

Control in the pipeline

Hoe beheers en bestuur je DevOps?

Door **Bart de Best** - 18 oktober 2019

Met de komst van DevOps is het begrip 'pipeline' erg in zwang geraakt. Het betreft hier de hele flow van *requirement* tot en met inproductienamen van een oplossing of een aanpassing daaraan.

Zowel mensen, methoden als middelen worden op een efficiënte en effectieve manier aan het werk gezet om business value op te leveren door deliverables voort te brengen door de traditionele ontwikkel-, test-, acceptatie- en productieomgeving waarbij vooral zoveel als mogelijk geautomatiseerd wordt.

Maar hoe zit het nu met de beheersing en besturing (*control*) van het op deze manier vormgegeven ontwikkel- en beheerproces?

Traditioneel is de *control* belegd in strak geprotocolleerde beheerprocessen zoals change management van ITIL en de processen van Prince2. De toekenning van de functies zoals projectmanager, change manager en release manager, die zelf geen deel uitmaakten van het voortbrengingsproces, moesten borgen dat de control geïmplementeerd en verantwoordbaar was.

Met de komst van het agile werken zijn deze controls echter steeds meer losgetrokken van deze functionarissen. Het zijn vaak rollen en zelfs alleen taken die toebedeeld worden aan deelnemers van de DevOps-teams. De control moet uitgeoefend worden door de DevOps-teams zelf en waar mogelijk verankerd worden in geautomatiseerde stappen in de pipeline, waarbij het normenkader geborgd is in de DoR (*definition of ready*) en DoD (*definition of done*).

Het risico

Deze loskoppeling heeft veel voordelen, omdat deze mensen met de vakkennis aangaande de deliverables nu zelf goed moeten nadenken over het in control zijn. Zo moet security een integraal onderdeel vormen van de flow. Dus vanaf non-functional requirement, of operationeel requirement zoals dat tegenwoordig heet. Dit geldt ook voor de continuïteit, beschikbaarheid, performance en capaciteit.

De vraag is alleen of een DevOps-team van maximaal negen mensen al deze competenties diepgaand genoeg kan beheersen om de juiste keuzen te maken. Ook is het de vraag of de DevOps-engineers zich deze competenties wel willen toe-eigenen. En ten slotte is het nog maar de vraag of er ook zoiets als een 'span of knowledge' geldt, ofwel de maximale hoeveelheid kennis en kunde die van een *E-shaped* (multiple expert) DevOps-expert verwacht mag worden.



“ER ONTSTAAT FRICTIE TUSSEN DE VERWACHTING VAN DE PRESTATIES VAN HET DEVOPS-TEAM EN DE GEREALISEERDE SERVICEVERLENING”

Het antwoord op al deze vragen blijkt te zijn, zo is mijn ervaring, dat control wel in de pipeline moet komen, maar het een willekeurig DevOps-team nog ontbreekt aan de nodige kennis en kunde. De pipeline is nog niet volwassen genoeg, vooral vanwege de onvolwassenheid van het DevOps-team.

Daar komt nog bovenop dat het managers soms eigenlijk te goed uitkomt dat er nu één team is dat integraal verantwoordelijk is voor alles. Door deze factoren ontstaat er frictie tussen de verwachting van de prestaties van het DevOps-team en de gerealiseerde serviceverlening.

De oplossing

Ik geloof alleen in kwaliteitsmanagers die met de voeten in de modder staan. Het moeten kwaliteitscoaches zijn die in zeer concrete vaktechnische taal aan de DevOps-teams uitleggen hoe ze control in de pipeline kunnen (of moeten) aanbrengen. Maar dus niet meer de managers die roepende zijn in de woestijn, of zonder leger.

Bij voorkeur zijn het de nieuwe scrum masters die E-shaped zijn in kwaliteitscompetenties. Zij moeten over ontwikkel- en beheervaardigheden beschikken om mensen uit te leggen hoe de pipeline tools, *way of working*, et cetera aangepast moeten worden om te komen tot de juiste control in de pipeline.

De borging

In veel gevallen is dat nog niet genoeg en is er ook nog een corporate DoD nodig waarin de verplichtingen van de organisatie naar de buitenwereld (lees wet- en regelgeving) zijn vertaald naar ontwikkel- en beheerproceseisen. Daarbij zou de DoD bij voorkeur uit vier control onderdelen moeten bestaan:

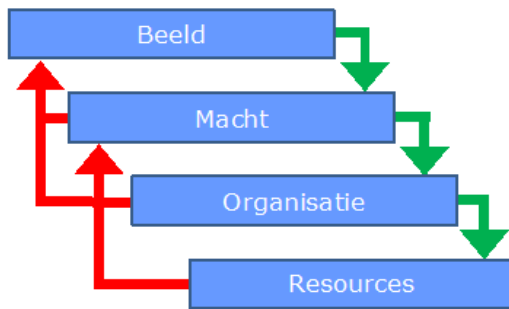
1. Organisatielevel-control met security, compliancy, risk-eisen, gevuld en bewaakt door kwaliteitscoaches.
2. Productlevel-control met de operationele requirements (SLA-normen), gevuld door de product owner.
3. Sprintlevel-control met de acceptatiecriteria per sprint, gevuld door de product owner.
4. Flow-level control met standard, rules en guidelines van het DevOps-team.

Deze vier controlsegmenten van de DoD zouden samen de eisen moeten omvatten die aan de pipeline worden gesteld. Daarmee is de control gedefinieerd en meetbaar geworden zodat verantwoording mogelijk is. De check op de DoD borgt in dat geval dat de control in de pipeline verankerd is en geeft een duidelijke en overzichtelijk definitie van de vereiste externe beïnvloeding op het DevOps-team.

De implementatie

Een visie zoals hierboven besproken vereist een gedegen implementatie. Een goede weg om te bewandelen zijn de gebruikelijke **acht stappen van Kotter**. Een extra idee om naar te kijken is het minder bekende veranderparadigma zoals in figuur 1 weergegeven.





Figuur 1. Veranderparadigma

Beeld. De beeldvorming is de eerste stap bij een verandering. Hierbij wordt vastgesteld of de stakeholders die betrokken zijn bij de verandering dezelfde beeldvorming hebben van waar ze nu staan en waar ze heen willen. Zij moeten de high-level requirements opstellen van de eindsituatie. De huidige situatie (*Ist*), de gewenste situatie (*Soll*) en het migratiepad moeten worden verkend. Hierbij speelt architectuur een belangrijke rol.

Macht. Nadat overeenstemming is bereikt over de route is het belangrijk om vast te stellen wie het voor het zeggen heeft. Wie is de eigenaar van de betrokken services, producten, processen, en dergelijke? Dit eigenaarschap is essentieel om lange discussies te voorkomen en om resources te kunnen claimen voor de inrichting en verrichting van de gekozen richting.

Organisatie. De werkwijze wordt pas bepaald als duidelijk is wat het beeld en de macht inhouden. Dit scheelt heel veel discussies over de way of working (WoW).

Resources. In de laatste stap wordt pas gekeken waar de verandering in de organisatie moet landen en welke functie, rollen, taken en tools er nodig zijn.

Bij een discussie over de resources moet deze gestopt worden. Het is dan zaak bij de betrokkenen na te gaan of duidelijk is wie het voor het zeggen heeft. Dit geldt ook bij een discussie over de WoW, ook dan moet er twee stappen terug worden gegaan en vastgesteld worden of de beeldvorming duidelijk is. De laatste discussie is die over macht. In dat geval is er maar één stap terug te gaan en dat is de beeldvorming.

Voorbeeld

Ter illustratie volgt hier een voorbeeldinvulling van het paradigma van de verandermanager. Figuur 2 geeft een plaat weer die in een workshop van de denkbeeldige organisatie Assuritas is gebruikt. Alle stakeholders zijn uitgenodigd om in een uur tijd te brainstormen over de vier perspectieven van het veranderparadigma. Het was niet toegestaan om te discussiëren. Per perspectief zijn de meningen gevraagd en met post-its op een whiteboard geplaatst. De verschillen van inzicht zijn vastgesteld en in een aantal workshops met de betrokkenen uitgediscussieerd. Het resultaat is in de figuren 3, 4, 5 en 6 weergegeven.



Figuur 2. De basisplaat van control in de pipeline bij Assuritas

Beeld. De eerste brainstormsessie betrof die van de beeldvorming. Al snel werd duidelijk dat control een zeer breed begrip is en dat het niet 'in control' zijn enorme consequenties heeft voor

Assuritas. Ook bleek dat iedereen vond dat je hier tot in het oneindige in kan doorschieten. Economisch omgaan met control werd dan ook als een rode draad gezien in de beheersing. De beste wijze hiertoe werd gevonden in het automatiseren van controls, hergebruik van gevonden oplossingen en beperking van de controls door de gates van de product-backlog en de corporate DoD.

Tot slot was de eis dat het niet in control zijn geborgd dient te worden door *fast feedback* met uitstekende *traceability*. Figuur 3 geeft de belangrijkste bevindingen van de brainstormsessie weer en de daarna gevoerde workshops.



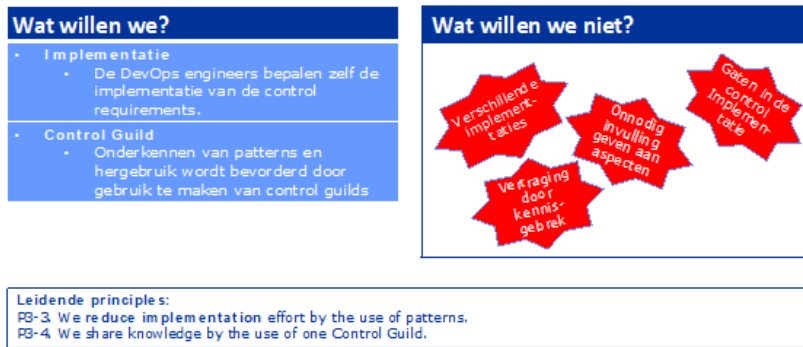
Figuur 3. Control-beeldvorming

Macht. De macht was zoals bij veel organisaties ook bij Assuritas een heikel punt. De brainstormsessie was kort maar de vervolg sessies duurden langer. De macht van de stakeholders was de belangrijkste discussie. Ten slotte is gekozen voor een corporate DoD waarin de control requirements van de stakeholders worden geplaatst als deze organisatiebreed van toepassing zijn. De invulling van de corporate DoD is conform de Kotter-aanpak vormgegeven waarbij vooral aandacht is gegeven aan de *guiding coalition* en de communicatie.



Figuur 4. Control-macht

Organisatie. De brainstormsessie leverde al snel een goed beeld bij de organisatie van de control. De zelfstandigheid van de DevOps engineers is evident. Een gebrek aan kennis en kunde ten aanzien van de controls en hoe deze te implementeren zou echter snel leiden tot versplintering en daarmee *waste* creëren. De zelfstandigheid van de DevOps engineers kan gehandhaafd blijven, mits ze hecht samenwerken in een *control guild* waarin oplossingen voor control worden gedeeld en gechallengeed.



Figuur 5. Control-organisatie



Figuur 6. Control-resources

Resources. De laatste stap was evident en snel besloten. Het tool-portfolio mag niet versnipperd raken en de oplossingen moeten deelbaar zijn. Wel is er nog het risico van fraude dat een vierogenprincipe vereist.

Nadat de vier stappen van het veranderparadigma waren vastgesteld werden de principes meegegeven aan de giding coalition die de verandering bestuurt. Het veranderparadigma werd door de giding coalition breed gecommuniceerd door gebruik te maken van posters, townhall meetings en aanwezigheid in de control guild.

“DE ZELFSTANDIGHEID VAN DE DEVOPS-ENGINEERS KAN GEHANDHAAFD BLIJVEN, MITS ZE HECHT SAMENWERKEN IN EEN CONTROL GUILD”

Tevens werden hackathons georganiseerd op oplossingen te vinden voor moeilijk te implementeren control requirements. Daarmee was de control in de pipeline binnen een jaar grotendeels af en de waste maximaal gereduceerd. Vooral het enthousiasme van de DevOps engineers heeft gezorgd voor vele positieve side effects zoals kennisdeling op vele vlakken en saamhorigheid.

Tot slot

Deze casus geeft een voorbeeldinvulling van de in dit artikel voorgestelde visie om control in de pipeline te krijgen. Het risico van kennisgebrek is in de casus gecompenseerd in de vorm van de control guild. Er zijn ook andere oplossingen denkbaar. De betrokkenheid van de specialisten in de control guild is een prima oplossing omdat hiermee een gelijkwaardige communicatie mogelijk is waar coaching eenvoudig is in te weven. De borging van wet- en regelgeving is ook in deze casus vormgegeven in de zin van een corporate DoD.

Met dank aan Niels Talens voor de review van dit artikel. Meer informatie over het gestructureerd vormgeven van DevOps kun je vinden in mijn publicatie DevOps Architectuur.

Bart de Best

Bart de Best is DevOps-trainer, -auditor en -architect.

