Engineer

Kontinuerlig levering

å holde programvare klar for trygg utgivelse ved behov

Hva det er

Kontinuerlig levering betyr at et team kan utgi en programvareendring trygt når den trengs. Endringen følger en dokumentert vei der hvert trinn forteller om den er klar til å gå videre. Tidspunktet kan avhenge av en ansvarlig godkjenner, et avtalt vedlikeholdsvindu eller andre forhold i driften. Når beslutningen tas, har teamet en ferdig testet pakke og en fremgangsmåte som kan brukes på nytt.

Ved kontinuerlig integrasjon samler utviklerne små endringer ofte og sjekker den felles koden. Testene gir tidlig beskjed om hva som virker sammen. Kontinuerlig levering omfatter også å installere pakken i de aktuelle miljøene og se at resultatet virker der. Ved kontinuerlig utrulling går endringer som oppfyller avtalte krav, automatisk til produksjon. Kontinuerlig levering fungerer også når produksjonstrinnet krever en uttrykkelig beslutning.

Installering og utgivelse kan skje til ulik tid. Koden kan være på plass i produksjon mens den nye funksjonen er slått av. Senere kan teamet åpne funksjonen for noen få brukere før alle får den. Begge trinn trenger en ansvarlig og en kontroll som viser hva som faktisk endret seg for dem som bruker tjenesten.

Bruk begrepet i en beslutning

Bruk «Hva er kontinuerlig levering?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Testet, utgitt og mulig å reversere, slik at endringer når kundene en tirsdag ettermiddag i stedet for sent en fredagskveld.

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. Automatiserte GIS-SaaS-utrulling i GitHub Actions — produksjonssetting, data og rapportering, alt uten manuelle grep

Automatiserte GIS-SaaS-utrulling i GitHub Actions Ta kontakt →

Hvorfor det er viktig

En utgivelse som avhenger av huskelisten til én person, blir en ny anledning til å gjøre feil. Et felles forløp viser hva som egentlig har skjedd. Det skiller en test som feilet, fra en tjeneste som ikke svarte, og fra en godkjenning som ennå mangler. Registrer den installerte pakkens identitet slik at den som har vakt, kan finne den uten å tolke gamle meldinger.

Når det er tungvint å utgi kode, blir flere uavhengige endringer liggende til neste anledning. Hvis én av dem skaper feil, må teamet vurdere hva hele samlingen kan ha påvirket. Med en innarbeidet leveringsvei går det an å sende ut én mindre endring om gangen og se virkningen. Den samme dokumentasjonen hjelper når det er grunn til å vente, for eksempel mens en avhengig tjeneste skifter konfigurasjon.

En utgivelse kan ha avhengigheter utenfor kodelageret. En databasemigrering, nettverksregler og identitetsinnstillinger kan styres av andre team. Undersøk hvilke kombinasjoner som må virke samtidig, og prøv dem i et miljø som ligner produksjon på de viktige områdene. En ferdigbygd pakke sier lite om en kombinasjon som aldri har vært i drift. Beskriv hvordan en kollega på vakt kan finne de avtalte testene og kjøre dem.

Slik ser det ut i praksis

I arbeidet med leveranseflyter i GitHub Actions beskrives egne arbeidsflyter for kodekontroll, tester og utrulling til forskjellige miljøer. Brukergrensesnittet, API-et og databasen fikk grunnpakker tilpasset de ulike tjenestene. Saken beskriver hvordan en definert rute og synlige kontroller gjorde utgivelsene til rutine. Valget av tekniske pakker og godkjennere kan se annerledes ut på andre prosjekter.

Se for deg et API som får en ny mulighet til å eksportere rapporter. Arbeidsflyten bygger pakken og knytter den til kilderevisjonen. Den installerer pakken i et testmiljø, der en testbruker henter rapporten og kontrollerer feltene og tilgangsreglene. Den som tar beslutningen, ser hvilket bygg som ble prøvd og hva resultatet ble. Deretter installeres samme pakke med produksjonsinnstillinger. En enkel kontroll bekrefter at brukere med tillatelse får riktig rapport, og at den gamle funksjonen fortsatt svarer.

En mislykket test må stoppe overføringen og vise en årsak neste person kan følge opp. Skulle produksjonen feile, kan den forrige pakken identifiseres sammen med signalet som endret seg. Øv på forløpet i et kontrollert miljø der én avhengighet er slått av med vilje. Når endringen berører databasen, bør testen også vise om den forrige applikasjonen kan kjøre mot tilstanden den får etter det første trinnet.

Hvor det går galt

Et nytt bygg i hvert miljø gjør det vanskelig å vite om produksjonen får det samme innholdet som ble testet. En ny kompilator eller en endret avhengighet kan gi en annen pakke fra samme commit. Før den identifiserte pakken videre så lenge miljøene kan bruke den. Hvis ulike mål faktisk trenger egne varianter, må hver variant ha sine egne spor og relevante tester.

En godkjenning kan vokse til en parallell leveranseprosess. Gi den som godkjenner, pakkens identitet, resultatet av testene og opplysninger om miljøet som skal endres. Da kan personen ta stilling til gjenværende risiko. Hvis vedkommende må skrive inn alle kommandoene igjen, stopper automatiseringen før en vesentlig del av arbeidet. Bestem hvilke avgjørelser som krever en person, og gjør de øvrige trinnene repeterbare.

En grønn kjøring gir svakt bevis hvis testen veksler mellom å bestå og å feile. Skaff et eksemplar av feilen og finn ut hva som utløser den. Ved en risikabel endring bør teamet få tid til å se hvordan tjenesten oppfører seg før flere brukere blir berørt. Avklar hvem som kan stanse videre utrulling. Neste forsøk bør starte med en kjent pakke og en tilstand som teamet har kontrollert.

Les mer

- DORA om kontinuerlig levering forklarer hva det innebærer å kunne utgi programvare ved behov.
- DORA om kontinuerlig integrasjon beskriver hyppig integrasjon og rask tilbakemelding.
- DORA om automatisert utrulling går gjennom repeterbare trinn i ulike miljøer.
- Styring av utrulling i GitHub Actions beskriver miljøer, jobber og beskyttelsesregler.

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 kontinuerlig levering?» 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/continuous-delivery/