**Engineer**

# Postmortem

en skriftlig gennemgang af en driftshændelse og dens årsager

## Hvad det er

En postmortem er den skriftlige gennemgang, et team laver, når en driftshændelse er overstået: hvad der skete, hvad det kostede, hvad der blev gjort for at stoppe det, hvorfor det skete, og hvad der bliver ændret, så det ikke sker på samme måde igen. Ordet staves også post-mortem med bindestreg, og mange danske teams bruger det engelske udtryk uændret. Googles bog Site Reliability Engineering definerer den netop sådan: en skriftlig optegnelse af hændelsen, dens konsekvenser, de handlinger der afbødede eller løste den, de underliggende årsager og de opfølgende tiltag, der skal forhindre gentagelse.

Bogen giver øvelsen tre formål: at dokumentere hændelsen, at forstå alle de årsager, der bidrog til den, og ikke kun den første, man faldt over, og at få forebyggende tiltag på plads, som rent faktisk virker. Flertallet er vigtigt. I systemer uanset størrelse har en hændelse sjældent én årsag, og en gennemgang, der stopper ved den første plausible forklaring, ender typisk med at rette udløseren og lade de forhold stå, som gav udløseren mulighed for at gøre skade.

Det, der adskiller praksissen fra en almindelig rapport, er, at den er skyldfri, på engelsk blameless. En skyldfri postmortem leder efter de bidragende årsager uden at anklage nogen person eller noget team for dårlig opførsel, og den går ud fra, at alle involverede handlede i god tro og traf beslutninger ud fra de oplysninger, de havde på tidspunktet. SRE-bogen fører tanken tilbage til sundhedsvæsenet og luftfarten, hvor en fejl kan koste liv, og hvor hver fejl ses som en anledning til at gøre systemet stærkere. Begrundelsen er praktisk snarere end venlig: et menneske kan man ikke rette med en kodeændring, men et system eller en proces kan man.

Der skrives ikke en postmortem for hvert lille hik, for den koster tid. Bogen nævner de typiske udløsere, og hvert team tilpasser grænserne:

- nedetid eller forringet service, som brugerne kan se, ud over en aftalt grænse;
- datatab af enhver art;
- at vagthavende måtte gribe ind, for eksempel ved at rulle en release tilbage eller omdirigere trafik;
- en løsningstid over en aftalt grænse;
- svigt i overvågningen, hvilket som regel betyder, at et menneske opdagede problemet først.

Kriterierne aftales, før noget går i stykker, så ingen skal diskutere midt i et nedbrud, om netop dette tæller, og enhver interessent kan bede om en postmortem, også når en hændelse falder uden for dem.

Brug begrebet i en beslutning

Brug »Hvad er en postmortem?« 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

En driftshændelse, der bliver rettet, men aldrig skrevet ned, kommer let igen. SRE Workbook gennemgår en postmortem, der blev offentliggjort fire måneder efter hændelsen; i mellemtiden gentog den samme hændelse sig, og teamet måtte stykke sammen igen, hvad det havde lært første gang. En gennemgang, der skrives hurtigt og ligger, hvor folk kan finde den, gør ét teams dårlige eftermiddag til viden, hele organisationen kan bruge.

Dokumentet tvinger også opfølgningen frem. Ben Treynor Sloss, som Workbook citerer, siger det ligeud: for brugerne er en postmortem uden efterfølgende handling ikke til at skelne fra slet ingen postmortem. Derfor kræver Google, at hver postmortem efter et nedbrud, som brugerne mærkede, har mindst én

opfølgningssag på en af de to højeste prioriteter. Postmortemmen er stedet, hvor argumentet for det arbejde bliver fremført med dokumentation, og hvor man senere kan se, om det blev gjort.

Skyldfriheden er det, der gør dokumentationen troværdig. Hvor folk bliver hængt ud for fejl, holder de op med at fortælle om problemer, skriver SRE-bogen, og Workbook tilføjer, at de kan komme til at skjule netop de fakta, der skulle forhindre en gentagelse. En skyldfri kultur får hele tidslinjen med, også den kommando, nogen kørte ved en fejl, fordi ingen bliver straffet for at fortælle om den. Bogen går et skridt videre: et team eller en person, der skriver mange postmortems, skal ikke behandles som et problem, for alternativet er hændelser, som ingen skriver om.

Samlet bliver postmortems til et datasæt. Når hver af dem har de samme metadata, kan en organisation følge udviklingen i tid til opdagelse, tid til løsning og hvor langt en hændelse bredte sig, og se, om pålidelighedsarbejdet betaler sig. De samme dokumenter bruges til oplæring: Google genopfører gamle postmortems som rollespil for nye ingeniører og har læseklubber og et nyhedsbrev med månedens postmortem.

## Sådan ser det ud i praksis

SRE-bogen bringer et komplet eksempel i appendiks D, skrevet for en fiktiv søgetjeneste, der var nede i 66 minutter under en pludselig trafikbølge. Afsnittene er et godt udgangspunkt: status, resumé, konsekvenser, grundårsager, udløser, løsning, opdagelse, opfølgning, erfaringer (hvad gik godt, hvad gik galt, hvor var vi heldige) og en tidslinje i UTC. Tidslinjen markerer fire øjeblikke hver for sig: hvornår nedbruddet begyndte, hvornår hændelsen blev erklæret, hvornår nedbruddet sluttede, og hvornår hændelsen formelt blev lukket. Det er forskellige tidspunkter, og det er i hullerne mellem dem, erfaringerne som regel ligger.

Her er en postmortem i den form for en opdigtet hændelse, skåret ned til det, der kan stå på én skærm:

```
Postmortem: fejl i betalingsflowet efter ændret timeout
Status:       endelig, gennemgået
Ejer:         vagthavende ingeniør, sammen med hændelseslederen
Resumé:       betaling fejlede for omkring hver tredje forespørgsel i 31 min
Konsekvens:   omkring 4.100 mislykkede betalinger, ingen data tabt;
```

```
               fejlbudgettet for 28 dage brugt på én aften
Grundårsager: timeout mod betalingsudbyderen sat ned fra 10 s til 1 s
               i en ændring, der blev gennemgået som »kun konfiguration«;
               intet afprøvede den under spidsbelastning før release
Udløser:      aftenens spidsbelastning gjorde udbyderen langsommere end 1 s
Opdagelse:    alarm på fejlrate kaldte vagten 6 min efter første fejl
Løsning:      ændringen rullet tilbage; fejlene stoppede 4 min senere

Opfølgning
  Type        Prio  Ejer            Færdig når
  forebyg     P1    platformleder   konfig-ændringer kører belastningstest
  opdag       P1    vagtansvarlig   alarm ved timeout mod betaling
  afbød       P2    betalingsleder  tilbagerulning af konfig er ét trin
  proces      P2    hændelsesleder  konfig-ændringer står i deploy-loggen

Erfaringer
  Gik godt:     tilbagerulningen virkede i første forsøg
  Gik galt:     en timeout-ændring blev ikke behandlet som kode
  Var heldige:  spidsen var kort; på en udsalgsdag ville den have
                været tre gange større

Tidslinje (UTC)
  18:02  ændringen merget og udrullet
  18:40  spidsbelastning; første fejlede betalinger   nedbrud begynder
  18:46  alarm på fejlrate kalder vagten
  18:52  hændelse erklæret, hændelsesleder udpeget    hændelse begynder
  19:07  timeout-ændringen fundet, tilbagerulning startet
  19:11  fejlene stopper                              nedbrud slutter
  19:41  tredive minutter med normal fejlrate         hændelse slutter
```

Læg mærke til, hvad der mangler: ingen navne under årsagerne og ingen opfølgning, der beder nogen om at være mere forsigtig. Hvert punkt har én ejer, en prioritet og en sluttilstand, som en anden kan efterprøve. Det meste af råmaterialet skal heller ikke skrives efter hukommelsen. SRE-bogens kapitel om håndtering af hændelser lader hændelseslederen føre et levende dokument under forløbet og beder om, at det gemmes til postmortemmen, og Workbook beskriver værktøjer, der selv udfylder roller, tidslinje og berørte tjenester.

Her på sitet har hændelsesbeskrivelsen om macOS-appens hændelsesbuffer, fra september 2026, meget af en postmortems anatomi. Den skelner mellem diagnoserapporter med 111 GB komprimerede sider på en maskine med 36 GB og en separat måling af hukommelsesforbruget, som viste en vækst på omkring 41 MB i sekundet. Den udpeger en ubegrænset hovedstrøm af hændelser og loghændelser kopieret til abonnenter, der ikke brugte dem, og afgrænser derefter buffere, filtrerer abonnementer efter hændelsestype og giver logningen et hastighedsbudget. En sammenligning af fulde testkørsler

registrerer et fald fra 610.996 core-loglinjer til 1.411. Arkitekturkontroller og budgettests understøtter designet, mens beskrivelsen angiver, hvad dokumentationen ikke fastslår: et målt hukommelsesforbrug efter rettelsen eller et benchmark fra produktion. Separate timeouts ved nedlukning krævede deres egen rettelse, og budgettet tillader med vilje, at gentagne diagnostiske hændelser kasseres. Casen om tre låse på døren, ingen af dem låst beskriver en gennemgang og ikke et nedbrud, men den slutter, hvor en god postmortem bør: de tre rettelser var små, og det, der kom ud af arbejdet, var to regler, der nu gælder for repositoryet, så hele klassen af fejl bliver fanget og ikke kun de tre tilfælde.

## Hvor det går galt

**Dokumentet udpeger en skyldig.** En postmortem, der skriver, at én person alene bar ansvaret, eller som taler om »skødesløs uvidenhed«, lærer alle andre at tie næste gang. Skriv om, hvad der skete, og hvad personen vidste på det tidspunkt, ikke om hvem vedkommende er, og hold sproget sagligt; Workbook peger på dramatiske formuleringer og udråbstegn som noget, der tager fokus fra fakta.

**Opfølgningen beder mennesker om at tage sig sammen.** »Vær mere forsigtig« og »forbedr drejebogen« kan aldrig lukkes, og det er mindre pålideligt at ændre adfærd end at ændre systemet. Giv hvert punkt en type (forebyg, afbød, opdag), én ejer, en prioritet og en sluttilstand, der kan efterprøves, for eksempel en alarm, der udløses, når en bestemt grænse overskrides, og opret en sag for hvert punkt i det system, teamet i forvejen holder øje med.

**Grundårsagen stopper ved udløseren.** »En konfigurationsændring forårsagede nedbruddet« navngiver udløseren. Grundårsagerne er, hvorfor ændringen kunne gøre så stor skade: hvorfor gennemgangen overså den, hvorfor ingen test fangede den, hvorfor alarmen kom sent. Hold de to overskrifter adskilt, som SRE-bogens eksempel gør.

**Konsekvensen er et skuldertræk.** Hvis varigheden er det eneste tal i dokumentet, kan ingen se, om rettelsen virkede. Anslå de berørte brugere, de tabte forespørgsler og det forbrugte fejlbudget; Workbooks råd er, at et velfunderet skøn er bedre end ingen data overhovedet.

**Fire personer ejer den, så ingen gør.** En postmortem skal have én ejer, der er det faste kontaktpunkt for både skrivningen og opfølgningen, med andre som hjælpere. Det samme gælder hvert enkelt opfølgningspunkt.

**Den skrives for sent eller læses af ingen.** Hukommelsen svækkes i løbet af dage, og et dokument, der rundsendes måneder senere, kommer efter næste gentagelse. Workbooks gode eksempel blev sendt ud mindre end en uge efter, at hændelsen var lukket, og til en bred kreds. Og gennemgå den: »En postmortem, der ikke er gennemgået, kunne lige så godt aldrig have eksisteret«, som SRE-bogen skriver. Hold faste gennemgangsmøder, og læg derefter resultatet i et arkiv, man kan søge i.

**Opfølgningen bliver aldrig lukket.** Når den samme hændelse bliver ved med at komme igen, så undersøg, om opfølgningspunkterne tager for lang tid, og om nye features bliver ved med at vinde over rettelserne. Mål afslutning, ikke forfatterskab: Google holder FixIt-uger to gange om året og belønner de ingeniører, der lukker flest opfølgningspunkter fra postmortems.

## Læs mere

- Postmortem Culture: Learning from Failure — kapitel 15 i Googles bog Site Reliability Engineering: udløsere, skyldfrihed, gennemgang og kulturen omkring det.
- Postmortem Culture: Learning from Failure — SRE Workbooks kapitel med en dårlig og en god postmortem af den samme hændelse og begrundelsen for, hvorfor de er, som de er.
- Example Postmortem — en komplet postmortem for et fiktivt nedbrud med opfølgning, erfaringer og tidslinje.
- Managing Incidents — kapitel 14 i SRE-bogen: hændelseslederen, det levende hændelsesdokument og de overdragelser, som postmortemmen senere bygger på.

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 postmortem?« 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/postmortem/