

# Procesbesturing sociale verzekering

## *Vernieuwen of vervangen?*

*"Wij worstelen als organisatie nu al zo lang met onze procesbesturing. Eén van de problemen is het houtje-touwtje systeem dat we daarvoor gebruiken en dat ieder moment om schijnt te kunnen vallen. Nu, dan zijn we pas echt ver van huis. Wat wij nodig hebben is een modern, uniform systeem dat het management 'met een druk op de knop' inzicht geeft in de stand van zaken. Kortom, we moeten ons huidige systeem vervangen door een modern workflow management systeem."*

Een instantie in de sociale zekerheid heeft een grote focus op operationele efficiëntie. De politiek oefent druk uit om sneller, tegen lagere kosten uitkeringen te verstrekken. Dat betekent dat de procesbesturing moet verbeteren. Weten we waar het goed gaat en waar niet? Weten we waar het beter zou kunnen? Waarom weten we dat niet? En, als we het al op afdelingsniveau weten, waarom weten we het dan niet op directieniveau?

Deze vragen zijn niet nieuw voor de organisatie. Sterker nog, men worstelt er al jaren mee. De processen zijn nauwkeurig beschreven en door de gehele organisatie geuniformeerd, er zijn allemaal prestatie-indicatoren benoemd waar ook trouw over gerapporteerd wordt – en toch heeft het management niet het gevoel de operatie in de greep te hebben. Al jaren is men voornemens een goede oplossing te implementeren, een systeem dat de operatie ondersteunt maar tegelijk ook management-informatie op verschillend detailniveau kan genereren. Maar altijd is er wel een reden om het niet te doen: een uitbreiding die moet worden verwerkt, een plotselinge bezuinigingsmaatregel, onduidelijkheid over de toekomst.... Uiteindelijk leidt het tot de situatie dat steeds weer een korte-termijn oplossing wordt gekozen die de meest urgente problemen weliswaar oplost, maar de organisatie niet dichterbij een structurele oplossing brengt.

### **Valkuil 1: korte termijn denken**

Telkens kiezen voor korte-termijn oplossingen die weliswaar de onmiddellijke problemen oplossen (en daarmee de business case voor een andere oplossing wegnemen), maar niet bijdragen aan een structurele oplossing en daardoor de organisatie uiteindelijk niet helpen.

Op een gegeven moment is de tijd toch rijp om de problemen structureel aan te pakken. De korte-termijn oplossing, die technologisch verouderd is en een risico betekent voor het netwerk van de betrokken organisatie, zal worden vervangen door een state-of-the-art workflow pakket dat nog een hoop extra functionaliteiten kan bieden ten opzichte van de huidige houtje-touwtje oplossing. En perfect passend in de architectuur van de organisatie, waar dit workflow pakket een uniforme procesbesturingslaag moet gaan vormen over alle primaire systemen heen.

Vol enthousiasme gaat een projectgroep aan de slag om één en ander te realiseren. Geheel volgens de normen van de organisatie wordt een strak projectplan gemaakt, worden milestones en deliverables gepland en de benodigde kosten geschat. Iedere maand wordt een voortgangsrapportage gemaakt die, samen met de voortgangsrapportages van de andere projecten, het management in één oogopslag de stand van zaken van de gehele projectportefeuille geeft.

Maar het loopt toch weer anders. Een nieuwe bezuinigingsronde zorgt ervoor dat alleen de meest urgente projecten overleefd blijven – en daar hoort dit project niet bij. Het

project wordt afgeserveerd: niet urgent, niet cruciaal. De korte termijn wint het weer eens van de lange termijn; de organisatie is de verliezer.

### **Valkuil 2: groot denken**

Organisaties die moeite hebben met veranderen zetten vaak grootschalige veranderprogramma's op om toch de nodige aanpassingen door te voeren. Het heeft echter vaak tot gevolg dat er onvoldoende ruimte is voor kleine verbeteracties, voor het type verbeteringen dat niet indrukwekkend klinkt (en niet een plekje binnen het grote veranderprogramma weet te veroveren), maar die de mensen op de werkvloer wel concreet helpt in hun dagelijkse werkzaamheden. Het gevolg is dat het autonome verandervermogen van de organisatie nog verder af neemt omdat er een mentaliteit ontstaat van 'we zien wel wat er gaat gebeuren'.

Of toch niet? Is het wel zo simpel als hierboven beschreven? Want de projectgroep die het nieuwe systeem van de grond zou moeten trekken raakt al snel aan het twijfelen of dat eigenlijk wel zo nodig is. De veelbekritiseerde houtje-touwtje oplossing blijkt toch helemaal niet zo slecht te zijn. Weliswaar technologisch verouderd, maar hij loopt nog lang niet op z'n eind. En veel van de voorgestelde verbeteringen blijken ook met het bestaande systeem te kunnen worden gerealiseerd. Er blijkt al jarenlang een dynamiek in de organisatie te zijn waarbij er door sommigen negatief over het bestaande systeem wordt gedacht – negatiever dan het wellicht vanuit andere perspectieven verdient. Informatie over het systeem blijkt gekleurd, moet gezien worden in relatie tot degene die het zegt. Er zijn veel verhalen in omloop die uiteindelijk niet blijken te kloppen en die het moeilijk maken een objectief beeld (zo dat al mogelijk is) te vormen en die de besluitvorming sterk hebben beïnvloed. Technologie (het huidige systeem met z'n voor- en nadelen) is toch niet zo waardevrij als gedacht.

Maar er is nog meer aan de hand. De architectuur waar het nieuwe systeem een onderdeel van zou moeten zijn, blijkt niet goed doordacht. Het idee is dat een out-of-the-box workflow pakket als een integrale besturingslaag over bestaande legacy-systemen wordt gelegd, en als een soort 'schil' om deze systemen ligt. Voor de eindgebruiker wordt het legacy-systeem in feite onzichtbaar: hij werkt in het workflow systeem, dat ook de informatie voor management-informatie levert. Voor het aanvraag-proces, dat met het 'houtje-touwtje systeem' wordt ondersteund, is dit aspect niet van belang: de legacy-systemen gaan hier pas een rol spelen aan het eind van dit proces. Voor de vervolgpacten is het echter wel van belang: de legacy-systemen spelen hier een cruciale rol.

En dan blijkt de architectuur-visie niet haalbaar. In de legacy-systemen zijn allerlei 'triggers' gedefinieerd die acties uitzetten bij de gebruikers; als je de besturing los zou willen koppelen van deze legacy-systemen dan zou je hetzij deze triggers in het workflow-pakket moeten nabouwen, hetzij een intensieve interface moeten bouwen. Maar de toegevoegde waarde daarvan is gering, omdat het kortcyclische activiteiten zijn die snel uitgevoerd kunnen worden. De procesbesturing waar de organisatie zo'n behoefte aan heeft betreft meer het overzicht van de openstaande acties en de rapportage achteraf over hoe ze zijn uitgevoerd, dan inzicht in de voortgang van de individuele acties. En dan heeft workflow niet zo veel toegevoegde waarde. Een architectuur die er op papier prima uitziet blijkt in de praktijk toch niet te werken.

Het perspectief op de toekomstige architectuur is echter de afgelopen jaren wel regelmatig de reden geweest om allerlei kleinschalige verbeteracties uit te stellen. Diverse initiatieven om de rapportage uit de legacy-systemen te verbeteren zijn tegengehouden onder het motto: straks hebben we een integraal workflow-pakket waar we ook allerlei rapportage-faciliteiten aan kunnen koppelen, dus we gaan nu niet investeren in dit soort kleine verbeteringen als straks al onze problemen worden opgelost.

**Valkuil 3: perfectie is de vijand van het goede**

Oftewel: korte-termijn verbeteringen worden vaak niet doorgevoerd omdat ze niet in het lange-termijn plaatje zouden passen

De frustratie op de werkvloer is dan ook aanzienlijk als duidelijk wordt dat het project (dat sowieso niet door gaat) geen oplossing zou kunnen bieden voor de problemen die zij als het meest urgent beschouwen. Is er dan nooit eens ruimte om gewoon pragmatisch kleine verbeteringen door te voeren?

Deze frustratie kan echter ook momentum genereren om de zaken nu dan wel in beweging te krijgen. Als de lange-termijn oplossing verder weg is dan ooit, als de van bovenaf opgelegde prioriteiten de werkvloer niet gaan helpen in hun dagelijkse werkzaamheden, dan zijn er wellicht mogelijkheden om (al dan niet formeel geaccordeerd) kleine initiatieven te ontwikkelen die 'onder de radar' van de grote programma's blijven. Het management hoeft alleen wat ruimte te bieden, wellicht hier en daar wat zaken te masseren. Veranderen is bewegen, en uiteindelijk moet die beweging op de werkvloer plaats vinden. Waarom zou je de beweging daar dan ook niet laten beginnen?

**Vernieuwen of vervangen?**

Het voorbeeld van de sociale verzekeringsinstelling laat in eerste instantie een heel algemeen probleem zien dat in vele organisaties speelt: moet een bestaand systeem worden vernieuwd, of moet het worden vervangen door een geheel ander systeem? Het bijzondere in dit geval was dat deze vraag in feite helemaal niet gesteld werd: de vervanging van het huidige systeem was het vertrekpunt. Immers, het bestaande systeem zou volledig verouderd zijn, en het was niet in staat de benodigde management-informatie op directieniveau op te leveren. En het paste niet in het architectuurmodel van de organisatie. De *business case* voor vervanging leek glashelder.

En toch bleek hij dat uiteindelijk niet te zijn. De argumenten vóór vervanging bleken gebaseerd te zijn op een onjuiste beoordeling van het bestaande systeem. En de overall architectuur, waar het nieuwe systeem zo goed in paste, bleek niet goed doordacht te zijn. Het vertrekpunt was dus niet goed gekozen – maar dat wordt vaak niet goed herkend en onderkend, waardoor ICT-projecten soms problemen oplossen die ook op geheel andere wijze, met veel minder inspanning en minder impact op de organisatie, opgelost zouden kunnen worden.

De technologische ontwikkeling verdient separate aandacht. Enerzijds is de voortdurende ontwikkeling op het gebied van ICT, en in het bijzonder van het type applicaties waar de sociale verzekeringsinstelling gebruik van maakt, een belangrijke impuls voor vernieuwing. Er kan gewoon steeds meer, en wie vast blijft houden aan verouderde technologie kan minder inspelen op deze nieuwe mogelijkheden. Daarom wordt wel eens gesteld dat een systeem slechts een levensduur van zo'n vijf jaar heeft en dan vervangen moet worden.

**Workflow**

Workflow management systemen zijn systemen waar een individueel geval, een 'case', een van te voren gedefinieerd proces doorloopt. Afhankelijk van het gekozen proces (en daarbij zijn allerlei mogelijkheden, inclusief vertakkingen in de procesgang) gaat een case van de ene medewerker of afdeling naar de volgende; de medewerker hoeft alleen zijn eigen taken uit te voeren, en het systeem zorgt ervoor dat de case vervolgens in het juiste werkbakje terecht komt voor de volgende stap. Zo kan het werk zeer

gestructureerd worden uitgevoerd, en heeft het management op ieder moment inzicht in de stand van zaken.

Workflow management systemen waren vooral in het begin van de jaren negentig een grote hype, waarbij de verwachting was dat steeds meer organisaties steeds meer processen met behulp van workflow zouden willen ondersteunen. De ervaringen van gebruikers waren echter niet onverdeeld positief. Workflow staat geen spontane variaties in de procesgang toe: wat niet van tevoren is gedefinieerd is niet mogelijk. Dat maakte het voor veel processen te rigide; gebruikers voelden zich in een keurslijf gedwongen en hun eigen professionaliteit miskend ("ik weet echt zelf wel wat er moet gebeuren, daar heb ik geen systeem voor nodig"). Workflow leek dan ook in de tweede helft van de jaren negentig een stille dood te sterven.

De laatste jaren is er echter sprake van een wederopstanding. Nieuwe workflow-pakketten zijn flexibeler, laten zich makkelijker integreren met andere applicaties en kunnen heel snel worden ingericht. Het fundamentele punt dat ieder proces van tevoren gedefinieerd moet zijn, is echter gebleven.

Maar de technologische ontwikkeling heeft nog een ander aspect, namelijk dat van de toenemende mogelijkheden om systemen met elkaar te laten communiceren ook als ze daar oorspronkelijk niet voor gebouwd waren. Juist op dit gebied is de vooruitgang de laatste tien jaar enorm geweest, waardoor 'verouderde' systemen een tweede jeugd kunnen krijgen.

Tenslotte speelt ook een rol dat er in iedere organisatie een ingebakken (en niet per sé onterechte) weerstand is tegen het vervangen van bestaande systemen, vooral als deze de behoeftes van de gebruikers redelijk vervullen. Systemen belichamen vaak de kennis en expertise van organisatie, vooral wanneer het de systemen zijn die de primaire processen ondersteunen; vervanging ervan is een majeure operatie die veel overhoop haalt en een gerede kans op mislukking heeft. Daarom kiezen veel organisaties er toch voor om hun 'legacy systemen' in stand te houden en ze liever aan te passen aan nieuwe omstandigheden dan ze te vervangen door iets geheel nieuws. Dit bleek ook heel duidelijk bij de milleniumwisseling, toen in veel systemen, diep onder de oppervlakte, code aanwezig bleek van vele decennia oud.

### **Legacy systemen**

Legacy systemen zijn oudere systemen die in een organisatie in gebruik zijn maar die niet voldoen aan 'moderne' inzichten over hoe applicaties opgebouwd moeten zijn. Vaak zijn het maatwerk-oplossingen, gebouwd in de zeventiger en tachtiger jaren op basis van technologieën die al lang verouderd zijn. Iedere organisatie worstelt met het probleem: moet ik het oude systeem vervangen, of moet ik er mee verder, eventueel na een vernieuwing (een 'mid-life revival')?

Legacy systemen hebben meestal betrekking op de primaire processen van een organisatie, en die zijn vaak redelijk uniek. De systemen zijn in de loop van de jaren doorontwikkeld en er zit een enorme hoeveelheid know-how van de organisatie in verwerkt; het is niet eenvoudig om (b.v. op basis van een standaard-pakket) dezelfde functionaliteit na te bouwen. Daarom zijn deze systemen vaak een lang leven beschoren.

Bij de keuze tussen vervangen of vernieuwen speelt een rol of het legacy systeem, ondanks de verouderde technologie, toch een moderne architectuur heeft (of kan krijgen). ICT-architecten zijn dol op het onderscheiden van verschillende 'lagen' in een applicatie, waarbij bijvoorbeeld onderscheid wordt gemaakt tussen een gegevenslaag, een 'business logica' laag en een presentatielaag. Legacy systemen die zo zijn opgebouwd of die zo kunnen worden aangepast dat deze lagen onderscheiden kunnen worden, kunnen goed geïntegreerd worden met andere, nieuwere systemen (waarbij het zelfs zo kan zijn dat ze als zodanig nooit worden vervangen, maar dat hun rol steeds verder wordt teruggedrongen). Als het legacy-systeem echter een kluwen spaghetti is

waarbij de hele hoop meekomt als je aan één sliertje trekt, dan is het onvermijdelijk het op enig moment te vervangen.

## Groeien of ontwerpen?

Een tweede 'richtingenstrijd' die het voorbeeld van de sociale verzekeringsinstelling laat zien, is die tussen de *ontwerpers*, degenen die willen veranderen door een blauwdruk van de toekomstige situatie te maken en daar gericht naartoe te bewegen, en de *groeiers*, die verandering het liefst van onderaf laten komen door ruimte te geven aan nieuwe initiatieven om daar vervolgens zo veel mogelijk van te leren. Het bestaande systeem was uit nood geboren: omdat de centrale systemen onvoldoende functionaliteiten voor procesbesturing boden, werd er lokaal een systeem gebouwd om deze lacune te vullen. En aangezien er weinig afstand was tussen de ontwikkelaars en de gebruikers maakte dit systeem vervolgens een grote ontwikkeling door waarbij gebruikerswensen snel gerealiseerd konden worden.

Uit oogpunt van verandermanagement is dit een krachtig model. Het genereren van beweging 'aan de basis' geldt als één van de noodzakelijke voorwaarden waaraan voldaan moet zijn om veranderingen te kunnen laten slagen; het ruimte geven voor initiatieven van onderop is iets dat bijvoorbeeld door de Japanners in hun kwaliteitsdenken (met de nadruk op kwaliteitskringen) enorm wordt gestimuleerd.

ICTers staan echter in het algemeen vrij huiverig tegenover zulk een 'bottom-up' model. Om een strakke applicatie te krijgen met lage beheerkosten is het nodig om vanuit een heldere architectuur te ontwikkelen en de flexibiliteit te beperken; bottom-up modellen hebben dat meestal niet. Wat dat betreft was de situatie bij de sociale verzekeringsinstelling vrij uniek: het systeem dat lokaal was ontwikkeld bleek geschikt voor een organisatiebrede uitrol. Meestal worden lokaal (Excel-)modellen gebouwd die, naarmate ze groeien, steeds ingewikkelder en moeilijker te beheren worden tot ze worden vervangen door een strak, centraal ontwikkeld systeem. Voor zulk een ontwikkelmodel is overigens veel te zeggen: door het werken met een lokale applicatie, hoe beperkt ook, worden de functionele behoeftes steeds duidelijker en kan helder worden gedefinieerd wat het centraal te ontwikkelen systeem moet kunnen.

## De objectieve en enige waarheid?

Ook ICT is mensenwerk. Mensen maken systemen, mensen identificeren zich met systemen en verbinden hun lot eraan. Tegelijk zijn er allerlei andere invloeden die juist weer tegenkrachten oproepen, zoals technologische ontwikkelingen die nieuwe mogelijkheden bieden, en bedrijven die deze nieuwe mogelijkheden onder de aandacht brengen. De discussie over het vernieuwen of vervangen van bestaande systemen wordt weliswaar gevoerd met zakelijke, waarde vrije argumenten – maar daarachter ligt vaak een wereld van emoties en belangen. Vanuit een verschillend perspectief ziet men heel verschillende dingen als men naar hetzelfde kijkt. Het blijkt in zulke omstandigheden moeilijk te zijn om een gedeeld beeld te krijgen, of zoals Morgan's metaforen ons leren om je bewust te zijn van die verschillende beelden en langs die weg met elkaar in gesprek te komen.

Bij de sociale verzekeringsinstelling was het bestaande systeem decentraal, in één van de districten ontwikkeld buiten de concern ICT-afdeling om. Deze had reeds vaak aangegeven dat dit systeem volgens hen niet toekomstvast was en dat het verstandig was om een modern systeem te bouwen en dat vervolgens landelijk uit te rollen. Om allerlei praktische en korte-termijn redenen was dat echter nooit gebeurd – dus toen de nood aan de man kwam en er urgent een besturingssysteem landelijk uitgerold moest worden, was het lokale systeem het enige dat beschikbaar was. Vanuit de visie van de ICT-afdeling werd een probleem dat klein begon zo steeds groter. En ideaal was het ook

niet, daar hadden ze helemaal gelijk in. Maar met wat goede wil zou er wel mee te leven zijn – doch die goede wil ontbrak goeddeels. De organisatorisch context beïnvloedde de visie op de technologie.

## **Voortschrijdend inzicht**

Wat het voorbeeld van de sociale verzekeringsinstelling ook laat zien (en ook diverse andere casussen in dit boek), is het inzicht dat het van vitaal belang is om voortschrijdend inzicht te vergaren en ruimte te bieden om het project hierdoor te laten beïnvloeden. De 'standaard' methodieken van projectmanagement benadrukken vrijwel allemaal een gedetailleerd plan van aanpak aan het begin van het project, en een strikte indeling in verschillende fases met duidelijke beslismomenten. Het terugkomen op eerdere besluiten of het kiezen van andere richtingen door nieuwe inzichten is hier vaak onmogelijk, of in ieder geval heel moeilijk. En het is ook erg riskant om laat in projecten nieuwe koersen te varen. Maar het niet open staan voor nieuwe inzichten is ook riskant en levert vaak resultaten op in de categorie 'operatie geslaagd, patient overleden'. Het is de uitdaging om hier tussen te laveren, om een heldere koers te varen, maar ook om de moed te hebben hier later toch van af te wijken als de overtuiging groeit dat dat beter is. Improviseren op het snijvlak van organisatie en ICT, kortom.

## **Conclusie**

Het voorbeeld van de sociale verzekeringsinstelling laat zien hoe moeilijk het kan zijn om een gedeeld, laat staan objectief beeld te krijgen van een situatie, en hoe belangrijk het is om open te staan voor nieuwe inzichten. Aan het begin van het project leek alles duidelijk, en was er weinig twijfel over de te varen koers. Maar later bleek dat die helderheid schijn was, dat de organisatorische context de meningen beïnvloedde. Technologie was toch niet zo eenduidig, zo waarde vrij als gedacht.