Engineer

Felbudget

andelen fel som ett tillförlitlighetsmål ännu tillåter

Vad det är

En felbudget är hur mycket fel en tjänst får ha under en given period, uträknat från dess servicenivåmål. Säger målet att 99,99 % av förfrågningarna ska lyckas det här kvartalet, är de återstående 0,01 % budgeten: förfrågningar som får misslyckas, av misstag eller för att någon medvetet tog en risk, utan att någon har brutit ett löfte. På engelska heter det error budget. Budgeten räknas i samma enhet som målet, oftast misslyckade förfrågningar, och den förbrukas i samma ögonblick som en användare möter ett fel.

Idén kommer från Googles bok om Site Reliability Engineering. I kapitlet Embracing Risk sätter produktledningen målet, övervakningssystemet fungerar som en neutral tredje part som mäter vad som faktiskt hände, och skillnaden mellan de två är den otillförlitlighet som återstår av kvartalet. Produktutvecklingen och SRE kommer överens om budgeten tillsammans, i förväg. Bokens exempel ligger i den krävande änden av skalan: en tjänst som ska besvara 99,999 % av förfrågningarna har en budget på 0,001 %, så ett enda problem som får 0,0002 % av kvartalets förfrågningar att misslyckas har förbrukat en femtedel av den.

Budgeten frågar inte vems felet var. Boken säger uttryckligen att ett nätverksavbrott eller ett datacenter som fallerar tär på den precis som en dålig release, och att resten av kvartalet då kan få färre utrullningar. Användarna märkte felet, så det räknas. Vem som måste agera på det är en annan fråga, och svaret hör hemma i en skriftlig policy som är överenskommen innan något går fel.

Använd begreppet i ett beslut

Använd ”Vad är en felbudget?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Övervakning och larm som säger till medan ett fel fortfarande är litet, och siffrorna som visar hur ofta det förblir så.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Byggde företagets första observerbarhetsdashboard — företagets första överblick över de egna systemen

Byggde företagets första observerbarhetsdashboard Hör av er →

Varför det spelar roll

De som bygger nya funktioner och de som håller en tjänst i drift vill olika saker, och utan en gemensam siffra vinner den som förhandlar bäst. SRE-boken säger att sådana beslut varken bör vila på politik, rädsla eller hopp, och påminner om Google SRE:s inofficiella motto: ”Hope is not a strategy.” Budgeten ersätter diskussionen med en siffra som båda sidor har godtagit i förväg. Så länge det finns budget kvar går releaser ut; när den är förbrukad pausas de, och kraften läggs på test och motståndskraft i stället. Boken beskriver också mjukare varianter av samma mekanism, till exempel att sänka takten eller rulla tillbaka releaser i takt med att budgeten närmar sig noll, och kallar i en fotnot den rena av-eller-på-varianten för bang/bang-styrning.

Den intressantare effekten gäller beteendet. När budgeten är stor kan utvecklarna ta risker: rulla ut oftare, pröva experimentet. När den nästan är tom, konstaterar boken, ber utvecklarna själva om mer test och långsammare utrullning, eftersom ingen vill stoppa sin egen lansering. Teamet blir självreglerande. Boken ställer ett villkor: det fungerar bara om SRE verkligen har mandat att stoppa lanseringar när målet är brutet.

Det är mest releaser budgeten styr, eftersom det mest är releaser som förstör något: exempelpolicyn i SRE Workbook uppskattar att ändringar ligger bakom ungefär 70 % av Googles avbrott. Det gör budgeten till en kompass för var insatsen bör läggas. Workbooks kapitel om att införa SLO:er räknar på en tjänst med ett mål på 97 % och en budget för fyra veckor på 109 897 misslyckade förfrågningar. En dålig API-release som fick alla förfrågningar att misslyckas i fyra timmar kostade 14 066 av dem, ungefär 13 %. En databasserver som dog och tog 20 timmar att återställa från säkerhetskopia kostade uppskattningsvis 72 000, eller 65 %. Serverhaveriet ser mycket värre ut, men företaget i exemplet råkar ut för ett sådant på fem år mot två eller tre dåliga releaser om året, så i genomsnitt kostar releaserna ungefär dubbelt så mycket budget. Det lönar sig att börja med releaseprocessen.

En budget som aldrig förbrukas säger också något. Boken påpekar att den synliggör vad ett för högt mål kostar i stelhet och långsam innovation, och att ett team som har svårt att få ut något kan välja att sänka målet och därmed göra budgeten större.

Så ser det ut i praktiken

Förbrukningen har en hastighet, inte bara en summa. Workbooks kapitel om larm kallar hastigheten burn rate, alltså förbrukningstakten: hur snabbt tjänsten gör av med sin budget i förhållande till målet. Vid en burn rate på 1 tar budgeten slut precis när perioden gör det. För ett mål på 99,9 % över 30 dagar ser Workbooks tabell ut så här:

Burn rate	Felfrekvens	Budgeten är slut efter
1	0,1 %	30 dagar
2	0,2 %	15 dagar
10	1 %	3 dagar
1 000	100 %	43 minuter

Att larma på burn rate är bättre än att väcka någon varje gång felfrekvensen passerar 0,1 %, vilket skulle ske vid varje kortvarig topp. Workbook rekommenderar att bevaka flera burn rates samtidigt, var och en över ett långt och ett kort fönster. Dess utgångspunkt för ett mål på 99,9 % larmar jouren när 2 % av 30-dagarsbudgeten går åt på en timme (en burn rate på 14,4) eller 5 % på sex timmar (en burn rate på 6), och skapar ett ärende när 10 % går åt på tre dagar (en burn rate på 1). Det korta fönstret, en tolftedel av det långa, gör att larmet upphör strax efter att problemet gör det. I Workbooks exempel sjunker medelvärdet i det korta fönstret under tröskeln fem minuter efter att felen upphör, medan timmedelvärdet ligger kvar över den en hel timme.

Som Prometheus-regelfil ser de två larmvillkoren ut så här. Kvoterna är recording rules, namngivna på samma sätt som i Workbook, och var och en innehåller andelen misslyckade förfrågningar över fönstret som namnet anger:

```
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
```

Följ en dålig driftsättning genom regeln. Klockan 14.00 börjar en release få 2 % av orderförfrågningarna att misslyckas, en burn rate på 20. Andelen över fem minuter passerar gränsen på 1,44 % inom fyra minuter, men andelen över en timme når den först när felen har pågått i omkring 43 minuter, så jouren larmas omkring klockan 14.43. Då är exakt 2 % av 30-dagarsbudgeten förbrukad, och det är den avvägning regeln gör: en kort topp väcker ingen, och en ihållande förbrukning kan inte gömma sig. Rulla tillbaka klockan 14.50, så sjunker andelen över fem minuter under gränsen inom två minuter och larmet upphör; andelen över en timme ensam hade hållit det igång till omkring klockan 15.07.

Att hålla larmen i proportion till problemet är samma omsorg som ligger bakom företagets egen felövervakning och spårning från 2025. Den upptäcker toppar och regressioner med en karenstid, så att en dålig driftsättning inte larmar någon fyrtio gånger, larmreglerna passar just den plattformen i stället för en generisk, och en av dess 11 operatörsvyer är avsedd för servicenivåmål.

Var det går fel

Att räkna ett fel förväxlas med att skuldbelägga någon. Allt användarna märker tär på budgeten, men inte allt bör stoppa teamets egna releaser. Exempelpolicyn i Workbook låter arbetet med nya funktioner fortsätta när ett nätverksproblem i hela företaget var orsaken, eller när ett annat teams tjänst var det och det teamet självt har stoppat sina releaser; har teamets egen kod eller procedur förbrukat budgeten kommer tillförlitligheten först. För ett avbrott som orsakas av ett beroende hos ett annat team beskriver kapitlet om att införa

SLO:er två skolor, fortsätt eller frys ändå, och säger att frysningen gör användarna nöjdare. Välj i förväg, skriv in valet i policyn och utse vem som avgör en oenighet om räknestycket; exempelpolicyn skickar den till CTO:n.

Fel hamnar i fel kategori. Ett lasttest som misslyckas eller en penetrationstestares avvisade förfrågningar tär på budgeten fast ingen användare drabbades, och exempelpolicyn nämner felaktigt kategoriserade fel som ett sätt för budget att försvinna i onödan. Det omvända är värre: verkliga fel som registreras som något harmlöst håller tjänsten inom målet på papperet medan användarna lider, och samma policy gör tillförlitlighetsarbete obligatoriskt när det upptäcks. Håll trafik som målet inte omfattar utanför indikatorn, och granska klassificeringen efter varje incident vars kostnad såg fel ut.

En stor incident betalas och glöms. En frysning köper tid, inte förståelse. Exempelpolicyn knyter ihop de två: en enskild incident som förbrukar mer än 20 % av budgeten för fyra veckor kräver en postmortem med minst en åtgärd av högsta prioritet, och en typ av avbrott som förbrukar mer än 20 % under ett kvartal ska stå med i planen för nästa kvartal.

Larmet kan aldrig gå. Trösklar som kopierats från tabellen för 99,9 % passar inte för andra mål. Vid 90 % tillgänglighet, påpekar Workbook, förbrukar ett totalt avbrott bara 1,4 % av budgeten på en timme, så en regel som larmar vid 2 % i timmen förblir tyst hela tiden. I andra änden förbrukas ett månatligt mål på 99,999 % av ett totalt avbrott på 26 sekunder, snabbare än många övervakningssystem ens hinner samla in ett mätvärde. Anpassa fönster och trösklar efter målet, och godta för mycket höga mål Workbooks slutsats att bara designen kan försvara dem: rulla först ut en ändring till 1 % av användarna, så ger samma fel 43 minuter innan budgeten är slut.

Låg trafik gör ett enda fel till ett larm. Vid tio förfrågningar i timmen är en misslyckad förfrågan en felfrekvens på 10 % för den timmen. Mot ett mål på 99,9 %, påpekar Workbook, larmar den enda förfrågan genast, eftersom den förbrukar 13,9 % av 30-dagarsbudgeten: hela månaden tillåter bara sju fel. Workbooks lösningar är att skapa konstgjord trafik, att slå ihop flera små tjänster till en i övervakningen eller att ändra produkten så att en enda misslyckad förfrågan betyder mindre.

Målet höjs innan tjänsten klarar det. Ett team som behöver ett striktare mål och inför det direkt lever i ständigt undantagstillstånd med en budget som alltid är förbrukad. Workbook föreslår i stället ett ambitiöst mål: mätt och följt bredvid det nuvarande, men beskrivet i policyn som ett mål som inte kräver någon åtgärd förrän tjänsten är redo för det.

Läs mer

- Embracing Risk — kapitel 3 i Googles bok om Site Reliability Engineering, där felbudgeten introduceras som ett sätt att fatta releasebeslut med data.
- Alerting on SLOs — SRE Workbook om burn rates, larm med flera fönster och tjänster med lite trafik.
- Example Error Budget Policy — en komplett policy: när releaser stoppas, vilka avbrott som är undantagna och när det krävs en postmortem.
- Implementing SLOs — hur man räknar ut en budget från ett mål och använder den för att välja mellan investeringar i tillförlitlighet.
- Alerting rules — Prometheus dokumentation för regelformatet som används ovan.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är en felbudget?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/error-budget/