

Het succesvol implementeren van een standaard softwaresysteem

Bachelorthesis

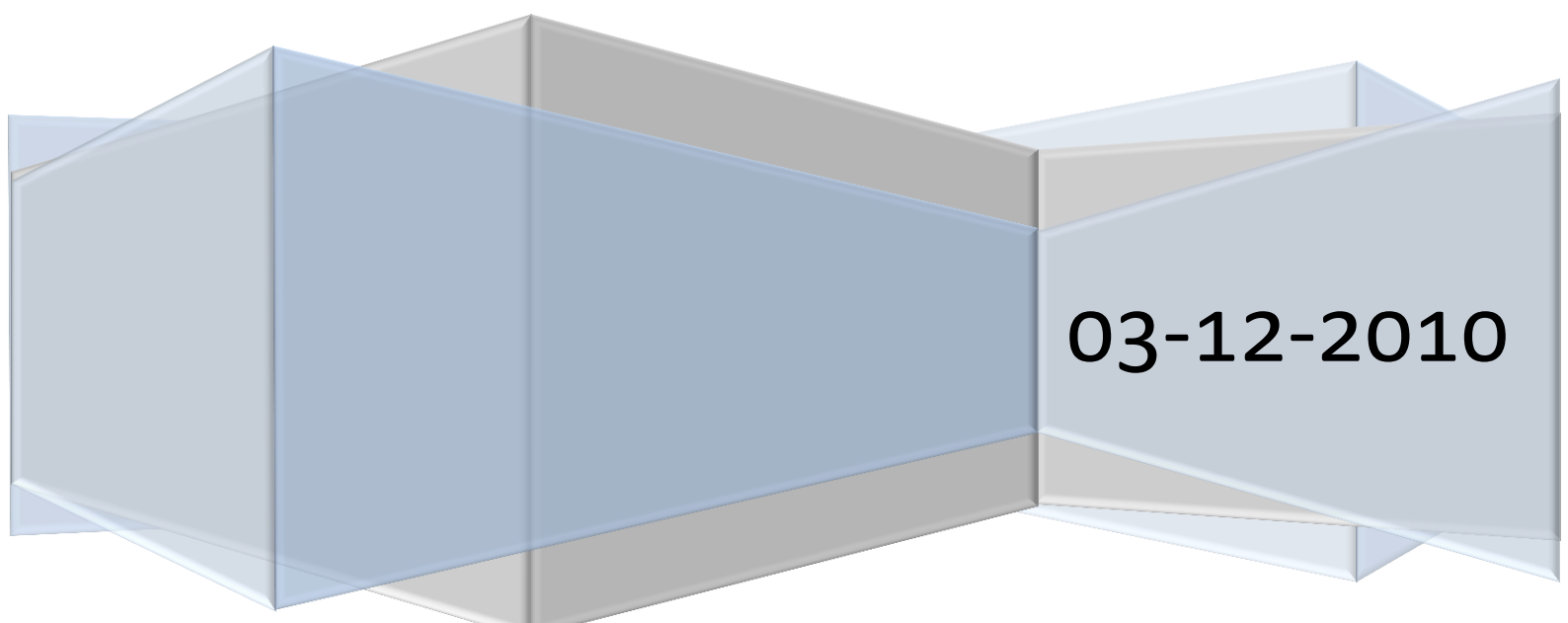
J.N. Zwikstra - 265948

Economie & Bedrijfseconomie

Erasmus Universiteit Rotterdam

Begeleider: prof. dr. G.J. van der Pijl

Meelezer: D. Vandic



03-12-2010

Inhoudsopgave

1.	Onderzoeksopzet.....	3
1.1	Inleiding	3
1.2	Onderzoeksvraag.....	3
1.3	Methodiek	4
1.4	Structuur.....	5
2.	Wat er gebeurt vóór een project wordt gestart.....	6
2.1	Gaan we investeren?.....	6
2.2	Welk project?	6
2.3	Het Plan van Aanpak.....	7
2.4	Scope	8
2.5	Eisen en Wensen	8
2.6	Pakketselectie: Welk systeem en bij wie?.....	10
3.	De implementatie van een project.....	12
3.1	Concieve	12
3.2	Design / Build.....	13
3.3	Develop.....	13
3.4	Deploy.....	13
3.5	Evaluate / Support.....	14
4.	Risico's en kritische succesfactoren tijdens een project	15
4.1	Risicoanalyse	15
4.2	Projectmanagement.....	16
4.3	Kritische Succesfactoren.....	18
4.3.1	Projectplan en visie	18
4.3.2	Change Management	18
4.3.3	Communicatie	19
4.3.4	Samenstelling van skills en personen in projeccteam.....	19
4.3.5	Ondersteuning vanuit het Senior Management en 'championship'	19
4.3.6	Projectmanagement.....	19
4.3.7	Systeem analyse, selectie en technische implementatie.....	20
5.	Praktijkonderzoek.....	21
5.1	Onderzoeksmethode.....	21
5.2	Resultaten.....	21

5.2.1	Scores	22
5.2.2	Analyse resultaten	24
6.	Discussie / Conclusie	28
	Nawoord	31
	Bibliografie	32
	Bijlage 1: Enquête	34

1. Onderzoeksopzet

1.1 Inleiding

Jaarlijks worden er vele miljoenen euro's geïnvesteerd in nieuwe ICT-projecten of nog lopende projecten. Er mag tegenwoordig wel verondersteld worden dat ICT een onmisbare factor is voor de bedrijfsvoering van de meeste bedrijven, zeker bij de meer geavanceerde en snelgroeiende bedrijven [1]. Aan de andere kant tonen onderzoeken echter aan dat de meeste topmanagers software als een groot zorgpunt beschouwen [2], zeker in tijden waarin het economisch minder goed gaat. Een deel van de lopende projecten wordt wel succesvol afgerond, maar lang niet alle projecten worden opgeleverd volgens de vooraf gestelde specificaties [2], [3], [4].

Redenen waarom een project niet succesvol afgerond wordt, kunnen zijn dat een project te laat, of erger nog, helemaal nooit afgerond wordt vanwege onderschatte of onverwachte complexiteit, teveel geld kost, uiteindelijk niet de gewenste functionaliteit bezit, enzovoorts. Ook de Nederlandse overheid worstelt met dit probleem: in november 2007 werd bijvoorbeeld een rapport van de Algemene Rekenkamer gepubliceerd in opdracht van de Tweede Kamer, waarin wordt onderzocht wat er waar is van berichten in de media dat er jaarlijks voor tussen de €4 en €5 miljard door de overheid wordt verspild [5]. Dit onderzoek gaat in op de vraag hoe projecten waarin een standaard software systeem wordt geïmplementeerd, succesvol afgerond kunnen worden.

1.2 Onderzoeksvraag

In dit onderzoek zal voornamelijk worden gekeken wat er in bestaande literatuur geschreven is over het uitvoeren van (ICT) projecten in het algemeen, met de nadruk op het implementeren van een standaard software systeem, zoals een *Enterprise Resource Planning* (ERP)-, *Customer Relationship Management* (CRM)- of *Human Resource Management* (HRM) systeem. Het gaat hierbij ondermeer om de verschillende fasen in een project, (scoping, implementatie e.d.), deliverables en mijlpalen. Grote projecten van de overheid vallen hier in het algemeen niet onder, omdat het daar meestal gaat om gespecialiseerde maatwerk toepassingen waar standaard systemen niet geschikt voor zijn.

De vraag die in dit onderzoek centraal zal staan is:

“Hoe moet de implementatie van een standaard software systeem ideaal gezien verlopen?”

Deelvragen hierbij zijn:

“Welke risicofactoren spelen hierbij een rol?”

“Welke kritische succesfactoren spelen hierbij een rol?”

Onder een standaard software systeem wordt verstaan een softwarepakket met standaard functionaliteit, dat een duidelijk gedefinieerd doel dient en alleen nog geconfigureerd hoeft te worden door de afnemende partij. Een deel maatwerk is hierbij in de regel nog wel nodig, bijvoorbeeld voor een koppeling met een ander systeem of het toevoegen van bedrijfsspecifieke functionaliteit. Zo heeft een CRM pakket als doel om relatiebeheer te faciliteren en zal daarom standaard al de functionaliteit bieden waarmee relatie- en persoonsgegevens bijgehouden kunnen worden.

Onder ‘ideaal verlopen’ wordt bedoeld dat het systeem wordt opgeleverd volgens de klassieke factoren die het succes van een project aanduiden:

- *Op tijd*
- *Binnen budget*
- *Voldoen aan de vooraf gestelde specificaties*

1.3 Methodiek

Dit onderzoek bestaat uit twee delen, een literatuuronderzoek en een praktijkonderzoek. In het literatuuronderzoek wordt de onderzoeksvraag bekeken vanuit een theoretisch kader, dit legt de noodzakelijke basis voor het kunnen uitvoeren van het praktijkonderzoek. Om een beeld te vormen van of, en hoe, deze theorieën ook in de praktijk bekend zijn en/of gebruikt worden, is een praktijkonderzoek gedaan. Hierin hebben ongeveer dertig personen een enquête ingevuld met vragen over de theorie die in dit onderzoek aan bod is gekomen. Deze personen zijn betrokken (geweest) bij een project waarin een standaard software systeem is geïmplementeerd. De resultaten van de enquête worden gepresenteerd in een tabel, waarna de uitkomst van de enquête wordt besproken.

1.4 Structuur

Het document is als volgt opgebouwd: In het volgende hoofdstuk wordt de fase besproken die vooraf gaat aan het daadwerkelijk starten van de implementatie van een standaard software systeem. Daarbij komen onder andere zaken als, Plan van Aanpak, scope en pakketkeuze aan bod. Daarna wordt de implementatiefase besproken in hoofdstuk 3. Hoofdstuk 4 behandelt de risico's en kritische succesfactoren waarmee je tijdens het project mee te maken kan krijgen, dit is het laatste theoriehoofdstuk. Hierna volgt een verslag van het praktijkonderzoek dat is uitgevoerd en tenslotte wordt het document afgesloten met discussie en conclusie.

2. Wat er gebeurt vóór een project wordt gestart

Voordat daadwerkelijk begonnen gaat worden met een nieuw project voor het implementeren van een standaard softwaresysteem, is al een hoop voorbereidend werk verricht en zijn er belangrijke keuzes gemaakt [6]. Dit hoofdstuk beschrijft deze gebeurtenissen.

2.1 Gaan we investeren?

De vraag of er in de eerste plaats wel geïnvesteerd moet gaan worden in een nieuw project, en zo ja, welk project, is er een die ieder bedrijf op zeker moment wel bezig houdt. Hoewel dit interessante problemen zijn, zal dit onderzoek hier verder niet dieper op ingaan, omdat die vragen omvangrijk genoeg zijn voor een geheel eigen onderzoek. Het starten van de implementatie van een standaard software systeem komt vaak voort uit een bepaalde behoefte, bijvoorbeeld het aanschaffen van een CRM systeem om beter op de behoeften van de klanten in te kunnen spelen, of een HRM systeem om het personeelsbeleid meer te structureren.

2.2 Welk project?

In theorie zou ieder project met een verwachte opbrengst, of beter gezegd, Netto Contante Waarde groter dan nul, uitgevoerd kunnen worden. Het uitvoeren van het project levert dan namelijk geld op voor het bedrijf. In de praktijk is dit echter niet mogelijk vanwege beperkte middelen (tijd, budget e.d.). Als is besloten dat een nieuw project uitgevoerd kan gaan worden, is de volgende vraag welk project dat gaat worden, er zal dus een keuze gemaakt moeten worden uit de verschillende investeringsmogelijkheden. Hiervoor bestaan verschillende methoden, bijvoorbeeld het selecteren van een project op basis van gangbare traditionele financiële maatstaven, als *Return on Investment (ROI)* of *Net Present Value (NPV, of Netto Contante Waarde, NCW)*. Bij deze methoden wordt gekeken welk project de hoogste opbrengst heeft volgens de gekozen maatstaf en op basis daarvan wordt bepaald welk project uitgevoerd gaat worden.

Er bestaan ook andere methoden die zich specifiek op ICT projecten richten, zoals *Information Economics* [7], hiermee wordt voor ieder project een lijst met een aantal factoren opgesteld, die negatief of positief meetellen. Ook wordt er onderscheid gemaakt tussen belangrijke en minder belangrijke factoren door het toekennen van een waarde, waardoor uiteindelijk een *gewogen gemiddelde* ontstaat. Dat gewogen gemiddelde is de

score voor het desbetreffende project, het project met de hoogste score is dan de beste keuze.

In het geval van het standaard software systeem waar dit onderzoek over gaat, is de vraag *of* er geïnvesteerd gaat worden, en zo ja, in welk project, al beantwoord. Wat nog wel open staat, is de vraag *welk* softwarepakket gebruikt gaat worden. Van groot belang hierbij is de pakketselectie, dit wordt besproken in sectie 2.6.

2.3 Het Plan van Aanpak

Nadat bekend is welk project uitgevoerd gaat worden, is de eerstvolgende belangrijke stap het opstellen van een Plan van Aanpak (PvA) [6]. In dit document komen de volgende zaken te staan:

1. *doelstelling*: Dit is een soort inleiding over het hoe en waarom van het project. Wat zijn de doelstellingen van het nieuwe systeem? Met andere woorden, wat verwacht men te bereiken en wat is de reden voor aanschaf? Hiermee wordt in het kort uitgelegd waarom dit project uitgevoerd gaat worden.
2. *scope*: Wat wordt de omvang van het project? Worden er huidige systemen vervangen door het nieuwe systeem en zo ja, welke? Welke bedrijfsprocessen gaan wel, en welke niet ondersteund worden? Het is belangrijk om een juiste scope vast te stellen. Als de scope te groot wordt, kan het budget voor het project niet meer toereikend zijn, terwijl bij een te kleine scope wellicht kansen voor verbetering blijven liggen.
3. *activiteiten*: Om welke activiteiten gaat het en hoe worden ze uitgevoerd? Er moet een eisen- en wensenlijst worden opgesteld aan de hand waarvan er pakketselectie kan worden toegepast.
4. *betrokkenen*: Wie wordt de projectmanager? Hoe groot wordt het projectteam, wat wordt ieders taak binnen het team? Welke andere stakeholders zijn er? Denk aan leveranciers, klanten, het senior management, enzovoorts.
5. *Tijd en capaciteit*: Hoe lang gaat het project naar verwachting duren, hoeveel capaciteit wordt ervoor ingezet? Wat gaat het kosten?

Het is erg belangrijk dat bij het opstellen van het Plan van Aanpak in voldoende detail wordt getreden en dat er genoeg tijd wordt genomen om na te denken over wat precies het doel is dat men wil bereiken met het project. Gebeurt dit niet voldoende, bestaat het gevaar dat het project al vanaf het begin de verkeerde weg inslaat en zodoende gedoemd is te mislukken.

2.4 Scope

Het bepalen van de scope houdt in dat wordt vastgesteld wat er wel, en wat niet in het project gedaan gaat worden. In het scope document zouden de volgende zaken moeten staan:

1. Zijn er processen in een huidig systeem die moeten worden meegenomen naar het nieuwe? Zijn er processen die niet meegenomen (kunnen) worden? Hoe is die keuze tot stand gekomen?
2. Wat zijn de knelpunten in de huidige situatie? Deze punten zouden verholpen moeten zijn met het nieuwe systeem en verdienen daardoor extra aandacht.
3. Wat zijn nu de onderdelen die juist goed lopen? Om er in het oog van de gebruikers niet op achteruit te gaan, is het belangrijk dat deze onderdelen ook in het nieuwe systeem zijn terug te vinden. Dit zal in de praktijk niet in alle gevallen haalbaar zijn.
4. Wat gaat er ondersteund worden door het nieuwe systeem (en wat niet)?

Het gaat bij deze punten niet alleen om software, ook de hardware en de organisatie rond het huidige en nieuwe systeem als geheel moeten aan bod komen [6].

2.5 Eisen en Wensen

Een ander belangrijk onderdeel in de voorbereidingsfase van het project is het opstellen van een document met een lijst van eisen en wensen. Ten eerste is het van groot belang dat het duidelijk is wat precies de eisen zijn aan het project, daarnaast verschaft het maken van zo'n lijst ook een goed inzicht in de bedrijfsprocessen, omdat het noodzakelijk is om kritisch naar die processen te kijken om een juiste lijst te krijgen. De wensen zijn onderdelen die men graag in het nieuwe systeem zou zien, maar niet noodzakelijk zijn [6].

Deze eisen en wensen zijn op te delen in de volgende gebieden:

- *Software*: Het gaat hierbij om zaken als geëiste en gewenste functionaliteit van het systeem, workflows, eventuele koppelingen met andere systemen en import van huidige data in het nieuwe systeem. Wat belangrijk is bij het opstellen van de lijst, is dat er in voldoende detail wordt getreden. Juist de details maken vaak het verschil tussen verschillende softwaresystemen. Ook kan hier worden aangegeven hoe er moet worden omgegaan met eventueel benodigd maatwerk, moet dat tot een minimum beperkt worden of niet, wat gebeurt er met het maatwerk bij een eventuele nieuwe release van het systeem e.d. Daarnaast kunnen er nog overige eisen en wensen zijn, bijvoorbeeld of er gekozen moet worden voor een systeem dat zich al heeft bewezen in de praktijk ("*bewezen technologie*").
- *Hardware en overige techniek*: Hieronder vallen hardware, besturingssysteem, client/server systemen en dergelijke. Wat duidelijk moet zijn, is of het nieuwe systeem op bestaande hardware kan gaan draaien of dat er nieuwe hardware aangeschaft moet worden, of dat het zelfs niet in eigen huis gaat draaien, maar op huurbasis wordt afgenomen via internet ASP applicaties, ook wel "Software as a Service", SaaS genoemd. ASP is de afgelopen jaren steeds meer in opkomst gekomen, mogelijk gemaakt door steeds snellere internetverbindingen waardoor het niet meer uitmaakt op welke fysieke locatie de software draait of data wordt opgeslagen. Deze software draait op servers van de leverancier in een datacentrum, zoals bijvoorbeeld een KPN CyberCenter¹. ASP betekent dus dat een bedrijf niet een licentie op een softwarepakket koopt en vervolgens het zelf installeert en onderhoudt, maar dat het de software als een dienst afneemt op huurbasis. Het grote voordeel hiervan is dat er geen hardware ingekocht hoeft te worden en dat installatie en onderhoud door de leverancier geschiedt. Dit kan dus een hoop kosten besparen tijdens de implementatie van een project omdat er geen hardware hoeft worden aangeschaft en bovendien neemt het een stuk complexiteit weg omdat het beheer van de servers uit handen is gegeven. Een nadeel is dat externe opslagruimte relatief duur is vergeleken met lokale opslag.

¹ <http://www.kpn.com/zakelijk/meer-diensten/cybercenters/cybercenter-services.htm>

Eventueel kan later altijd nog besloten worden om de hard- en software in eigen huis te gaan beheren.

Het besturingssysteem waar het systeem op moet kunnen draaien is uiteraard ook van belang, het moet aansluiten op eventuele andere in gebruik zijnde systemen en client-pc's van werknemers en er moet kennis van het besturingssysteem in huis zijn als de software in eigen beheer wordt genomen.

- *Leverancier:* Dit is de partij die het uiteindelijke systeem zal gaan leveren en zal meehelpen met de implementatie. Omdat dit meestal een traject is dat meerdere maanden of zelfs een paar jaar duurt, kan het een goed idee zijn om eventuele eisen en wensen met betrekking tot de leverancier op te nemen. Bijvoorbeeld de grootte van de leverancier, branche- / domeinkennis, expertiseniveau, of internationaal opererend, dat een voordeel kan zijn bij een onderneming met vestigingen in meerdere landen. Als de leverancier tevens optreedt als implementatiepartner, is het ook van belang dat deze 'past' bij de organisatie. De implementatiepartner moet zich namelijk goed in kunnen leven in de situatie van het bedrijf om het systeem op de juiste wijze in te kunnen richten.

2.6 Pakketselectie: Welk systeem en bij wie?

De laatste stap in het proces is het kiezen van een leverancier. In de regel zijn er altijd meerdere pakketten mogelijk die aan de gestelde eisen en wensen voldoen en verschillende leveranciers die hetzelfde pakket aanbieden, dus is het belangrijk om alle opties uit te werken om zo tot de beste keuze te komen. Vaak kan de leverancier van een pakket ook meehelpen bij de implementatie van het systeem door het leveren van consultants en eventueel benodigde hardware, of het systeem aanbieden op huurbasis. Het voordeel hiervan is dat alle zaken met maar één partij afgehandeld hoeven te worden. Er kan ook gekozen worden voor een partij die is gespecialiseerd in het betreffende softwarepakket, zo'n partij heeft veel expertise in huis over de software en 'best practices' op implementatie gebied.

De gebruikelijke methode om tot een keuze voor een bepaalde aanbieder te komen, is het samenstellen van een *long list*, met daarop maximaal tien verschillende aanbieders. Deze

lijst wordt opgesteld door aan de hand van de eisen- en wensenlijst die aanbieders te kiezen die de meeste overeenkomsten hebben met de punten uit de lijst. Uit de long list wordt vervolgens een *short list* opgemaakt, met maximaal twee à drie aanbieders, die de beste 'fit' met de onderneming hebben, zodat er op het gebied van software zo weinig mogelijk maatwerk nodig is en zo weinig mogelijk overbodige functionaliteit wordt ingekocht en dat naar verwachting de aanbieder zich zo goed mogelijk kan inleven in de situatie van de onderneming. Voor overheidsinstanties geldt bovendien nog dat bij projecten boven een bepaalde drempelwaarde, deze Europees moeten worden aanbesteed, dit naar aanleiding van Europese richtlijnen (2004/17/EG [8] en 2004/18/EG [9]) om concurrentie te bevorderen.

De short list fase is een belangrijke fase, want hierin wordt uiteindelijk de beslissing genomen welk pakket gebruikt gaat worden. De aanbieders die nu nog op de short list staan, wordt om een uitgebreide offerte en demonstratie van hun systeem gevraagd. Aan de hand van deze demonstraties moet nu een goed onderbouwde beslissing gemaakt kunnen worden welke aanbieder de uiteindelijke voorkeur heeft, en kan het implementatietraject gestart worden.

3. De implementatie van een project

Het uitvoeren van een project wordt in de literatuur in het algemeen ingedeeld in vier of vijf fasen [6], [10]. De indeling kan dus soms enigszins verschillen, net als de namen die aan de fasen worden gegeven, maar de punten die aan bod komen zijn in de regel wel gelijk. Hier wordt uitgegaan van vijf fasen:

1. *Conceive (initiatie)*
2. *Design/Build (ontwerp)*
3. *Develop (ontwikkelen)*
4. *Deploy (uitrollen)*
5. *Evaluate/Support (evaluatie/nazorg)*

Iedere fase heeft zijn eigen karakteristieken en 'deliverables' en wordt bij voorkeur formeel afgesloten en afgetekend: een 'mijlpaal' in het project. Een deliverable is meestal een document dat aan het eind van de fase moet worden opgeleverd, bijvoorbeeld het functioneel ontwerp voor benodigd maatwerk, of het draaiboek voor tijdens de uitrolfase. Deze documenten moeten officieel door de projectmanager en eventuele andere belanghebbenden (bijvoorbeeld een externe derde partij) worden goedgekeurd. In dit hoofdstuk worden de bovenstaande fasen stuk voor stuk besproken.

3.1 Conceive

In de eerste fase vindt de start van het project plaats. Een deel van het werk dat is uitgevoerd voor de eigenlijke start kan in deze fase worden hergebruikt. Bijvoorbeeld de indeling van het projectteam, het Plan van Aanpak e.d. In deze fase wordt het plan voor de verdere implementatie uitgewerkt, zodat het duidelijk is wat er allemaal gaat veranderen op het gebied van bedrijfsprocessen, mensen, technologie en organisatie, en wat er nodig is om dat te realiseren. Consultants van de implementatiepartner kunnen erg behulpzaam, of zelfs onmisbaar zijn bij het (helpen) doorgronden van de bedrijfsprocessen. Deze fase wordt afgerond met de oplevering van het definitieve implementatieplan, dat tijdens de rest van de implementatie als leidraad zal worden gebruikt.

3.2 Design / Build

In de ontwerpfase wordt een detailplan (functioneel ontwerp) gemaakt over hoe het systeem moet worden ingericht. Ook functionele ontwerpen voor maatwerkapplicaties, zoals rapportage, koppelingen met andere systemen of andere specifieke applicaties, worden in deze fase opgesteld. Op basis van de functionele ontwerpen kunnen ontwikkelaars een technisch ontwerp opstellen voor het te ontwikkelen maatwerk. Aan het eind van deze fase zijn er documenten opgeleverd die in de ontwikkelfase gebruikt worden als handleiding voor de inrichting van het systeem.

3.3 Develop

In deze fase vindt de daadwerkelijke inrichting en configuratie van het systeem plaats, deze fase wordt soms ook samen met de Design/Build fase genomen. Het maatwerk wordt nu ontwikkeld en getest en procedures voor de import en conversie van data uit oude systemen worden opgesteld. Ook wordt het systeem ingericht en geconfigureerd zodat het past bij de specifieke situatie van het bedrijf. Vervolgens wordt het systeem als geheel getest, bij voorkeur in een testomgeving met bestaande data uit huidige systemen. In deze fase vinden ook trainingen plaats voor de gebruikers, zodat zij straks kunnen omgaan met het nieuwe systeem.

Aan het eind van deze fase is het systeem gereed om in productie te worden genomen.

3.4 Deploy

Het systeem is nu klaar om daadwerkelijk in gebruik genomen te worden. In de vorige fasen is het systeem helemaal ingericht en maatwerk ontwikkeld, er is getest en de nieuwe gebruikers zijn getraind om het systeem te kunnen gebruiken. Deze fase wordt afgesloten met het moment waar tijdens de hele implementatiefase naartoe is gewerkt: de “Go Live” van het project, als het project goed is verlopen meestal een feestelijke gebeurtenis. Data uit oude systemen wordt nu overgezet naar de live omgeving van het nieuwe systeem en getest op integriteit. Bij een erg ingrijpende verandering als het in gebruik nemen van een ERP systeem, wordt hiervoor vaak een datum gekozen waarop het rustig is in de onderneming, bijvoorbeeld op 1 januari als de meeste medewerkers een dag vrij hebben en er weinig bedrijfsactiviteit is.

3.5 Evaluate / Support

Nadat het systeem in gebruik is genomen volgt een evaluatie over het eindresultaat versus het beoogde resultaat bij aanvang van het project. Ook zijn er altijd problemen en foutjes die in de eerste weken na de Go Live boven water komen, hier kan de partij die bij de implementatie betrokken was bijvoorbeeld de eerste zes weken nog bij helpen en tijdens die periode geleidelijk de verantwoordelijkheid voor het beheer en het oplossen van problemen aan de onderneming overdragen. Deze laatste fase en daarmee het hele project wordt afgesloten met het tekenen van de formele overdracht van het systeem aan de organisatie. Meestal blijft de implementatiepartner wel support leveren in de vorm van een supportcontract om problemen op te lossen waar het eigen beheer in de organisatie niet uit komt.

4. Risico's en kritische succesfactoren tijdens een project

Er zijn een hoop dingen die kunnen gebeuren tijdens de uitvoer van een project die de mate van succes van een project nadelig beïnvloeden, of zelfs volledig laten mislukken. In dit hoofdstuk worden deze risico's in kaart gebracht, samen met mogelijke oplossingen om deze tegen te gaan. Wat hiermee samenhangt, zijn de zogenaamde kritische succesfactoren: factoren die van kritisch belang zijn voor het slagen van een project en dus extra aandacht vereisen.

4.1 Risicoanalyse

Sleutelwoorden bij risico zijn complexiteit en onzekerheid. Hoe groter de complexiteit, hoe moeilijker het is om overzicht te houden en alle risico's te onderscheiden. Risico en onzekerheid zijn begrippen die op het eerste gezicht bijna hetzelfde lijken, maar er zit toch enig verschil in. Risico kan namelijk omschreven worden als een gebeurtenis die nadelig voor is voor het project en zich *kan* voordoen. De kans dat die gebeurtenis optreedt kan bekend zijn, worden geschat, of helemaal onbekend zijn. Het verschil met onzekerheid zit hem in het feit dat een risicofactor voor aanvang van een project al bekend is, terwijl een onzekere gebeurtenis niet van tevoren bekend is. Met risico kan dus rekening worden gehouden in de planning, met onzekerheid niet. Een onzekere gebeurtenis hoeft overigens niet altijd een nadelig effect te hebben, er kan ook een kans gecreëerd worden die positief uitvalt voor het project [11].

Looijen stelt dat complexiteit door maar liefst negen factoren bepaald wordt: hoeveelheid, heterogeniteit, spreiding, dynamiek, functionaliteit, samenhang, geavanceerd, eigenaarschap en gebruik [12]. Deze factoren legt hij op de volgende manier uit:

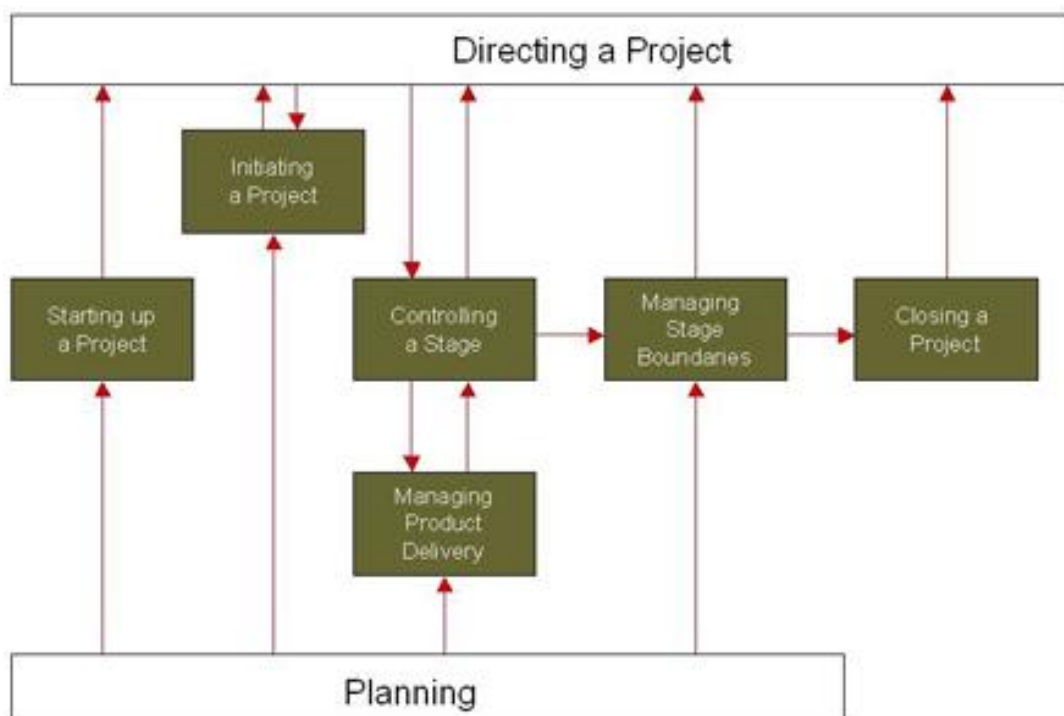
“Integratie van een informatiesysteem, een technische infrastructuur en een gegevensinfrastructuur kan grote aantallen (*hoeveelheid*), automatiseringsmiddelen omvatten die gekenmerkt worden door velerlei typen (*heterogeniteit*). De middelen kunnen in hoge mate gedeconcentreerd zijn (*spreiding*) en regelmatig onderhevig zijn aan veranderingen (*dynamiek*). Voorts kan het gaan om een grote verscheidenheid aan functies (*functionaliteit*) die vanuit tal van aan elkaar gekoppelde componenten (*samenhang*) wordt geleverd. Deze componenten kunnen buitengewoon gecompliceerd zijn (*geavanceerd*). Tenslotte kunnen de automatiseringsmiddelen aan verschillende eigenaars toebehoren

(eigenaarschap) en de gebruikers kunnen zeer uiteenlopende eisen en randvoorwaarden stellen (gebruik).”

Gedurende de hele implementatie moet dus voldoende aandacht worden besteed aan risico's, zodat er tijdig ingegrepen kan worden, mocht er een probleem op dreigen te treden. Adequaat projectmanagement kan hiervoor zorgen.

4.2 Projectmanagement

Voor ICT-projecten is de bekendste methode voor projectmanagement waarschijnlijk Prince2 (PProjects IN Controlled Environments) [13]. Prince2 is de opvolger van Prince, dat in 1989 is ontwikkeld in het Verenigd Koninkrijk als methode voor ICT-project management, en is uitgegroeid tot de *de facto* standaard voor project management. De methode bestaat uit procedures en richtlijnen die een onderneming gestructureerd door de fasen van een project moeten leiden.

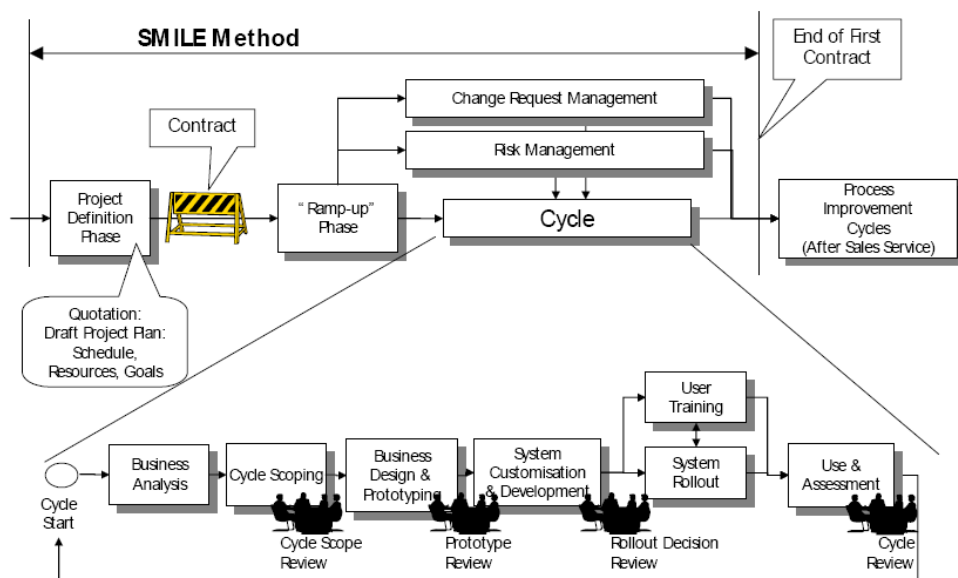


Figuur 1: Het Prince2 proces model

De methode probeert de implementatie beheersbaar te houden door het project op te delen in verschillende fasen en processen, zodat deze beheersbaar blijven door het projectmanagement. Deze fasen en processen komen in principe overeen met de project

fasen die hier al zijn besproken. Door het project als het ware in hapklare brokken op te delen wordt de complexiteit verminderd waardoor risico's beheersbaar blijven. Een nadeel van deze methode, die vooral bij kleinere projecten zichtbaar wordt, is de extra overhead die aan het project wordt toegevoegd, bijvoorbeeld omdat alle stappen gedocumenteerd dienen te worden.

Een methode die zich specifiek richt op CRM implementaties is de SMILE methode, opgesteld in 2002/2003 door het Fraunhofer Instituut in opdracht van de Europese Unie [14]. SMILE staat voor “SME oriented Method for successful CRM Implementation with Low Effort”, waarbij SME weer staat voor “Small- and Medium sized Enterprises”.



Figuur 2: De SMILE methode

In de SMILE methode staan de volgende kenmerken centraal:

- Een iteratieve manier van werken (bekend als het “spiraal model” bij systeem ontwikkeling ([15]).
- Verschillende cycli waar doorheen wordt gelopen. Iedere fase heeft zijn eigen specifieke cycli, deze staan niet vast, maar worden bepaald op basis van de specifieke omstandigheden van de onderneming.
- Hoge mate van betrokkenheid van gebruikers tijdens de cycli

Parallel aan de cycli loopt het normale project- en change management [14].

4.3 Kritische Succesfactoren

Kritische succesfactoren zijn factoren die van groot (kritisch) belang zijn voor het slagen van een project, deze factoren verdienen daarom in het bijzonder aandacht tijdens de implementatie [4]. Kritische succesfactoren zijn in weze risicofactoren, hoewel een risicofactor niet perse een kritische succesfactor hoeft te zijn. De kritische succesfactoren kunnen gezien worden als de belangrijkste risicofactoren. In een artikel over kritische succesfactoren bij ERP implementaties [16] worden zeven verschillende gebieden onderscheiden waarin deze factoren verdeeld kunnen worden:

1. *projectplan en visie*
2. *change management*
3. *communicatie*
4. *samenstelling van skills en personen in projectteam*
5. *ondersteuning vanuit het management en 'championship'*
6. *projectmanagement*
7. *systeem analyse, selectie en technische implementatie*

Deze factoren zullen in de volgende subsecties één-voor-één nader worden toegelicht.

4.3.1 Projectplan en visie

Het is erg belangrijk voor een project dat er een heldere visie, doelstellingen en plan bestaan. Een heldere visie en missie moeten een soort richtlijn zijn tijdens de implementatie, daarnaast moeten er ook meetbare doelen worden gespecificeerd. De doelstellingen van het project moeten duidelijk zijn en door het projectteam begrepen, zodat het project in goede banen geleid kan worden.

4.3.2 Change Management

Change management houdt in dat de veranderingen die het nieuwe systeem brengt in goede banen geleid worden. Dit betekent ondermeer dat de gebruikers training moeten krijgen om met het nieuwe systeem te werken en dat gebruikers betrokken moeten worden bij het project en feedback kunnen leveren om zo tijdig eventuele bugs of knelpunten die zij aangeven te verhelpen. Dit vergemakkelijkt de acceptatie van het nieuwe systeem door de gebruikers.

4.3.3 Communicatie

Gedurende het project moet er steeds een goede communicatie zijn tussen alle betrokken personen, zodat men steeds op de hoogte is van de laatste ontwikkelingen. Het is dus belangrijk dat er zeer regelmatig overleg plaats vindt tussen de betrokken personen.

Ook moet feedback afkomstig van gebruikers zichtbaar worden verwerkt, zodat zij kunnen zien dat er daadwerkelijk wat met hun opmerkingen gedaan wordt.

4.3.4 Samenstelling van skills en personen in projectteam

De samenstelling van het projectteam moet een afspiegeling zijn van de gebieden die door het nieuwe systeem gedekt worden, met business- en technische experts uit het bedrijf, samen met gewone gebruikers, en daarnaast bestaan uit business consultants en functionele consultants afkomstig van de leverancier, met grondige kennis van het pakket. Verder moet het team een gebalanceerd geheel zijn van zowel de interne mensen, als de consultants van de verkopende partij. Dit gebied wordt door de auteurs van het artikel na onderzoek in twee organisaties als meest belangrijk aangewezen [16].

4.3.5 Ondersteuning vanuit het Senior Management en 'championship'

Een positieve instelling en steun vanuit het hogere management is cruciaal voor het succes van ieder project. Er moeten voldoende middelen: tijd, geld en mankracht, beschikbaar worden gesteld om het project goed uit te kunnen voeren. Als er niet genoeg commitment is voor het project vanuit het senior management, kan dat bijvoorbeeld problemen opleveren als er het project over het budget heen dreigt te gaan.

Een 'kampioen' is iemand die ervoor zorgt dat het project blijft leven onder de mensen die ermee bezig zijn, hij is de drijvende kracht achter het geheel en moet de deelnemers motiveren met zijn enthousiasme en kan kritiek en weerstand tegen het project weerleggen. Een kampioen kan een belangrijke factor zijn in het slagen van het project.

4.3.6 Projectmanagement

Een goede projectmanager is uiteraard van groot belang bij een project. Deze persoon is verantwoordelijk voor het project en aan te spreken als er zaken fout dreigen te lopen. De projectmanager dient er zorg voor te dragen dat de scope van het project bewaard blijft en dat het project soepel blijft verlopen, het budget en planning bewaken, en moet

weloverwogen beslissingen kunnen nemen als het aankomt op eventuele wijzigingen in het originele plan.

4.3.7 Systeem analyse, selectie en technische implementatie

Als het nieuwe systeem in eigen beheer gebruikt gaat worden, is het erg belangrijk dat de complexiteit die komt kijken bij het integreren van de nieuwe hardware met eventuele bestaande (legacy) systemen goed gemanaged wordt. De systeemarchitectuur zoals deze eruit moet komen te zien na afloop van het project, moet voor aanvang van de implementatie vastgesteld worden, zodat er tijdens het project geen wijzigingen in de architectuur gemaakt hoeven te worden. Grondig testen is ook altijd belangrijk bij het in gebruik nemen van nieuwe hardware. In het geval van een ASP oplossing hoeft er natuurlijk geen nieuwe hardware aangeschaft en geconfigureerd te worden, testen blijft wel erg belangrijk.

Om een nieuw systeem optimaal te kunnen benutten is integratie van de bestaande data in andere systeem van groot belang. Hiervoor kan een onderneming zelf tools ontwikkelen, maar indien mogelijk moet gebruik worden gemaakt van tools en richtlijnen die de leverancier ter beschikking kan stellen. Dit bespaart zowel tijd als geld voor de onderneming.

5. Praktijkonderzoek

In de voorgaande hoofdstukken van deze thesis is de theorie besproken over het implementeren van een standaard software systeem. In dit gedeelte volgt een verslag van het onderzoek naar hoe er in de praktijk gewerkt wordt. Eerst wordt in sectie 5.1 de onderzoeksmethode uiteengezet, waarna in sectie 5.2 de resultaten worden gepresenteerd, geanalyseerd en besproken.

5.1 Onderzoeksmethode

Er is aan ongeveer dertig projectmanagers, afdelingshoofden, software-ontwikkelaars en algemene leden van projectteams een vragenlijst voorgelegd (zie Bijlage 1: Enquête), waarin hun mening wordt gevraagd over de verschillende onderwerpen die in het theoriegedeelte aan bod zijn gekomen. Als referentiemateriaal voor de respondenten is een samenvatting van het theorieonderzoek (de hoofdstukken 2 tot en met 4) meegestuurd, de vragen zijn vervolgens opgesteld op basis van de samenvatting.

Bij de meeste vragen konden punten gegeven worden op een 5-punts Likert schaal (sterk oneens – sterk mee eens), alleen bij de laatste vragen over kritische succesfactoren kon een ‘rapportcijfer’ op een schaal van 1 tot en met 10 worden gegeven. De scores van de eens-oneens vragen zijn omgerekend naar een 10-punts schaal door ze met twee te vermenigvuldigen. Dit is gedaan om beide soorten vragen dezelfde schaal te laten gebruiken. Dit was achteraf gezien een wat ongelukkige keuze, maar niet meer terug te draaien omdat de enquête al was ingevuld.

Bij alle vragen was er ook de mogelijkheid om commentaar en/of uitleg te geven bij het antwoord, deze opmerkingen worden meegenomen in de analyse van de resultaten. Voor iedere vraag is een gemiddelde score berekend, om te kunnen zien in hoeverre de meningen verspreid liggen, is ook de standaardafwijking weergegeven.

5.2 Resultaten

In deze sectie bespreken we de resultaten van de enquête. Eerst wordt de uitkomst verwerkt in een tabel, waarbij de gemiddelde score en standaarddeviatie wordt getoond om de eens/oneens vragen te kwantificeren. Tenslotte wordt Cronbach’s Alpha [17] berekend om de kwaliteit van de resultaten te bepalen. In sectie 5.2.2 wordt de uitkomst van de enquête geanalyseerd en besproken.

5.2.1 Scores

Tabellen 4.1 en 4.2 geven een overzicht van de uitkomsten van de enquête.

\bar{x} s

"Het opstellen van een goed en duidelijk PvA is belangrijk"	9,1	1,0
"Als er een PvA is, moet deze zoveel mogelijk gevolgd worden"	8,1	1,56
"Ik heb een PvA eigenlijk nooit nodig gehad, of noodzakelijk gevonden"	3,2	1,1
"Het bepalen van de juiste scope is belangrijk"	9	1,0
"Het is belangrijk om te inventariseren wat huidige knelpunten zijn, zodat deze in het nieuwe systeem opgelost kunnen worden"	8,9	1,23
"Het is belangrijk om te weten wat de goedlopende punten zijn, zodat deze er na de implementatie nog steeds zijn."	8,3	1,35
"Naast een PvA en bepalen van de scope, is een document met eisen en wensen ook belangrijk"	8,5	1,13
"Het software gedeelte is belangrijk (welk pakket kiezen we?)"	8,4	1,38
"Het hardware gedeelte is belangrijk (welke hardware/ASP?)"	7,6	1,56
"De leverancier van het systeem is belangrijk (bijv. Microsoft, Peoplesoft, Oracle)"	7	1,75
"Er moet ook rekening worden gehouden met de eisen en wensen van de gebruikers"	9,2	1,32
"Het is belangrijk om uitgebreid onderzoek te doen naar welk softwarepakket gekozen moet worden"	7,9	1,79
"De leverancier van het gekozen softwarepakket is belangrijk"(de implementatiepartner)	8,1	1,34
"Ik ben bekend met de verschillende fasen tijdens een implementatie"	8,1	1,39
"Het opstellen van een implementatieplan tijdens de Concieve (Scoping) fase is belangrijk"	8	1,24
"Het opstellen van functionele ontwerpen voor de inrichting van het systeem en te ontwikkelen maatwerk is belangrijk voor een goed ingericht systeem."	8,8	1,5
"Er moet eerst getest worden voor een nieuw systeem in productie wordt genomen."	9,5	0,88
"Trainingen voor gebruikers zijn noodzakelijk voor een vlotte acceptatie en goed gebruik van het nieuwe systeem."	9	1,0
"Ook na de Go Live is ondersteuning van externe consultants, zeker in de eerste twee maanden, van belang."	7,7	2,04

“Het is belangrijk om risicofactoren tijdens een project goed in de gaten te houden”	8,6	0,91
“Ik weet wat de methode PRINCE2 inhoudt”	6,5	2,73
“De PRINCE2 methode is gebruikt bij een project waarbij ik betrokken was of ben” (oneens = nee; eens = ja)*	51,7%	

Tabel 4.1: Overzicht gemiddelde scores ‘eens/oneens’ vragen. \bar{X} is de gemiddelde score, s de standaardafwijking

*Score is percentage ‘ja’

	\bar{X}	s
“Het hebben van een degelijk projectplan en visie is van kritisch belang voor het slagen van het project”	8,1	1,09
“Change management is van kritisch belang voor het slagen van het project”	7,6	1,31
“Communicatie is van kritisch belang voor het slagen van het project”	8,6	0,95
“De samenstelling van skills en personen binnen het projectteam is van kritisch belang voor het slagen van het project”	8,2	0,96
“Ondersteuning vanuit het management en het hebben van een “champion”	8,5	0,89
“Een goede projectmanager is van kritisch belang voor het slagen van het project”	8,2	0,93
“ Een analyse van de nieuwe systeemeisen en bestaande systemen is van kritisch belang voor het slagen van het project”	8	0,84

Tabel 4.2: Overzicht waardering kritische succesfactoren. \bar{X} is de gemiddelde score, s de standaardafwijking

De berekening voor Cronbachs Alpha is als volgt:

$$\alpha = \frac{N}{N - 1} \left(1 - \frac{\sum_{i=1}^N \sigma_{Y_i}^2}{\sigma_X^2} \right)$$

Hierbij is N het aantal items, is $\sigma_{Y_i}^2$ de variantie voor de afzonderlijke itemscores en σ_X^2 de variantie van de totale score X voor alle items van alle respondenten.

Het aantal vragen was 23, maar de vraag “De PRINCE2 methode is gebruikt bij een project waarbij ik betrokken was of ben” is uiteindelijk buiten beschouwing gelaten, omdat deze teveel afwijkt van de andere vragen. Daarom heeft N de waarde van 22. De varianties zijn berekend door alle scores te verwerken, wat resulteert in de volgende waarde van Cronbachs Alpha:

$$\alpha = \frac{22}{22 - 1} \left(1 - \frac{42,438}{103,95} \right) = 0,62$$

Als we de vuistregel hanteren die zegt dat Cronbachs Alpha minimaal 0,70 dient te zijn, moeten we constateren dat de gebruikte enquête helaas niet goed genoeg is opgebouwd om heel betrouwbaar te zijn.

5.2.2 Analyse resultaten

Het blijkt dat een Plan van Aanpak vrijwel unaniem als erg belangrijk en onmisbaar wordt ervaren. Het PvA vormt de basis en is een leidraad/handvat voor de hele verdere implementatieprocedure en definieert mijlpalen en de opleverdata ervan. Door een aantal personen wordt aangegeven dat er nog voor het PvA een Programma van Eisen (PvE) moet zijn, dit document vormt de basis van een Plan van Aanpak. Er moet ook geprobeerd worden om aan het plan vast te houden (*“plan your work and work your plan”*), tenzij er door (onvoorziene) omstandigheden aanpassingen nodig zouden zijn.

Ook het bepalen van de scope vindt de meerderheid erg belangrijk, een project zou niet gestart mogen worden als de scope nog niet is bepaald, of het project is gedoemd te mislukken. Er moet worden bepaald wat wel, maar ook wat niet wordt meegenomen, dit schept duidelijkheid bij de betrokkenen. Hierbij vindt men het belangrijker dat knelpunten worden opgelost, dan dat bestaande onderdelen die goed lopen behouden blijven. Wel wordt aangegeven dat gebruikers eerder het nieuwe systeem aanvaarden als ze de goede punten van een vorig systeem weer terug zien in het nieuwe systeem.

Een document met eisen en wensen wordt ook als zeer belangrijk ervaren, voor zowel de opdrachtgever, als de implementatiepartner. De opdrachtgever kan hiermee tijdens de implementatie controleren of de doelstellingen in het document gevolgd worden, terwijl de implementatiepartner hier een handvat aan heeft voor wat betreft de opdrachtgever noodzakelijk vindt en wat eventueel kan vervallen, mocht het project over het budget dreigen te gaan.

Bij het bepalen van het hoe/wat/waar van het te gebruiken software systeem is de software het belangrijkste, daarna de hardware en als laatst de software leverancier. Met betrekking tot de software is het vooral belangrijk dat deze aansluit bij huidige hardware en/of systemen, zeker als er een koppeling tussen die systemen nodig is. In dat geval kan het nodig zijn dat de leverancier van zo'n systeem ook bij het project betrokken moet worden om de koppeling te realiseren. Tenslotte moet het pakket natuurlijk ook kunnen wat ervan

verwacht wordt en de gebruikers moeten ermee om kunnen en willen gaan. Dit hangt ook samen met hoe gebruiksvriendelijk het pakket is.

Hardware wordt niet als een heel belangrijk punt gezien. Het moet wel aan kunnen sluiten op bestaande hardware, verder is vooral de prijs van belang. Een ASP oplossing is een goed alternatief als er binnen de organisatie geen (voldoende grote) ICT afdeling is.

Bij de keuze voor een bepaalde leverancier van de software (Microsoft, Peoplesoft e.d.), let men vooral op betrouwbaarheid en continuïteit en reeds aanwezige systemen binnen de organisatie. Eén persoon wist het mooi te zeggen: “Het is niet de tent, maar de vent die er toe doet”, ofwel de consultants van een implementatiepartner zijn belangrijker dan de leverancier van het systeem.

Rekening houden met de eisen en wensen van gebruikers wordt door de meeste ondervraagde personen als erg belangrijk gezien. De gebruikers zijn uiteindelijk degenen die met het nieuwe systeem gaan werken, er moet dus voldoende draagvlak zijn bij deze groep. Als het nieuwe systeem niet aansluit op hun wensen zal het nooit effectief gebruikt gaan worden. Er wordt echter ook opgemerkt dat dit voor een deel opgevangen kan worden met bijscholing en ‘de juiste persoon op de juiste plek’, deze kan dan toch de gebruikers stimuleren om met het nieuwe systeem te gaan werken. Ook is het niet vanzelfsprekend dat alle eisen en wensen ook worden meegenomen, dit dient wel goed gecommuniceerd te worden naar de gebruikers.

Over onderzoek doen naar het gewenste softwarepakket zijn de meningen wat meer verdeeld, de meeste personen vinden dit wel belangrijk, zeker omdat je meestal jaren gebruik gaat maken van het pakket. Anderen vinden dit minder belangrijk, als het maar voldoende functionaliteit biedt en binnen het budget valt.

Bij het kiezen van een leverancier van het gekozen pakket, bijna altijd ook de implementatiepartner, is het vooral belangrijk dat er een ‘klik’ is tussen de twee partijen, dat betekent dat de partijen goed met elkaar overweg kunnen op zakelijk, maar vooral op menselijk gebied. (Denk aan de eerder aangehaalde: “Het is niet de tent, maar de vent die er toe doet”) Daarnaast spelen ook de referenties van de leverancier een rol, wat het bedrijf kan bieden en hoe het presteert is ook van belang.

Alle ondervraagde personen waren in meer of mindere mate op de hoogte van de verschillende fasen in een project, dit is voor de opdrachtgever ook belangrijk, deze moet namelijk de regie over het proces in handen houden. Daarvoor is het noodzakelijk om op de hoogte te zijn van het verloop van het project.

Functionele ontwerpen maken voor de inrichting en vooral bij te ontwikkelen maatwerk is ook een belangrijk punt. Zonder een degelijk functioneel ontwerp bestaat het gevaar dat er uiteindelijk iets anders wordt opgeleverd dan de bedoeling was. Ook hier kan het weer nuttig zijn om ook bij gebruikers te rade te gaan.

Het testen van de nieuwe omgeving blijkt het allerbelangrijkst gevonden te worden. Dit testen kan bijvoorbeeld gedaan worden op een ASP omgeving die tijdens de Go Live wordt overgezet naar de productieomgeving. Testen voor de Go Live is daarnaast een stuk makkelijker en goedkoper dan fouten opsporen en oplossen als het systeem al in productie is genomen. Ook is het erg belangrijk dat het systeem van het begin af goed werkt, er is namelijk maar één eerste indruk die de nieuwe gebruikers van het systeem krijgen, als die niet goed is, kan dit nadelige gevolgen voor draagvlak en efficiënt gebruik van het systeem hebben.

Om nieuwe gebruikers om te leren gaan met een nieuw systeem is het noodzakelijk dat er trainingen worden gegeven. Hier zijn wel een aantal nuances in te vinden, bijvoorbeeld hoe complexer het systeem, hoe meer training er nodig is. Het kan zijn dat een systeem zo intuïtief werkt, dat training niet tot nauwelijks nodig is. Ook kunnen er een aantal 'key users' opgeleid worden die de rest van de gebruikers wegwijs kunnen maken in het nieuwe systeem, ("train the trainer") dit kan tijd en geld besparen en daardoor efficiënter zijn.

Wat betreft het nodig hebben van nazorg na afloop van een project lopen de meningen sterk uiteen. Hier volgen een aantal punten die bepalen hoeveel nazorg nodig is:

- De mate van eigen kennis. Als er binnen de eigen organisatie personen aanwezig zijn met kennis van het systeem, bijvoorbeeld opgedaan tijdens de implementatie, zal er minder nazorg nodig zijn.
- De mate waarin het systeem overeenkomt/afwijkt van andere of eerder gebruikte systemen.

- Hoeveelheid maatwerk. Een standaardsysteem zal minder nazorg nodig hebben dan een volledig nieuw gebouwd systeem.
- Gebruik van het systeem in eigen beheer, of als ASP oplossing. Als het systeem in eigen beheer draait, zullen technische mankementen door de eigen IT afdeling worden opgelost, terwijl bij een ASP omgeving de ASP provider hiervoor verantwoordelijk is.

Het in de gaten houden van risico wordt natuurlijk ook erg belangrijk gevonden, er wordt hierbij aangegeven dat deze zoveel mogelijk van tevoren in kaart zijn gebracht (dit is ook onderdeel van de Prince2 methode), inclusief maatregelen die hier tegen genomen kunnen worden. Het is niet de vraag óf er iets fout gaat, maar wanneer dit gebeurt. Hier moet dus van tevoren al rekening mee worden gehouden. De ervaringen die de implementatiepartner heeft opgedaan tijdens eerdere projecten kan een belangrijke rol spelen bij het in kaart brengen van risico's en het tijdig signaleren van problemen.

De bekendheid van de Prince2 methode onder de ondervraagde personen varieert van 'geen idee' tot 'Prince2 gecertificeerd'. Het hangt ook erg af van de rol die iemand speelt binnen het projectteam of hij/zij daar mee te maken heeft (gehad). In precies de helft van alle gevallen is de Prince2 methode gebruikt bij een project. Een veel gehoord bezwaar bij Prince2 is het extra administratieve werk dat bij een project komt kijken.

Tenslotte zijn de zeven kritische succesfactoren gewaardeerd, hierbij wordt "communicatie" het belangrijkste gevonden, goede communicatie zorgt ervoor dat iedereen op de hoogte blijft van de stand van zaken en ook naar de gebruikers toe is communicatie belangrijk, zo blijft het onderwerp ook onder die groep levend, dat komt het draagvlak ten goede. "Ondersteuning vanuit het management en het hebben van een 'kampioen'" komt op de tweede plaats. De kampioen kan een belangrijke rol spelen door het stimuleren van het project en door goed contact te onderhouden met de gebruikers. Zonder steun vanuit het senior management zou een project niet eens gestart moeten worden. De top drie wordt afgesloten met "samenstelling van skills en personen binnen het projectteam", als hier een scheve verdeling is, kan dat nadelige gevolgen hebben voor het slagen van het project. "Change management" wordt als minst belangrijk beschouwd. De scores liggen echter wel vrij dicht bij elkaar, tussen de 7,6 en 8,6, dus heel veel verschil in waardering is er niet.

6. Discussie / Conclusie

In dit onderzoek is geprobeerd om een antwoord te geven op de vraag:

“Hoe moet de implementatie van een standaard software systeem ideaal gezien verlopen?”

Met als deelvragen:

“Welke risicofactoren spelen hierbij een rol?”

“Welke kritische succesfactoren spelen hierbij een rol?”

De volgende theoretische aspecten zijn hierbij aan bod gekomen:

Ten eerste de fase vóór aanvang van een nieuw project, waarin het voorbereidende werk gebeurt: het kiezen van een project, het opstellen van een plan van aanpak en eisen en wensen lijst, het bepalen van de scope en het kiezen van een software en aanbieder.

Ten tweede de verschillende fasen tijdens de implementatie van een nieuw systeem: de Concieve fase waarin het project wordt gestart en de planning en plannen voor de implementatie worden gemaakt, de Design/Build fase waarin concrete plannen en functionele ontwerpen worden gemaakt voor de inrichting van het systeem, de Develop fase waarin het systeem wordt ingericht, maatwerk wordt ontwikkeld en gebruikers worden getraind, de Deploy fase waarin de Go Live plaatsvindt en het systeem in gebruik wordt genomen en als laatste de Evaluate/Support fase waarin de implementatiepartner tijdens de eerste weken na de Go Live helpt bij het oplossen van problemen en daarnaast geleidelijk het beheer overdraagt aan de onderneming.

Tenslotte het omgaan met risico, risicomangement en kritische succesfactoren. Bij risico gaat het voornamelijk om complexiteit en onzekerheid. Als de complexiteit te groot is, is het niet meer goed mogelijk om alles te overzien, dat kan ervoor zorgen dat een mogelijk probleem niet, of te laat wordt opgemerkt. Onzekerheid zorgt ervoor dat onvoorziene gebeurtenissen kunnen optreden, al kunnen deze zowel een positief als negatief effect hebben. Om de risico's te kunnen beheersen is risicomangement van belang, dit kan ervoor zorgen dat problemen tijdig worden opgemerkt en opgelost en moet zorg dragen voor een soepel implementatietraject.

Vergeleken met een nieuw te ontwerpen software systeem liggen de complexiteit en onzekerheid bij een standaard software systeem lager. Het systeem is namelijk al gebouwd en hoeft alleen nog naar eigen wens worden ingericht. Dit betekent dat er veel minder ontwikkeld (en getest) hoeft te worden. Dit scheelt tijd, vermindert kans op ontwerpfouten en neemt een deel van de onzekerheid weg over de datum waarop het systeem opgeleverd kan worden. Het is namelijk een stuk eenvoudiger om de tijd in te schatten voor het inrichten van een standaard systeem, dan het is voor het ontwerpen en bouwen van een nieuw systeem, aangezien een goede implementatiepartner dit al vele malen heeft gedaan bij andere organisaties. Daarbij zal een standaard systeem meestal al gebruikt worden bij andere organisaties, zodat ook bekend is hoe het systeem in de praktijk zijn dienst bewijst.

Een zeer bekende en veel gebruikte methode voor projectmanagement is Prince2. Deze methode moet een project beheersbaar houden door het op te delen in verschillende processen, dit vermindert de complexiteit. Het gebruik van Prince2, zeker als het naar de letter wordt gevolgd, brengt echter zelf ook weer extra complexiteit met zich mee. Kritische succesfactoren zijn factoren die van groot (kritisch) belang zijn voor het slagen van het project, deze moeten dus nauwlettend in de gaten worden gehouden. Er is een zevental kritische succesfactoren besproken die bij een project van belang zijn.

In deze eerste hoofdstukken van het onderzoek in de literatuur zijn theorieën en methoden naar voren gekomen, die als ze precies gevolgd worden, zouden leiden tot een vlekkeloos implementatietraject en succesvolle afsluiting van het project: Op tijd, binnen budget en volgens de specificaties. Daarmee is de vraag beantwoord hoe de implementatie van een standaard software systeem ideaal gezien zou moeten verlopen.

Verder is beschreven wat risico inhoudt en welke risicofactoren en kritische succesfactoren een rol kunnen spelen tijdens de implementatie.

Om te zien in hoeverre de theorieën verschillen van hoe het er in de praktijk aan toe gaat, is er ook een onderzoek gedaan naar hoe bekend de besproken theorieën zijn en of er in de praktijk ook gebruik van gemaakt wordt.

De uitkomsten van dit onderzoek lagen eigenlijk erg in lijn met de theorie die is besproken. Het is wel duidelijk dat de theorie algemeen bekend is onder de ondervraagde personen,

helaas is door de vraagstelling in de enquête niet duidelijk naar voren gekomen of, en hoe, deze in de praktijk dan wordt toegepast en of dit specifiek voor standaard software systemen geldt. De vraagstelling had achteraf gezien dus beter gekund. Toch is het praktijkonderzoek wel zinvol geweest, de opmerkingen van de ondervraagde personen hebben een hoop extra achtergrondinformatie en praktijkervaringen uit eerste hand opgeleverd.

De conclusie aan het eind van dit onderzoek is dat het toch wel lastig is gebleken om in de discussie over standaard software systemen niet af te dwalen naar een algemene discussie over software implementaties of zelfs projecten in het algemeen, aan de andere kant is het onvermijdelijk dat de implementatie van een standaard software systeem voor een groot deel samenvalt met de implementatie van een willekeurig softwaresysteem en projecten in het algemeen. Of er specifieke verschillen zijn, bijvoorbeeld een hogere kans op succes bij een standaard systeem, omdat bijvoorbeeld de complexiteit bij een standaard systeem lager zou kunnen zijn, is niet aangetoond in dit onderzoek.

Nawoord

Ten slotte een woord van dank, ten eerste aan dhr. Van der Pijl, de begeleider tijdens dit onderzoek en Damir Vadic, de meezeer. Daarnaast aan Marc la Chapelle van CRM-Resultants die heeft geholpen met het praktijkgedeelte.

Tenslotte de bedrijven/organisaties die hebben meegewerkt aan de enquête, in willekeurige volgorde: VCK Travel, Royal Lankhorst Euronet Group BV, Triple I, Wegener ICT|Media, Spectrum Benelux, GeoStick BV, Bookall BV, RDW, Kempen & Co, Delta Lloyd, Franssen Gerrits BV, ROC Twente, ArtEZ Hogeschool, Planon BV, Bosman, ROC van Amsterdam, UPC, ROC ter AA, Rabo Bouwfonds, Berenschot, Hot Item.

Bibliografie

- [1] M. Raskino and J. Mahoney, "Leading-Edge Ideas in IT Management, Business and Architecture From CIOs," Gartner, G00151540, 2007.
- [2] B. Waters, "Software as a Service, a look at the customer benefits," *Journal of Digital Asset Management*, vol. 1, no. 1, pp. 32-39, 2005.
- [3] L. Rengersen, Sustainable Software Implementation, Using emergent change to increase Enterprise Resource Planning (ERP) systems' implementation success, 2006.
- [4] Y. Kim, Z. Lee, and S. Gosain, "Impediments to successful ERP implementation process," *Business Process Management Journal*, vol. 11, no. 2, pp. 158-170, 2005.
- [5] "Rapport Lessen uit ICT-projecten bij de overheid, Deel A," Algemene Rekenkamer, 2007.
- [6] A. Koedijk and A. Verstelle, *ERP in bedrijf.*: KMPG e-Solutions, 2001.
- [7] Marilyn M. Parker, Robert J. Benson, and H.E. Trainor, *Information Economics: Linking Business Performance to Information Technology*. Upper Sadle River, NJ: Prentice-Hall, Inc., 1988.
- [8] Europese Unie. (2004, April) Publicatieblad van de Europese Unie. [Online]. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2004:134:0001:0113:nl:PDF>
- [9] Europees Parlement en de Raad. (2004, April) Publicatieblad Nr. L 134. [Online]. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32004L0018:NL:HTML>
- [10] J.Y. Ruwanpura, T.N. Ahmed, K. Kaba, and G.P. Mulvany, "Project Planning and Scheduling and its Impact to Project Outcome; A Study of EPC Projects in Canada," *AACE International Transactions*, pp. 201-209, 2006.
- [11] O. Perminova, M. Gustaffson, and K. Wilkstrom, "Defining uncertainty in projects, a new perspective," *International Journal of Project Management*, vol. 26, pp. 73-79, 2008.
- [12] M. Looijen, *Beheer van Informatiesystemen.*: Ten Hagen & Stam Uitgevers, 2004.
- [13] Prince2 organisatie. [Online]. http://www.ogc.gov.uk/methods_prince_2.asp
- [14] Fraunhofer Instituut: Smile methode. [Online]. <http://www.swm.iao.fraunhofer.de/Projektbeispiele/forschung/smile.jsp>
- [15] B. W. Boehm, "A Spiral Model of Software Development and Enhancement," *IEEE*, 1988.
- [16] F. Fui-Hoon Nah and S. Delgado, "Critical success factors for Enterprise Resource Planning implementation and upgrade," *The Journal of Computer Information Systems*, vol. 46, 2006.
- [17] L.J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no.

3, pp. 297-334, 1951.

- [18] L. Bielski, "CRM R.I.P.? Not exactly," *American Bankers Association Banking Journal*, vol. 96, September 2004.
- [19] Marc La Chapelle and Bas van Sluis, *CRM Implementaties volgens de Result Track Methode*, 2007.
- [20] I. Corner and M. Hinton, "Customer Relationship Management systems: Implementation risks and relations," *Qualitative Market Research*, vol. 5, 2002.
- [21] I. Hyväri, "Success of projects in different organizational conditions," *Project Management Journal*, vol. 37, September 2006.
- [22] B. Orr, "SaaS may be the end of software as we know it," *American Bankers Association Banking Journal*, vol. 98, no. 8, pp. 51-52, August 2006.
- [23] C. Besner and B. Hobbs, "The perceived value and potential contribution of project management practices to project success," *Project Management Journal*, vol. 37, no. 3, pp. 37-48, August 2006.

Bijlage 1: Enquête

“Enquête t.b.v. het onderzoek ‘succesvol implementeren van standaard software systemen’”

Naam:

Werkzaam bij:

Functie:

Korte beschrijving dagelijkse werkzaamheden:

De vragen zijn net als de samenvatting verdeeld in drie delen en zullen ook ongeveer de samenvatting volgen:

- fase voor het daadwerkelijk starten van een implementatie
- de implementatie fase
- onderkennen van -, en omgang met risico's tijdens de implementatie

De vragen bestaan voornamelijk uit stellingen die u kunt beoordelen met vijf mogelijkheden (sterk oneens – sterk mee eens). U mag uiteraard eventuele uitleg, opmerkingen of commentaar toevoegen.

Indien u in uw functie geen ervaring heeft met een van deze delen, of specifieke vragen, kunt u deze natuurlijk overslaan. Als u wilt, mag u de vragen in dat deel natuurlijk wel invullen.

Fase vóór het daadwerkelijk starten van een implementatie

Plan van Aanpak (PvA)

“Het opstellen van een goed en duidelijk PvA is belangrijk”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Als er een PvA is, moet deze zoveel mogelijk gevolgd worden”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Ik heb een PvA eigenlijk nooit nodig gehad, of noodzakelijk gevonden”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

Scope

“Het bepalen van de juiste scope is belangrijk”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Het is belangrijk om te inventariseren wat huidige knelpunten zijn, zodat deze in het nieuwe systeem opgelost kunnen worden”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Het is belangrijk om te weten wat de goedlopende punten zijn, zodat deze er na de implementatie nog steeds zijn.”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

Eisen en wensen

“Naast een PVA en bepalen van de scope, is een document met eisen en wensen ook belangrijk”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Het software gedeelte is belangrijk (welk pakket kiezen we?)”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Het hardware gedeelte is belangrijk (welke hardware/ASP?)”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“De leverancier van het systeem is belangrijk (bijv. Microsoft, Peoplesoft, Oracle)”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Er moet ook rekening worden gehouden met de eisen en wensen van de gebruikers”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

Pakketkeuze en leverancier

“Het is belangrijk om uitgebreid onderzoek te doen naar welk softwarepakket gekozen moet worden”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“De leverancier van het gekozen softwarepakket is belangrijk” (de implementatiepartner)

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

De implementatie fase

“Ik ben bekend met de verschillende fasen tijdens een implementatie”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Het opstellen van een implementatieplan tijdens de Concieve fase is belangrijk”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Het opstellen van functionele ontwerpen voor de inrichting van het systeem en te ontwikkelen maatwerk is belangrijk voor een goed ingericht systeem.

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Er moet eerst getest worden voor een nieuw systeem in productie wordt genomen.”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Trainingen voor gebruikers zijn noodzakelijk voor een vlotte acceptatie en goed gebruik van het nieuwe systeem.

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“Ook na de Go Live is ondersteuning van externe consultants, zeker in de eerste twee maanden, van belang.”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

Onderkennen van -, en omgang met risico's tijdens de implementatie

Risicoanalyse

“Het is belangrijk om risicofactoren tijdens een project goed in de gaten te houden”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

Risicomangement

“Ik weet wat de methode PRINCE2 inhoudt”

Sterk oneens Oneens Neutraal Mee eens Sterk mee eens

Opmerkingen:

“De PRINCE2 methode is gebruikt bij een project waarbij ik betrokken was of ben”

Ja Nee

“Indien nee, werd er wel een andere methode voor project/risicomangement gebruikt?”

Ja:

Kritische succesfactoren – waardeer deze uitspraken met een cijfer tussen 1 en 10

“Het hebben van een degelijk projectplan en visie is van kritisch belang voor het slagen van het project”

1 2 3 4 5 6 7 8 9 10

Opmerkingen:

“Change management is van kritisch belang voor het slagen van het project”

1 2 3 4 5 6 7 8 9 10

Opmerkingen:

“Communicatie is van kritisch belang voor het slagen van het project”

1 2 3 4 5 6 7 8 9 10

Opmerkingen:

“De samenstelling van skills en personen binnen het projectteam is van kritisch belang voor het slagen van het project”

1 2 3 4 5 6 7 8 9 10

Opmerkingen:

“Ondersteuning vanuit het management en het hebben van een “champion” (een drijvende kracht in het projectteam) is van kritisch belang voor het slagen van het project”

1 2 3 4 5 6 7 8 9 10

Opmerkingen:

“Een goede projectmanager is van kritisch belang voor het slagen van het project”

1 2 3 4 5 6 7 8 9 10

Opmerkingen:

“ Een analyse van de nieuwe systeemeisen en bestaande systemen is van kritisch belang voor het slagen van het project”

1 2 3 4 5 6 7 8 9 10

Opmerkingen: