

DEVOPS

eficode

WWW.EFICODE.COM

DevOps er en softwareudviklingsmetode til en post-agil verden. Den er baseret på automatisering, virtualisering og korrekte valg af værktøjer. Kedeligt, gentaget arbejde overlades til robotter, hvilket giver projektdeltagerne mulighed for at fokusere på den bedste del af softwareproduktionen: kreativt og kundeorienteret udviklingsarbejde.

For ledelsen

04 - 10

For eksperter

11 - 18



DEVOPS



For ledelsen



**”Hvor mange ting er ikke blevet benægtet
den ene dag, hvorefter det er blevet til realiteter
den næste dag?”**

Jules Verne (1828 - 1905)

DEVOPS - EN TREDJEGENERATIONS SOFTWAREUDVIKLINGSMETODE

DevOps er en del af en global ændring i hverdagen for private og offentlige organisationer, hvor IT og IT infrastruktur i højere grad bliver virtualiseret. Konsulentfirmaet Deloitte har nævnt DevOps som en af de vigtigste teknologitrends i 2014. ¹

EFICODE UNDERSØGER

66 %

af Finlands IT beslutningstagere mener, at DevOps er interessant for deres virksomhed. ²

En virksomhed søger måske en webservice, der kan få kasseapparatet til at synge eller et digitalt værktøj, der gør livet lettere for virksomhedens ansatte. Når idéen er blevet formuleret og målene er sat, kan softwareudviklingen begynde. Efter et stykke tid kan udviklingsteamet præsentere det færdige produkt og brugerne vil tage imod det med kyshånd. Eller vil de? Alt for ofte sker det at budgetterne bliver overskredet, deadlines ikke overholdes og - værst af alt - at brugerne er utilfredse

Hvad er problemet? Som regel skyldes det den måde, som software udvikles på.

Den først udbredte softwareudviklingsmetode går under navnet 'vandfaldsmodellen'. Den er centreret omkring rigide, på forhånd fastlagte krav, hvis opfyldelse bliver grundigt overvåget i henhold til en møjsommeligt detaljeret aftale. Da ingen kan forudse hvad der sker i løbet af et udviklingsprojekt, resulterer denne proces typisk i katastrofale brugeroplevelser. Som et minimum bliver budgettet sprængt og den originale tidsplan skrottet. Dette sker ofte i forbindelse med ændringer i softwaren.

1 - Deloitte Development, LLC (2014) Tech Trends 2014: Inspiring Disruption. Deloitte University Press.
2 - Markedsundersøgelseskontor Value Clinic Oy (2014) DevOps i Finland-undersøgelserapport.

FAKTA

18 %

Færre omkostninger til udvikling og vedligeholdelse
= Næsten en femtedel besparelse på udgifterne. ³

50 %

Færre fejl
= Dobbelt så stor chance for en vellykket release af produktet. ⁴

30 ×

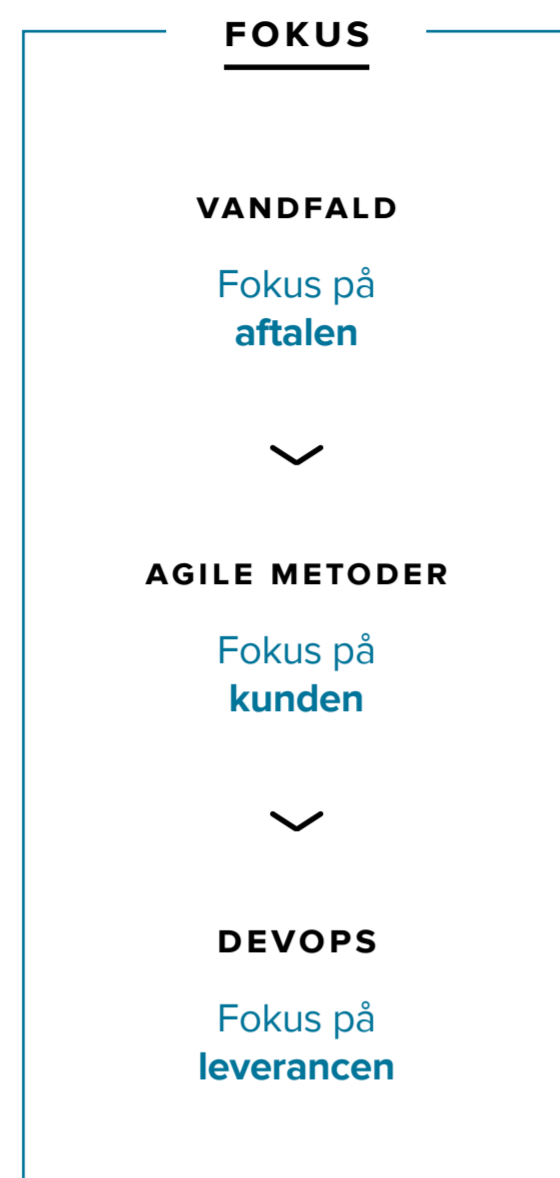
Hurtigere release
= Implementering 30 gange så ofte og 8000 gange hurtigere. ⁴

20 %

Mere forretning
= Vækst i omsætning og nye kunder på 17-23%. ³

³ - CA Technologies (2013) TechInsights Report: What Smart Businesses Know About DevOps. CA Technologies.

⁴ - Puppet Labs (2013) 2013 State of DevOps Report. IT Revolution Press.



For at løse problemet med krav, der ændres undervejs i udviklingsforløbet, blev "agil" softwareudvikling introduceret. Agile metoder håndterer udvikling som en iterativ proces og lægger vægt på kommunikation. De agile metoder har elimineret mange af de udfordringer, der plagede vandfaldsmodellen, men de adresserer ikke samarbejdet med IT driftsorganisationen. Købere af software er også bekymrede over den "pengetræs-mentalitet", der forbindes med agil udvikling og derfor truer med at sprænge projekternes budgetter.

Softwareproduktionen virtualiseres og automatiseres

De gode ting ved agil udvikling er også en del af DevOps, en tredjegenérations udviklingsmetode, der bygger på virtualisering og automatisering. Med DevOps bliver softwareudviklingen cloudbaseret og mængden af manuelt gentaget arbejde - for eksempel i forbindelse med kvalitetssikring og releases - reduceret.

Resultaterne af udviklingsarbejdet kan præsenteres for

købere og slutbrugere af softwaren, i takt med at de enkelte dele færdiggøres. DevOps gør det også muligt at kombinere kravspecifikationer med automatisk kvalitetssikring, hvilket betyder at alle involverede medarbejdere lige fra udviklere til ledere kan få realtidsindsigt i, hvordan udviklingen skrider frem. Produktudviklingsprocessen fokuserer altid på at levere fordele til kunden.

En organisation, der gør brug af DevOps principperne, tester og releaser software kontinuerligt og automatisk. Udviklingsarbejdet udføres som regel i et virtualiseret miljø. Dette reducerer antallet af fejl i og omkostninger ved infrastrukturen og udviklingsarbejdet. Når udviklere og driftsfolk, der ellers ofte foretrækker at arbejde uafhængigt af hinanden, deler samme mål og måles efter samme skala, hjælper det dem med at blive bedre til at arbejde sammen, hvilket i sidste ende giver en bedre brugeroplevelse.

DevOps løser sin forgængers udfordringer

Agil udvikling løste en stor del af vandfaldsmodellens problemer. DevOps fortsætter, hvor agilmetodikken slipper og eliminerer samtidig en del af de ulemper som agil udvikling har.

Fri for produktionskædens flaskehalse

Agile metoder gjorde softwareproduktion mere fleksibel. Til gengæld tog den ikke hensyn til IT miljøer og de værktøjer, der bruges i softwareproduktionen. Dette skaber flaskehalse

PROBLEMER / LØSNINGER

VANDFALDSMODEL PROBLEM	vs	LØSNING MED AGIL UDVIKLING
Software specificeres, ligesom byggeprojekter, i detalje før arbejdet går i gang. Det er dyrt eller umuligt at ændre planer undervejs.	—	Software udvikles iterativt og i samarbejde med kunden under hele projektet.
Projekter er ofte aftalt til fast pris med en bunden kravspecifikation. Forhandlinger er langsomme og baseret på antagelser.	—	Aftalen, der indgås i starten af projektføreløbet, omhandler den vigtigste funktionalitet og en prioriteret liste af krav, og ikke en beskrivelse af det endelige produkt.
Software udvikles i faser og test påbegyndes først når udviklingen er afsluttet.	—	Softwareudviklingen gennemføres i mindre, uafhængige iterationer, der kan tilpasses inden hver ny iteration påbegyndes.
Det er svært at skabe sig et overblik over udviklingsprocessen.	—	Kunden får mulighed for kontinuerligt at se den reelle projektstatus undervejs i forløbet.
Man er ikke forberedt på at håndtere ændringer, der opstår undervejs i projektføreløbet.	—	Ændringer er velkomne, og de prioriteres som en del af de fremtidige iterationer.
Systemet testes først i slutningen af implementeringsfasen, hvor det er dyrest at gennemføre ændringer. Nødvendige rettelser kan i nogle tilfælde være umulige at gennemføre.	—	Systemet testes allerede under implementeringsfasen for at sikre at de krav, der er stillet til den enkelte iteration, er opfyldt før næste iterationen påbegyndes.

PROBLEM VED AGIL UDVIKLING	vs	DEVOPS' LØSNING
Levering af ny funktionalitet til kunden er ofte forsinket.	—	DevOps værktøjer bruges til både test og release af ny funktionalitet, når den er færdiggjort.
Udviklede softwarekomponenter er ikke indbyrdes kompatible.	—	Åbne grænseflader og testautomatisering gør det muligt at opdele udviklingen i mindre - men kompatible - dele.
Kvalitetssikring af produktet er ikke gennemført inden release.	—	DevOps værktøjer og metoder hjælper til med at automatisere kvalitetssikringen og reducerer behovet for gentaget manuelt arbejde.
Ny funktionalitet ødelægger eksisterende funktionalitet.	—	Kvaliteten af eksisterende funktioner sikres hurtigt og automatisk efter hver ændring.
Budgetter sprænges og deadlines overskrides.	—	DevOps værktøjer og procedurer øger gennemsækeligheden og forudsigeligheden af udviklingsarbejdet.
Udviklings- og driftspersonalet samarbejder ikke.	—	Udviklingsteams og driftsafdelingen er enige om fordelingen af ansvar. Deres mål er fælles.

i udviklingskæden, hvilket forsinker hele udviklingsprocessen. I DevOps spiller de tekniske forhold en vigtig rolle. Eksisterende datacenterløsninger kan erstattes af cloudbaserede eller virtualiserede løsninger, og manuelle processer automatiseres.

Det er ingen mirakelkur

DevOps har meget at tilbyde til de organisationer, der er villige til at lære, men det er ingen mirakelkur. DevOps overflødiggør ikke manuelt arbejde fuldstændigt, men gentagent og kedeligt arbejde, samt arbejde der kræver høj arbejdshastighed, flyttes ved hjælp af DevOps til maskiner. DevOps giver udviklerne mulighed for at fokusere på det, der tilføjer værdi: produktets kvalitet og brugeroplevelsen.

Den lette vej til effektivitet

DevOps kræver ikke gigantiske implementeringsprojekter eller totale kulturskift. Kortlægning af status på softwareproduktionen og identifikation af hvor det gør ondt i udviklingsteamet, er nok til at man kan komme i gang.

Her er lederens to skridt hen mod en mere effektiv fremtid:

01

Tal med den IT ansvarlige i din organisation om status på softwareproduktionen og om hvordan DevOps principperne måske kan dække jeres behov. Vis også den IT ansvarlige den tekniske del af denne guide.

02

Hvis du ikke kan finde en person, som viser interesse for DevOps internt i organisationen, så prøv at kigge efter en udenfor. Vær opmærksom på at serviceudbyderen ikke kun skal levere en teknisk ydelse, men også skal gøre DevOps til videnskapital for din organisation.

SYSTEMUDVIKLING

SYSTEMUDVIKLING I ET TRADITIONELT MILJØ

VS

SYSTEMUDVIKLING MED DEVOPS

IT-miljøer er ufleksible og dyre at vedligeholde.

Et virtualiseret miljø er en økonomisk overkommelig måde til at opretholde basal funktionalitet. Det kan også hjælpe med at identificere belastningspukler.

Leverandørafhængighed reducerer paletten af færdigheder og tvinger en til at indgå unødvendige kompromisser.

Principperne i DevOps understøtter modularitet og åbne grænseflader. Man er ikke bundet til en bestemt leverandør.

Det er svært at overvåge opfyldelsen af kundens krav. Det kan være umuligt at estimere en release-plan.

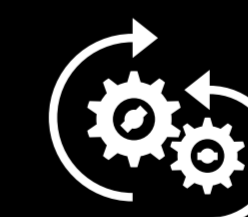
Med DevOps kan kundens krav centraliseres og vedhæftes i forbindelse med automatisk kvalitetssikring. Kunden kan adviseres automatisk, når der har været release af ny funktionalitet.

Menneskelige fejl ved kvalitetssikring medfører ujævn kvalitet.

Testautomatisering håndterer altid kvalitetskontrol med højeste effektivitet og ensartethed.

Udviklerne bliver frustrerede over gentagne arbejdsopgaver.

Kedelige opgaver overtages af maskiner og menneskers tid frigøres til kreativt arbejde.



DEVOPS ER IDEELT FOR DE FORANDRINGSVILLIGE

DevOps er ideelt for organisationer, der:

1. Udvikler, vedligeholder eller køber software
2. Ønsker at udvikle bedre digitale services
3. Leder efter besparelser
4. Ønsker at udvikle deres virksomhed

MERE OM DEVOPS

Du fik altså ikke slukket din tørst for information? Godt! Dyk længere ned i DevOps-verdenen med disse informationskilder:

THE PHOENIX PROJECT: A NOVEL ABOUT IT, DEVOPS, AND HELPING YOUR BUSINESS WIN (GENE KIM ET AL.) · En hit-novelle skrevet af tre tunge IT-folk, som fortæller en fiktiv historie om en organisation, der kæmper med et forsinket projekt med et budget, der er overskredet. Knuden løses, når DevOps kommer til byen.

BUILDING A DEVOPS CULTURE (MANDI WALLS) · En kort og præcis guide til, hvordan en moderne udviklingskultur kan opbygges og hvordan ændringer i organisationen kan gennemføres.

DEVOPS.COM · DevOps.com indeholder en omfattende samling af DevOps-relaterede artikler - lige fra nyheder og længere historier til blogs.

EFICODE.FI/EN/CATEGORY/BLOG · På en del af Eficodes blog åbner denne guides forfattere op for deres tanker og idéer om DevOps. Læs for eksempel om at bygge bro over gab i kommunikation eller om DevOps robotten, der booster Finlands konkurrenceevne.

EFICODE AUDIT



Eficodes gennemprøvede og troværdige audit service er en god måde at komme i gang på. En audit indebærer at vores eksperter upartisk bedømmer tilstanden af dine interne og dine outsourcede softwareprojekter og giver dig en detaljeret udviklingsplan, der kan hjælpe til med at øge effektiviteten af dine processer. Planen beskriver blandt andet hvor der er mest pres på i din softwareproduktion, infrastruktur, miljøer, processer, værktøjer og udviklingskultur. Når planen er udarbejdet, er det op til dig at beslutte hvor stor en del af anbefalingerne du vil følge.

WWW.EFICODE.COM



DEVOPS MODENHEDSMODEL

WWW.EFICODE.COM

	001	002	003	004
LEDELSE	Udvikling holdes adskilt fra forretningen. Det er komplekst at starte nye udviklingsprojekter.	Udviklingsprojekter kan startes agilt og metoder til projektstyring er fastlagt.	Nye projekter kan kobles direkte op på organisationens strategiske mål. Det er nemt at starte nye pilotprojekter.	Der er mulighed for realtids-indblik i udviklingsforløbet, hvilket understøtter beslutningstagningen og holder styr på opfyldelsen af de strategiske mål.
ORGANISATION OG KULTUR	Design, udvikling og kvalitetssikring holdes adskilt. Kommunikation foregår hovedsageligt skriftligt.	Arbejdet udføres i teams, men udvikling og kvalitetssikring holdes adskilt.	De enkelte teams arbejder selvstændigt og har det fulde ansvar for udvikling og kvalitetssikring af funktioner.	De enkelte teams kommunikerer regelmæssigt med hinanden og samarbejder om at forbedre den måde de arbejder på. Der er konstant kommunikation med driftsafdelingen.
MILJØER OG RELEASES	Produkterne er miljø-specifikke og kompiles manuelt. Miljøer installeres og konfigureres manuelt.	Systemet opdeles i mindre enheder og kompilermiljøet er kendt. Enkelte releases foregår automatisk.	Miljøer kan installeres og konfigureres automatisk. Build- og releaseprocesser automatiseres.	Releases kan foregå automatisk og kontinuerligt. Migrations- og retableringsprocesser fungerer som de skal.
BUILDS OG KONTINUERLIG INTEGRATION	Produkt-integration foregår automatisk, men konfiguration og ibrugtagning styres manuelt. Ingen logføring af ændringer.	Integrationsprocessen startes af de enkelte teams ved enhver ændring. Værktøjer deles. Integration uden test.	Integrationen dækker hele produktet og er knyttet til accepttesten. Afhængigheder er kendte og håndteres.	Udviklingsorganisationen har regelmæssige møder. Indsamlede resultater er med til at øge hastigheden i feedback-fasen og giver bedre indblik.
KVALITETSSIKRING	Kvalitetssikring foretages manuelt og foregår som regel efter endt udvikling.	Unit test og/eller statisk kodeanalyse anvendes for dele af produktet.	Funktionalitet rettet mod slutbrugere testes automatisk. Testpersonale deltager i udviklingsprocessen.	Accepttests reflekterer systemkravene tydeligt og bruges i videst muligt omfang som guide i udviklingen af systemet.
SYNLIGHED OG RAPPORTERING	Rapportering foregår manuelt og efter behov.	Kodeintegration og resultater af unit tests og kodeanalyse kan ses af hele teamet.	Status på opfyldelse af krav til systemet kan overvåges i realtid baseret på tests og frigivet funktionalitet.	Resultater og erfaringer indsamles under udviklingsforløbet og bruges som grundlag for forbedringer.
TEKNOLOGIER OG ARKITEKTUR	Teknologier og værktøjer er forældede eller lever ikke op til nutidige krav.	Teknologier er ved at blive forældede og arkitekturen er kun delvis fleksibel. Grænseflader er mangelfulde eller helt fraværende.	Teknologier er moderne eller bredt understøttede. Grænseflader er grundigt dokumenterede og er til rådighed for al nøglefunktionalitet.	Arkitektur og teknologi er dækkende og gør det muligt at opfylde målene for forretningen effektivt.

DEVOPS

eficode

For eksperter

“Måske har vi selv bestemt vores skæbner, og fortjener dermed alt det vi bliver udsat for, fordi vi ikke havde modet til at forestille os noget bedre.”

Iain Banks (1954 - 2013)

HVAD ER DEVOPS?

DevOps binder hele softwareproduktionskæden sammen, lige fra kundens krav til et leveret produkt eller service.

EFICODE UNDERSØGER**72 %**

Af de Finske beslutningstagere mener, at time-to-market er DevOps' vigtigste egenskab.¹

Hvor agil udvikling fokuserer på kommunikation og softwareudvikling i små enheder, tager DevOps udgangspunkt i kunden, automatisering og gennemskelighed. Udtrykket DevOps beskriver en kultur hvor man benytter sig af moderne værktøjer og processer.

DevOps inddrager alle parter, der er involveret i udviklingsprocessen: kunden, udviklere, partnere, kvalitetssikringen og driften. Mange DevOps principper, som for eksempel kontinuerlige leverancer, adskiller sig fra den traditionelle opdeling af ansvar i organisationen. DevOps fokuserer på gnidningsløst samarbejde, der typisk understøttes af centraliseret kravstyring og automatisk kvalitetssikring.

Principperne i DevOps er ikke helt nye. Mange elementer i DevOps - for eksempel virtualisering og automatisk kvalitetssikring - har allerede været i brug lang tid før udtrykket DevOps blev introduceret. Formålet med DevOps er at inddrage de bedste og mest effektive teknikker og metoder, og bruge dem til at skabe en større sammenhæng.

¹ - Markedsundersøgelseskontor Value Clinic Oy (2014) DevOps i Finland-undersøgelsesrapport.



FORDELE

HVAD ER FORDELENE VED DEVOPS?

DevOps binder udvikling og drift sammen og bringer overblikket helt ud til kunden.

- 01. Systemkrav samles på et klart defineret sted.
- 02. Udviklingsmiljøer er kendte og kan oprettes automatisk.
- 03. Effektiv versionskontrol giver bedre kontrol over udvikling og drift.
- 04. Ny funktionalitet testes automatisk på alle trin i udviklingsprocessen.
- 05. Automatisk kvalitetssikring, der er koblet direkte op på de oprindelige krav.
- 06. Nye produktversioner kan enten releases automatisk eller ved et enkelt klik.
- 07. Servermiljøer er virtuelle eller cloud-baserede. De kan konfigureres og sættes op automatisk.
- 08. Driften og effektiviteten af services kan overvåges af hele organisationen.

PROBLEM / LØSNING

PROBLEM VED TRADITIONEL SYSTEMUDVIKLING

vs

DEVOPS' LØSNING

Kundens krav, fejlmeldinger og nye idéer til udvikling er spredt ud over det hele - og hvor er dokumentationen?

For at øge muligheden for samarbejde, skal krav og dokumentation findes på ét sted, hvor alle kan se dem.

Systemintegration er risikofyldt og tidskrævende.

Kontinuerlig integration understøtter sikringen af funktionalitet og kompatibilitet, i takt med at der foretages ændringer i produktet.

Systemtests udføres kun i forbindelse med release af nye versioner.

Automatisk accepttest og kontinuerlig integration gør det muligt at gennemføre kvalitetssikring, når der sker ændringer.

Udviklere og driftsfolk samarbejder ikke med hinanden.

Release af produkter foregår automatisk. Ansvar for funktionalitet i en ny release deles mellem udviklere og driftsfolk.

Servermiljøer konfigureres til 'worst case scenarios', hvilket resulterer i øgede udgifter og mindre fleksibilitet.

Virtualisering af servermiljøer gør det muligt at nedskalere servicen og dermed reducere omkostningerne i perioder, hvor der ikke er større belastninger på systemet.

Udviklingsforløb tager alt for lang tid.

Automatisering og fælles mål for udviklingsteams sikrer at hele produktionskæden arbejder med lynets hast.

Nogle teams i produktionskæden har forskellige mål, der kan være indbyrdes modstridende.

Hele produktionskæden har et fælles mål, der kan inddeles i flere delmål.

De forskellige dele af produktionskæden modtager feedback for sent og feedback sendes ikke altid til den korrekte modtager.

Automatisk genereret feedback sendes til den korrekte person.

KERNEKOMPONENTER I DEVOPS

DevOps er en kombination af værktøjer og en udviklingskultur, der deles af hele organisationen. Automatisering er kernen i DevOps. Det understøtter en produktudvikling, der flyder som en konstant strøm gennem hele organisationen. Kravene, der defineres af kunden, bliver automatisk overført fra ét trin i processen til det næste – hele vejen til release og vedligeholdelse af færdigudviklet funktionalitet.

KRAVSTYRING

Problem: Krav og fejlmeldinger ligger spredt forskellige steder.

1. Krav og systemdokumentation samles på ét sted, som hele udviklingsorganisationen har adgang til.
2. Status på de enkelte krav overvåges gennem hele udviklingsforløbet indtil frigivelsen af softwaren. På denne måde kan man altid vide hvornår ny funktionalitet er tilføjet og hvilke ændringer der er inkluderet i en release.

UDVIKLINGSMILJØER

Problem: Etablering af udviklingsmiljøer er meget tidskrævende. Udviklerne arbejder i forskellige udviklingsmiljøer.

1. Udviklingsmiljøerne bør kunne etableres automatisk ved hjælp af konfigurationsbiblioteker og -værktøjer.
2. Udviklingsmiljøerne bør være tæt forbundne med central krav- og versionsstyring samt – via kontinuerlig integration – til automatisk kvalitetssikring og implementering.
3. Udviklingsarbejdet guides ved hjælp af hurtig og automatisk feedback via kontinuerlig integration, automatisk kvalitetssikring og kodeanalyse.

KONTINUERLIGE RELEASES OG IMPLEMENTERING

Problem: Det tager lang tid at lave produktions-releases og der er stor risiko for fejl.

1. Kontinuerlige leverancer sigter mod at skabe en situation hvor et produkt automatisk kan releases til produktion efter hver ændring.
2. Konfigurationer, anvendt i udviklingsfasen, bør være knyttet til konfigurationen og virtualiseringen af services, så releaseprocessen kan automatiseres.
3. Processen med kontinuerlige leverancer kan indeholde kvalitetskontrol, der stopper den automatiserede proces, indtil resultaterne er gennemgået og processen får lov til at fortsætte.
4. Kontinuerlige leverancer er en fremgangsmåde, der sigter mod at automatisere gentagne processer så pålideligt, at man har tillid til automatiserede releases helt ud til slutbrugerne.

CASE ROVIO: Angry Birds med automatiseret softwareproduktionslinje.
eficode.fi/en/cases/angry-birds-hit-automated-software-production-line

ACCEPTTEST

Problem: Krav er ikke bundet direkte op på tests. Test foretages hovedsageligt manuelt og først efter endt udviklingsarbejde.

1. Krav knyttes til kvalitetssikringen, så den aktuelle status på dem hele tiden er synlig.
2. Som minimum automatiseres den del af kvalitetssikringen, som skal udføres efter hver ændring.
3. Automatiserede test cases skrives i naturligt sprog, så både kunder og eksperter forstår hvorfor og hvordan systemets kvalitet sikres.

```
Transferring money using a phone number
Login to payment service      ${user}
Input recipients phone number  ${phone}
Input the payment amount      €50
Accept payment
Confirm the payment has been transferred
```

VIRTUALISERING

Problem: Vedligeholdelse af servere er dyr og kapaciteten overstiger normale behov. Konfigurationerne på test- og præproduktionsystemer afviger fra konfigurationen på produktionssystemerne.

1. En af DevOps' hjørnesten er virtualisering.
2. Virtualisering giver mulighed for identiske miljøer under hele udviklingsfasen.
3. Virtualisering giver mulighed for en signifikant reduktion i udgifter til drift da virtuelle systemer kan skaleres efter behov.
4. Virtualisering gør udvikling og kvalitetssikring hurtigere. Antallet af fejl, der kan henføres til miljøet reduceres markant.

MONITORERING

Problem: Systemet kører langsomt, men ingen ved præcist hvorfor.

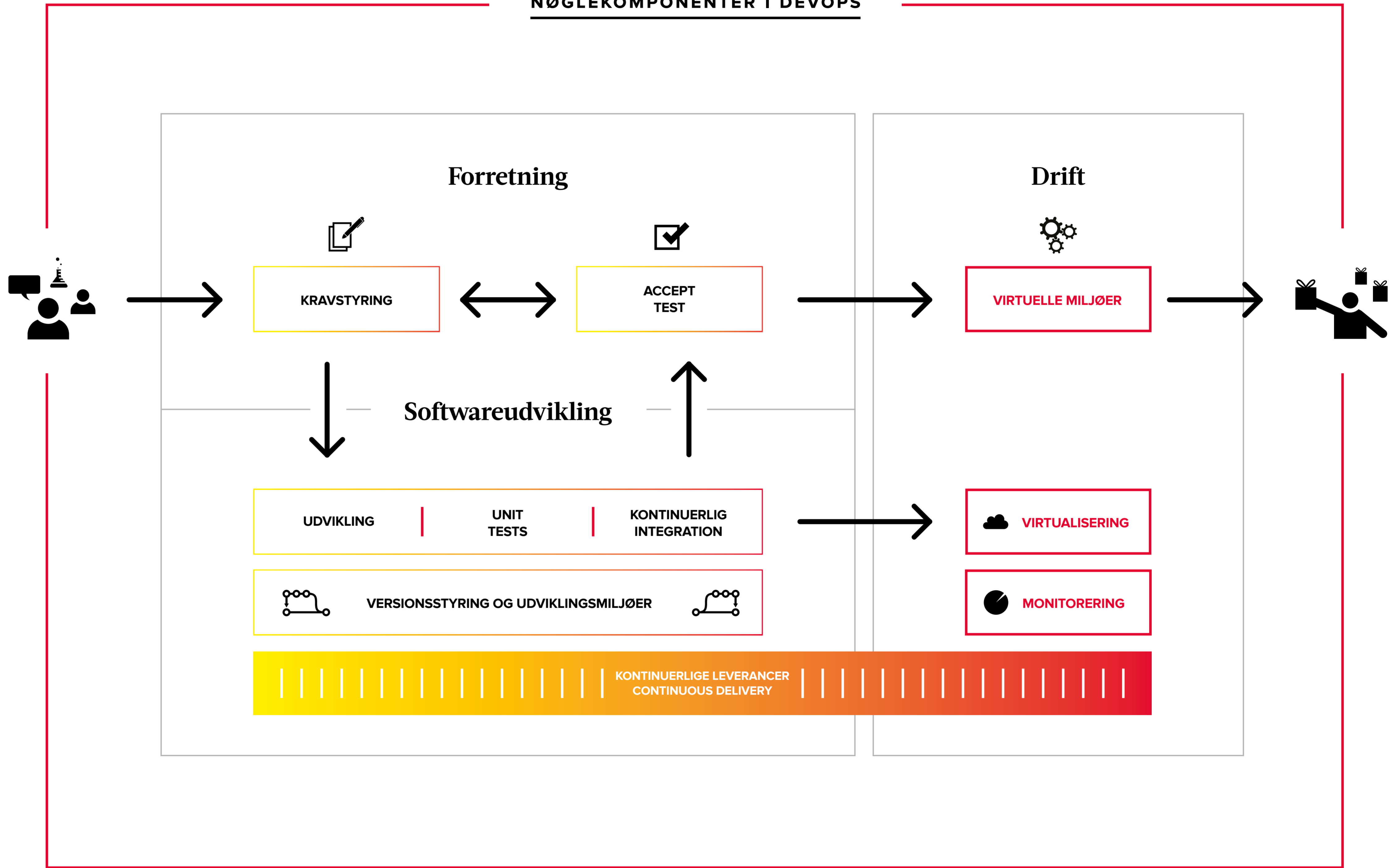
1. Systemmonitorering giver mulighed for at måle ydelsen på de enkelte services og på selve serveren.
2. Moderne systemmonitoreringsværktøjer er i stand til at rapportere de største problemer og bidrager til forbedring af systemet ydelse.
3. Monitorering hjælper til med at forbedre udviklingen på lang sigt og til at tracke fejl i systemet.

GRÆNSEFLADER

Problem: Systemet kan ikke testes automatisk.

1. DevOps flytter fokus i produktudviklingen fra at bygge en monolith til at bygge mindre uafhængige enheder, der kommunikerer ved brug af veldokumenterede grænseflader.
2. Grænseflader dokumenteres grundigt, standarder overholdes og de bedste metoder benyttes.

NØGLEKOMPONENTER I DEVOPS



HVORDAN KOMMER MAN I GANG?

Du kan begynde at lære DevOps bedre at kende allerede nu. Start med at overveje hvor den største flaskehals er i din produktionskæde og forsøg at løse problemet med en af de metoder, der er nævnt tidligere. Prøv at automatisere dine manuelle tests eller undersøg om dine servere kan virtualiseres.

VIL DU INTRODUCERE DIT TEAM TIL DEVOPS? PRØV DISSE TRIN:

01. Kortlæg tilstanden af jeres softwareudviklingsprocesser.
02. Lav et roadmap, der muliggør softwareudvikling i flere mindre trin.
03. Eksperimentér på områder hvor kulturen er forandringsvillig, eller hvor dine skarpeste og mest innovative medarbejdere befinder sig.
04. Foretag en kortlægning af teknisk know-how og afprøv forskellige metoder.
05. Etablér fælles principper og gør dem til faste arbejdsgange for organisationen.
06. Hop ud i det ukendte! Giv eksempelvis udviklerne lov til at release ændringer direkte i produktionsmiljøet.

PÅ EFICODES BLOG:

Automatic testing with Robot Framework (video tutorial)

eficode.fi/en/blog/maintainable-automatic-tests-for-your-web-application-2

EFICODE DEVOPS

For ledelsen

04 - 10

For eksperter

11 - 17



DEVOPS MODENHEDSMODEL

WWW.EFICODE.COM

	001	002	003	004
LEDELSE	Udvikling holdes adskilt fra forretningen. Det er komplekst at starte nye udviklingsprojekter.	Udviklingsprojekter kan startes agilt og metoder til projektstyring er fastlagt.	Nye projekter kan kobles direkte op på organisationens strategiske mål. Det er nemt at starte nye pilotprojekter.	Der er mulighed for realtids-indblik i udviklingsforløbet, hvilket understøtter beslutningstagningen og holder styr på opfyldelsen af de strategiske mål.
ORGANISATION OG KULTUR	Design, udvikling og kvalitetssikring holdes adskilt. Kommunikation foregår hovedsageligt skriftligt.	Arbejdet udføres i teams, men udvikling og kvalitetssikring holdes adskilt.	De enkelte teams arbejder selvstændigt og har det fulde ansvar for udvikling og kvalitetssikring af funktioner.	De enkelte teams kommunikerer regelmæssigt med hinanden og samarbejder om at forbedre den måde de arbejder på. Der er konstant kommunikation med driftsafdelingen.
MILJØER OG RELEASES	Produkterne er miljø-specifikke og kompiles manuelt. Miljøer installeres og konfigureres manuelt.	Systemet opdeles i mindre enheder og kompilermiljøet er kendt. Enkelte releases foregår automatisk.	Miljøer kan installeres og konfigureres automatisk. Build- og releaseprocesser automatiseres.	Releases kan foregå automatisk og kontinuerligt. Migrations- og retableringsprocesser fungerer som de skal.
BUILDS OG KONTINUERLIG INTEGRATION	Produkt-integration foregår automatisk, men konfiguration og ibrugtagning styres manuelt. Ingen logføring af ændringer.	Integrationsprocessen startes af de enkelte teams ved enhver ændring. Værktøjer deles. Integration uden test.	Integrationen dækker hele produktet og er knyttet til accepttesten. Afhængigheder er kendte og håndteres.	Udviklingsorganisationen har regelmæssige møder. Indsamlede resultater er med til at øge hastigheden i feedback-fasen og giver bedre indblik.
KVALITETSSIKRING	Kvalitetssikring foretages manuelt og foregår som regel efter endt udvikling.	Unit test og/eller statisk kodeanalyse anvendes for dele af produktet.	Funktionalitet rettet mod slutbrugere testes automatisk. Testpersonale deltager i udviklingsprocessen.	Accepttests reflekterer systemkravene tydeligt og bruges i videst muligt omfang som guide i udviklingen af systemet.
SYNLIGHED OG RAPPORTERING	Rapportering foregår manuelt og efter behov.	Kodeintegration og resultater af unit tests og kodeanalyse kan ses af hele teamet.	Status på opfyldelse af krav til systemet kan overvåges i realtid baseret på tests og frigivet funktionalitet.	Resultater og erfaringer indsamles under udviklingsforløbet og bruges som grundlag for forbedringer.
TEKNOLOGIER OG ARKITEKTUR	Teknologier og værktøjer er forældede eller lever ikke op til nutidige krav.	Teknologier er ved at blive forældede og arkitekturen er kun delvis fleksibel. Grænseflader er mangelfulde eller helt fraværende.	Teknologier er moderne eller bredt understøttede. Grænseflader er grundigt dokumenterede og er til rådighed for al nøglefunktionalitet.	Arkitektur og teknologi er dækkende og gør det muligt at opfylde målene for forretningen effektivt.

For ledelsen



MICHAEL WILLUMSEN

Business Manager
michael.willumsen@eficode.com
+45 31248985



HEIKKI HÄMÄLÄINEN

Sales Director
heikki.hamalainen@eficode.com
+358 40 766 2610

For eksperter



MARKO KLEMETTI

CTO
marko.klemetti@eficode.com
+358 445 225 927



JESPER IO KARUP

Lead Architect
jesper.karup@eficode.com
+45 25382802



Eficode MBC ApS

Købmagergade 26C, 3.tv, 1150 København K

Eficode er et software- og konsulenthus, med kontorer i Helsinki, Tampere, Peking, Stockholm (2017) og København, som arbejder efter ét overordnet princip: intet nonsens, kun resultater.