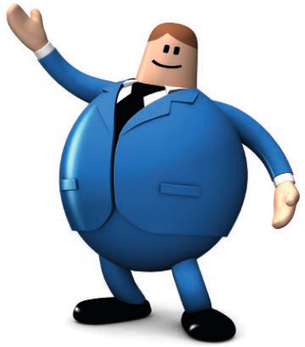
DEVOPS IS GEEN DOEL, VAAK RELEASEN WEL

BELANGRIJKSTE DOELSTELLING: RELEASE KLEIN EN FAAL SNEL

ONDER INVLOED VAN AGILE WERKEN EN DOOR DE TOENEMENDE DYNAMIEK IN DE MARKT WORDT RELEASEN AL SNEL EEN BOTTLENECK. DE VRAAG IS: HOE KUNNEN WE HET RELEASE-PROCES FUNDAMENTEEL VERSNELLEN? RINI VAN SOLINGEN LAAT ZIEN WELKE MAATREGELEN IN DE PRAKTIJK WERKEN EN ILLUSTREERT DIT AAN DE HAND VAN ERVARINGEN BIJ BOL.COM.

door Rini van Solingen



VEEL ORGANISATIES HEBBEN VANUIT HET VERLEDEN EEN STRIKTE SCHEIDING AANGEBRACHT TUSSEN ONTWIKKELING EN BEHEER, of zoals het ook vaak wordt genoemd: tussen Dev (development) en Ops (operations). Deze scheiding is op zichzelf logisch. Het maakt duidelijk wie verantwoordelijk is voor vernieuwing (Dev) en wie verantwoordelijk voor het operationele systeemlandschap (Ops).
Echter, door het invoeren van agile komt deze splitsing vaak onder druk te staan. Door het iteratieve karakter van agile gaan Dev-teams zo frequent nieuwe versies opleveren dat de Ops-teams deze niet meer kunnen releasen. Het gevolg is een groeiende berg software die maar niet live wordt gezet. En dat is voor beide teams onplezierig.
Als reactie praat men daarom vaak over DevOps: agile-teams waarin zowel ontwikkeling (Dev) als beheer (Ops) plaatsvindt. ‘Eat your own dogfood’ is dan het mantra, met het idee dat als teams zelf verantwoordelijk zijn voor Dev én Ops zij op een heel andere manier naar kwaliteit en beschikbaarheid kijken. Dat is slechts ten dele waar. Impliciet neemt men dan namelijk aan dat Dev-teams soms geneigd zijn slechte kwaliteit software te maken omdat zij zelf toch de Ops-taken niet uitvoeren. En dat is onzin. Het ligt niet aan hun persoonlijke professionaliteit maar aan de praktische (on)mogelijkheden om de kwaliteit door hen te laten beheersen en controleren. Als veel teams heel veel versies opleveren en die gaan maar één keer per kwartaal gezamenlijk live, dan is het logisch dat dit compleet fout gaat. Dat is de oorzaak, niet het ontbreken van DevOps-teams.

Het label DevOps is niet waar het echt om draait

OPKNIPPEN

De enige manier om goed te worden in releasen is: door het heel vaak te doen. Bol.com heeft zich, in de periode 2014 tot nu, geconcentreerd op het verhogen van de releasesnelheid. Zij hebben geleerd dat concrete maatregelen het releaseproces drastisch versnellen

AUTEUR

RINI VAN SOLINGEN is deeltijdhoogleraar aan de TU-Delft en CTO bij Prowareness (r.vansolingen@prowareness.nl). Hij is auteur van onder meer: De Kracht van Scrum, De Bijenherder en De Responsive Enterprise.

REACTIES EN BIJDRAGEN
Voor reacties en nieuwe bijdragen van IT-experts:
Henk Ester
020-2356415
h.ester@agconnect.nl

> “Opknippen is de enige juiste oplossings-richting.”

(zie kader ‘Zes maatregelen’).
De praktijk laat zien dat het veel handiger is om heel vaak en veel kleine releases te doen. Hierdoor worden problemen eerder inzichtelijk en kunnen daardoor veel sneller verholpen worden. Dat klinkt eenvoudiger dan het is. De onderlinge afhankelijkheid tussen systemen en componenten is in de praktijk vaak zo groot dat het onmogelijk is losse onderdelen apart te releasen. En dat is dus het bronprobleem. Opknippen is de enige juiste oplossings-richting. Zelfs wanneer dat in eerste instantie erg lastig lijkt. Dit kan echter stap voor stap en levert ook direct meetbare voordelen op (zie kader ‘Praktijkvoordelen in cijfers’). Bovendien zorgt elke kleine verbetering, die de releasesnelheid verhoogt, ervoor dat extra tijd en ruimte ontstaat om de volgende stappen te zetten.

LASTIG

Kortom, de oplossing ligt in het rigoureus versnellen van het releaseproces zelf. Releasesnelheid verhogen is een investe-ring die tijd en geld kost. En het is in het begin vaak lastig. Maar helaas, er zijn feitelijk geen alternatieven. Snel starten is daarom sterk aanbevolen. Focus op het wegnemen van afhankelijk-heden tussen losse onderdelen van een systeem, zodat deze onafhankelijk van

ZES MAATREGELEN OM RELEASESNELHEID TE VERGROTEN

1 ALLES TEN DIENSTE VAN HET RELEASE-PROCES STELLEN

De manier om te zorgen dat een bottle-neck de minste problemen geeft, is door hem het meest belangrijk te maken. Alle andere activiteiten staan dan ten dienste van het releaseproces en dit wordt met militaire precisie gemanaged. Dit kan een (tijdelijke) oplossing zijn, maar lost het bronprobleem niet op, en blijkt vaak op termijn toch onvoldoende.

2 MONOLIETEN OPKNIPPEN

De fundamentele maatregel is om grote monolithische systemen op te knippen in componenten die elkaar onderling met services bedienen en die separaat te releasen zijn. De keuze hoe deze knipbeurt er uitziet is een architectuurvraagstuk waarbij SOA en microservices een belangrijke rol vervullen. Een bijkomend voordeel is dat het totale systeem daardoor veel betrouwbaarder wordt. Een probleem in een van de delen zorgt er dan namelijk niet meer voor dat alles platligt.

3 VEEL EN KLEINE RELEASES

Niet per se vaker releasen maar kleiner releasen. Leg de focus op het doen van kleine releases, zodat veel duidelijker is of iets werkt en dat het niet ergens anders aan ligt. ‘Release small, fail fast’ is het credo. Als gevolg wordt er dan vanzelf vaker gereleased; gewoon omdat het kan. Losse delen separaat kunnen releasen en automatische testen kunnen opstarten, is daarvoor een randvoor-waarde (zie ook het AG Connect artikel van april 2017: Binnen 90 dagen over op Continuous Delivery).

4 ALLE RELEASE-STAPPEN AUTOMATISEREN

Alle activiteiten die een Ops-engineer uitvoert voor een release, voorzien van tools zodat agile-teams dit zelf kunnen doen. De rol van de Ops-engineer verschuift daardoor volledig. Niet achteraf handmatig releasestappen zetten, maar vooraf releasetools neerzetten voor anderen. Feitelijk betekent dit dat Ops-teams zichzelf ‘wegautomatiseren’ en zich bezighouden met problemen voorkomen in plaats van achteraf oplossen. Brandpreventie in plaats van blussen.

5 TEAMS AUTONOOM MAKEN EN ZELF LATEN RELEASEN

Om er voor te zorgen dat teams volledig eigenaarschap pakken, is het belangrijk dat ze autonoom zijn. Een component is van henzelf en zij zijn er end-to-end verantwoordelijk voor. In de traditionele scheiding tussen Dev en Ops betekent dit dat alle Ops-taken en Dev-taken door een agile-team worden gedaan, met behulp van tools die door de Ops-teams beschikbaar worden gesteld. Daardoor komen de Ops-teams van het kritieke pad af en zijn zij niet langer de bottleneck.

6 VOORDELEN HOGE RELEASESNELHEID INZICHTELIJK MAKEN

Elke verbetering vraagt ook om een investering, in aandacht, inspanning, tijd, geld en geduld. Als voor iedereen duidelijk is wat de voordelen zijn van vaak releasen, is de moeite die het kost makkelijker te accepteren. Dit geldt vooral voor de businessafdelingen waarvoor de agile-teams werken. Het creëren van enthousiasme bij de businessteams werkt als een belangrijk vliegwiel voor alle andere maatregelen.

PRAKTIJKVOORDELEN IN CIJFERS

Het toepassen van de, in het andere kader genoemde, maatregelen heeft bij bol.com geleid tot een aantal meetbare voordelen:

- Modulariteit website vervijfvoudigd
  De hele website moest opgeknipt worden in losse delen die apart van elkaar gereleased konden worden. Dit bleek in de praktijk de grootste klus, maar dit is gelukt. Eind 2017 is het aantal services gegroeid naar meer dan 250, die elk separaat te releasen zijn. En deze onderdelen functioneren ook los van elkaar. Als bijvoorbeeld de zoekfunctie uitvalt, dan werkt de rest van de website nog wel gewoon en kunnen klanten die reeds aan het betalen zijn, gewoon verder. Of wanneer het tonen van een gepersoni-ficeerde aanbieding is uitgevallen, dan verschijnt een generieke aanbieding.

De rol van Ops-engineer verschuift volledig

- Aantal releases extreem gegroeid
  Het aantal releases is gestegen, van 1 release per maand in 2014 naar meer dan 550 releases per maand in 2017. Dit was een direct gevolg van het opknippen van monolieten in losse delen. Hierdoor worden componenten separaat gere-leased. Bovendien gebeurt het live zetten niet langer door Ops-teams maar door de agile-teams zelf. Alle Ops-engineers zijn overgeplaatst naar een agile-team of naar een SRT (Site Reliability Team). De SRT’s maken infrastructuur en tools voor de agile-teams zodat die daarmee zelf automatische testen kunnen draaien, zelf monitoring kunnen doen, zelf loadtesten kunnen uitvoeren of zelf kunnen releasen.
- Kritische incidenten met meer dan 95 procent gereduceerd
  Het aantal kritische incidenten (met significante impact op reputatie, klanttevredenheid en omzet) is gedaald van gemiddeld meer dan 25 per maand in 2014 naar minder dan 1 per maand in 2017.
- Geplande downtime gereduceerd tot nul
  In 2014 ging de website gemiddeld 4 uur per maand uit de lucht om een release te doen. Sinds 2016 gaat de website niet meer uit de lucht bij een release. Voor een enkele update van een onderliggend platform lukt het nog niet om deze te updaten zonder downtime. De downtime voor dit soort releases is gedaald naar minder dan 18 minuten per jaar.
- Ongeplande downtime praktisch tot nul gereduceerd
  Naast geplande downtime kan de website ook uit de lucht zijn door problemen. Dit is met meer dan 98 procent gereduceerd naar minder dan 10 minuten ongeplande downtime per jaar.

elkaar kunnen worden gereleased. Liefst door de agile-teams zelf, met veel autonomie en zonder een strikte scheiding tussen Dev en Ops. En ja, daar kan dan eventueel best het label DevOps op worden geplakt, maar dat is eigenlijk niet waar het echt om draait.

Het onderzoek dat in dit artikel ter sprake komt, is uitgevoerd in nauwe samenwerking met Frede-rieke Ubels, Thomas de Jong en Nick Tinnemeier van bol.com. Deze webwinkel heeft inmiddels 65 agile DevOps-teams en richt zich sinds 2014 actief op het verhogen van zijn releasesnelheid.