Engineer

Service level objective

◎ et målbart pålidelighedsmål, en tjeneste skal leve op til

Hvad det er

En service level objective er et mål for, hvor godt en tjeneste opfører sig, formuleret så præcist, at enhver kan efterprøve, om det blev nået: en andel af forespørgslerne, der skal lykkes eller besvares hurtigt nok, over en bestemt periode. I daglig tale hedder den næsten altid SLO, og på dansk ses også serviceniveaumål. Googles bog om Site Reliability Engineering definerer den som en målværdi eller et interval af værdier for ét niveau af service, målt med en indikator.

Indikatoren er en service level indicator, en SLI: en omhyggeligt defineret størrelse, der beskriver én side af den service, brugerne faktisk får. Latenstid er det sædvanlige første valg, med fejlrate og gennemløb lige efter. SRE Workbook anbefaler at skrive hver SLI som en brøk, gode hændelser divideret med alle hændelser, så den går fra 0 %, hvor intet virker, til 100 %, hvor intet er i stykker. Selve målet er en streg på den skala: indikatoren skal holde sig på den rigtige side af en grænse eller mellem en nedre og en øvre grænse.

En komplet SLO nævner alt, hvad der skal til for at efterprøve den. Bogens eksempel lyder: 99 % af Get RPC-kaldene, som gennemsnit over ét minut, afsluttes på under 100 ms, målt på tværs af alle backend-servere. Én indikator kan bære flere mål på én gang, for eksempel 90 % af kaldene under 1 ms, 99 % under 10 ms og 99,9 % under 100 ms, fordi én percentil sjældent beskriver en hel fordeling. SRE Workbook skelner også mellem specifikationen, det udfald, der betyder noget for brugerne, og implementeringen, stedet hvor det måles: en serverlog, en prober, der kører en browser, eller kode i selve siden. Hvert sted ser sine egne fejl, og valget hører hjemme i definitionen.

En SLO er ikke en SLA. En service level agreement er en aftale med brugerne, udtrykkelig eller underforstået, der knytter konsekvenser til at nå eller misse de mål, den indeholder, typisk en rabat eller en bod. Bogens test er at spørge, hvad der sker, når målet ikke nås: følger der ingen udtrykkelig konsekvens, er det en SLO. En fodnote tilføjer, at folk, der taler om en brudt SLA, næsten altid mener en misset SLO, for et egentligt brud på en SLA kan ende i retten som kontraktbrud. Google Search har ifølge bogen ingen SLA over for offentligheden, men når søgningen er utilgængelig, koster det stadig omdømme og annonceindtægter.

Brug begrebet i en beslutning

Brug »Hvad er en service level objective?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Overvågning og alarmer, der siger til, mens fejlen stadig er lille, og tallene, der viser, hvor tit den bliver ved med at være det.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har bygget det første observerbarhedsdashboard — virksomhedens første overblik over sine egne systemer

Har bygget det første observerbarhedsdashboard Kontakt os →

Hvorfor det betyder noget

Uden et udtalt mål laver brugerne deres eget. SRE-bogen beskriver begge måder, det går galt på: folk, der stoler mere på en tjeneste, end de burde, og mulige brugere, der tror, den er mere ustabil, end den er, og derfor holder sig væk. Eksemplet er Chubby, Googles låsetjeneste til distribuerede systemer. Globale udfald af den var så sjældne, at andre tjenester stille og roligt kom til at regne med, at den aldrig fejlede, og de tjenester gik i stykker de få gange, den gjorde. Løsningen var at lade den globale instans nå sin SLO uden at overgå den ret meget: i hvert kvartal, hvor rigtige fejl ikke havde trukket tilgængeligheden under målet, tog SRE-teamet med vilje systemet ned.

Målet afgør også diskussionen om, hvor pålideligt pålideligt nok er. SRE Workbook kalder 100 % pålidelighed det forkerte mål, og begrundelserne er jordnære. Komponenter fejler samtidig, uanset hvordan de er bygget. Kæden af netværk og enheder mellem tjenesten og kunderne fejler også, så kunderne ville aldrig opleve de 100 %. Hvert ekstra 9-tal koster mere end det forrige, mens værdien for brugerne nærmer sig nul. Og da ændringer er den største kilde til udfald, kan en tjeneste, der aldrig må fejle, heller aldrig opdateres. Når målet først ligger under 100 %, er afstanden op til perfektion plads, der kan bruges på udgivelser, eksperimenter og planlagt vedligeholdelse. Den plads hedder et error budget, på dansk også fejlbudget, og den er blot 100 % minus SLO'en. Nogen

skal eje afvejningen mellem nye funktioner og stabil drift; SRE Workbook peger på CTO'en i en lille organisation og på product owner eller product manager i en større.

Endelig gør målet overvågning til beslutninger. Bogen beskriver en løkke: man måler indikatorerne, holder dem op mod målene, afgør, om noget skal gøres, og gør det. Stiger latenstiden og vil den krydse grænsen inden for et par timer, undersøger teamet, om serverne er CPU-bundne, og sætter flere i drift. Uden målet, påpeger bogen, kan ingen sige, om der skal handles, eller hvornår.

Sådan ser det ud i praksis

Tag et checkout-API og en SLO skrevet, som SRE Workbook foreslår, med indikatoren adskilt fra stedet, hvor den måles, vinduet angivet og begrundelsen skrevet ned ved siden af:

```
Tjeneste:     checkout-API
SLI:          andel af POST /checkout-forespørgsler besvaret uden
              5xx-status, talt i serverens requestlog
Mål:          99,9 % over et glidende vindue på 28 dage
Begrundelse:  observeret, rundet ned fra fire ugers data
Ejer:         product manager sammen med vagtholdet
Gennemgang:   månedligt, indtil målet har sat sig, derefter kvartalsvis
```

Vinduet betyder lige så meget som procenten. SRE Workbook foretrækker et glidende vindue frem for en kalendermåned, fordi brugerne ikke glemmer et udfald på månedens sidste dag, blot fordi en ny måned begynder, og et helt antal uger, så hvert vindue rummer lige mange weekender; fire uger er dens valg til almindeligt brug. Det vindue er 28 × 24 × 60 = 40.320 minutter, og 0,1 % af det er godt 40 minutter med fuldt udfald. På en måned med 30 dage giver de samme 99,9 % plads til 43,2 minutter, tallet i bogens tilgængelighedstabel.

Tid er dog den grove måde at tælle på. SRE-bogen forklarer, at Google måler tilgængelighed som andelen af forespørgsler, der lykkes, frem for som oppetid, fordi en tjeneste, der er spredt over hele verden, sjældent er helt oppe eller helt nede. Talt i forespørgsler lyder eksemplet i SRE Workbook sådan: ved 99,9 % giver 3 millioner forespørgsler over fire uger plads til 3.000 fejl, og et udfald, der giver 1.500 af dem, bruger halvdelen af budgettet. At tælle forespørgsler sætter

også pris på de delvise fejl, som oppetid overser. Ved jævn trafik koster 20 minutter, hvor halvdelen af alle forespørgsler fejler, lige så meget som 10 minutters fuldt udfald, og det er omtrent en fjerdedel af budgettet for 28 dage.

I Prometheus, med en tæller for forespørgsler, der har status som label, er indikatoren over vinduet én brøk, bygget efter eksemplet i SRE Workbook:

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

Her på sitet ligger siden om serviceniveau på aftalens side af stregen. For et system, virksomheden driver for en kunde, skrives svartiden ved en driftshændelse, de dækkede timer, den tilgængelighed, systemet skal leve op til, og hvornår der må laves vedligeholdelse, ind i aftalen for netop det system. Tallene fastsættes system for system, fordi en natlig rapport og en betalingsløsning kræver forskellige svar. Den egenudviklede fejlovervågning og tracing, som grundlæggeren har bygget i sit aktuelle arbejde hos NextMariner, vises gennem en række operatørvisninger, heriblandt én til service level objectives. Og noten om at sælge den sidste billet præcis én gang gør en lagerinvariant til et serviceniveaumål: kontrollér den efter en tidsplan, vis den i en graf og alarmér ved enhver afvigelse, så nul oversalg er et mål med en alarm bag sig og ikke et håb.

Hvor det går galt

Målet er kopieret fra sidste måneds dashboard. At tage det, systemet klarer i dag, føles trygt, men SRE-bogen advarer om, at et mål, der vælges uden eftertanke, kan låse et team fast i heroiske anstrengelser for at blive ved med at nå det, på et system, der ikke kan blive bedre uden et redesign. Arbejd baglæns fra det, brugerne har brug for, og skriv begrundelsen ned ved siden af tallet: SRE Workbook beder om rationalet og om, hvorvidt tallene kom fra forsøg eller observation, også når de ærligt talt er ad hoc. Gennemgå dokumentet månedligt i starten og kvartalsvis, når det har sat sig.

Målet er 100 %, eller en række 9-taller, ingen har regnet prisen på. Et system, der altid er tilgængeligt eller skalerer uden grænser, er dyrt at bygge og drive og bedre, end brugerne ville bemærke. Begynd med et løst mål, og stram det,

efterhånden som I lærer systemet at kende; bogen foretrækker det frem for et stramt mål, der senere må lempes, fordi det viste sig uopnåeligt.

Et gennemsnit står i stedet for det, brugerne mærker. I bogens eksempel tager en typisk forespørgsel omkring 50 ms, mens 5 % af forespørgslerne tager tyve gange så lang tid, og overvågning af gennemsnittet alene ville ikke vise nogen ændring i løbet af dagen. Skriv latensmål som percentiler eller som en andel af forespørgsler under en grænse som 300 ms, aldrig som et gennemsnit.

Der er flere mål, end nogen kan forsvare. Et dusin SLO'er pr. tjeneste udvander dem alle. Bogens test er kontant: kan det aldrig vinde en diskussion om prioriteter at henvise til et mål, er målet nok ikke værd at have. Behold de få, der dækker det, brugerne faktisk gør.

Et misset mål får ingen følger. SRE Workbook kalder det en SLO uden tænder: overholdelsen bliver blot ét KPI mere i en rapport i stedet for en grund til at beslutte noget. Aftal på forhånd, hvad der sker, når budgettet er brugt, og få product manageren, udviklerne og SRE'erne til at skrive under. Eksemplet på en error budget policy i SRE Workbook stopper alle ændringer undtagen fejlrettelser med højeste prioritet og sikkerhedsrettelser, indtil tjenesten igen er inden for sit mål, og siger rent ud, at politikken ikke er en straf.

SLO'en er det samme tal som SLA'en. Når det interne mål svarer til kontrakten, er bruddet det første, teamet hører om problemet. Bogen anbefaler et strammere internt mål end det, der loves udadtil, som buffer mod kroniske problemer. Den modsatte fejl tæller også: en tjeneste, der kører langt bedre end sit mål, lærer brugerne at regne med det, så bogen foreslår lejlighedsvis planlagt nedetid, throttling eller et system, der ikke er hurtigere under let belastning.

Vinduet er længere end dataene. En SLO på 28 dage, der beregnes på en Prometheus-server, som gemmer 15 dages samples, standarden når der ikke er sat nogen retention, måler i stilhed 15 dage og rapporterer dem som fire uger. Sæt retention længere end vinduet, før tallet bruges til noget:

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

Læs mere

- Service Level Objectives — kapitel 4 i Googles bog om Site Reliability Engineering: indikatorer, mål og aftaler, og hvordan man vælger et mål.
- Implementing SLOs — SRE Workbooks kapitel om SLI'er som brøker, error budgets, vinduer og om at skrive målet ned.
- Example Error Budget Policy — en komplet politik for en fiktiv spiltjeneste: hvornår udgivelser stopper, og hvem der afgør en uenighed.
- Embracing Risk — hvorfor Google måler tilgængelighed i lykkede forespørgsler frem for oppetid, og hvad hvert trin af pålidelighed koster.
- Availability Table — hvor meget nedetid hvert niveau af tilgængelighed tillader pr. år, kvartal, måned, uge, dag og time.
- Storage — hvordan Prometheus gemmer samples på disk, og de retention-flag, der afgør, hvor langt tilbage en SLO-forespørgsel kan se.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er en service level objective?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/service-level-objective/