalb nie sehen. Jede weitere Neun kostet mehr als die vorherige, während ihr Nutzen gegen null geht. Und weil Änderungen die häufigste Ausfallursache sind, kann ein Dienst, der nie ausfallen darf, auch nie aktualisiert werden. Liegt das Ziel unter 100 %, lässt der

Abstand zur Perfektion Spielraum für Veröffentlichungen, Experimente und geplante Wartung: das Fehlerbudget, schlicht 100 % minus SLO. Jemand muss diese Abwägung zwischen neuen Funktionen und Verfügbarkeit verantworten. Das Workbook nennt dafür den CTO in einer kleinen Organisation und den Product Owner oder Produktmanager in einer größeren.

Schließlich macht das Ziel aus Überwachung Entscheidungen. Das Buch beschreibt einen Regelkreis: Indikatoren messen, mit den Zielen vergleichen, Handlungsbedarf feststellen und handeln. Steigt die Latenz und wird sie die Grenze in einigen Stunden überschreiten, prüft das Team, ob die CPU der Engpass auf den Servern ist, und fügt weitere Server hinzu. Ohne Ziel, so das Buch, kann niemand sagen, ob oder wann gehandelt werden soll.

Wie es in der Praxis aussieht

Nehmen Sie eine Checkout-API und ein nach dem Vorschlag des Workbooks formuliertes Ziel: Indikator und Messort sind getrennt, das Zeitfenster ist benannt und die Begründung steht daneben:

```
Dienst:       Checkout-API
SLI:          Anteil der POST /checkout-Anfragen mit Antwort ohne 5xx-Status,
              gezählt im Anfrageprotokoll des Servers
Ziel:         99.9% über ein rollierendes Zeitfenster von 28 Tagen
Begründung:   Beobachtung, aus vier Wochen Daten nach unten gerundet
Verantwortung: Produktmanager zusammen mit den Bereitschaftsingenieuren
Prüfung:      anfangs monatlich, nach der Stabilisierung vierteljährlich
```

Das Zeitfenster ist ebenso wichtig wie der Prozentsatz. Das Workbook bevorzugt ein rollierendes Fenster gegenüber einem Kalendermonat. Nutzer vergessen einen Ausfall am letzten Tag eines Monats nicht, sobald der nächste beginnt. Es bevorzugt außerdem ganze Wochen, damit jedes Fenster gleich viele Wochenenden enthält; vier Wochen sind seine allgemeine Empfehlung. Dieses Fenster umfasst 28 × 24 × 60 = 40.320 Minuten. 0,1 % davon sind etwas mehr als 40 Minuten vollständiger Ausfall. In einem Monat mit 30 Tagen erlauben dieselben 99,9 % 43,2 Minuten, die Zahl aus der Verfügbarkeitstabelle des Buchs.

Zeit ist allerdings das gröbere Zählmaß. Das SRE-Buch erklärt, dass Google Verfügbarkeit als Erfolgsquote von Anfragen statt als Betriebszeit misst. Ein weltweit verteilter Dienst ist selten vollständig verfügbar oder vollständig ausgefallen. In Anfragen gerechnet lautet das Beispiel des Workbooks: Bei 99,9

% lassen 3 Millionen Anfragen über vier Wochen Raum für 3.000 Fehler. Ein Ausfall, der 1.500 davon verursacht, verbraucht das halbe Budget. Das Zählen von Anfragen erfasst auch die Kosten von Teilausfällen, die eine reine Betriebszeitmessung übersieht. Bei gleichmäßigem Verkehr kosten 20 Minuten mit der Hälfte fehlgeschlagener Anfragen ebenso viel wie 10 Minuten Totalausfall. Das ist ungefähr ein Viertel des Budgets für 28 Tage.

Mit einem nach Status gekennzeichneten Anfragezähler lässt sich der Indikator für das Zeitfenster in Prometheus als ein Verhältnis ausdrücken, nach dem Vorbild des Workbook-Beispiels:

```
sum(rate(http_requests_total{job="checkout", status!~"5.."}[28d]))
  /
sum(rate(http_requests_total{job="checkout"}[28d]))
```

Auf dieser Website steht die Seite zu Service Levels auf der Vertragsseite dieser Trennlinie. Für ein System, das das Unternehmen im Auftrag eines Kunden betreibt, stehen Reaktionszeit bei Vorfällen, abgedeckte Zeiten, geforderte Verfügbarkeit und Wartungsfenster in der Vereinbarung für dieses System. Sie werden pro System festgelegt, weil ein nächtlicher Bericht und ein Zahlungsdienst unterschiedliche Antworten brauchen. Die eigene Fehlerüberwachung und das Tracing, die die Gründerin im Rahmen ihrer aktuellen Arbeit für NextMariner aufgebaut hat, sind über mehrere Ansichten für den Betrieb zugänglich, darunter eine für Service Level Objectives. Und der Beitrag zum genau einmaligen Verkauf der letzten Eintrittskarte macht aus einer Bestandsinvariante ein solches Ziel: regelmäßig prüfen, als Diagramm darstellen und bei jeder Abweichung die Bereitschaft alarmieren. So wird das Ausbleiben von Überverkäufen zu einem Ziel mit Alarmierung statt zu einer Hoffnung.

Wo es schiefgeht

Der Zielwert stammt aus dem Dashboard des letzten Monats. Den heutigen Istwert zu übernehmen fühlt sich sicher an. Das SRE-Buch warnt jedoch: Ein unüberlegt gewähltes Ziel kann ein Team zu heldenhaften Anstrengungen zwingen, es weiter zu erfüllen, obwohl das System ohne Neuentwurf nicht besser werden kann. Leiten Sie das Ziel von den Bedürfnissen der Nutzer her und schreiben Sie die Begründung neben die Zahl. Das Workbook verlangt die Begründung und die Angabe, ob die Werte aus Experimenten oder

Beobachtungen stammen, selbst wenn es ausdrücklich Ad-hoc-Werte sind. Prüfen Sie das Dokument anfangs monatlich und nach der Stabilisierung vierteljährlich.

Das Ziel lautet 100 % oder besteht aus Neunen, deren Kosten niemand berechnet hat. Ein immer verfügbarer oder unbegrenzt skalierender Dienst ist teuer zu bauen und zu betreiben und besser, als Nutzer es wahrnehmen würden. Beginnen Sie mit einem lockeren Ziel und verschärfen Sie es mit wachsender Systemkenntnis. Das Buch zieht dies einem strengen Ziel vor, das später gelockert werden muss, weil es sich als unerreichbar erwiesen hat.

Ein Mittelwert soll das Nutzererlebnis darstellen. Im Buchbeispiel dauert eine typische Anfrage etwa 50 ms, während 5 % der Anfragen zwanzigmal so lange dauern. Ein Diagramm des Mittelwerts zeigt den ganzen Tag keinerlei Auffälligkeit. Formulieren Sie Latenzziele als Perzentile oder als Anteil von Anfragen unter einer Schwelle, etwa 300 ms, niemals als Mittelwert.

Es gibt mehr Ziele, als jemand begründen kann. Ein Dutzend SLOs pro Dienst verwässert sie alle. Der Test des Buchs ist unverblümt: Kann das Zitieren eines Ziels nie einen Streit über Prioritäten entscheiden, lohnt es sich vermutlich nicht. Behalten Sie die wenigen Ziele, die das tatsächliche Handeln der Nutzer abdecken.

Ein verfehltes Ziel ändert nichts. Das Workbook nennt das ein zahnloses SLO: Die Einhaltung wird zu einem weiteren KPI im Bericht statt zu einem Entscheidungsgrund. Vereinbaren Sie vorab, was nach Verbrauch des Budgets geschieht, und lassen Sie Produktmanager, Entwickler und SREs unterschreiben. Die beispielhafte Fehlerbudgetrichtlinie des Workbooks setzt gewöhnliche Änderungen aus, solange der Dienst sein Ziel verfehlt. Erlaubt bleiben nur Fehlerbehebungen höchster Priorität und Sicherheitsarbeiten. Sie sagt ausdrücklich, dass die Richtlinie keine Strafe ist.

SLO und SLA nennen dieselbe Zahl. Entspricht das interne Ziel dem Vertrag, ist die Vertragsverletzung das erste Warnsignal für das Team. Das Buch empfiehlt ein strengeres internes Ziel als das nach außen zugesagte, als Puffer für chronische Probleme. Auch der umgekehrte Fehler zählt: Ein Dienst, der sein Ziel deutlich übertrifft, lehrt Nutzer, sich darauf zu verlassen. Das Buch empfiehlt deshalb gelegentliche geplante Ausfallzeiten, Drosselung oder ein System, das unter geringer Last nicht schneller wird.

Das Zeitfenster ist länger als der Datenbestand. Ein Ziel über 28 Tage auf einem Prometheus-Server, der ohne eingestellte Aufbewahrung standardmäßig 15 Tage Messwerte behält, misst unbemerkt 15 Tage und meldet sie als vier Wochen. Stellen Sie eine Aufbewahrungsdauer jenseits des Fensters ein, bevor Sie der Zahl vertrauen:

```
prometheus --storage.tsdb.retention.time=35d
```

Weiterführende Lektüre

- Service Level Objectives — Kapitel 4 in Googles Buch Site Reliability Engineering: Indikatoren, Ziele und Vereinbarungen sowie die Wahl eines Zielwerts.
- Implementing SLOs — Das Kapitel des SRE Workbooks über SLIs als Verhältnisse, Fehlerbudgets, Zeitfenster und die schriftliche Festlegung des Ziels.
- Example Error Budget Policy — Eine vollständige Richtlinie für einen fiktiven Spieledienst: Wann Veröffentlichungen stoppen und wer bei Uneinigkeit entscheidet.
- Embracing Risk — Warum Google Verfügbarkeit anhand erfolgreicher Anfragen statt der Betriebszeit misst und was jede weitere Zuverlässigkeitsstufe kostet.
- Availability Table — Wie viel Ausfallzeit jede Verfügbarkeitsstufe pro Jahr, Quartal, Monat, Woche, Tag und Stunde erlaubt.
- Storage — Wie Prometheus Messwerte auf der Platte speichert und welche Aufbewahrungsoptionen den Rückblick einer SLO-Abfrage bestimmen.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein Service Level Objective?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/service-level-objective/