Engineer

Feilbudsjett

andelen feil et pålitelighetsmål fortsatt tillater

Hva det er

Et feilbudsjett er hvor mye svikt en tjeneste har lov til å ha i en gitt periode, regnet ut fra tjenestenivåmålet. Sier målet at 99,99 % av forespørslene skal lykkes dette kvartalet, er de resterende 0,01 % budsjettet: forespørsler som kan feile, ved et uhell eller fordi noen tok en bevisst risiko, uten at noen har brutt et løfte. På engelsk heter det error budget. Budsjettet telles i samme enhet som målet, som regel feilede forespørsler, og det brukes i det øyeblikket en bruker møter en feil.

Idéen kommer fra Googles bok om Site Reliability Engineering. I kapittelet Embracing Risk setter produktledelsen målet, overvåkingssystemet opptrer som en nøytral tredjepart som måler hva som faktisk skjedde, og avstanden mellom de to er den upåliteligheten som fortsatt er igjen av kvartalet.
Produktutviklingen og SRE avtaler budsjettet sammen, på forhånd. Bokens eksempel ligger i den krevende enden av skalaen: en tjeneste som skal besvare 99,999 % av forespørslene, har et budsjett på 0,001 %, så ett enkelt problem som får 0,0002 % av kvartalets forespørsler til å feile, har brukt opp en femtedel av det.

Budsjettet spør ikke om hvem som hadde skylden. Boken sier uttrykkelig at et nettverksbrudd eller et datasenter som svikter, tærer på det akkurat som en dårlig utgivelse, og at resten av kvartalet da kan by på færre utrullinger. Brukerne merket feilen, så den teller. Hvem som må handle på den, er et annet spørsmål, og svaret hører hjemme i en skriftlig policy som er avtalt før noe går galt.

Bruk begrepet i en beslutning

Bruk «Hva er et feilbudsjett?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Overvåking og varsler som sier fra mens en feil fortsatt er liten, og tallene som viser hvor ofte den holder seg liten.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Bygde selskapets første observerbarhetsdashbord — selskapets første innsyn i egne systemer

Bygde selskapets første observerbarhetsdashbord Ta kontakt →

Hvorfor det er viktig

De som lager nye funksjoner, og de som holder en tjeneste i drift, vil ulike ting, og uten et felles tall vinner den som forhandler best. SRE-boken sier at slike beslutninger verken bør hvile på politikk, frykt eller håp, og minner om Google SREs uoffisielle motto: «Hope is not a strategy.» Budsjettet erstatter diskusjonen med et tall begge parter har godtatt på forhånd. Så lenge det er budsjett igjen, går utgivelser ut; når det er brukt opp, tar de pause mens innsatsen går til testing og robusthet i stedet. Boken beskriver også mildere varianter av samme mekanisme, for eksempel å senke tempoet eller rulle tilbake utgivelser etter hvert som budsjettet nærmer seg null, og kaller i en fotnote den rene av-eller-på-varianten for bang/bang-styring.

Den mer interessante virkningen er på atferden. Når budsjettet er stort, kan utviklerne ta sjanser: rulle ut oftere, prøve eksperimentet. Når det nesten er tomt, påpeker boken, ber utviklerne selv om mer testing og langsommere utrulling, fordi ingen vil stoppe sin egen lansering. Teamet blir selvregulerende. Boken setter én betingelse: dette virker bare hvis SRE faktisk har myndighet til å stanse lanseringer når målet er brutt.

Det er mest utgivelser budsjettet styrer, fordi det mest er utgivelser som ødelegger noe: eksempelpolicyen i SRE Workbook anslår at endringer står bak omtrent 70 % av Googles driftsavbrudd. Dermed blir budsjettet et kompass for hvor innsatsen bør gå. Workbooks kapittel om å innføre SLO-er regner på en tjeneste med et mål på 97 % og et budsjett for fire uker på 109 897 feilede forespørsler. En dårlig API-utgivelse som fikk alle forespørsler til å feile i fire timer, kostet 14 066 av dem, omtrent 13 %. En databaseserver som døde og tok 20 timer å gjenopprette fra sikkerhetskopi, kostet anslagsvis 72 000, eller 65 %. Serverfeilen ser langt verre ut, men selskapet i eksempelet opplever én slik på fem år mot to eller tre dårlige utgivelser i året, så i snitt koster utgivelsene omtrent dobbelt så mye budsjett. Det lønner seg først å rette opp utgivelsesprosessen.

Et budsjett som aldri blir brukt, sier også noe. Boken påpeker at det avslører hva et for høyt mål koster i stivhet og treg innovasjon, og at et team som sliter med å få lansert noe, kan velge å løsne på målet og dermed gjøre budsjettet større.

Slik ser det ut i praksis

Forbruket har en fart, ikke bare en sum. Workbooks kapittel om varsling kaller denne farten burn rate, altså forbrukstakten: hvor raskt tjenesten bruker opp budsjettet sitt, sett i forhold til målet. Ved en burn rate på 1 er budsjettet brukt opp nøyaktig ved slutten av perioden. For et mål på 99,9 % over 30 dager ser Workbooks tabell slik ut:

```
Burn rate   Feilrate   Budsjettet er brukt opp etter
1           0,1 %      30 dager
2           0,2 %      15 dager
10          1 %        3 dager
1 000       100 %      43 minutter
```

Å varsle på burn rate er bedre enn å tilkalle noen hver gang feilraten passerer 0,1 %, for da blir noen vekket av hver kortvarige topp. Workbook anbefaler å følge med på flere burn rates samtidig, hver over et langt og et kort vindu. Utgangspunktet for et mål på 99,9 % tilkaller vakten når 2 % av 30-dagersbudsjettet går med på én time (en burn rate på 14,4) eller 5 % på seks timer (en burn rate på 6), og oppretter en sak når 10 % går med på tre dager (en burn rate på 1). Det korte vinduet, en tolvtedel av det lange, gjør at varselet stopper kort tid etter at problemet gjør det. I Workbooks eksempel faller snittet i det korte vinduet under terskelen fem minutter etter at feilene stanser, mens timesnittet blir liggende over den i en hel time.

Som Prometheus-regelfil ser de to betingelsene for å tilkalle vakten slik ut. Forholdstallene er recording rules, navngitt slik Workbook navngir dem, og hver av dem holder andelen feilede forespørsler over vinduet som står i navnet:

```
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 produksjonssetting gjennom regelen. Klokken 14.00 begynner en utgivelse å få 2 % av ordreforespørslene til å feile, en burn rate på 20. Andelen over fem minutter passerer streken på 1,44 % i løpet av fire minutter, men andelen over én time når den først når feilene har pågått i omkring 43 minutter, så vakten blir tilkalt omkring klokken 14.43. Da er nøyaktig 2 % av 30-dagersbudsjettet brukt, og det er byttehandelen regelen gjør: en kort topp vekker ingen, og et vedvarende forbruk kan ikke gjemme seg. Rull tilbake klokken 14.50, så faller andelen over fem minutter under streken innen to minutter og avslutter varselet; andelen over én time alene ville holdt det gående til omkring klokken 15.07.

Å holde varslingen i rimelig forhold til problemet er det samme hensynet som ligger bak selskapets egen feilovervåking og sporing fra 2025. Den oppdager topper og regresjoner med en avkjølingsperiode, slik at én dårlig produksjonssetting ikke varsler noen førti ganger, varslingsreglene passer denne plattformen i stedet for en generisk, og én av de 11 driftsvisningene er satt av til tjenestenivåmål.

Hvor det går galt

Å telle en feil blir forvekslet med å gi noen skylden. Alt brukerne merker, tærer på budsjettet, men ikke alt bør stanse teamets egne utgivelser. Eksempelpolicyen i Workbook lar arbeidet med nye funksjoner fortsette når et nettverksproblem i hele selskapet var årsaken, eller når et annet teams tjeneste var det og det teamet selv har stanset sine utgivelser; har teamets egen kode eller prosedyre brukt opp budsjettet, kommer påliteligheten først. For et driftsavbrudd som skyldes en avhengighet hos et annet team, beskriver kapittelet om å innføre SLO-er to skoler, fortsett eller frys likevel, og sier at en frysing gjør brukerne mer fornøyde. Velg på forhånd, skriv valget inn i policyen, og utpek hvem som avgjør en uenighet om regnestykket; eksempelpolicyen sender den til CTO-en.

Feil havner i gal kategori. En lasttest som feiler, eller en penetrasjonstesters avviste forespørsler tærer på budsjettet selv om ingen bruker ble rammet, og eksempelpolicyen nevner gal kategorisering som en måte budsjett forsvinner på uten grunn. Det motsatte er verre: ekte feil som blir registrert som noe harmløst, holder tjenesten innenfor målet på papiret mens brukerne lider, og samme policy gjør pålitelighetsarbeid obligatorisk når det blir oppdaget. Hold trafikk som målet ikke dekker, utenfor indikatoren, og kontroller klassifiseringen etter hver driftshendelse der prisen så feil ut.

En stor hendelse blir betalt og glemt. En frysing kjøper tid, ikke forståelse. Eksempelpolicyen knytter de to sammen: én enkelt hendelse som bruker mer enn 20 % av budsjettet for fire uker, krever en postmortem med minst ett tiltak av høyeste prioritet, og én type driftsavbrudd som bruker mer enn 20 % i et kvartal, skal stå på planen for neste kvartal.

Varselet kan aldri gå. Terskler som er kopiert fra tabellen for 99,9 %, passer ikke for andre mål. Ved 90 % tilgjengelighet, påpeker Workbook, bruker et totalt driftsavbrudd bare 1,4 % av budsjettet på en time, så en regel som tilkaller vakten ved 2 % i timen, forblir taus hele veien gjennom. I den andre ytterligheten er et månedlig mål på 99,999 % brukt opp av et totalt avbrudd på 26 sekunder, raskere enn mange overvåkingssystemer i det hele tatt samler inn en måling. Tilpass vinduer og terskler til målet, og godta for svært høye mål Workbooks konklusjon om at bare designet kan forsvare dem: rull en endring ut til 1 % av brukerne først, så gir den samme feilen 43 minutter før budsjettet er borte.

Lav trafikk gjør én feil til et varsel. Ved ti forespørsler i timen er én feilet forespørsel en feilrate på 10 % for den timen. Mot et mål på 99,9 %, påpeker Workbook, tilkaller den ene forespørselen vakten med en gang, fordi den bruker 13,9 % av 30-dagersbudsjettet: hele måneden tillater bare sju feil. Workbooks løsninger er å generere kunstig trafikk, å slå sammen flere små tjenester til én i overvåkingen eller å endre produktet slik at én feilet forespørsel betyr mindre.

Målet heves før tjenesten klarer det. Et team som trenger et strammere mål og innfører det med en gang, lever i permanent unntakstilstand med et budsjett som alltid er brukt opp. Workbook foreslår i stedet et ambisiøst mål: målt og fulgt ved siden av det gjeldende, men beskrevet i policyen som et mål som ikke krever handling før tjenesten er klar for det.

Les mer

- Embracing Risk — kapittel 3 i Googles bok om Site Reliability Engineering, der feilbudsjettet introduseres som en måte å avgjøre utgivelser på med data.
- Alerting on SLOs — SRE Workbook om burn rates, varsler med flere vinduer og tjenester med lite trafikk.
- Example Error Budget Policy — en fullstendig policy: når utgivelser stanser, hvilke driftsavbrudd som er unntatt, og når det kreves en postmortem.
- Implementing SLOs — hvordan man regner ut et budsjett fra et mål og bruker det til å velge mellom investeringer i pålitelighet.
- Alerting rules — Prometheus-dokumentasjonen for regelformatet som er brukt ovenfor.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er et feilbudsjett?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/error-budget/