

Afstudeerverslag

Virtualiseren mail en databaseservices

Afstudeerverslag Jaco Kranenburg
Opleiding: Technische Informatica Haagse Hogeschool
Datum: 25/5/2011
Versie: 1.0

Referaat

Afstudeerverslag virtualiseren mail- en database services, Utrecht 18 mei 2011.

Afstudeerverslag Jaco Kranenburg, Haagse hogeschool Technische Informatica. Project voor Pimarox uitgevoerd van 4 september 2010 tot en met 29 april 2011.

Dit rapport beschrijft het onderzoek naar de functionele en technische eisen en wensen voor een gevirtualiseerd Microsoft Exchange server en Microsoft SQL server ontwerp. Daarnaast wordt het implementeren van de ontworpen omgevingen besproken.

Descriptoren:

- Afstudeerverslag
- Virtualisatie
- Microsoft Exchange server
- Microsoft SQL server
- Implementatie infrastructuur

Voorwoord

Dit afstudeerverslag is geschreven naar aanleiding van het afstuderen van Jaco Kranenburg voor de opleiding Technische Informatica aan de Haagse Hogeschool. De afstudeeropdracht is uitgevoerd voor Pimarox van 4 september 2010 tot en met 29 april 2011. De beschreven netwerkinfrastructuren zijn in werkelijkheid van een grote klant van Pimarox waar ik in deze periode gedetacheerd was. In dit verslag gebruik ik consequent de organisatienaam Pimarox.

Het verslag is geschreven voor studenten in een ICT richting en voor mensen die werken in dezelfde richting. Enige voorkennis van de beschreven Microsoft® producten (Windows, SQL server en Exchange server) en van VMware vSphere is noodzakelijk.

De totstandkoming van dit project staat beschreven in hoofdstuk 1 en 2. Het definiëren van de eisen en wensen en het ontwerpen van de nieuwe infrastructuur wordt beschreven in hoofdstuk 3 en 4. Het opstellen van het migratieplan staat beschreven in hoofdstuk 5. In hoofdstuk 6 evalueer ik het project en het proces.

In de toekomst kunnen de hoofdstukken 3, 4 en 5 van pas komen bij het ontwerpen van en migreren naar een nieuwe infrastructuur.

Tot slot wil ik graag iedereen bedanken die een bijdrage heeft geleverd. In het bijzonder noem ik mijn vrouw Annemieke voor haar eindeloze geduld en steun, mijn collega's Remko Francken en Gertjan van Bruchem voor het helpen definiëren van dit project en het kritisch kijken naar de resultaten en de docenten Rutger Spaans en Hans Witkamp voor hun nuttige tips bij het schrijven van dit verslag.

Utrecht, 25 mei 2011

Jaco Kranenburg

Inleiding

Een grote klant van Pimarox is een uitvoerder van financiële regelingen en informatietaken in de zorg- en welzijnssector. Als gevolg van een grote groei van de organisatie is het aantal ICT systemen erg hard gegroeid, zo hard dat de datacentres nu vol zitten. Tegelijkertijd is er onvoldoende aandacht geweest voor vervangingsinvesteringen aan de bestaande infrastructuur.

Op dit moment is het noodzakelijk om de bestaande Microsoft SQL server en Microsoft Exchange server omgeving te vervangen en te virtualiseren. Dit project is geïnitieerd om nieuwe ontwerpen en implementatieplannen te maken voor deze omgevingen.

De beschreven omgeving is van een grote klant van Pimarox. Gedurende de looptijd van het project was ik bij deze klant gedetacheerd als senior systeembeheerder bij het team ICT beheer. Bij de start van mijn project was er echter geen duidelijkheid over mijn contractduur bij de klant. De opdrachtgever voor dit project is dus Pimarox, deze naam komt ook terug in dit document.

In dit project is onderzocht welke functionaliteiten nu geleverd worden door de genoemde omgevingen en hoe de omgevingen ontworpen zouden moeten worden om weer toekomstvast te zijn. Belangrijke randvoorwaarden hierbij zijn dat er gevirtualiseerd moet worden en dat de ondersteuning van de leverancier gewaarborgd moet zijn. Tot slot is op basis van een Proof of Concept bepaald hoe de migratie zou moeten verlopen.

In hoofdstuk 1 en 2 wordt uitgelegd hoe dit project tot stand gekomen is. Het onderzoeken van de eisen volgt in hoofdstuk 3. In hoofdstuk 4 en 5 worden de nieuwe omgevingen ontworpen en de implementatie beschreven. Ten slotte volgt in hoofdstuk 6 een evaluatie van het project en het proces.

Belangrijk om te vermelden is verder dat dit project gestart is vanuit mijn afstudeerbehoefte, maar op dit moment is de klant wel bezig met een migratie naar een Exchange 2010 omgeving die gebaseerd is op mijn ontwerp.

Inhoudsopgave

Inleiding.....	4
Inhoudsopgave.....	5
1. Tot stand komen van de opdracht	6
1.1. Het begin	6
1.2. Bepalen van de scope	6
1.3. Bepalen van de doelstellingen en het resultaat	8
1.4. Bepalen van de projectmethodiek	8
1.5. Bepalen van de fasering	9
2. De initiatieffase.....	12
2.1. Projectbeschrijving.....	13
2.2. Planning	15
2.3. Projectbeheersing.....	16
3. Definitiefase	18
3.1. Informatie verzamelen en verwerken	18
3.2. Het netwerk	20
3.3. Mailomgeving	21
3.4. SQL omgeving	24
3.5. Virtualisatieplatform	26
3.6. Planning volgende fase.....	27
4. Ontwerpfase.....	28
4.1. Verzamelen kennis	28
4.2. Bepalen eisen en wensen	29
4.3. Ontwikkeling SQL ontwerp.....	29
4.4. Uitwerking van het high level design	32
4.5. Configuratie virtuele machines.....	36
4.6. Replicatie MC	37
4.7. Ontwikkeling Exchange ontwerp	38
4.8. Ontwerp mailbox server	39
4.9. Ontwerp Client Access server	42
4.10. Ontwerp Hub Transport Server	43
5. Realisatiefase.....	44
5.1. Schrijven implementatieplan	44
5.2. Inrichting Proof of Concept	46
5.3. Implementatie SQL ontwerp.....	46
5.4. Implementatie Exchange ontwerp	47
6. Evaluatie	50
6.1. Productevaluatie	50
6.2. Procesevaluatie	52
6.3. Competenties	53
Bijlagen	55
Literatuurlijst.....	56

1. Tot stand komen van de opdracht

In dit hoofdstuk beschrijf ik hoe de afstudeeropdracht tot stand gekomen is en waarom de gemaakte keuzes zo gemaakt zijn.

1.1. Het begin

Pimarox is door de veranderde regelgeving op het gebied van de zorg hard gegroeid en zal dat in de toekomst blijven doen. Deze groei vertaalt zich in het aantrekken van veel nieuwe medewerkers en de aanschaf of ontwikkeling van nieuwe informatie systemen. Zoals vaak het geval zijn niet alle onderdelen van het bedrijf even hard mee gegroeid, of is de groei ten koste gegaan van andere projecten.

Bij Pimarox valt op dat het datacentre vol zit en dat er sprake is van een groot aantal verouderde systemen. Gesteld kan dus worden dat het datacentre niet hard genoeg mee gegroeid is en dat de groei ten koste is gegaan van investeringen om de bestaande systemen te vervangen.

Inmiddels blijkt echter dat "de rek eruit is". Het datacentre zit nu echt vol, tegelijkertijd wordt het voor beheer steeds lastiger om de bestaande systemen in de lucht te houden door het gebruik van verouderde hard- en software. Vervanging van deze fysieke systemen door nieuwe fysieke systemen is niet mogelijk vanwege het ruimtegebrek. Om deze deadlock situatie op te heffen wordt er gedacht aan virtualisatie. Niet alle systemen zijn zonder meer geschikt voor virtualisatie, soms zijn er aanpassingen nodig.

Vanuit ICT beheer is er een dringende wens om de huidige mail- en database services te upgraden. Dat is alleen mogelijk als deze services gevirtualiseerd kunnen worden.

Ik heb dit project geïnitieerd om te onderzoeken hoe de huidige mail- en database services gevirtualiseerd kunnen worden.

1.2. Bepalen van de scope

Voor dit project bepaal ik de scope aan de hand van onderstaande vragen:

1. Waar zit de ergste nood?
2. Wat kan er vanuit ICT Beheer gedaan worden?
3. Welke producten zijn zo toekomstvast dat een investering mogelijk wordt geacht?

Ik wil om te beginnen een globaal antwoord op deze vraag krijgen. Een verdere uitwerking komt wel als ik het plan van aanpak maak. Daar komt bij dat snelheid nu het belangrijkste is. Ik wil in deze fase mijn collega's bij systeembeheer nadrukkelijk hierbij betrekken om op deze wijze draagvlak te creëren voor de voorgestelde veranderingen.

Veel gebruikte manieren om aan informatie te komen zijn:

- Tweegesprekken met collega's;
- Een enquête;
- Agenderen voor een afdelingsoverleg;
- Een brainstormsessie organiseren.

Tweegesprekken en een enquête hebben als nadeel dat ze de inzichten van een enkele collega opleveren, zonder dat er sprake is van een groepsgesprek. Agenderen voor een afdelingsoverleg en een brainstormsessie kunnen deze beperkingen opheffen. De brainstormsessie heeft als extra voordeel dat de agenda en de deelnemers beperkt zijn. Ik kies in dit geval dus voor een brainstormsessie met de meest betrokken collega's. Ik richt de brainstormsessie als volgt in:

- Brainstormronde over de vraag waar de ergste nood zit;
- Discussie over de randvoorwaarden.

De uitkomst van de brainstormronde is dat er grote behoefte is aan vervanging van de volgende servers:

1. een aantal webservers op basis van Microsoft Windows 2000;
2. de mailservers op basis van Microsoft Exchange server 2003 en
3. het Microsoft SQL2000 database platform.

De fysieke servers zijn inmiddels meer dan 5 jaar oud en over hun economische- en technische levensduur heen. Daarnaast zijn het besturingssysteem (Microsoft Windows 2000) en de middleware (Internet Information Server 5, SQL server en Exchange server 2003) dermate verouderd dat ze niet meer ondersteund worden door de leverancier¹.

Direct na de brainstorm is de discussie gestart met als centrale vraag "Wat komt er kijken bij het vervangen van deze servers?"

In de discussie kwam ook naar voren dat vervanging van de webservers niet gerealiseerd kan worden zonder ondersteuning van de afdeling Applicatie Ontwikkeling als gevolg van een aantal zelf ontwikkelde applicaties die op deze servers draaien. De vervanging van de mail- en SQL servers daarentegen kan volledig door ICT beheer uitgevoerd worden, er zijn geen voorbereidende werkzaamheden van andere afdelingen noodzakelijk.

Ik zie de afhankelijkheid van de afdeling Applicatie ontwikkeling als een risico. Ik kan namelijk geen invloed uitoefenen op de prioriteitenlijst van deze afdeling. Daar komt bij dat betrekken van een andere afdeling met zich meebrengt dat er kosten gemaakt worden die verrekend moeten worden. Deze risico's spelen niet alleen tijdens de voorbereiding, maar ook als ik eenmaal een goedgekeurd projectplan zou hebben, dan nog vind ik het risicovol om zonder duidelijke opdracht van de directie (zie ook de volgende paragraaf) afhankelijk te zijn van een andere afdeling.

Vervanging van de mail- en database servers leveren wat dat betreft minder risico's op, want er zijn geen voorbereidende werkzaamheden van andere afdelingen noodzakelijk.

Gezien de opstelling van de directie heb ik besloten om de scope te beperken tot de mail- en database services.

¹ Zie: <http://support.microsoft.com/gp/lifeselectindex#top>

1.3. Bepalen van de doelstellingen en het resultaat

De aanleiding² en de scope van het project geven een richting aan de doelstellingen en het beoogde resultaat die gehaald moeten worden, maar het resultaat³ moet wel voldoen aan de SMART criteria:

Specifiek
Meetbaar
Acceptabel
Realistisch
Tijdgebonden

Het resultaat is specifiek want er is duidelijkheid over wat er bij Pimarox verstaan wordt onder de mail- en Microsoft SQL database services.

Het resultaat is Meetbaar door het criterium “goedgekeurd”. Alle opgeleverde documenten worden goedgekeurd door de opdrachtgever.

Ik heb het resultaat acceptabel en realistisch gemaakt door mij te beperken tot het opleveren van een implementatiescenario.

De daadwerkelijke implementatie zit hier dus niet in. De directie wil namelijk pas een definitief besluit over de benodigde investeringen nemen als er een goed onderbouwd ontwerp is. Ik heb dus vrijwel geen invloed of mijn ontwerp binnen de looptijd van het project geïmplementeerd kan worden.

Het opnemen van implementatie van de ontwerpen in het resultaat van mijn project vind ik daarom niet acceptabel en realistisch.

Het criterium tijdgebonden zit in de beperkte looptijd van het project.

1.4. Bepalen van de projectmethodiek

Om de kans van slagen van het project te vergroten is het van belang om op gestructureerde wijze aan de slag te gaan. Ik verwacht dat een goede projectmethodiek mij helpt om gestructureerd dit project aan te pakken. De vraag die hieruit voort komt is: “Wat is een goede projectmethodiek?” Dit antwoord is natuurlijk subjectief en verschilt per project. Ik begin met het formuleren van mijn verwachtingen van de projectmethodiek.

Een goede projectmethodiek:

- Helpt mij om gestructureerd te werk te gaan zonder dat het veel extra werk met zich meebrengt. Dit project kent een eenvoudige opzet en dat wil ik graag zo houden.
- Is geschikt voor een klein project (weinig betrokkenen en beperkte doorlooptijd).
- Kent een duidelijke fasering. Een valkuil voor mij is dat een duidelijke fasering ontbreekt. Ik loop dan het risico dat ik te snel te diep ga en dan alsnog halverwege verzand in allerlei onbekende details.
- Heeft goede en toegankelijke documentatie. Ik ben niet in de gelegenheid om veel tijd te steken in het mij ‘eigen’ maken van de “in’s en out’s” van een bepaalde projectmethodiek. Ik geef de voorkeur aan een methodiek waar ik direct mee aan de slag kan.

² Zie: hoofdstuk 2 “aanleiding” in het projectplan, bijlage 1

³ Zie: hoofdstuk 5 “resultaat” in het projectplan, bijlage 1

Om tot een definitieve selectie te komen maak ik gebruik van een shortlist. Vervolgens beoordeel ik de verschillende methodieken op hun toepasbaarheid in mijn specifieke situatie. De samenstelling van de shortlist is gebaseerd op de adviezen van anderen. Mijn shortlist ziet er als volgt uit:

1. Timeboxing
2. Grit
3. Prince2

Ad 1. Timeboxing is interessant vanwege de strakke tijdsaders waarin het project gegoten wordt. Voor mijn project geldt dat er activiteiten in zitten waar ik een afweging moet maken hoeveel tijd ik er in wil steken. Binnen een bepaalde bandbreedte geldt namelijk dat meer tijd voor een beter resultaat zorgt. De vraag is dan: “wanneer is het goed genoeg?”

Ad 2. De Grit methodiek is interessant vanwege de praktische aanpak en de toegankelijke documentatie.

Ad 3. Prince2 is erg uitgebreid gedocumenteerd en heel erg geschikt voor grote projecten.

De afweging van de verschillende alternatieven staat in de volgende tabel. De totaalscore bestaat uit de som van het aantal plusjes.

	Gestructureerd werken	Klein project	Duidelijke fasering	Toegankelijke documentatie	Totaal
Timeboxing	-	+	-	-	1
Grit	+	+	+	+	4
Prince2	+	-	+	+	3

Tabel 1: Afweging projectmethodieken

De best scorende methodieken zijn Grit en Prince2. Grit scoort echter beter op het criterium ‘klein project’. Ik heb het project dan ook ingericht volgens de Grit methodiek.

1.5. Bepalen van de fasering

Grit schrijft in zijn boek⁴ een projectfasering voor. Deze fasering ziet er als volgt uit:

- Initiatief
- Definitie
- Ontwerp
- Voorbereiding
- Realisatie
- Nazorg

In de terminologie van Grit is de initiatieffase bedoeld om te kijken of er een project gestart moet en kan worden. Het resultaat van deze fase is doorgaans een projectvoorstel. Het afstudeerplan wat vooraf gaat aan mijn project is al te zien als projectvoorstel.

Na de initiatieffase volgt bij Grit de definitiefase. In de definitiefase wordt duidelijk wat de doelstellingen van het project zijn. Het eindresultaat van deze fase is een plan van aanpak.

⁴ Project management, Roel Grit, 2008

Na de definitiefase volgt bij Grit de ontwerpfase waarin de doelstellingen van het project ontworpen wordt. De gedachte hier achter is dat er in de voorgaande fase (redelijk) nauwkeurig is wat er gemaakt moet worden en aan welke eisen dat moet voldoen. Het eindproduct van deze fase is het ontwerpdocument.

Na afloop van de ontwerpfase volgt de voorbereiding. Tijdens de voorbereiding wordt het ontwerp geschikt gemaakt voor realisatie. Het resultaat van deze fase is een gedetailleerd ontwerp.

Tijdens de realisatiefase wordt er dan uiteindelijk gebouwd en komt het vooraf afgesproken projectresultaat tot stand.

De nazorgfase zoals Grit die omschrijft gaat mijns inziens voorbij aan de beperkte looptijd van een project. Grit beschouwt onderhoud aan een systeem als onderdeel van een project. In mijn ervaring is onderhoud aan een systeem een verantwoordelijkheid van de beheerorganisatie. Ik vind dat nazorg beperkt moet blijven tot complete invoering van het systeem, inclusief eventuele noodzakelijke aanpassingen. Als een systeem volledig geïmplementeerd is en naar tevredenheid werkt kan het project afgesloten worden.

Ik vind de fasering van Grit niet zo geschikt voor dit project omdat mijn project op een aantal punten afwijkt van de gedachten achter de fasering van Grit:

1. Het vaststellen van de eisen van het te ontwerpen systeem zijn een belangrijk onderdeel van het project. De fasering van Grit gaat er van uit dat deze eisen al bekend zijn.
2. De gedachte achter de voorbereidende fase is dat duidelijk is op welke termijn en onder welke randvoorwaarden de ontwerpen gerealiseerd gaan worden. Bij mijn project is dat niet het geval. Ik heb daadwerkelijke implementatie van de ontwerpen niet als projectresultaat opgenomen. Uit ervaring weet ik dat een op voorhand geschreven implementatieplan nog een aantal variabelen bevat die pas kort voor implementatie ingevuld kunnen worden (denk aan servernamen, IP adressen etc.). Daarnaast bestaat de mogelijkheid dat er ook wijzigingen aan het netwerk komen voor er geïmplementeerd gaat worden waardoor het ontwerp opnieuw beoordeeld moet worden. Ik kan dus geen implementatieplan maken dat zover gedetailleerd is dat het zo uitgevoerd kan worden. Er blijven een aantal vraagtekens open.
3. Een nazorgfase is zinloos als er niets gebouwd is.

Ik wijk daarom af van de voorgeschreven fasering om te komen tot een fasering die beter past in mijn project.

Om een beter passende fasering voor het project te bepalen maak ik eerst een globaal stappenplan. Dit plan bevat op abstract niveau de stappen die gezet moeten worden om het einddoel te halen. Om de benodigde stappen te bepalen ga ik terug redeneren vanuit het eindresultaat:

Het eindresultaat is een implementatiescenario voor een gevirtualiseerd mail- en SQL-serverplatform. Implementeren gebeurt op basis van een ontwerp, er moet dus ook een ontwerp worden gemaakt.

Om een systeem te kunnen ontwerpen is het noodzakelijk om te weten welke eisen en wensen aan het systeem gesteld worden. Het probleem in dit geval is dat er geen vastgestelde lijst met eisen en wensen is. Als organisatiebreed moet worden bepaald wat de eisen en wensen voor een mail- en databaseplatform zijn nemen de tijd en kosten snel toe, terwijl het eindresultaat niet per definitie beter wordt, is mijn ervaring.

Om nu toch de eisen en wensen voor de systemen te kunnen bepalen ga ik uit van de huidige situatie. De achterliggende gedachte hierbij is dat ik een nieuw ontwerp maak voor de mail- en SQL servers waarin de huidige functionaliteiten gewaarborgd zijn.

Op abstract niveau moeten dus de volgende fasen doorlopen worden: Beschrijven van het project, definiëren van de huidige situatie, ontwerpen van een nieuw systeem, beschrijven van de implementatie van dit nieuwe systeem. Als ik deze activiteiten vergelijk met de fasering van Grit, dan kom ik tot de volgende aanpassingen aan de standaard fasering:

- a) Initiatieffase. Het afstudeerplan is al een projectvoorstel, maar ik ben nog niet eens echt begonnen. Ik wijzig de doelstelling van deze fase in overeenstemming met de opdrachtgever over de projectdoelstellingen en het verloop van het project. Het eindresultaat van deze fase is dan een goedgekleurd plan van aanpak.
- b) Definitiefase. Ik heb in de voorgaande fase al een plan van aanpak gemaakt. Kenmerkend voor mijn project is dat de opdrachtgever geen duidelijkheid heeft over de eisen en wensen die aan het resultaat worden gesteld. Het vaststellen van de eisen en wensen is daarmee een taak van het project geworden. Ik wijzig de doelstelling van deze fase in ‘het vaststellen van de huidige situatie en de eisen en wensen die aan de nieuwe ontwerpen gesteld worden’. Het gewijzigde resultaat is een definitierapport waarin de huidige situatie beschreven staat en waarin de eisen en wensen voor het eindresultaat vastgelegd zijn.
- c) Voorbereidende fase. Deze fase kan vervallen.
- d) Realisatiefase. Ik ga niet implementeren, maar wel uitschrijven hoe de implementatie moet gebeuren. Het resultaat van deze fase wordt dus een uitgewerkt implementatieplan.
- e) Nazorgfase. Deze fase kan ook vervallen.

De uiteindelijke fasering van mijn project ziet er dus als volgt uit:

1. De *initiatieffase* waarin ik overeenstemming met de opdrachtgever bereik over de projectdoelstellingen en het verloop van het project.
2. De *definitiefase* waarin ik de huidige situatie analyseer om op basis van die analyse de eisen en wensen voor de toekomstige ontwerpen bepaal.
3. De *ontwerpfase* waarin ik de toekomstige situatie ga ontwerpen.
4. De *realisatiefase* waarin het implementatieplan wordt geschreven.

2. De initiatieffase

In de initiatieffase moet het globale plan van aanpak uit het projectplan verder geconcretiseerd worden. Ook moet overeenstemming met de opdrachtgever worden bereikt over de doelstellingen en het verloop van het project. Het eindproduct van deze fase is het plan van aanpak⁵.

Grit schrijft in zijn boek de volgende hoofdstukindeling voor een plan van aanpak voor:

1. Achtergronden
2. Projectopdracht
3. Projectactiviteiten
4. Projectgrenzen
5. Producten
6. Kwaliteit
7. Projectorganisatie
8. Planning
9. Kosten en baten
10. Risico's

Deze hoofdstukindeling wijkt af van de hoofdstukindeling die Pimarox hanteert in een plan van aanpak:

1. Beschrijving van het project
2. Business case
3. Planning
4. Risicobeheersing
5. Beheersingsmechanismen
6. Projectorganisatie

Ik heb besloten om voor zoveel mogelijk de hoofdstukindeling van Pimarox te volgen omdat alle betrokkenen daar het meest mee vertrouwd zijn. In de inhoud zijn er weinig verschillen tussen een 'Grit-plan van aanpak' en een 'Pimarox-plan van aanpak'.

Pimarox heeft de gewoonte om de aangeboden oplossing in een business case uit te werken. De aangeboden oplossing is echter al grotendeels vastgelegd in het projectplan, waardoor een business case geen toegevoegde waarde heeft. In mijn projectplan besteed ik ook geen aandacht aan de kosten en baten, omdat er alleen door het maken van een implementatieplan nog geen baten te behalen zijn en er geen kosten zijn begroot voor dit project. Ik voeg de hoofdstukken risicobeheersing, beheersingsmechanismen en projectorganisatie samen, want mijn projectorganisatie is klein en er zijn weinig externe factoren of belanghebbenden waaraan aandacht besteed moet worden.

Uiteindelijk ziet de hoofdstukindeling van het Plan van Aanpak er dus als volgt uit:

1. Projectbeschrijving
2. Planning
3. Projectbeheersing

⁵ Zie bijlage 2: "Plan van aanpak virtualiseren mail- en database services"

Een deel van de inhoud van het projectplan komt ook terug in plan van aanpak. De doelstelling van beide documenten is echter anders.

Het plan van aanpak is daarmee gedetailleerder dan het projectplan.

Het projectplan is geschreven om goedkeuring van de opleiding te krijgen om dit project als afstudeerproject in te zetten. In de relatie met de opdrachtgever is het projectplan te zien als “intentieverklaring” tussen opdrachtgever en projectleider en is primair bedoeld om toestemming te krijgen om aan de slag te gaan. Het plan van aanpak is bedoeld om met de opdrachtgever overeenstemming te bereiken over de doelstellingen en het verloop van het project.

2.1. Projectbeschrijving

Het hoofdstuk projectbeschrijving⁶ in het plan van aanpak geeft een beschrijving van het project. De inhoud van dit hoofdstuk is grotendeels gelijk aan het projectplan.

In tegenstelling tot het projectplan besteedt dit plan van aanpak expliciet aandacht aan de scope van het project. De scope is wel aan de orde geweest bij het schrijven van het projectplan (zie hoofdstuk 1). Door nu veel aandacht te besteden aan de scope van het project, beperk ik het risico dat er later discussie met de opdrachtgever ontstaat over de reikwijdte van mijn ontwerp en implementatieplan.

Er is een belangrijk verschil tussen het projectplan en het plan van aanpak:

- De scope in het projectplan bepaalt wat ik uiteindelijk moet opleveren.
- De scope in het plan van aanpak bepaalt wat ik ga onderzoeken.

Ik heb de scope van mijn project bepaald door de relaties van de mail- en database servers met andere systemen in kaart te brengen. Vervolgens kan ik per relatie bepalen of ik deze relatie wel of niet kan meenemen en wat de mogelijke gevolgen zijn.

Een belangrijke relatie van gevirtualiseerde mail- en database services is het virtualisatieplatform, de hard- en softwarelaag die het mogelijk maakt om te virtualiseren.

Ik neem in het plan van aanpak het virtualisatieplatform expliciet binnen de scope van het project op⁷. Mijn overwegingen hiervoor zijn als volgt:

- a) Om een gevirtualiseerde Exchange en SQL omgeving te kunnen draaien moet er in elk geval ruimte in de datastores zijn.
- b) Mogelijk moet er ook iets gedaan worden aan (RAM)geheugen- en processorcapaciteit.
- c) Het ontwerp moet passen binnen de huidige opzet en beheersprocedures van het virtualisatieplatform.

⁶ Zie: hoofdstuk 1 “projectbeschrijving” in het plan van aanpak (bijlage 1)

⁷ Zie Paragraaf 1.7 “Scope” van het Plan van aanpak (bijlage 2)

- d) Door het virtualisatieplatform expliciet binnen scope te plaatsen, neem ik het op in mijn onderzoek en maak ik voor implementatie duidelijk wat er gevraagd wordt van het virtualisatieplatform en kan de directie een afgewogen beslissing nemen. Het alternatief is dat ik geen onderzoek doe naar de capaciteit van het virtualisatieplatform, met als gevolg dat het ontwerp en implementatieplan niet getoetst zijn aan de mogelijkheden van het virtualisatieplatform en de directie dus geen afgewogen beslissing kan nemen.

Ik ga dus geen nieuw ontwerp maken voor het virtualisatieplatform, maar wel onderzoek doen naar de capaciteiten van het virtualisatieplatform.

De volgende onderdelen zijn buiten scope⁸ van dit project geplaatst:

1. Aanpassingen aan de werkplekken om volledig gebruik te kunnen maken van de nieuwe infrastructuur
2. Aanpassingen aan gerelateerde systemen
3. Aanpassingen aan gerelateerde software
4. Implementatie van de nieuwe infrastructuur
5. Kostenraming voor apparatuur

Ad 1: Bij implementatie van Exchange kan ik kiezen om de werkplekken ook aan te passen, maar dat is niet strikt noodzakelijk. Ik neem de werkplekken niet in scope op met het oog op de risico's en de voor het project benodigde tijd. Als ik de werkplekken in de scope zou opnemen maak ik het mogelijk dat mijn project veranderingen aan de werkplekken met zich meebrengt. Dat is aantrekkelijk om optimaal van nieuwe features gebruik te kunnen maken. Tegelijkertijd maak ik mijn project afhankelijk van vele andere systemen en projecten. Mijn project kent dan meer risico's en kost veel meer tijd.

In scope nemen van de werkplekken maakt het project veel groter en afhankelijk van andere projecten. Het eindresultaat komt dan in gevaar. De consequentie van het niet in scope nemen van de werkplekken is een extra eis aan het ontwerp, namelijk om Exchange zodanig te ontwerpen dat het ook met de huidige werkplek standaard overweg kan.

Ad 2: Gerelateerde systemen zijn bijvoorbeeld de firewalls en applicatieservers die gebruik maken van de Exchange services. Ook hier geldt dat ik een keuze moet maken tussen het zoveel mogelijk benutten van nieuwe features of het voor mij realiseerbaar houden van het project.

Ik weet op basis van mijn kennis van Exchange server dat er nieuwe features zijn die kunnen leiden tot grootscheepse wijzigingen aan infrastructuur en systemen. Deze wijzigingen zorgen ervoor dat de tijd die nodig is voor het project veel groter wordt en dat er mogelijk externe leveranciers betrokken moeten worden bij het project. Door de gerelateerde systemen in zijn geheel buiten scope te plaatsen geef ik strakkere kaders aan mijn project en beperk ik de hoeveelheid tijd en externe partijen die benodigd zijn.

Ad 3: Bij gerelateerde software kan bijvoorbeeld gedacht worden aan de applicaties die gebruik maken van de SQL database services. Voor aanpassingen hieraan is Pimarox afhankelijk van de leveranciers. Daarnaast heeft de software weer relaties met andere systemen. Binnen scope nemen van deze software heeft tot gevolg dat ik verantwoordelijk word voor het ontwerpen en implementeren van de benodigde aanpassingen. Hierdoor wordt de kans van slagen van het project kleiner en de hoeveelheid benodigde tijd voor het project groter. Buiten scope plaatsen doet ook recht aan de bijzondere positie van mijn project, namelijk dat ik van te voren niet weet of, wanneer en in welke mate mijn ontwerp geïmplementeerd gaat worden.

⁸ Zie Paragraaf 1.8 “Niet in scope” van het Plan van aanpak (bijlage 2)

Ad 4: Implementatie van de nieuwe ontwerpen houdt in dat een aantal randvoorwaarden geregeld moet zijn. Hier staan kosten tegenover. De directie wil nog niet besluiten dat het ontwerp ook daadwerkelijk geïmplementeerd wordt, pas als de benodigde kosten bekend zijn wil de directie implementatie van de ontwerpen overwegen. Ik vind het risico dat de ontwerpen niet geïmplementeerd worden te groot. Door de implementatie buiten scope te plaatsen is mijn opdracht om een ontwerp neer te zetten dat in een later stadium met relatief weinig inspanning alsnog geïmplementeerd kan worden.

Ad 5: De prijs van hardware fluctueert. Het is niet duidelijk of en wanneer de nieuwe infrastructuur daadwerkelijk geïmplementeerd kunnen worden, een kostenraming als onderdeel van dit project is daarmee niet relevant.

2.2. Planning

Ik begin het hoofdstuk planning met het onderwerp aannames⁹. Ik heb besloten om onderscheid te laten bestaan tussen aannames en risico's. Zowel de aannames als de risico's zijn randvoorwaarden van het project. Het verschil is echter dat de *risico's* een bedreiging vormen voor het behalen van de doelstellingen¹⁰. De risico's zijn deels beïnvloedbaar door de opdrachtgever of de projectleider. De *aannames* zijn geen directe bedreiging voor de doelstellingen. Ze liggen geheel buiten de invloedssfeer van de projectleider. Als aan één of meer aannames niet is voldaan dan heeft dat weinig invloed op het kunnen realiseren van de doelstellingen.

Een voorbeeld van een aanname is:

De Active Directory is voorbereid voor Exchange 2010 of kan zonder problemen geschikt worden gemaakt. Er is in elk geval geen extra hardware of design wijziging noodzakelijk

Ik kom later in dit document nog terug op de risico's.

Om een inzichtelijke planning voor het project te kunnen maken begin ik op abstract niveau met het noemen van de activiteiten die voor elke fase uitgevoerd moeten worden. Dat ziet er voor de definitiefase¹¹ als volgt uit:

Activiteit:	Doel:
Bepalen functionele eisen Exchange	Bij gebrek aan een functioneel en technisch ontwerp voor de bestaande Exchange infrastructuur wordt tijdens de definitiefase de huidige omgeving in kaart gebracht om daarmee de functionele eisen te bepalen.
Bepalen functionele eisen SQL server	De huidige SQL server omgeving wordt in kaart gebracht om daarmee de gestelde functionele eisen te kunnen bepalen.
Bepalen kenmerken VMware omgeving	De huidige VMware infrastructuur wordt geanalyseerd op de bruikbaarheid voor gevirtualiseerde Exchange en SQL server omgevingen.
Bepalen algemene afspraken	De algemene afspraken over het beheer maken een functioneel ontwerp compleet. Voor zover deze afspraken niet op papier staan zullen deze uit gesprekken duidelijk moeten worden.

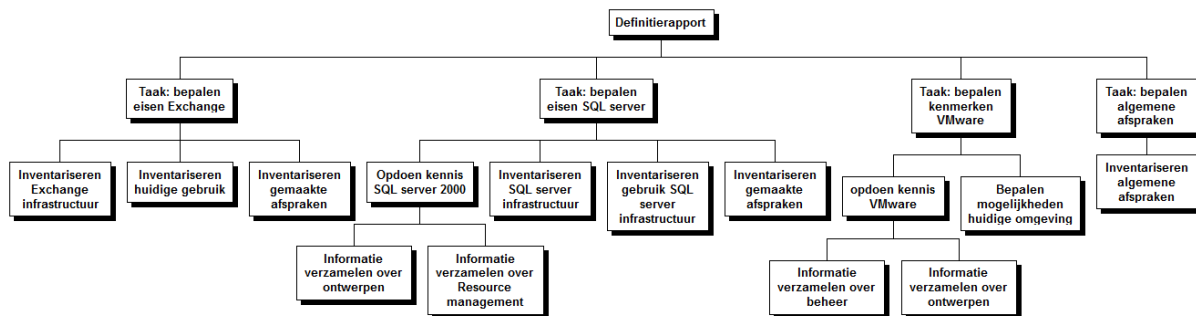
Tabel 2: globale activiteitenplanning definitiefase

⁹ Zie paragraaf 2.1 “Aannames” van het plan van aanpak (bijlage 2)

¹⁰ Zie ook de definitie van risico's in het boek “Projectmanagement” van Grit

¹¹ Zie paragraaf 2.3 “Globale activiteitenplanning” van het plan van aanpak (bijlage2) voor de overige fases

Ik heb de abstracte activiteitenplanning voor de definitiefase verder uitgewerkt in een meer concrete activiteitenplanning. Een Work Breakdown Structure maakt het mogelijk een grote lap activiteiten toch samenhangend te tonen:



Figuur 1: WBS definitiefase

Nu ik duidelijk heb wat er precies gedaan moet worden voor de volgende fase kan ik daar ook een tijdsplanning voor maken¹² door per activiteit een inschatting te maken hoeveel tijd ik er voor nodig denk te hebben. Deze inschatting maak ik op basis van mijn eigen ervaring met dit soort werkzaamheden.

2.3. Projectbeheersing

In een plan van aanpak is het gebruikelijk om behalve aan de risicobeheersing ook expliciet aandacht te besteden aan de kwaliteitsborging. Ik beschrijf in het hoofdstuk kwaliteitsborging van mijn plan van aanpak¹³ welke maatregelen er genomen zijn en worden om een kwalitatief goed eindproduct neer te kunnen zetten.

De vraag is echter: “wanneer is er sprake van een kwalitatief goed eindproduct”? Kwaliteit is immers niet concreet meetbaar. Ik definieer voor dit project een kwalitatief goed eindproduct als “een implementatieplan dat naadloos aansluit op de huidige infrastructuur en met weinig inspanning te realiseren is”. Concreet bedoel ik met “naadloos aansluiten op de huidige infrastructuur” dat er geen aanpassingen benodigd zijn aan het netwerk- of het Active Directory ontwerp. Met “weinig inspanning” bedoel ik dat bij gelijkblijvende omstandigheden het plan vrijwel direct uitgevoerd kan worden. Daarnaast houdt “kwalitatief goed” in dat het gehele proces voor de opdrachtgever inzichtelijk is en hij voldoende mogelijkheid heeft om het resultaat te beïnvloeden.

Door de regelmatige afstemming met de opdrachtgever bereik ik dat het proces inzichtelijk is en hij voldoende mogelijkheden heeft tot bijsturing. Het gebruik van door leveranciers goedgekeurde documenten levert op dat, ondanks het mogelijke gebrek aan ‘fysiek bewijs’, het implementatieplan toch vertrouwen wekt voor de daadwerkelijke implementatie.

In mijn kwaliteitsdefinitie staat dat het implementatieplan naadloos aansluit op de huidige infrastructuur. Ik ben zelf de beheerder van de huidige infrastructuur en er zijn geen externe partijen betrokken bij dit project. Ik hoef dus geen extra maatregelen te nemen om hieraan te kunnen voldoen.

¹² Zie paragraaf 2.4 “Activiteitenplanning definitiefase” van het plan van aanpak (bijlage 2)

¹³ Zie paragraaf 3.2 “Kwaliteitsborging” van het plan van aanpak (bijlage 2)

De risico's voor dit project zijn factoren die een bedreiging kunnen vormen voor het behalen van de projectdoelstellingen. Voor mijn project geldt dat door de beperkte scope (geen implementatie) en de kleine projectorganisatie de risico's beperkt zijn. Ik heb de risico's voor mijn project bepaald door twee keer voor mijzelf na te gaan wat er redelijkerwijs voor kan zorgen dat de doelstellingen van mijn project in gevaar komen. In mijn ervaring loop ik bij één keer nadenken een grote kans dingen over het hoofd te zien. Twee keer nadenken levert al snel nieuwe inzichten op. Bij drie keer nadenken komt er 'meer van hetzelfde'. De impact en de maatregelen zijn een logisch gevolg van de geïnventariseerde risico's. Een voorbeeld van een risico dat is opgenomen in het plan van aanpak¹⁴ is:

Risico: Geen hardware voor Proof of Concept

Omschrijving: Om een proof of concept opstelling te bouwen is hardware nodig. Zonder een proof of concept is het niet mogelijk om de realiseerbaarheid van het ontwerp in de praktijk aan te tonen.

Impact: Als de realiseerbaarheid van het ontwerp niet in de praktijk aangetoond wordt, dan moet de realiseerbaarheid door middel van literatuurstudies aangetoond worden, er blijft dan altijd een risico bestaan dat enkele onderdelen niet geverifieerd kunnen worden.

Maatregelen: Waar mogelijk wordt gebruik gemaakt van documentatie van leveranciers om de ontwerpen te maken.

¹⁴ Zie paragraaf 3.3 "Risicobeheersing" in het plan van aanpak (bijlage 2)

3. Definitiefase

De definitiefase is bedoeld om de huidige situatie en de eisen en wensen voor de nieuwe ontwerpen in kaart te brengen. Het eindproduct van deze fase is het definitierapport¹⁵.

Om de huidige toestand te beschrijven moet er informatie verzameld en verwerkt worden. In de eerste paragraaf ga ik dieper in op het informatie verzamelen en verwerken, in de volgende paragrafen ga ik dieper in op de keuzes die in het verleden voor de verschillende onderwerpen gemaakt zijn.

Belangrijk bij een onderzoek is het vaststellen van het onderzoeksdomein. In dit geval is het onderzoeksdomein gelijk aan de scope uit het plan van aanpak (mailomgeving, database servers en het virtualisatieplatform). In de komende paragrafen wordt het onderzoeksdomein verder ingevuld.

Met het oog op de leesbaarheid van het rapport heeft elk onderwerp uit dit onderzoeksdomein een eigen hoofdstuk gekregen in het definitierapport. Deze hoofdstukken uit het definitierapport komen ook weer terug in dit afstudeerverslag.

3.1. Informatie verzamelen en verwerken

Om voldoende informatie te verzamelen om de huidige toestand te beschrijven gebruik ik de volgende werkwijze:

1. Hoofdstukindeling bepalen
2. Tekst schrijven op basis van parate kennis
3. Vragen formuleren
4. Onderzoek doen
5. Tekst verder invullen

De hoofdstukindeling van het definitierapport is bedoeld om vooraf aan te geven wat beschreven moet worden en vooral wat er niet beschreven hoeft te worden, een verdere afbakening dus. Bij het bepalen van de hoofdstukindeling maak ik gebruik van meerdere informatiebronnen. In de eerste plaats heb ik van de hoofdonderwerpen uit mijn onderzoeksdomein de relaties met andere onderdelen van de ICT infrastructuur tot twee keer toe overdacht.

De tweede informatiebron die ik gebruikt heb bij het maken van de hoofdstukindeling is mijn verantwoordelijkheid als systeembeheerder bij Pimarox. Dat is (kort door de bocht) het ijzer, het besturingssysteem en de middleware (software die noodzakelijk is om de gevraagde functie uit te voeren).

¹⁵ Zie bijlage 3: “Definitierapport virtualiseren mail- en database services”

Op basis van bovenstaande informatiebronnen is voor de mailomgeving de hoofdstukindeling als volgt (de relevantie van de onderwerpen volgt in paragraaf 3.3):

1. Servers
2. Mailflow internet
3. Storage
4. Mail databases
5. Mailboxen
6. Public Folders
7. Retention, backup en restore
8. Beschikbaarheid mailservices
9. Connectiviteit gebruikers
10. Overige eisen en wensen

De hoofdstukindeling voor de database servers is (de relevantie van de onderwerpen volgt in paragraaf 3.4):

1. Opzet productiecluster
2. Storage
3. Databases
4. Ontsluiting databases
5. Beveiliging databases
6. Backup en recovery
7. Beschikbaarheid SQL databases
8. Overige eisen en wensen

Voor het virtualisatieplatform zijn de volgende onderwerpen belangrijk (de relevantie volgt in paragraaf 3.5):

1. Hardware
2. Networking
3. Storage
4. Aandachtspunten migratie

Het netwerk en de Active Directory zijn belangrijke relaties van alle onderwerpen van mijn onderzoeksdomein. Om te voorkomen dat ik deze onderdelen twee of drie keer moet beschrijven in het definitierapport voeg ik aan het definitierapport een extra hoofdstuk toe. (Zie volgende paragraaf)

3.2. Het netwerk

Bij het beschrijven en beoordelen van een Exchange ontwerp of een SQL server ontwerp is het noodzakelijk om ook inzicht te hebben in de structuur van het netwerk en de Active Directory, zo weet ik uit ervaring. Zonder dit inzicht is het moeilijk, zo niet onmogelijk, om de achterliggende redenen van de gemaakte keuzes te begrijpen.

Om te beoordelen wat ik van het netwerk en de Active Directory moet beschrijven kijk ik naar de relaties met het primaire onderzoeksdomein. Op basis hiervan kom ik tot de volgende paragrafen in dit hoofdstuk¹⁶:

- | |
|--|
| <ol style="list-style-type: none">1. Gebouwen2. Subnetten3. Routing4. Fysieke verbindingen5. Beveiliging6. Active Directory |
|--|

Ad 1. De gebouwen zijn relevant omdat een Exchange ontwerp aandacht moet besteden aan de bereikbaarheid van de mailservices voor alle gebruikers. Het is dus noodzakelijk om te weten waar deze gebruikers zich bevinden en hoe de fysieke netwerkinfrastructuur op hun locatie er uit ziet.

Ad 2. De subnetten zijn relevant omdat hiermee duidelijk wordt gemaakt welke logische onderverdeling gemaakt is in het netwerk.

Ad 3. De routing sluit aan op de subnetten. Zonder routing zijn er geen gescheiden subnetten mogelijk.

Ad 4. De fysieke verbindingen geven weer richting aan de ontwerpkeuzes die in een later stadium gemaakt moeten worden, bijvoorbeeld voor de plaatsing van centrale apparatuur.

Ad 5. In een aantal gevallen brengt de aanwezige netwerkbeveiliging ontwerpkeuzes met zich mee. Ik heb daarom de netwerkbeveiliging besproken in dit hoofdstuk.

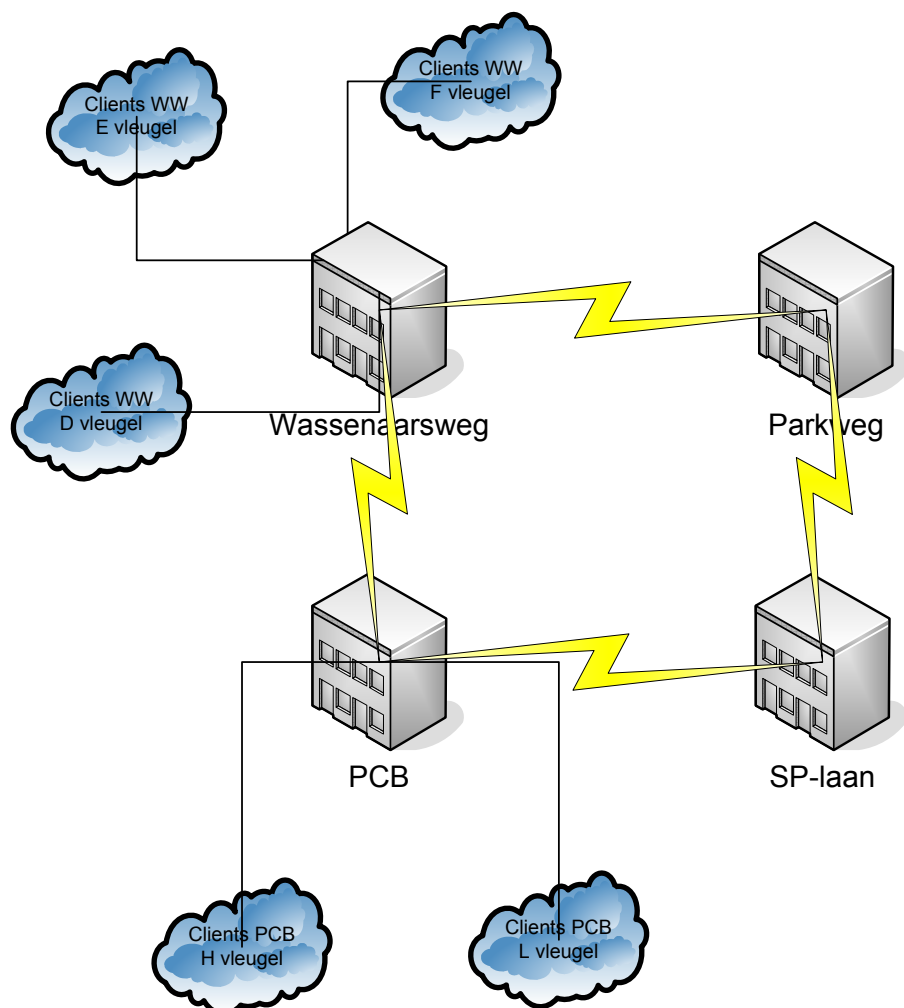
Ad 6. Zowel Microsoft Exchange server als Microsoft SQL server zijn afhankelijk van Active Directory. In het geval van Microsoft Exchange server is het ontwerp van Active Directory bepalend voor het ontwerp van Exchange. Door in de definitiefase hier aandacht aan te besteden, beperk ik de kans dat in een later stadium het ontwerp of implementatieplan niet realiseerbaar blijkt te zijn.

Voorbeelden van onderwerpen die geen duidelijke relatie met het onderzoeksdomein hebben en daarom buiten beschouwing gelaten zijn:

- Merk en type van de core- en access switches;
- OU structuur van de Active Directory.

Het netwerk van Pimarox is één van mijn verantwoordelijkheden als systeembeheerder. De aanwezige tekeningen waren echter niet heel geschikt voor het definitierapport (te gedetailleerd). Daarom heb ik een nieuwe tekening gemaakt met daarin de relevante informatie overzichtelijk gepresenteerd:

¹⁶ Zie hoofdstuk 1 “Netwerk” van het definitierapport (bijlage 3)



Figuur 2: Netwerk Pimarox

Uit de bovenstaande tekening blijkt dat Pimarox vier gebouwen gebruikt. In het vervolg van het project gebruik ik afkortingen voor deze gebouwen:

Wassenaarseweg: WW
 Parkweg: PW
 PC Boutenselaan: PCB
 Schimmelpennicklaan: SP-Laan

3.3. Mailomgeving

De bedoeling van dit hoofdstuk¹⁷ in het definitierapport is om voor mij en de opdrachtgever inzichtelijk te maken hoe de huidige mailomgeving, gebaseerd op Exchange Server 2003, nu in elkaar zit en welke functionaliteiten dit systeem nu levert. Deze functionaliteiten moeten namelijk terug komen in het nieuwe ontwerp, dat gebaseerd zal zijn op Exchange Server 2010. Eerder is immers vastgesteld dat er behoefte is aan vervanging van de huidige versie (zie paragraaf 1.2).

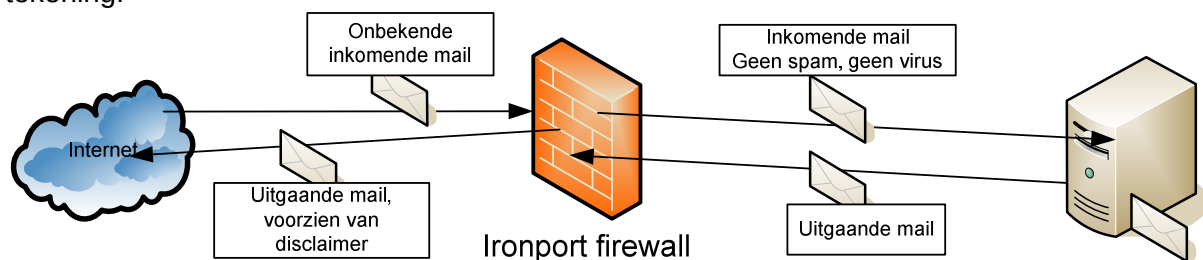
¹⁷ Zie hoofdstuk 2 “Mailomgeving” in het definitierapport (bijlage 3)

Op basis van mijn kennis en ervaring met de omgeving van Pimarox en het ontwerpen en beheren van Microsoft Exchange mailomgevingen heb ik bepaald welke onderdelen relevant genoeg zijn om in dit hoofdstuk te bespreken. Dat zijn:

1. Servers
2. Mailflow internet
3. Storage
4. Mail databases
5. Mailboxen
6. Public Folders
7. Retention, Backup en restore
8. Beschikbaarheid mailservices
9. Connectiviteit gebruikers
10. Overige eisen en wensen

Ad 1: De servers zijn relevant om te bepalen welke ontwerpkeuzes in het verleden gemaakt zijn.

Ad 2: Ik weet dat er in nieuwe versies van Exchange Server (grote) wijzigingen zijn in de wijze waarop de mailrouting van en naar het internet verloopt. Ik beschrijf de routing om daar in het ontwerpproces rekening mee te kunnen houden. Ter illustratie gebruik ik deze tekening:



Figuur 3: Mailrouting internet

Ad 3: Een 'hot item' bij Pimarox is storage. De nieuwe architectuur van Exchange 2010 (zie ook hoofdstuk 4) heeft ook grote gevolgen voor de hoeveelheid benodigde storage.

Ad 4, 5, 6: De mail databases, mailboxen en public folders vormen het resultaat van het systeem 'Exchange 2003' en moeten daarom onderzocht worden.

Ad 7: De configuratie op dit onderdeel heeft gevolgen voor de hoeveelheid storage die benodigd is. Als ik hier geen aandacht aan zou besteden in mijn onderzoek, dan heeft dat tot gevolg dat de storage calculatie voor het nieuwe ontwerp aan betrouwbaarheid inlevert.

Ad 8: Met de beschikbaarheid van de mailservices probeer ik duidelijkheid te krijgen over de afspraken die er zijn met de gebruikersorganisatie.

Ad 9: De wijze waarop de gebruikers verbinding maken met de Exchange 2003 omgeving kan invloed hebben op het ontwerp of de implementatie. Door dit nu te onderzoeken voorkom ik verrassingen in een later stadium.

Ad 10: In deze paragraaf som ik de eisen en wensen die aan het nieuwe ontwerp en de implementatie worden gesteld op.

Een voorbeeld van een onderwerp dat niet relevant is om in dit hoofdstuk te bespreken is journaling¹⁸ in Exchange, want dat wordt niet gebruikt bij Pimarox.

Ik ben eerst gaan schrijven op basis van mijn parate kennis over dit onderwerp. Op basis van deze resultaten heb ik voor mijzelf een vragenlijst gemaakt voor de nog ontbrekende informatie, voornamelijk details als exacte naamgeving en IP adressen. Om het gebruik van de mailboxen in kaart te brengen heb ik een export gemaakt van een aantal statistische gegevens uit Exchange 2003. De export bevat de volgende gegevens per mailbox:

- Naam
- account laatste log on;
- Grootte (in KB);
- Aantal items;
- Datum en tijd laatste log on;
- Datum en tijd laatste log off.

Deze export heb ik geïmporteerd in Excel om ze te kunnen analyseren. Aan de hand van de analyse kan ik gemiddelden berekenen voor het gebruik van Exchange.

Een belangrijk resultaat uit het onderzoek naar de connectiviteit van de gebruikers dat er geen gebruik wordt gemaakt van Outlook Web Access. Als Outlook Web Access wel in gebruik is, dan moet daar in de migratie naar Exchange 2010 op geanticipeerd worden.

Een aantal ontwerpkeuzes die in het verleden gemaakt zijn worden nu door het beheer in twijfel getrokken. Ik heb getwijfeld of ik deze vraagtekens in het definitierapport moet opnemen. Ik heb de voor- en nadelen van wel opnemen tegen elkaar afgewogen:

Nadeel: Doelstelling van het definitierapport is het beschrijven van de huidige situatie om daarmee de functionele eisen en wensen te achterhalen. Twijfels over het huidige ontwerp hebben geen invloed op de huidige functionaliteit en horen niet thuis in het definitierapport.

Voordeel: Door aandacht te besteden aan de twijfels kan ik het nieuwe ontwerp mogelijk verbeteren, maar in elk geval de acceptatie van het ontwerp door de beheerorganisatie verbeteren.

Ik vind acceptatie van het nieuwe ontwerp dusdanig belangrijk dat ik de twijfels dus wel terug laat komen in het definitierapport (zie ook hoofdstuk 4).

Het resultaat is een beschrijving van de huidige mailomgeving van Pimarox waarbij ook aandacht wordt besteed aan ontwerpkeuzes die in het verleden gemaakt zijn en die dus van invloed zijn op het toekomstige ontwerp.

¹⁸ Journaling is het archiveren van binnenkomende en uitgaande mails.

3.4. SQL omgeving

De bedoeling van dit hoofdstuk¹⁹ in het definitierapport is om voor mij en de opdrachtgever inzichtelijk te maken hoe de huidige SQL omgeving nu in elkaar zit en welke functionaliteiten dit systeem nu levert. Deze functionaliteiten moeten namelijk terug komen in het nieuwe ontwerp.

In vergelijking met Exchange is de bepaling van de ontwerpscope bij SQL server lastiger voor mij. De betrokken partijen bij een Exchange omgeving zijn technisch beheer en de gebruikersorganisatie. Interactie met andere applicaties gaat doorgaans via gestandaardiseerde protocollen (http, MAPI/RPC en SMTP). Ontwerpkeuzes die daarmee samenhangen zijn simpel. Wel ondersteunen, of niet ondersteunen.

Bij SQL server is het niet zo eenvoudig. Om te beginnen is er doorgaans geen directe relatie tussen SQL server en de eindgebruiker. Ook wordt de inrichting van een SQL server grotendeels bepaald door de applicaties die er gebruik van willen maken. Bij Pimarox is dit krachtenveld tussen applicaties en infrastructuur sterk in beweging en de neiging bestaat om bij gebrek aan duidelijkheid te wachten.

Om een deadlock waarbij iedereen op elkaar wacht te voorkomen, heb ik in overleg met de opdrachtgever besloten om mijzelf te beperken tot die onderdelen van een infrastructuur waarvoor systeembeheer verantwoordelijk is. Het nadeel van deze keuze is dat er mogelijk in een later stadium alsnog wijzigingen in de applicaties aangebracht moeten worden, maar het voordeel is dat er vanuit ICT beheer wel duidelijkheid gegeven kan worden aan de rest van de organisatie over de mogelijkheden van de SQL infrastructuur en de wijze waarop invulling wordt gegeven aan hoge beschikbaarheid. In hoofdstuk 4 kom ik nog terug op de definitie en invulling van hoge beschikbaarheid bij Pimarox.

Om het onderzoek af te bakenen zoek ik naar de onderdelen die op basis van bovenstaande afwegingen voor mijn project relevant zijn. Dat zijn:

1. Opzet productiecluster
2. Storage
3. Databases
4. Ontsluiting databases
5. Beveiliging databases
6. Back-up en recovery
7. Beschikbaarheid SQL databases
8. Overige eisen en wensen

Ad 1: Ik weet al uit ervaring dat er sprake is van een geclusterde SQL server met meerdere instances. Een SQL server instance is een verzameling services die gezamenlijk alle taken voor een SQL server verrichten. Op een enkele Windows server kunnen meerdere instances geconfigureerd zijn. Ze werken onafhankelijk van elkaar, maar zijn natuurlijk wel afhankelijk van hetzelfde besturingssysteem en software. Het is daarom belangrijk om nader te beschrijven hoe de instances verdeeld zijn over de fysieke servers.

Ad 2: Een ‘hot item’ bij Pimarox is storage. De vereiste virtualisatie (zie het projectplan) heeft ook grote gevolgen voor de storage configuratie. Ik onderzoek daarom hoeveel storage de huidige SQL-omgeving gebruikt. Dit geeft enig inzicht in het gebruik van deze omgeving.

¹⁹ Zie hoofdstuk 2 “SQL server” in het definitierapport (bijlage 3)

Ad 3, 4, 5: De (ontsluiting en beveiliging van) databases vormen het hart van de geleverde functionaliteiten. Het spreekt voor zich dat het ontwerp hierop moet aansluiten. Ik onderzoek dan ook welke databases aanwezig zijn en hoe deze in de huidige situatie worden gebruikt.

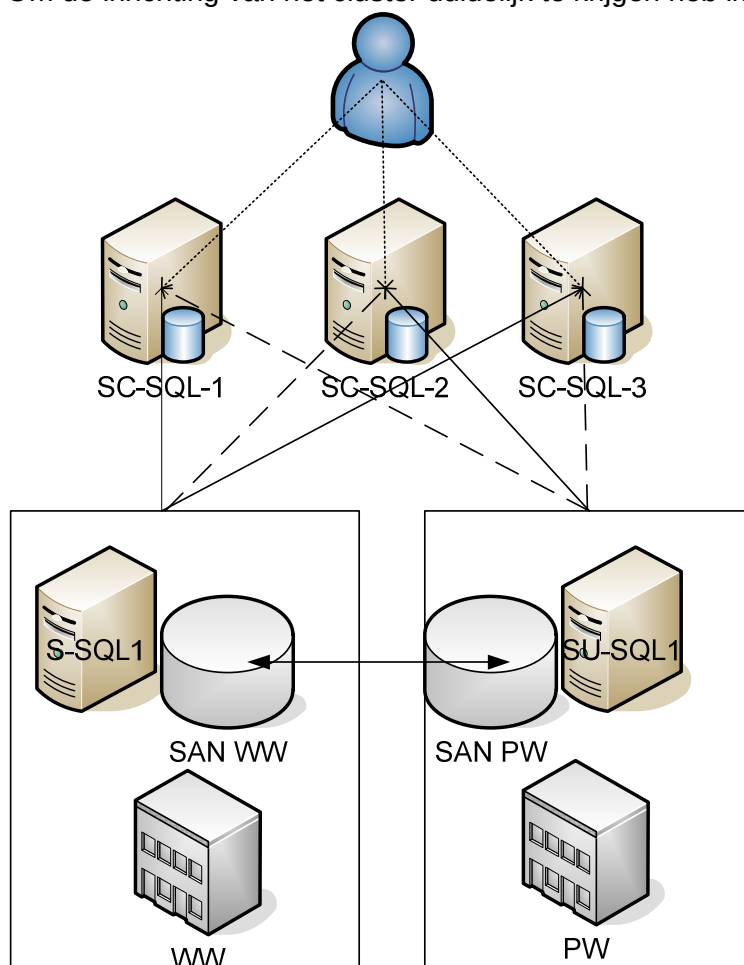
Ad 6: De wijze waarop de back-ups gemaakt worden kan invloed hebben op de hoeveelheid benodigde storage, weet ik uit ervaring. Ik onderzoek dit op deze plaats, om verrassingen bij implementatie te voorkomen.

Ad 7: Het onderzoek naar de beschikbaarheid van de databases helpt om de meest geschikte configuratie voor hoge beschikbaarheid te bepalen.

Ad 8: In deze paragraaf som ik de eisen en wensen die aan het nieuwe ontwerp en de implementatie worden gesteld op.

Een voorbeeld van een onderwerp dat niet relevant genoeg is om te beschrijven in het definitierapport is het gebruik van database jobs. Database jobs kunnen van invloed zijn op de migratie naar SQL server 2008, maar het functioneren van de database jobs is bij Pimarox geen verantwoordelijkheid van systeembeheer en is dus niet relevant voor mijn onderzoek.

Net als bij het onderzoeken van de mailomgeving ben ik eerst gaan schrijven op basis van mijn eigen kennis. Voor de ontbrekende vragen heb ik een vragenlijst naar mijzelf gestuurd. Om de inrichting van het cluster duidelijk te krijgen heb ik een tekening gemaakt:



Figuur 4: Inrichting SQL productie cluster

Het resultaat van dit onderzoek is een beschrijving van het huidige SQL 2000 productiecluster, inclusief de databases die daar nu op draaien. Op basis van deze beschrijving bepaal ik de eisen en wensen voor het nieuwe ontwerp (zie hoofdstuk 4).

3.5. Virtualisatieplatform

Het is niet de bedoeling om een nieuw ontwerp te maken voor het virtualisatieplatform. De bedoeling van mijn onderzoek naar het virtualisatieplatform is om later vast te kunnen stellen of mijn ontwerp realiseerbaar is op het virtualisatieplatform van PimaroX.

Ik bepaal op basis van bovenstaande afweging welke onderdelen relevant zijn voor mijn onderzoek. Dat zijn:

1. Hardware
2. Networking
3. Storage
4. Aandachtspunten migratie

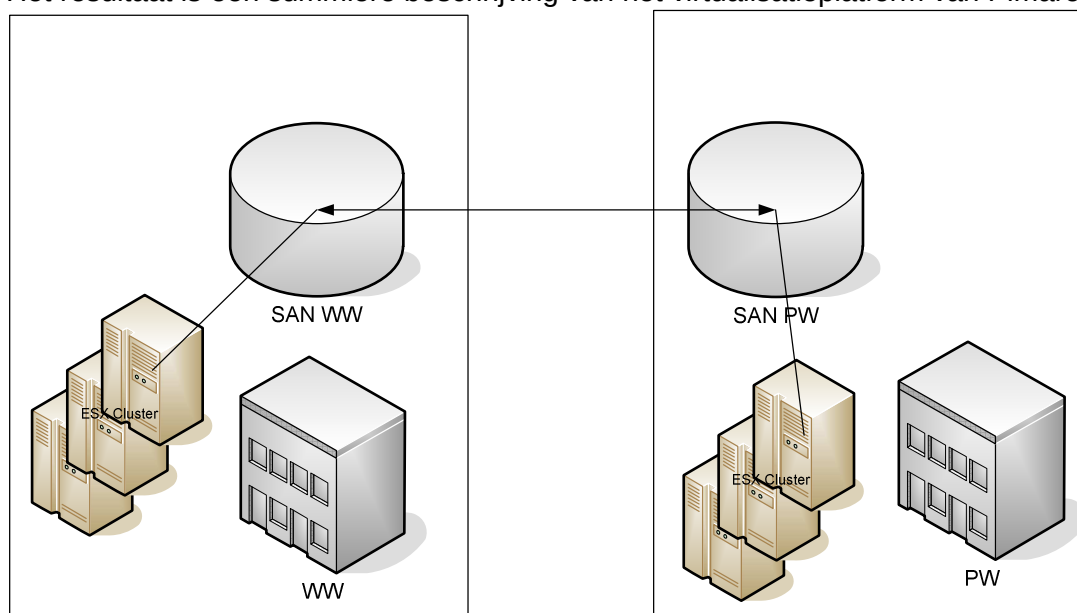
Ad 1: De fysieke hardware bepaalt wat de capaciteiten van het virtualisatieplatform zijn. Voor Exchange Server 2010 is vooral van belang hoeveel megacycles de processoren kunnen leveren.

Ad 2: Ik wil vaststellen of alle benodigde subnetten beschikbaar zijn op het virtualisatieplatform.

Ad 3: Bij het ontwerp moet ook aan virtuele machines storage toegekend worden. Daarom is het van belang om te onderzoeken welke vormen van storage mogelijk zijn binnen het virtualisatieplatform.

Ad 4: Ik vraag mij af welke aandachtspunten voortvloeien uit het huidige gebruik van het virtualisatieplatform.

Het resultaat is een summiere beschrijving van het virtualisatieplatform van PimaroX²⁰.



Figuur 5: vSphere clusters PimaroX

²⁰ Zie hoofdstuk 4 “Virtualisatieplatform” van het definitierapport (bijlage 3)

Voor dit onderzoek heb ik gebruik gemaakt van de vSphere client waarin de eerste drie onderwerpen van dit onderzoek geconfigureerd en beheerd worden.

Ik heb in mijn onderzoek geen aandacht besteed aan overcommitment van processoren en geheugen, hoewel dat over het algemeen wel belangrijke ontwerpcriteria zijn. Overcommitment wil zeggen dat er meer fysiek (RAM) geheugen en processorcores verdeeld zijn dan er fysiek aanwezig zijn.

Er worden regelmatig nieuwe vm's (virtuele machines) aangemaakt. Daarnaast worden de vm's ook regelmatig opnieuw verdeeld tussen de hoofd- en uitwijklocatie. Het gevolg is dat een meting van de overcommitment van processoren en geheugen alleen nuttig is bij het aanmaken en alloceren van de vm's. Overcommitment is daarom niet relevant voor mijn onderzoek.

3.6. Planning volgende fase

De laatste stap in de definitiefase is het maken van een planning voor het vervolg²¹. Tijdens de gesprekken met de opdrachtgever is dit al besproken, waarop de opdrachtgever met het voorstel kwam om maar één ontwerp te maken zodat hij geen keuze hoeft te maken.

Oorspronkelijk was mijn plan om verschillende ontwerpen te maken, deze onderling te vergelijken en het geheel aan de opdrachtgever voor te leggen zodat hij een keuze kan maken. De opdrachtgever stelde voor dat ik maar één ontwerp zou maken en deze inclusief onderbouwing ter goedkeuring voorleg aan de opdrachtgever. Het voordeel van deze werkwijze voor mij is dat ik maar één ontwerp hoeft te maken. Een ander voordeel is dat er meer recht wordt gedaan aan mijn expertise op het gebied van ontwerpen.

Het gevolg is een nieuwe activiteitenplanning voor de ontwerpfase:

Activiteit:	Doel:
Vaststellen ontwerpcriteria	Het vaststellen van de criteria waaraan de ontwerpen moeten voldoen.
Exchange server ontwerpen	Maken van het ontwerp voor de gevirtualiseerde Exchange services
SQL ontwerpen	Maken van het ontwerp voor de gevirtualiserde SQL services
Impact VMware bepalen	Bepalen van de gevolgen voor de Vmware omgeving.

Tabel 3: Globale activiteitenplanning ontwerpfase

De ontwerpcriteria zijn deels de eisen die worden gesteld door de softwareleveranciers in het kader van de ondersteuning. Voor een ander deel komen de criteria voort uit de resultaten van de definitiefase.

²¹ Zie hoofdstuk 5 “Planning volgende fase” van het definitierapport (bijlage 3)

4. Ontwerpfase

In de definitiefase is de huidige toestand van de mail- en database services bepaald. Het doel van de ontwerpfase is het opnieuw ontwerpen van deze services zodat dezelfde functionaliteiten geleverd kunnen worden door een nieuw platform. Het eindproduct van deze fase is het ontwerprapport²² waarin de nieuwe ontwerpen uitgewerkt zijn.

Ik weet dat ik te weinig kennis van Microsoft SQL server heb om te kunnen bepalen wat belangrijke factoren voor een ontwerp zijn. Ik begin dus met kennis vergaren.

4.1. Verzamelen kennis

Om kennis over SQL server te kunnen vergaren zijn er veel verschillende hulpmiddelen voor handen. Ik heb eerst een grofmazige indeling gemaakt:

1. Een cursus;
2. Uitleg van ervaren SQL beheerders;
3. Boeken.

Het voordeel van een cursus is dat er in een beperkte tijd veel informatie, inclusief “hands-on training” wordt aangeboden. Uit een zoektocht onder cursusaanbieders blijkt dat de bestaande cursussen vooral gericht zijn op het vakgebied van de database administrator. Dat is nu net niet waar ik naar op zoek ben. Daarnaast zijn cursussen behoorlijk duur, veel duurder dan de andere alternatieven.

Het voordeel van uitleg door ervaren SQL beheerders is dat de aangeboden informatie beter toegespitst kan worden op mijn specifieke situatie. Het nadeel is echter dat de aangeboden informatie niet per definitie overeenkomt met de inzichten van Microsoft, de software leverancier. Het recht op ondersteuning door Microsoft zou daardoor in gevaar kunnen komen.

Het voordeel van boeken is dat ik geheel op eigen tempo de benodigde kennis kan verzamelen en dat er velen beschikbaar zijn. De nadelen van boeken is echter dat ze vaak ook gericht zijn op het vakgebied van de database administrator en de mening van de auteur of uitgeverij verkondigen.

Vanwege prijs en beschikbaarheid kies ik voor boeken. De beschreven nadelen van boeken vang ik op door te kiezen voor drie boeken die elkaar deels overlappen. Twee boeken zijn uitgegeven door Microsoft zodat het recht op ondersteuning niet in gevaar komt.

Ik heb uiteindelijk gekozen voor een drietal boeken die elkaar aanvullen:

1. Een boek over SQL server 2008 High Availability²³;
2. Een cursusboek over implementatie en onderhoud van een SQL server omgeving²⁴;
3. Een boek met nuttige informatie over beheer en gebruik van SQL server²⁵.

²² Zie bijlage 4: “Ontwerprapport virtualiseren mail- en database services”

²³ “SQL server 2008: high availability with clustering & database mirroring”, Michael Otey, uitg: McGraw-Hill

²⁴ “SQL server 2008: implementation & maintenance”, Mike Hotek, uitg: Microsoft Press

²⁵ “SQL Server 2008 Administrator's Pocket Consultant, second edition (updated for R2)”, William R. Stanek, uitg: Microsoft Press

4.2. Bepalen eisen en wensen

Ik begin het ontwerpproces met het helder maken van de eisen en wensen. De eisen en wensen zijn geïnventariseerd in het definitierapport. Het hoofdstuk ontwerpcriteria in het ontwerprapport²⁶ bevat dus geen nieuwe informatie, maar is een samenvatting van de eisen en wensen zoals die in het definitierapport verwoord zijn.

Ik maak in mijn ontwerprapport een onderscheid tussen eisen en wensen.

Zoals de termen al aangeven zijn de eisen criteria die in elk geval in het ontwerp terug moeten komen. Wensen zijn criteria waar in elk geval aandacht aan besteed moet worden, maar die mogelijk niet (geheel) gerealiseerd kunnen worden.

4.3. Ontwikkeling SQL ontwerp

Om tot een definitief SQL ontwerp²⁷ te komen ga ik in een aantal stappen van een high level ontwerp naar een gedetailleerd ontwerp. Het high level ontwerp besteedt aandacht aan zaken als load-balancing en hoge beschikbaarheid. In het gedetailleerd ontwerp komen ook andere criteria zoals scheiding van database- en logbestanden en beperkte servertoegang aan de orde.

De belangrijkste eis voor het high level design is hoge beschikbaarheid. Om hoge beschikbaarheid te bereiken is redundantie noodzakelijk. Het ontwerp is redundant als alle onderdelen dubbel aanwezig zijn.

Er is geen gangbare definitie over hoge beschikbaarheid die ook op Pimarox van toepassing is. Belangrijke variabelen bij hoge beschikbaarheid zijn de gevraagde beschikbaarheid (in percentage) en de storingsen (aantal en / of impact) die opgevangen moeten kunnen worden. Er zijn bij Pimarox geen waarden bepaald voor deze variabelen. Het begrip “hoge beschikbaarheid” is dus in principe niet gedefinieerd.

Ik hanteer in overleg met de opdrachtgever het volgende criterium: “De architectuur is hoog beschikbaar als de uitval van een compleet datacentre opgevangen kan worden, terwijl de impact in termen van hersteltijd en dataverlies minimaal is.”

Het is technisch niet mogelijk om uitval op te vangen zonder enige impact. Wel kunnen er maatregelen genomen worden waardoor de dienstverlening snel hervat wordt. Het verlies aan data moet minimaal zijn. De snelheid waarmee de dienstverlening hervat wordt en het verlies aan data bepalen dus de hoge beschikbaarheid.

De belangrijkste keuze die in dit verband gemaakt moet worden is: “welke technische oplossing ga ik toepassen om het ontwerp hoog beschikbaar te maken?”

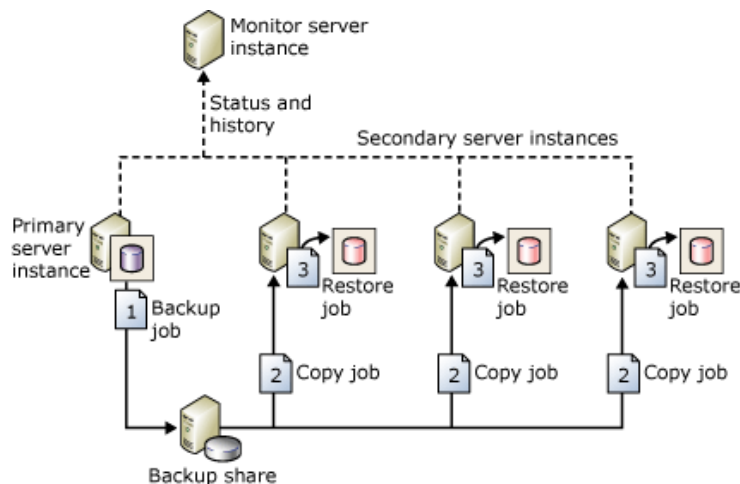
²⁶ Zie hoofdstuk 1 “ontwerpcriteria” in het ontwerprapport (bijlage 4)

²⁷ Zie hoofdstuk 2 “SQL ontwerp” in het ontwerprapport (bijlage 4)

Uit de geraadpleegde literatuur (Zie paragraaf 4.1) is gebleken dat een drietal technieken in aanmerking komen om het ontwerp te laten voldoen aan de vereiste hoge beschikbaarheid. Dat zijn:

1. Log Shipping;
2. Shared Storage clustering;
3. Database mirroring.

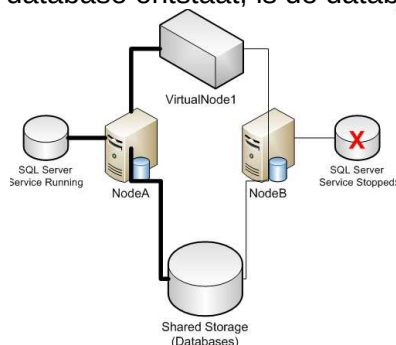
Ad 1. Log shipping is een techniek waarbij het logbestand van een database (hierin zijn alle transacties vastgelegd) periodiek (in principe elke 15 minuten) wordt gekopieerd naar een andere server. Daar wordt het logbestand opnieuw toegepast op de database.



Figuur 6: Schematische weergave Log shipping (Bron: Microsoft)

Log shipping voldoet aan de eis van redundantie, omdat de database op verschillende, van elkaar onafhankelijke servers of vm's, geplaatst is. Als de primaire database (of de server / vm) uitvalt is er echter geen mogelijkheid de partner database zonder tussenkomst van een beheerder op te brengen. Het dataverlies is dan hooguit de geconfigureerde vertraging (standaard een kwartier). Deze oplossing is dus wel redundant, maar door de mogelijk forse hersteltijd en dataverlies niet hoog beschikbaar.

Ad 2. Shared storage clustering maakt gebruik van een windows shared storage cluster. Een windows shared storage cluster kan gevormd worden als twee (of meer) windows servers toegang hebben tot dezelfde gedeelde storage. Theoretisch zijn deze servers verbonden met dezelfde SAN oplossing, maar Pimarox beschikt over een techniek waardoor een shared storage cluster ook te gebruiken is in verschillende vestigingen²⁸. Het nadeel van een shared storage cluster is echter dat de database maar 1 keer bestaat. Indien er dus corruptie aan de database ontstaat, is de database daarmee niet beschikbaar.

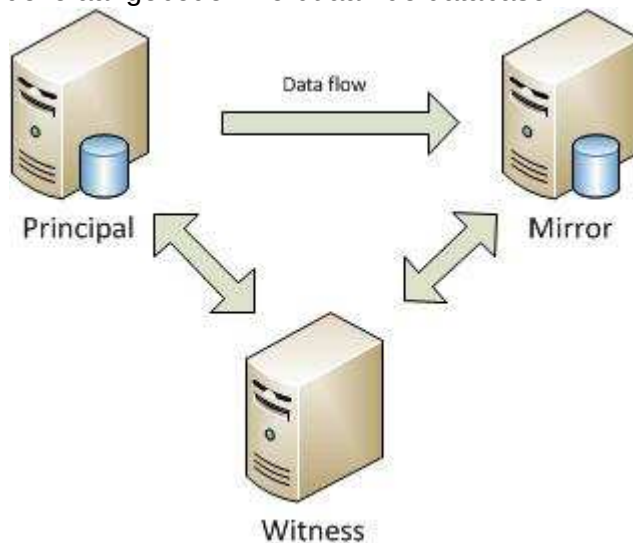


Figuur 7: Schematische weergave Shared storage clustering (Bron: Microsoft)

²⁸ Zie voor meer informatie de paragrafen 2.3 en 3.2 uit het definitierapport

Als de actieve server uitvalt, zal de andere server proberen om de gedeelde resources (storage) zich toe te eigenen en vervolgens worden de SQL services en de database opgestart. Indien de actieve server met een transactie bezig was op het moment van uitvallen, dan zal de database eerst hersteld moeten worden voordat de database weer beschikbaar gesteld kan worden. Uit de geraadpleegde bronnen²⁹ blijkt dat Microsoft geen ondersteuning geeft voor geclusterde servers waarvoor VMware HA of DRS geconfigureerd is.

Ad 3. Database mirroring is in de basis dezelfde techniek als log shipping, maar er zijn wel wat verschillen. Net als bij log shipping zijn er twee servers die allebei over de database beschikken, maar het belangrijkste verschil is dat elke transactie gerepliceerd wordt zodra deze aangeboden wordt aan de database.



Figuur 8: Schematische weergave database mirroring (Bron: Microsoft)

Database mirroring in combinatie met een ‘witness server’ zorgt ervoor dat de stand-by database automatisch actief kan worden als de primaire database uitvalt. De hersteltijd is korter dan bij een shared storage cluster. Er zijn namelijk geen gedeelde resources die overgenomen moeten worden, tevens is er geen risico op onvoltooide transacties in de database die eerst terug gedraaid moeten worden.

Om een definitieve keuze te maken ga ik de alternatieven wegen. Het meest voor de hand liggende wegingcriterium “toepasbaarheid huidige infrastructuur”³⁰ heeft geen zin, omdat alle alternatieven in principe even goed toepasbaar zijn in de huidige infrastructuur. Ik ga de alternatieven vergelijken op basis van criteria die wel relevant en onderscheidend zijn:

- | |
|---|
| <ul style="list-style-type: none">a) Toepasbaarheid in een gevirtualiseerde omgevingb) Risico op en omvang van dataverliesc) Licentiekostend) Hersteltijd. |
|---|

Ad a) Doel van het project is dat de SQL services gevirtualiseerd worden. In principe zijn alle drie de alternatieven geschikt voor toepassing in een gevirtualiseerde infrastructuur, maar er zijn wel verschillen. Een Shared Storage cluster is minder geschikt voor virtualisatie omdat Microsoft een aantal belangrijke VMware-features niet ondersteunt.

²⁹ “SQL server on Vmware; availability and recovery options”, PDF uitgave van VMware

³⁰ Zie paragraaf 2.3 van dit document

Ad b) Enig dataverlies is niet altijd te voorkomen, maar hoe minder, hoe beter. Bij Log Shipping is het risico op dataverlies het grootst. Ook is bij deze techniek de mogelijke omvang van dataverlies het grootst.

Ad c) Wat voor Windows server licentie is er benodigd? Een fail-over cluster, gebaseerd op shared storage vereist een Windows server enterprise editie. De overige alternatieven hebben voldoende aan een (goedkopere) standaard editie³¹.

Ad d) Hoeveel tijd is er nodig voor de services weer beschikbaar zijn. Op basis van mijn ervaring kom ik tot de volgende verdeling:

- Actief maken van een standby database (bij mirroring) is het snelst (enkele seconden);
- Gevolgd door automatisch starten van de SQL services en databases (bij een shared storage cluster) op een andere server (ong. 2 minuten);
- Handmatig actief maken van een standby database (bij log shipping) kost de meeste tijd (afhankelijk van notificatie en reactietijd).

In onderstaande tabel zijn de alternatieven voor het high level design onderling vergeleken op basis van de bovenstaande argumenten. De totaalscore bestaat uit de som van het aantal plusjes.

	Toepas- baarheid	Data- verlies	Licentie- kosten	Herstel -tijd	Totaal
Shared storage	-	+	-	-	1
Mirroring	+	-	+	+	3
Log shipping	+	-	+	-	2

Tabel 4: Vergelijking SQL HA alternatieven

Op grond van bovenstaande vergelijking valt voor het high level design de keuze op een architectuur gebaseerd op database mirroring.

4.4. Uitwerking van het high level design³².

In het high level design is bepaald hoe de architectuur er globaal uit moet komen te zien. Bij het uitwerken van dit design komt als eerste de vraag hoe de huidige applicaties verdeeld kunnen worden over de servers en hoe de benodigde server rollen verdeeld worden.

In de huidige situatie zijn er drie applicaties afhankelijk van de SQL omgeving. Los van de discussie of deze applicaties gemigreerd worden zal ik in elk geval uitwerken hoe deze applicaties gemigreerd kunnen worden. In de huidige situatie is er sprake van één fail-over cluster met daarop drie SQL instances.

Een instance is het kleinste niveau binnen SQL server waarop resource management kan plaatsvinden.

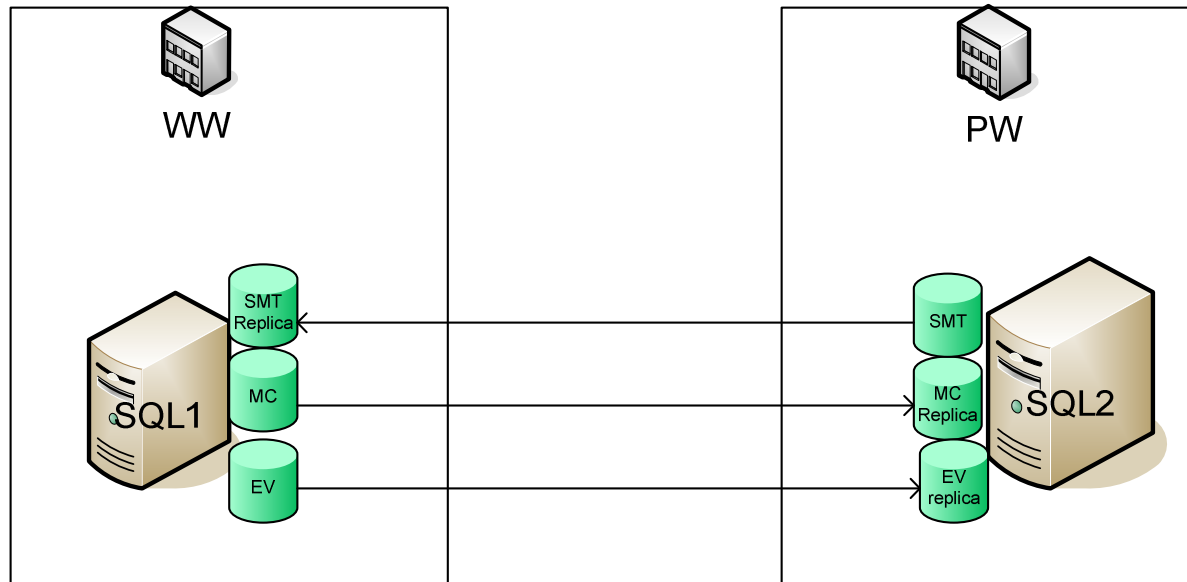
³¹ Zie: <http://www.microsoft.com/windowsserver2008/en/us/r2-compare-specs.aspx>

³² Zie paragraaf 2.1 "high level design" van het ontwerprapport (bijlage 3)

In de nieuwe configuratie zal er gebruik worden gemaakt van database mirroring. Op basis van de keuze voor database mirroring zijn er drie mogelijke alternatieven om de databases te hosten:

1. Alle databases in een enkele SQL instance;
2. Alle databases verdeeld over een drietal instances op een tweetal servers;
3. Elke applicatie z'n eigen servers;

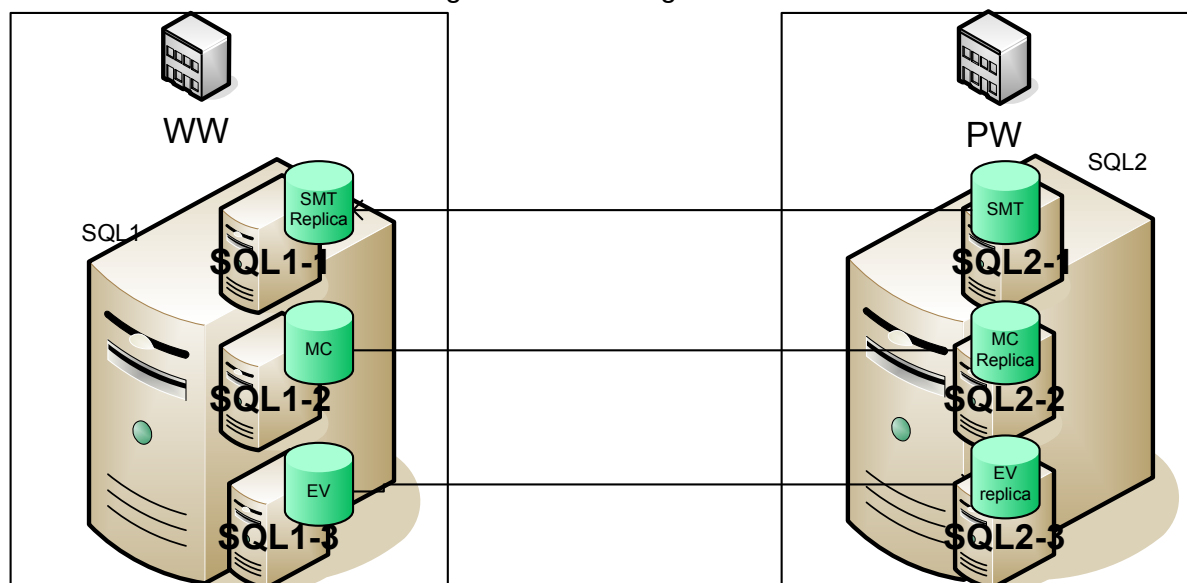
Ad 1. Schematisch ziet deze configuratie er als volgt uit:



Figuur 9: Schematische weergave SQL alternatief 1

In deze configuratie is er sprake van twee virtuele machines (SQL1 en SQL2) die beide voldoende gedimensioneerd zijn om alle databases te kunnen draaien. Er kan geen resource management plaatsvinden op het niveau van de database omdat er maar één SQL instance aanwezig is. Deze architectuur kent geen witness server. Bij implementatie moet dan besloten worden of daar een extra virtuele machine voor ingericht wordt, of dat de witness server rol op andere wijze ingevuld kan worden.

Ad 2. Schematisch ziet deze configuratie er als volgt uit:

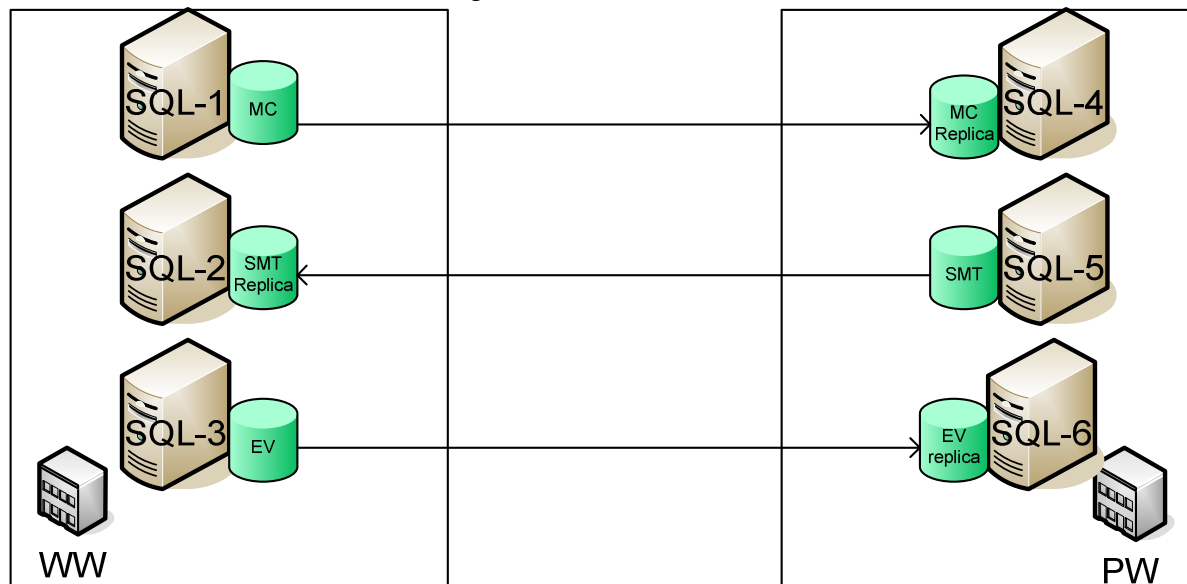


Figuur 10: schematische weergave alternatief 2

Bij deze architectuur is er sprake van twee virtuele machines (SQL1 en SQL2) die beide voldoende gedimensioneerd zijn om alle databases te kunnen hosten, maar door het gebruik van meerdere instances (SQL1-1, SQL1-2 etc) per server zijn er duurdere versies van SQL server nodig³³.

Deze architectuur maakt het mogelijk om enig resource management op het niveau van de applicatie uit voeren, maar de instances op één virtuele machine zijn wel afhankelijk van dezelfde software en hetzelfde besturingssysteem.

Ad 3. Schematisch ziet dit er als volgt uit:



Figuur 11: Schematische weergave SQL alternatief 3

Bij deze architectuur is er spraken van zes virtuele machines (SQL1-6), evenredig verdeeld over de beide locaties. De virtuele machines worden kunnen dusdanig gedimensioneerd worden dat ze in staat zijn hun eigen database te draaien.

Elke SQL server draait hier z'n eigen instance. Hierdoor zijn de instances niet afhankelijk van hetzelfde besturingssysteem of software.

Het kwaliteitscriterium “naadloos aansluiten op de huidige infrastructuur” is in dit geval niet voldoende onderscheidend. Om toch een keuze uit deze drie alternatieven te maken ga ik de alternatieven onderling vergelijken op de volgende criteria:

- a) Toepasbaarheid virtuele infrastructuur;
- b) Licentiekosten;
- c) Beheersbaarheid.

Ad a) Een virtuele infrastructuur is zeer flexibel. Technisch is het bijvoorbeeld geen enkel probleem om virtuele machines net zo te dimensioneren als fysieke machines, maar daarmee gaat men wel voorbij aan de voordelen van een virtuele infrastructuur. Het resource scheduling proces van VMware kan beter overweg met lichter geconfigureerde vm's dan met zwaarder geconfigureerde vm's.

³³ Zie ook: <http://www.microsoft.com/sqlserver/2008/en/us/editions-compare.aspx>

Dit criterium is moeilijk objectief meetbaar te maken, aangezien ik geen mogelijkheden heb om de ontwerpen in de praktijk te testen. Daarom kies ik ervoor om een rangorde aan te geven door middel van een puntenverdeling. Op basis van ervaring kan ik wel aangeven dat alternatief 3 beter past dan de beide andere alternatieven. Alternatief 3 krijgt daarom 1 punt en de beide andere alternatieven krijgen 0 punten.

Ad b) In dit stadium komt de keuze voor een SQL enterprise of een SQL standaard editie aan bod. Een enterprise editie is ongeveer 6 keer duurder dan een standaard editie. Gebruik maken van de standaard editie in dit ontwerp is dus goedkoper dan de enterprise editie. In dit project is deze keuze alleen afhankelijk van het wel of niet toepassen van verschillende instances op een virtuele machine. De waardering is 0 punten voor het alternatief dat een enterprise editie vereist en 1 punt voor de beide andere alternatieven.

Ad c) Dit criterium is moeilijk absoluut en objectief meetbaar te maken. Om hier toch een waardering aan te geven beoordeel ik de alternatieven op de vraag hoeveel mogelijkheden er voor de beheerders zijn om de databases optimaal te laten presteren. Per instance is er één sqlsrvr proces (het laagste niveau in SQL server waarop resource management mogelijk is). Meerdere vm's geeft meer mogelijkheden voor resourcemanagement dan meerdere instances. Een vm functioneert immers onafhankelijk van andere vm's. Ik kom tot de volgende puntenverdeling:

- Alternatief 1: 0 punten;
- Alternatief 2: 1 punt;
- Alternatief 3: 2 punten.

	Toepasbaarheid Virtuele infra	Licentiekosten	Beheersbaarheid	Totaal
Alternatief 1	0	1	0	1
Alternatief 2	0	0	1	1
Alternatief 3	1	1	2	4

Tabel 5: vergelijking alternatieven applicatieverdeling

Op basis van bovenstaande vergelijking trek ik de conclusie dat Alternatief 3 het beste past in de casus van dit project.

Ten slotte beschrijf ik nog de verdeling van de databases over de verschillende vm's³⁴. Om database mirroring met automatische fail-over te laten functioneren is het noodzakelijk dat er behalve een principal en mirror server ook een witness server aanwezig is. Als de witness en stand-by server beiden ontdekken dat de principal server afwezig is wordt een fail-over geïnitieerd.

Het ontwerp moet een hoge beschikbaarheid van de SQL services en databases garanderen. Hier hoort in de casus van dit project ook bij dat de SQL services en databases beschikbaar moeten blijven in het geval van uitval van een compleet datacentre. Om aan deze eis te voldoen is het advies om de witness server in hetzelfde datacentre te plaatsen als de mirror server. Ik heb in het ontwerprapport een voorstel gedaan om twee databases primair te draaien vanuit het datacentre aan de WW en een applicatie primair in het datacentre aan de PW. De exacte plaatsing van databases kan plaatsvinden bij implementatie en achteraf nog aangepast worden.

³⁴ Zie paragraaf 2.2 "Verdeling databases" van het ontwerprapport (bijlage 4)

4.5. Configuratie virtuele machines

Deze paragraaf³⁵ gaat in op de vraag hoe de virtuele machines geconfigureerd moeten worden. Deze vraag is van belang om enerzijds een goede performance van de database server mogelijk te maken. Anderzijds moet voorkomen worden dat er een onnodig grote vraag naar de fysieke resources wordt gedaan.

Bij het opstellen van de configuratie loop ik echter tegen twee problemen aan:

- Het is op dit moment niet bekend welke eisen de databases aan de hardware stellen om een acceptabele performance te leveren;
- Het begrip ‘acceptabele performance’ is niet omschreven.

Ik ga in mijn antwoord uit van de best practices voor een gevirtualiseerde SQL server³⁶ om de hierboven aangegeven problemen te ondervangen.

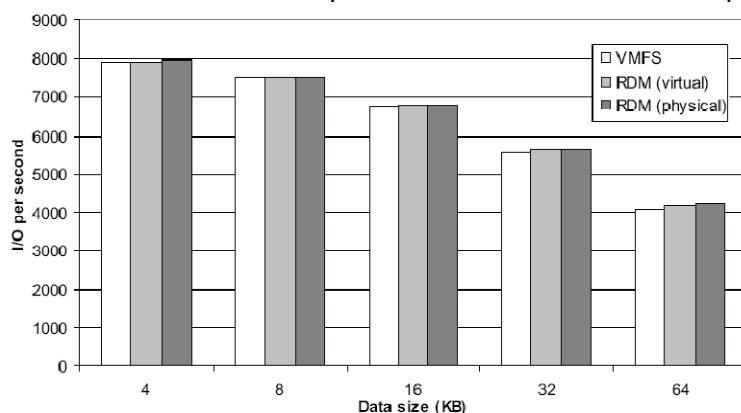
Bij de configuratie van een virtuele machine zijn er minder variabelen dan bij de configuratie van een fysieke machine. De configuratie opties voor een virtuele machine beperken zich tot:

- Het aantal virtuele CPU's;
- Gereserveerde CPU capaciteit;
- De hoeveelheid geheugen;
- De hoeveelheid gereserveerd geheugen;
- Type en hoeveelheid storage;
- Het aantal netwerkkaarten.

Bij fysieke machines kan het raadzaam zijn om meer netwerkkaarten te gebruiken vanwege redundantie en teaming. In een virtuele machine speelt dit echter niet, extra (virtuele) netwerkkaarten zijn alleen noodzakelijk als de vm met meerdere VLANs wordt verbonden.

Het virtualisatieplatform van Pimarox (VMware Vsphere) kan overweg met verschillende typen storage. Voor Pimarox is alleen de keuze tussen VMFS of RDM van belang. Een VMFS datastore is storage (doorgaans een lun op het SAN) waar het virtualisatieplatform (ESX) een eigen bestandssysteem op heeft gezet (zoals NTFS bij Windows). Op deze datastore zijn in elk geval de configuratie van de vm weggeschreven, evenals een swap file voor het fysiek geheugen en een logfile. De storage die aan een vm wordt aangeboden kan een virtuele hard disk zijn op de datastore (een VMDK bestand), maar ook een lun op het SAN dat rechtstreeks aan de vm wordt gekoppeld, een RDM.

Het is noodzakelijk om RDM storage te gebruiken bij shared storage clusters. Daarbij heeft een RDM een iets betere performance dan een VMDK op een VMFS:



Figuur 12: performance vergelijking RDM en VMFS. (Bron: VMware)

³⁵ Zie paragraaf 2.3 “Configuratie virtuele machines” in het ontwerprapport (bijlage 4)

³⁶ Zie “Microsoft® SQL server on VMware® best practices guide”

Het nadeel van het gebruik van RDM's zit in het maximale aantal lun's dat aan een ESX host toegekend kan worden. Elke RDM neemt op elke fysieke host een lun in beslag.

Een groot voordeel van RDM is: Bestaande luns op het SAN kunnen als RDM aan een vm gekoppeld worden zonder dat de data gekopieerd of gemigreerd moet worden.

Het is gebruikelijk om bij database servers OS, databases en logs te verdelen over meerdere fysieke schijven om een betere performance te halen. Bij vm's is dit echter niet automatisch het geval. Als de verschillende verschillende virtuele disken op dezelfde datastore geplaatst zijn, dan zijn de virtuele disken afhankelijk van dezelfde fysieke schijven. Er is dan geen sprake van een betere performance. Het gebruik van RDM's ondervangt dit, omdat een RDM een verwijzing naar andere storage is.

Op basis van bovenstaande afwegingen neem ik in mijn advies het gebruik van RDM's op.

4.6. Replicatie MC

Zoals in het definitierapport vermeld is, wordt de MC database ten behoeve van een dagelijkse rapportage gerepliceerd³⁷. Deze rapportage is bij het beheer bekend als de QBB lading.

Op dit moment wordt er gebruik gemaakt van een feature van de storage apparatuur om de lun's met daarop de database en logfile te synchroniseren met een ander lun. Deze feature vereist dat de target lun tijdens de synchronisatie niet verbonden is met een fysieke host.

Als gevolg van de keuze voor een volledig gevirtualiseerde infrastructuur is deze feature niet langer bruikbaar, de ESX hosts (de basis van het virtualisatieplatform) hebben immers altijd verbinding met de lun's, ongeacht de status van de virtuele machine.

Er zijn nu twee alternatieven mogelijk:

1. De rapportage database draaien op een fysieke server.
2. De rapportage database onderbrengen op een andere SQL server vm en gebruik maken van de replicatie technologie van SQL server.

Ad 1: Dit alternatief vereist alsnog een fysieke server, een Windows server enterprise licentie en een SQL licentie.

Ad 2: De synchronisatie loopt bij dit alternatief over het netwerk en is daarmee trager. Door de gerepliceerde MC database op een mirror server te plaatsen zijn er echter geen extra kosten voor hardware of licenties benodigd.

Gezien de voordelen van alternatief twee werk ik dit alternatief verder uit.

Mijn idee is om de gerepliceerde MC database (MCR) te plaatsen bij de vm met de SMT database. Om te bepalen of dit mogelijk was heb ik het gebruik van MCR onderzocht. Op basis van mijn ervaring had ik al de indruk dat de database alleen voor queries wordt gebruikt. Het doel van het onderzoek is dus om dit beeld bevestigd te krijgen bij de gebruikers.

³⁷ Zie laatste alinea paragraaf 3.3 "Databases" van het definitierapport (bijlage 3)

Veel gebruikte middelen voor een onderzoek onder personen zijn:

- Een enquête;
- Een interview;
- Een groepsgesprek.

In dit onderzoek zoek ik naar feiten over de QBB lading. Bij Pimarox zijn voor de verschillende business processen procesdeskundigen aangewezen. Het ligt dus voor de hand om feiten over de QBB lading te vragen bij de betreffende procesdeskundige.

De enquête en het groepsgesprek zijn vooral geschikt voor het uitvragen van feiten of meningen bij verschillende mensen. In dit geval heb ik om te beginnen maar één informatiebron.

Ik heb besluit dus om de procesdeskundige een kort telefonisch interview af te nemen bestaande uit 3 gesloten vragen:

1. Is de QBB lading nog steeds afhankelijk van de MCR omgeving?
2. Worden er bij deze lading wijzigingen aangebracht in de database?
3. Wordt de database buiten de lading om nog gebruikt?

Het antwoord op de eerste vraag is "ja". Het antwoord op de laatste twee vragen is "nee". Dit komt overeen met mijn ervaring dat de database alleen gebruikt wordt in een automatisch proces om een rapportage omgeving van informatie te voorzien. Voor nu is dit voldoende informatie.

Blijft over de vraag: "welke andere SQL server vm is geschikt om de MCR aan te bieden?" Van de databases die in dit project onderzocht zijn, kent de SMT database geen nachtelijke verwerkingen³⁸. Mijn advies is dus om de MCR database te plaatsen op de vm met de SMT database.

4.7. Ontwikkeling Exchange ontwerp

Exchange 2010 kent vijf verschillende server rollen:

- Mailbox server;
- Client Access server;
- Hub Transport Server;
- Unified Messaging server;
- Edge Transport server.

De Mailbox server, de Client access server en de Hub Transport server zijn essentieel in een Exchange 2010 ontwerp. De Unified Messaging server en de Edge Transport server zijn optioneel³⁹.

Om te bepalen of deze servers nodig zijn voor Pimarox bepaal ik of de geleverde functionaliteiten onderdeel zijn van de scope van de migratie zoals dat in het plan van aanpak en het definitierapport bepaald is.

³⁸ Zie paragraaf 3.7 "Beschikbaarheid SQL databases" van het definitierapport (bijlage 3)

³⁹ Zie ook de architectuur poster op:

<http://social.technet.microsoft.com/wiki/contents/articles/exchange-posters-visio-stencils-and-more.aspx>

De Unified Messaging server maakt het mogelijk om telefonie te integreren in de mailservices. Op dit moment wordt deze functionaliteit niet gebruikt⁴⁰. Ik ga de Unified Messaging server dus ook niet meenemen in het ontwerp.

De Edge Transport server kan in tegenstelling tot de overige servers in een afgescheiden netwerk geplaatst worden. De Edge Transport server is bedoeld om het mailverkeer van en naar het internet af te handelen en daarmee de intern geplaatste Hub Transport server te beschermen. Pimarox gebruikt voor de mailafhandeling met internet op dit moment de Ironport® Firewalls. De Edge transport server heeft niet meer functionaliteiten dan deze firewalls. Er is ook geen noodzaak om de huidige firewalls te vervangen. Gebruik van de Edge transport server brengt daarnaast wijzigingen aan het netwerk met zich mee. Ik neem de Edge transport server rol dus niet mee in mijn ontwerp.

4.8. Ontwerp mailbox server

Het belangrijkste van een Exchange infrastructuur is de mailbox server rol. Om de mailbox server optimaal te ontwerpen is een Mailbox role calculator op internet beschikbaar⁴¹. Deze calculator berekent op basis van een groot aantal variabelen de optimale sizing van de mailbox server rol.

Uit praktische overwegingen is het niet mogelijk om alle variabelen mee te nemen bij het bepalen van de optimale configuratie. De eerste stap is het vullen van een aantal omgevingswaarden. De tweede stap is het kiezen van de variabelen waarmee geëxperimenteerd gaat worden om de optimale configuratie te bepalen. Een overzicht van de mogelijke variabelen en hun betekenis is opgenomen in de calculator.

Om te bepalen met welke variabelen ik wel ga experimenteren kijk ik naar de eisen en wensen die gedefinieerd zijn. Hieruit volgt dat in elk geval alle mailboxen gemigreerd moeten worden zonder verlies van data of functionaliteit. In de definitiefase is vastgesteld dat 30% van de gebruikers een mailbox heeft van 1-2 GB. De rest zit daar onder. Ook is vast komen te staan dat er vanuit beheer vragen zijn over de lange retention periode.

Een andere eis is dat de omgeving hoog beschikbaar moet zijn. Hoge beschikbaarheid van de mailboxserver rol bij Exchange server 2010 is te bereiken door meerdere mailboxservers in een Database Availability Group (cluster van mailboxservers) op te nemen en de mailboxdatabases te dupliceren over meerdere mailboxservers⁴².

Op basis van bovenstaande overwegingen ga ik eerst experimenteren met de volgende variabelen:

- | |
|--|
| <ul style="list-style-type: none">a) Aantal mailboxservers (2, 4 of 6)b) Aantal database kopieën (2 of 3)c) Retentieperiode (14 of 62 dagen)d) Mailboxgrootte (1 GB of 2 GB)e) Aantal gebruikersprofielen (1 of 2) |
|--|

Ad a) Uitgangspunt is een redundant ontwerp vanwege de vereiste hoge beschikbaarheid. Dat betekent dat er minimaal 2 mailboxservers (vm's) moeten zijn. Meer kan interessant zijn om de load te verdelen.

⁴⁰ Zie ook hoofdstuk 2 “Mail omgeving” van het definitierapport (bijlage 3)

⁴¹ Zie: <http://msexchange.com/archive/2009/11/09/453117.aspx>

⁴² Zie: <http://technet.microsoft.com/en-us/magazine/ee835711.aspx>

Ad b) Het minimale aantal kopieën van een database om redundantie te kunnen bieden is 2 (inclusief de actieve kopie). Het voordeel van meer kopieën is meer redundantie en de mogelijkheid om goedkopere, niet redundante storage te kunnen gebruiken. Uit de calculator moet blijken of meer database kopieën ook voordelen bieden op het gebied van geheugen en processorcapaciteit.

Ad c, d en e) Retentieperiode, Mailboxgrootte en Gebruikersprofielen: De calculator die ik gebruik, berekent de maximale behoefte aan storage. Alle mailboxen hebben dan de maximale grootte. De calculatie voeden met een verschillend gebruikersprofiel geeft een meer realistisch beeld van de behoefte aan storage.

Aan de hand van deze variabelen heb ik een aantal mogelijke scenario's ontwikkeld. In onderstaande tabel worden verschillende scenario's onderscheiden waarin de hierboven genoemde variabelen verschillende waardes krijgen. Elk scenario is ingevoerd in de calculator en het resultaat is opgeslagen.

Ter verduidelijking van de tabel de volgende voorbeelden:

VB 1: Scenario 1 bestaat uit 2 mailboxservers en twee kopieën van elke database. Er is één gebruikersprofiel (laag 1, 1500 mailboxen) bestaande uit een maximale mailboxgrootte van 2 GB en een retentieperiode van 62 dagen.

VB 2: Scenario 11 bestaat uit 4 mailboxservers en twee kopieën van elke database. Er zijn twee gebruikersprofielen (laag 1 en 2). Laag 1, 450 mailboxen (30%) heeft een limiet van 2 GB. Laag 2, 1050 mailboxen heeft een limiet van 1 GB. De retentieperiode is voor beide lagen 62 dagen.

	Mailbox configuratie							
	Aantal		Laag 1			Laag 2		
	Servers	kopieën	aantal	grootte	retentie	aantal	grootte	retentie
1	2	2	1500	2	62	0	0	0
2	2	2	1500	2	14	0	0	0
3	2	2	1500	1	62	0	0	0
4	2	2	1500	1	14	0	0	0
5	2	2	450	2	62	1050	1	62
6	2	2	450	2	14	1050	2	14
7	4	2	1500	2	62	0	0	0
8	4	2	1500	2	14	0	0	0
9	4	2	1500	1	62	0	0	0
10	4	2	1500	1	14	0	0	0
11	4	2	450	2	62	1050	1	62
12	4	2	450	2	14	1050	1	14
13	4	3	450	2	14	1050	1	14
14	4	3	450	2	62	1050	1	62
15	6	3	450	2	14	1050	1	14
16	6	3	450	2	62	1050	1	62
17	6	2	1500	2	62	0	0	0
18	6	2	1500	2	14	0	0	0
19	6	2	1500	1	14	0	0	0
20	6	2	1500	1	62	0	0	0
21	6	2	450	2	62	1050	1	62
22	6	2	450	2	14	1050	1	14

Tabel 6: Mogelijke scenario's Exchange

Om een keuze te maken uit de verschillende scenario's onderzoek ik eerst de consequenties van de verschillende scenario's voor de benodigde hoeveelheid RAM geheugen, de benodigde CPU capaciteit en het benodigde aantal databases. De calculator berekent dit aan de hand van de ingevulde variabelen.

De berekende CPU capaciteit is gebaseerd op de volgende aannames⁴³:

- Het hosten van een actieve mailbox met 100 mailtjes van 75KB per dag kost 2 megacycles;
- Het hosten van een passieve mailbox kost 0,3 megacycles;
- Bij het ontwerp uitgaan van het maximale aantal actieve mailboxen per server;
- Elke CPU core van het virtualisatieplatform (bij Pimarox) levert 2400 megacycles⁴⁴;
- Bij het ontwerp uitgaan van een maximale processorbelasting van 70%;
- Minimaal aantal CPU cores is 2.

Voor het berekenen van het benodigde aantal databases ga ik uit van een maximale grootte van 150 GB. Dat is gebaseerd op de huidige praktijk van luns van maximaal 300 GB op het virtualisatie platform en de benodigde tijd om een database te kunnen repareren. Bij reparatie moet het database bestand gekopieerd worden. Hoe groter de database, hoe meer tijd dat kost. Volgens de ontwerpcriteria mag reparatie van een database hooguit 4 uur duren⁴⁵. Het maximum van 150 GB is gebaseerd op ervaring. De logfiles komen op hetzelfde lun als de database.

In onderstaande tabel zijn de uitkomsten per scenario weer gegeven.

Scenario	Geheugen	Aantal DB's	Schijfruimte totaal	Benodigde Megacycles	CPU cores
1	16 GB	32	12927 GB	3000	2
2	16 GB	28	11210 GB	3000	2
3	16 GB	20	7722 GB	3000	2
4	16 GB	16	6005 GB	3000	2
5	16 GB	24	9283 GB	3000	2
6	16 GB	20	7566 GB	3000	2
7	8 GB	16	12927 GB	1725	2
8	8 GB	14	11210 GB	1725	2
9	8 GB	10	7722 GB	1725	2
10	8 GB	8	6005 GB	1725	2
11	8 GB	12	9283 GB	1725	2
12	8 GB	10	7566 GB	1725	2
13	8 GB	15	11350 GB	1725	2
14	8 GB	18	13925 GB	1725	2
15	8 GB	12	11350 GB	1300	2
16	8 GB	12	13925 GB	1300	2
17	8 GB	12	12927 GB	1300	2
18	8 GB	10	11210 GB	1300	2
19	8 GB	6	6005 GB	1300	2
20	8 GB	8	7722 GB	1300	2
21	8 GB	8	9283 GB	1300	2
22	8 GB	8	7566 GB	1300	2

Tabel 7: doorrekening scenario's

⁴³ Zie: Exchange 2010 on VMware – Best practices guide

⁴⁴ Zie het definitierapport, hoofdstuk 4 en www.spec.org

⁴⁵ Zie paragraaf 1.2 "Eisen en wensen mailservices" van het ontwerprapport (bijlage 4)

Uit bovenstaande analyse blijkt dat de scenario's 1 t/m 6 aanzienlijk meer RAM geheugen en meer CPU cycles nodig hebben dan de overige configuraties. Gezien de keuze van Pimarox om de Exchange infrastructuur te virtualiseren laat ik deze scenario's buiten beschouwing bij de uiteindelijke keuze. “Lichtere” vm's zijn namelijk beter inpasbaar op de fysieke infrastructuur dan “zwaardere” vm's. Hiermee vallen de scenario's 1 t/m 6 af.

De scenario's 13 tot en met 16 zijn gebaseerd op 3 kopieën van elke database. Uit de analyse blijkt dat deze scenario's wat betreft geheugen en processorcapaciteit geen voordelen bieden, maar wel extra opslagruimte vereisen. Daar komt bij dat de extra functionaliteit van 3 kopieën per database door Pimarox niet gevraagd worden. Ik laat deze scenario's dus ook buiten beschouwing.

De keuze gaat dus tussen de scenario's 7 tot en met 12 en 17 tot en met 22.

Het criterium uit de kwaliteitsborging “een implementatieplan dat naadloos aansluit op de huidige infrastructuur” is niet voldoende onderscheidend om een keuze te kunnen maken uit de scenario's. Om toch een keuze te kunnen maken ga ik de alternatieven op andere criteria beoordelen:

- | |
|--|
| <ol style="list-style-type: none">1. Toepasbaarheid in een gevirtualiseerde infrastructuur;2. Overeenkomsten met de huidige praktijk. |
|--|

Ad 1. De impact op geheugengebruik en processorbelasting per vm is beduidend lager bij de scenario's die gebaseerd zijn op 6 virtuele machines. De benodigde processorbelasting staat al in de tabel, het geheugengebruik is door de mailboxcache rechtstreeks afhankelijk van het aantal actieve mailboxen per server. Het aantal actieve mailboxen is het totaal aantal mailboxen gedeeld door het aantal mailbox servers. Voor Pimarox betekent dit dat er hooguit 500 respectievelijk 300 actieve mailboxen per server zijn.

Ad 2. Hoewel kiezen voor een kortere retentieperiode en kleinere mailboxgrootte aanzienlijk kan schelen in de behoefte aan storage, kies ik toch voor de huidige retentieperiode van 62 dagen en een maximale mailboxgrootte van 2 GB voor tenminste 30% van de gebruikers. Op basis van dit criterium vallen de scenario's 8, 9, 10, 12, 18, 19, 20 en 22 alsnog af.

Van de overgebleven scenario's blijkt scenario 21 de minste hoeveelheid geheugen en processorcapaciteit te gebruiken. Ik werk scenario 21 dan ook verder uit in het ontwerp⁴⁶.

De overige scenario's worden wel beschikbaar gehouden voor ICT beheer, zodat de impact van de mailboxgrootte en retentieperiode later nog bekeken kan worden.

4.9. Ontwerp Client Access server⁴⁷

De Client access server (CAS) is een essentieel onderdeel van een Exchange 2010 ontwerp. De CAS neemt vrijwel alle communicatie met de clients voor z'n rekening. Ook de gewone Outlook clients die via MAPI/RPC de mailbox benaderen maken contact met de CAS (dit is een belangrijke wijziging ten opzichten van de voorgaande versies). De CAS onderhoudt vervolgens contact met de Mailbox server.

Het benodigde aantal CAS servers is afhankelijk van de processorcapaciteit en de belasting van de mailboxservers. De mailbox role calculator die ik gebruik heb voor het ontwerp van de mailbox role geeft ook een advies voor het aantal benodigde CAS processorcores. Volgens de calculator is 1 processorcore voldoende.

⁴⁶ Zie paragraaf 3.1 “Ontwerp mailboxserver” in het ontwerprapport (bijlage 4)

⁴⁷ Zie paragraaf 3.2 “Ontwerp CAS server” in het ontwerprapport (bijlage 4)

Zonder CAS is Exchange 2010 voor de gebruiker feitelijk niet bereikbaar. Een ontwerp dat hoge beschikbaarheid garandeert moet daarom ook aandacht besteden aan het hoog beschikbaar maken van de CAS rol. De CAS rol is hoog beschikbaar te maken door deze redundant uit te voeren en beide servers op te nemen in een Network Load Balancing cluster en CAS array.

Microsoft schrijft verder voor dat elke Active Directory site met mailbox servers minimaal één CAS bevat. De Active Directory van Pimarox bevat maar 1 site (zie het betreffende hoofdstuk in het definitierapport), dus aan deze eis is voldaan.

4.10. *Ontwerp Hub Transport Server*

Net als de CAS is de Hub Transport Server (HTS) een essentieel onderdeel van een Exchange 2010 ontwerp. De HTS verzorgt het transport van alle mails. Ook als de afzender en de geadresseerde zich in dezelfde database bevinden zal de mail via de HTS gaan.

Het benodigde aantal HTS servers is afhankelijk van de processorcapaciteit en de belasting van de mailboxservers. De mailbox role calculator die ik gebruikt heb voor het ontwerp van de mailbox role geeft ook een advies voor het aantal benodigde HTS processorcores. Volgens de calculator is 1 processorcore voldoende.

Zonder HTS is de gebruiker niet in staat om mails te ontvangen of te versturen. Voor de gebruiker functioneert in dit geval het systeem niet. Mijn ontwerp moet onder andere hoge beschikbaarheid garanderen. Ik neem daarom twee HTS servers op in mijn ontwerp. In beide datacentres komt één HTS.

Redundantie en fouttolerantie van de HTS rol zijn ingebouwd in Exchange 2010. De ontvangst van mail van buiten de Exchange organisatie valt daar echter buiten. Bij Pimarox verzorgen de Ironport firewalls de routing van mail van Internet naar de Exchange omgeving (zie de afbeelding in paragraaf 3.3). Deze firewalls kunnen voorzien worden van een extra afleveradres. Het is dus niet nodig om het afleveradres redundant te maken.

De HTS kan goed gecombineerd worden met de CAS rol op een enkele virtuele server. Het nadeel daarvan is echter dat de server dus zwaarder belast en mogelijk zwaarder geconfigureerd moet worden. Met het oog op de keuze van Pimarox voor een gevirtualiseerde omgeving adviseer ik de CAS en HTS rollen te scheiden over meerdere virtuele machines, zodat de impact van een enkele virtuele machine op de host beperkt blijft en het makkelijker is alle virtuele machines optimaal over de fysieke hosts te verdelen.

5. Realisatiefase

De laatste fase in mijn project is de realisatiefase. De doelstelling van deze fase is het ontwerp verder uit te werken tot een implementatieplan⁴⁸ zodat, als er later wordt besloten tot implementatie, het plan al klaar ligt. Daar komt bij dat het schrijven van het implementatieplan gebaseerd is op een proof of concept omgeving waardoor het de haalbaarheid van het ontwerp aantoont.

5.1. Schrijven implementatieplan

Het schrijven van vrijwel ieder document, dus zeker ook een implementatieplan, begint met het definiëren van het beoogde publiek. Bij een installatiehandleiding is dit nog net wat belangrijker. Een installatiehandleiding voor een compleet SQL- of exchange ontwerp is al snel een omvangrijk document (voor dit project 85 pagina's!). Als het niveau van het document niet aansluit op de lezer, dan is er een grote kans dat de aandacht van de lezer verslapt en de lezer mogelijk belangrijke informatie mist en daardoor fouten maakt (De handleiding heeft het dan gedaan!). De risico's de andere kant op, dat wil zeggen: de handleiding is te moeilijk, zijn kleiner. Zo is mijn ervaring.

Ik heb mij voorbereid op het schrijven van deze handleiding door eerst te bedenken wie dit document moeten kunnen volgen:

- a) In elk geval de "techneuten" die de installatie moeten uitvoeren;
- b) Beslissers die beslissen over uitvoering.

Ad a) De techneut verwacht informatie over het doel van de activiteit en de informatie die voor de uitvoering benodigd is.

Ad b) De beslisser wil weten wat de consequenties van uitvoering zijn en wil graag zeker weten dat het werkt.

Ik heb aan beide partijen aandacht besteed in het implementatieplan. Het gebruik van een Proof of Concept en de screendumps maken de beslisser duidelijk dat het ontwerp realiseerbaar is. In de tekst besteed ik aandacht aan de consequenties van de verschillende stappen. Zie bijvoorbeeld de aangehaalde tekst vanuit hoofdstuk 1 van het implementatieplan:

Aan het eind van dit hoofdstuk is SQL server geïnstalleerd en zijn de database mirroring en log shipping geconfigureerd. Tevens wordt er aandacht besteedt aan het installeren van databases en de beheerprocedures rondom mirroring en log shipping.

...

Voor de installatie van een nieuwe SQL server zijn de paragrafen 1.1, 1.2 en 1.3 van belang. De configuratie van database mirroring wordt besproken in paragraaf 1.4. De log shipping komt aan de orde in paragraaf 1.5.

⁴⁸ Zie bijlage 5: "implementatieplan virtualiseren mail- en database services"

De techneut kan ook op overzichtelijke wijze terugvinden wat er van hem verlangd word. Zie daarvoor bijvoorbeeld het volgende citaat uit paragraaf 1.2 van het implementatieplan:

Om SQL server te kunnen installeren is het volgende benodigd. Tussen de () de gegevens uit de POC:

- De SQL server installatie DVD;
- Een service account, Domain user, password never expires, de setup regelt de permissies (PIMAROX\SQLService);
- Locatie van database- en logbestanden (C:\databases en C:\Logs);
- De licentiecode;
- Een gebruiker voor Database beheer (Pimarox\admin).

De installatiehandleiding heb ik behoorlijk technisch gehouden. Aan de ene kant houd ik de technische lezer geboeid, aan de andere kant verklein ik het risico van "ondeskundig gebruik". In de inleiding van het implementatieplan staat aangegeven welk kennisniveau ik van de lezer verwacht.

Een andere aanpassing om aan te sluiten bij het veronderstelde kennisniveau van de lezer is het summier beschrijven van het besturingssysteem. Zowel het startpunt als het configureren van de storage zijn in het implementatieplan summier beschreven, maar begrijpelijk voor een ervaren Windows beheerder. Zie bijvoorbeeld onderstaand citaat uit hoofdstuk 1 van het implementatieplan:

In dit hoofdstuk wordt de installatie en configuratie van SQL server besproken. Het startpunt voor installatie is een schone installatie van Windows server 2008 R2. Hieraan zijn de kritieke patches toegevoegd, de firewall is uitgeschakeld voor het Domein verkeer en de storage voor de databases is geconfigureerd.

Uit ervaring weet ik dat het gebruikelijk is om de uitrol van een nieuw ontwerp gefaseerd aan te pakken. Ik ben daar bij het schrijven ook van uit gegaan door de implementatie in verschillende activiteiten onder te verdelen die los van elkaar uitgevoerd kunnen worden. De complete uitvoering is onderverdeeld in verschillende activiteiten. De hoofdstukindeling is erop gericht dat duidelijk is welke taken los van elkaar uitgevoerd kunnen worden. Bijkomend voordeel van deze opzet is dat delen van de handleiding (bijvoorbeeld de installatie van extra servers of het configureren van database mirroring) ook goed bruikbaar zijn als beheerdocumentatie.

5.2. Inrichting Proof of Concept

Bij het schrijven van de handleiding maak ik gebruik van een Proof of Concept omgeving (POC). De POC vertoont behoorlijke afwijkingen van de productie omgeving, maar is wel bruikbaar als bewijs dat het ontwerp gerealiseerd kan worden.

De belangrijkste verschillen zijn:

- Een enkele fysieke host in de POC ten opzichte van meerdere, geclusterde fysieke hosts in productie. Het enige voordeel van het gebruik van clusters is dat er sprake kan zijn van load balancing en redundantie.
- Twee datacentres in productie ten opzichte van één fysieke host in productie. In beide datacentres is hetzelfde subnet aanwezig, ook de centrale storage apparatuur verschilt niet tussen beide datacentres. Op een enkele fysieke host zijn daardoor geen aanpassingen nodig om het ontwerp te kunnen realiseren. Alleen redundantie bij uitval van een fysieke host is in de POC niet te testen.
- Andere en minder hardware in de POC dan in productie. Het gevolg is dat er in de POC geen betrouwbare snelheidstesten uit te voeren zijn. Ook de configuratie verloopt deels anders, omdat de storage configuratie in de POC geheel afwijkt van de productie.

De gebruikte POC is wel bruikbaar als referentie om het ontwerp vast te stellen, want in verkleinde vorm zijn de ontwerpen van SQL en Exchange 2010 nagebouwd.

5.3. Implementatie SQL ontwerp

Zoals uitgelegd bevat de POC minder hardware dan de productie. Ik heb daarom een kleiner SQL ontwerp gebouwd bestaande uit drie vm's. Met behulp van deze vm's heb ik de technieken zoals die beschreven zijn in het ontwerp (database mirroring en transaction log shipping) kunnen configureren en testen. Het opvallendste verschil is dat de POC maar één database heeft. De implementatie van het SQL ontwerp is volledig uitgewerkt in hoofdstuk 1 van het implementatieplan.

Ik heb in mijn ontwerp gesteld dat de database- en logbestanden gescheiden moeten worden van elkaar en het besturingssysteem. In de POC heb ik dat niet gedaan, bij gebrek aan fysieke hardware. De gevolgen voor de installatieprocedure zijn in het implementatieplan aangegeven⁴⁹.

Voor het configureren van database mirroring en transaction log shipping maak ik gebruik van de standaard SQL2000 oefendatabase "northwind". Het voordeel hiervan is dat ik direct bewijs dat een SQL2000 database in principe op een SQL 2008 server kan draaien.

Om de hoge beschikbaarheid van het ontwerp aan te tonen heb ik voor het schrijven van de beheersparagraaf⁵⁰ een fail-over geïnitieerd door de uitval van een vm te simuleren.

⁴⁹ Zie paragraaf 1.2 "Installatie SQL server" van het implementatieplan (bijlage 5)

⁵⁰ Zie paragraaf 1.6 "aandachtspunten met betrekking tot beheer" van het implementatieplan (bijlage 5)

5.4. Implementatie Exchange ontwerp

Als gevolg van de afwijkende POC omgeving is ook de Exchange 2010 omgeving die gebruikt is voor de installatie kleiner dan ontworpen. Zoals uitgewerkt in het betreffende hoofdstuk⁵¹ bestaat de minimale installatie uit zes servers. Met behulp van deze zes servers heb ik alle technieken uit het ontwerp kunnen installeren en configureren. Ook de migratie vanaf exchange 2003 is uitgevoerd in de POC.

Aan het eind van het implementatieplan zijn er minder databases dan ontworpen (2 in plaats van 24) en er zijn minder mailboxservers dan ontworpen (2 in plaats van 6). Ik heb aangegeven welke activiteiten uitgevoerd moeten worden voor het toevoegen van extra databases of servers. Zie het onderstaande citaat uit het implementatieplan:

Dit hoofdstuk gaat in op de installatie, configuratie en migratie van de Microsoft Exchange Server. Na afronding van de gehele migratie zijn de volgende paragrafen van belang indien er nieuwe mailboxservers moeten worden toegevoegd: 2.2, 2.2.3, 5.5.1, 2.6, 2.9.2, 2.9.3, 2.9.4. Het aanmaken van extra databases is beschreven in 2.9.2 en 2.9.4.

Net als bij de SQL installatie werk ik de configuratie van RDM's of aparte SCSI controllers niet uit. Ik veronderstel dat de configuratie van RDM's of virtuele SCSI controllers bekend is voor het technische deel van mijn publiek. Wel maak ik duidelijk hoe de schijfruimte voor de databases geconfigureerd moet worden⁵².

Exchange server leent zich uitstekend voor installatie en configuratie door middel van de command line en powershell (uitbreidbare command line interface van Microsoft). Ik pas deze methodes toe waar dit nuttig is. Gebruik van deze methodes in een implementatiehandleiding vermindert het aantal afbeeldingen in het document en daarmee de lengte. Bijkomend gevolg is dat de aandacht van de techneut minder snel verslapt, tekst lezen kost immers meer moeite dan een plaatje zien.

Niet alle stappen van de totale installatie, configuratie en migratie lenen zich voor command prompt en powershell. Soms blijft de grafische interface noodzakelijk. Ik maak van de grafische interface gebruik in de volgende situaties:

- a) installatie van de benodigde hotfixes;
- b) installatie van de eerste server in een rol;
- c) aanpassing van de receive connectors;
- d) verhuizen van de mailboxen.

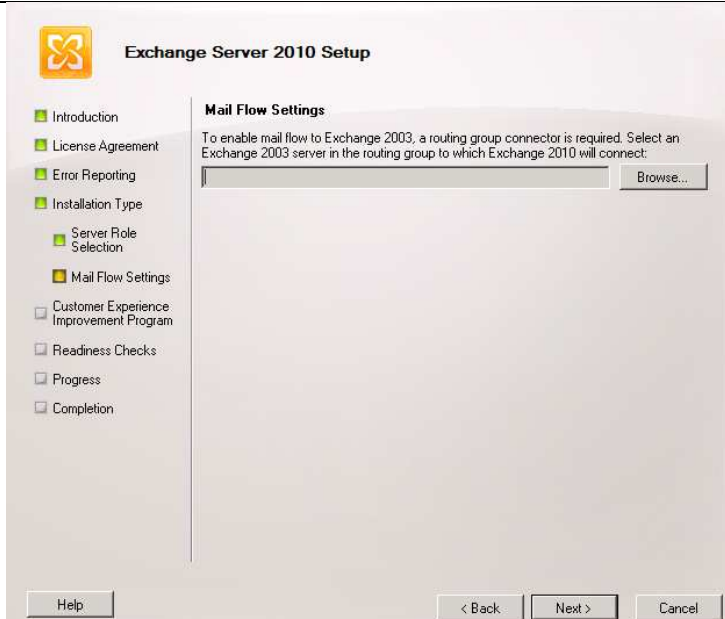
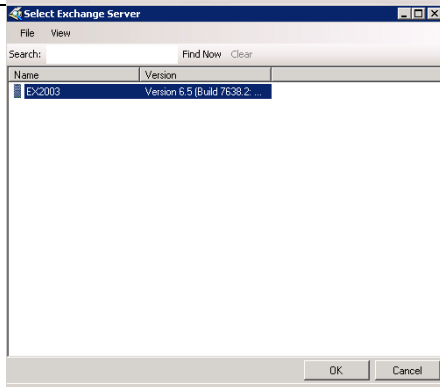
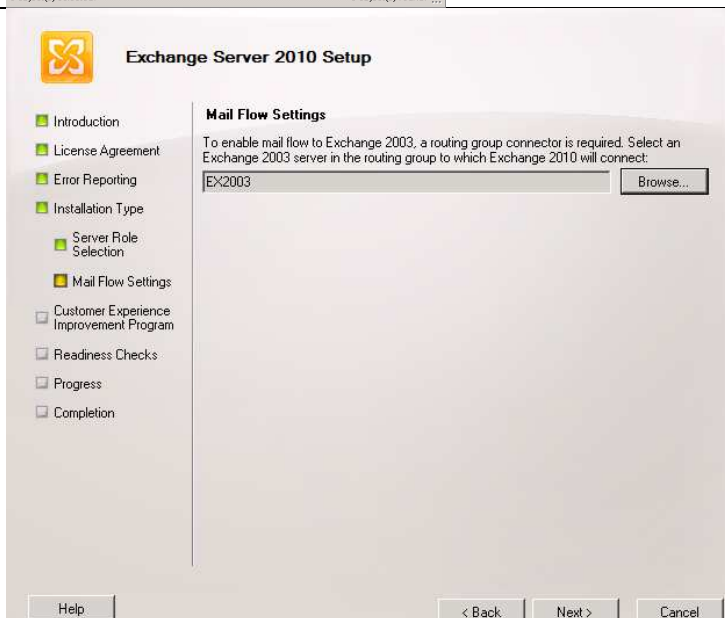
Ad a) De installatie van de hotfixes kan ook via een commando gebeuren, maar installatie via de grafische interface is meer gebruikelijk en makkelijker te controleren.

Ad b) Bij de installatie van de eerste server in een rol is het extra belangrijk dat aan de voorwaarden voor installatie aandacht wordt besteedt. Daarbij kan het zijn dat er nog extra vragen worden gesteld. Zie bijvoorbeeld de onderstaande stappen uit het implementatieplan⁵³:

⁵¹ Zie hoofdstuk 2 "Installatie exchange server" van het implementatieplan (bijlage 5)

⁵² Zie paragraaf 2.2.3 "Vorbereidingen Mailboxservers" van het implementatieplan (bijlage 5)

⁵³ Afkomstig uit: Paragraaf 2.4 "Installatie Hub transport server" van het implementatieplan (bijlage 5).

9.	Om de mailflow met de oude Exchange 2003 omgeving te kunnen realiseren moet er een Routing Group Connector aangemaakt worden. Klik op Browse om de Exchange 2003 server te selecteren die daarvoor gebruikt moet worden.	 The screenshot shows the 'Exchange Server 2010 Setup' window, specifically the 'Mail Flow Settings' step. On the left, a list of steps includes 'Introduction', 'License Agreement', 'Error Reporting', 'Installation Type', 'Server Role Selection', 'Mail Flow Settings' (which is highlighted), 'Customer Experience Improvement Program', 'Readiness Checks', 'Progress', and 'Completion'. The main area contains the text: 'To enable mail flow to Exchange 2003, a routing group connector is required. Select an Exchange 2003 server in the routing group to which Exchange 2010 will connect:'. Below this text is an empty text box and a 'Browse...' button. At the bottom, there are 'Help', '< Back', 'Next >', and 'Cancel' buttons.				
10.	Er is maar één exchange 203 server, de keuze is dus niet moeilijk. Klik op OK	 The screenshot shows a 'Select Exchange Server' dialog box. It has a 'File' menu and a 'View' button. Below is a search bar with 'Find Now' and 'Clear' buttons. A table lists available servers: <table><thead><tr><th>Name</th><th>Version</th></tr></thead><tbody><tr><td>EX2003</td><td>Version 6.5 (Build 7638 2...</td></tr></tbody></table> At the bottom, there are 'OK' and 'Cancel' buttons. The status bar at the very bottom indicates '1 object(s) selected.' and '1 object(s) found.'.	Name	Version	EX2003	Version 6.5 (Build 7638 2...
Name	Version					
EX2003	Version 6.5 (Build 7638 2...					
11.	Je bent weer terug in het Mail flow scherm. Klik Next	 This screenshot is identical to the one in row 9, but the text box in the 'Mail Flow Settings' section now contains the text 'EX2003'. The 'Browse...' button is still present to the right of the text box. All other elements, including the step list on the left and the bottom navigation buttons, remain the same.				

Het gebruik van de grafische setup zorgt ervoor dat deze vragen niet genegeerd kunnen worden.

Ad c) De aanpassing van de receive connectors bestond uit enkele clicks waar geen handig powershell commando voor is.

Ad d) Er zullen veel mailboxen verhuisd moeten worden. Verhuizen via de grafische tools heeft als voordeel dat er gemakkelijk meer mailboxen geselecteerd kunnen worden, via powershell is dat lastiger te bereiken.

Bij het verhuizen van mailboxen moet altijd rekening worden gehouden met de volgende 2 aandachtspunten:

1. In de Active Directory is vastgelegd welke Exchange server verantwoordelijk is voor de mailbox. Bij de verhuizing zal dat wijzigen in de CAS server die gekoppeld is aan de betreffende mailboxdatabase. Het gevolg is dat Outlook de verbinding kwijt zal raken zodra de exchange 2003 server de mailbox niet meer heeft.
2. De mailtjes worden verstuurd van de ene database naar de andere database. Tenzij circular logging aan staat, zal elk mailtje dat verstuurd of ontvangen wordt genoteerd worden in de Exchange log. Pas nadat er een volledige back-up van de exchange database gemaakt is zullen de logbestanden worden opgeschoond. Het verhuizen van veel mailboxen in korte tijd, kan leiden tot het vollopen van de ruimte voor logging en het vastlopen van Exchange server veroorzaken.

Op basis van deze twee bovenstaande factoren adviseer ik om de daadwerkelijke verhuizing gefaseerd buiten kantooruren uit te voeren. Beter iets langer bezig dan productieverstoringen riskeren.

6. Evaluatie

In dit hoofdstuk geef ik mijn evaluatie van het product (het implementatieplan) en het proces dat hiertoe gevolgd is. Ik sluit het hoofdstuk af met een demonstratie van de competenties.

6.1. Productevaluatie

De doelstelling van mijn project was om een goedgekeurd technisch ontwerp en implementatieplan op te leveren om de bestaande Exchange 2003 mailomgeving en de bestaande SQL 2000 database server te upgraden en te virtualiseren. Ik ben daarin geslaagd⁵⁴.

De extra eis waaraan deze omgevingen moeten voldoen is ondersteuning van de leverancier. Concreet houdt dit bijvoorbeeld in dat er gebruik moet worden gemaakt van de nieuwste versies van de software (Exchange 2010 en SQL server 2008). Ook betekent dit dat de VMware features DRS en HA niet gebruikt kunnen worden bij geclusterde servers (met uitzondering van NLB).

Ik heb de SQL omgeving hoog beschikbaar gemaakt door database mirroring toe te passen. Ik ben blij dat DRS en vmotion wel grotendeels ondersteund worden in mijn ontwerp, ik vind dat zeer nuttige features van VMware. Daarnaast ben ik blij dat het nieuwe ontwerp een snellere fail-over mogelijk maakt dan de huidige omgeving. Wanneer niet voor virtualisatie gekozen was, zou een voortzetting van een geclusterd ontwerp (Shared Storage clustering) overigens beter scoren bij de afweging van de verschillende ontwerpalternatieven.

Het criterium uit de kwaliteitsborging “Het implementatieplan moet naadloos aansluiten op de huidige infrastructuur” is continu meegenomen tijdens het ontwerpproces, vandaar dat dit criterium niet terugkomt in de afweging tussen de alternatieven. Zonder dit criterium waren er andere ontwerpen mogelijk geweest, in het bijzonder de compleet nieuwe architectuur van Exchange server 2010 zou een aanleiding kunnen zijn om aanpassingen door te voeren in de netwerkarchitectuur van Pimarox.

Mijn SQL ontwerp (en daarmee het implementatiescenario) voldoet aan de eisen die ik geïnventariseerd heb tijdens de definitiefase:

- a) Hoge beschikbaarheid;
- b) ODBC koppeling mogelijk;
- c) Gescheiden datafiles en logfiles;
- d) Replicatie van de MC database naar een rapportage omgeving;
- e) Ondersteuning vanuit Microsoft;
- f) Beperkte toegang tot de servers;
- g) Geen dataverlies bij migratie.

Ad a) Het ontwerp is redundant en daarmee hoog beschikbaar, want bij uitval van een database of vm neemt de mirror de taken automatisch over.

Ad b) De ODBC ondersteuning is niet gewijzigd in SQL 2008 ten opzichte van SQL 2000.

Ad c) De data- en logbestanden zijn gescheiden door het toevoegen van extra disks (RDM's) aan de SQL vm's.

⁵⁴ Zie ook bijlage 6: goedkeuring opdrachtgever.

Ad d) De MC database wordt via transaction log shipping gerepliceerd naar een andere database die in de nacht en overdag gewoon bereikbaar is.

Ad e) ondersteuning van Microsoft is al aan de orde geweest tijdens het ontwerpproces.

Ad f) Beperkte toegang tot de servers is niet expliciet aan de orde geweest tijdens de ontwerp- en realisatiefase. De beperkte toegang is echter inherent aan het gebruik van Active Directory en VMware vSphere. Alleen beheerders kunnen via deze omgevingen controle krijgen over de vm's.

Ad g). De SQL2000 database kan zonder aanpassingen geïmporteerd worden in SQL server 2008. Hierbij treedt geen dataverlies op.

Mijn ontwerp (en daarmee het implementatieplan) voldoet aan de eisen die gesteld zijn voor de Exchange 2010 migratie:

- a) Hoge beschikbaarheid;
- b) Moet werken met huidige en toekomstige clients;
- c) koppeling met firewall moet intact blijven;
- d) Geen dataverlies bij de migratie;
- e) Ondersteuning door Microsoft moet gewaarborgd blijven;
- f) Hersteltijd na een totale uitval is maximaal 4 uur.

Ad a). Het ontwerp is redundant en daarmee hoog beschikbaar omdat alle kritische onderdelen van een Exchange 2010 configuratie (CAS servers, Hub transport server, mailboxservers en mailboxdatabases) redundant uitgevoerd zijn en elkaars taak zonder tussenkomst van het beheer overnemen. De gebruikte technieken hiervoor zijn geconfigureerd volgens de procedures van Microsoft.

Ad b) Het client platform van Pimarox (Outlook 2003) wordt nog steeds ondersteund door Exchange server 2010⁵⁵.

Ad c) De Ironport firewall wordt niet gewijzigd in het ontwerp.

Ad d) Alle mailboxen worden gemigreerd, daarbij hoeft geen data verloren te gaan.

Ad e) De criteria voor ondersteuning zijn meegenomen in het ontwerp.

Ad f) Op basis van mijn ervaring bij Pimarox kan gesteld worden dat een restore van backup (traditioneel het worst-case scenario) van 150 GB aan data in minder dan een uur gerealiseerd kan zijn. Er zijn geen aanpassingen in de infrastructuur voorzien die een negatieve impact hebben op de bestaande calamiteiten procedures.

Ik vind het jammer dat er geen mogelijkheid was om in het kader van mijn project de nieuwe ontwerpen in productie te nemen of om een pilot te doen in de productie omgeving. De resultaten van mijn project zijn wel gebruikt in de migratie naar Exchange 2010 die onlangs is gestart bij de klant.

⁵⁵ Zie voor de ondersteunde client software: <http://technet.microsoft.com/en-us/library/aa996719.aspx>

Tijdens de definitiefase zijn de volgende aandachtspunten genoemd ten aanzien van inpassing in de virtuele infrastructuur:

- a) De maximale grootte van de datastores is 500 GB;
- b) Verdelen van vm's over meerdere datastores wordt nu niet toegepast;
- c) Er wordt bij voorkeur geen gebruik gemaakt van raw device mappings;
- d) Alle datastores worden door de SAN apparatuur gerepliceerd naar beide datacentre's, replicatie door de software is in principe niet noodzakelijk.

Ad a) De bestanden die op een vmfs datastore geplaatst moeten worden ten behoeve van een vm hoeven niet groter te worden dan 500 GB.

Ad b) Door het gebruik van RDM's is er geen noodzaak om vm's te verdelen over meerdere datastores.

Ad c) Het was niet mogelijk om deze wens in mijn ontwerp op te nemen. Zowel bij het ontwerp van SQL als bij het ontwerp van Exchange speelt dat er eigenlijk meer disken aan de vm aangeboden moet worden, ik heb geadviseerd gebruik te maken van RDM's. Ik heb hierbij de volgende afwegingen gemaakt:

- Meerdere virtuele disken op één datastore maakt de virtuele disken alsnog afhankelijk van dezelfde fysieke schijven.
- Een vm verdelen over meerdere datastores is in mijn ervaring een zeer grote bedreiging voor de stabiliteit en beheersbaarheid van een virtuele infrastructuur.

Ad d) Het was niet mogelijk om deze wens op te nemen in mijn ontwerp. Met behulp van de SAN replicatie is het niet mogelijk om te kunnen voldoen aan de eisen van hoge beschikbaarheid. Gebruik van de Microsoft technologieën biedt grote voordelen op het gebied van hoge beschikbaarheid in een gevirtualiseerde infrastructuur. Het belangrijkste verschil ten opzichte van de huidige situatie is dat bij een gevirtualiseerde infrastructuur niet alleen de data gerepliceerd wordt over het SAN, maar de hele vm. Hoge beschikbaarheid houdt dan in dat bij uitval van een vm in het eerste datacentre, de vm in het tweede datacentre online gebracht moet worden. Hier zitten zulke grote risico's aan dat ik deze opties niet meegenomen heb in mijn ontwerpproces. De bescherming is ook minder groot dan bij de nu ontworpen configuraties.

Het grote voordeel van LAN replicatie boven SAN replicatie dat de database logisch twee keer bestaat en alleen de transacties worden gerepliceerd. Bij SAN based replicatie worden de fysieke blokken gesynchroniseerd. Logisch is er dan sprake van één database.

6.2. Procesevaluatie

Ik ben blij dat ik vooraf gekozen heb voor een goed en toegankelijk gedocumenteerde projectmanagement methodiek. Deze methodiek heeft mij vooraf gedwongen goed na te denken over de fasering en de tijdsplanning. Dit heeft mij geholpen in het op tijd en volgens specificaties opleveren van de tussenproducten en het eindproduct. Een bekende valkuil voor mij is dat ik te vroeg over ga tot "bouwen". De gebruikte fasering dwong mij om eerst goedkeuring over het ontwerp te krijgen (en er dus nog eens kritisch naar te kijken) voor ik kon gaan "bouwen".

Bij dit project was ik als projectleider ook de enige projectmedewerker. De opdrachtgever had geen ervaring met de beschreven omgeving. Het was een uitdaging voor mij om de opdrachtgever van voldoende informatie te voorzien zodat hij z'n rol goed kon uitvoeren.

Bij grotere projecten zijn er doorgaans meer mensen betrokken bij een project. In de huidige opzet van het project had dat ertoe bij kunnen dragen dat de afstemming tussen de projectmedewerkers een probleem had kunnen worden. Bij een project met meer deelnemers zou ik meer aandacht besteden aan de onderdelen planning en projectbeheersing.

De Grit methodiek (en de meeste andere waarschijnlijk) gaan er wel van uit dat er aan het eind van het project opgeleverd moet gaan worden en mogelijk ook nog nazorg verleend moet worden. Nu de daadwerkelijke implementatie niet in de doelstellingen van het project opgenomen was, heeft dat geleid tot wat 'kunstgrepen' met betrekking tot de fasering. Ik ben blij dat ik uiteindelijk wel tot een passende fasering ben gekomen.

Terugkijkend op mijn project zijn er drie zaken die beter hadden gekund. Ten eerste heb ik snel en gemakkelijk gekozen voor deze projectmethodiek en fasering. Ik heb gemerkt dat ik daar een volgende keer meer aandacht aan moet besteden.

Ten tweede had ik het netwerk en de Active Directory niet meegenomen in de definitie van mijn onderzoeksdomein. Vanwege het belang van deze onderdelen voor zowel Exchange server als SQL server heb ik tijdens de definitiefase alsnog besloten om hier expliciet aandacht aan te besteden. Een volgende keer is het beter om het netwerk en de Active Directory ook op te nemen in het onderzoeksdomein.

Ten derde was er tijdens de laatste fase van dit project toch wel sprake van een behoorlijke tijdsdruk. De oorzaak van de tijdsdruk was dat ik in mijn plan van aanpak een te korte doorlooptijd heb gereserveerd voor de realisatiefase. De tijdsdruk heeft er toe geleid dat de inrichting van de gebruikte Proof of Conceptomgeving niet in één keer goed gegaan is. Bij een volgende keer zou ik wellicht voor een andere fasering kiezen waarbij ik eerst een product helemaal afrond (inclusief implementatie) voor ik met het volgende aan gang ga. De tijdsdruk heeft ook tot gevolg gehad dat ik concessies heb moeten doen aan de kwaliteit (wat betreft technische diepgang) van mijn implementatieplan. Als ik meer tijd had ingeruimd voor de laatste fase, dan had dat kunnen leiden tot een hogere kwaliteit.

Ik ben erg tevreden met de manier waarop ik mijn keuzes heb gemaakt en verantwoord. Ik heb ontdekt dat ik mij, ook buiten dit project om, bewuster ben geworden van de wijze waarop ik mijn keuzes maak door het scheiden van afwegingen en consequenties.

6.3. Competenties

Aan de hand van dit verslag en de bijbehorende bijlagen demonstreer ik dat ik de volgende

- A1 Analyseren van het probleemdomein;
- C9 Ontwerpen van een infrastructuur;
- E21 Migreren van een technische infrastructuur.

competenties op niveau 3 beheers:

Competentie A1

Het plan van aanpak en het definitierapport besteden aandacht aan het analyseren van het probleem van de opdrachtgever. Het beschreven probleem is vervolgens uitgewerkt in ontwerpcriteria.

Complicerende factoren bij deze competentie waren onder andere:

- Ik was de enige deskundige in het project;
- Er zijn veel gebruikers betrokken;
- De taken die ik moest uitvoeren waren deels nieuw.

Het was voor mij een uitdaging om de scope van het onderzoek in proporties te houden. Op basis van mijn ervaring en de drive om dingen te verbeteren bestond het risico dat ik 'te veel hooi op mijn vork' zou nemen. De uitdaging was dus om de scope groot genoeg te houden zodat een nieuw ontwerp de moeite waard is, maar ook klein genoeg zodat het realiseerbaar blijft. Ik ben daarin geslaagd. Met behulp van mijn ontwerp kan de afdeling ICT beheer de mailomgeving upgraden en een nieuw, stabiel SQL server database platform bieden aan de business.

Competentie C9

In het ontwerp rapport is een beschrijving gegeven van een nieuwe architectuur voor Exchange 2010 en SQL server 2008 bij Pimarox. Ik heb tijdens het ontwerpen continu rekening gehouden met:

- De eisen en wensen die ik geïnventariseerd heb tijdens de definitiefase;
- De best practices van VMware (de leverancier van het virtualisatieplatform);
- De best practices van Microsoft (leverancier van Windows server, SQL server en exchange server);
- Mijn eigen ervaringen met de producten;
- De voorwaarden die Microsoft stelt voor ondersteuning.

Het gevolg is dat de ontwerpen geschikt zijn voor uitrol op een virtuele infrastructuur en het recht op ondersteuning behouden blijft. De onderbouwing van de gemaakte keuzes is verwerkt in dit afstudeerverslag. Het implementatieplan geeft een zodanige detaillering dat mijn ontwerpen geïmplementeerd kunnen worden.

Complicerende factoren voor deze competentie waren onder andere:

- Het feit dat ik de enige deskundige in dit project was;
- De techniek database mirroring. Ik had hier nog nooit eerder mee gewerkt, maar het bleek wel de best passende techniek te zijn in dit project.

Competentie E21

Op basis van de gebruikte Proof of Concept omgeving heb ik aangetoond dat mijn implementatieplan leidt tot een succesvolle migratie van de SQL 2000 databases en Exchange 2003 mailomgeving. Met behulp van het implementatieplan is het mogelijk om de mail- en database services zodanig in te richten en te migreren dat er voldaan kan worden aan de eisen die de organisatie stelt.

Complicerende factoren voor deze competentie zijn onder andere:

- Implementatie van de ontwerpen heeft grote gevolgen voor ICT beheer. Ik heb hier in het implementatieplan aandacht aan besteed;
- Ik moest ervaring krijgen in het gebruik van de technieken database mirroring en Exchange Database Availability Groups.

Bijlagen

Bijlage I:	Afstudeerplan
Bijlage II:	Plan van aanpak
Bijlage III:	Definitierapport
Bijlage IV:	Ontwerpdokument
Bijlage V:	Implementatieplan
Bijlage VI:	Goedkeuring opdrachtgever

Literatuurlijst

Boeken:

- Grit, R. (2008) *Project management*. Houten: Noordhoff Uitgevers
- Otey, M. *SQL server 2008: high availability with clustering & database mirroring*. McGraw-Hill
- Hotek, M. *SQL server 2008: implementation & maintenance*. Microsoft Press
- Stanek, W *SQL Server 2008 Administrator's Pocket Consultant, second edition (updated for R2)*. Microsoft Press

Websites:

- Lifecycle software: <http://support.microsoft.com/gp/lifeselectindex#top>
- Vergelijking edities Windows server 2008: <http://www.microsoft.com/windowsserver2008/en/us/r2-compare-specs.aspx>
- Vergelijking edities SQL server 2008: <http://www.microsoft.com/sqlserver/2008/en/us/editions-compare.aspx>
- Architectuurposter Exchange server 2010: <http://social.technet.microsoft.com/wiki/contents/articles/exchange-posters-visio-stencils-and-more.aspx>
- Exchange storage calculator: <http://msexchangeteam.com/archive/2009/11/09/453117.aspx>
- Ondersteunde Client software Exchange 2010: <http://technet.microsoft.com/en-us/library/aa996719.aspx>

Digitale uitgaven:

- "SQL server on Vmware; availability and recovery options", PDF uitgave van VMware
- "Microsoft® SQL server on VMware® best practices guide", PDF uitgave van VMware
- "Exchange 2010 on VMware – Best practices guide", PDF uitgave van VMware