

De digitale transformatie
vanuit technisch perspectief

ZGW
NU

De digitale transformatie vanuit technisch perspectief

In Hoe zaakgericht werken de digitale transformatie kan ondersteunen beschreven we drie perspectieven op de digitale transformatie. Deze white paper beschrijft die digitale transformatie vanuit het technisch perspectief. We kijken hierin naar een aantal aspecten van de digitale transformatie die een impact gehad hebben die de wereld -niet alleen de wereld van IT- getransformeerd heeft.

Van ERP naar open source bouwblokken

In de jaren tachtig ontstonden ERP-pakketten die tot doel hadden de IT voor



standaardprocessen van organisaties volledig af te dekken. Hoewel deze kostbaar waren en lang niet altijd goed pasten, kregen deze een groot marktaandeel bij grote organisaties. Aan de andere kant werden organisatie- en marktspecifieke processen vooral ondersteund met zelf

gebouwde applicaties en kleine standaardpakketten. De integratie van deze kleine pakketten en ERP-pakketten laat vaak te wensen over. De digitale transformatie heeft de voorwaarden gecreëerd voor het breken van de houdgreep op de markt die ERP-leveranciers hebben, door het combineren van verschillende functionele blokken mogelijk te maken.

Open source

In de jaren negentig ontstonden steeds meer initiatieven die gericht waren op het aanbieden van software op basis van open source. In eerste instantie waren dat vooral uit de hand gelopen hobby-projecten van studenten, maar met de tijd kregen die soort initiatieven meer aanhangers en ondersteuning. Hierdoor kon een aanzienlijk aantal van deze initiatieven zich ontwikkelen tot oplossingen die voldoende van kwaliteit, stabiliteit en ondersteuning waren om die ook te gebruiken in zakelijk omgevingen waarin beschikbaarheid een belangrijke speelt. Oplossingen als Linux (besturingssysteem dat vooral veel gebruikt wordt op servers), Apache (webserver) en Alfresco (Document management) zijn open source, maar niet weg te denken uit de markt waarin ze opereren.



Het belang van integratie met open standaarden

Met het toenemen van het aantal open source oplossingen werd het steeds belangrijker



deze op elkaar aan te laten sluiten. In het verleden had alle pakketten hun eigen interface. Dat maakte het kostbaar en moeizaam om die op elkaar aan te sluiten. Deze problemen leidden in eerste instantie tot het ontstaan van service-bussen. Hoewel dit in concept een behoorlijke verbetering was, waren de kosten -zowel financieel als qua inspanning- die met de eerste service-bus-oplossingen gepaard gingen aanzienlijk.

Vanuit de open source wereld kwam hiervoor een betere oplossing: open standaarden.

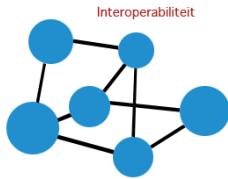


Pakketten kregen een interface die gegevens in een gestandaardiseerd formaat opleverde. Dit nam de noodzaak weg voor het implementeren van de dure service-bus-oplossingen. In eerste instantie werden open standaarden vooral gebruikt voor open source software, maar onder druk van marktvragen en wet- en regelgeving werden steeds meer closed source leveranciers gedwongen communicatie via open standaarden aan te bieden.

Standaard bouwblokken vs standaardpakketten

In het ERP-tijdperk werd het gangbaar om software van een leverancier te gebruiken om het volledige functionele bereik van een organisatie af te dekken. Het nadeel hiervan is dat een leverancier zelden voor alle functionele aandachtsgebieden een goede oplossing kan bieden. Door het gebrek aan goede aansluitmogelijkheden op andere pakketten was het echter ook problematisch om voor ieder gebied de beste oplossing aan te kopen.

De opkomst van open standaarden heeft deze situatie opengebroken. Door de sterke interoperabiliteit van oplossingen op basis van open standaarden is het mogelijk geworden om die beste oplossing voor ieder van de functionele gebieden te implementeren. Dit zorgt voor betere invulling van functionele eisen en wensen van organisaties en gebruikers. Bovendien zorgt het voor meer transparantie in de beschikbare gegevens. Daarnaast verlaagt het de drempel naar hybride oplossingen waarin externe oplossingen of componenten gecombineerd worden met zelfgebouwde software.



Ook deze ontwikkeling zorgt voor een verbetering van de functionele fit die een organisatie kan krijgen op de functionele eisen en wensen.

Het ondersteunen van specifieke processen met eigen oplossingen of kleine marktpakketten wordt hierdoor ook een stuk makkelijker. Het wordt makkelijker voor leveranciers om dit soort software te vermarkten, mede doordat het voor organisaties makkelijker wordt om deze oplossingen te integreren in het systeemlandschap.

Scheiding van procesgegevens en documenten

Het is wenselijk procesgegevens en document van elkaar te scheiden. Dit komt de veiligheid ten goede (toegang tot een soort gegevens betekent niet direct toegang tot alle gegevens)



Het scheiden van procesgegevens en documenten draagt daarnaast bij aan het makkelijker aanpassen van de manier van werken (het proces), zonder dat dit direct impact heeft op de gegevens die een organisatie opslaat (documenten).

Beheer van expertwerk naar commodity

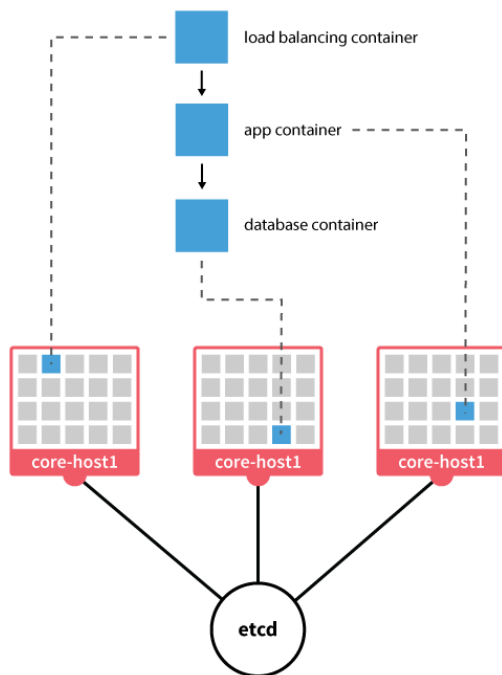
De lappendeken aan oplossingen en interfaces die eerder beschreven werd levert ook in beheer grote problemen. Niet alleen wordt hiervoor een behoorlijke inspanning gevraagd, door de verscheidenheid van oplossingen vraagt het beheer vaak een brede kennis. Daarnaast liet de praktijk nog wel eens zien dat co-existentie van verschillende oplossingen een probleem kon zijn.

Containers vs virtuele servers

In de jaren tachtig en negentig hielden de ontwikkelingen van hardware en software elkaar in balans; weliswaar werd hardware sneller, maar software vroeg ook steeds meer processorkracht, bijvoorbeeld doordat we van command line besturingssystemen als DOS en Unix overgingen naar grafische interfaces als Windows. De afgelopen tien jaar heeft de groei van eisen aan rekenkracht veel aan snelheid verloren. Hierdoor is ruimte ontstaan voor andere zaken.

Virtualisatie maakt het mogelijk een fysieke machine te gebruiken voor meer taken, door meer dan een virtuele server te draaien. Bovendien maakt dit het mogelijk om verschillende

technische omgevingen (combinatie van hardware en besturingssysteem) na te bootsen op een server. Dit biedt organisaties de kans het hardware-park in te perken zonder daarbij concessies te hoeven doen op het gebied van beschikbaarheid van omgevingen.



Het gebruik van container-vorming gaat nog een stap verder; hierbij wordt voor iedere applicatie een eigen container gecreëerd. Zo'n container functioneert volledig zelfstandig, en heeft tegelijk de mogelijkheid om te communiceren met andere containers. Dit biedt een aantal voordelen:

- Problemen rondom co-existentie zijn volledig opgelost: iedere applicatie draait in een eigen omgeving zonder daarbij andere applicaties te kunnen verstoren.
- Het inrichten van verschillende omgevingen voor ontwikkeling, testen en productie wordt veel eenvoudiger. Een container kan zelfstandig getest worden. Omhangen naar een andere omgeving vereist geen enkele aanpassing van de container met daarin de applicatie, waardoor risico's bij oplevering veel kleiner worden. Bovendien vereist een rollback niets anders dan het terugzetten van de oude container.

Continue integratie

Agile manieren van werken zorgen voor het kort-cyclisch opleveren van software. Hierdoor wordt voor de business veel sneller functionaliteit met een toegevoegde waarde opgeleverd. Om dit goed te laten lopen moet de oplevering van de functionaliteit wel gebeuren op zo'n manier dat de risico's en benodigde inspanning beperkt zijn; als iedere oplevering grote impact heeft op de productieomgeving, zijn de kosten en risico's van dat kort-cyclisch werken zo groot dat de business case van agile werken onder druk komt te staan.



Het gebruik van containers zorgt ervoor dat opleveringen snel en met een beperkt risico uitgevoerd kunnen worden. Hiermee vormen ze een sterke basis voor continue integratie.

Standaardisatie voor de gebruiker



De digitale transformatie heeft gezorgd voor standaardisatie van gegevensuitwisseling via open standaarden. Die open standaarden zorgden er bovendien voor dat gewerkt kon worden met standaard bouwblokken. Mede hierdoor, maar ook door het gebruik van de browser als standaardgebruikers-interface voor apps, ontstaat de mogelijkheid om de manier waarop gebruikers met oplossingen werken te standaardiseren.

De standaardisatie van de gebruikersinterface draagt er toe bij dat IT laagdrempeliger wordt; door het leren van een app leert een gebruiker meteen de gebruiksaspecten van een groot scala aan apps. Alleen de specifieke doelen van een nieuwe app hoeven dan nog geleerd te

worden, aangezien de bediening over applicaties heen hetzelfde kan blijven. Dit verkleint inleertijd en versnelt de implementatie in een organisatie van nieuwe applicaties.

En dan nog iets

Ik zit dit artikel te schrijven in de achtertuin op een warme dag, op mijn laptop die verbonden is met mijn wifi. Zodra ik opsla, wordt het artikel direct in de cloud bewaard. Ook als we niet op kantoor werken, is de manier waarop we werken nauwelijks vergelijkbaar met die van vroeger.



Roberto Lambooy & Onno Haldar, ZaakgerichtWerken.Nu