

**Flexibiliteit,
efficiëntie en
snelheid in
software-
ontwikkeling**

Continuous delivery

Opkomende technologieën stellen organisaties voor nieuwe uitdagingen, maar ze bieden ook mogelijkheden. De eindgebruiker is steeds op zoek naar snellere, betere en stabielere oplevering van software. Nieuwe ontwikkelmethoden als 'continuous delivery' bieden hier uitkomst.

Dennis Joosten en Joop Snijder



De kwaliteit van de producten of diensten die een organisatie levert of aanbiedt, is belangrijk voor iedere organisatie. Maar hoe goed deze producten of diensten ook zijn, het succes staat of valt met goede dienstverlening. Hier is voor veel organisaties nog veel te winnen. Klanttevredenheid wordt vooral bepaald door de mate van service die klanten ervaren. Deze service wordt grotendeels gedreven door proceskwaliteit en daarmee is IT een van de meest bepalende factoren in de strategie van een organisatie. Het is daarom belangrijk voor organisaties om kritisch te kijken naar hun ontwikkelmethodes en na te gaan of de huidige situatie tot IT-oplossingen leidt met optimale functionaliteit voor de eindgebruiker. Tegenwoordig dient de 'time to market' kort te zijn en anno 2014 zijn er vele nieuwe ontwikkelmethoden die dit bewerkstelligen. Eén daarvan is 'continuous delivery.'

Iteratieve ontwikkelaanpak

Met het complexeren van IT volstaat de traditionele 'waterval'-methode in de meeste gevallen niet meer. Bij deze methode worden duidelijk afgebakende fasen stap voor stap doorlopen. Maar de omgevingsfactoren binnen veel organisaties zijn inmiddels zo onzeker, dat meer flexibiliteit vereist is. Daarnaast is er een opmars

van apps en cloudapplicaties die continu worden geüpdatet. Lang wachten op een nieuwe versie is niet meer van deze tijd.

Organisaties zoeken steeds vaker naar flexiblere manieren om software te ontwikkelen. Ze kiezen voor een agile- en iteratieve aanpak. Een agile-aanpak zorgt ervoor dat teams de juiste zaken ontwikkelen voor de business en er snel kan worden bijgestuurd door kortcyclisch te werken. Dit heeft als grote voordeel dat het de flexibiliteit van het ontwikkelproces bevordert doordat functionaliteit in stappen wordt toegevoegd.

Het ontwikkelteam functioneert beter, omdat ze de eerste ontwikkelfasen als analyse, ontwerp en bouw niet afzonderlijk afhandelen. Het nadeel van deze aanpak is dat de opvolgende testfase lang kan duren en dat er tijd overheen gaat voordat de software-implementatie is doorgevoerd in de productieomgeving. Beheerders hebben immers een drukke agenda en hun streven naar stabiliteit zorgt ervoor dat zij het aantal releases laag houden.

De denkfout die men hierbij maakt, is dat deze aanpak tijd bespaart. In beginsel klopt de redenering van de agile-aanpak. Maar omdat de implementatie vaak handmatig gebeurt, neemt het risico op fouten toe, waardoor beheerders

Samenvatting

Continuous delivery is een iteratief proces waarbij continu kleine stukken functionaliteit automatisch worden uitgerold. Door zeer korte cycli is de impact van wijzigingen klein, is bijsturen makkelijker en worden minder fouten gemaakt. Om *continuous delivery* te realiseren, is het nodig veranderingen aan te brengen in zowel processen als structuur van de beheer- en ontwikkelafdeling.

alsnog extra tijd nodig hebben om problemen op te lossen. Ook moet men zich hierbij realiseren dat software weliswaar agile opgeleverd kan worden, maar dat een eindproduct doorgaans bestaat uit verschillende software-onderdelen waardoor de oplevering veel minder iteratief is. Kortom, de agile-ontwikkelmethode is een stap in de goede richting, maar de grotere uitdaging voor organisaties is om zowel flexibeler als efficiënter en sneller software te kunnen opleveren. Voor het behalen van deze flexibiliteit, efficiëntie en snelheid in softwareontwikkeling is de 'continuous delivery'-methode zeer geschikt. Deze vrij nieuwe methode is er juist op gericht om nieuwe ideeën zo snel en efficiënt mogelijk in productie te krijgen.

Flexibel, efficiënter en sneller

Continuous delivery zorgt voor het aanpakken van de gehele levenscyclus van software. Het is een end-to-end-werkwijze; van analyse, ontwerp en cyclisch opleveren, naar test, deployment en feedback.

Het opleveren van software wordt bij *continuous delivery* een iteratief proces waarbij continu kleine stukken functionaliteit automatisch worden uitgerold. Hierdoor komen deploy-stappen vaker in het OTAP (Ontwikkel, Test, Acceptatie en Productie)-proces voor, waardoor software-releases vaker plaatsvinden en functionaliteit in een doorlopend proces wordt getest en stap voor stap wordt verbeterd. Doordat functionaliteit vaker en sneller kan worden opgeleverd, kun je beter op de behoeften van klanten inspelen. Door zeer korte cycli is de impact van wijzigingen klein. Dit zorgt ervoor dat zowel het team als de klant de impact van wijzigingen goed kan overzien. Dat maakt bijsturen makkelijker en er worden minder fouten gemaakt. Aanpassen van de software wordt ook minder 'bedreigend', want de impact is immers klein. Dit stimuleert de orga-

nisatie om te experimenteren. Het automatisch testen zorgt er bovendien voor dat men eerder inzicht krijgt in de kwaliteit van applicaties. Het voordeel is dat ontwikkelteams ook sneller feedback krijgen van de eindgebruikers. Eventuele fouten kunnen zo ook snel worden opgelost en verbeteringen eenvoudiger toegevoegd. Bij *continuous delivery* wordt alleen ontwikkeld wat nodig is. Dit zorgt voor een veel efficiëntere manier van werken. Veel features in huidige software worden namelijk niet of nauwelijks gebruikt. Als men weet wat in de praktijk de essentiële features zijn, wordt overbodig werk bespaard. Door de software kortcyclisch in productie te brengen en gebruikers er mee te laten werken, komt de organisatie erachter welke functionaliteit ontbreekt en wat overbodig is. Hierdoor ontwikkelt men alleen features waar echt behoefte aan is.

De verantwoordelijkheid voor de hele levenscyclus wordt bij *continuous delivery* ondergebracht in gezamenlijke teams, en onderscheid tussen ontwikkelen en beheer vervaagt hierbij. Alle fasen van de levenscyclus worden verregaand geautomatiseerd met standaard tooling die zich in de markt bewezen heeft. De invoering van *continuous delivery* zal leiden tot het vergroten van de kwaliteit van IT en het reduceren van de kosten die gemeoid zijn met IT.

Noodzakelijke procesverandering

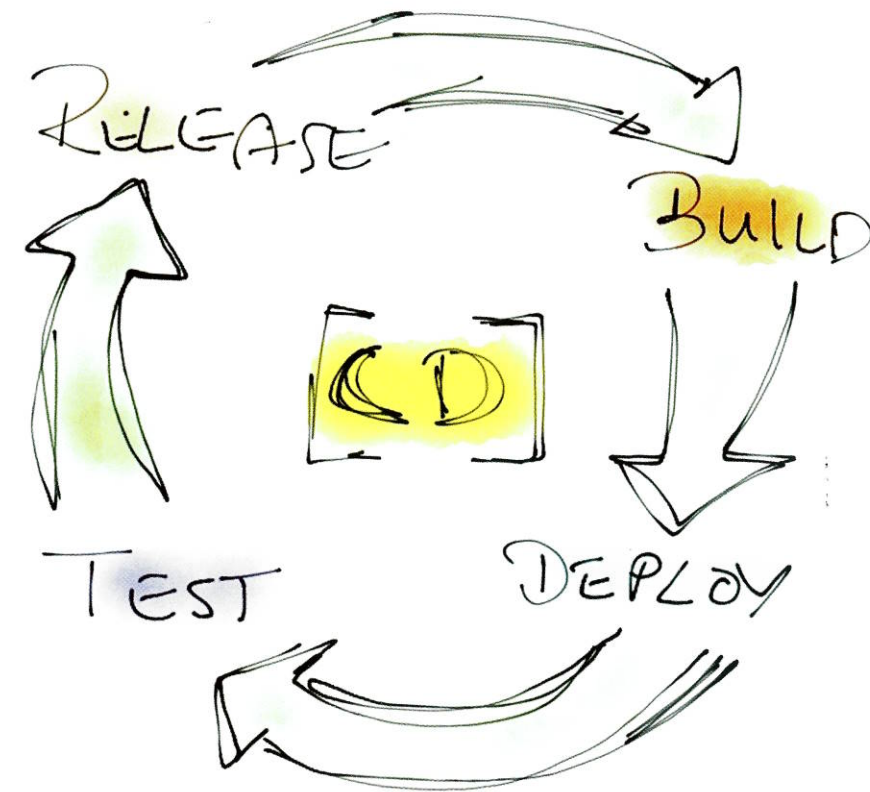
Bij *continuous delivery* is het belangrijk dat mensen, processen en tools goed zijn afgestemd. Ontwikkelaars, testers en beheerders werken intensief samen om in iteraties functionaliteit van hoge kwaliteit op te leveren. Hierbij is niet alleen goede communicatie belangrijk, maar ook het volgen van de agile- en iteratieve processen. Verder moet voor iedereen precies duidelijk zijn hoe het uitrolproces opgezet is. IT-afdelingen die volgens de waterval- of agile-

methode werken, zullen de verschillende fasen, zoals ontwikkelen, testen, acceptatie en securityafstemming, stap voor stap en afzonderlijk uitvoeren. Bij *continuous delivery* daarentegen is het belangrijk dat ontwikkelaars, testers en andere IT'ers juist continu samenwerken en zoveel mogelijk taken automatiseren zodat snel kan worden overgegaan naar productie. Om *continuous delivery* succesvol te realiseren, is het noodzakelijk om gebruik te maken van tools die dit proces ondersteunen. Zo is het belangrijk om een 'deployment pipeline' samen te stellen. Deze pipeline zorgt ervoor dat wat geprogrammeerd is in de ontwikkelomgeving, snel overgaat naar functionaliteit en dat deze snel in productie kan worden gebracht.

Herstructurering IT-afdeling

Continuous delivery vraagt ook om een herinrichting van de IT-afdeling. De verschillende fasen in het ontwikkelproces moeten niet langer worden gezien als afzonderlijke onderdelen en ontwikkelaars en beheerders werken meer als één team samen in een DevOps-structuur. Deze structuur dicht de kloof tussen ontwikkeling en beheer. Vanuit het streven om zo snel mogelijk en op een betrouwbare manier nieuwe functionaliteit voor software op te leveren, valt er vaak winst te behalen door de overdracht van software van de ontwikkel- naar de beheerafdeling te verbeteren of zelfs op te heffen. Overdracht zorgt namelijk vaak voor kennisverlies en levert communicatieproblemen op.

Naast het veranderen van werkprocessen vraagt de transformatie ook een andere rol van de teamleden. Belangrijke vaardigheden zijn communiceren en samenwerken. Het is essentieel dat de teamleden elkaar door en door begrijpen om samen tot het gewenste resultaat te komen. Dit kan wel eens lastig zijn, aangezien ontwikkelaars en beheerders traditioneel gezien andere belangen hebben. Een ontwikkelaar wil zo snel en zoveel mogelijk functionaliteit naar productie brengen. Een beheerder daarentegen zoekt juist naar stabiliteit en zo min mogelijk veranderingen. Daarnaast zijn beheerders van nature defensiever van aard en ontwikkelaars juist opportunistisch. Maar door beide persoonlijkheden in een team te plaatsen, creëer je balans en worden besluiten beter afgewogen. Vanuit de DevOps-aanpak streven beiden namelijk hetzelfde doel na; het opleveren van kwalitatief goede functionaliteit waar de klant direct iets aan heeft. Goede communicatie en nauwe samenwerking is nodig om



» Bij *continuous delivery* worden continu kleine stukken functionaliteit automatisch uitgerold «

focus op kwaliteit te behouden en het ontwikkelproces efficiënt te laten verlopen.

Rollen veranderen

Door de veranderende rollen is niet iedereen even enthousiast over *continuous delivery*. Beheerders zijn met *continuous delivery* bijvoorbeeld minder tijd kwijt aan het uitrollen van de software en er is meer tijd nodig voor samenwerking met het team, de automatisering van zoveel mogelijk taken en het verder verbeteren van het deployment-proces. Teams die zich volledig richten op het testen van software passen niet meer in de aanpak. Met *continuous delivery* zijn multidisciplinaire teams nodig die deze verschillende onderdelen uit ontwikkeltrajecten in nauwe samenwerking en in een iteratief proces doorlopen. Het automatiseren van terugkerende taken speelt een belangrijke rol. Enerzijds om de kans op fouten door handmatig werken te verminderen. Anderzijds kunnen teamleden meer tijd steken in samen-

»Om continuous delivery te realiseren, is het nodig veranderingen aan te brengen in zowel processen, als structuur van de beheer- en ontwikkelafdeling«

werking en kwaliteitsbewaking en tegelijkertijd sneller en efficiënter werken.

De consequenties voor de IT-afdeling kunnen groot zijn. Beheerders en ontwikkelaars kunnen makkelijk denken dat ze overbodig worden. Maar de ontwikkeling naar een continuous delivery-georiënteerde organisatie gaat stapsgewijs. Alle medewerkers worden hierin meegenomen. Het uiteindelijke doel is het samenbrengen van alle competenties en vaardigheden in één DevOps-team waarbij software-engineering centraal staat. Binnen een team heeft iedereen zijn eigen rol, maar heeft iedereen ook kennis van, en begrip vóór de andere rollen. Een groot voordeel van deze manier van werken is dat alle teamleden bij het ontwikkelen rekening houden met elkaar. Ze zijn op de hoogte van de stappen die voorafgegaan zijn in het ontwikkelproces, en houden rekening met de stappen die nog gaan komen. Hierdoor raken taken verweven en worden fouten en problemen snel opgespoord en opgelost. Een DevOps-team is als geheel verantwoordelijk voor het opleverproces. Hoe goed een team ook is, er is altijd ruimte voor verbetering. Het is belangrijk dat teams alert blijven om verbeterpunten te signaleren. Dit helpt om steeds meer onderdelen te automatiseren of op een andere manier te verbeteren zodat in steeds kortere tijd, efficiëntere software van toenemende kwaliteit kan worden opgeleverd.

Transformatie

Verandering binnen een organisatie zorgt regelmatig voor onrust en verzet. Niet iedereen is meteen even enthousiast over het werken met een nieuwe ontwikkelmethode en teamstructuur. Maar zowel terughoudende als vooruitstrevende medewerkers kunnen prima een rol vervullen in een DevOps-team. Een optimaal DevOps-team is in balans en heeft alle ontwikkel-skills in huis.

Daarnaast beschikt het over medewerkers die verschillend zijn van karakter.

Voor het samenstellen van een DevOps-team, heb je medewerkers nodig die bereid zijn om op deze nieuwe manier te werken. Zij zullen minder in eigen taken moeten denken, en meer het geheel van het ontwikkelproces moeten inzien. Verantwoordelijkheden worden grensoverschrijdend. Idealiter zijn alle teamleden 'T-shaped': kennis en interesse zowel in de breedte als in de diepte, met een specialisme in een aantal gebieden. Ze hebben basiskennis over alle onderdelen van de keten zoals testen, ontwikkeling en beheer, maar ze worden vooral ingezet op een eigen vakgebied.

Het transformeren van een bestaande ontwikkelafdeling naar een DevOps-team is een traject voor de lange termijn. Het vergt niet alleen een andere samenwerkingsvorm, maar een gehele verandering van de bedrijfscultuur. Meestal duurt het dan ook enkele maanden tot jaren voor een DevOps-team gerealiseerd is.

Start-up

Hoewel de continuous delivery methode ook is toe te passen zonder DevOps-teams in te richten, maakt het continuous delivery wel veel makkelijker. Bij continuous delivery doorloopt software bijzonder snel de gehele keten. Soms zelfs binnen een dag. Wanneer je aparte afdelingen handhaaft, is het haast onmogelijk om de gehele keten binnen een dag te doorlopen. DevOps-teams verhogen de snelheid aanzienlijk waardoor resultaat verbetert en kosten verlagen.

De voordelen van continuous delivery zijn aantrekkelijk. Continuous delivery staat als ontwikkelmethode pas aan het begin. Waar start-ups dus vaak niet anders werken dan via deze methode, zijn grotere organisaties nu pas de mogelijkheden aan het verkennen. Veel organisaties

VECOZO

Een organisatie die op dit moment bezig is met de overstap naar continuous delivery is VECOZO. Deze organisatie biedt een portaal waarin ketenpartijen in de zorg snel, veilig en eenvoudig gegevens met elkaar kunnen uitwisselen. De organisatie biedt een twintigtal verschillende diensten en heeft vijf ontwikkelteams die volgens de Scrum-methodiek bezig zijn met de ontwikkeling van applicaties. In de oude situatie werkte VECOZO met een hele korte ontwikkelcyclus (sprint) om snel en agile op te kunnen leveren. Een aparte applicatie-beheerafdeling was verantwoordelijk voor de uitrol van applicaties naar de productieomgeving. Dit gebeurde één keer per maand, op een vast moment. Op dat moment ging het systeem uit de lucht en werden updates en wijzigingen voor alle diensten doorgevoerd. Voor de organisatie was

dit geen ideale situatie, men wilde van de downtime af. Niet alleen omdat het hele systeem dan plat lag, maar eens per maand was een grote groep medewerkers tot 's nachts bezig met de updates. VECOZO wilde dit anders en keek naar oplossingen.

Inrichting ontwikkelproces

Continuous delivery lijkt voor VECOZO de ideale oplossing. Om over te kunnen stappen moet het deployment-proces anders ingericht worden. Een belangrijke eis hierbij is dat de organisatie



dezelfde kwaliteit en beschikbaarheid moet kunnen blijven leveren als voorheen. Het testproces moet daarvoor geautomatiseerd worden. Dit is gedaan door SpecFlow in te zetten. Binnen de nieuwe ontwikkelomgeving wordt gebruikgemaakt van continuous deployment. Met als doel dat alle wijzigingen daarin geautomatiseerd uitgerold worden en de bijbehorende testen worden uitgevoerd. Eerder was de uitrol van nieuwe applicaties en functionaliteit met name gericht op het downtime-moment waarin alles in één keer werd uitgerold. Nu is VECOZO bezig om binnen het huidige proces de automatisering zo in te richten dat het per dienst kan gebeuren en dus in verschillende fases. Het uiteindelijke hoofddoel voor VECOZO is het volledig terugbrengen van de downtime. Dit gebeurt nu in verschillende stappen waarmee de werkwijze van de volledige ontwikkelafdeling wordt omgevormd naar de continuous delivery-methode.

vinden het lastig over te stappen. Om continuous delivery stapsgewijs in te voeren, kunnen grotere organisaties een voorbeeld nemen aan start-ups. Voor kleinere bedrijven is het, vanwege de kleinere ontwikkelafdelingen, vaak makkelijker om in een DevOps-structuur samen te werken. Bij grotere organisaties vergt dit meer verandering en aanpassing.

Met het inrichten van één of twee kleinere DevOps-teams, kunnen grotere organisaties de samenwerkingsvorm op eenzelfde manier als bij start-ups vormgeven en testen. Ook de rest van de IT-afdeling en zelfs de gehele organisatie kan op deze manier geleidelijk kennis maken met DevOps. Om de gehele IT-afdeling in te delen in DevOps-teams, zal uiteindelijk natuurlijk de bedrijfsvoering veranderd moeten worden. Kortom, om continuous delivery te realiseren, is het nodig veranderingen aan te brengen in zowel

processen als structuur van de beheer- en ontwikkelafdeling. Hoewel dit intern wat strubbelingen met zich mee kan brengen, is het belangrijk dat de eindgebruiker altijd centraal blijft staan. Functionaliteiten worden gebouwd voor klanten en het ontwikkelproces is pas afgerond als de eindgebruiker er succesvol mee kan werken.

Dennis Joosten

Joosten is businessunit-manager.
E-mail: Dennis.Joosten@infosupport.com

Joop Snijder

Snijder is product owner Endeavour knowNow.
E-mail: Joop.Snijder@infosupport.com