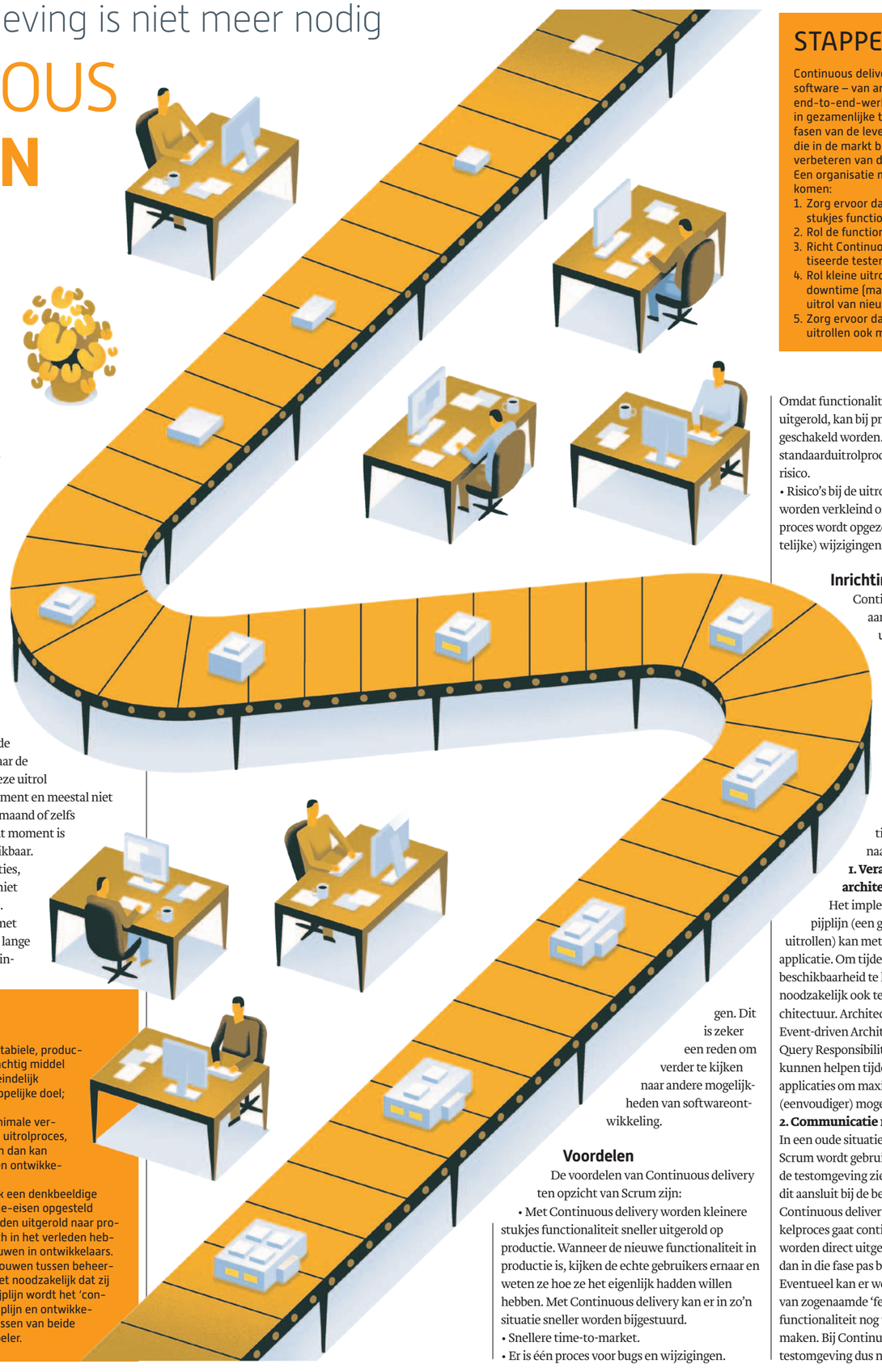
Wachten op een testomgeving is niet meer nodig

MET CONTINUOUS DELIVERY GEEN DOWNTIME

‘Continuous delivery’ is een snelle en flexibele manier van softwareontwikkeling en een goede aanvulling op Scrum. Dé manier, zegt Erik Steinebach, voor organisaties om zich te onderscheiden. Hij laat zien wat ervoor nodig is om het ontwikkelproces uit te bouwen naar deze methode.

door: ERIK STEINEBACH **beeld:** MARC KOLLE



Organisaties staan door opkomende technologieën voor nieuwe uitdagingen. Door de crisis zijn bedrijven echter steeds voorzichtiger met hun uitgaven en nemen ze minder risico’s. Terwijl klanten juist meer vragen van de (online) dienstverlening en er kansen ontstaan die organisaties niet moeten laten liggen. Organisaties die zich willen onderscheiden moeten zich richten op de dienstverlening, en nieuwe technologieën optimaal benutten. ‘Continuous delivery’ kan organisaties hierbij helpen. Maar hoe werkt het en hoe kun je de business de voordelen van deze ontwikkelmethode laten zien?

Overstappen van bijvoorbeeld Scrum naar Continuous delivery is niet iets wat binnen een dag is gebeurd. De Scrum-methode biedt wel uitstekende mogelijkheden voor een overstap. Continuous delivery is er namelijk een goede aanvulling op. Wanneer binnen een organisatie gebruik wordt gemaakt van Scrum voor de ontwikkeling van applicaties, dan zijn er vaak hele korte ontwikkelcycli (sprints) om snel en agile op te kunnen leveren. Vaak is dan een aparte applicatiebeheerafdeling verantwoordelijk voor de uitrol van applicaties naar de productieomgeving. Deze uitrol gebeurt op een vast moment en meestal niet vaker dan één keer per maand of zelfs minder frequent. Op dat moment is het systeem niet beschikbaar. Deze uitrol van applicaties, inclusief downtime, is niet ideaal voor organisaties. Ook is de afstemming met de klant lastig door een lange doorlooptijd van wijzigingen. Dit is zeker een reden om verder te kijken naar andere mogelijkheden van softwareontwikkeling.

Voordelen

De voordelen van Continuous delivery ten opzicht van Scrum zijn:

- Met Continuous delivery worden kleinere stukjes functionaliteit sneller uitgerold op productie. Wanneer de nieuwe functionaliteit in productie is, kijken de echte gebruikers ernaar en weten ze hoe ze het eigenlijk hadden willen hebben. Met Continuous delivery kan er in zo’n situatie sneller worden bijgestuurd.
- Snellere time-to-market.
- Er is één proces voor bugs en wijzigingen. Omdat functionaliteit snel kan worden uitgerold, kan bij productiestoringen snel geschakeld worden. Bij een storing wordt het standaarduitrolproces gebruikt en loop je minder risico.
- Risico’s bij de uitrol van nieuwe functionaliteit worden verkleind omdat een geautomatiseerd proces wordt opgezet en kleinere (dus overzichtelijke) wijzigingen worden doorgevoerd.

Inrichting

Continuous delivery sluit dus goed aan op Scrum. Maar om Scrum uit te bouwen naar Continuous delivery, moet wel het uitrolproces anders worden ingericht. Een belangrijke eis hierbij is dat de organisatie dezelfde kwaliteit en beschikbaarheid moet kunnen blijven leveren als voorheen. Onderstaande best practices kunnen organisaties helpen bij een overstap naar Continuous delivery:

1. Verandering aan applicatiearchitectuur

Het implementeren van een deliverypijplijn (een geautomatiseerd proces van uitrollen) kan met bijna elke bestaande applicatie. Om tijdens de uitrol volledige beschikbaarheid te kunnen bieden, is het noodzakelijk ook te kijken naar de applicatiearchitectuur. Architectuurpatronen zoals Event-driven Architecture (EDA) en Command Query Responsibility Segregation (CQRS) kunnen helpen tijdens het ontwerp van applicaties om maximale beschikbaarheid (eenvoudiger) mogelijk te maken.

2. Communicatie met eindgebruikers

In een oude situatie, wanneer bijvoorbeeld Scrum wordt gebruikt, kan de eindgebruiker in de testomgeving zien wat het resultaat is en of dit aansluit bij de behoefte. Bij de overstap naar Continuous delivery verandert dit. Het ontwikkelproces gaat continu door en wijzigingen worden direct uitgerold in productie. Er wordt dan in die fase pas bijgestuurd en/of aangepast. Eventueel kan er wel gebruik worden gemaakt van zogenaamde ‘feature toggles’ om nieuwe functionaliteit nog tijdelijk niet zichtbaar te maken. Bij Continuous delivery is wachten op de testomgeving dus niet nodig, waardoor processen veel sneller zijn, bijsturen eenvoudiger is, en men niet afhankelijk is van planningen van de eindgebruikers. Dit vraagt in een aantal gevallen wel om andere communicatie met eindgebruikers. Een maandelijks overzicht van uitgevoerde wijzigingen verandert in een altijd inzichtelijk overzicht van wijzigingen die in ontwikkeling zijn met daarbij de status van deze wijziging. Het is belangrijk deze communicatie niet uit het oog te verliezen.

3. Voordelen voor de business

Voor IT is het eenvoudig om de toegevoegde waarde van Continuous delivery aan de business te tonen. De organisatie wordt namelijk een stuk flexibeler als het gaat om aanpassingen in applicaties. Er is een veel snellere time-to-market. Wanneer je met een maandelijkse uitrol werkt, kan het soms bijna twee maanden duren voordat functionaliteit wordt doorgevoerd.

Overstappen

Bij een overstap naar Continuous delivery geldt ook dat niet elke organisatie hetzelfde is en eenzelfde aanpak zal kiezen. Een publieke website moet bijvoorbeeld 24/7 online en beschikbaar zijn, downtime is daar niet mogelijk. Voor zakelijke applicaties is het misschien een minder groot probleem wanneer een applicatie na 18:00 uur ‘s avonds een keer offline is. In het laatste geval heb je als organisatie dus de kans om Continuous delivery in stappen in te voeren. Bijvoorbeeld door functionaliteit eerst ‘s avonds geautomatiseerd uit te rollen, misschien zelfs eerst met wat downtime. Daarna kan het ‘s avonds zonder downtime worden uitgerold. Mocht er dan iets fout gaan dan is er op dat moment een minder groot risico dat gebruikers worden getroffen. Wanneer alles optimaal werkt en getest is, kan er vervolgens overdag en zonder downtime worden uitgerold. Een volledige overstap naar Continuous delivery dus. «

Erik Steinebach is software engineer bij Info Support (www.infosupport.nl)

STAPPENPLAN

Continuous delivery zorgt kortgezegd voor het aanpakken van de gehele levenscyclus van software – van analyse, ontwerp en cyclisch opleveren naar test, uitrol en feedback – in één end-to-end-werkwijze. De verantwoordelijkheid voor de hele levenscyclus wordt ondergebracht in gezamenlijke teams en het onderscheid tussen ontwikkelen en beheer vervaagt hierbij. Alle fasen van de levenscyclus worden verregaand geautomatiseerd met bewezen, standaardtooling die in de markt beschikbaar is. De invoering van Continuous delivery zal uiteindelijk leiden tot het verbeteren van de kwaliteit van IT en het reduceren van de kosten die hiermee gemoeid zijn.
Een organisatie moet een aantal stappen doorlopen om uiteindelijk tot Continuous delivery te komen:

1. Zorg ervoor dat er volgens een methode als Scrum of Kanban ontwikkeld wordt, dus kleine stukjes functionaliteit per keer.
2. Rol de functionaliteit uit en automatiseer acceptatie- en regressietesten.
3. Richt Continuous deployment in naar een testomgeving (ter ondersteuning van de geautomatiseerde testen).
4. Rol kleine uitrollen uit door de hele OTAP-straat (Continuous delivery). Dit is dan nog wel met downtime (maar bijvoorbeeld na 18:00 uur), maar minder lang dan voorheen en er is vaker een uitrol van nieuwe functionaliteit.
5. Zorg ervoor dat applicaties tijdens een uitrol beschikbaar blijven, hierdoor wordt overdag uitrollen ook mogelijk.

DEVOPS

Het doel van DevOps is dat ontwikkelaars moeten leren om hoge kwaliteit, stabiele, productierijpe software te ontwikkelen en beheerders dat agile-technieken een krachtig middel zijn om effectief, snel en met een laag risico wijzigingen door te voeren. Uiteindelijk bereiken ontwikkelaars en beheerders met DevOps samen het gemeenschappelijke doel; het creëren van waarde voor de klant.
De doelstelling van Continuous delivery, namelijk het regelmatig en met minimale verstoring voor de gebruiker uitrollen van een applicatie, vraagt om een stabiel uitrolproces, waarbij de applicatie-architectuur en het uitrolproces in elkaar vallen. Alleen dan kan maximale beschikbaarheid behaald worden en hiervoor is afstemming tussen ontwikkelaar en beheerder noodzakelijk.
Bij organisaties waar men Continuous delivery wil implementeren moet vaak een denkbeeldige muur doorbroken worden. De beheerorganisatie heeft een set van acceptatie-eisen opgesteld waaraan de ontwikkelafdeling moet voldoen voordat een applicatie kan worden uitgerold naar productie. Deze set van eisen is vaak opgesteld als reactie op problemen die zich in het verleden hebben voorgedaan, opgesteld door beheerders uit angst en gebrek aan vertrouwen in ontwikkelaars. De deliverypijplijn (het geautomatiseerde uitrolproces) kan helpen het vertrouwen tussen beheerders en ontwikkelaars te herstellen. Om een deliverypijplijn in te richten is het noodzakelijk dat zij met elkaar praten over hoe die pijplijn ingericht moet worden. De deliverypijplijn wordt het ‘contract’ tussen beheerders en ontwikkelaars. Beheer heeft controle over de pijplijn en ontwikkelaars hebben de gewenste transparantie over het uitrolproces. Omdat processen van beide afdelingen zo op elkaar zijn afgestemd, is de organisatie efficiënter en flexibeler.

ag
Voor reacties en nieuwe bijdragen van IT-experts:
Henk Ester, 070 3046812
h.ester@automatiseringgids.nl