10 succesfactoren voor continu verbeteren proces-IT match

Hoe creëer je een symbiose tussen systeem, proces en gebruiker in het cyclisch verbeteren van je proces-IT match? Oftewel, hoe richt je een structuur in die continu resulteert in vernieuwing en verbetering van de IT-ondersteuning van je processen, én in verbeterenergie bij je gebruikers? Hierbij de do's and don'ts bij het continu verbeteren van jouw proces-IT match.

Wat gaat er op dit moment mis?

Organisaties zijn in allerlei vormen bezig met het steeds verder ontwikkelen van hun processen en ondersteunende IT. Het feit dat proces en IT goed op elkaar moeten worden afgestemd is niet nieuw. Toch gaat het nog vaak mis in termen van continuïteit en kwaliteit. Aan goede initiatieven geen gebrek, maar potentiële verbeteringen blijven regelmatig hangen in een complex en diffuus proces tussen idee en realisatie. Het lukt veel organisaties nog steeds niet om structureel en iteratief verbeteringen te realiseren. Hoe komt dit? Welke fouten blijven we maken en hoe maken we de doorontwikkeling van business en IT tot een succes? Op deze vragen gaat dit artikel in.

Voorheen werd IT-ondersteuning zoals 'workflow management systemen' enkel ingezet om de gebruiker te ondersteunen bij zijn/haar werkzaamheden in het proces (input-driven). Tegenwoordig wordt het ook steeds meer ingezet om procesdata te genereren (output-driven). Dit benadrukt het belang van een goede proces-IT match des te meer. Daarnaast scherpt het de discussie aan over de balans tussen het 'dichtbouwen' van het systeem met checks (data-perspectief), versus het open en flexibel houden van het systeem t.b.v. het optimaal ondersteunen van het proces (gebruikersperspectief).
Deze balans is in veel organisaties een enorme uitdaging, met ontevreden gebruikers, overspannen functioneel beheerders en een radeloze informatiemanager als gevolg. De gezamenlijke stip op de horizon is vaak duidelijk, maar het ontbreekt aan de juiste afspraken, spelregels en randvoorwaarden om het gedeelde beeld om te zetten in een werkende applicatie die ondersteunt, en niet vertraagt. Organisaties moeten voorkomen dat ze in een vicieuze cirkel blijven hangen waarbij de gebruikers 'meldings-moe' en de functioneel beheerders 'uitleg-moe' zijn geworden (zie figuur 1). Het wederzijdse vertrouwen dat beide partijen continu werken aan het doorontwikkelen van de match tussen proces en IT is essentieel.

Figuur 1: 'Meldingsmoe' versus 'Uitlegmoe'

Het effect van een suboptimale doorontwikkeling

Het is geen eenvoudige opgave om als proces-IT matcher hoge ogen te gooien tijdens het jaarlijkse medewerkerstevredenheidsonderzoek (MTO). Gebruikers verwachten dat het systeem-landschap te allen tijde goed werkt en hen optimaal ondersteunt. Een kleine misser wordt dan al snel fataal. Toch mag deze kritische blik geen excuus zijn om de ambitie te laten varen. Als procesarchitect/procesmanager heb je de verantwoordelijkheid om de IT-ondersteuning continu door te ontwikkelen.

Toch blijkt, zoals eerder gezegd, dat continu verbetering realiseren makkelijker is gezegd dan gedaan. Regelmatig wordt de pijn bij het management onvoldoende gevoeld, terwijl de gehele organisatie (en al haar stakeholders) de nadelen ondervindt of zelfs compleet vastloopt.

Wat kan er beter? De 10 succesfactoren

Om eerder genoemde valkuilen te vermijden zijn er 10 succesfactoren te benoemen die bijdragen aan een optimale wisselwerking tussen business en systeem en daarmee het doorontwikkelen van de proces-IT match.

1. Werk met korte verbetercycli (sprints)

Ondanks dat het gedachtegoed van Scrum/Agile en haar meerwaarde voor productontwikkeling wijdverspreid is, werken teams nog regelmatig met logge en trage verbetercycli. Het karakter van doorontwikkeling is vaak onregelmatig. Er wordt pas geacteerd wanneer er zich een fout voordoet wat resulteert in het continu 'achter de feiten aanlopen'. Ook stapelen gewenste of benodigde veranderingen zich op deze manier op, met een toename in complexiteit en doorlooptijd tot gevolg.

Door je verbeterverzoeken op te knippen in kleine stukjes 'werkbaar product' ben je in staat elke 2 tot 4 weken een set verbeteringen te ontwerpen, bouwen, testen en in productie te nemen. Dit maakt doorontwikkeling flexibel, efficiënt en behapbaar. Deze methode stelt je in staat om na elke stap de nieuwe 0-situatie te inspecteren en waar nodig de koers te wijzigen.

2. Maak je activiteiten transparant

Continu verbeteren blijft nog te vaak een feestje van de technische mensen. Voor gebruikers blijft de doorontwikkeling van 'hun systeem' een black box. De 'technische heren' lijken te bepalen welke verbeteringen wanneer worden doorgevoerd. Gebruikers worden keer op keer verrast met de zoveelste update waar ze niet om hebben gevraagd. Maak je verbetersystematiek transparant! Gebruik tools zoals *Trello* en *MS Planner* om inzichtelijk te maken welke 'changes' (wijzigingen in het IT-landschap) in de pijplijn zitten en in welk stadium van ontwikkeling ze zich bevinden. Hiermee betrek je de gebruiker in de ontwikkelcyclus en geef je de organisatie de kans om bij te sturen waar nodig. Deze transparantie voorkomt ook dat er achteraf discussie ontstaat over de inhoud en prioritering van de changes. Met het real-time delen van je 'product-backlog' maak je de hele organisatie verantwoordelijk voor de ontwikkelroute van het systeem.

3. Manage de verwachtingen

Overpromise en underdeliver blijft een veel gemaakte fout. Gebruikers wordt een worst voorgehouden, terwijl de oplossing nog alles behalve in zicht is. Maak als ontwikkelteam daarom eerst inzichtelijk welke capaciteit er beschikbaar is en gebruik deze gegevens vervolgens om een realistische planning neer te leggen richting de organisatie. Bied ook kaders waarbinnen gebruikers verbeteringen kunnen initiëren. Er is niets zo frustrerend als te horen krijgen dat je zorgvuldig uitgedachte idee niet kan worden doorgevoerd vanwege technische onmogelijkheden. Daarnaast voorkom je hiermee een hoop analyse-tijd die achteraf gezien niet nodig was geweest. Dit wil overigens niet zeggen dat het systeem leidend is in het continu verbeteren van je proces-IT match. Het proces blijft leidend, omkaderd door de spelregels van het systeem.

4. Positioneer de juiste 'middle man'

Dat business en IT regelmatig slecht met elkaar communiceren is geen geheim. Dat daarom een 'olieman' of 'middle man' moet worden ingeschakeld om beide werelden samen te brengen is ook bekend. Echter, het karakter en de methode van deze 'middle man' verandert door de jaren heen. Waar voorheen de business IT adviseur/analist actief deelnam aan het ontwikkelproces, vraagt de huidige omgeving om iemand met een meer coachende rol die het ontwikkelteam in staat stelt te floreren. Dit wordt ook wel 'dienend leiderschap' genoemd, waarbij de 'leider' het ontwikkelteam ondersteunt om het doel te bereiken.

5. Creëer wederzijds begrip van een 'gereed product'

Derveaux & Sadeleer beschreven het al in 1993: bedoeld is nog niet gezegd, gezegd is nog niet gehoord, gehoord is nog niet begrepen, begrepen is nog niet akkoord en akkoord is nog niet gedaan. Deze 'communicatie-muur' is in zekere zin ook van toepassing op veel productontwikkelingstrajecten. Er bestaan in een organisatie vaak verschillende definities van een gereed product. Dit creëert ambiguïteit en minimaliseert daarmee de kans op succes. Bepaal en documenteer daarom vooraf expliciet waar een opgeleverde change aan moet voldoen.

6. Documenteer je systeemkeuzes en vermijdt 'technical debt'

Volledige documentatie over zowel het functionele als het technische aspect van een change ontbreekt vaak. Programmeurs kennen de technische achtergrond en gebruikers weten, met een beetje mazzel, nog de functionele argumenten waarom changes in het verleden zijn doorgevoerd. Dit maakt dat (onderdelen van) een change regelmatig overgedaan moeten worden, omdat men niet meer kan achterhalen hoe en waarom iets is gebouwd. Dit raakt de discussie over korte termijn denken versus lange termijn denken. Programmeurs worden vaak op projectbasis ingevlogen en staan onder flinke tijdsdruk. Hun doel is om de klant op dat moment tevreden te maken met de op dat moment gevraagde wijziging. Op lange termijn is het echter van belang dat de gevraagde change ook toekomstbestendig is gebouwd. Er mag geen 'technical debt' achter blijven. Dit noemt men ook wel de 'boyscout-rule'. Laat de programmeercode beter achter dan hoe je het aantrof. Hiermee wordt er verwezen naar scouting-leden die de regel hanteren dat ze de camping netter achterlaten dan hoe ze het aantroffen.

7. Creëer ambassadeurs

Om changes succesvol te implementeren zijn ambassadeurs van wezenlijk belang. Zonder goede ambassadeurs blijft de doorontwikkeling 'een feestje van techneuten of externen' en bereikt het niet of nauwelijks de werkvloer. Een goede ambassadeur staat met zijn/haar poten in de klei, maar is ook key-user voor de applicatie. Deze persoon kent dus zowel de applicatie als het proces door en door en is in staat om wensen van de business te vertalen naar functionele eisen aan het systeem. Daarnaast heeft de ambassadeur binnen de organisatie een betrouwbare en toegankelijke status waardoor hij/zij makkelijk mensen meekrijgt in de benodigde verandering.
Ook al voldoet een medewerker aan alle bovenstaande criteria, een ambassadeur is pas echt een ambassadeur als hij/zij zelf oprecht gelooft in de meerwaarde van de change. Richt je aandacht daarom eerst op de ambassadeur, alvorens je de rest van de medewerkers kennis laat maken met de wijziging. Komt de wijziging niet langs de beoogd ambassadeur, dan weet je dat je er nog werk aan de winkel is voordat de rest van de organisatie kan worden betrokken.

8. Kies bewust het soort softwarepakket

Een bekend dilemma dat menig informatiemanager bezig houdt is de keuze voor het soort softwarepakket. Kies je voor een standaardpakket dat garanties geeft op een goede werking van bepaalde functionaliteiten en een bewezen statuur heeft bij andere organisaties? Of kies je voor de flexibiliteit van een maatwerkoplossing? Wat je ook kiest, houd rekening met de volgende kenmerken bij de inrichting van een continu verbeterstructuur:

Standaardpakket: Bewezen techniek; profiteren van doorontwikkeling gevoed door andere gebruikers; lagere ontwikkel- en implementatiekosten; eenvoudige data-uitwisseling met andere organisaties.

Maatwerkpakket: Veel flexibiliteit in ontwikkeling; niet afhankelijk van wensen van andere gebruikers; beter toegespitst op specifieke werkprocessen; snellere doorontwikkeling.

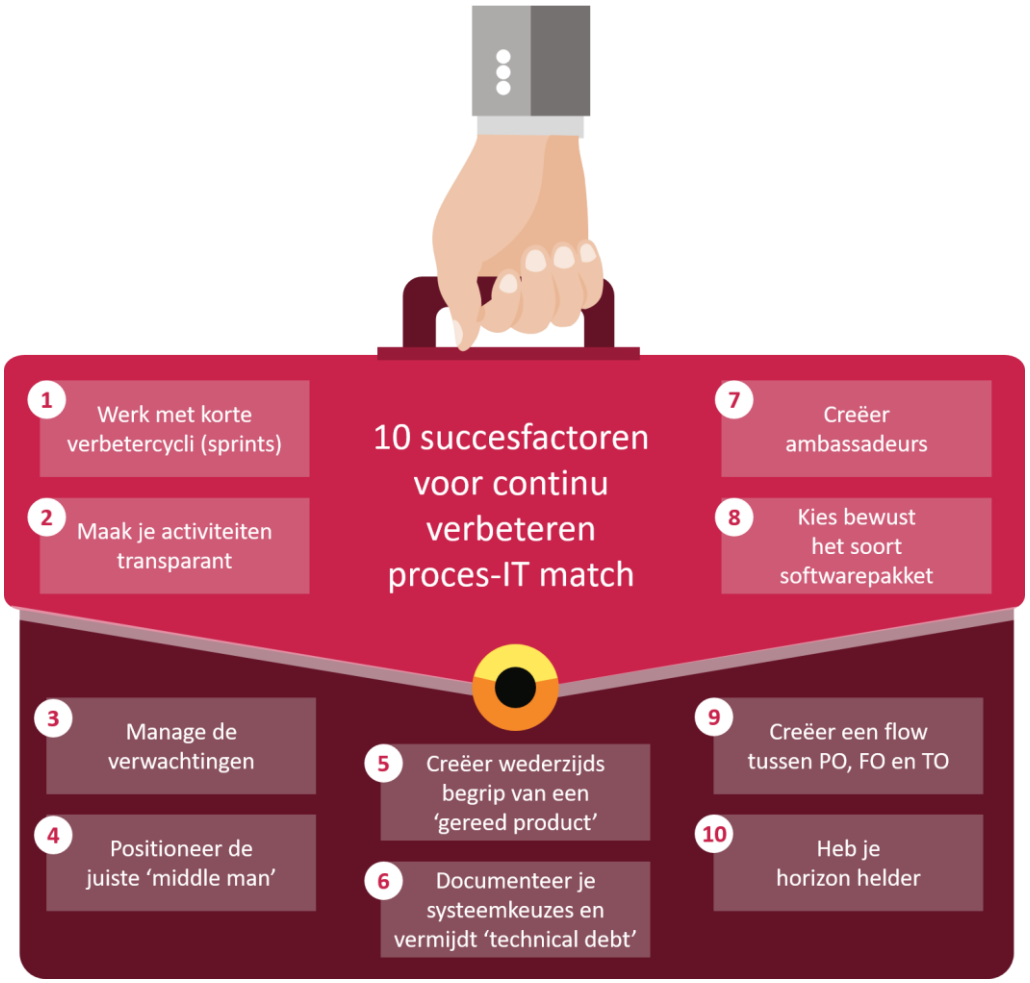
9. Creëer een flow tussen PO, FO en TO

Een procesontwerp (PO), functioneel ontwerp (FO) en technisch ontwerp (TO) zijn alle drie belangrijke elementen bij het succesvol afronden van een change. Wanneer één van deze ontwerpen ontbreekt of onvoldoende aansluit

op de andere ontwerpen is dit een garantie voor falen. De 3 ontwerpen zijn essentieel in de samenwerking tussen business en de IT.
Het gebruik van de zogenoemde PO's, FO's en TO's kun je vergelijken met het wilde westen. Er bestaan geen regels en er is dus ook geen standaard. Vorm en detailniveau van de ontwerpen spelen echter een grote rol in het onderling begrip tussen procesanalist en programmeur. Zorg ervoor dat je dezelfde taal spreekt. Iedere functie heeft een eigen rol, met eigen werkwoorden, maar de uitdaging ligt in het creëren van een gedeeld beeld. Het is als een voetbalteam dat 90 minuten lang een nieuwe aanval opzet, waarbij keeper, verdediger, middenvelder en aanvaller volgens afgesproken patronen met elkaar samenwerken om tot een optimaal resultaat te komen.

10. Heb je horizon helder

De wereld staat niet stil. Het is een dooddoener, maar veranderingen volgen elkaar steeds sneller op. Des te meer reden om de ontwikkeling van je IT-ondersteuning continu te spiegelen aan 'de stip op de horizon' van je processen. Elke ontwikkeling moet in lijn liggen c.q. bijdragen aan de ontwikkelvisie van de proces-IT match. Kijk kritisch naar elke doorontwikkeling en bepaal op basis van een vaste set criteria of de wijziging meerwaarde en geen 'sunk costs' oplevert. Dit betekent overigens niet dat elke investering bij moet dragen aan de lange termijnvisie van het applicatielandschap. Bij tijd en wijlen is een korte termijn fix nodig om systemen 'functioneel te kunnen laten draaien'.



Figuur 2: 10 succesfactoren voor continu verbeteren proces-IT match

Conclusie: van droom naar realisatie

Met de toenemende waarde (zowel vanuit input- als outputperspectief) van een optimale proces-IT match neemt de noodzaak tot een strakke continu verbeter cyclus steeds verder toe. Het creëren van de juiste randvoorwaarden

en het vastleggen van spelregels is daarbij van essentieel belang. Kort-cyclische verbeterslagen waarbij participatie, transparantie en commitment vanuit zowel de business als de IT zijn geborgd, zijn bepalend voor het succes van de doorontwikkeling. Creëer verbeterenergie door verspilling in het ontwikkelproces te elimineren en successen te vieren. Ga aan de slag met deze 10 succesfactoren en maak continu verbeteren leuk, simpel en succesvol!

Over de auteur:

Jentel Mulder is organisatieadviseur bij BPM Consult. Hij specialiseert zich in procesmanagement, en dan met name het adviseren van publieke organisaties over hoe zij hun proces-IT match kunnen optimaliseren, teneinde hun prestaties continu te blijven verbeteren.