Engineer

Error budget

den andel fejl, et pålidelighedsmål stadig tillader

Hvad det er

Et error budget er den mængde fejl, en tjeneste har lov til at have i en given periode, regnet ud fra dens service level objective. Siger målet, at 99,99 % af forespørgslerne skal lykkes i dette kvartal, er de resterende 0,01 % budgettet: forespørgsler, der må fejle, ved et uheld eller fordi nogen tog en risiko med vilje, uden at nogen har brudt et løfte. På dansk kaldes det også et fejlbudget. Det tælles i samme enhed som målet, som regel fejlede forespørgsler, og det bliver brugt i samme øjeblik, som en bruger møder en fejl.

Idéen kommer fra Googles bog om Site Reliability Engineering. I kapitlet Embracing Risk fastsætter produktledelsen målet, overvågningssystemet optræder som en neutral tredjepart, der måler, hvad der faktisk skete, og forskellen mellem de to er den upålidelighed, der stadig er tilbage i kvartalet. Produktudviklingen og SRE aftaler budgettet sammen og på forhånd. Bogens eksempel ligger i den krævende ende af skalaen: en tjeneste, der skal besvare 99,999 % af forespørgslerne, har et budget på 0,001 %, så et enkelt problem, der får 0,0002 % af kvartalets forespørgsler til at fejle, har brugt en femtedel af det.

Budgettet spørger ikke, hvis skyld en fejl var. Bogen siger udtrykkeligt, at et netværksudfald eller et datacenter, der svigter, tærer på det præcis som en dårlig udgivelse, og at resten af kvartalet så kan byde på færre udrulninger. Brugerne mærkede fejlen, så den tæller. Hvem der skal handle på den, er et andet spørgsmål, og svaret hører hjemme i en skriftlig politik, der er aftalt, før noget går galt.

Brug begrebet i en beslutning

Brug »Hvad er et error budget?« 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

De, der bygger nye funktioner, og de, der holder en tjeneste kørende, vil forskellige ting, og uden et fælles tal vinder den, der forhandler bedst. SRE-bogen siger, at den slags beslutninger hverken bør hvile på politik, frygt eller håb, og minder om Google SRE's uofficielle motto: »Hope is not a strategy.« Budgettet erstatter diskussionen med et tal, som begge parter har accepteret på forhånd. Så længe der er budget tilbage, går udgivelser ud; når det er brugt, holder de pause, mens kræfterne går til test og robusthed i stedet. Bogen beskriver også blidere udgaver af samme mekanisme, for eksempel at sætte tempoet ned eller rulle udgivelser tilbage, efterhånden som budgettet nærmer sig bunden, og kalder i en fodnote den rene tænd-og-sluk-udgave for bang/bang-styring.

Den mere interessante virkning er på adfærden. Når budgettet er stort, kan udviklerne tage chancer: udrulle oftere, prøve eksperimentet. Når det næsten er tømt, bemærker bogen, beder udviklerne selv om mere test og langsommere udrulninger, fordi ingen har lyst til at stoppe deres egen lancering. Teamet holder selv justits. Bogen stiller én betingelse: det virker kun, hvis SRE reelt har myndighed til at stoppe lanceringer, når målet er brudt.

Det er mest udgivelser, budgettet styrer, fordi det mest er udgivelser, der ødelægger noget: eksempelpolitikken i SRE Workbook anslår, at ændringer står bag omkring 70 % af Googles nedbrud. Det gør budgettet til et kompas for, hvor indsatsen bør lægges. Workbooks kapitel om at implementere SLO'er regner på en tjeneste med et mål på 97 % og et budget for fire uger på 109.897 fejlede forespørgsler. En dårlig API-udgivelse, der fik alle forespørgsler til at fejle i fire timer, kostede 14.066 af dem, omkring 13 %. En databaseserver, der døde og tog 20 timer at gendanne fra backup, kostede anslået 72.000, eller 65 %. Serverfejlen ser langt værre ud, men virksomheden i eksemplet oplever én af dem på fem år mod to eller tre dårlige udgivelser om året, så i gennemsnit koster udgivelserne omkring dobbelt så meget budget. Det betaler sig først at rette udgivelsesprocessen.

Et budget, der aldrig bliver brugt, siger også noget. Bogen påpeger, at det afslører, hvad et for højt mål koster i stivhed og langsom innovation, og at et team, der har svært ved at få noget lanceret, kan vælge at løsne målet og dermed gøre budgettet større.

Sådan ser det ud i praksis

Forbruget har en fart, ikke kun en sum. Workbooks kapitel om alarmering kalder farten for burn rate: hvor hurtigt tjenesten bruger sit budget i forhold til målet. Ved en burn rate på 1 er budgettet brugt præcis ved periodens udgang. For et mål på 99,9 % over 30 dage ser Workbooks tabel sådan ud:

```
Burn rate   Fejlrate   Budgettet er brugt efter
1           0,1 %      30 dage
2           0,2 %      15 dage
10          1 %        3 dage
1.000       100 %      43 minutter
```

At alarmere på burn rate er bedre end at kalde nogen ud, hver gang fejlraten passerer 0,1 %, for så vækkes nogen ved hver kortvarig spids. Workbook anbefaler at holde øje med flere burn rates på én gang, hver over et langt og et kort vindue. Dens udgangspunkt for et mål på 99,9 % kalder ud, når 2 % af 30-dagesbudgettet går på én time (en burn rate på 14,4) eller 5 % går på seks timer (en burn rate på 6), og opretter en sag, når 10 % går på tre dage (en burn rate på 1). Det korte vindue, en tolvtedel af det lange, får alarmen til at stoppe kort efter, at problemet gør det. I Workbooks eksempel falder gennemsnittet i det korte vindue under tærsklen fem minutter efter, at fejlene stopper, mens timegennemsnittet bliver over den i en hel time.

Som Prometheus-regelfil ser de to udkaldsbetingelser sådan ud. Forholdstallene er recording rules, navngivet sådan som Workbook navngiver dem, og hver af dem rummer andelen af fejlede forespørgsler over det vindue, navnet angiver:

```
groups:
  - name: orders-error-budget
    rules:
      - alert: OrdersErrorBudgetBurn
        expr: |
          (
            job:slo_errors_per_request:ratio_rate1h{job="orders"}
              > (14.4 * 0.001)
```

```
            and
            job:slo_errors_per_request:ratio_rate5m{job="orders"}
              > (14.4 * 0.001)
          )
          or
          (
            job:slo_errors_per_request:ratio_rate6h{job="orders"}
              > (6 * 0.001)
            and
            job:slo_errors_per_request:ratio_rate30m{job="orders"}
              > (6 * 0.001)
          )
        labels:
          severity: page
```

Følg én dårlig udrulning gennem reglen. Klokken 14.00 begynder en udgivelse at få 2 % af ordreforespørgslerne til at fejle, en burn rate på 20. Femminutterstallet passerer stregen ved 1,44 % inden for fire minutter, men timetallet når den først, når fejlene har stået på i omkring 43 minutter, så udkaldet går ud omkring klokken 14.43. På det tidspunkt er præcis 2 % af 30-dagesbudgettet brugt, og det er den byttehandel, reglen laver: et kort udsving vækker ingen, og et vedvarende forbrug kan ikke gemme sig. Rul tilbage klokken 14.50, og femminutterstallet falder under stregen inden for to minutter og afslutter udkaldet; timetallet alene ville have holdt alarmen i gang til omkring klokken 15.07.

At holde udkald i et rimeligt forhold til problemet er samme hensyn, som ligger bag virksomhedens egen fejlovervågning og tracing fra 2025. Den opdager spikes og regressioner med en cooldown, så én dårlig udrulning ikke kalder nogen ud fyrre gange, dens alarmregler er skræddersyet til netop den platform og ikke til en generisk, og én af dens 11 operatørvisninger er afsat til service level objectives.

Hvor det går galt

At tælle en fejl forveksles med at give nogen skylden. Alt, hvad brugerne mærker, bruger af budgettet, men ikke alt bør stoppe teamets egne udgivelser. Eksempelpolitikken i Workbook lader arbejdet med nye funktioner fortsætte, når et netværksproblem i hele virksomheden var årsagen, eller når et andet teams tjeneste var det, og det team selv har stoppet sine udgivelser; har teamets egen kode eller procedure brugt budgettet, kommer pålideligheden først. For et nedbrud hos et andet team, man er afhængig af, beskriver kapitlet om at implementere SLO'er to skoler, fortsæt eller frys alligevel, og siger, at en frysning

gør brugerne gladere. Vælg på forhånd, skriv valget ind i politikken, og udpeg, hvem der afgør en uenighed om regnestykket; eksempelpolitikken sender den til CTO'en.

Fejl havner i den forkerte kategori. En belastningstest, der fejler, eller en penetrationstesters afviste forespørgsler bruger af budgettet, selv om ingen bruger blev ramt, og eksempelpolitikken nævner forkert kategoriserede fejl som en måde, budget forsvinder på uden grund. Det omvendte er værre: rigtige fejl, der registreres som noget harmløst, holder tjenesten inden for målet på papiret, mens brugerne lider, og samme politik gør pålidelighedsarbejde obligatorisk, når det bliver opdaget. Hold trafik, som målet ikke dækker, ude af indikatoren, og efterprøv klassificeringen efter enhver hændelse, hvis pris så forkert ud.

En stor hændelse bliver betalt og glemt. En frysning køber tid, ikke forståelse. Eksempelpolitikken binder de to sammen: en enkelt hændelse, der bruger mere end 20 % af budgettet for fire uger, kræver en postmortem med mindst én opfølgning af højeste prioritet, og én type nedbrud, der bruger mere end 20 % i et kvartal, skal stå på planen for næste kvartal.

Alarmen kan aldrig gå. Tærskler, der er kopieret fra tabellen for 99,9 %, passer ikke til andre mål. Ved 90 % tilgængelighed, påpeger Workbook, bruger et totalt nedbrud kun 1,4 % af budgettet på en time, så en regel, der kalder ud ved 2 % i timen, forbliver tavs hele vejen igennem. I den anden yderlighed er et månedligt mål på 99,999 % opbrugt af et totalt nedbrud på 26 sekunder, hurtigere end mange overvågningssystemer overhovedet indsamler en måling. Tilpas vinduer og tærskler til målet, og accepter ved meget høje mål Workbooks konklusion, at kun designet kan forsvare dem: rul en ændring ud til 1 % af brugerne først, så giver den samme fejl 43 minutter, før budgettet er væk.

Lav trafik gør én fejl til et udkald. Ved ti forespørgsler i timen er én fejlet forespørgsel en fejlrate på 10 % for den time. Mod et mål på 99,9 %, bemærker Workbook, udløser den ene forespørgsel straks et udkald, fordi den bruger 13,9 % af 30-dagesbudgettet: hele måneden tillader kun syv fejl. Workbooks løsninger er at skabe kunstig trafik, at samle flere små tjenester til én i overvågningen eller at ændre produktet, så én fejlet forespørgsel betyder mindre.

Målet hæves, før tjenesten kan nå det. Et team, der har brug for et strammere mål og indfører det med det samme, lever i permanent undtagelsestilstand med et budget, der altid er brugt. Workbook foreslår i stedet et ambitiøst mål: målt

og fulgt ved siden af det nuværende, men beskrevet i politikken som et mål, der ikke kræver handling, før tjenesten er klar til det.

Læs mere

- Embracing Risk — kapitel 3 i Googles bog om Site Reliability Engineering, hvor error budget introduceres som en måde at afgøre udgivelser på med data.
- Alerting on SLOs — SRE Workbook om burn rates, alarmer med flere vinduer og tjenester med lav trafik.
- Example Error Budget Policy — en komplet politik: hvornår udgivelser stopper, hvilke nedbrud der er undtaget, og hvornår der kræves en postmortem.
- Implementing SLOs — hvordan man regner et budget ud fra et mål og bruger det til at vælge mellem investeringer i pålidelighed.
- Alerting rules — Prometheus-dokumentationen for det regelformat, der er brugt ovenfor.

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 et error budget?« 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/error-budget/