

**BraintainEr**

# “Ergens zijn wij wat gebruikers kwijtgeraakt”

Ir. E.J.K. Rensen

Lezing voor de Vereniging Nederlandstalige SAP gebruikers

Themadag Projecten en Onderhoud  
Ede, 5 oktober 2011

## Inhoudsopgave

Verantwoording.....	2
Inleiding: Kennedy en de Appolo 11.....	3
ERP-projecten falen nog wel eens.....	4
Lemmingengedrag in ERP-projecten .....	5
Ontkennen en negeren van complexiteit.....	7
Data en informatie zijn niet hetzelfde.....	8
De aandacht voor de gebruiker is een mythe .....	9
Streven naar eenvoud voor de gebruiker betekent keuzes maken .....	13
Projecten, coördinatie, en managen van verwachtingen .....	15
Er is een verschil tussen risico's en onzekerheden .....	17
Tips om de gebruikers niet kwijt te raken .....	17
Tip 1:    Formuleer het doel .....	18
Tip 2:    Maak verwachtingen expliciet: wederzijdse aanpassing.....	18
Tip 3:    Organiseer het project.....	18
Tip 4:    Formaliseer het ontwerp en benoem beheerders .....	20
Tip 5:    Organiseer het ERP-project! .....	20
Tip 6:    Zorg voor betrouwbaarheid .....	21
Tot besluit.....	23

## Verantwoording

Deze tekst is gebaseerd op een lezing die is gehouden op 5 oktober 2011 tijdens de themadag "Projecten en Onderhoud" van de Vereniging Nederlandstalige SAP Gebruikers.

De inhoud van deze tekst komt volledig voor de verantwoordelijkheid van de auteur. Omdat het een weergave van een lezing betreft is er hier en daar sprake van opinies en stellingen die vooral tot doel hadden de toehoorders te prikkelen, uit te dagen, en hen aan te zetten tot reflectie. Bovendien is de gekozen invalshoek die van Onderhoud. Dat betekent dat sommige uitspraken geldig zijn in de onderhoudsomgeving, maar mogelijk niet in een ander functioneel gebied, zoals Productie.

Bij het meeste van wat gezegd en nu dan ook geschreven is, kunnen uiteraard kanttekeningen worden gemaakt. Een lezing aan het eind van een uitgebreid programma is namelijk niet zo geschikt om de gebruikersaspecten van ERP-implementaties diepgravend te verkennen. Het is aan de lezer te ontdekken welke nuances zouden kunnen worden aangebracht. Wie die moeite wil doen, mag echter één ding niet uit het oog verliezen:

*ERP-systemen zijn ontwikkeld voor de échte gebruikers, de mensen die er dagelijks hun werk mee moeten doen en die we nog wel eens uit het oog verliezen.*

Eric Rensen

Nuenen, 5 oktober 2011

Voor verdere informatie verwijzen wij graag naar onze website: [www.braintainer.nl](http://www.braintainer.nl)

## Inleiding: Kennedy en de Appolo 11

Het is 25 mei 1961. President Kennedy houdt een speech voor het voltallige Congres van afgevaardigden, en kondigt aan dat vóór het einde van het decennium (het werd 1969) een mens op de maan zal landen om daarna weer veilig terug te keren naar de aarde. Geen enkel ruimtevaartproject zal zo moeilijk en kostbaar zijn om te verwezenlijken.



Neem nu eens de organisatie in gedachten die met deze missie werd belast. U zult misschien denken, met uw kennis van ERP-implementaties en Prince2, dat het doel bereikt is door middel van een strak opgetuigd project, met bijbehorende projectorganisatie, projectmethoden, en strikt management.

Niets was minder waar. De missie vroeg om een bijna onvoorstelbare en veelomvattende opsplitsing van activiteiten, met duizenden specialisten die allerlei specifieke taken vervulden. Maar uiteindelijk was er initieel niemand die exact wist wat er gedaan moet worden. Die kennis ontwikkelde zich tijdens het werk.<sup>1</sup>

Elk voorbeeld, en elke vergelijking gaat mank, ook in dit geval. Je mag het Appolo 11 project niet vergelijken met de implementatie van een ERP-systeem. Over de kosten werd bijvoorbeeld niet moeilijk gedaan. Maar het doel was helder en werd wél bereikt én binnen de gestelde termijn. En zelfs met oneindige resources is dat geen vanzelfsprekendheid. We komen er op terug als we het straks gaan hebben over de manier waarop het meest essentiële deel van een ERP-implementatie zou kunnen, en misschien wel zou moeten worden georganiseerd.

---

<sup>1</sup> Henry Mintzberg, "The structuring of organizations", (p3), 1979, Prentice Hall Inc, Englewood Cliffs

## ERP-projecten falen nog wel eens

ERP-projecten hebben een tamelijk slechte naam. Ze duren te lang, ze kosten meer dan voorzien, en ze

METRIC	AVERAGE
Take Longer Than Expected	57%
Implementation Duration	18.4 months
Total Implementation Cost (\$)	\$6.2M
Total Implementation Cost (% of Revenue)	6.9%
Total Implementation Cost Exceeds Budget	54%

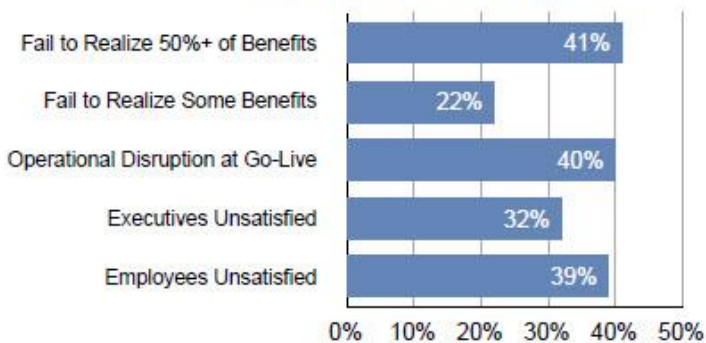
leveren minder waarde dan is voorgespiegeld. In een recent onderzoek werd dat wederom bevestigd.<sup>2</sup>

Als we Onderhoud gaan aansluiten op ERP-software, en we kijken naar de cijfers uit het onderzoek, lijken bescheidenheid en voorzichtigheid geboden.

Ik ben geen ERP-expert, en ik kan de juistheid van het onderzoek niet goed beoordelen. Maar in 30 jaar adviseren ben ik met veel leed geconfronteerd dat overeenkomt met de bevindingen. En je kunt ze nog steeds tegenkomen, die minder succesvolle implementaties.

Nu is de hamvraag natuurlijk: hoe komt het dat die implementaties in de ogen van velen niet het succes opleveren dat bij aanvang voor ogen stond?

Figure D: ERP Results (%)



Het is wat gemakkelijk om dan maar een

zondebok te zoeken. Of een zwart schaap aan te wijzen. Toch gebeurt dat nog wel. En we maken er ons allemaal bewust en onbewust schuldig aan. En het zwarte schaap hoeft geen mens te zijn, het kan ook "het bedrijf" zijn, of "het hoofdkantoor", of "de leiding", of een andere abstracte entiteit die maakt dat wij ons geen diepgaande vragen hoeven te stellen. En die daarmee voor iedereen uiteindelijk toch een ogenschijnlijk nuttige functie vervullen.



*"We failed the project but don't worry, I think I know who we can blame."*

Opdrachtgevende managers, zo is mijn ervaring, hebben vooral de neiging om de oorzaken buiten henzelf te zoeken. Gebruikers ook, en consultants zeker. Ze hebben allemaal gelijk, of juist geen gelijk. Dat is wat hen bindt. De echte vraag is natuurlijk: hoe komt het dat in zoveel gevallen een grote groep van betrokken (in beide betekenissen van het woord) mensen in staat is er een (gedeeltelijke) mislukking van te maken?

De plaatjes met de cijfers over mislukte projecten laten iets heel interessants zien: het gaat vooral over verwachtingen die niet werden waargemaakt. Blijkbaar, en hopelijk, waren de verwachtingen fout of

niet specifiek genoeg. Want het alternatief is dat er gewoon maar wat is aangerommeld, dat de projecten slecht werden geleid, of dat de begeleiding in handen was van gewetenloze, ondeskundige knoeiers. We moeten niet uitsluiten dat zo iets wel eens voorkomt, maar vooralsnog kunnen we er beter van uitgaan dat het probleem zit in het managen van verwachtingen. En die verwachtingen worden nota bene gewekt in het prille begin van het project. We plannen als het ware de mislukking al in het vroegst mogelijke stadium. En

<sup>2</sup> Blog van Michael Krigsman, februari 2010 naar aanleiding van een onderzoek door Panorama Consultants.

op de een of andere manier is een groep van individueel verstandige individuen niet in staat om onduidelijke of onjuiste verwachtingen te corrigeren. We gaan gewoon door, en we luisteren niet naar waarschuwingen. Of misschien is de bedrijfscultuur er niet naar om iets te doen met kritische opmerkingen en vragen. In zo'n cultuur leren mensen snel af iets te zeggen dat als kritiek kan worden opgevat. Typisch zo'n cultuur waarin managers graag zeggen dat ze geen problemen willen horen, maar oplossingen.

## Lemminggedrag in ERP-projecten

Een illustratief voorbeeld van een aantal jaren geleden uit mijn eigen praktijk. Ik had het voordeel dat ik er vanuit een ander project maar zijdelings bij was betrokken. Voor de goede orde: we hebben het over een, ook door mij, gerespecteerde onderneming. Er was door adviseurs een zogenoemde landschapstudie gemaakt. De conclusie: verschillende applicaties voor dezelfde dingen, software die niet meer aan de



**Come back, you fools! That's just a myth! We lemmings don't really do that!**

hedendaagse eisen voldeed, en onderhoudsorganisaties die op heel verschillende manieren werkten. Tot zover niet echt bijzonder. Wat gebeurde er vervolgens?

Er werd een project gestart om het bestaande ERP-systeem naar onderhoud uit te breiden. Een van de bestaande stand-alone onderhoudsapplicaties was als voorbereiding op wat zou gaan komen al uit de lucht gehaald. Er was door de gebruikers gewaarschuwd dat dit beter na de implementatie van het ERP-systeem kon gebeuren.

Het ERP-systeem zou ervoor gaan zorgen dat iedereen in onderhoud op dezelfde manier ging werken. De vooronderstelling van de externe projectmanager was *“dat de software de mensen nu eindelijk zal dwingen op dezelfde manier te werken, en dat daarmee een einde wordt gemaakt aan oeverloze discussies over wie de beste werkwijze heeft.”* Een achteraf treffend eufemisme hiervoor was het woord *“harmonisatie”*. Achteraf treffend omdat het proces zelf niet zo harmonieus zou verlopen.

De voorgespiegelde jaarlijkse baten bedroegen een veelvoud van de kosten (een van de *“onderbouwingen”* daarvan betrof de uitspraak dat er dankzij het systeem minder storingsonderhoud zou zijn ...).

De implementatie, zo werd de Raad van Bestuur beloofd, zou maximaal 6 maanden duren, en het benodigde budget was ongeveer 1 miljoen Euro voor externe ondersteuning. Omdat de RvB veel kennis had van cijfers, en omdat er al wat ervaring was opgedaan met niet helemaal gelukke IT-projecten, werden tijdsduur en kosten als belangrijkste parameters voor het project neergezet. Wat harmonisatie nu precies inhield was minder duidelijk, en er werd ook niet op doorgevraagd. Wat er zou worden opgeleverd was ook niet zo duidelijk gespecificeerd.

Om te voorkomen dat budget en tijdsduur zouden worden overschreden werd Prince2 als projectmanagement systeem gehanteerd. Het systeem werd vervolgens vooral gebruikt om beslissingen af te dwingen waarvan veel gebruikers zeiden dat er nog onvoldoende was nagedacht over de consequenties. Maar het tijdspad was tot uitermost doel verheven. Uiteindelijk werd ook de scope steeds beperkter: *“na de*

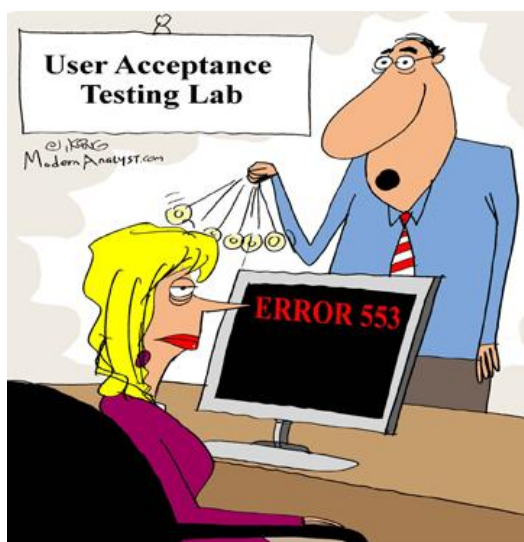
*implementatie zullen we de details wel verder invullen*". Dit betekende o.a. dat binnen het project een equipmentstructuur werd ingevoerd, die niet voldeed aan de eisen van gebruikers in Onderhoud.

Wat waren de resultaten van dit project? Het bleef vrijwel binnen budget (o.a. door de projectorganisatie eerder dan voorzien grotendeels te ontdoen van externen), en het was bijna op tijd klaar. Maar de geboden functionaliteit en de ermee verbonden harmonisatie waren minimaal en veel minder dan de gebruikers hadden verwacht en hen was voorgespiegeld. Men is nu nog steeds bezig om de functionaliteit voor hen in orde te krijgen. Over de baten is nooit meer gesproken.

Is dit een dom bedrijf? Neen! Waren het domme mensen? Zeker niet! Neem van mij aan: het is een bedrijf dat in zijn core-business trendsetter is, winstgevend is, en veel en terecht respect afdwingt. Niettemin waren er medewerkers die zeiden dat dit project niet zou gaan leveren wat werd toegezegd. Er waren ook mensen die zeiden dat je niet tegelijkertijd een zware implementatie van nieuwe en gecompliceerde engineering software moest starten omdat dat teveel beslag zou leggen op de mensen die ook bij de ERP-implementatie nodig waren. Er waren mensen die zeiden dat je eerst moest harmoniseren, en daarna pas het ERP-systeem moest invoeren. Er waren mensen die vroegen om nadere duiding. Er waren gebruikers die vanaf het begin sceptisch waren of er wel voldoende budget was, en of de tijdsduur niet te optimistisch was ingeschat. Nou, die lastige mensen kom je toch in de meeste IT- projecten tegen?

Toch waren er veel signalen die vanaf het begin aangaven dat de zaken echt anders waren dan men dacht of had aangenomen. Daarmee werd niets gedaan, ze werden niet herkend, of ze werden ontkend. De meer

concrete signalen kwamen bij de harmonisatie-sessies. Die liepen anders dan gepland, de voorbereidingen bleken niet aan te sluiten bij de gebruikers. En er was weinig tijd voor reflectie. Gebruikers spraken de taal niet van de consultants, en vice versa. Er was dus veel discussie over semantiek. Gebruikers bleven weg uit sessies. Het project dreigde vertraging op te lopen, de kosten veel hoger. Door de projectorganisatie werd voorgesteld de scope te wijzigen zodat de doelen, waarop afgerekend werd, gehaald zouden kunnen worden. De geharmoniseerde werkwijzen werden door de consultants op papier gezet onder het motto "zo zou het moeten kunnen". Veel gebruikers hadden inmiddels mentaal afscheid genomen.



***"You love the system! ... You love the system!"***

Ondanks de onverdachte en goede bedoelingen: iedereen draafde gewoon door. Als lemmingen. En er was uiteindelijk niemand die zich opwierp als winnaar van deze moeizame strijd. En de reactie van de direct verantwoordelijke managers was zoals gebruikelijk: ontkennen en overtuigen. En mocht u

denken dat dit wel een heel uitzonderlijk voorbeeld is: onderwerp uw eigen project(en) eens aan kritische analyse. En maak een lijstje met wat achteraf beter had gekund en had moeten. En ga de reflectie over verwachtingen dan niet uit de weg. Er is een gerede kans dat ook u tot de conclusie komt dat er in aanvang wel wat meer aandacht had kunnen worden besteed aan de gebruikers en hun verwachtingen.

## Ontkennen en negeren van complexiteit

Heel, heel veel mensen zouden in hun dagelijks leven kunnen volstaan met een pizza-model van de aarde. Ze hebben de meer diepgaande kennis niet nodig. Richard Feynman, winnaar van de Nobelprijs voor Natuurkunde in 1965 en de ontdekker van de oorzaak van de ramp met de Challenger, heeft eens een interessant betoog gehouden over de waarom-vraag.<sup>3</sup> Elke volgende “waarom” legt een interessanter



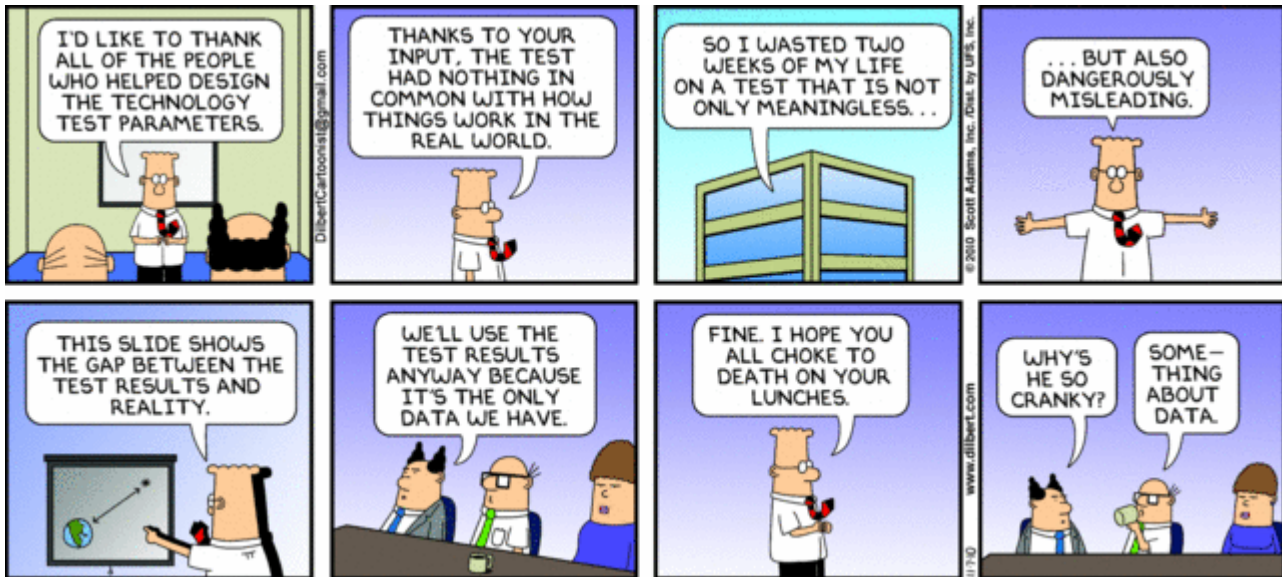
fenomeen bloot, maar lopende het traject wordt de werkelijkheid complexer, en het vinden van de antwoorden moeilijker. Bovendien kan de “waarom”-vraag alleen maar worden beantwoord binnen het kader van de kennis en ervaring van degenen die een antwoord moeten geven en van degenen die een antwoord verwachten. Voor mensen die van simpel houden, of voor mensen die niet over de vereiste kennis beschikken, is het eenvoudig: vraag niet naar de achterliggende waaroms, en ontken verschijnselen die in strijd zijn met wat je graag wilt denken vanuit je eigen model van de werkelijkheid.

ERP-systemen gaan onvermijdelijk uit van een holistisch model van de bedrijfs“wereld”. En hoe “vollediger” het model des te complexer de software. Maar die wordt daarmee niet beter of begrijpelijker voor de gebruiker. Want die leeft in zijn eigen deel van de wereld, en dat is al complex genoeg. Vandaar het succes van relatief eenvoudige stand-alone software voor onderhoud.

<sup>3</sup> [http://www.youtube.com/watch?v=wMFPe-DwULM&feature=player\\_detailpage](http://www.youtube.com/watch?v=wMFPe-DwULM&feature=player_detailpage)



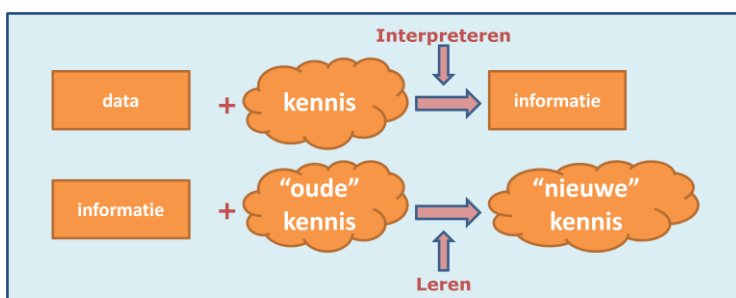
Voor gebruikers is het aangenaam om te denken dat hun aarde een pizza is. En ze hebben weinig interesse in de derde dimensie, want ze hebben daar geen voordeel van. Althans dat denken ze. En probeer ze maar eens te overtuigen van het tegendeel. En laten we eerlijk zijn: zelfs de pizza van een eenvoudige werkvoorbereider is zó complex dat een systeem niet meer kan zijn dan een grove, maar hopelijk voor de gebruiker bruikbare benadering van diens wereld.



IT-consultants weten vaak heel goed hoe complex de werkelijkheid is, en hoeveel tijd, moeite, energie, en geld het kost om iets echt bruikbaar neer te zetten. Maar velen van hen negeren de complexiteit onder de druk van managers met gebrek aan kennis, hun eigen bazen, beperkte budgetten, en een te kort ingeschatte implementatietijd. Het ontkennen en negeren van die complexiteit kost uiteindelijk een veelvoud van de projectkosten.

## Data en informatie zijn niet hetzelfde

Het is een wijd verbreid misverstand dat een informatiesysteem informatie geeft. Dat is, zeker voor onderhoudsinformatiesystemen, voorlopig nog onjuist. Een onderhoudsinformatiesysteem genereert data, en al dan niet bewerkt of geaggregeerd: het blijven gegevens. Gegevens waarmee we hopelijk iets belangrijks kunnen doen, namelijk: van gegevens informatie maken. Hoe maken we van gegevens informatie? Door onze kennis los te laten op de gegevens. En wie beschikt over die kennis? Juist, het is de gebruiker. De informatie die de gebruiker genereert zorgt ervoor dat zijn "oude" kennis wordt opgefrist, en leidt tot "nieuwe" kennis. Dat is de essentie van



leren. Data leveren nooit kennis op, het gaat om de interpretatie van de gegevens. Het is vervolgens de gebruiker die concreet aan de slag gaat met de door hem gegenereerde informatie. Een minder deskundige gebruiker haalt uit gegevens minder dan een terzake kundige gebruiker. Of trekt zelfs foute conclusies. Discussies tussen een boekhouder en een onderhoudsmanager over de onderhoudskosten zijn dan ook vaak erg gemakkelijk om aan te horen: twee werelden die beschikken over dezelfde data, en er met hun eigen

specifieke kennis hun eigen informatie van maken. En dus met verschillende interpretaties van de werkelijkheid komen. En daar dan een gesprek over aangaan.

Het is opvallend dat gebruikers in onderhoud heel vaak en graag praten over onjuiste, ontbrekende, of onvolledige data. Een IT-consultant belooft dan dat dit met een ERP-systeem allemaal goed komt. Maar nu stellen we de waarom-vraag maar weer eens. Hoe komt het dat de gebruiker nu niet beschikt over de juiste data? U zult tot de conclusie komen dat voor het genereren van betrouwbare gegevens (bijvoorbeeld gegevens over het uitgevoerde werk) meer en andere dingen nodig zijn dan alleen een ERP-systeem.

Het gebiologeerd zijn door data is ook niet zo verwonderlijk: dat is precies wat de gebruiker in de onderhoudsdienst nodig heeft om met onnaspeurbare en impliciete hersenactiviteiten data om te zetten in informatie die hem helpen zijn werk goed en beter te doen.<sup>4</sup>

Wat dus een onverdachte zegen is van een ERP-systeem voor onderhoud: de mogelijkheid om statische gegevens gestructureerd op te slaan en te onderhouden. De equipmentstructuur en de BoM zijn daarvan prachtige voorbeelden, die een functionaliteit bieden die zonder automatisering ondenkbaar is. Maar ze genereren geen informatie.

## De aandacht voor de gebruiker is een mythe

Er is bij implementaties van een ERP-systeem meestal veel aandacht voor de toekomstige gebruiker. “We moeten ze erbij betrekken, want we richten het systeem in voor hen” is dan het motto. Dit zijn nobele gedachten, maar de praktijk is toch wat weerbarstiger. Ik noem een paar zaken die mijns inziens een rol spelen bij het feit dat we onderweg gebruikers verliezen:

1. Gebruikers hebben helemaal niet om een ERP- systeem gevraagd. Wij richten het systeem ook niet in voor hen, maar voor managers. Het is meestal het management dat vindt dat onderhoud te weinig transparant is, en dat een systeem daarvoor de oplossing is. Wie zou ze dat toch hebben wijsgemaakt?
2. De meeste gebruikers hebben wel een idee aan welke gegevens ze behoefte hebben. Maar dat zijn nu juist vaak zaken die in een implementatie onder de kar terecht komen. Equipment tree, BoM, planning van werk, een transparant voor onderhoud geschikt budgetterings- en kostensysteem. Functionaliteit ontsluiten is soms (maar zeker niet altijd) mogelijk, maar er voor zorgen dat gegevens in het systeem zijn ingebracht, voordat de zaak live gaat, is iets anders want dat kost heel veel geld! We kopen een auto, maar er is geen geld voor benzine.
3. Een andere fout die regelmatig gemaakt wordt, is de vraag aan gebruikers welke informatie zij graag willen hebben. Dat is een outputvraag en dus een vraag om hun brein te expliciteren. Gebruikers in onderhoud praten liever over data die ze nodig hebben, die ze graag op een eenvoudige manier zouden willen kunnen benaderen. En natuurlijk zijn gebruikers bereid om data aan te leveren, als ze er daarna voordeel van hebben (en er dus informatie van kunnen maken). Ik was er zelf bij toen een consultant aan een gebruiker vroeg wat hij dan allemaal ging doen met die data die hij zo graag zou zien, want het was wel erg veel en erg gedetailleerd. Een duidelijker bewijs van zijn onkunde kon hij

---

<sup>4</sup> Ik moet hierbij een aantekening maken. Deze stelling gaat vooral op voor onderhoudsomgevingen. Voor productieplanners bijvoorbeeld, kan een ERP-systeem een zegen zijn, omdat in MRPII-runs (met al zijn gecompliceerde berekeningen) echte informatie wordt geleverd die een mens, hoe slim ook, niet snel en goed zou kunnen genereren. In onderhoud hebben we echter weinig tot geen voorbeelden van dit soort geautomatiseerde slimmigheden. En waar dit al zou kunnen (capaciteitsplanning en spare parts management) is de functionaliteit voor onderhoud nog steeds gebrekkig.

aan de gebruiker niet laten zien. En ik geef het je te doen om als werkvoorbereider/planner uit te leggen waarom je een gestructureerde en eenvoudig toegankelijke skill-matrix van uitvoerenden nodig hebt, en dat die skills vooral betrekking moeten hebben op soorten equipment. Ik heb het zelf meegemaakt dat een werkvoorbereider die vroeg om een aantal werkorder statussen (wacht op scoping, in voorbereiding, voorbereid, wacht op materiaal, gereed voor uitgifte, in weekplanning, in uitvoering, wacht op voorbereidingsanalyse, wacht op technische analyse, technisch gereed, administratief gereed) werd afgeserveerd. 11 verschillende statussen was wel erg complex, en daar moesten we maar niemand mee lastig vallen. En het allerergste: de betrokken consultant werd geprezen door de opdrachtgever, want we kunnen natuurlijk niet op alle “nice to haves” ingaan...

4. Pigeonholing door consultants. Oplossingsgericht denken in plaats van probleemgericht denken. Een van de fouten die bedrijven maken die regelmatig iets ergs overkomt. Ook managers zeggen graag: “ik wil geen problemen horen, maar oplossingen”.



En daar gaan we dan: regelrecht richting de afgrond. De gebruiker kent wel het probleem, maar kan in veel gevallen de oplossing niet zelf verzinnen of de geboden ERP-oplossing niet beoordelen. Totdat die oplossing niet blijkt te werken. Pigeonholing is noodzakelijk, want het maakt het inrichtingsproces van het systeem efficiënter. Maar niets vervelender voor de gebruiker dan een consultant die zegt (zonder echt te luisteren): daarin voorziet het systeem (ja,ja

vanuit het model van de werkelijkheid dat in het systeem is ingebouwd, en vanuit het model dat de consultant in zijn hoofd heeft).

5. Herdersgedrag. Zowel de ERP-software, als de consultants én de opdrachtgevers vertonen dit gedrag. Zij weten wat goed is voor de gebruikers. De filosoof Cornelis Verhoeven heeft ooit een mooie metafoor beschreven over herders en schapen:



*“... als mensen namelijk met schapen worden vergeleken, samenlevingen met kuddes en hun leiders met herders, mag dat wel goed gebeuren en een beetje steunen op de gang van zaken in de werkelijkheid buiten de vergelijking. ... Van pastorale zijde wordt voorgesteld alsof de herders en niet de schapen deskundig zijn op het gebied van gras. Dat is een merkwaardige pretentie. Niet de herder, maar de*

*kudde heeft verstand van gras en weet waar het te vinden is. Een wezen dat nooit gras heeft gegeten, moet zich daarover geen oordeel aanmatigen. Schapen mogen dan de naam hebben dom te zijn, van gras weten ze alles. Een tweede punt is dat de symbolische herders doen alsof zij de kudde wijzen waar die moet grazen en alsof ze dus de kudde voorgaan naar de grazige weiden. In werkelijkheid volgen de herders de kudde en lopen de belhamels voorop.”<sup>5</sup>*

---

<sup>5</sup> Cornelis Verhoeven, “Een register, bedenkingen over woordjes en glossen”, (p 139) Van Genneep, Amsterdam 1995

6. Last but not least: het ontbreken van een Lingua franca. Als twee mensen niet dezelfde taal spreken is de Lingua franca de taal waarvan zij zich beiden kunnen bedienen om met elkaar te communiceren. Het Engels is een goed voorbeeld van zo'n taal. Bij de invoering van een ERP-systeem hebben gebruikers en consultants vaak hun eigen vakjargon. Ze spreken niet dezelfde taal,

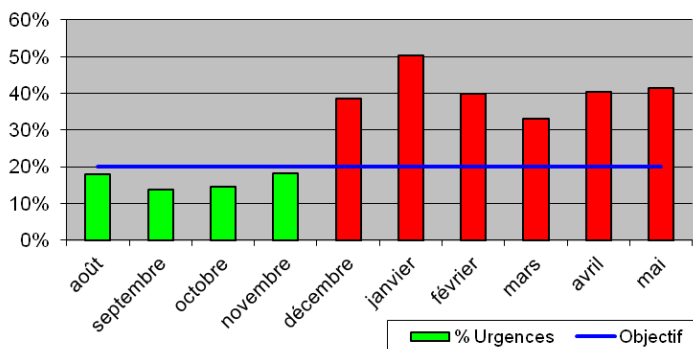


al zal het in veel gevallen toch Nederlands zijn. Maar de gemiddelde gebruiker weet niet goed wat het IT-jargon inhoudt, niet in de laatste plaats omdat er veel engelstalige termen worden gebruikt en niet te vergeten een stortvloed aan afkortingen (ook op beeldschermen). En een IT-consultant heeft helaas niet altijd kennis van het onderhoudsjargon of van de precieze betekenis van woorden. Of van de betekenis die de gebruiker aan een woord geeft.

Een van de voorbeelden van het negeren van de gebruiker uit mijn praktijk betrof een onderhoudsproject in het franstalige deel van België. En wat ik nu ga bespreken komt veel voor bij SAP-implementaties. In het project hadden wij moeite gedaan om het verschil duidelijk te maken tussen urgentie en prioriteit (vooral aan Productie). Voor de niet ingewijden: urgentie zegt iets over het belang dat een aanvrager van werk hecht aan de mate van spoed waarmee het werk gestart wordt. Een urgentie 0 is dan bijvoorbeeld werk in verband met een calamiteit. Een urgentie 1 is ook spoed, maar met enig overleg over starttijdstip. Een urgentie 2 betekent dat het werk voorbereid en gepland kan worden.

Voor planbaar werk hadden wij Productie opgevoed om voor urgentie 2 altijd een gevraagde en realistische leverdatum op te geven. Na feedback vanuit Onderhoud werd een definitieve leverdatum afgesproken, en het was aan Onderhoud om er vervolgens voor te zorgen dat die leverdatum gerespecteerd werd. Een en ander had veel moeite gekost, maar uiteindelijk werd Productie overtuigd en de gevraagde routine werd gedisciplineerd uitgevoerd: het stellen van de juiste urgentie en een gevraagde leverdatum voor aanvragen met urgentie 2.

% Urgences



Onderhoud bleek in staat (omdat niet alles bloedspoed was en ASAP ter hand moest worden genomen) het planbare deel van het werk vrijwel altijd op de afgesproken leverdatum klaar te hebben. Iedereen tevreden.

Nu vraag ik u om voor u verder leest naar het plaatje hiernaast te kijken met de KPI "percentage urgente aanvragen". De aanvragen met urgentie 0 en 1, die in feite het werk van Onderhoud verstoren.

Wat u ongetwijfeld opvalt: vanaf december steeg het percentage urgente aanvragen dramatisch. Begin maart kwamen de eerste ongeruste vragen vanuit het hoofdkantoor: wat is er aan de hand? Zijn de installaties aan het instorten? Wordt het preventief onderhoud niet meer uitgevoerd?

Het antwoord was eenvoudig: in december was SAP PM live gegaan. Dat leidde uiteraard niet tot meer urgent werk, al leek dat wel zo. Waarom was het percentage urgente opdrachten dan zo dramatisch gestegen? Vrij eenvoudig.

Ten eerste heet “urgentie” in SAP “prioriteit”. Waarom dat zo is, weet ik niet. En waarom de afkorting van prioriteit de letter Z is, weet ik ook niet en de gebruikers trouwens ook niet. Nu weet elke derde-rangs bedrijfskundige dat urgentie een onveranderlijk kenmerk van een aanvraag is, en dat een prioriteit kan veranderen in de tijd: het toekennen van prioriteiten is hetzelfde als het vaststellen van een plek in een wachtrij. Als de leverdatum van een geplande, niet urgente opdracht nabij is, en er is nog geen werk uitgevoerd wordt de prioriteit van het werk hoger. Prioriteiten toekennen is dus een dynamisch proces met eigen regels (prioriteitsregels), en een ervaren werkvoorbereider weet hoe hij daarmee om moet gaan, opdat leverdata gehaald worden.<sup>6</sup>

Het toekennen van een prioriteit in SAP zou beperkt kunnen blijven tot een semantische kwestie: “we noemen het prioriteit, maar we bedoelen urgentie”. Maar dat is helaas niet zo. Bij mijn klant werd voor elke SAP-prioriteitscode automatisch met een gestandaardiseerde doorlooptijd gerekend (die aanname zat in het systeem verborgen) en een leverdatum gegenereerd. Het tabelletje geeft verduidelijking:

SAP			“Project”		
Code		Leverdatum	Code		Leverdatum
Z1	Emergency	Zelfde dag	0	Calamiteit	Meteen starten tot klaar
Z2	Urgent	1 week	1	Potentieel zeer hoog risico	Z.s.m. starten in overleg
Z3	Routine	1 maand	2	Planbaar	Opgeven gevraagde leverdatum
Z4	Shutdown	3 maanden	3	Shutdown	-

De plotselinge stijging in urgente opdrachten was dus te wijten aan de wijze waarop SAP was ingericht, en aan het feit dat Productie “zag” dat Z3 automatisch van een leverdatum van 1 maand verder werd voorzien. Dat was in veel gevallen te laat (vond Productie), en dus werd een groot deel van de aanvragen een Z2 (en dus urgent), met een automatisch gegenereerde leverdatum die in veel gevallen onnodig dichtbij lag.

Wij hebben een aantal maanden vruchteloos geprobeerd om dit automatisme van het systeem op te heffen. Voor de goede orde: dit was een systeem dat in alle meer dan 100 fabrieken van het bedrijf werd ingevoerd met een verplicht uniforme inrichting. Volgens de projectleider en de stuurgroep was het daarom onmogelijk een uitzondering te maken... Bovendien zou het verplicht door Productie moeten opgeven van een realistische gevraagde leverdatum overal (behalve dan bij ons) leiden tot protesten. Drogredenen en misschien wel leugens, want dit detail was natuurlijk in geen enkele fabriek besproken.

Voordat deze “duidelijkheid” ontstond, waren we een half jaar verder. Uiteraard hebben we het vervolgens maar procedureel opgelost: Productie moest voor niet urgent werk code Z3 gebruiken en elke door het systeem gegenereerde leverdatum vervangen door een realistische gevraagde leverdatum. Om meerdere redenen niet ideaal, maar uit de grafiek bleek vervolgens dat de installaties gelukkig weer onder controle waren...

Wellicht aardig te vermelden dat in de meeste andere fabrieken na enkele maanden een enorm backlog-probleem ontstond. Werkvoorbereiders hadden geen overzicht meer over het werkpakket, en hadden door het ontbreken van realistische leverdata heel veel moeite om de juiste prioriteiten te stellen.

---

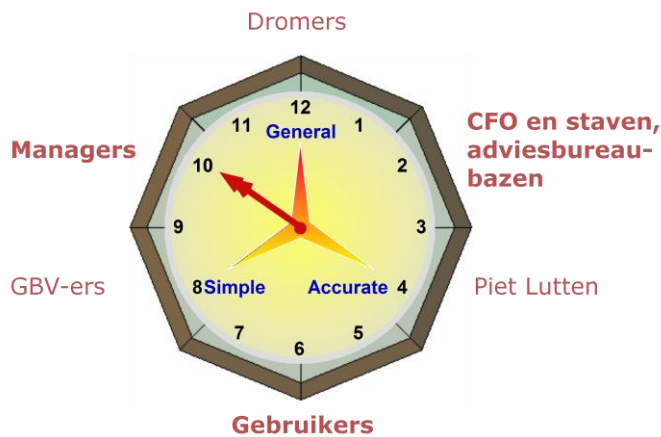
<sup>6</sup> In feite goochelt de werkvoorbereider met drie optimalisatiecriteria: aanhouden leverdatum, benutting van capaciteit, en minimale doorlooptijd. In productiebesturingssystemen een redelijk geautomatiseerd proces, in onderhoud nog een individuele vaardigheid van de werkvoorbereider.

## Streven naar eenvoud voor de gebruiker betekent keuzes maken

In het ideale geval zou een voor Onderhoud ingericht ERP-systeem aan drie eisen moeten voldoen:

1. Algemeen toepasbaar; de software is generiek ontworpen en toepasbaar in alle bedrijfssituaties, waar ook ter wereld; Verschillende fabrieken van hetzelfde bedrijf op verschillende lokaties kunnen met dezelfde software en met dezelfde inrichting werken.
2. Accuraat; de software vraagt en levert nauwkeurige gegevens, op het voor de gebruiker juiste niveau van detail;
3. Eenvoudig; de software is gebruikersvriendelijk, en aangepast aan de eisen van de gebruiker die er in zijn specifieke situatie mee moet werken.

Naar analogie van wat Weick<sup>7</sup> hierover heeft geschreven mogen we aannemen dat er slechts 2 van de 3 eisen kunnen worden gerealiseerd. Welke van de 2, is een keuze die expliciet gemaakt moet worden, omdat de derde eis wordt opgeofferd. Als we onze drie eisen afbeelden op een klok, zoals in onderstaande figuur, dan kunnen we ons punt wat verduidelijken. Op 12 uur zetten we de eis voor "Algemeen". Op 4 uur die van "Accuraat". En op 8 uur de eis voor "Eenvoud".



Op de 10-uur positie vinden we dan de systemen die generiek zijn, en eenvoudig. De accuratesse is opgeofferd. De casus in de vorige paragraaf is een voorbeeld. Veel managers voelen zich van nature thuis op deze plaats. In hun dagelijkse management-praktijk bedienen zij zich ook van algemeenheden, die eenvoudig klinken. "De kosten moeten omlaag", "we moeten harder werken", "We moeten de concurrenten op afstand zetten". "Het onderhoud moet transparanter". U kunt er vast nog meer noemen. Nauwkeurig zijn dit soort

opmerkingen niet, eerder vaag. De Engelsen noemen deze opmerkingen non-brainers.

Met dit managers-voorbeeld komt nog een ander aardig fenomeen naar boven: waar je je ook positioneert in de klok, je moet altijd heel goed blijven luisteren naar degenen die precies tegenover je staan. Managers op 10 uur moeten dus goed luisteren naar de 4-uur mensen die maar voor 1 eis kiezen, die van nauwkeurigheid. Managers hebben meestal een hekel aan dit soort lieden (de bekende Piet Lut, waarvan er in elk bedrijf wel een aantal te vinden zijn), en die bijvoorbeeld vragen stellen als "Welke kosten moeten omlaag?" "Wat bedoel je precies met kosten?" "Hoeveel moeten die kosten dan omlaag?" en "Wanneer moeten die kosten dan lager zijn?"

Gebruikers kiezen graag voor de 6 uur positie: eenvoud en accuratesse. Zij offeren daarmee de algemene toepasbaarheid op, en kiezen voor exact het tegenovergestelde: geen algemene oplossing, maar een specifieke oplossing, een inrichting van het systeem die precies voor hen passend is gemaakt en voldoet aan hun eigen specifieke eisen. Wat collega's in andere fabrieken of afdelingen met het systeem doen, moeten die zelf maar uitzoeken. Een niet helemaal terechte houding, maar wel begrijpelijk.

<sup>7</sup> Karl E. Weick, "The Social Psychology of Organizing", p 35-42, McGraww-Hill, New York, 1979, 2nd edition

We kennen natuurlijk allemaal mensen die zich beroepen op hun gezonde boerenverstand. Zij staan op 8 uur en kiezen maar voor 1 eis: simpel/eenvoud. Ze doen dat vaak op een gevaarlijke manier, een manier waarop consultants nogal eens verschrikt en bevestigend reageren: “het moet wel pragmatisch blijven. Ik ben een man van de praktijk, en al dat theoretisch gedoe dient tot niets”. Je bent geneigd de man gelijk te geven. Totdat je je realiseert dat hij pragmatisch gelijk stelt aan “eenvoudig”. En dan heeft de GBV’er natuurlijk geen gelijk. De praktijk is juist ontstellend complex, de theorie die over die praktijk gaat is daarentegen eenvoudig en begrijpelijk, meestal zelfs té eenvoudig.

De GBV’er komt in twee soorten (we doen nu even aan pigeonholing): degenen die de complexiteit van de praktijk hebben geïnternaliseerd, en daardoor hun praktijk als eenvoudig ervaren. Zij zijn onbewust vaardig. Zij kunnen een belangrijke rol spelen bij de inrichting van een systeem. Niet omdat ze mogen blijven hameren op de noodzaak voor eenvoud of pragmatisme, maar omdat we ze kunnen gebruiken om de precisie van het systeem te vergroten. We schuiven ze als het ware op naar de 6 uur positie. Aan de andere soort GBV’er moeten we niet veel tijd besteden. Te dom om complexiteit te begrijpen, en achter de steeds weer gedeclameerde mantra’s van pragmatisme en eenvoud zit meestal een forse dosis incompetentie verstopt. Verwar deze twee typen GBV’ers dus niet met elkaar.

Voor de ERP-ers onder ons is de 2 uur positie van belang. ERP-systemen worden in eerste instantie gekocht om de financiën van een bedrijf te administreren, en alles wat daarmee samenhangt (en dat is veel). Multinationals hebben een ERP-systeem ook nodig om de financiële resultaten van hun werkmaatschappijen te consolideren. Dat zijn erg moeilijke problemen, die met de vereiste accuratesse dienen te worden opgelost. Vanuit deze optiek wordt dus veel gekozen voor de 2 uur positie. Algemeen en accuraat. Maar er is niemand die zal beweren dat het daarmee ook simpel is. Integendeel, het ERP-systeem, en met name de inrichting ervan, is complex en daarmee voor velen in het bedrijf onbegrijpelijk. Wat echter ook onbegrijpelijk is, is het veel voorkomende feit dat de voor de financiën noodzakelijke complexiteit in een of ander automatisme wordt doorvertaald naar andere toepassingen van het systeem.

Het probleem met veel SAP PM implementaties is, dat we de eisen die vanuit een financieel perspectief terecht tot complexiteit leiden, één op één doorvertalen naar Onderhoud. Bazen van adviesbureaus vinden dit heerlijk, want het staat garant voor een omvangrijke hoeveelheid complex onderhoudsadvieswerk met bijbehorende hoge omzetten. Verwacht van hen dus niet teveel tegenwerk. Sterker nog, er zijn er die zullen benadrukken dat op enig moment met één dashboard de onderhoudsperformance van alle fabrieken zichtbaar en daarmee vergelijkbaar zal worden. Het klinkt goed, maar het zijn leugens.<sup>8</sup> En het is ook niet goed voorstelbaar waarom de PM-module voor alle bedrijven van een concern identiek moet worden ingericht (behalve dat bazen van adviesbureaus stellen dat daarmee de advieskosten lager zullen zijn, maar vraag je dan af waarom je die adviseurs nodig hebt). U moet zich twee zaken goed realiseren:

- Een boekhouder heeft (naast uiteraard financiële informatie over transacties met externen, maar die komt niet uit de onderhoudsadministratie) met betrekking tot Onderhoud niet veel meer nodig dan 10 tot 20 cijfers per maand, en die hoeven niet eens heel nauwkeurig te zijn. Hij heeft die nodig om de kostprijs voor de vervaardiging van een product te kunnen bepalen en te vergelijken met de vooraf berekende standaard-kostprijs. Dat alleen rechtvaardigt geen complex onderhoudsinformatie (vooruit dan maar) systeem dat volledig is geïntegreerd met de complexe financiële softwaremodule.
- Een adequate opvolging van de prestaties van een onderhoudsdienst heeft niets te maken met een financiële module van het ERP-systeem, hoe graag sommigen u dat ook willen laten geloven. Onderhoud gaat in essentie over beschikbaarheid van installaties, over technische activiteiten, en

---

<sup>8</sup> Voor wie geïnteresseerd is in dit soort leugens adviseer ik het boek te lezen van Jos Verveen: “Bullshit Management, terug naar de essentie van organisaties”, SDU, Den Haag, 2011.

over productiviteit van monteurs. Zaken die zonder enige link met een financiële administratie kunnen (en moeten) worden opgevolgd. Natuurlijk willen we graag weten wat een uur productieverlies kost, en zeker als dat verlies is veroorzaakt door onderhoud (of gebrekkig onderhoud). Dat soort overzichten kan echter ook wel op een eenvoudiger manier worden gemaakt. En als een onderhoudsdienst niet in staat is om dit zonder een ERP-systeem te maken, dan kunnen ze het zeker niet mét een ERP-systeem. Er zijn namelijk nog wat andere zaken nodig, zoals een gedisciplineerde registratie van betrouwbare gegevens.

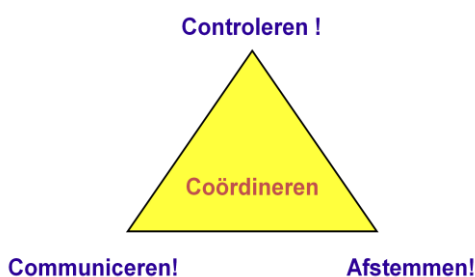
Hetzelfde geldt mutatis mutandis voor de MM-module. Als dromer zie ik het voordeel dat een fabriek in Nederland kan zien dat hun dringend benodigde reservedeel in een magazijn van een collega-fabriek in België ligt. Als dromer begrijp ik ook dat daarvoor een standaardisatie van benamingen nodig is (en een flexible slang heet in België nu eenmaal darm, en een schroef heet vijs). Dat is in een MM-module allemaal oplosbaar (met een goede equipmentstructuur en BoM is het zelfs niet eens heel relevant), maar het kost ook wat (alleen al aan mankracht, hopelijk niet van externe consultants). Toch ken ik meer voorbeelden van een goed geïmplementeerde MM-module dan van een goed ingevoerde PM-module. Dat komt met name omdat de dynamiek van het beheer van bijvoorbeeld 60.000 goed gespecificeerde artikelen uiteindelijk van een heel andere complexiteit is dan de dynamiek van een onderhoudsdienst die 60.000 manuur per jaar besteedt aan een veelheid van onderhoudsactiviteiten en installaties.

In het kader van de trade-off's zou ik voor Onderhoud altijd eerst kiezen voor eenvoudig en precies, en dus voor specifiek, en ik zou de algemene toepasbaarheid opofferen. Want de alternatieven zijn voor de gebruiker nauwelijks acceptabel. En de gebruiker is de klant. En de klant staat centraal in bijna alle bedrijven. Jammer dat ze dat dan vooral belangrijk vinden voor hun externe klanten, en niet voor hun interne klanten.

En last but not least: laten we als adepten van de 6 uur positie toch vooral goed luisteren naar de voorstanders van 'algemeen toepasbaar', de dromers. Dat kan ons in elk geval behoeden voor het indutten in de comfortable zetel van zelfgenoegzaamheid en "satisfactory underperformance".

## Projecten, coördinatie, en managen van verwachtingen

Coördinatie is een woord dat wij graag gebruiken als we het hebben over een project, of algemener: als we het hebben over het realiseren van meerdere met elkaar samenhangende activiteiten. Een projectmanager of een projectcoördinator is degene die verantwoordelijk is voor de coördinatie. Het kan daarom geen kwaad om het begrip wat nauwkeuriger te omschrijven. Daarvoor gaan we eerst naar een stukje theorie.



Zodra je een activiteit gaat opsplitsen in deel-activiteiten en die verdeelt over meerdere mensen ontstaat een coördinatieprobleem. Coördinatie heeft hier een heel precieze betekenis: het is afstemmen, communiceren, én controleren. Er zijn verschillende manieren om coördinatie van activiteiten te realiseren<sup>9</sup>. We kennen allemaal die van direct toezicht. Het betekent dat één persoon, meestal een manager maar het mag ook een teamleider zijn, of een projectcoördinator, verantwoordelijk is voor het controleren van alle activiteiten van

alle medewerkers. De communicatie over het werk verloopt ook via de manager, dus van medewerker naar manager en van de manager weer naar een andere (of meerdere) medewerker(s). Er is geen rechtstreekse communicatie tussen medewerkers, althans voorzover het over het werk gaat. Afstemming over de

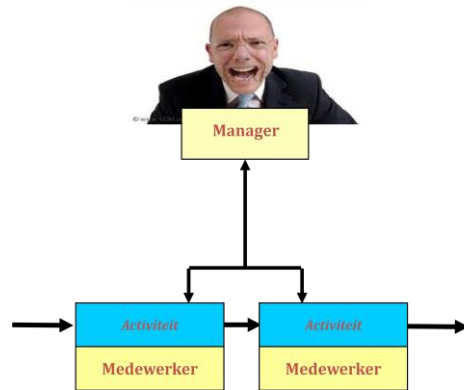
---

<sup>9</sup> Henry Mintzberg, "The structuring of organizations", (p 4-16), 1979, Prentice Hall Inc, Englewood Cliffs



werkzaamheden (bijvoorbeeld over de consequenties van opgelopen vertragingen, of over de vraag wanneer een bepaalde activiteit afgerond moet zijn of zal zijn) verlopen via hetzelfde pad: de manager als spin in het web. Deze mensen hebben het druk, en staan feitelijk voor een onmogelijke opgave, zelfs als het aantal medewerkers en het aantal verschillende activiteiten beperkt zijn.

U begrijpt dat de man op de maan daar niet is gekomen met dit coördinatiemechanisme. Er was echt geen

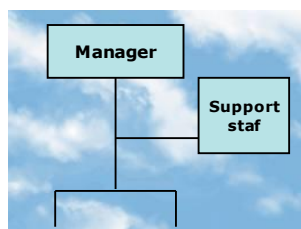


generaal via wie alle communicatie liep en die als de grote roerganger het hele project coördineerde. Bedenk dat alle communicatie en afstemming en controle loopt van de medewerker naar de baas. En via de baas weer naar de andere medewerker(s). Soms cultureel bepaald, soms noodzakelijk. Maar niet in de beginfase van het Apollo 11 project.

Het coördinatiemechanisme dat wél gebruikt is, heet wederzijdse aanpassing. Denk aan een een kano met twee roeiers: het heeft geen nut om van de kant af aanwijzingen te roepen hoe de twee moeten samenwerken. Het doel is duidelijk voor die twee: ze

willen ergens naar toe, hopelijk naar dezelfde plek. Hoe ze dat samen regelen is hun verantwoordelijkheid.

De fase waarin wederzijdse aanpassing noodzakelijk is om het uiteindelijke projectdoel te bereiken, is erg eng voor managers, die nu eenmaal vinden dat niets goed kan verlopen zonder hun inbreng. Dat laatste is



waar, maar die inbreng is een andere dan zij misschien denken. Zoals gesteld: de medewerkers zorgen zelf voor de coördinatie. De rol van manager is vooral: faciliteren. Zorgen dat er koffie is, en zo. Maar vooral niet vragen waar we staan, en ook niet vragen wat er klaar is, als het klaar is. De antwoorden zullen een tijdlang frustrerend zijn. Projectmanagers kunnen al helemaal niet tegen dit coördinatie-mechanisme. Het is in strijd met bijna alles wat ze geleerd hebben over

projectmanagement. Je moet ze er in deze fase dan ook niet bij betrekken. En consultants (als ze er al moeten zijn) doen mee als expert, een van de velen. Samen met de anderen vinden ze het pad, en ze worden dus niet ingeschakeld als de expert met betrekking tot het vinden van de vooraf opgelegde oplossing, en de weg daar naartoe.

Het moge duidelijk zijn dat een project uiteindelijk in een stadium komt dat wederzijdse aanpassing niet meer werkt. Uiteindelijk moet er toch iets gebouwd en ingevoerd worden. En dat weten projectmanagers dan weer heel goed voor elkaar te krijgen. Maar die initiële fase is cruciaal, want daarin worden de verwachtingen expliciet gemaakt, en op elkaar afgestemd, en het pad wordt uitgestippeld. Gezamenlijkheid, begrip van de complexiteit, expliciete verwachtingen en draagvlak zijn zodoende de belangrijkste uitkomsten, en vormen daarmee het fundament voor een implementatie met grote kans op succes.

## Er is een verschil tussen risico's en onzekerheden

Als u houdt van disco en uw partner van klassieke muziek; als u houdt van rumoer en uw partner van stilte; u van asfalt en uw partner van groen; u zuinig bent van aard en uw partner graag geld uitgeeft. Dan zijn er wat risico's voor een bestendige relatie. Maar risico's kunnen besproken worden, en ook hoe ermee om te gaan. Je kunt zelfs afspreken dat je de verschillen accepteert.

Maar de toekomst herbergt ook veel onzekerheden. En die kun je niet allemaal in kaart brengen, laat staan dat je er afspraken over kunt maken. Je wilt graag samen kinderen, en het lukt niet. Je wilt geen kinderen, en je krijgt er toch een. Jij komt in een rolstoel, en je partner moet duwen. Je verliest je baan, en je moet je huis verkopen. Je wint een miljoen. Je krijgt nieuwe bureaus die het afval over de schutting gooien. We kunnen nog wel even doorgaan.

Er is een analogie met ERP-projecten en degenen die daarin deelnemen. Er is geen projectplan dat de complexiteit van voorbereiding en implementatie kan vatten en regelen. Het gaat dus vooral om het in kaart brengen van activiteiten op hoofdlijnen en vooral van de risico's (om die vervolgens af te dekken), én om het accepteren van onzekerheden en daarop adequaat kunnen reageren op het moment dat ze zich manifesteren. Dat is ook de essentie van een goede projectmanager: adequaat reageren op onverwachte gebeurtenissen. We komen er later op terug.

Maar... als er te weinig tijd en energie is gestoken in het expliciet maken van verwachtingen, dan zal er weinig vertrouwen zijn in een goed verloop van het project, laat staan in een goede afloop. En als er geen vertrouwen is, worden onzekerheden verheven tot risico's. En die risico's moeten vervolgens allemaal beheerst worden, het liefst via een risicomangement-methodiek. En dus worden projectplan, projectorganisatie, communicatie, afstemming en controle zó ingewikkeld dat alleen ingewijden er nog iets van begrijpen. Zo ingewikkeld dat we een complex projectmanagementsysteem nodig hebben, en mensen die dit systeem "voeden". Een beter systeem om los te raken van de werkelijkheid en om niet meer adequaat te kunnen reageren op onverwachte en ongewenste gebeurtenissen is nauwelijks denkbaar.

Weick en Sutcliffe<sup>10</sup> hebben onderzoek gedaan naar organisaties die succesvol omgaan met onverwachte gebeurtenissen, en met name organisaties die in heel risicovolle omstandigheden moeten werken. Op basis van hun onderzoek komen zij tot de conclusie dat dit soort organisaties (zij noemen ze "High Reliability Organizations") niet falen in datgene te doen wat verwacht wordt vanwege een vijftal kenmerken die we in de volgende paragraaf zullen bespreken.

## Tips om de gebruikers niet kwijt te raken

De voorgaande paragrafen behandelden vooral wat er mis kan gaan en waarom, hier en daar met een theoretisch uitstapje. Dat ERP-implementaties regelmatig mislukken behoeft geen betoog. Maar hoe kunnen we zorgen dat een project geen mislukking wordt? Dat de gebruikers een systeem krijgen waar ze graag mee werken, en dat ze graag voeden met betrouwbare gegevens omdat ze er zelf wat voor terugkrijgen? Er is natuurlijk geen panklare oplossing, want dan hadden we die al wel gevonden. Maar er zijn wel zaken waarmee we rekening kunnen houden, en die de kans op succes aanmerkelijk vergroten. We geven er een aantal in de vorm van tips, en de tips staan in een weldoordachte volgorde...

---

<sup>10</sup> Karl E. Weick en Kathleen M. Sutcliffe, "Managing the Unexpected", Jossey-Bass, San Francisco, (2001)

## Tip 1: Formuleer het doel

Het formuleren van een doel lijkt triviaal, maar er worden regelmatig fouten mee gemaakt. Te ingewikkeld, teveel oplossingsgericht. Denk dus aan Kennedy en het doel dat hij formuleerde. Dat zorgde voor beweging en enthousiasme. Een van de fraaiste doelen, die ik zelf heb mogen meemaken, werd uitgesproken door de directeur van Volvo Cars Gent als start van een immens complex verbeterproject waarvan niemand (ook hij niet) kon vermoeden hoe ingewikkeld het was, en wat het allemaal zou vragen:

*“Tot nu toe worden alle nieuwe modellen van Volvo eerst geproduceerd in Zweden, want zij denken dat zij beter zijn dan wij. En dat is misschien ook zo. Maar wij gaan ervoor zorgen dat het eerstvolgende nieuwe model als eerste bij ons zal worden geproduceerd. Omdat niemand zal kunnen betwisten dat wij de beste automakers in het concern zijn”.*

En zo geschiedde. De eerste Volvo 850 werd in Gent geproduceerd, en de Zweedse fabrieken hadden het nakijken. Een revolutie in de rijke historie van het concern, zonder precedent.

In het geval van een ERP-systeem kan en mag het doel dus nooit zijn: “we gaan SAP implementeren”. Het doel moet concreet zijn en begrijpelijk voor iedereen, en mensen moeten begrijpen waarom hun bedrijf kiest voor dit doel. En het formuleren van het doel is een verantwoordelijkheid van het top-management, en van niemand anders. En het doel gaat dus nooit, nooit over een ERP-systeem.

## Tip 2: Maak verwachtingen expliciet: wederzijdse aanpassing

Het doel creëert al verwachtingen. Maar iedereen leidt daar eigen verwachtingen uit af. Daarom is het nodig al die individuele verwachtingen expliciet, concreet, en wederzijds te maken. Nog voordat er een project wordt opgetuigd. En let op: het woord ERP is nog niet genoemd! Het is tijd om een proces van wederzijdse aanpassing te starten. Dat voorkomt dat we elkaar een rad voor ogen draaien, en zaken beloven die we niet kunnen waarmaken. Dit proces is niet hetzelfde als het maken van afspraken. Verwachtingen zijn vaak veel subtieler en minder kwantificeerbaar, en dus ook minder meetbaar.

Het proces van wederzijdse aanpassing zorgt voor beweging en ideeën. Het zorgt ervoor dat gebruikers het vooral zelf gaan uitzoeken, samen met elkaar. Met het helder geformuleerde doel voor ogen. Verwar dit proces niet met brainstormsessies. Het gaat om werken, écht werken en vanuit de expertise van de gebruikers. Het moet breed en diep gebeuren, en een ERP-implementatie is nog helemaal niet aan de orde. Sommigen zullen zeggen dat de kosten van deze fase dan wel erg hoog worden. Dat is juist. Hoger dan u denkt. En het kost waarschijnlijk ook nog meer tijd dan u denkt. Maar de kosten van een mislukte ERP-implementatie zijn hoger.

## Tip 3: Organiseer het project

Het is niet waarschijnlijk, maar een van de uitkomsten van de initiële fase kán zijn dat de implementatie van een ERP-systeem noodzakelijk is om het doel te kunnen bereiken. Er ligt een nieuwe valkuil voor u. De verleiding is groot om met de inrichting en implementatie van de ERP-software te starten. Bedenk nu waarom ERP-projecten de naam hebben lang te duren. Dat ligt echt niet aan de weerbarstigheid en ondoorgrondelijkheid van de software. Een ERP vraagt om duidelijkheid, en als die er niet is dan moet die er komen. En voor je het weet besteed je kostbare maanden aan het duidelijk krijgen hoe we in de toekomst willen gaan werken, terwijl het management niet begrijpt waarom het ERP-systeem maar niet live kan gaan. Ook hier gaat het om het managen van verwachtingen.

Om het doel te bereiken zal er waarschijnlijk het nodige veranderd en formeel geregeld moeten worden. Dat inzicht is ontstaan tijdens de initiële fase. Het is beter om nu een projectfase te starten waarin de puntjes op de i gezet gaan worden. Ter verduidelijking wat voorbeelden uit onderhoud:

*Onze werkvoorbereiders hebben geen zicht op de status van opdrachten. Ze willen graag weten welke klussen nog voorbereid moeten worden. En welke klussen in de weekplanning kunnen worden opgenomen. En welke klussen nog wachten op de levering van materiaal.* Een klein groepje van deskundigen (werkvoorbereiders?) zal de statussen moeten vaststellen.

*We willen graag kosten van uren en materialen bijhouden.* Een wat omvangrijker groepje zal zich het hoofd moeten breken over vragen als:

1. Wat is het laagste niveau van de installaties waarop wij kosten willen registreren?
2. Hoe definiëren we dat laagste niveau? En wie is verantwoordelijk voor het opstellen en onderhouden van de installatiestructuur?
3. Hoe hoog is het uurtarief, en hoe bepalen we dat? Werken we met een integraal uurtarief? Met welk aantal productieve uren per jaar rekenen we, wat is de berekeningsmethode, en wie maakt de berekening en wie autoriseert?

*We willen graag technische historie bijhouden.* Ook weer een groepje dat moet aangeven welke gegevens we willen bijhouden, en op welk niveau van de installatie. Misschien moeten er zelfs storingscodes worden gedefinieerd.

*We vinden dat niet alle werkzaamheden op dezelfde manier moeten worden afgehandeld. Een klein klusje is toch iets heel anders dan een calamiteit of een modificatie.* Een groepje zal de verschillende soorten werk moeten vaststellen, én hoe die moeten worden afgehandeld. Procesbeschrijvingen met verantwoordelijkheden, bevoegdheden, en taken komen in beeld.

Al dit soort voorbeelden (en er zijn er uiteraard veel meer) kun je uitwerken op basis van de functionaliteit van het ERP-systeem. Maar het is niet aan te raden. Enerzijds omdat je er de doorlooptijd van de ERP-implementatie optisch mee verlengt. Anderzijds omdat gebruikers zich dan niet alleen moeten verdiepen in hoe ze hun activiteiten willen organiseren, maar tezelfdertijd ook de geheimen van het ERP-systeem moeten zien te doorgronden. De gebruikers zijn in deze fase nog leidend, en niet het systeem.

Deze ontwerpfase moet projectmatig worden aangepakt. Het leidend coördinatiemechanisme binnen de groepjes is standaardisatie van vaardigheden<sup>11</sup>, en dat betekent dat er groepjes worden samengesteld met deskundigen op de gebieden die nodig zijn om het werk van een groepje tot een goed einde te brengen. Maar om al die groepjes te coördineren, is een projectorganisatie absoluut noodzakelijk. In deze fase kan het nodig zijn om extern advies in te schakelen. Maar dan wel organisatieadviseurs die gespecialiseerd zijn in onderhoud. En die dus als volwaardig deelnemer mee kunnen doen in de ontwerp-groepjes.

In de ontwerpfase wordt de complexiteit van de werkelijkheid, van de praktijk pas echt duidelijk. Dit is daarom ook de fase waarin moet worden gestreefd systematieken en processen zo eenvoudig mogelijk te houden, om verantwoordelijkheden duidelijker te benoemen, en om steeds weer de vragen van effectiviteit

---

<sup>11</sup> Dit mechanisme lichten we niet verder toe. Voor de geïnteresseerden verwijzen we naar ref 9.

en efficiency te stellen. Ook de vraag “waarom” we bepaalde zaken die we graag willen invoeren nog niet hebben, kan tot leerzame inzichten leiden waarmee we ons voordeel kunnen doen bij de implementatie. Met een continue kritische toetsing van de bijdragen aan het oorspronkelijk vastgestelde doel, streven we dus naar eenvoud en acuratesse:

*“Alles moet zo eenvoudig mogelijk worden gemaakt, maar niet eenvoudiger” (A.Einstein)*

#### **Tip 4: Formaliseer het ontwerp en benoem beheerders**

De ontwerpfase resulteert in een model van de werkelijkheid voor alle voor Onderhoud relevante gebieden. Het is belangrijk dat de bedrijfsleiding dit model, de ontwerpen, formeel goedkeurt. “Zo gaan we het doen, en niet anders”.

Dit is ook het moment dat er beheerders voor het ontwikkelde gedachtengoed moeten worden benoemd. Niet alleen essentieel om zaken te blijven aanpassen aan veranderende omstandigheden en voortschrijdend inzicht, maar ook om nieuwe gebruikers te trainen. De beheerders dienen tevens als vraagbaak voor de gebruikers, zij zijn de “most trusted advisor” voor hun expertisegebied.

Het aanwijzen van beheerders is ook belangrijk voor het vervolgtraject, als we het ERP-systeem gaan inrichten en implementeren. Want het is voor de IT-consultants helemaal niet meer nodig om alle gebruikers te raadplegen. Hun gesprekspartners bij uitstek zijn de beheerders, die de gebruikers vertegenwoordigen. In IT-termen heten ze ook wel key-users, maar dat is niet helemaal hetzelfde. Door deze mensen al voorafgaand (en om andere redenen) te benoemen zijn ook op het punt van functionele expertise de verwachtingen expliciet gemaakt. Een eenvoudiger organisatie van het vervolgtraject is daarmee gerealiseerd.

Door het benoemen van “systematiekbeheerders”, wordt ook de connotatie van het woord key-user anders. Dit zijn geen mensen die veel weten van het systeem, en hoe het kan worden aangepast. Dit zijn mensen die verzoeken tot aanpassingen kunnen beoordelen op hun merites, en die ze met redenen omkleed kunnen afwijzen of accepteren. Dat ze de aanpassingen ook in het systeem kunnen aanbrengen is slechts een deel van hun verantwoordelijkheid en vaardigheid. De rol van key user wordt duidelijker. En de karikaturen van deze lieden minder valide: het zijn geen mensen die overal tegen zijn, omdat ze opzien tegen het werk, geen zin hebben zich te verdiepen in de vraag, geen notie hebben van waar het over gaat, of domweg geen tijd hebben. En die dus lak hebben aan de gebruikers. En aan de andere kant van het karikatuuerspectrum: het zijn ook geen mensen die op elke vraag ingaan onder het motto “u vraagt, wij draaien”. Dat lijkt gebruikersvriendelijk, maar doordat onvoldoende wordt gekeken naar consequenties voor anderen is het dat juist niet.

#### **Tip 5: Organiseer het ERP-project!**

Als alle voorgaande tips zijn opgevolgd is het inrichten en implementeren van het ERP-systeem nog steeds geen sinecure. Maar het werk is voorbereid, de verwachtingen zijn duidelijk en misschien zelfs hooggespannen, het draagvlak aanzienlijk, en de gesprekspartners voor de IT-consultants zijn benoemd. En bovendien: het doel is bekend en nog steeds leidend.

Hoe moet het project worden georganiseerd? Daarover zijn boeken vol geschreven, en projectmanagementsystemen geven ook nog wel wat aanwijzingen. We beperken ons tot een paar essentiële punten, die nog wel eens uit het oog worden verloren.

1. Bouw de organisatie bottom-up. We beginnen graag met het benoemen van een projectmanager. Die zal wel nodig zijn, en als hij of zij de leiding moet nemen, is het maar beter om hem/haar te betrekken bij de inrichting van de projectorganisatie. Maar start vervolgens met de systematiekbeheerders. Voeg aan elke beheerder een IT-consultant toe, die diepgaande kennis heeft van de functionaliteit van het in te voeren systeem én voldoende materiekennis heeft om de juiste vragen te kunnen stellen aan de beheerder en om de antwoorden te kunnen vertalen naar systeemfunctionaliteit. Voeg aan deze duo's alles toe wat nodig is om het systeem te implementeren. Dat zijn meestal de mensen die de technische IT-aspecten voor hun rekening moeten nemen, en dat mag zeker niet onderschat worden. Wees voorzichtig met het benoemen van allerlei mensen die het project moeten volgen en administreren. Een assistent van de projectmanager is meestal voldoende, en zelfs die is niet altijd noodzakelijk.
2. Maak een projectplan en planning op hoofdlijnen. Een te gedetailleerde planning betekent veel wijzigingen tijdens het traject. Heb enig vertrouwen in de groepjes van deskundigen. De beste eigenschap van vakmensen is dat ze weten wanneer ze moeten ophouden, en hulp moeten vragen of een signaal moeten afgeven. Met andere woorden: geen nieuws is goed nieuws. Zet dus deadlines met een duidelijke omschrijving van wat er moet zijn opgeleverd op de betreffende datum. En zeg ook nog iets over de toegestane tijdsbesteding.
3. Hoe u de activiteiten ook inricht: zorg dat eerst de functionaliteit met betrekking tot statische gegevens wordt ontsloten en ingevoerd. Zorg ook dat de statische gegevens zijn ingebracht voordat het systeem live gaat. Ik noem met name de installatiestructuur en de BoM. Want die vormen de ruggengraat van het onderhoud en van het onderhoudssysteem. De gebruikers zullen u dankbaar zijn. (Zorg wel in het projectplan dat de consequenties helder zijn aangegeven, want hier kan ook nog wel wat misgaan met betrekking tot verwachtingen).
4. Benoem een goede projectmanager. Dat hadden we in punt 1 al gedaan, maar nu komen we terug op een aantal eigenschappen. Stel u hebt de keuze tussen twee gelijkwaardige kandidaten. Een van de kandidaten heeft veel ervaring en een bewezen goed track-record. De andere ook, maar heeft u ook gezegd dat hij heel vaak geluk heeft gehad. Kies de laatste, want hij beschikt blijkbaar over de competentie om naar bevind van zaken op de juiste manier te handelen. Moet een projectmanager over ervaring beschikken? Ja, het helpt zeker, en kennis van een doorwrocht projectmanagementsysteem is altijd meegenomen. Maar veel belangrijker zijn de eigenschappen die hem (of haar uiteraard) in staat stellen om goed om te gaan met onverwachte gebeurtenissen. En onverwachte gebeurtenissen zullen er altijd zijn. En het ermee goed omgaan is het verschil tussen succes en mislukking. Daarover gaat de laatste tip.

### **Tip 6: Zorg voor betrouwbaarheid**

Betrouwbaarheid is “niet falen in datgene te doen wat verwacht wordt”. Veel projectmanagementsystemen gaan uit van de maakbaarheid van de wereld. Als we maar gedetailleerd genoeg activiteiten specificeren en plannen, en strikte controlemechanismen inbouwen: dan komt het goed. Dat is helaas niet het geval. Zeker, u moet het project organiseren, en zo goed mogelijk. Maar er zullen zich toch onverwachte gebeurtenissen voordoen. Gebeurtenissen die u niet heeft voorzien, niet heeft kunnen voorzien, en vooral: niet heeft willen voorzien (denk aan de uitleg over het verschil tussen risico's en onzekerheden).

## Anticiperen en voorkomen

Onverwachte gebeurtenissen dienen zich vaak op heel subtiele wijze aan. Ze hebben zich nog niet gemanifesteerd, maar het rommelt al wat. Om op onverwachte gebeurtenissen te anticiperen, en om te voorkomen dat ze zich manifesteren (meestal op een moment dat niet slechter zou kunnen zijn uitgekozen) zijn drie eigenschappen van de organisatie (en vooral van de projectmanager) essentieel.

1. Alert zijn op alles dat afwijkt van wat verwacht werd. Geen grote afwijkingen of grote dingen, neen juist kleine dingetjes. Zaken waarvan je zou kunnen zeggen: dat is zo onbelangrijk, daar hoeven we even niks mee te doen. Of dat hoeven we niet te rapporteren. Het tegendeel is waar: de kleine afwijkingen zijn voorboden van iets heel groots, misschien zelfs van een calamiteit. Er moet dus op gereageerd worden. En in tegenstelling tot wat je zou denken: er moet fors op worden gereageerd. Een van de eigenschappen van een goede projectmanager is dus dat hij alle antennes op maximale ontvangst zet. Alles dat afwijkt is belangrijk, de rest nauwelijks. Heel begrijpelijk ook, want we hebben juist veel tijd en energie besteed aan het expliciet en gemeenschappelijk maken van verwachtingen.
2. Terughoudend zijn met het simplificeren van je interpretaties. Niets is wat het lijkt. Het is altijd complexer dan je zou willen. Als mensen de moeite nemen om uit te leggen hoe "iets eigenlijk zit", neem de moeite om te luisteren en te doorgronden. Verdiep en nuanceer. En voorkom daarmee toekomstige problemen.
3. Heb gevoel voor en erkenning van de gebruikers en hun werk. Wees alert en bewust. Blijf gebruikers informeren over de stand van zaken (want zij worden vertegenwoordigd door de systematiekbeheerders, en kunnen dus gemakkelijk het contact met het project verliezen), en dus niet over het tijdsplan en de milestones maar vooral over de inhoud van het werk dat gedaan wordt. Zorg dat ze betrokken worden bij het testen van het systeem, en neem opmerkingen heel serieus (zie punt 1) en doe er iets mee. En er iets mee doen betekent ook concreet iets doen, en niet zeggen dat het wel goed komt of zeggen dat het later wel "wordt meegenomen" of nog erger: zeggen dat de opmerkingen interessant zijn maar nu niet relevant. Of dat er geen tijd is om iets met de opmerkingen te doen.

## Corrigeren en in bedwang houden

Ondanks het feit dat we voor de condities hebben gezorgd om te kunnen anticiperen op overwachte gebeurtenissen, kunnen ze zich toch manifesteren. Dan komt het er echt op aan. Er zijn twee eigenschappen van de organisatie (en wederom vooral van de projectmanager) die cruciaal zijn.

1. Veerkracht, het heft in handen nemen. Een van de belangrijkste consequenties: alle afspraken en regels worden terzijde gezet, sterker nog: ze worden overtreden. Het waarom is duidelijk: de afspraken en regels zijn gemaakt voor wat we hebben voorzien. Maar nu is er iets onverwachts gebeurd. En daarvoor zijn die afspraken en regels niet gemaakt. En dus zijn ze niet geldig. Het vraagt moed om de zogenaamde overtredingen te begaan, maar het is essentieel. Een goede projectmanager weet dit, en als hij het niet weet dan doet hij het intuïtief. Hij legt het achteraf wel uit. Zijn enige focus is de situatie weer stabiel te krijgen, desnoods ten koste van een reprimande of gezeur van opdrachtgevers. Mensen die in een oorlog hebben gevochten weten hierover veel te vertellen. Dát ze het nog kunnen vertellen is te danken aan hun veerkracht, en

aan het feit dat ze de richtlijnen van de hogere legerleiding en staven (die meestal redelijk veilig en op afstand zitten) tijdelijk even naast zich hebben neergelegd.

2. Diep respect voor expertise, voor de deskundigen. Als er een probleem is dus niet escaleren via de hiërarchie, want uiteindelijk komt het bij iemand die niet begrijpt waarover het gaat, maar wel zegt “beter een foute beslissing, dan geen beslissing”. En die neemt hij dan ook, die foute beslissing. Bij een onverwachte en ongewenste gebeurtenis moet er dus naar opzij of naar beneden gedelegeerd worden. Naar de deskundige die het probleem zou kunnen begrijpen en doorgronden. En die dus bij uitstek gekwalificeerd is om een oplossing te bedenken. Een goede projectmanager kent de organisatie, en niet alleen “zijn” projectorganisatie. Hij kent de mensen, en hun vaardigheden en specialismen. En hun telefoonnummers. Om die kennis op te doen, zou een externe projectmanager best wat tijd mogen krijgen. En opdrachtgevers zouden blij moeten zijn met een projectmanager die ook het terrein buiten zijn projectorganisatie wenst te verkennen. Helaas wordt dat door de opdrachtgever vaak gezien als nodeloze tijdverspilling (focus op het project!) en daarmee als een te vermijden kostenpost. Niets is minder waar. Maar dan gaan we wel uit van een projectmanager die begrijpt waarom hij het doet, en niet om iemand die als een toerist de omgeving wil verkennen.

## Tot besluit

In het begin hebben we gezien dat een gemiddeld ERP-project een doorlooptijd heeft van ruim 18 maanden. Mijn pleidooi is die 18 maanden zó te gebruiken dat het ERP-traject een succes wordt. Ik denk ook niet dat het korter kan. En misschien moeten we die verwachting ook maar vanaf het begin communiceren met opdrachtgevers. Die 18 maanden moeten we nuttig gebruiken. Ik zou daarom het volgende tijdsplan willen voorstellen:

- 6 maanden: Doel formuleren en verwachtingen expliciet maken in een proces van wederzijdse aanpassing.
- 6 maanden: Ontwerpen en detailleren. Goedkeuren van het ontwerp en benoemen van beheerders.
- 6 maanden: Inrichten en implementeren van het ERP-systeem.

Afhankelijk van de complexiteit en van de mate waarin een organisatie al “volwassen” is, kan de doorlooptijd korter of langer zijn. Maar het verdient aanbeveling om de 1/3, 1/3, 1/3 verhouding te respecteren. En realiseer u dat het ERP-traject pas na 2/3 van de doorlooptijd actueel zal worden. En dan krijgen de leveranciers van ERP-software tóch gelijk: “het kán in 6 maanden”.

*‘All organizations are perfectly aligned to get the results they get’ (Stephen R. Covey)*