**Engineer**

# Kontinuerlig leverans

att hålla programvara redo att släppas säkert vid behov

## Vad det är

Kontinuerlig leverans innebär att ett team kan släppa en mjukvaruändring säkert när den behövs. Ändringen rör sig genom ett förlopp som går att upprepa. Varje steg visar något om hur klar den är för nästa steg. Produktion kan fortfarande ha ett bestämt underhållsfönster eller en person som fattar det slutliga beslutet. När det väl är dags finns den testade versionen och en känd väg för att installera den.

Med kontinuerlig integration för utvecklarna ofta samman små ändringar och kontrollerar den gemensamma koden. Det visar bland annat om ett program går att bygga och om tidigare beteenden håller. Kontinuerlig leverans innebär också att en version kan installeras och kontrolleras i de miljöer där den behöver fungera. Vid kontinuerlig driftsättning går ändringar som uppfyller kraven automatiskt till produktion. Ett godkännande före produktionssteget är förenligt med kontinuerlig leverans när personen tar ställning till en färdig kandidat.

Installation och lansering kan ske vid olika tillfällen. Den nya koden kan köras i produktion med funktionen avstängd. Senare kan några användare få tillgång innan funktionen öppnas för alla. Teamet behöver kontrollera både att den avstängda funktionen går att driftsätta säkert och att själva öppnandet följer en tydlig ordning.

### Använd begreppet i ett beslut

Använd ”Vad är kontinuerlig leverans?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Testade, släppta och möjliga att backa, så att ändringar når kunderna en tisdagseftermiddag i stället för sent en fredagskväll.

**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. Automatiserade GIS-driftsättning i GitHub Actions — driftsättning, data och rapportering, helt utan manuellt arbete

**Automatiserade GIS-driftsättning i GitHub Actions** Hör av er →

## Varför det spelar roll

En leverans som bygger på att en person minns en rad manuella steg kan ge nya resultat varje gång. Ett dokumenterat förlopp gör utfallet läsbart. Den som tar över kan se skillnad på en misslyckad test, en tjänst som inte är tillgänglig och ett godkännande som inte är klart. Versionsnumret för det som faktiskt installerats går att slå upp, även när den som stod för ändringen har gått hem.

Om en release tar mycket tid är det frestande att vänta tills flera ändringar kan följa med samtidigt. Skapar en av dem problem måste flera möjliga orsaker undersökas. Ett repeterbart förlopp gör det lättare att släppa små ändringar och se vad var och en gör. Den som beslutar kan också vänta med öppna ögon, till exempel om en närliggande tjänst befinner sig mitt i ett planerat byte.

Leveransen kan bero på ändringar från flera kodlager. En databas, en nätverksregel och en anslutning till inloggningen kan skötas av andra personer än dem som bygger programmet. Beskriv vad som måste fungera tillsammans. Prova representativa kombinationer i en miljö som liknar produktionen på de punkterna. Då kan en annan person upprepa en kontroll med samma konfiguration när det behövs.

## Så ser det ut i praktiken

I arbetet med CI/CD och GitHub Actions beskrivs separata arbetsflöden för kodkontroll, tester och installation i olika miljöer. Gränssnittet, API:t och databasen fick var sin imagebas utifrån sina behov. Portföljexemplet beskriver hur tydliga kontroller och en fast väg mellan miljöer gjorde releaser till rutin. En annan tjänst kan ha helt andra behov för paketering eller godkännande.

Tänk dig att en tjänst ska få en ny rapportexport. Arbetsflödet registrerar källrevisionen och bygger ett unikt identifierat paket. Paketet installeras i en testmiljö där en behörig användare hämtar en rapport från realistiska testdata. Samma kontroll tittar på kolumnerna och på vilka personer som ska få åtkomst. Den som godkänner för produktion ser att testet kördes på just den version som sedan ska installeras. Efter installationen kontrollerar teamet en export och ett anrop till den gamla funktionen.

När testmiljön signalerar ett fel stannar paketet där med en förklaring. Om en produktionskontroll sedan misslyckas går det att ta reda på vilken version som körde tidigare och vilket tecken på försämring som först syntes. Öva med en avsiktligt avstängd beroendetjänst i testmiljön. När en databas ingår i förändringen ska övningen också visa om föregående programversion kan använda det tillstånd som finns under övergången.

## Var det går fel

Den som bygger om paketet i varje miljö kan ändra vad som faktiskt provas. En beroendeversion kan ändras eller kompilatorn ha bytts. För därför vidare det identifierade paket som de tidigare kontrollerna läste, när miljöerna kan använda samma innehåll. Om olika plattformar behöver olika byggen ska varje paket ha egna resultat och ett namn som går att koppla till rätt revision.

Ett godkännande kan bli ett andra system för utgivning. Den som beslutar behöver se vilken version det gäller, i vilken miljö den provas och vilka risker som återstår. Personen ska kunna ta ställning utan att uppfinna en egen installationsprocedur. Bestäm vilka frågor som behöver ett mänskligt omdöme. Gör de återkommande tekniska momenten användbara också för en annan behörig operatör.

En grön markering på en test kan ge svaga skäl att gå vidare om den ofta blir grön först vid en omkörning. Undersök sådana fel, och kontrollera att ett övervakningsvärde faktiskt speglar hur tjänsten fungerar. Om en ändring är svår att bedöma kan teamet planera ett observationsfönster och ange vilken signal som ska stoppa utvidgningen. Vid nästa försök ska både paketet och det aktuella tillståndet vara kända.

## Läs mer

- DORA om kontinuerlig leverans beskriver vad som krävs för att kunna släppa ändringar vid behov.
- DORA om kontinuerlig integration förklarar hur små ändringar får snabb återkoppling.
- DORA om automatiserad driftsättning går igenom återkommande steg i olika miljöer.
- Styrning av driftsättning i GitHub Actions beskriver miljöer, jobb och skyddsregler.

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 kontinuerlig leverans?” 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/continuous-delivery/**