

# Afstudeerverslag

**Afstudeeropdracht:**

**Het ontwerp en de ontwikkeling van het Infodis CRM**

**INFODIS**

SUPPLY CHAIN SOLUTIONS

Author  
Date  
Version  
Status

Thijs van Hemessen  
24/9/2012  
1.6  
Final

**Version control and document owner**

Version	Date	Author	Description
1.0	13/4/2012	Thijs van Hemessen	Indeling concept, stukje over de organisatie toegevoegd.
1.1	17/4/2012	Thijs van Hemessen	Hoofdstuk 2 en 3.1
1.2	22/4/2012	Thijs van Hemessen	Hoofdstuk 3
1.3	10/5/2012	Thijs van Hemessen	Hoofdstuk 4, 5, 9 aangevuld
1.4	28/6/2012	Thijs van Hemessen	Hoofdstuk 6, 7, 8 aangevuld
1.5	29/6/2012	Thijs van Hemessen	Hoofdstuk 10, 11, 12 aangevuld
1.6	27/7/2012	Thijs van Hemessen	Algehele controle en correctie
1.7			

## VOORWOORD

Voor u ligt het afstudeerverslag van Thijs van Hemessen. Bedrijfskundige Informatica student van de Haagse Hogeschool.

Naar aanleiding van dit verslag zullen de begeleider en de expert van de Haagse Hogeschool mijn afstuderen beoordelen. Dit verslag is bedoeld om hun inzicht te geven in het verloop van het project dat ik heb uitgevoerd deze afstudeerperiode. De titel van de afstudeeropdracht luidt: **Het ontwerp en de ontwikkeling van het Infodis CRM.**

Hierbij wil ik ook direct een aantal mensen bedanken voor hun hulp, steun, begeleiding en inzet gedurende de uitvoering van mijn afstudeeropdracht:

Kees Alblas, voor de vele momenten dat hij de tijd heeft genomen om mij uitleg te geven over Infodis en de systemen van Infodis. En daarnaast ook voor zijn hersenkracht bij het meedenken over bepaalde problemen en oplossingen.

Willem Kosmeijer, voor de uitstekende hulp bij het bouwen en vullen van de database. En natuurlijk voor het uitleggen van programmeer werkzaamheden, nadenken over de kwaliteit van de database en de gegevens daarin.

Natuurlijk bedank ik ook alle medewerkers van Infodis voor de vriendelijkheid en de hulp die ze mij geboden hebben.

En van de Haagse Hogeschool zou ik graag mijn begeleiders Paul de Vries en Theo van Gerwen bedanken.

Alphen aan den Rijn, 23-9-2012

Thijs van Hemessen

# CONTENT

<b>VOORWOORD .....</b>	<b>III</b>
<b>CONTENT .....</b>	<b>IV</b>
<b>1 INLEIDING.....</b>	<b>7</b>
1.1 LEESWIJZER .....	7
<b>2 DE ORGANISATIE.....</b>	<b>8</b>
2.1 ORGANIGRAM INFODIS.....	9
2.2 CULTUUR.....	9
<b>3 AANLEIDING AFSTUDEERPROJECT.....</b>	<b>10</b>
3.1 PROBLEEMSTELLING .....	10
3.2 DOELSTELLING .....	10
3.3 UITVOERING PROJECT .....	10
<b>4 FASE 0: KEUZE METHODE EN TECHNIEKEN.....</b>	<b>11</b>
4.1 OPENUP .....	11
4.1.1 Iteratief werken.....	11
4.2 UML.....	12
4.3 INTERVIEWS.....	12
4.4 STAKEHOLDERANALYSE .....	12
<b>5 FASE 1: INCEPTION .....</b>	<b>13</b>
5.1 INITIATE A PROJECT .....	13
5.1.1 Opstellen projectplan.....	13
5.1.2 Opstellen iteratie plan.....	13
5.2 OPSTELLEN VISION .....	14
5.2.1 Kennismaking met Infodis .....	14
5.2.2 Inventarisatie van de knelpunten.....	14
5.2.3 Gevonden knelpunten.....	14
5.2.4 Vision opstellen .....	15
5.2.5 Glossary opstellen .....	16
5.3 MODELLEREN BEDRIJFSPROCES .....	16
5.3.1 Bedrijfsproces aanpassen.....	17
5.4 IDENTIFY AND REFINE REQUIREMENTS .....	17
5.4.1 Wensen en eisen verzamelen.....	17
5.4.2 Requirements tool bouwen.....	18
5.4.3 Requirements elicitation .....	19
5.4.4 Requirements analyse .....	20
5.4.5 Requirements specificatie en validatie .....	20

5.4.6	Use cases beschrijven.....	21
5.4.7	Inventarisatie informatie objecten.....	22
5.4.8	Proof of concept ontwerpen .....	25
5.5	AGREE ON TECHNICAL APPROACH.....	26
5.5.1	Opstellen Architecture Notebook .....	26
5.5.2	Technische werking van het systeem vastleggen.....	26
5.6	ITERATION ASSESSMENT .....	28
5.6.1	Besluit geschikte oplossing.....	28
<b>6</b>	<b>FASE 2: ELABORATION 1.....</b>	<b>29</b>
6.1	PLAN ITERATION .....	29
6.2	IDENTIFY AND REFINE REQUIREMENTS .....	30
6.2.1	Stakeholder analyse .....	30
6.2.2	Wensen en eisen verzamelen.....	31
6.2.3	Requirements valideren .....	32
6.3	DEVELOP THE ARCHITECTURE .....	32
6.4	DEVELOP SOLUTION .....	35
6.4.1	Risico analyse use cases .....	35
6.4.2	Use cases detailleren.....	35
6.4.3	Interfaces ontwerpen.....	36
6.4.4	Database ontwerp.....	38
6.5	TEST SOLUTION .....	43
6.6	ITERATION ASSESSMENT .....	45
6.6.1	Besluit volgende iteratie.....	45
<b>7</b>	<b>FASE 2: ELABORATION 2.....</b>	<b>46</b>
7.1	PLAN ITERATION .....	46
7.2	IDENTIFY AND REFINE REQUIREMENTS .....	47
7.2.1	Wensen en eisen verzamelen.....	47
7.2.2	Requirements valideren .....	47
7.3	DEVELOP THE ARCHITECTURE .....	48
7.4	DEVELOP SOLUTION .....	50
7.4.1	Bouwen database.....	50
7.4.2	ETL schema's opstellen.....	50
7.4.3	Vullen Infodis CRM database .....	51
7.4.4	Overzicht geïmporteerde gegevens .....	55
7.4.5	Documenteren database.....	55
7.4.6	Front end concept bouwen .....	56
7.5	TEST SOLUTION .....	58

7.6	ITERATION ASSESSMENT .....	59
<b>8</b>	<b>FASE 3: AFRONDING .....</b>	<b>60</b>
8.1	INTRODUCEREN CONCEPT SYSTEEM.....	60
8.2	OPSTELLEN OVERDRACHT DOCUMENT .....	60
8.3	ITERATION ASSESSMENT .....	61
<b>9</b>	<b>AFWIJKINGEN MET HET AFSTUDEERPLAN.....</b>	<b>62</b>
9.1	PROOF OF CONCEPT .....	62
9.2	SYSTEM SEQUENCE DIAGRAM .....	62
9.3	ETL.....	62
9.4	SQL (STRUCTURED QUERY LANGUAGE).....	62
9.5	DOCUMENTATIE IPV HANDLEIDING .....	63
<b>10</b>	<b>COMPETENTIEVERANTWOORDING.....</b>	<b>64</b>
10.1	BUSINESS ANALYSE .....	64
10.2	INFORMATIE ANALYSE .....	65
10.3	SYSTEEM ARCHITECTUUR .....	67
10.4	INFORMATIE ARCHITECTUUR .....	69
<b>11</b>	<b>EVALUATIE.....</b>	<b>71</b>
11.1	PROCES .....	71
11.2	PRODUCT .....	72
<b>12</b>	<b>LITERATUURLIJST.....</b>	<b>73</b>

# 1 INLEIDING

Het uiteindelijke doel was om binnen de afstudeerperiode van 17 weken een basis te hebben gelegd aan een oplossing waarmee de knelpunten bij Infodis zouden worden opgelost. Ook zou er al een klein deel van de oplossing gebouwd moeten zijn.

## 1.1 LEESWIJZER

Dit verslag is als volgt opgebouwd:

Allereerst zal de aanleiding van deze afstudeeropdracht behandeld worden. Vervolgens volgt het verslag de verschillende fases die doorlopen zijn tijdens dit project. In fase 0 wordt mijn keuze voor bepaalde methoden en technieken behandeld. In fase 1: Inception beschrijf ik hoe het project van start is gegaan en hoe ik ben begonnen met het inventariseren van mogelijke oplossingen voor het probleem. Ook wordt hier beschreven voor welke oplossing is gekozen. In fase 2: Elaboration 1 wordt dieper ingegaan op het ontwikkelen van de oplossing. Fase 2: Elaboration 2 laat zien hoe het eindproduct daadwerkelijk is gebouwd. Fase 3: Afronding toont hoe ik het project heb overgedragen aan Infodis. Ik behandel per fase ook een evaluatie van het proces en het product waaraan is gewerkt in die fase.

Daarna zal er een hoofdstuk besteed worden waarin ik de afwijkingen op mijn originele afstudeerplan behandel. Deze wordt opgevolgd door een hoofdstuk met mijn competentieverantwoording. Vervolgens zal ik ook globaal het hele project en het eindproduct evalueren. En ik zal eindigen met een literatuurlijst.

De bijlagen zullen gebundeld naast dit verslag worden geleverd.

## 2 DE ORGANISATIE

Infodis is een ambitieus bedrijf en is de afgelopen jaren sterk gegroeid, het gevestigd in Woerden. Het bedrijf is ooit opgezet door een aantal logistieke bedrijven die een oplossing zochten om zendingen digitaal te kunnen verzenden en in een systeem op te slaan. Infodis is na verloop van tijd zelfstandig geworden maar deze bedrijven zijn nog steeds de aandeelhouders. De belangrijkste drijfveer van de organisatie is haar klanten ten dienst staan met een goed werkend systeem. Hierin zijn ze al vele jaren succesvol. De sfeer is informeel en iedereen steekt graag zijn handen uit de mouwen om de ander te helpen.

Overal op de wereld werken bedrijven met de systemen van Infodis voor de optimale aansturing van transportprocessen. Met de diensten Message Exchange, Shipment Tracking en Carrier Management verlagen bedrijven transport gerelateerde kosten. Voor een lage prijs per zending wordt antwoord gegeven op alle vervoersvragen. Infodis faciliteert met deze diensten:

- Digitale communicatie tussen klanten, vervoerders en derde partijen
- Internationaal "tracking" van zendingen
- De keuze voor meest geschikte verzendwijze en de prijs daarvan
- Inzicht in de performance van vervoerders

Omdat Infodis deze diensten via Internet aanbiedt hoeven klanten niet zelf te investeren in hard- of software. Maar profiteren zij wel van kostenbesparingen en efficiëntere transportprocessen.

Infodis werkt voor circa 30 klanten waaronder bedrijven als Bavaria, Canon, Philips en Sony en het bedrijf heeft een sterke positie in de logistieke markt. Ook werkt Infodis met circa 250 grote en kleine vervoerders waaronder: DHL, DPD, TNT en UPS. Ongeveer een derde hiervan heeft een directe koppeling met het informatiesysteem van Infodis.

Er wordt gewerkt met systemen die Infodis zelf heeft ontwikkeld: Transport Management System (TMS) en Message Exchange System (MXS) zijn de backoffice systemen van Infodis. TMS is het systeem waarin informatie over zendingen van klanten en vervoerders is opgeslagen.

Het MXS zorgt ervoor dat totaal verschillende informatiesystemen van klanten en vervoerders aan elkaar gekoppeld worden. Het vertaalt de berichten, die onderling vaak verschillen van opbouw, zodat deze te gebruiken zijn in andere systemen.

Shipment Tracking System (STS) is het frontoffice systeem dat de klanten gebruiken. De naam van dit systeem dekt inmiddels niet meer de lading omdat het veel meer kan dan alleen "Tracking". Het systeem wordt in vorm van een website aangeboden aan klanten. Hiermee kunnen klanten zendingen aanmelden, inzien, aanpassen, volgen, factureren en verschillende soorten rapporten genereren.

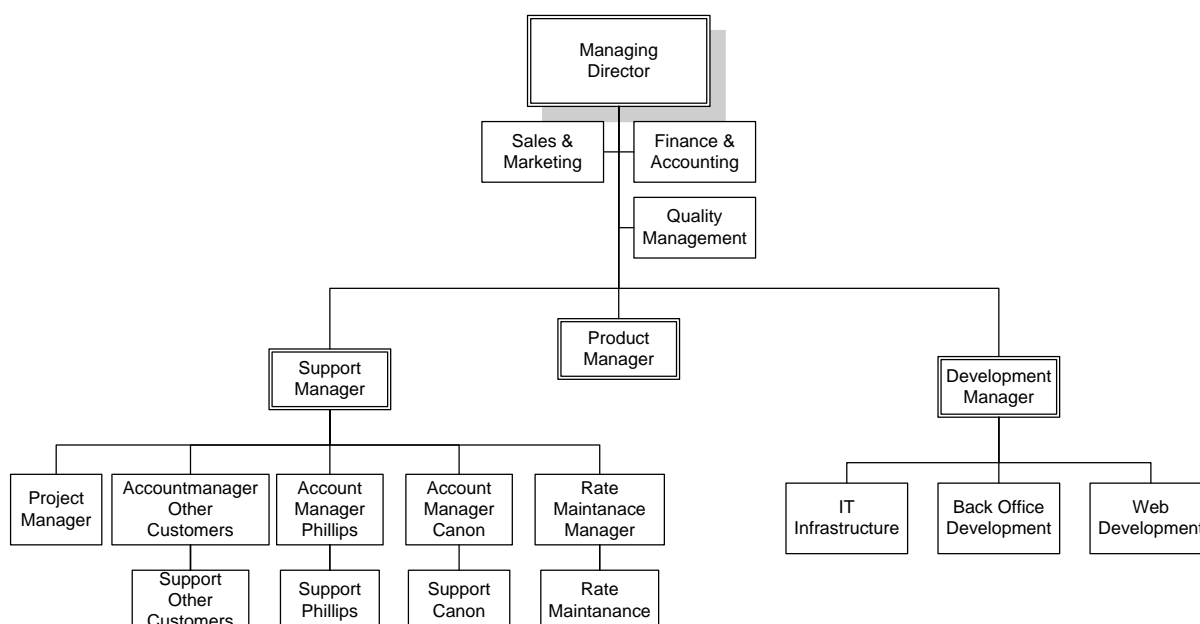


## 2.1 ORGANIGRAM INFODIS

Om een beeld van de organisatie te krijgen heb ik een organigram van Infodis gemaakt. Door dit model wist ik wat mijn positie in de organisatie was en waar ik moest zijn om bepaalde informatie te vinden. Ook heb ik met behulp van dit model een stakeholder analyse kunnen maken.

Hieronder is te zien hoe de rollen in de organisatie van Infodis zijn verdeeld. Allereerst is er een waarnemend directeur met daaronder een aantal stafafdelingen. Vervolgens is er een Support Manager die leiding geeft aan het hele support team. Deze bewaakt de performance en de voortgang van projecten zodat klanten op tijd geholpen worden. Er is een Product Manager die verantwoordelijk is voor de producten en diensten die Infodis biedt. Hierbij wordt gekeken of de producten en diensten aansluiten op de vraag vanuit de markt en of er wellicht nieuwe producten of diensten ontwikkeld kunnen worden. Ten slotte zorgt de Development Manager voor het ontwikkelen en up-to-date houden van de producten/diensten/systemen.

Niet in dit model opgenomen zijn de aandeelhouders van Infodis.



Omdat ik mij bezig hield met de ontwikkeling van een ICT oplossing ben ik in het Back Office Development team ingedeeld. Mijn werkplek was ook op de afdeling samen met alle andere developers. Mijn opdrachtgever en begeleider, dhr. Kees Alblas, is de Product Manager bij Infodis maar tevens ook een developer.

## 2.2 CULTUUR

Ondanks de niveaus in het organigram is de cultuur bij Infodis zeer informeel. Er is in de dagelijkse gang van zaken geen sprake van medewerkers onderaan of bovenaan de ladder. Als er besluiten genomen moeten worden is deze hiërarchie er uiteraard wel. Ik heb ervaren dat iedereen vrij aan te spreken is en ook zeer behulpzaam. Er heerst een open deuren cultuur en bij vragen kan je daarom gewoon even langs lopen.

## 3 AANLEIDING AFSTUDEERPROJECT

In dit hoofdstuk wordt de aanleiding van dit afstudeerproject behandeld.

### 3.1 PROBLEEMSTELLING

Binnen Infodis is informatie over klanten is nooit centraal opgeslagen. Het gaat hierbij om informatie over het bedrijf zelf (de klant), contactpersonen en technische gegevens wat betreft de inregeling van systemen. Deze informatie is verspreid over meerdere informatiesystemen, configuratiebestanden, e-mails, Excel bestanden en is niet altijd up-to-date. Infodis wil echter heel graag een totaaloverzicht van al deze informatie. Maar omdat de informatie op zoveel verschillende plekken staat is het lastig dit te verkrijgen. Dit was nooit een probleem omdat er een aantal medewerkers waren die van alles wel iets wisten. Dat ging goed totdat Infodis is gaan groeien. Naarmate het aantal klanten en medewerkers groeide kwam niet meer alle informatie altijd bij iedereen aan. Medewerkers gebruikten en bewaarden alleen de informatie die zij voor hun eigen klanten nodig hadden. Ook dat was nog steeds geen probleem op zich. Het werd echter wel een probleem als een klant een vraag of probleem had en de medewerkers die verantwoordelijk waren voor die bepaalde klant er niet waren. Op die momenten moest er heel veel zoekwerk verricht worden om de vraag van de klant te beantwoorden of om het probleem op te lossen.

### 3.2 DOELSTELLING

Het ontwerpen van een informatiesysteem (of eventueel aanpassen van een bestaand systeem) en het implementeren van de belangrijkste functionaliteiten daarvan zodat aan de knelpunten tegemoet gekomen kan worden.

### 3.3 UITVOERING PROJECT

Infodis heeft besloten om dit project als afstudeeropdracht aan te bieden aan mij, een bedrijfskundige informatica student van de Haagse Hogeschool. De reden hiervoor is dat er geen medewerker beschikbaar is die genoeg tijd heeft om dit project naast de gewone werkzaamheden uit te voeren. Ook wilden ze graag een frisse blik op het probleem. Daarnaast is hun doel natuurlijk ook om een leerplaats te bieden aan een afstudeerder en ook daarvan wat te leren.

Ik zal dit project uitvoeren gedurende een afstudeerperiode van 20 weken. Ik word binnen Infodis ondersteund en begeleid door Kees Alblas. Dit is tevens de opdrachtgever. Hieraan zal wekelijks worden gerapporteerd over de stand van zaken en deze zal samen met mij de voortgang bewaken.

Willem Kosmeijer uit het development team zal zijn aandacht richten op de technische kant van dit project. Hij zal mij helpen door het script- en programmeerwerk te verrichten. Daarnaast zal hij ook meedenken vanuit zijn rol als kwaliteitsmanager bij Infodis.

Vanuit de Haagse Hogeschool wordt ik begeleid door Paul de Vries en Theo van Gerwen. Deze zal ik ook op de hoogte houden van de voortgang.

## 4 FASE 0: KEUZE METHODE EN TECHNIKEN

Dit hoofdstuk zal mijn keuze voor verschillende methode en technieken beargumenteren.

### 4.1 OPENUP

In dit project zal gebruikt gemaakt worden van Open Unified Process, ook wel OpenUP genoemd. Dit is een manier voor iteratieve ontwikkeling van software en is gebaseerd op “best practices”.

Ik heb gekozen voor OpenUP omdat er tijdens dit project software ontwikkeld moet worden op een gestructureerde manier. OpenUP is uitstekend schaalbaar naar grote van een project. Daarnaast heb ik al eerder ervaring met OpenUP opgedaan. En vond ik het niet slim om een voor mij nieuwe methode te gaan gebruiken gezien de relatief korte duur van de afstudeerperiode en het vele werk. Mijn begeleider bij Infodis vond dit ook een prima idee en had geen bezwaar tegen een andere methode dan die ze zelf gebruiken. De medewerkers van de afdeling development zijn zelfs nieuwsgierig naar deze andere manier van werken.

OpenUP lijkt het in grote lijnen op de ontwikkel methode (SCRUM) die binnen Infodis zelf gebruikt wordt. Alleen wordt er bij SCRUM een stuk minder gedocumenteerd. SCRUM genereert ook geen uitgebreid gedocumenteerde motivatie voor de start van project, in tegenstelling tot bij OpenUP. Bij OpenUP moeten de richting en de visie van het project duidelijk vastgelegd worden.

Als ik toch SCRUM zou hebben gebruikt had ik dit moeten ondersteunen door onderdelen uit een andere projectmanagementmethode erbij te nemen. Bijvoorbeeld onderdelen om de reden en het bestaansrecht van dit project te documenteren. Ik vond dat een rommelige keuze aangezien al deze onderdelen al in OpenUP zitten en op elkaar afgestemd zijn.

Alle artefacten die opgeleverd worden kunnen worden getoetst aan checklists die OpenUP aanlevert. Hierin staan punten waarop gelet moet worden om de artefacten zo compleet mogelijk te krijgen en potentiële risico's te vermijden. Deze checklists zijn gegeneraliseerd en daardoor van toepassing op allerlei projecten en verschillende situaties. Het controleren van de artefacten met deze checklists helpt mij om de kwaliteit van de artefacten hoog te houden.

#### 4.1.1 Iteratief werken

Bij OpenUP realiseer je in iteraties, periodes van een aantal weken, een software oplossing. Gedurende elke iteratie worden er onderdelen van het systeem gebouwd. Dat past goed bij het project omdat de afstudeerperiode niet oneindig is. Er wordt dus elke iteratie iets opgeleverd waar Infodis iets aan heeft. Hiermee is er ook geen risico dat er aan het einde van de afstudeerperiode iets wordt opgeleverd waar Infodis helemaal niets mee kan. Iets wat kan gebeuren als een systeem niet in iteraties opgeleverd wordt. Dit komt omdat er met de iteraties rekening kan worden gehouden met veranderingen in wensen en eisen van stakeholders.

Veranderingen worden vaak niet ontdekt als men in één keer een heel systeem gaat bouwen. Maar ze kunnen er bijvoorbeeld voor zorgen dat je onderdelen weer helemaal opnieuw moet bouwen. Een groot voordeel is dus dat risico's niet aan het eind pas boven water komen. Dit omdat er elke iteratie al een testbaar onderdeel wordt opgeleverd.

Een ander voordeel is dat de kwaliteit van de software hoog is. Een belangrijke vuistregel bij OpenUP is dat “de software doet wat het moet doen”. Bij OpenUP wordt daarom gezorgd dat alle stakeholders met elkaar eens zijn wat de software moet doen. Als men het eens is wordt dat vastgelegd in de documentatie van OpenUP. Pas dan wordt er gebouwd wat er gebouwd moet worden. De software die wordt gebouwd is altijd op basis van diverse OpenUP documenten die het bestaansrecht van dat stukje software van meerdere kanten verdedigen.

Per OpenUP iteratie zal een Work item list worden gemaakt met items die gedurende die iteratie voltooid moeten worden. Met behulp van deze lijst kan de voortgang van de iteratie bewaakt worden.

Aan het eind van elke iteratie is er een assessment. Hierin worden de werkzaamheden gedurende de iteratie geëvalueerd en beoordeeld. Zaken die niet goed zijn verlopen moeten worden verbeterd en/of worden meegenomen naar de volgende iteratie.

## **4.2 UML**

Om processen en software onderdelen schematisch te weergeven zal ik gebruik maken van Unified Modeling Language, ook wel afgekort als UML. Hiermee is het mogelijk om processen, systemen, etc.. op een inzichtelijke manier weer te geven zodat deze ook door buitenstaanders en niet technische personen zijn te begrijpen. Hierdoor wordt het makkelijker om personen uit verschillende disciplines op dezelfde lijn te krijgen.

Binnen Infodis zullen ook niet technische personen betrokken zijn bij het ontwikkelen van de oplossing. Hun rol zal voor het grootste deel bestaan uit het formuleren van hun wensen en eisen. Om die wensen en eisen voor iedereen duidelijk te maken kunnen deze in een model weergegeven worden. Zo kan gecontroleerd worden of de wensen en eisen op de juiste manier zijn opgevat. En ook om deze opvatting met iedereen te delen.

Het proces van informatie verzamelen en het opstellen van modellen is voor mijzelf ook nuttig omdat ik op die manier snel en veel leer over het domein waarin ik het project uitvoer.

## **4.3 INTERVIEWS**

Door middel van interviews zal ik een groot deel van de benodigde informatie die voor dit project nodig is verzamelen. Ik heb ervoor gekozen om de initiële interviews kort te houden. Dit omdat er geen hoge drempel is bij het maken van contact. Binnen Infodis heerst een informele cultuur en is er een open deuren beleid.

Op basis van de uit interviews verzamelde informatie moet het systeem ontworpen worden. Tijdens de interviews zullen notulen worden gemaakt waar ook wensen en eisen uit gehaald kunnen worden.

## **4.4 STAKEHOLDERANALYSE**

Om een beeld te krijgen van alle stakeholders die invloed hebben op dit zal ik een stakeholderanalyse maken. Hiermee kan aangetoond worden dat alle stakeholders een stem hebben in dit project en dat de requirements de behoefte dekken vanuit alle oogpunten.

Na het maken van deze stakeholderanalyse zal duidelijk worden welke stakeholders direct- en welke indirect betrokken zijn. Ook zal er besloten worden welke stakeholders echt belangrijk zijn voor dit project. De interviews zullen worden gehouden met verschillende medewerkers uit de verschillende stakeholder groepen.

## 5 FASE 1: INCEPTION

In dit hoofdstuk worden de werkzaamheden van de Inception fase behandeld. De Inception fase is gericht op het opstarten van het project.

### 5.1 INITIATE A PROJECT

#### 5.1.1 Opstellen projectplan

Deze fase bestond uit een enkele iteratie. Het doel van deze iteratie was om een beeld te krijgen van de organisatie, de problemen die er speelde en het bepalen van een geschikte oplossing.

De eerste fase ben ik begonnen met het opstellen van een project plan, een artefact van OpenUP. Hierin zijn de achtergrond van het project, organisatie, methoden, milestones en de globale planning beschreven. Een aantal onderdelen hiervan kon ik rechtstreeks uit mijn reeds opgestelde afstudeerplan halen. Die onderdelen van het project plan zijn dan ook een meer gedetailleerde en uitgebreide vervolg hiervan.

Ik heb besloten om de planning in het projectplan globaal te houden en een gedetailleerdere planning per fase te maken. Deze planning is terug te vinden in het iteratie plan van elke fase. Dit heb ik gedaan omdat de planning zeer afhankelijk was van de te realiseren oplossing. En wat de oplossing zou worden moest bepaald worden tijdens de eerste fase van het project. Een volledige detailplanning was daarom niet realistisch. Als ik dit wel had gedaan had de planning hoogstwaarschijnlijk vele malen aangepast moeten worden. Door per fase een planning te maken met meer detail kon de planning meer op de oplossing gericht worden. In plaats van dat de oplossing op de planning werd gericht.

#### 5.1.2 Opstellen iteratie plan

Na het opstellen van het project plan heb ik voor de eerste iteratie een iteratie plan opgesteld. Daarin zijn de high level objectives, work items, milestones, evaluation criteria en een assessment opgenomen.

High level objectives zijn doelen die bereikt moeten worden tijdens de iteratie. Hieronder de doelen van deze iteratie:

- Een duidelijk beeld krijgen van de knelpunten
- Wensen en eisen verzamelen
- Requirements eliciteren
- Inventariseren beschikbare informatiebronnen met klantgegevens
- Proof of Concept

Om deze doelen te halen wordt er een lijst met work items opgesteld. Dit zijn de werkzaamheden die verricht moeten worden om de doelen te bereiken. Milestones zijn de producten die het resultaat zijn van de werkzaamheden. Dat waren in deze iteratie:

- Interview uitwerkingen
- Business activity diagram
- Information model concept
- Vision document
- Architecture Notebook

Om de iteratie te evalueren worden evaluation criteria opgenomen in het iteratie plan. Met behulp van de evaluation criteria kon bepaald worden of de iteratie een succes was of niet. De evaluation criteria worden opgesteld op basis van de high level objectives. Aan de hand van deze criteria wordt het assessment ingevuld. Als aan alle criteria is voldaan betekent het dat de high level objectives gehaald zijn en kan de iteratie afgesloten worden. Als er nog criteria zijn waar niet aan is voldaan moet er actie ondernomen worden om hier alsnog aan te voldoen. Tevens moet in het assessment uitleg gegeven worden waarom het wel en/of niet gelukt is om aan de criteria te voldoen. Het is assessment is daardoor ook een stuk reflectie voor de projectorganisatie.

Aan de hand van het iteratieplan kan de voortgang van de iteratie bewaakt worden. Hierin heb ik bijgehouden hoeveel uur ik bezig was met bepaalde werkzaamheden en hoeveel uur ik nog verwachtte ermee bezig te zijn. Ook heb ik project issues en risico's hierin vastgelegd. Het assessment van de iteratie wordt besproken aan het einde van deze fase.

Zie het iteratie plan van Fase 1 in de bijlagen.

## **5.2 OPSTELLEN VISION**

### **5.2.1 Kennismaking met Infodis**

De eerste dag kreeg ik een rondleiding en introductie waarbij ik alle medewerkers even heb gesproken. Daarbij zat ook een korte uitleg van mijn opdracht. Het leek me verstandig om later iedereen nog een keer te spreken over haar/zijn werkzaamheden om zo inzicht te krijgen in het bedrijf. Door dit inzicht kon ik later in de iteratie beter bepalen wat voor oplossing paste bij Infodis.

In de dagen die daarop volgden heb ik meerdere medewerkers gesproken. Daardoor heb ik een beeld van de werkzaamheden en de organisatie gekregen. Ook heb ik gezien op welke fronten de knelpunten zich manifesteerden.

### **5.2.2 Inventarisatie van de knelpunten**

Toen ik begon met mijn afstudeerstage bij Infodis wist ik al iets over het bedrijf zelf en over de knelpunten die er speelden. Maar wat er nu precies voor knelpunten waren wist ik niet. Ook wist ik niet hoe breed het probleem in de organisatie gedragen werd. En wat de impact ervan op de medewerkers was. Dit waren vragen waarop ik het antwoord wilde hebben. De antwoorden zouden mij helpen met het bepalen van een geschikte oplossing.

Om deze antwoorden te krijgen heb ik interviews gehouden. Hierbij heb ik besloten om van alle afdelingen een aantal medewerkers te selecteren. Daardoor had elke afdeling een stem in deze interviews. Het interview startte ik door nogmaals te vertellen wat mijn opdracht was bij Infodis en het overhandigen van een korte vragenlijst. Ik had van tevoren slechts een beperkt aantal vragen opgesteld omdat ik medewerkers de ruimte wilde geven om hun verhaal te doen. Tijdens het interview zou ik doorvragen op hetgene wat werd verteld.

### **5.2.3 Gevonden knelpunten**

De knelpunten die naar boven kwamen waren op bijna alle afdelingen gelijk. En knelpunten die afweken waren zeer specifiek op een bepaalde afdeling gericht. Bij de afdeling systeembeheer kwam bijvoorbeeld ter sprake dat zij erg lang bezig waren met technische gegevens opzoeken over communicatielijnen tussen Infodis en klanten. Dat is een knelpunt wat alleen op die afdeling speelt.

De gevonden knelpunten spelen ook niet in elk geval op. Ze spelen vooral op als bepaalde medewerkers er niet zijn. Bijvoorbeeld de medewerkers die heel veel van een onderwerp afweten. Of medewerkers die verantwoordelijk zijn voor een bepaalde klant. In sommige gevallen moet een ander de taken overnemen of snel een probleem oplossen. Maar omdat informatie niet centraal is opgeslagen is het lastig voor diegene om de benodigde informatie te vinden om zijn/haar werk te doen.

Hieronder de belangrijkste knelpunten.

- Opzoeken informatie duurt lang.
- De informatie is gefragmenteerd.
- Informatie moet via een workaround gevonden worden.
- Sommige informatie wordt helemaal niet vastgelegd.

Voor een complete lijst met knelpunten verwijs ik naar het Vision artefact in de bijlagen.

#### 5.2.4 Vision opstellen

In dit artefact is duidelijk de “Wat en waarom” van het project beschreven. De Vision definieert het beeld van de stakeholders over de oplossing die gerealiseerd dient te gaan worden. Deze definitie uit zich in een lijst van essentiële wensen en eisen en een beschrijving van de mogelijkheden en functies die het systeem biedt. Hiermee is de Vision ook basis voor meer gedetailleerde technische eisen aan het systeem.

Ik heb de Vision goed laten keuren door mijn opdrachtgever zodat ik een goede basis had om naar een oplossing te gaan zoeken. De Vision was voor mij een manier om potentiële oplossingen te beoordelen. Belangrijke beslissingen over oplossingen heb ik altijd gemaakt door naar de Vision te kijken. Aan de hand van de Vision kon ik bepalen of de oplossing de gewenste resultaten zou geven. Als een oplossing haaks stond op wat er in de Vision beschreven was gaf dit mij aan dat het nog niet de goede oplossing was.

De eerste versie van het Vision document heb ik opgesteld aan de hand van een template. Dit template is te vinden op de website van OpenUP. Ik gebruikte een template omdat deze is gebaseerd op de best practices van vele OpenUP aanhangers.

Ik heb de Vision geschreven op basis van gesprekken met mijn opdrachtgever, onderdelen van het afstudeerplan en interviews met de medewerkers. De inhoud is meerdere malen aangepast en aangevuld omdat er na vervolg interviews meer informatie over de knelpunten beschikbaar was. Latere versies zijn in de loop van het project aangevuld. De kern van het document is echter hetzelfde gebleven.

De Vision bevat globaal:

- Problem statement
- Stakeholder analyse
- Onion stakeholder model
- User environment
- Product overview

In de problem statement wordt het probleem wat aan de orde is besproken. De knelpunten zijn hierin uitgebreid beschreven. Verder heb ik beschreven waarop de knelpunten invloed hebben en wat de impact daarvan is.

Onder stakeholder analyse heb ik een opsomming van alle stakeholders opgenomen met daarbij een beschrijving en hun verantwoordelijkheden. Aan de hand hiervan is te zien wie verantwoordelijk zijn en wie invloed hebben op het project. In de Elaboration fase heb ik een uitgebreidere stakeholder analyse gemaakt met daarbij ook een uitwerking van het Onion model. Hierover meer in de hoofdstukken van Fase 2: Elaboration 1.

Verder is er een inventarisatie opgenomen van de gebruikersomgeving en applicaties bij Infodis. Hier is wat over de medewerkers zelf verteld en over hun werkzaamheden. Ook heb ik alle applicaties benoemd die binnen Infodis regelmatig worden gebruikt. Dit heb ik gedaan zodat er bij de oplossing rekening gehouden kan worden met eventuele koppelingen tussen die applicaties en de te realiseren oplossing.

Er was een grote kans was dat de te realiseren oplossing door Infodis ontwikkeld zou gaan worden. Daarom heb ik de binnen Infodis gebruikte programmeertalen vermeldt. Het zou niet efficiënt zijn geweest om een oplossing te kiezen die niet in één van deze talen zou worden ontwikkeld. Dat zou betekenen dat er iemand een nieuwe programmeertaal moest gaan leren.

Daarop volgend heb ik beschreven wat er nodig was om de knelpunten op te lossen (Needs). En beschreven welke functionaliteiten (Features) in de oplossing ervoor moesten zorgen dat dit ook daadwerkelijk gebeurde. Aan de hand hiervan kon ik beoordelen of bepaalde oplossingen voldeden en de knelpunten konden verhelpen.

Zie de Vision in de bijlagen.

#### **5.2.5 Glossary opstellen**

Omdat er in veel projecten terminologie gebruikt wordt die niet voor iedereen is te begrijpen heb ik een glossary opgesteld. In deze glossary zijn alle project specifieke en vaktechnische termen opgenomen met daarbij een betekenis of omschrijving. Ook termen die op meerdere manieren geïnterpreteerd kunnen worden zijn hierin beschreven en de verschillen zijn duidelijk gemaakt. Hierdoor ontstaan er een uniforme kennis betreffende het taalgebruik binnen het project. Dit verminderd het risico op miscommunicatie en het verkeerd interpreteren van bepaalde zaken.

Met het opstellen van een glossary ben ik gelijk de eerste dag begonnen om de terminologie mijzelf eigen te maken. Elke keer als ik een term hoorde die ik niet begreep heb ik deze opgeschreven en gevraagd wat het precies betekende. Soms moest ik voor een echt precieze definitie langs verschillende medewerkers. Dit omdat de medewerkers wel wisten wat het ongeveer betekende maar het niet makkelijk op mij over konden brengen. Dat was gelijk het bewijs dat een glossary echt nodig was voor dit project.

Alle termen die ik had opgeschreven heb ik vervolgens op alfabetische volgorde vastgelegd in een glossary artefact. Natuurlijk met een omschrijving. De omschrijving heb ik laten goedkeuren door een key-user.

Zie bijlage 11.7.

### **5.3 MODELLEREN BEDRIJFSPROCES**

Het bedrijfsproces “Klantgegevens bijhouden” is waar project op van toepassing is. Dit proces is verspreidt over meerdere afdelingen binnen Infodis. Op elke afdeling wordt er een deel van de gegevens toegevoegd. Dat deel is meestal alleen het deel wat toegevoegde waarde heeft voor de desbetreffende afdeling. Ik heb besloten om dit bedrijfsproces voor meerdere afdelingen te modelleren om inzicht te krijgen in de verschillende werkwijzen rondom het bijhouden van klantgegevens. Hierdoor kreeg ik ook beter inzicht in welke soort gegevens op welke plekken gebruikt worden.

De gegevens van een klant worden niet altijd via dezelfde stappen aangevuld. Er zijn verschillende manieren waarop een nieuwe klant bij Infodis bekend wordt. Dit ligt aan het soort klant. Klanten die een totaalpakket willen waarin zij al hun zendingen kunnen beheren en volgen komen vaak binnen via een accountmanager. Hieraan zit meestal een project van weken tot maanden vast waarin het systeem voor de klant ingeregeld wordt. Gegevens van de klanten worden in deze situatie door de accountmanagers vastgelegd.

Maar er zijn ook klanten die simpelweg berichten willen uitwisselen en daarvoor Infodis als tussenpartij nodig hebben. Het uitwisselen van berichten is een technisch verhaal en wordt daarom door developers van Infodis en voor de klant ingeregeld. Gegevens van de klant moeten in deze situatie door de developers worden vastgelegd.



Wat er dan precies wordt vastgelegd verschilt ook per afdeling. Accountmanagers maken een record aan in Topdesk, een map op de netwerkschijf en een gedeelde map voor e-mail op de Exchange server. Dit laatste kan echter ook door de developers gedaan worden. Developers maken, als deze er nog niet is, ook een map aan op de netwerkschijf. Daaraan voegen zij weer hun eigen mappen toe. Ook maken zij een party\_code aan (de code waarmee elke klant/partij uniek wordt geregistreerd) en registreren deze in een Excel sheet. Systeembeheerders beheren de gegevens over verbindingen met de klant. Deze worden weer opgeslagen in aparte mappen.

Dit alles is een nogal ingewikkeld proces waarin wel eens wat fout kan gaan. Afdelingen kunnen langs elkaar heen werken en gegevens worden niet ingevoerd of incompleet. Waar ik me op moest gaan richten was het bijhouden van de meest simpele klantgegevens. De naam van de klant, adresgegevens, contactpersonen, naam van de klant in Infodis systemen, etc.

Na het modelleren van de bedrijfsprocessen kon ik concluderen dat er geen procedures over het toevoegen van klantgegevens waren. Dit gebeurt daarom niet altijd even correct of volledig. Het probleem hier is dat iedereen zijn eigen stukje klantgegevens opslaat en zelf bedenkt op welke plek het wordt opgeslagen. Niet iedereen gebruikt dezelfde structuur en daarom is het voor anderen lastig om iets te vinden. In de oplossing kunnen maatregelen genomen worden om dit aan te pakken. Bijvoorbeeld door velden te maken die verplicht moeten worden ingevuld. Echter deze kunnen alleen zorgen dat klantgegevens volledig ingevoerd worden. Er kan niet mee afgedwongen worden dat klantgegevens überhaupt worden ingevoerd. Hiervoor moeten veranderingen aan het bedrijfsproces worden gemaakt die buiten de scope van het systeem vallen.

Zie ook het Business Activity Diagram in de bijlagen.

#### **5.3.1 Bedrijfsproces aanpassen**

Nadat ik heb bedrijfsproces “Klantgegevens bijhouden” had gemodelleerd ontstond bij mij de vraag of ik niet beter het proces kon aanpakken in plaats van een systeem ontwikkelen om de knelpunten op te lossen. Het bijhouden van klantgegevens begint tenslotte bij het proces zelf, als het hier niet goed loopt helpt ook een nieuw systeem niet. Na dit besproken te hebben is er besloten dat er inderdaad iets gedaan moet worden aan het bedrijfsproces. Echter zal dit niet mijn verantwoordelijkheid zijn. De kwaliteitsmanager bij Infodis heeft aangegeven dat het inderdaad aan de grondslag ligt van het probleem. Hij gaf aan dat Infodis al bezig was met een project om zaken omtrent klanten op te slaan in een nieuwe gestandaardiseerde SharePoint omgeving. Hierbij ging het voornamelijk om documenten en procedures. Met het introduceren van de SharePoint omgeving zouden ook noodzakelijke veranderingen in het bedrijfsproces gepaard gaan.

## **5.4 IDENTIFY AND REFINE REQUIREMENTS**

#### **5.4.1 Wensen en eisen verzamelen**

Toen ik voldoende was georiënteerd binnen Infodis kon ik beginnen met het verzamelen van wensen en eisen. Voldoende betekend hier dat ik wist wat de werkzaamheden op de afdelingen waren en welke knelpunten er speelden.

De wensen en eisen moesten helpen om een compleet beeld van een oplossing te vormen. Sommige wensen waren al naar boven gekomen in de oriëntatie en had ik al genoteerd. In vervolg interviews met de medewerkers kwamen er nog meer wensen en eisen naar voren. Omdat ik tijdens deze interviews ook notuleerde kon ik de wensen en eisen direct uit mijn notulen halen. Per interview hield ik bij wie de aanwezigen waren en zo kon ik ook bijhouden van wie de wensen en eisen waren.

Sommige wensen en eisen kwamen van een enkel persoon en sommige kwamen bij meerdere personen naar voren. Wensen en eisen van meerdere personen hadden niet gelijk meer waarde hadden dan die van een enkel persoon. Dat kon ik pas gaan concluderen tijdens later onderzoek.

Deze wensen en eisen zijn meegenomen in de Vision, het Architecture Notebook en de requirements.

#### 5.4.2 Requirements tool bouwen

Voor het vastleggen van de requirements heb ik mijn eigen Excel requirements management tool gemaakt. Dit heb ik gedaan omdat er binnen Infodis geen tool of pakket was om requirements te managen. Omdat ik de requirements op een gestructureerde manier wilde vastleggen was een requirements management tool noodzakelijk.

Tijdens mijn opleiding heb ik al eerder een requirements tool gemaakt. Deze heb ik niet opnieuw gebruikt omdat hierin veel onhandige functionaliteiten waren verwerkt. De functionaliteiten waren onhandig omdat maar één persoon snapte hoe ze werkte. Diegene die het geheel in elkaar geknutseld had. Ik heb daarom zelf een nieuwe tool gemaakt. De nieuwe tool is meer op dit project en de omvang daarvan gericht. Om de tool compleet te maken heb ik informatie uit het boek "Succes met de requirements!" gebruikt. Van elke requirement worden nu de volgende zaken ingevuld:

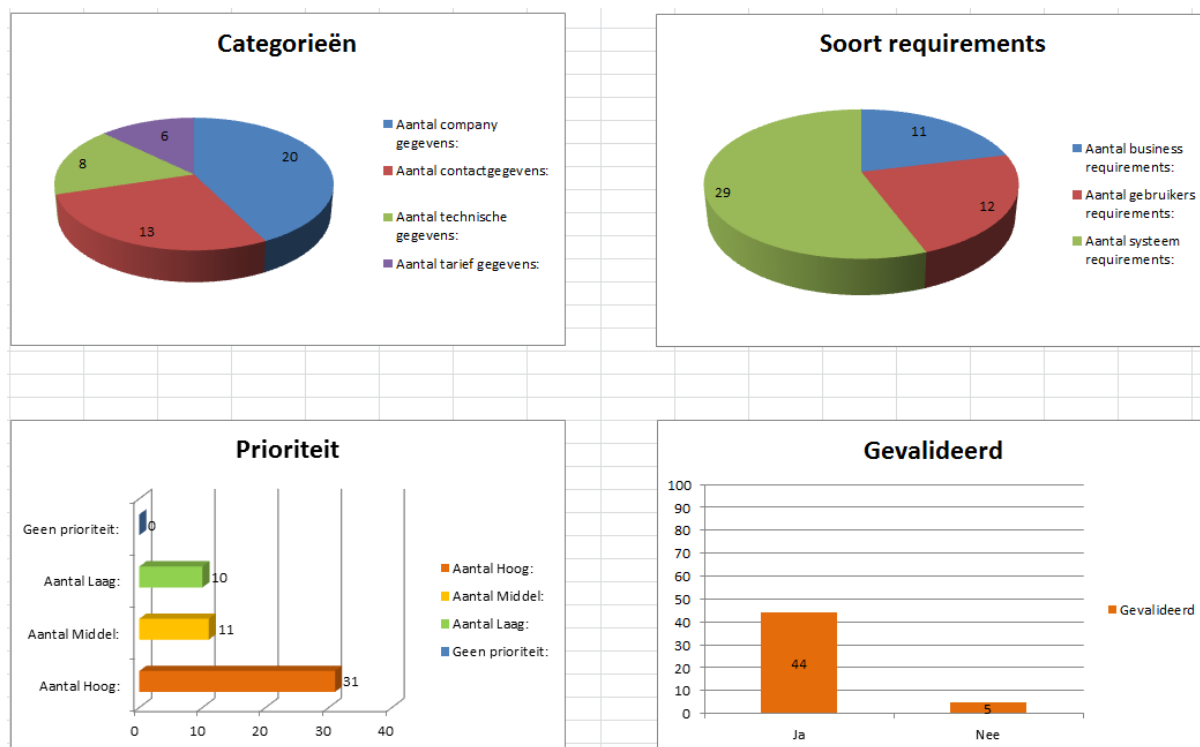
- Nummer – Uniek nummer
- Categorie – Met wat voor gegevens heeft de requirement te maken
- Type – Functioneel of niet functioneel
- Soort requirement – Business-, Gebruiker- of systeemrequirement
- Prioriteit – Welke prioriteit heeft de requirement
- Use case – Bij welke use case hoort de requirement
- Aangemeld door – Wie heeft de requirement "aangemeld"
- Gevalideerd – Is de requirement geverifieerd?
- Gevalideerd door – Door wie is de requirement geverifieerd?

Zie hieronder hoe de tool eruit ziet met een aantal requirements die zijn ingevuld:

	Requirements	Categorie	Type requirement	Soort requirement	Prioriteit	Use Case	Aangemeld door:	Gevalideerd	Gevalideerd door
1	Van 80% van de klanten is geregistreerd wat voor soort communicatie zij met Infodis hebben	Technische gegevens	Functioneel	Business requirement	Hoog	Technical information	Rene	Ja	Rene, Kees
1.1	De gebruiker kan invoeren wat soort voor communicatie een klant met Infodis heeft	Technische gegevens	Functioneel	Gebruikers requirement	Hoog	Technical information	Rene	Ja	Rene, Kees
1.2	Het systeem kan de techniek waarmee onderling gecommuniceert wordt vastleggen	Technische gegevens	Functioneel	Systeem requirement	Hoog	Technical information	Rene	Ja	Rene, Kees
1.3	Het systeem kan vastleggen wat voor soort berichten een klant met Infodis uitwisselt	Technische gegevens	Functioneel	Systeem requirement	Hoog	Technical information	Rene	Ja	Rene

In de tool heb ik ook een dashboard gemaakt waarbij ik door middel van grafieken verschillende soorten informatie kon bekijken over de requirements. Dit heb ik gebruikt om overzicht te krijgen over de lijst met requirements. Ik kon bijvoorbeeld in één overzicht zien hoeveel requirements er waren, hoeveel van een bepaalde categorie, etc... Op deze manier kon ik bijvoorbeeld zien welke use cases veel requirements hadden met hoge prioriteit. Ook kon ik bewaken hoeveel van de requirements gevalideerd waren. Dit hielp mij om te bepalen waarop ik me moest focussen. Als ik zag dat er nog veel requirements niet waren gevalideerd plande ik vervolgens validatie activiteiten in.

Hieronder een impressie van het dashboard uit de requirements tool:



### 5.4.3 Requirements elicitation

Ik ben begonnen met het lezen van documentatie over de Infodis systemen. Hierin werd uitgelegd hoe de systemen functioneerden en wat ze precies voor taken hadden. Daarnaast heb ik ook de product sheets van de verschillende diensten gelezen om hier meer inzicht in te krijgen. Ook kreeg ik veel uitleg van mijn begeleider over de systemen en diensten. Dit hielp mij om meer kennis over het domein te krijgen waarin dit project zich afspeelt.

In interviews waren inmiddels al wensen en eisen naar voren gekomen. De medewerkers die ik interviewde waren zich ervan bewust van dat hun input verwerkt zou worden in de oplossing. De wensen en eisen waren daarom ook erg professioneel. Hieruit kon ik direct al requirements verzamelen. Vervolgens heb ik ook nog de bestaande systemen als TMS en TopDesk geanalyseerd waardoor ik nog meer requirements verzamelen. Bij de analyse van de systemen kwam ik erachter dat er veel incomplete gegevens waren. Hier heb ik vervolgens vragen over gesteld aan verschillende medewerkers en het bleek dat dit als zeer vervelend werd ervaren. Hieruit zijn meerdere requirements over de compleetheit van informatie ontstaan. Als voorbeeld:

- Er moet iemand verantwoordelijk worden voor de gegevens in het systeem
- Bij alle contactpersonen moeten een telefoonnummer zijn ingevuld

#### 5.4.4 Requirements analyse

De volgende stap was om alle verzamelde requirements te analyseren om daarvan de essentie te bepalen. Daarbij heb ik gebruik gemaakt van een checklist uit het boek "Succes met de requirements!". Hiermee heb ik bepaald of het requirement goed is geformuleerd. Deze checklist is als volgt:

- Waarom hoort dit requirements thuis in de uiteindelijke verzameling requirements?
- Wat is de bron van het requirement en is die bron valide?
- Betreft het hier een samengesteld requirement dat moet worden opgesplitst in afzonderlijke, elementaire requirements?
- Is er een onnodige ontwerpbeslissing genomen in het requirement?

Ook heb ik de verzamelde requirements verdeeld onder een aantal categorieën. De categorieën zijn bepaald op basis van de verschillende gegevens soorten die in de oplossing bijgehouden moesten worden. En er zijn aparte categorieën voor system-wide requirements en requirements die van toepassing waren op de database. Zie hieronder de verschillende categorieën die ik heb benoemd:

- Contactgegevens
- Technische gegevens
- Company gegevens
- Tarief gegevens
- Service gegevens
- System-wide
- Database

Het plaatsen van requirements in categorieën helpt om overlappende requirements te ontdekken. Dit is omdat de requirements op deze manier worden gegroepeerd waardoor het makkelijker is om de requirements met elkaar te vergelijken. En als bepaalde requirements niet in een categorie geplaatst kunnen worden wijst dit op ontbrekende categorieën of nog niet ontdekte delen van de oplossing.

#### 5.4.5 Requirements specificatie en validatie

De geanalyseerde requirements zijn vervolgens gespecificeerd. Hierbij zijn deze definitief in de requirements tool ingevoerd. In de Elaboration fase zijn de requirements verder gespecificeerd. Op basis van de gespecificeerde requirements zijn use cases en een use case diagram opgesteld.

In de Inception zijn nog niet alle requirements gevalideerd. Dit komt omdat het eliciteren van requirements een doorlopend proces was. Requirements die aan het einde van de Inception zijn gespecificeerd zijn daarom in de eerste Elaboration fase gevalideerd.

Het valideren is gedaan door medewerkers van support, de product manager en mijzelf. Hierbij het ik zelf vooral gekeken of de requirements goed waren opgesteld. Inhoudelijk zijn de requirements beoordeelt door personen uit de organisatie.

#### 5.4.6 Use cases beschrijven

Om de functionaliteiten van het systeem gedetailleerd te beschrijven heb ik gebruik gemaakt van use cases. In de use cases is beschreven hoe de interactie tussen systeem en gebruiker verloopt. Het benoemen en beschrijven van use cases helpt om een beeld te krijgen van alle functionaliteiten. Om een totaal beeld van het systeem te krijgen moeten alle functionaliteiten benoemd worden. Dit totaal beeld laat zien hoe alle functionaliteiten op elkaar aansluiten. Met behulp van een use case model is het systeem ook voor relatieve buitenstaanders te begrijpen.

Aan de hand van de Vision en gespecificeerde requirements had ik genoeg informatie beschikbaar om use cases te benoemen. Bij elke use case heb ik ook bepaald welke actoren ermee verbonden waren. Deze actoren waren al eerder benoemd in de Vision als stakeholders. De use cases met de actoren heb ik opgenomen in een use case diagram.

Voordat ik de use cases kon gaan uitschrijven heb ik eerst bepaald wat de inhoud van mijn use case template zou zijn. Use cases kunnen heel simpel zijn maar ook heel uitgebreid. Dit moet afgestemd worden op het doel waarvoor ze gebruikt worden. Ik heb ervoor gekozen om de template hieronder te gebruiken. Deze template is voldoende gedetailleerd voor het doel in dit project. Het doel van de use cases is namelijk het eenduidig vastleggen en overbrengen van de functionaliteiten van het systeem.

Use case	
Doel	
Scenario (80%)	<b>Pre conditie:</b> <b>Scenario:</b> <b>Post conditie:</b>
Scenario (20%)	
Business activity	
Belangrijkste actors	
Secundaire actors	

Voor het schrijven van de use cases heb ik gebruik gemaakt van het boek "Writing Effective Use Cases" van Alistair Cockburn. Hierin wordt het hele concept van use cases simpel maar toch gedetailleerd uitgelegd. Er wordt in beschreven hoe je use cases bruikbaar maakt en de kwaliteit hoog houdt. De stappen in de werkwijze die hij beschrijft sloten vaak goed aan op stappen in OpenUP. Het doel van het systeem en de functionaliteiten had ik bijvoorbeeld al beschreven in het Vision artefact.

De use cases heb ik in eerste instantie globaal uitgeschreven. Dit heb ik gedaan zodat ik niet veel tijd kwijt was aan details die mogelijk toch nog zouden veranderen. Verder in de diepte uitwerken zou ik in opvolgende iteraties doen. In het boek van Alistair Cockburn wordt deze manier van werken als tip meegegeven. Verder heb ik vooral geprobeerd om de use cases simpel te houden. Ze moesten makkelijk leesbaar zijn en mij ondersteunen in het vastleggen van de functionaliteiten van het systeem.

#### 5.4.7 Inventarisatie informatie objecten

In het nieuwe systeem moet de informatie die op verschillende plekken is opgeslagen samengevoegd worden op één locatie, de centrale database van het systeem. Om erachter te komen welke informatie het allemaal betreft heb ik diverse medewerkers geïnterviewd. Mijn aanpak was een korte vragenlijst samenstellen en vervolgens tijdens het interview veel door te vragen. Hieronder de kernvragen die ik tijdens de interviews wilde behandelen:

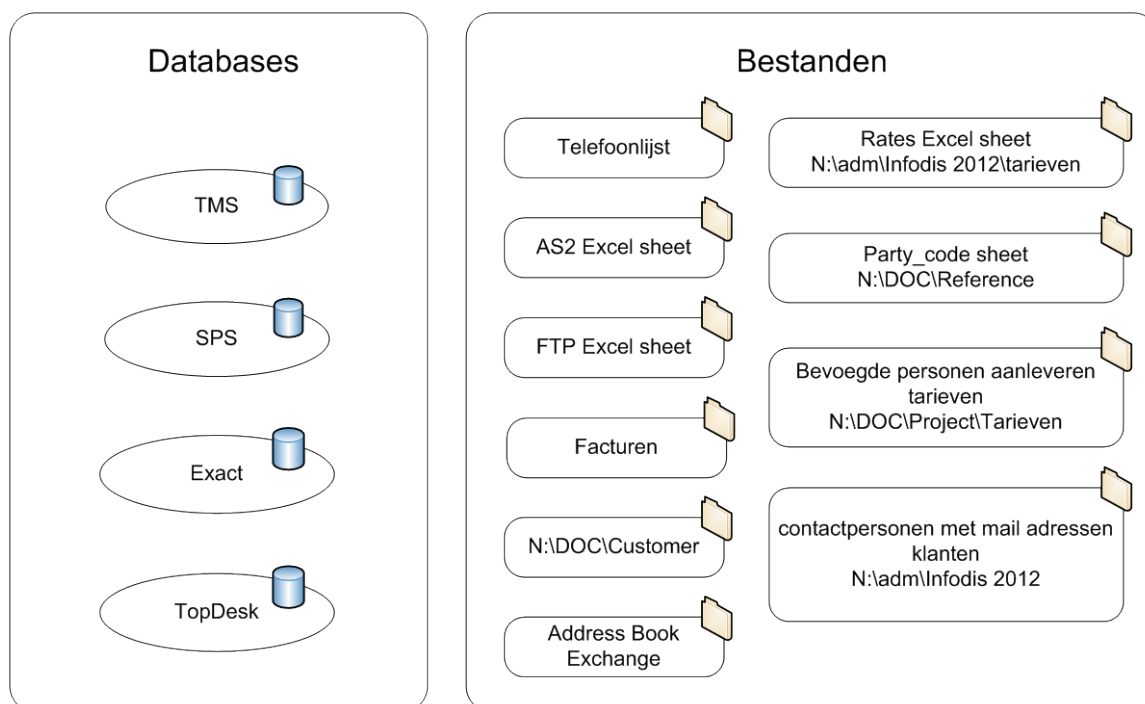
- *Welke klantgegevens worden er nu vastgelegd? (Infodis breed)*
  - *Waar zijn deze klantgegevens te vinden? (Welk systeem, welk bestand, etc)*
- *Welke klantgegevens (specifiek) heb jij nodig?*
- *Hoe vaak maak je gebruik van deze gegevens?*
- *Ontbreken er soms ook gegevens?*
- *Waar loop je tegenaan bij het vinden van klantgegevens?*

Deze aanpak werkte goed binnen Infodis. Dit omdat zeer formele interviews en grote vragenlijsten niet passen binnen de bedrijfscultuur bij Infodis. De cultuur is meer informeel en open.

De vragen zijn opgesteld om voor alle afdelingen/functies bruikbaar te zijn. Hieronder zal ik per vraag uitleggen wat ik ermee wilde bereiken.

- De eerste vraag gaat over meerdere afdelingen heen. Hiermee wilde ik bekijken in hoeverre informatie objecten organisatie breed bekend zijn en of er overlap is.
- Ik wilde ook weten welke gegevens er dan per functie specifiek van belang waren. Hoe vaak die gegevens dan gebruikt werden per dag. Veel mutaties of weinig?
- Vervolgens wilde ik weten of er gegevens ontbraken omdat ze niet ingevuld worden of omdat ze simpelweg niet bijgehouden worden. En of de wens er is om die gegevens toch bij te gaan houden.
- Tenslotte een vraag over knelpunten bij het vinden van de gegevens. Hiermee wilde ik kijken welke functionaliteiten een nieuwe systeem zou moeten bieden om hieraan tegemoet te komen.

Na de interviews uitgewerkt te hebben heb ik een opsomming gemaakt van de informatie objecten. Hieraan is niet te zien wat de inhoud is van de objecten, die ben ik pas later gedetailleerd gaan inventariseren. Hieronder de opsomming van de belangrijkste informatie objecten. Ik heb een scheiding gemaakt tussen databases en losse bestanden met informatie.



Nu ik deze opsomming had kon ik de informatie objecten beter gaan analyseren. Daardoor kon ik echt zien wat voor informatie er allemaal in opgeslagen was. Om de kwaliteit van de informatie hoog te houden ben ik gaan bepalen wat nuttige informatie was om in het nieuwe systeem op te nemen. Dit heb ik uiteraard samen met de medewerkers gedaan.

Om beter te snappen wat nuttige informatie was ben ik gaan lezen over "Information Quality Management" in het boek: Improving Data Warehouse and Business Information Quality. Daarbij heb ik me vooral gefocust op de "Pragmatic Information Quality". Dat houdt ook wel in hoeveel waarde de informatie toevoegt aan de organisatie. Een voorbeeld uit de situatie bij Infodis is als volgt. In de SPS database worden voor elke booker ook adresgegevens vastgelegd. Deze adresgegevens worden eigenlijk nooit gebruikt omdat een booker een virtueel iets is. Deze bestaat alleen in het systeem. De informatie levert daarom ook geen meerwaarde aan de organisatie. Dit is een voorbeeld van lage pragmatische informatie kwaliteit.

Om te bepalen welke informatie van hoge pragmatische kwaliteit was heb de informatie naast de volgende definitie gehouden:

#### Quality characteristic

The *right* data  
With the right *completeness*  
In the right *context*  
With the right *accuracy*  
In the right *format*  
At the right *time*  
At the right *place*  
For the right *purpose*

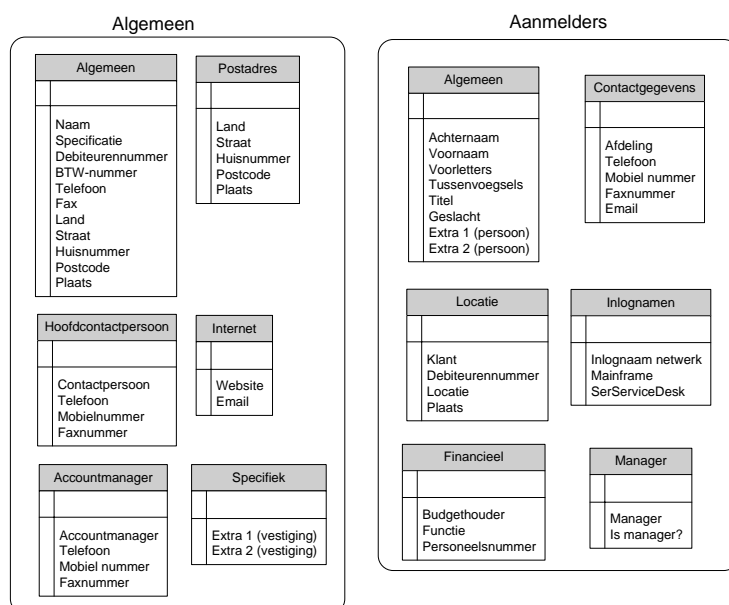
#### Knowledge worker benefit

The data I *need*  
All the data I need  
Whose *meaning* I know  
I can *trust* and rely on it  
I can *use* it *easily*  
When I need it  
Where I need it  
I can *accomplish* our objectives and delight our customers

Er moest ook bepaald worden welke van alle opgesomde gegevens er allemaal onder klantgegevens zouden vallen. Dit heb ik in overleg met mijn bedrijfsmentor bepaald. Er is afgesproken dat de volgende gegevens tot klantgegevens zouden behoren:

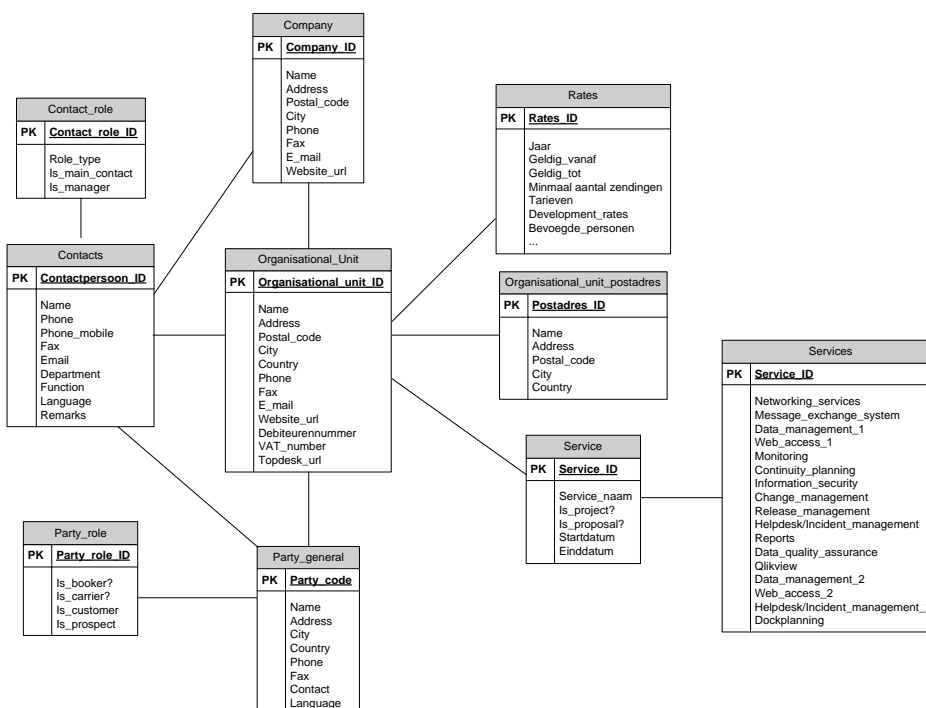
- Company gegevens
  - Naam, adres, telefoon, etc.
  - Eventuele organisational units
- Party gegevens
  - Unieke code in het systeem
- Contact gegevens
  - Naam, telefoon, e-mail, etc.
- Service gegevens
  - Welke diensten levert Infodis aan de klant

Van de informatie objecten die ik had gevonden heb ik de gegevens die van toegevoegde waarde waren gegroepeerd en vervolgens tot tabellen met veldnamen gevormd. Dit om ze geschikt te maken om in een database gebruikt te worden. De informatie uit Topdesk is bijvoorbeeld tot de volgende tabellen en veldnamen gevormd:





Op deze manier heb ik de basis gelegd voor een eenvoudig data model. Dit datamodel moest alle informatie gaan bevatten die in het nieuwe “systeem” beschikbaar moest zijn. Het datamodel is gegroeid naarmate er meer informatie objecten geanalyseerd waren. Zie hieronder een voorbeeld van de opzet aan het einde van de Inception fase:



In de Elaboration fase is het model uitgegroeid tot een volwaardig relationeel datamodel.

#### 5.4.8 Proof of concept ontwerpen

Het Proof of Concept (POC) was bedoeld om een concept oplossing te kunnen demonsteren mijn opdrachtgever. Dit in vorm van een aantal geanimeerde schermontwerpen.

Het maken van een POC had ik al besloten tijdens het opstellen van het afstudeerplan en is meegenomen naar het project plan. Zonder erbij na te denken heb ik het in de eerste iteratie ingepland. Ik heb me daarbij niet gelijk gerealiseerd dat dit geen eenvoudig project is met een paar eenvoudige use cases. Het project was veel groter en complexer dan dat. Dit komt waarschijnlijk omdat ik met zoveel dingen tegelijk bezig was dat mijn aandacht voor dit onderdeel verslapte. Maar naarmate het einde van de Inception fase in zicht kwam had ik nog lang niet alle informatie objecten geanalyseerd. Daarnaast was nog maar net duidelijk wat voor vorm de uiteindelijke oplossing zou krijgen. Ik realiseerde me toen dat het veel verstandiger was om het maken van een POC in een volgende iteratie in te plannen. Ik heb daarom besloten om het POC te verschuiven naar de Elaboration 1 iteratie. In deze volgende iteratie zou veel concreter worden welke vorm de oplossing zou krijgen. Ook zou er dan meer voortgang zijn geboekt op het gebied van de informatie objecten en de analyse daarvan. Op basis daarvan zou ik vervolgens een POC kunnen maken.

## 5.5 AGREE ON TECHNICAL APPROACH

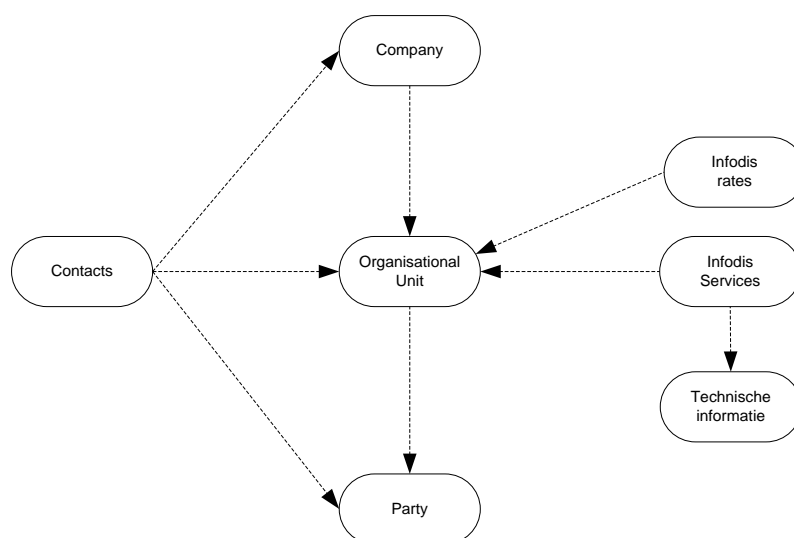
### 5.5.1 Opstellen Architecture Notebook

Om de architectuur van de oplossing vast te leggen heb ik een Architecture Notebook opgesteld. In dit document wordt beschreven wat de gedachte achter de architectuur is en wat hiermee bereikt moet worden. Het is in feite een verlengstuk van de Vision maar dan met een technische inhoud. Net als bij de Vision heb ik hierbij gebruik gemaakt van een OpenUP template.

Bij het Architecture Notebook had ik net als bij het POC het probleem dat er nog geen besluit genomen was over de meest geschikte oplossing. Daarom kon ik in de Inception fase vooralsnog alleen globaal invulling geven aan het Architecture Notebook. De besluiten die wel al genomen waren zijn wel opgenomen.

### 5.5.2 Technische werking van het systeem vastleggen

In de Inception is er een besluit genomen ten opzichte van de structuur die de oplossing moest aanhouden. Het systeem moest een front-office laag en back-office laag krijgen. Hieronder is een model van de structuur te zien zoals dit is bedacht in de Inception fase.



De Company en Organisational unit vallen onder de front-office laag. De Party valt onder de back-office laag. Het idee is ontstaan uit het feit dat er een verschil is tussen de entiteiten voor de front-office en voor de back-office van Infodis. Hiertussen is een scheiding in de situatie bij Infodis. Op de volgende pagina een uitleg over deze scheiding en waarom deze in het systeem opgeheven moet worden.

## Party's

Het systeem van Infodis werkt op basis van party's. Een party kan van alles zijn, een carrier, booker, warehouse, pickup address, etc... Deze party's zijn echter niet altijd duidelijk getypeerd. Zo is niet direct te zien of een party een carrier, booker of iets anders is. Ter illustratie een voorbeeld van een party die carrier is in het TMS van Infodis:

```

DSU07  DSU NO
Addresses
General
Name1  DSU Noorwegen
Addr
City
Cntry  NO
Cntry
Phone
Fax
Contactperson

```

En een party die booker is in het TMS:

```

CAN18  CBS XSales
Addresses
General
Name1  CBS XSales
Addr
City
Cntry  NL
Cntry
Phone
Fax
Contactperson

```

Tussen deze twee party's zijn zichtbaar geen verschillen, buiten de naam om. Om de verschillen toch zichtbaar te maken moet verder gezocht worden in het TMS. De informatie over party's is daardoor niet bruikbaar voor front-office medewerkers omdat deze niet veel werken met het TMS systeem.

Daarnaast is het zo dat één company meerdere party's kan hebben in het systeem. Canon heeft bijvoorbeeld wel tien verschillende party's in het TMS. Het TMS heeft deze party's nodig om zijn verschillende functies uit te voeren. Het is de Development afdeling die party's gebruikt om het TMS in te regelen.

## Company's

Company's daarentegen zijn fysieke bedrijven in de werkelijkheid. Als men binnen Infodis praat over de klanten dan praten ze over company's, niet over party's. Daarom is het voor de front-office veel belangrijker om gegevens over company's vast te leggen. De front-office wilt weten wat het adres is, welke contactpersonen er zijn, welke diensten er afgenomen worden, hoe die diensten zijn ingeregeld, etc...

De bedachte structuur waaraan de oplossing moet voldoen helpt om een brug te slaan tussen de front-office en de back-office. Deze gegevens van deze twee moeten samen zijn om compleet te zijn. De structuur die nu bedacht is maakt het mogelijk om in twee richtingen duidelijkheid te scheppen.

## 5.6 ITERATION ASSESSMENT

De iteratie is binnen de geplande tijd afgerond maar wel met afwijkingen op de planning. Zo is het Proof of Concept verschoven naar de Elaboration fase. Dit is besloten omdat er nog niet genoeg informatie beschikbaar was waarmee een Proof of Concept ontwikkeld kon worden. Dit is tevens een leer punt voor toekomstige projecten. Ik moet goed opletten tijdens het maken van de planning zodat ik geen zaken opneem die ik op dat moment niet waar kan maken.

Het inventariseren van de knelpunten, wensen en eisen en de informatie objecten is succesvol verlopen. Deze zaken zijn opgenomen in het Vision document, requirements tool en in het datamodel. Hier heeft het meeste tijd in gezeten deze iteratie maar het heeft ook het meest opgebracht voor het project. Hierdoor is een heldere omschrijving van de knelpunten beschikbaar en ook een beeld van wat een oplossing moet bieden. Dit bewijst voor mij opnieuw dat het verstandig is om rustig te beginnen met een project ook al kost dat veel tijd. Eerst het domein verkennen, de problemen begrijpen en pas dan werken naar een oplossing. Toen ik begon met deze afstudeerstage kon ik mij namelijk veel minder inleven in de werkzaamheden van Infodis en de knelpunten die er speelden. Na deze iteratie is dat veranderd omdat ik mij nu zelf onderdeel voel van Infodis en de problemen heb meegemaakt.

Het Architecture Notebook is nog zeer oppervlakkig. Dit komt omdat er nog niet veel duidelijk was omtrent het technische verhaal achter de te ontwikkelen oplossing. In de Elaboration fase kan hieraan meer invulling gegeven worden. Dit vanwege een belangrijk besluit dat aan het einde van deze iteratie is genomen, meer hierover in het volgende hoofdstuk. Er is echter wel voortgang geboekt omdat de structuur van de oplossing al is vastgelegd. En er is duidelijk welke gegevens binnen dit project onder klantgegevens vallen.

### 5.6.1 Besluit geschikte oplossing

Aan het einde van deze iteratie is er een besluit genomen t.o.v. de doelstelling van de opdracht. De doelstelling van de opdracht is als volgt:

*“Het ontwerpen van een informatiesysteem (of eventueel aanpassen van een bestaand systeem) en het implementeren van de belangrijkste functionaliteiten daarvan zodat de knelpunten tegemoet kunnen worden gekomen.”*

De conclusie die er was na de Inception fase is dat er geen bestaand systeem is wat voldoende aangepast kan worden. De systemen die er zijn kunnen niet aangepast worden (closed source). Of het zijn systemen die dan worden belast met allerlei data die daar eigenlijk niet thuis hoort. Het is bijvoorbeeld niet de bedoeling om allerlei klantgegevens op te gaan slaan in de database van het Transport Management System. In dat systeem zijn namelijk alleen de naw gegevens nodig van de klanten. Ook een systeem aanschaffen zou niet voldoen omdat daar nog veel kostbaar maatwerk aan gekoppeld zou zitten. Als er een pakket aangeschaft moet worden dan zou er in de richting van een CRM pakket gekeken moeten worden. Dit zou echter betekenen dat het customer support pakket TopDesk ook vervangen zou worden. Dat was geen optie omdat dit vrij recentelijk was aangeschaft door Infodis. Ook waren er veel werkzaamheden afhankelijk van TopDesk.

Het uiteindelijke besluit is geworden dat er een nieuw systeem ontwikkeld zou gaan worden. Het Infodis CRM met daarachter de Infodis CRM database. Dit systeem zou op maat worden ontwikkeld voor Infodis en door Infodis.

## 6 FASE 2: ELABORATION 1

Deze iteratie was erop gericht om zoveel mogelijk informatie objecten te analyseren en daarvan te bepalen welke gegevens van hoge pragmatische kwaliteit waren. Dit was nodig zodat er een volledig ontwerp van de database gemaakt zou kunnen worden. Omdat de database de basis is van het Infodis CRM heeft dat onderdeel een hoge prioriteit gekregen.

Daarnaast zou er ook verder gewerkt worden aan de use cases en een Proof of Concept. Tijdens het werken aan deze producten zouden hoogstwaarschijnlijk zaken aan het licht komen die van toepassing zouden zijn op de database. Deze zaken konden dan gelijk meegenomen worden in het database ontwerp.

### 6.1 PLAN ITERATION

Omdat ik in dit document bij het iteratie plan van de Inception al heb uitgelegd waarom een iteratie plan wordt gemaakt zal ik dat hier niet nogmaals uitleggen. Wel worden belangrijke besluiten rondom de iteratie planning beschreven.

High level objectives van deze iteratie:

- Risico's use cases inschatten
  - Aan de hand van de risico's de prioriteit van use cases vaststellen.
- Requirements specificeren
  - Verder gaan met het eliciteren van requirements en het specificeren hiervan. Er wordt vanuit gegaan dat in de Inception fase nog niet alle requirements zijn gespecificeerd. Of dat er wellicht requirements zijn veranderd.
- Proof of Concept maken
  - Aan de hand van verzamelde requirements een POC maken om de werking van het systeem aan te tonen.
- Database ontwerpen
  - Er is besloten dat er een nieuwe CRM systeem gemaakt moet worden. Hieraan moet een database gekoppeld zijn waarin de klantgegevens worden opgeslagen. Deze iteratie zal het ontwerp daarvan plaatsvinden.

De work items zijn de werkzaamheden die verricht moeten worden om de doelen te bereiken. Dat waren in deze iteratie:

- Specificeren requirements
- Valideren requirements
- Vision document bijhouden
- Risico analyse use cases
- Glossary bijhouden
- Database ontwerpen
- Use cases uitwerkingen
- Proof of Concept ontwerpen
- Opstellen test cases
- Architecture Notebook aanvullen

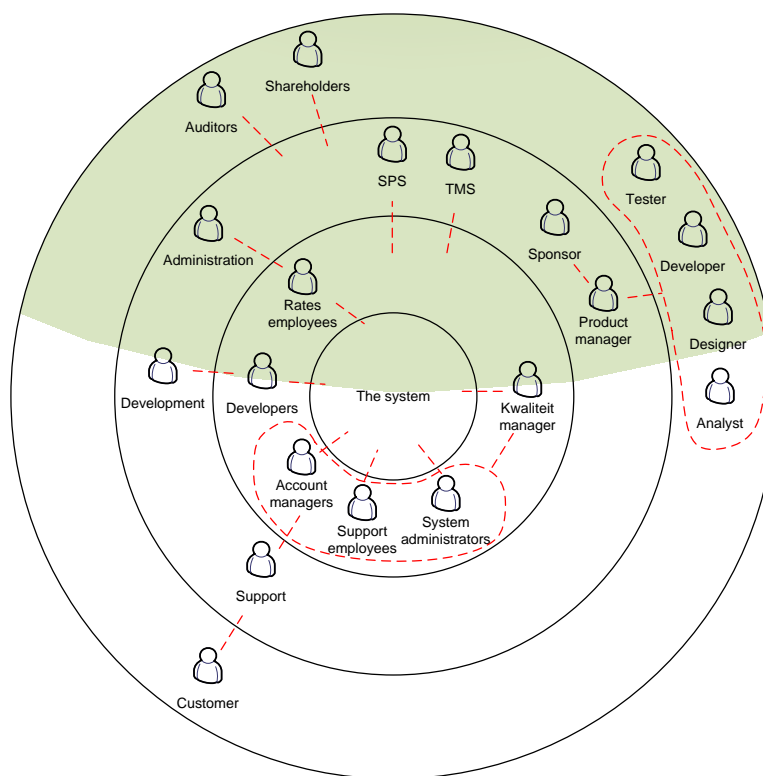
Zie ook het iteratie plan in de bijlage.

## 6.2 IDENTIFY AND REFINE REQUIREMENTS

### 6.2.1 Stakeholder analyse

Op advies van mijn begeleider vanuit school heb ik een meer gedetailleerde stakeholder analyse gemaakt. Dit als middel om alle stakeholders te ontdekken en hun belangen bij het project te begrijpen. Deze stakeholder analyse zou die eerdere stakeholder analyse uit de Inception vervangen.

Ik heb bij het maken van de stakeholder analyse gebruik gemaakt van een artikel van Ian F. Alexander genaamd *"A Better Fit – Characterising the Stakeholders"*. Hierin beschrijft hij dat het noodzakelijk is om alle stakeholders rondom een project te analyseren. Dit wordt vaak niet gedaan omdat een systeem voornamelijk op operationele rollen is gericht. Daarbij worden meestal alleen de operationele stakeholders geanalyseerd. Er wordt verder niet geanalyseerd of er wellicht andere stakeholders zijn met een belang in het systeem. Dat kan resulteren in een systeem dat voor niet-operationele stakeholders volledig ongewenst is. Om dat te voorkomen beschrijft hij een manier waarbij alle stakeholders ontdekt en gemodelleerd kunnen worden, het Onion model.



Onion model met de stakeholders van Infodis

Het Onion model is een grafische weergave van de stakeholders en hun positie ten opzichte van een systeem/product. Het model is afgeleid van de vele lagen die een ui heeft. De lagen van een ui hebben een steeds grotere afstand vanaf de "kern". In het Onion model hebben deze lagen allemaal een andere betekenis:

- Binnenste ring (De kern): Het systeem of product.
- Eerste ring: De stakeholders of operators die direct met het systeem werken. Deze verwachten dat de binnenste ring hun kan ondersteunen in hun werkzaamheden.
- Tweede ring: Business stakeholders of beneficiaries die niet direct met het systeem werken maar er wel belang bij hebben. Bevat vaak managers van afdelingen, zij sturen de

operationele werkzaamheden aan. De afdelingen/managers hebben er profijt bij als de eerste ring profiteert van een goed systeem (binnenste ring).

- Derde ring: De ruimere omgeving. Externe stakeholders en projectleden van het team dat de oplossing ontwikkeld. De buitenste ring heeft er belang bij dat alle lagen eronder goed functioneren. Een klant bijvoorbeeld heeft belang bij het functioneren van onderliggende lagen omdat deze dan sneller geholpen kan worden.

Ter aanvulling van het Onion model heb ik ook gebruik gemaakt van een template waarin vele stakeholder types zijn beschreven. Deze "Stakeholder Analysis Template" heb ik van Volere.co.uk. Aan de hand hiervan kon ik systematisch alle stakeholder types langsgaan om te bepalen of er binnen Infodis soortgelijke stakeholder waren. Dankzij deze manier van stakeholder analyse kan ik met zekerheid zeggen dat ik alle stakeholders binnen dit project heb benoemd.

Aan de hand van het Onion model is besloten welke stakeholders nog benaderd moesten worden voor vragen en welke niet. Zo is besloten dat de aandeelhouders en klanten van Infodis niet benaderd zullen worden. Aandeelhouders niet omdat deze al kennis hebben van het project en het project steunen. Er zijn echter wel plannen voor een overname waarbij dit zou kunnen veranderen. Hiermee hoefde ik echter geen rekening te houden van mijn begeleider. Dit omdat het project ook voor de nieuwe eigenaar van toegevoegde waarde zal zijn. Zij kunnen alleen maar profiteren van een database waarin alle klantgegevens centraal zijn vastgelegd.

Klanten worden niet benaderd omdat zij geen idee hebben wat Infodis voor klantgegevens nodig heeft en wilt gaan vastleggen. Dat is logisch omdat klanten zich helemaal niet bezig houden met wat Infodis intern doet.

Wel moest ik nog bepalen wat de rol zou worden van het TMS en SPS systeem in het tot stand komen van een CRM database. Er waren namelijk al klantgegevens in deze systemen vastgelegd. Deze zouden een goed startpunt kunnen vormen voor het vullen van een CRM database.

En uiteraard is naar voren gekomen dat ik in dit project afhankelijk ben van de Development afdeling. Deze zal mij moeten ondersteunen bij het ontwerpen en bouwen van het Infodis CRM. Samen met mijn begeleider is daarom besloten om Willem Kosmeijer aan dit project te koppelen. Hij is een developer en zou het grootste gedeelte van zijn tijd samen met mij aan dit project gaan werken.

### **6.2.2 Wensen en eisen verzamelen**

In de eerste Elaboration iteratie ben ik verder gegaan met het verzamelen van wensen en eisen. Ik ben vragen gaan stellen over de voorgestelde functionaliteiten in de Inception fase om te kijken of deze compleet waren. Hierbij merkte ik dat het steeds meer de diepte in ging qua detail. Er werden vrijwel geen "nieuwe" functionaliteiten aan het systeem toegevoegd. Dat gaf mij aan dat ik de functies in de Inception fase goed had geïnventariseerd.

Er kwamen wel wensen om bepaalde gegevens types toe te voegen die nog ontbraken. Als voorbeeld het ontbreken van velden voor financiële gegevens. Een wens was het toevoegen van een BTW nummer en debiteurnummer aan de gegevens van een company.

Eisen aan de performance van het systeem kwamen er ook. Bijvoorbeeld dat het systeem in ieder geval sneller dan Topdesk zou moeten zijn. Alle medewerkers gebruiken dit pakket en ondervinden er regelmatig hinder aan omdat het erg traag is. Een snel systeem zou daarom de drempel tot het gebruik ervan heel erg verlagen.

En ook eisen aan de gebruikersvriendelijkheid en het uiterlijk van het systeem. Bijvoorbeeld dat het standaard Infodis lettertype zou worden gebruikt en dat "Infodis blauw" in het ontwerp werd verwerkt. Het systeem moest herkenbaar zijn wat betreft de interface. Hiermee werd bedoeld dat

de interface van het systeem niet significant verschilde van andere systemen die in gebruik waren. Dit omdat er al meerdere verschillende systemen in gebruik waren. Een suggestie was om de interface te matchen met of te baseren op de Shipping Tracking System website van Infodis.

Een eis aan de database was dat deze bij introductie van het systeem gevuld zou zijn. Veel medewerkers konden van te voren al zeggen dat ze geen tijd hadden om handmatig tientallen records toe te gaan voegen. Daarom wilden ze graag dat de het merendeel van de gegevens al ingevoerd zou zijn en dat ze eventueel zelf nog de details in konden voeren.

Ten slotte waren er nog wenselijke features zoals het kunnen instellen van notificaties. Hierbij ging het om notificaties bij het verlopen van connection certificaten of diensten die geleverd werden. Als de start- en einddatum werden ingevoerd in de database kon het systeem hier een notificatie van maken. Het vergeten van bepaalde zaken zou hiermee voorkomen kunnen worden.

### 6.2.3 Requirements valideren

Deze fase is er meer aandacht geweest voor het valideren van requirements. Dit was nodig omdat in de Inception fase nog niet alle requirements gevalideerd waren en er in de Elaboration nieuwe requirements bij kwamen. Ik wilde niet achterlopen op dit gebied dus heb ik hiervoor tijd ingepland.

Binnen Infodis wordt er niet actief aan requirements management gedaan. Daarom hadden zij ook geen eisen om de requirements mee te valideren. Ik heb daarom de eisen uit het boek "Succes met de requirements!" gebruikt om de requirements te valideren. Daarbij ben ik de volgende checklist langs gegaan:

- Uniek identificeerbaar
- Atomair
- Eenduidig
- Bevat geen implementatie details
- Traceerbaar
- Testbaar
- Prioriteit bepaald

Als een requirement aan al deze eisen had voldaan gaf ik in de requirements tool aan dat de requirement was gevalideerd. Daarmee was echter nog niet bepaald of de requirement inhoudelijk voldeed aan de behoefte van de stakeholders. Daarom zijn de requirements inhoudelijk beoordeeld door medewerkers van support en de product manager. Als dit gedaan was gaf ik in de requirements tool aan door wie de requirement was gevalideerd. Door deze werkwijze te hanteren kon ik zien of de requirement zowel aan de eisen uit het boek als aan de eisen van de stakeholders had voldaan.

## 6.3 DEVELOP THE ARCHITECTURE

Aan het einde van de Inception fase was duidelijk geworden dat er een nieuwe systeem ontwikkeld zou gaan worden, het Infodis CRM. Dankzij deze beslissing kon er nu meer invulling gegeven worden aan het Architecture Notebook. De belangrijkste beslissingen worden hieronder vermeldt. Meer informatie hierover is te vinden in het Architecture Notebook.

### Systeem in het Engels

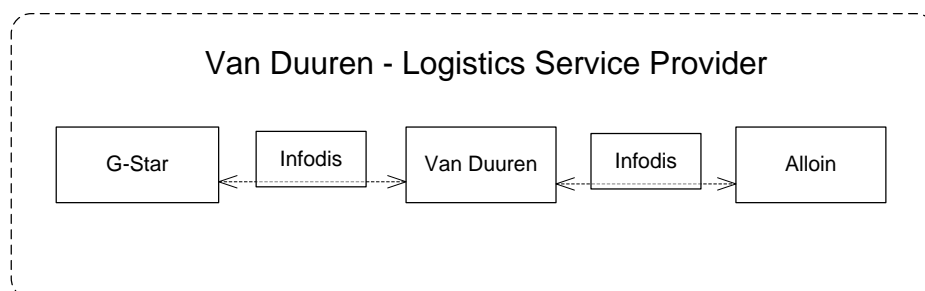
Zo is er besloten dat het nieuwe systeem volledig in het Engels ontwikkeld gaat worden. Infodis is namelijk steeds vaker internationaal operationeel en zijn veel gebruikte vaktermen in het Engels. Voor de applicatie betekend dit dat alle user interfaces in het Engels gemaakt zullen moeten worden. Ook de achterliggende Infodis CRM database zal volledig in het Engels ontwikkeld worden. Namen van entiteiten en velden zullen allemaal in het Engels zijn.



## Gegevens van allerlei soorten company's

Het systeem moet geschikt zijn om gegevens voor allerlei soorten company's op te slaan. Dit moet omdat Infodis niet alleen met klanten in contact staat maar ook met klanten van de klant. Het is soms noodzakelijk om direct contact met zo'n "klant van klant" te hebben. Daarom moeten ook gegevens van deze "klant van klant" company vastgelegd worden. Er zijn veel van zulke uitzonderingen bij Infodis. Deze kwam ik op het spoor tijdens gesprekken met mijn begeleider. Hij wist dat er van zulke uitzonderingen waren en dat ook deze in het systeem verwerkt moesten worden. Zie een voorbeeld van een soortgelijke situatie op de volgende pagina.

Van Duuren is klant van Infodis en regelt de logistiek weer voor hun eigen klant G-Star. Het vervoer besteed Van Duuren dan weer uit naar Alloin. Van Duuren wordt daarom ook wel een Logistics Service Provider genoemd. De zendingen worden wel geboekt en vastgelegd via het systeem van Infodis. In het model hieronder is te zien hoe Infodis tussen de verschillende company's zit.

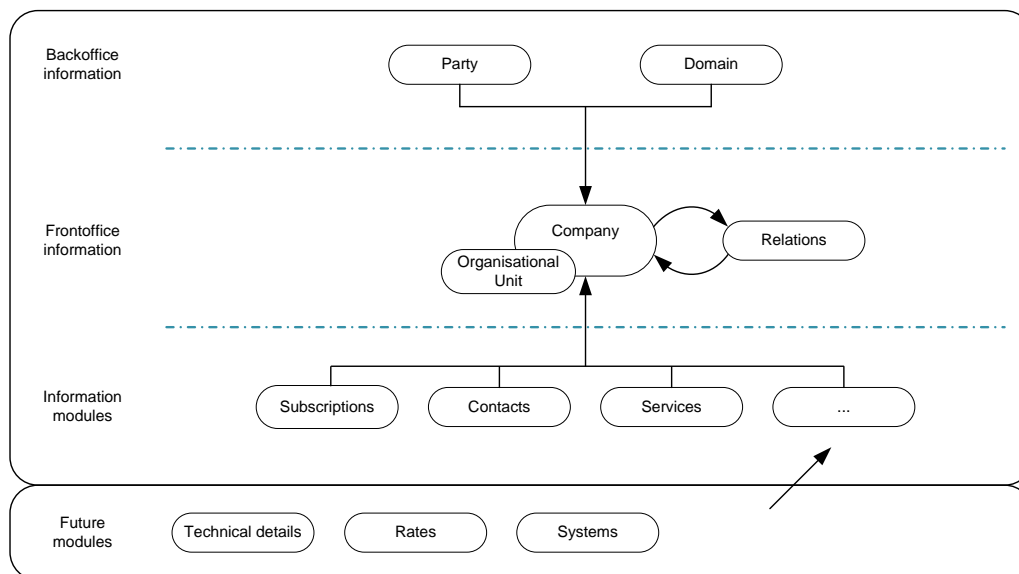


Infodis moet soms contact opnemen met deze verschillende company's. Daarom moeten ze opgeslagen worden in het Infodis CRM ook al zijn deze partijen niet allemaal een directe klant van Infodis. Er moet echter wel onderscheid blijven tussen echte klanten van Infodis en klanten van de klant. De oplossing hiervoor is dat er types en rollen aan company's worden gekoppeld. Types zijn bijvoorbeeld: Customer, Prospect, Third Party, etc... Dit is een type in relatie tot Infodis. Een company is een customer van Infodis of een potentiële customer. Rollen daarentegen zijn bijvoorbeeld: Shipper, Carrier, Warehouse, etc... Dit is gebaseerd op de werkzaamheden die een company uitvoert, een company kan bijvoorbeeld een warehouse zijn.

Met behulp van modellen als deze heb ik Willem Kosmeijer kunnen tonen waarom ik bepaalde beslissingen heb gemaakt in het ontwerp van de database. En ook waarom ook klanten van klanten in de database opgenomen moeten worden. Dit was nodig omdat hij in eerste instantie een heel andere oplossing had voor het database ontwerp. Na wat uitleg aan de hand van deze modellen kon ik hem overtuigen dat mijn ontwerp beter aansloot op de realiteit in verband met de vele uitzonderingen die bestonden.

## Opbouw van het Infodis CRM

Het model hieronder is ontwikkeld om te tonen hoe de relaties zijn tussen de verschillende gegevens groepen. Dit is een nieuwe versie van het model uit hoofdstuk 4.4.2.



## Interfaces

Het ontwerp van de interfaces is gebaseerd op een oude helpdesk applicatie en op Topdesk. Van beide zijn bepaalde elementen overgenomen in het ontwerp van de nieuwe interfaces. Deze ruwe versies van de interfaces heb ik ook gebruikt om een POC mee te maken. Hieronder is een ontwerp te zien waarin Company gegevens getoond moeten worden.

The screenshot shows a web application interface for the 'Infodis database'. The main content area displays a 'Company' record form with the following sections:

- General information:** Includes fields for Name, Address, Postal\_code, City, Phone, Fax, E-mail, Website\_url, Debiturennummer, VAT\_number, Organisational Unit of, and Notes. There are 'New' and 'Edit' buttons.
- Post address:** Includes fields for Name, Address, Postal\_code, City, and Country.
- Shortcuts:** Includes buttons for TopDesk, SharePoint, N., TMS, and SP5.

The left sidebar contains a navigation menu with 'Home' and 'Company' tabs. Under 'Company', there are 'Party' and 'Reports' sub-tabs. A search bar is present, and a list of company entries is shown below it. A 'Help' button is located at the bottom of the sidebar.

De elementen die in bijna alle interfaces gebruikt gaan worden zijn in het Architecture Notebook beschreven. Dit zijn de volgende elementen:

- Een navigatie paneel aan de linkerkant van de interface.
- Er worden tabbladen gebruikt per gegevens categorie.
- Als er details getoond moeten worden gebeurd dit in een pop-up.

### **Snelkoppelingen naar applicaties**

Een wenselijke feature was dat het systeem snelkoppelingen kan bevatten naar andere applicaties die binnen Infodis gebruikt worden. Als er bijvoorbeeld een company geselecteerd is moet er via een snelkoppeling naar diezelfde klant in TopDesk geswitcht kunnen worden. Dit is wenselijk om te voorkomen dat gebruikers onnodig veel tijd kwijt zijn om tussen verschillende applicaties moeten switchen.

## **6.4 DEVELOP SOLUTION**

### **6.4.1 Risico analyse use cases**

Met een risico analyse heb ik bepaald welke use cases de hoogste prioriteit zouden krijgen. Volgens OpenUP is de use case met de hoogste prioriteit ook de use case waaraan als eerst aandacht besteed moet worden. Dit omdat deze vaak moeilijk te realiseren is en veel waarde toevoegt aan de organisatie. Aan de hand van de risico analyse heb ik dan ook bepaald in welke volgorde ik zou gaan werken en welke use cases aandacht kregen in de eerste Elaboration.

Dit heb ik gedaan door te bepalen wat het verwachte risico is en hoeveel waarde de use case toevoegt aan de business. Het verwachte risico is onder andere afhankelijk van hoe complex de use case is, hoe moeilijk is het programmeerwerk, afhankelijkheid van meerdere actoren, interfaces met andere applicaties, etc.. Het andere punt is de waarde toevoeging van de use case voor de business. Daarbij is gekeken naar hoe vaak een use case gebruikt gaat worden, één keer per maand of elke dag? Hoe breed de vraag is vanuit de organisatie, gebruikt over meerdere afdelingen, hoeveel tijd wordt ermee bespaart? Het inschatten hiervan is in overleg gebeurd met mijn begeleider en met Willen Kosmeijer van Development. Vervolgens is bepaald wat het risico van de use cases was.

Om de risico analyse van de use cases overzichtelijk weer te geven heb ik een tabel opgesteld met alle use cases en daarbij gezet wat het verwachte risico is en de waarde die de use case toevoegt aan de business. Van de resulterende tabel hebt ik ook nog een grafiek gemaakt waarin de verschillende criteria naast elkaar worden gezet.

Zie het Use case document in de bijlagen.

### **6.4.2 Use cases detailleren**

Ook al lag de focus in deze fase op de Infodis CRM database moesten er alsnog use cases verder gedetailleerd worden die in de Inception globaal waren opgesteld. Nu bekend was dat er een nieuw systeem ontwikkeld zou gaan worden betekende dit dat bedacht moest gaan worden hoe het systeem zou interacteren met de gebruiker. Het ontwerp van de Infodis CRM database speelde ook een rol in dit proces. Deze moest namelijk de interactie kunnen ondersteunen.

Omdat dit een totaal nieuw systeem was kon ik mij nergens op baseren. Ik ben daarom begonnen met het doel van elke use case te beschrijven. Dit doel kon ik gebruiken bij het schrijven van de scenario's. Als ik het doel wist hoefde ik alleen nog te bedenken welke stappen er genomen moesten worden om dat doel te bereiken. Deze stappen heb ik beschreven in naam van het systeem en in naam van de medewerker. Hieronder een voorbeeld van een 80% scenario uit de

use case View company information. 80% betekend dat dit zich in het grootste gedeelte van de gevallen op deze manier afspeelt.

1. *De medewerker klikt op het "Company" tabblad*
2. *Het systeem toont het company scherm*
3. *De medewerker bladert door de lijst met company's*
4. *De medewerker selecteert een company*
5. *Het systeem toont desbetreffende company informatie*

Dit heb ik voor alle use cases gedaan. Vervolgens heb ik ook voor een aantal use cases 20% scenario's opgesteld. Dit waren use cases waarbij alternatieve scenario's mogelijk waren. Als voorbeeld het annuleren van het aanmaken van een contactpersoon.

1. *De medewerker klikt op het "Contacts" tabblad*
2. *De medewerker klikt op de knop "New"*
3. *Het systeem toont een "New contact" pop-up*
4. *De medewerker wilt toch geen gegevens invullen en klikt op de "Cancel" knop*
5. *Het systeem sluit de pop-up*

De scenario beschrijvingen van de belangrijkste use cases zijn deze iteratie vrijwel voltooid. Op een aantal kleine toevoegingen of wijzigingen na. Aan de hand van deze scenario's ben ik ook begonnen met het schetsen van interfaces het Proof of Concept. Hierover meer in het volgende hoofdstuk.

Zie ook het Use case document in de bijlagen.

#### **6.4.3 Interfaces ontwerpen**

Omdat ik maar beperkte tijd had voor het ontwerpen van interfaces moest ik een manier vinden waarop dit niet veel tijd zou kosten. Daarom heb ik het onderwerp rapid prototyping bestudeerd. Ik wilde dit gebruiken voor het maken van concept interfaces. Het is een vorm van prototyping waarbij er snel een tastbaar iets wordt opgeleverd waarop feedback gegeven kan worden. Diegene die requirements heeft aangeleverd kan snel zien hoe zijn requirements zijn overgekomen en of ze goed verwerkt zijn in het prototype. Een ander voordeel is dat je ook laat zien dat het snel gemaakt kan worden. Daardoor is de drempel om kritiek te geven voor medewerkers lager. Bij een product waar iemand maanden aan heeft gewerkt zal er minder snel scherpe kritiek worden gegeven. Door die kritiek kan immers maanden aan werk als verspild gezien worden. Als iemand iets in een paar uur maakt zal men eerder geneigd zijn om kritiek te geven.

Ik heb een aantal eisen aan de interfaces gesteld die ik met rapid prototyping zou gaan maken:

- Opgebouwd uit herkenbare elementen
- Kern van het systeem is vormgegeven
- Er zijn echte gegevens ingevoerd
- Detail alleen waar nodig

Vanuit mijn opleiding had ik wel wat basiskennis over het ontwerpen van User Interfaces. Maar op de opleiding wordt geen theorie gegeven over de opbouw en structuur van interfaces. Vragen als: "Wat is bruikbaar, wat verhoogt productiviteit, hoe denk je aan de gebruikersvriendelijkheid, etc..." worden niet behandeld. Ik heb daarom veel gelezen op sites over User Interfaces en User Experience. (Bijvoorbeeld: <http://www.uxnewsfeed.com/>) Dit heeft me geholpen omdat er veel informatie te vinden is over best practices op user interface gebied.

Ook heb ik op [www.guui.com](http://www.guui.com) een template gevonden waarmee in Microsoft Visio snel user interfaces gemaakt kunnen worden. Deze interfaces kunnen opgebouwd worden uit meest

gebruikte elementen van user interfaces. Aan de hand van deze template ben ik begonnen met het ontwerpen van user interfaces.

Ik ben hierbij begonnen met de interface voor de use case Company information. Ik heb hiervoor gekozen omdat deze een hoge prioriteit heeft. Daarnaast kon ik delen van deze interface hergebruiken voor de interfaces van andere use cases.

Hieronder een voorbeeld van een interface ontwerp.

Company information			
* Company	Contacts	Technical	Rates
Services	Relations		
<b>Edit</b>			
<b>General information</b>			
Name	Canon Europe NV	Debiteurennummer	116285
Address	Bovenkerkerweg 59-61	VAT_number	NL005916343B01
Postal_code	1185 XB	Organisational Unit of	
City	Amstelveen	Notes	
Phone	+31 205458545		
Fax	+31 205458222		
E-mail			
Website_url	<a href="http://www.canon-europe.com/">http://www.canon-europe.com/</a>		
<b>Postal address</b>			
Name	Canon Europe NV	City	Amstelveen
Address	Bovenkerkerweg 59-61	Country	NL
Postal_code	1185XB		

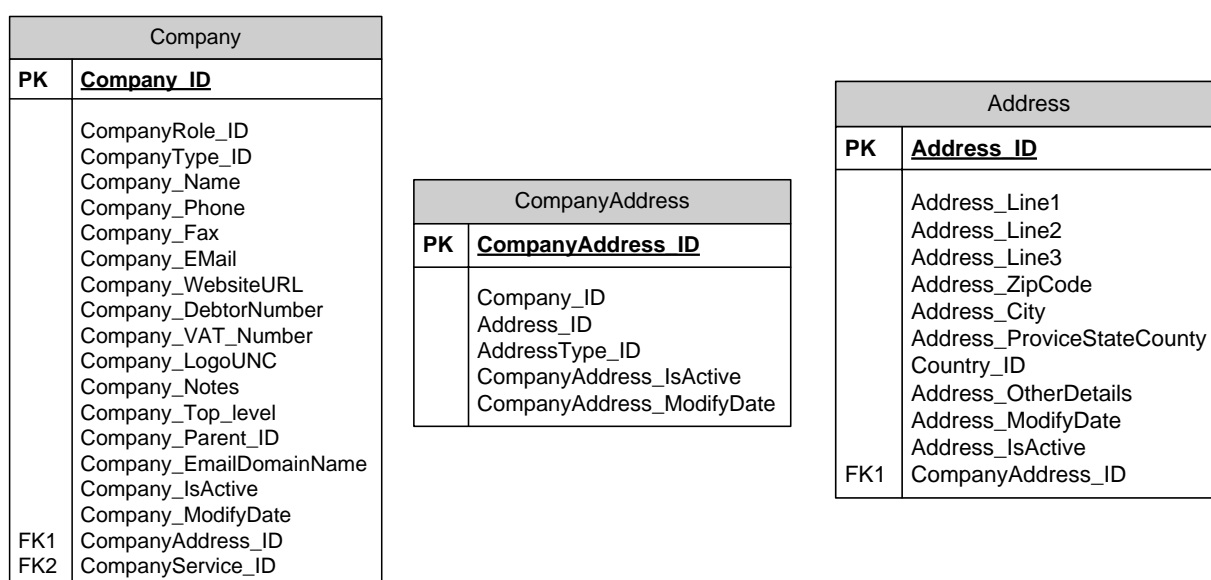
De interfaces moesten aansluiten op de use cases. Daarom heb ik het ontwerpen van de interfaces vaak gecombineerd met het schrijven van de use case scenario's. Daardoor kon ik gemakkelijk bepalen wat gebruikersvriendelijk was of juist wat niet. En op deze manier waren de use cases en interfaces altijd consistent met elkaar.

#### 6.4.4 Database ontwerp

Het ontwerpen van het datamodel was al gestart in de Inception fase. Deze ben ik verder gaan detailleren in de Elaboration fase. De eerste versies van het datamodel uit de Inception waren vooral een opsomming van alle klantgegevens die in het CRM opgeslagen moesten worden. In de Elaboration ben ik meer gaan focussen op de opbouw en structuur van de gegevens.

Om de database te structureren heb ik gebruik gemaakt van de “Sterke entiteiten – zwakke entiteiten” regel. Door middel van deze regel kan bepaald worden welke gegevens in welke entiteiten opgenomen worden. De gevormde sterke entiteiten zullen daarbij de essentie van de database zijn. De entiteiten kunnen vervolgens door middel van zwakke entiteiten aan elkaar gekoppeld worden.

Om te bepalen welke entiteiten sterk zijn moet er een simpele vraag gesteld worden. Kan de entiteit op zichzelf bestaan of is deze afhankelijk van andere entiteiten om te bestaan? Wanneer de desbetreffende entiteit onafhankelijk kan bestaan is het een sterke entiteit. Een sterke entiteit bevat gegevens die waardevol zijn onafhankelijk van andere gegevens. In het geval van het Infodis CRM zijn bijvoorbeeld de entiteiten Company en Address een sterke entiteit.



De Company en Address entiteiten kunnen in principe onafhankelijk bestaan van andere entiteiten. Door middel van een zwakke entiteit als CompanyAddress kan de entiteit Company aan de entiteit Address gerelateerd worden. Op dezelfde manier kunnen er nog andere entiteiten aan Address gerelateerd worden via een zwakke entiteit.

Na het bepalen van alle entiteiten is het datamodel op de volgende pagina ontworpen. De sterke entiteiten in het model zijn:

- Company
- Address
- Contact
- Service
- Party
- Subscription



Er is ook veel rekening gehouden met de naamgeving van tabellen en velden om deze zo consistent mogelijk te maken. Dit is gedaan omdat Willem in mijn initiële ontwerp inconsistente naamgeving constateerde. Vanuit zijn rol als kwaliteitsmanager heeft hij daarom gelijk ingegrepen en mij regels voor de naamgeving gegeven. In tabelnamen begint elk woord met een hoofdletter, zwakke entiteiten bevatten de namen van de tabellen die met elkaar verbonden worden. En alle veldnamen in tabellen beginnen met de naam van de tabel waarin ze voorkomen gevolgd door een liggende streep.

Gedurende de Elaboration fase zijn er een aantal delen van het datamodel afgevallen omdat de prioriteit van de achterliggende use cases laag was. Dit is gedaan om het primaire datamodel zo overzichtelijk mogelijk te houden. De afgevallen onderdelen zijn opgeslagen als aparte datamodellen. Van deze onderdelen was de basis ontworpen maar er moest nog meer detail aan toegevoegd worden. Dit zal nu later gebeuren als de onderdelen met hogere prioriteit afgebouwd zijn. En dat is hoogstwaarschijnlijk pas na de afstudeerperiode. Deze onderdelen zullen dan ook overgedragen worden aan de ontwikkelaars bij Infodis.

### **Company**

Dit is de centrale entiteit van de hele database Infodis CRM database. Alle sterke entiteiten zijn verbonden met deze entiteit. Aan de company zijn onder andere de Contact, Party en Service entiteiten gekoppeld. De zijn doormiddel van primary en foreign keys direct of indirect aan elkaar gekoppeld.

Omdat er veel verschillende company's opgenomen moeten worden in de database krijgt elke company een rol en een type mee. Deze zijn als aparte zwakke entiteiten vastgelegd en definiëren wat voor soort company het betreft.

### **Contact**

De Contact entiteit bevat alle contactpersonen waarmee Infodis communiceert. Deze is gekoppeld aan de Company entiteit. De Contact entiteit bevat gegevens als naam, geslacht, taal van de contactpersoon. Gegevens waarmee contactpersonen daadwerkelijk benaderd kunnen worden zijn apart vastgelegd. Dit omdat deze afhankelijk zijn van het medium waarover contact gelegd moet worden.

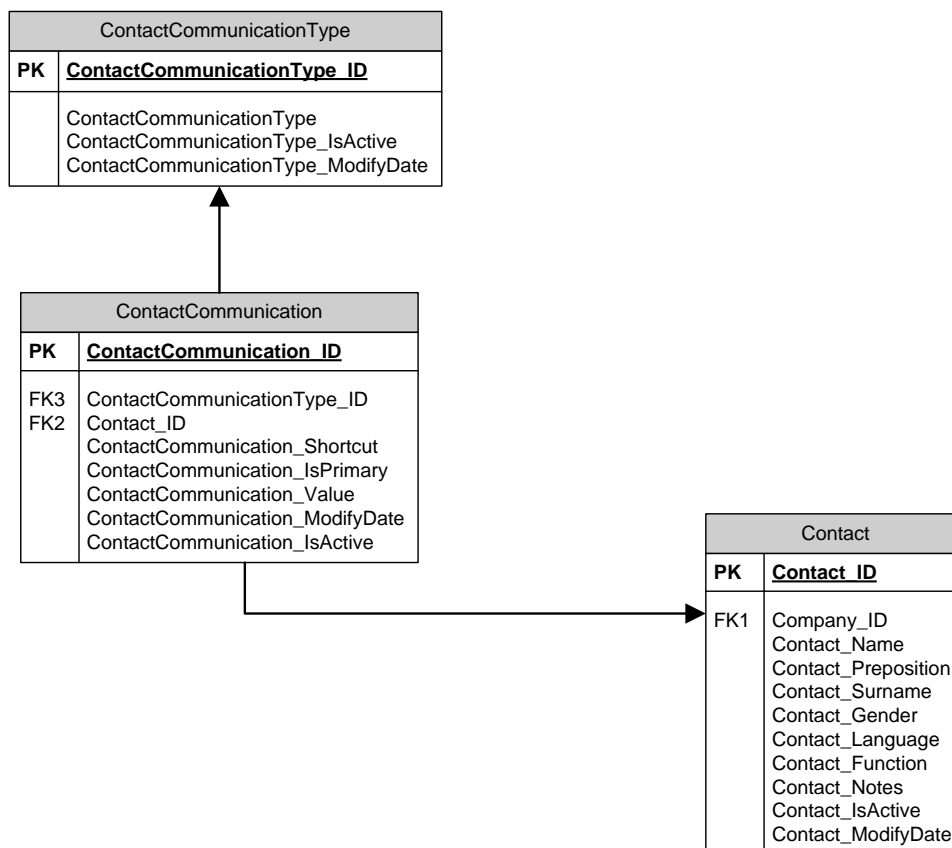
Contactpersonen kunnen op vele manieren benaderd worden. Hier een opsomming van:

- Telefoon vast
- Mobiel
- Mobiel privé (In sommige gevallen)
- Verkort nummer (Voor de Infodis telefooncentrale)
- Fax
- E-mail

Daarbij komt nog dat een contactpersoon meerdere telefoonnummers, email adressen, etc... kan hebben. Dit is nog een reden waarom deze gegevens in een aparte entiteit zijn vastgelegd. Om hierin te kunnen voorzien heb ik nagedacht over hoe deze gegevens ingevoerd zouden gaan worden door de gebruiker. Ik was niet van plan om voor al deze communicatie media een apart veld vast te gaan leggen. Omdat dit vrijwel onvermijdelijk zou lijden tot een grote hoeveelheid NULL waardes. Mijn stagebegeleider gaf me de tip om naar al bestaande applicaties te kijken om te zien hoe dit daar geregeld is. Ik heb daarom gekeken naar het invoeren van contactgegevens op een Google Android en Apple IOS telefoon. Hierbij kan je bij een contactpersoon steeds velden toevoegen wanneer deze nodig zijn. Velden voor telefoonnummers, e-mail adressen, naw gegevens, etc... Deze opbouw heb ik vervolgens gebruikt voor in het database ontwerp voor contactgegevens en ook voor adresgegevens.



Elke contactpersoon kan met deze opzet een in principe onbeperkt aantal e-mail adressen of telefoonnummers hebben. Hieronder is te zien hoe dit onderdeel in de database is verwerkt.



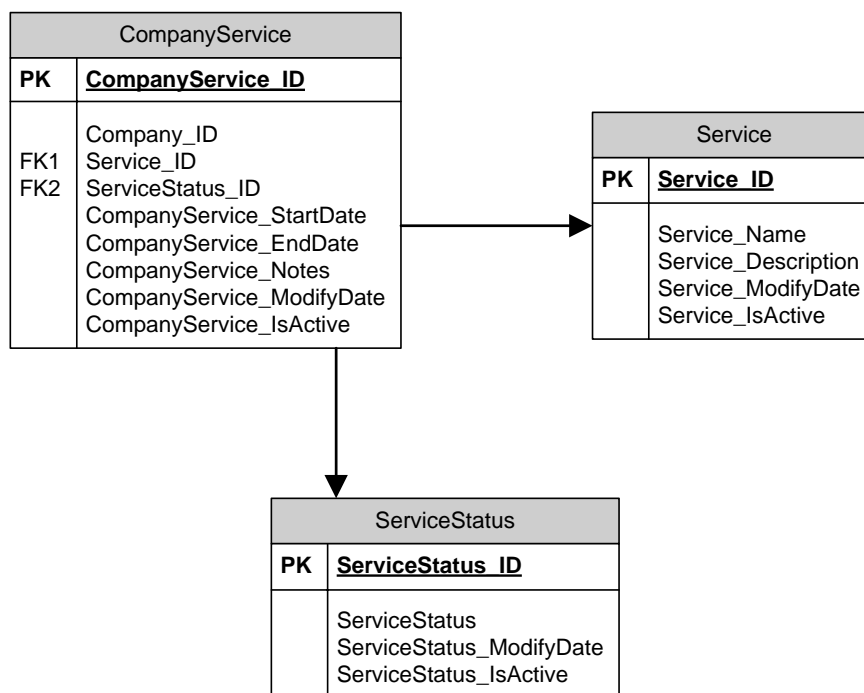
De entiteit Contact is verbonden aan de entiteit ContactCommunication, deze is weer verbonden aan ContactCommunicationType. In deze laatstgenoemde entiteit worden de verschillende soorten communicatie media vastgelegd. In ContactCommunication wordt vervolgens aangegeven wat de waarde is van het desbetreffende type. Bijvoorbeeld het type is "E-mail" en de waarde ervan is "[test@infodis.nl](mailto:test@infodis.nl)".

In de concept database zien de gegevens er dan uit zoals hieronder. Door met SQL joins te leggen kunnen de verschillende ID's vervangen worden door makkelijk leesbare namen.

	ContactCommunication_ID	ContactCommunicationType_ID	Contact_ID	ContactCommunication_Value	ContactCommunication_ModifyDate	ContactCommunication_IsActive
1	777	6	448	'Andreas.Mischke@canon.de'	2012-06-12	1
2	778	2	449	'*31(0)207145180'	2012-06-12	1
3	779	6	449	'Pedro.Da.Costa.Silva@contractors.canon-europe.co...	2012-06-12	1
4	780	2	450	'*31(0)205458647'	2012-06-12	1
5	781	6	450	'Emily.De.Vries@canon-europe.com'	2012-06-12	1
6	782	2	451	'*31 (0) 181 355454 (867)'	2012-06-12	1

## Service

In de Service entiteit worden alle services vastgelegd die Infodis biedt aan klanten. Dit wordt gedaan zodat van elke klant inzichtelijk is welke services deze gebruikt. Infodis heeft dat nu nog nergens centraal vastgelegd. Het heeft drie redenen om dit te gaan vastleggen. Ten eerste omdat het slim is om te weten met welke services je als bedrijf geld verdient. Ten tweede zodat klanten geïnformeerd kunnen worden over storings en/of onderhoud aan services. En ten derde zodat er actief (nieuwe) services aan klanten aangeboden kunnen worden.



Van de services wordt bijgehouden aan welke company deze is gekoppeld, wat de start- en einddatum is en wat de status van de service is.

Hieronder is te zien hoe de gegevens er in de concept database uitzien. Hierbij zijn SQL joins gebruikt om in plaats van ID's makkelijk leesbare namen te tonen.

	Company_Name	Service_Name	ServiceStatus	CompanyService_StartDate	CompanyService_EndDate
1	Sony	Business Intelligence	Implementatie fase	2012-04-10	NULL
2	Sony	Virtual Consolidation	Productie fase	2011-02-01	NULL
3	Philips CL	CO2 Calculation	Productie fase	2010-02-14	NULL
4	Philips CL	Pre-Billing	Productie fase	2010-07-03	NULL
5	Bavaria	Geo Consolidation	Productie fase	2009-07-08	NULL
6	CANON BRATISLAVA	Freight Charge Calculation	Productie fase	2010-05-20	NULL
7	CANON BRATISLAVA	Delivery Windows	Productie fase	2011-08-17	NULL

## 6.5 TEST SOLUTION

Het testen van de oplossing heb ik gedaan door middel van het bouwen van een Proof of Concept (POC). Daarnaast heb ik ook SQL query's gemaakt voor de test database. Ik had bedacht dat ik door het maken van een POC alle requirements en use cases kon testen. Deze zijn immers allemaal nodig bij het bouwen van het POC. Het database ontwerp kon ik testen met de SQL query's. Deze query's zouden de interactie van het POC met de database laten zien.

Het POC bestaat uit schermontwerpen van de kern van het systeem. De belangrijkste en hoogst geprioriteerde use cases. De schermontwerpen zouden vervolgens geanimeerd worden aan de hand van de use case scenario's. Hierdoor is goed te zien hoe de oplossing zou moeten gaan werken. En zo kon ook gelijk het doel van het POC getoond worden. Namelijk bewijzen dat het ontwerp werkt en realistisch is.

The screenshot displays the Infodis CRM web application. The main window is titled 'Infodis database' and contains a 'Contact information' section with tabs for General, Contacts, Technical, Rates, Services, and Relations. A 'New' button is visible. Below this, a 'Contact details' form is shown with fields for Name (David), Surname (Romero), Title (Mr.), Gender (Male), Language (Chinese), Company (Canon), Department (Logistics), and Function (Transport Coordinator). To the right, there are fields for Phone work (+31 (0) 181 355436), Mobile phone private (06 34590305), and E-mail work (david.romero-leal@canon-europe.com). At the bottom right, a modal window titled 'Add phonenummer / E-mail address' is open, showing a 'Categorie' dropdown set to 'Phone\_private', a 'Primary' dropdown set to 'No', and a 'Number' field containing '0658475487'. A blue arrow points to the 'Save' button in this modal. The interface also includes a search bar, a list of company names, and a footer with 'Help', 'TopDesk', and 'SharePoint' buttons.

Het POC heeft ook daadwerkelijk getoond dat het ontwerp van het Infodis CRM goed was. En de test database kon de gewenste gegevens aanleveren door middel van de SQL query's. Een voorbeeld van een POC interface met de bijhorende query is te zien op de volgende pagina.

Zie ook het POC in de bijlagen.

## Interface van de Contact information pagina:

Infodis database

Home Company  
Party Reports

☒ Customer  
☐ Carrier  
☐ Prospect

Search

Company  
Company  
Company  
- Organisational Unit  
- Organisational Unit  
Company  
Company  
Company  
Company  
- Organisational Unit  
Company  
Company

Help

General Contacts Technical Rates Services Relations

**Contact information** New

Name	Phone	Phone mobile	E-mail	Function
Marjanka Haxe	+31 (0)20-5458995	06 24978640	marjanka.haxe@canon-europe.com	Central Transport Manager
Pedro Da Costa Silva	+31(0)207145180		Pedro.Da.Costa.Silva@contractors.canon-europe.com	
David Romero	+31 (0) 181 355436	06 34590305	david.romero-leal@canon-europe.com	Transport Coordinator

Shortcuts

TopDesk SharePoint N: TMS SPS

## De bijhorende SQL query:

Alle contactpersonen voor een bepaalde klant

```
select C.Company_Name as CustomerName
      ,Con.Contact_Name as ContactName
      ,Con.Contact_Surname as ContactsSurname

from Company Com

join Contact Con on Con.Company_ID = Com.Company_ID

where Com.Company_ID = '1008'
```

## Resultaat van de query:

	CustomerName	ContactName	ContactsSurname
1	Sony	Jan-Willem	Boezelman
2	Sony	Anne	Bootsma
3	Sony	Jose	Cabrera Gutierrez
4	Sony	Tomas	Chrastina

## 6.6 ITERATION ASSESSMENT

De iteratie is binnen de vooraf geplande tijd afgerond en er zijn goede resultaten geboekt. Het plannen van deze iteratie is daarom naar mijn idee beter gegaan dan in de Inception. Dit keer geen verborgen verrassingen waar ik niet op had gelet.

De grootste mijlpaal is het afronden van het database ontwerp. Het heeft veel tijd gekost om alle informatie te inventariseren en om een logische structuur te bedenken. Maar deze tijd is nuttig besteed want er is nu een ontwerp wat zich al bewezen heeft in een test omgeving. Dit belooft veel goeds voor de volgende iteratie waarin de database volledig gebouwd moet gaan worden.

Ook is er een risico analyse gemaakt van de use cases. Aan de hand hiervan heb ik samen met de opdrachtgever bepaald welke onderdelen de hoogste prioriteit hebben. Dit waren tevens de onderdelen waaraan ik als eerst ben gaan werken. Van deze use cases zijn scenario's beschreven die consistent zijn met de requirements en interfaces. Het consistent maken was veel werk omdat ik constant heen en weer moest switchen tussen verschillende documenten. Omdat ik de requirements had gegroepeerd scheelde het consistent maken met de requirements wel veel tijd.

Requirements zijn volop aangevuld, aangepast en ook voor het grootste gedeelte gevalideerd. Dit was nodig omdat ik het POC erop moest baseren. Het POC is gemaakt zodat het consistent is aan de use case scenario's, requirements en het database ontwerp. Aan de hand van het POC is getest of het ontworpen systeem kon voldoen aan de eisen en om aan te tonen dat het ontwerp werkt. Deze tests zijn allemaal voorspoedig verlopen en geven daarmee ook groen licht om de volgende fase te starten.

Het is goed dat ik heb samengewerkt met Willem omdat hij veel tips had rondom het ontwerpen van de database. Hij heeft vaak een frisse blik op problemen waar ik tegenaan loop en helpt me dan snel weer op het juiste pad. Ook heeft hij me veel uitleg gegeven over de testomgeving en de werking van SQL Management Studio. Dat heeft mij veel tijd bespaart die ik hard nodig had deze iteratie. Ik heb deze iteratie wel een aantal dagen nodig gehad om mijn SQL kennis op te frissen. Dat is ook wel nodig voor de volgende iteratie waarin de hele database op een SQL Server gebouwd moet gaan worden.

### 6.6.1 Besluit volgende iteratie

In overleg met mijn opdrachtgever en Willem Kosmeijer hebben we besloten wat de volgende stap zou worden. We hebben besloten dat een passend vervolg het volledig bouwen van de Infodis CRM database is. Dit zal voorlopig nog op de test server van Infodis plaatsvinden. Na het bouwen moeten we ons focussen op het vullen van de database vanuit bestaande informatiebronnen. Daar gaat waarschijnlijk de meeste tijd in zitten.

Ook is besloten dat er voorlopig minder focus op lager geprioriteerde use cases ligt. Deze moeten in opvolgende fases volledig gedetailleerd en ontwikkeld worden. De reden voor de lagere prioriteit is dat mijn begeleider en tevens opdrachtgever liever heeft dat de kern van het systeem helemaal af is. De lagere geprioriteerde onderdelen van het systeem zullen dan in een vervolgtraject na de afstudeerperiode afgebouwd worden.

## 7 FASE 2: ELABORATION 2

De focus van deze iteratie ligt op het bouwen en vullen van de Infodis CRM database. Het database ontwerp zal volledig worden gebouwd op een test server. Hierin zullen alle functionaliteiten verwerkt worden die ook in de productie database komen.

Ook zal de database gevuld gaan worden aan de hand van gegevens die nu in bestaande database vastliggen. Het is gedurende deze iteratie de bedoeling dat de volgende onderdelen van de database gevuld zullen gaan worden:

- Company's
  - -Namen
  - -Adressen
- Party's
  - -Carriers
  - -Bookers
- Contacts
  - -Namen
  - -Contactgegevens

De bouw zal waarschijnlijk minder tijd kosten dan het vullen van de database. Dit aangezien het ontwerp simpelweg overgenomen wordt in SQL Server. Het vullen zal meer uitzoekwerk kosten en dit zal ook in stappen moeten gebeuren.

Uiteindelijk zal er ook een concept gemaakt worden van een front-end applicatie.

### 7.1 PLAN ITERATION

High level objectives zijn doelen die bereikt moeten worden tijdens de iteratie. Hieronder de doelen van deze iteratie:

- Overige use cases verder detailleren
  - Als er tijd voor is dit oppakken
- Requirements bijhouden
  - Deze iteratie ga ik ervanuit dat er niet veel nieuwe requirements bij zullen komen. Toch blijft er aandacht bij dit onderwerp omdat er wellicht aanvullende requirements bij zullen komen tijdens de bouw van de database.
- Database bouwen
  - Dit heeft deze iteratie een hoge prioriteit. De database is de fundering van het hele Infodis CRM en met het bouwen hiervan wordt een mijlpaal in het project bereikt.
- Database vullen
  - Het vullen van de database heeft ook een hoge prioriteit deze iteratie. De focus zal liggen op het zoveel mogelijk automatiseren van het vullen.
- Front-end concept
  - Tenslotte zal er ook gewerkt worden aan een front-end applicatie van het Infodis CRM. Deze zal de kernfunctionaliteiten bevatten zodat het systeem gepresenteerd zal kunnen worden.

De work items zijn de werkzaamheden die verricht moeten worden om de doelen te bereiken. Milestones zijn de producten die het resultaat zijn van de werkzaamheden. Dat waren in deze iteratie:

- Use cases
- Requirements
- Test cases
- Vision document
- Architecture Notebook
- Database bouw
- Database vullen
- Front-end concept

## 7.2 IDENTIFY AND REFINE REQUIREMENTS

### 7.2.1 Wensen en eisen verzamelen

In deze iteratie werden er ook wensen en eisen verzameld. Er lag wel minder focus op deze werkzaamheden omdat er veel tijd nodig was voor het bouwen van de database. De wensen en eisen die er toch bijkwamen waren ook bijna allemaal van toepassing op de database. Veelal waren die toevoegingen op al bestaande requirements.

Een wens was dat ingevoerde data zoveel mogelijk gevalideerd zou worden door de Infodis CRM applicatie. Bijvoorbeeld door het toepassen van invoervalidatie op velden in de applicatie. Hiermee kon voorkomen worden dat er vervuiling van de database zou optreden. Iets wat in de huidige situatie vaak het geval was en veel irritatie veroorzaakte. Tijdens het vullen van de database zou hier ook gelijk op gelet worden door bijvoorbeeld geen NULL waardes over te nemen.

Ook waren er veel eisen omtrent de opslag van de data en de betrouwbaarheid daarvan. Omdat het vullen van de database zoveel werk zou worden mochten gegevens niet zomaar verwijderd worden uit de database. Aangezien het veel werk zou kosten als er per ongeluk een groot deel van de data verwijderd wordt.

Omdat geïmporteerde data een probleem zou kunnen veroorzaken met de al bestaande data in de database moet er iets bedacht worden om het importeren terug te draaien. Tenslotte is het een vereiste dat er regelmatig back-ups gemaakt worden van de data.

### 7.2.2 Requirements valideren

De requirements zijn deze iteratie vrijwel allemaal gevalideerd. Sommige requirements zijn afgevallen en een paar requirements staan nog ter discussie. Zie een voorbeeld van deze requirements hieronder:

<b>9</b>	Het systeem onthoudt ingevoerde informatie ook bij storingen/onderbrekingen
<b>9,1</b>	Het systeem houdt een log bij met acties die zijn uitgevoerd
<b>9,2</b>	Het systeem houdt een log bij met acties die mislukt zijn

Deze requirements stellen hoge eisen aan het systeem die misschien overbodig zijn. Het zijn eigenlijk eisen die van toepassing zouden zijn op real time systemen met veel in- en output. De requirements zijn nog niet volledig van de tafel maar er wordt nog nagedacht over hoe zo iets makkelijk te implementeren zou zijn.

## 7.3 DEVELOP THE ARCHITECTURE

In de tweede Elaboration fase zijn er een aantal toevoegingen geweest aan het Architecture Notebook. Deze hadden allemaal te maken met de bouw of het vullen van de Infodis CRM database.

### Data validatie

Om zoveel mogelijk af te vangen dat er onjuiste informatie in de database wordt ingevoerd moet er gebruik gemaakt worden van data validatie regels. Dit zijn regels op invoervelden in de database. Deze zorgen ervoor dat de invoermogelijkheden per veld beperkt worden tot waar het veld voor bedoeld is. Hiermee kan de kwaliteit van de gegevens zo hoog mogelijk gehouden worden. Een aanvulling hierop zal zijn het instrueren van medewerkers om gegevens zo correct mogelijk in te voeren. De kwaliteitsmanager bij Infodis zal hierop gaan toezien.

Een voorbeeld van data validatie is veelvoorkomend op websites. Als men een e-mail adres moet invullen is het verplicht dat deze eindigt met @emailadres.nl. Als dit niet zo is geeft de CRM applicatie een melding en wordt er nog niks opgeslagen in de database. Hiermee wordt voorkomen dat gegevens opgeslagen worden die van geen waarde zijn.

Bij het zoeken naar data validatie methodes viel het mij op dat er bijna geen standaarden zijn op dit gebied. En al helemaal geen internationale standaarden. Dit komt omdat er grote verschillende zijn tussen bijvoorbeeld telefoonnummers in Nederland en Amerika. Ditzelfde geldt voor de notatie van adressen en postcodes. Hierdoor is het niet mogelijk om een universele validatie te maken binnen dit project. Dit zou een project op zichzelf zijn. De data validatie methoden die voor het Infodis CRM gebruikt worden controleren nu of velden gebruikt worden waarvoor ze bedoeld zijn.

### Vullen van de database door middel van transacties

De Infodis CRM database moet gevuld worden vanuit diverse andere databases. Dit proces zal vele malen getest worden en zal wellicht af en toe herhaald moeten worden wanneer de database operationeel is. Als de database operationeel is mogen er onder geen voorwaarde verkeerde gegevens worden geïmporteerd. Ook mogen de geïmporteerde gegevens niet zorgen voor problemen met de gegevens die al in de database staan. Samen met Willem kwam ik er al snel achter dat deze maatregelen noodzakelijk zijn. Door een fout werd de test database gevuld met gegevens uit de verkeerde brontabel. Hierdoor stond de database vol met gegevens die eigenlijk niet geïmporteerd hadden mogen worden.

Om deze zaken te voorkomen is besloten om de database te vullen op basis van SQL transactions. Gegevens die geïmporteerd moeten worden vallen binnen een transactie. Elke transactie krijgt een uniek nummer wanneer deze wordt uitgevoerd. Op basis van dit nummer kan elke transactie gevolgd worden. Als er problemen zijn ontstaan door de gegevens in de transactie kan men de transactie terug draaien. Hierdoor wordt de database weer hersteld tot het punt voordat de desbetreffende transactie gestart werd. Deze maatregel voorkomt problemen door het verkeerd importeren van gegevens. En hierdoor kan er ook efficiënter gezocht worden naar problemen mochten deze zich voordoen.



## Records archiveren

Records in de database zullen niet direct verwijderd worden. In plaats daarvan zullen de records gearchiveerd worden. Deze kunnen dan na verloop van tijd permanent verwijderd worden. Bijvoorbeeld als ze al jaren geleden gearchiveerd zijn. Dit wordt geregeld door twee velden in de database die aan elke tabel worden toegevoegd. Dit zijn de velden:

- Tabelnaam\_IsActive
- Tabelnaam\_ModifyDate

Hiervan zal de IsActive kolom bij het aanmaken van een record altijd standaard de waarde 1 (True) krijgen. Bij het "verwijderen" van de record zal de waarde veranderd worden in 0 (False). In de applicatie zal gefilterd worden zodat alleen records getoond worden als de waarde 1 is. Hierdoor worden records nooit gelijk permanent verwijderd. Dit heeft als voordeel dat je blijft beschikken over oude gegevens. Ook kunnen per ongeluk verwijderde records terug gezet worden zonder hiervoor gebruik te moeten maken van een back-up.

De kolom ModifyDate krijgt standaard de waarde van de huidige datum (Getdate()). Hiermee kan in eerste instantie gezien worden wanneer een record is aangemaakt. Als het record op een latere datum wordt aangepast krijgt deze de datum van deze aanpassing mee. Op basis van deze datum kan uiteindelijk ook besloten worden om de records permanent te verwijderen.

Contact_Name	Contact_Surname	Contact_IsActive	Contact_ModifyDate
Remco	Eichelsheim	1	2012-06-12
Adela	Elekes	1	2012-06-12
Jean-Sebastien	Epaud	1	2012-06-12
Panagiotis - Panos	Giannoulis	1	2012-06-12
Marjanka	Haxe	1	2012-06-12

De IsActive en ModifyDate kolommen in de database

## Contactpersonen exporteren uit Microsoft Exchange Server

Om het Contact gedeelte van de Infodis CRM database zo compleet mogelijk te krijgen moeten contactgegevens van diverse plekken geïmporteerd worden. Een van deze plekken is de Exchange Server van Infodis.

Omdat er vaak mail contact is tussen Infodis en klanten zijn in de adresboeken van Exchange veel contactgegevens terug te vinden. Deze contactgegevens kunnen uit de Exchange server gehaald worden en vervolgens in de Infodis CRM database geplaatst worden. De contactgegevens kunnen automatisch aan een company gekoppeld worden. Hiermee is rekening gehouden bij het ontwerp van de database. Van elke company wordt het e-mail domeinnaam opgeslagen. Op deze manier kunnen contactgegevens op basis van e-mail adres gekoppeld worden aan een company.

Zie ook het Architecture Notebook in de bijlagen.

## 7.4 DEVELOP SOLUTION

### 7.4.1 Bouwen database

De database is gebouwd aan de hand van het database ontwerp dat is voltooid in de eerste Elaboration fase. Dit ontwerp is nog aangepast om het vullen van de database te ondersteunen. Hierbij zijn een aantal tijdelijk tabellen toegevoegd die later weer worden verwijderd.

Het bouwen is gedaan met behulp van Microsoft SQL Server Management Studio. Dit is een pakket voor het configureren en managen van alle onderdelen van de SQL server. Hiermee is het mogelijk om zowel via een grafische omgeving als via scripts een database te implementeren. Ik heb vooral gebruik gemaakt van de grafische ontwikkelomgeving omdat het voor mij lang geleden was dat ik met SQL werkte. Naarmate de bouw vorderde werd het voor mij steeds makkelijker om ook via scripts zaken te implementeren.

Bij de bouw kwam meer kijken dan ik dacht. Het waren vooral kleine dingen die toch veel tijd in beslag namen. Hieronder een overzicht van werkzaamheden die ik heb uitgevoerd bij het bouwen van de database:

- Tabellen toevoegen
- Kolommen toevoegen
- Veldtypes en lengte configureren
- Auto increment keys toewijzen
- Default values configureren
  - IsActive
  - ModifyDate
- Naamgeving consistent houden
- Bepalen welke velden NULL values mochten hebben
- Self generated database diagram configureren
- Relations configureren
- Triggers configureren
- Test data invoeren
- Query's testen

Na de bouw en het kort testen van de database moest deze gevuld gaan worden vanuit bestaande bronnen. Deze bronnen zijn de Topdesk en SPS databases. De data moest op een gestructureerde manier uit die databases gehaald worden. Dit is gedaan doormiddel van ETL schema's en conversie scripts. Deze scripts heeft Willem Kosmeijer gemaakt aan de hand van de gemaakte ETL schema's. Tevens zijn er een aantal tools gemaakt waarmee de data eerst bekeken en gefilterd kon worden voordat deze werd geïmporteerd.

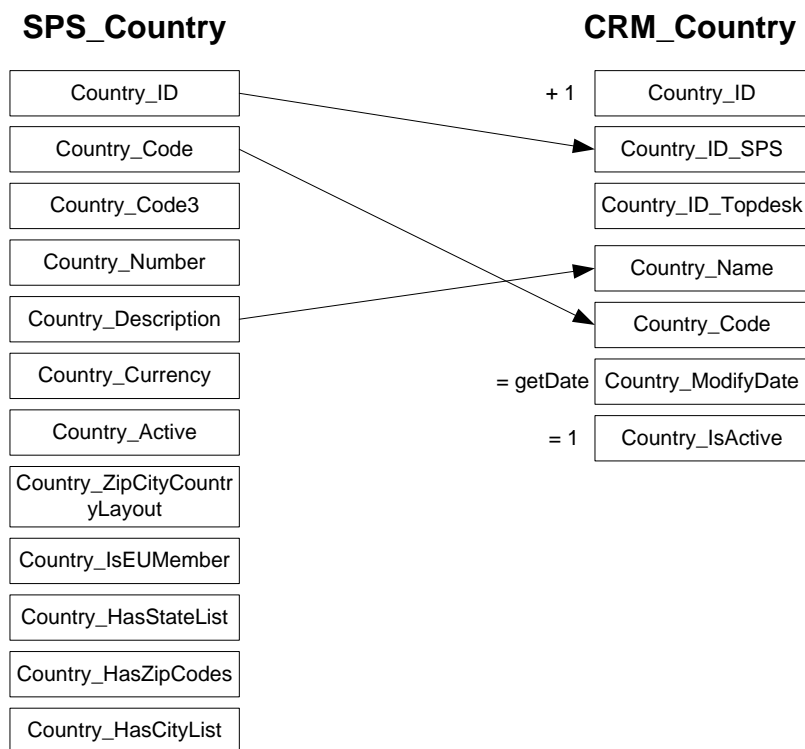
### 7.4.2 ETL schema's opstellen

ETL is een techniek die gebruikt wordt om data uit verschillende bronnen op te halen en deze te importeren in een andere bron. Bronnen zijn hierbij meestal databases. In de meeste gevallen wordt ETL gebruikt bij het vullen van datawarehouses en andere grote databases.

Ik heb besloten dat met deze techniek het vullen van de CRM database goed werd gestructureerd en gedocumenteerd. De schema's die ik heb gemaakt zijn afgeleid van de theorie uit de reader Datawarehouse ontwerp (2009) van Jeroen Vuurens. Hier waren wel wat aanpassingen nodig omdat in de theorie alleen schema's besproken worden die van toepassing zijn op datawarehouses. Relationele databases waar in dit project mee gewerkt word hebben een totaal andere opzet als een datawarehouse. Het grootste verschil met de theorie is dat mijn ETL schema's meerdere doeltabellen bevatten.

Op basis van ETL schema's konden developers scripts schrijven om data uit de ene database naar de andere te exporteren. De ETL schema's bevatten hiervoor alle benodigde informatie. Hieronder een voorbeeld van een ETL schema voor het importeren van Country gegevens. Op basis van dit ETL schema is een script gemaakt. Dit script is te vinden in de bijlagen.

Op basis van **SPS Country's** creëren we **CRM Country's**



De overige ETL schema's zijn terug te vinden in het Infodis CRM database document. Zie bijlage.

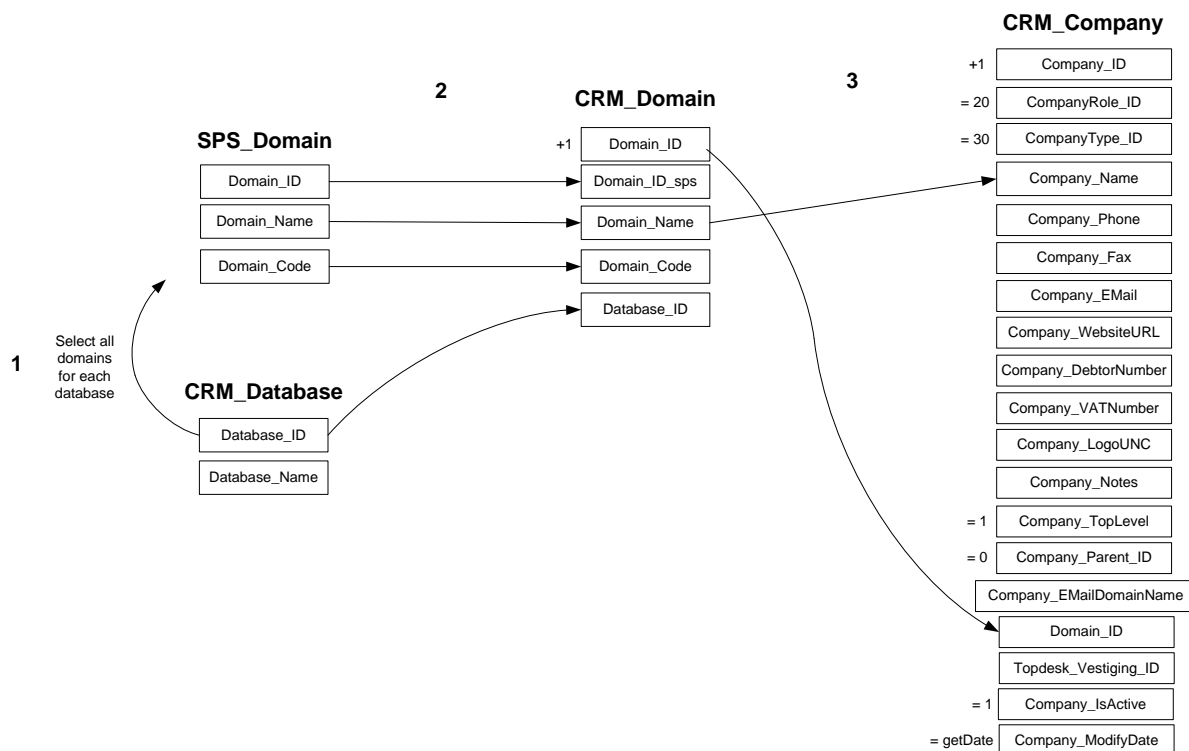
#### 7.4.3 Vullen Infodis CRM database

Het vullen van de database moest gedaan worden vanuit de bestaande SPS en Topdesk database. De data hierin was echter niet zomaar geschikt om in de Infodis CRM database geïmporteerd te worden. Er was namelijk sprake van vervuiling in de brondatabases. Deze vervuiling uit de brondatabases wordt weg gefilterd tijdens het importeren van de gegevens. In de code zijn regels gedefinieerd waaraan de brongegevens moeten voldoen. Als de gegevens niet voldoen worden ze niet geïmporteerd. De volgende regels zijn van toepassing:

- NULL waarden in essentiële velden worden niet meegenomen.
- Verplichte kolommen gevuld met loze informatie worden niet meegenomen.
  - Hierop wordt vooraf gescreend.
- Verouderde informatie wordt niet meegenomen.
- Informatie die verwijderd had moeten worden wordt niet meegenomen.
- Duplicaten worden niet meegenomen.

Nadat de gegevens deze regels gepasseerd zijn worden ze gebruikt in door de conversie scripts en tools.

Er waren ook grote verschillen in de opzet van de verschillende databases. Om de verschillen in opzet te overkomen zijn er diverse aanpassingen gemaakt aan de Infodis CRM database. Zo zijn er keys uit de SPS en Topdesk databases overgenomen om het importeren van gegevens te vereenvoudigen. Het beste voorbeeld hiervan is het proces waarmee customer en carrier company's gecreëerd worden. In dit verslag zal ik alleen van dat proces een beschrijving maken omdat de andere conversies soortgelijk zijn.



ETL schema SPS\_Domain naar CRM\_Domain naar CRM\_Company

## Eerste stap

Elke klant bij Infodis heeft zijn eigen domain in SPS. Daarom zijn op basis van domains alle customer company's aangemaakt. In het ETL schema hierboven is te zien dat dit niet direct is gebeurd. De SPS domains worden ook opgeslagen als CRM domains. Reden hiervoor is dat ik deze SPS domains heb gebruikt om verschillende gegevens aan elkaar te knopen. Elk SPS domain krijgt in het Infodis CRM zijn eigen CRM Domain\_ID. Elke customer company krijgt ook een Domain\_ID.

## Tweede stap

In de SPS database komt het vaak voor dat een carrier veel verschillende party's heeft met allemaal een net iets andere naam. Van deze party's kunnen niet zomaar company's gemaakt worden. Dan zouden er bijvoorbeeld vijf company's van DHL aangemaakt worden met een net iets verschillende naam. Deze dubbele moesten er eerst uitgefilterd worden. Hiervoor heeft Willem Kosmeijer een tool gemaakt. Met deze tool konden party's die veel op elkaar leken samengevoegd worden tot een enkele company. Dit was nog wel veel handwerk omdat er per party bepaald moest worden of deze samengevoegd moest worden. Maar het was ook noodzakelijk omdat de database anders vervuild zou worden met nutteloze data.

Op de volgende pagina is een screenshot te zien van de tool. In het voorbeeld zijn de company's al geïmporteerd in de database en wordt het resultaat getoond van het samenvoegen. Links zijn de party's te zien uit de brontabel in SPS. Rechts zijn de company's te zien die in de CRM database zijn aangemaakt.

	Party	Party_Cc	Party_Name	Company_ID	Company_Name	Party_Address	Party_ZipCc	Country	Party_City
21	AAC01		A & A Cronins	2358	A & A Cronins	66 Baldoyle Industri	13	65	Dublin
22	ABE01		Abendland	2359	Abendland	..		45	REMSECK
23	APL02		APL	2360	APL			97	
24	BSP01		Bosman System Plus	2361	Bosman System Plus			97	
25	BOX01		Boxlines	2362	Boxlines	Schiedamsedijk 54	3011EE	97	Rotterdam
26	CEVA4		Ceva Logistics	2363	Ceva Logistics				
27	CEVA2		Ceva Logistics Limited	2364	Ceva Logistics	Excelsior Road	LE65 1NU	56	Leicestershire
28	CRE01		Cremer	2365	Cremer			97	
29	DAC02		Dachser	2366	Dachser			97	
30	DAC04		Dachser2	2366	Dachser			97	
31	DFDS4		DFDS Container line bv	2367	DFDS Container line bv	Oever 5	3161 GR	97	Rhoon

### Derde stap

Bij het importeren van de carrier company's komt nog meer kijken. Elke carrier in SPS is verbonden met een customer door middel van een domain. De carriers zijn namelijk party's die vallen in een customer domain. Voorbeeld:

- Domain: Canon (Customer)
  - Party: DHL (Carrier)
  - Party: P&O Ferrymasters (Carrier)
  - Etc...

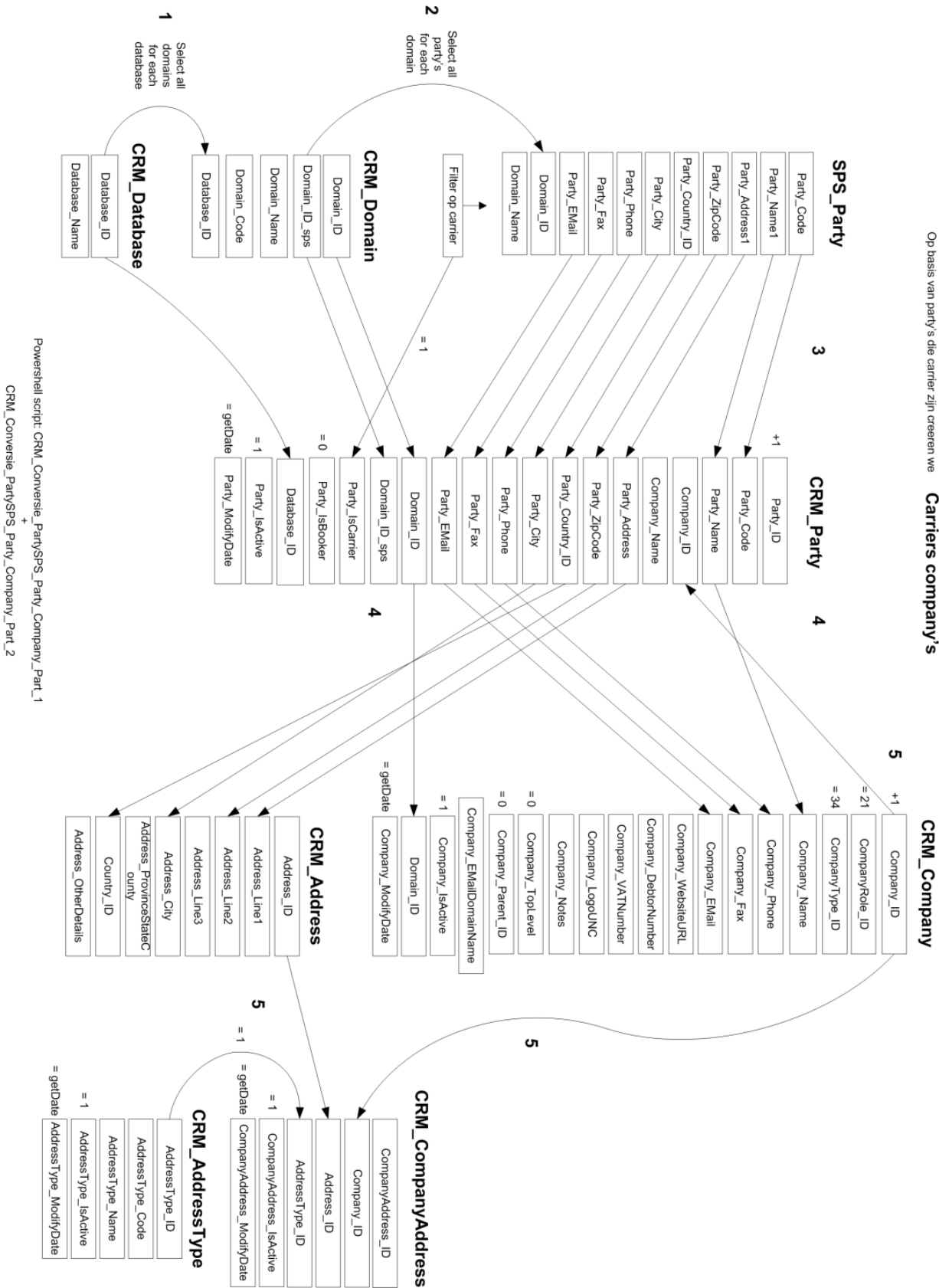
Op basis van de party's moeten company's worden aangemaakt met een bijhorend Domain\_ID. Aan de hand van het Domain\_ID kunnen carrier company's vervolgens gekoppeld worden aan een customer company. We weten namelijk van beide het Domain\_ID en wanneer deze voor beide company's identiek zijn betekent dit dat ze bij elkaar horen.

De party's kunnen echter in meerdere domains voorkomen omdat deze niet voor een één klant carrier zijn. Zo komt DHL zowel voor als carrier in het Sony domain als in het Philips domain. En bij beide kunnen ook weer verschillende contactpersonen horen. In dat geval is het belangrijk dat er per domain een DHL company wordt aangemaakt. Echter zijn dubbele company's niet gewenst. Om dit probleem op te lossen wordt van elke carrier party een Organisational unit (OU) gemaakt. Dit is in principe hetzelfde als een company alleen wordt in de database meegegeven dat het een OU van... betreft. De naam van de OU is samengesteld uit de Party\_Name en de Domain\_Name. Van party's die in meerdere domains te vinden zijn worden OU's gemaakt met een bijhorende Top Level Company. Zie hieronder een voorbeeld van Maersk, een bedrijf wat carrier is voor drie klanten:

- Maersk (Top Level Company)
  - Maersk (Canon Bratislava)
  - Maersk (Philips)
  - Maersk (Bavaria)

Op deze manier kunnen contactpersonen op Top Level niveau aan een company gekoppeld worden als het algemene contactpersonen zijn. Maar ook op OU niveau als de contactpersonen specifiek voor een bepaalde klant van Infodis zijn. Zie op de volgende pagina een ETL schema waarin is te zien hoe van carrier party's in SPS via een tussenstap CRM carrier company's worden gemaakt.

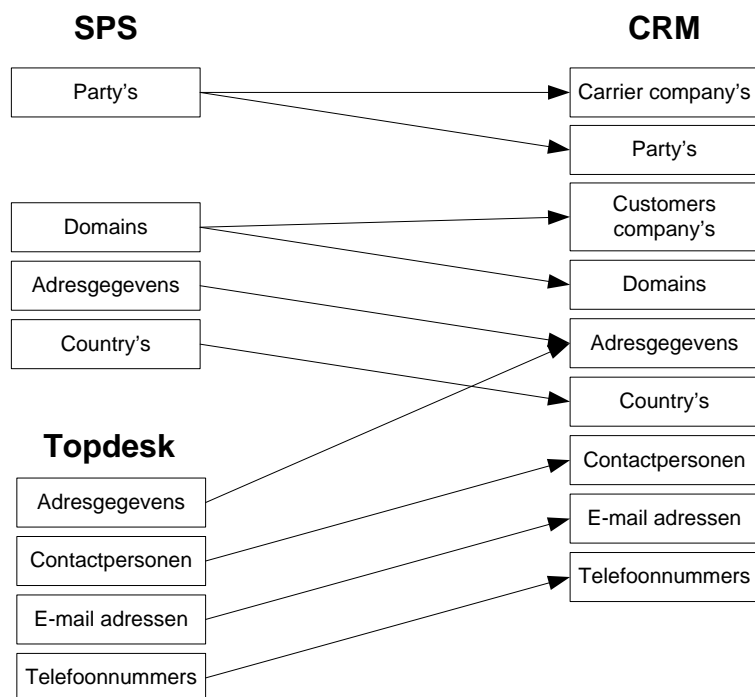
Op basis van party's die carrier zijn creëren we **Carriers company's**



#### 7.4.4 Overzicht geïmporteerde gegevens

Uiteindelijk zijn de gegevens uit SPS en Topdesk in het CRM samengevoegd. Daarmee is een basis gelegd voor het compleet vullen van de CRM database. Er is echter nog veel incompleet en dit komt simpelweg omdat de brondatabases incompleet zijn. Een volgende stap is nog om contactpersonen uit Microsoft Exchange te importeren. Dit zal echter pas na de afstudeerperiode plaats vinden omdat hier geen tijd meer voor was. Dat is ook de laatste database waaruit gegevens gehaald kunnen worden. Vanaf dat punt wordt het vullen van de database handwerk wat door medewerkers zelf bijgehouden moeten worden.

Het volgende schema laat zien welke gegevens tot nu toe uit de brondatabases zijn gehaald en hoe deze zijn gebruikt in het CRM:



#### 7.4.5 Documenteren database

De database is een belangrijk en ingewikkeld onderdeel van het Infodis CRM. Daarom is ook veel tijd besteed aan het documenteren van de database. In de documentatie is uitgelegd hoe de database is ontstaan. Waarom het deze structuur heeft en wat de gedachte achter verschillende mechanismen is.

Uiteraard is er een volledig database model opgenomen. Ook is van alle tabellen en kolommen in de database vastgelegd wat hun inhoud en/of betekenis is. Daardoor kan nooit onduidelijkheid ontstaan over het doel van een bepaalde tabel of kolom.

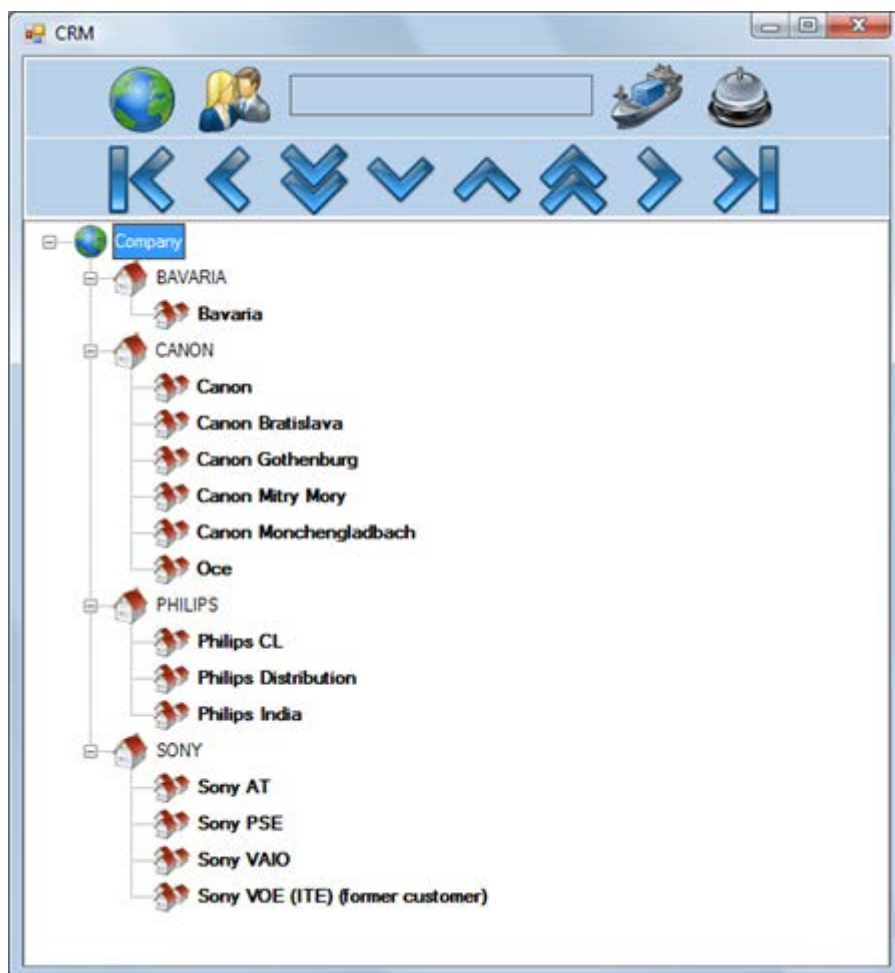
Alle databases waaruit brongegevens zijn overgenomen zijn vermeld. Daarmee wordt getoond waar de originele gegevens vandaan komen. Het proces van het vullen van de database is beschreven en geïllustreerd met behulp van ETL schema's. De ETL schema's beschrijven stap voor stap hoe de CRM database gevuld kan worden.

Tenslotte is er vermeld waar alle documentatie, scripts, tools en ETL schema's terug te vinden zijn.

#### 7.4.6 Front end concept bouwen

Nadat we bewezen hadden dat het vullen van de database goed werkte in alle gevallen konden we verder werken aan een front-end applicatie. Deze is geschreven in C# met behulp van Visual Studio. Willem Kosmeijer heeft hier verreweg het meeste aan gewerkt. Ik heb vaak meegekeken tijdens het programmeren en geprobeerd te helpen als er bugs naar boven kwamen. Het was voor mij 6 jaar geleden dat ik voor het laatst had geprogrammeerd en daardoor was mijn hulp beperkt. Willem heeft ondanks dat heel veel uitleg gegeven over de stukken code en ook laten zien waar programmeurs vaak tegenaan lopen. Dit heeft mij ook doen beseffen dat programmeren een heel creatief proces is waarin je constant naar oplossingen moet zoeken. Daardoor kost het ook veel tijd.

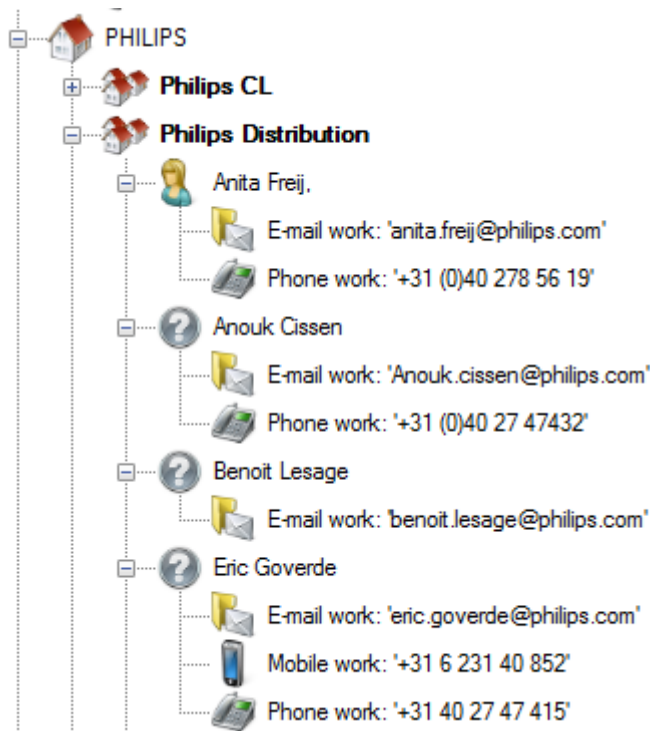
Hieronder is een screenshot te zien van de front-end applicatie van het Infodis CRM. Deze applicatie bevat alleen nog maar basis functies zoals het tonen en doorzoeken van klantgegevens. De gegevens worden getoond in een alfabetisch geordende boomdiagram. Met behulp van de bovenste rij knoppen kan tussen company's, contacts, party's en services gewisseld worden. De hiervoor gekozen symbolen zijn nog tijdelijk, hiervoor moeten duidelijkere symbolen gevonden worden. Met de onderste rij knoppen kan door de lijst genavigeerd worden. Hiermee kan de lijst volledig uitgeklaapt worden, er kan genavigeerd worden tussen de records, etc. Er wordt nog getwijfeld of deze knoppen blijven bestaan in toekomstige versies aangezien ze niet erg instinctief werken.



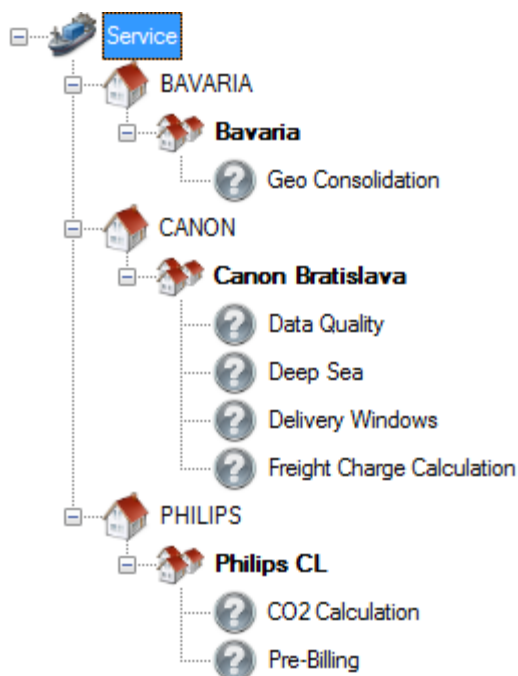
Er is te zien hoe company's getoond worden, een Top Level Company met daaronder de verschillende Organisational Units.



Company's kunnen vervolgens ook uitgekapt worden waardoor de contactpersonen getoond worden. Daarbij kunnen vervolgens ook de contactgegevens getoond worden. Zoals hieronder is te zien:



Via de service knop kan bekeken worden wat voor services (diensten) klanten bij Infodis afnemen.



De front-end applicatie zal na de afstudeerperiode nog veel werk vereisen. Functies als het toevoegen en bewerken van klantgegevens zijn nu nog niet gebouwd. Deze zijn wel essentieel als de applicatie in gebruik genomen gaat worden. Willem zal hiermee verder gaan als het zover is.

## 7.5 TEST SOLUTION

De gebouwde database is vervolgens ook getest. Ik heb het testen gedaan aan de hand van requirements en use case scenario regels. Er is per onderdeel van de database een test case gemaakt.

De test cases zijn gemaakt aan de hand van een template van de OpenUP website. Deze template is op basis van best practices uit de praktijk tot stand gekomen. In de template staan ook richtlijnen beschreven waaraan een test case moet voldoen. Deze richtlijnen heb ik als checklist gebruikt om mijn test cases te "testen". Zie hieronder de richtlijnen voor een test case:

### General

- Does the Test Case identify the requirement it evaluates? This linking might be informal through a naming convention or formalized through a requirements traceability matrix.
- Does the Test Case reference the preconditions and post conditions that apply to it?

### Name

- Is the Test Case name unique?
- Does the name express a test condition or an expected result?
- Is it unambiguous to a stakeholder?

### Brief Description

- Is the logical test condition clearly identified in the description?
- Does the description clearly state the expected result?
- Is the expected result stated as a concrete outcome?
- Can a casual reader distinguish this Test Case from a similar one?

Per test case zijn alle requirements en use case scenario regels opgenomen die specifiek van toepassing zijn op de desbetreffende test case. Dit heb ik gedaan om de test cases overzichtelijk te houden. Als ik per test case alle requirements op moest nemen zou dit resulteren in test cases van vele pagina's lang.

Als de Infodis CRM front-end applicatie is ontwikkeld kan een nieuwe ronde tests uitwijzen of de applicatie ook voldoet aan alle requirements en use case scenario's. Hiervan is nu alleen nog een front-end concept ontwikkeld als stand-alone applicatie. Dit front-end concept is nu alleen geschikt voor demonstratie doeleinden.

Zie het test document in de bijlagen.

## 7.6 ITERATION ASSESSMENT

De tweede Elaboration fase is langer geworden dan gepland. Dit kwam omdat het vullen van de database en het maken van een concept front-end applicatie langer duurde dan ik had verwacht. Voortaan zal ik bij het plannen van zulke werkzaamheden raad moeten vragen bij iemand die hier meer ervaring mee heeft. Het gevolg van deze uitloop is dat er een week van de afronding fase is afgehaald. Dit was mogelijk omdat er al in het begin van het project rekening is gehouden met eventuele uitloop van een fase. Daarom heeft de afrondingsfase in het begin extra tijd toegewezen gekregen. Verder is de iteratie succesvol verlopen en is er veel nuttig werk verricht.

De database is volledig gebouwd en operationeel. Dit heeft wel meer tijd gekost als van tevoren ingeschat. Dit is een wijze les voor vervolg projecten. Ik heb geleerd dat er bij het bouwen van een database meer komt kijken dan tabellen en relaties invoeren. Er moet ook gelet worden op een consistente naamgeving. En het testen door middel van query's uit te voeren neemt veel tijd in beslag.

Ook is de database gevuld met relevante informatie. Dit is doormiddel van verschillende zelfgemaakte scripts en tools gedaan. Deze scripts en tools halen data uit bestaande database en converteren deze zodat deze bruikbaar is voor de CRM database. Dit heeft veel tijd gekost omdat daarvoor tijdelijke aanpassingen nodig waren in de database. Er moesten bijvoorbeeld een aantal tijdelijke tabellen toegevoegd worden. Ook moesten relaties in de database regelmatig worden afgebroken tijdens het testen van het vullen. Dit moest omdat tabellen regelmatig leeg gehaald moesten worden maar dit was niet mogelijk omdat het DBMS integriteit wilde afdwingen tussen de tabellen. Tenslotte waren er veel velden waarvoor bepaald was dat deze niet NULL mochten zijn. Omdat er in de bron data echter wel veel NULL waardes zaten ging het importeren gepaard met veel fouten. Om deze fouten te voorkomen is besloten om tijdelijk op sommige velden toch wel NULL waardes toe te staan.

Ik heb besloten om de database wel te testen maar de front-end applicatie nog niet. Laatstgenoemde was immers nog lang niet uitontwikkeld en een test zou daarom voor het grootste deel falen. De database heb ik wel getest aan de hand van de requirements en de use case scenario's. Voor het testen zijn test cases opgesteld waarin alle voor de database relevante requirements en use case scenario's zijn opgenomen. Hiermee kon ik bepalen of de database kon voldoen aan de gestelde eisen. De resultaten van deze tests zijn opgenomen in het test document.

Tenslotte is er ook een concept gemaakt van de front-end applicatie van het Infodis CRM. Deze applicatie was puur ter demonstratie dat het systeem werkt. Ook konden medewerkers hierdoor zien waaraan tot die tijd is gewerkt. Deze concept applicatie kan uiteindelijk doorontwikkeld worden tot een definitieve applicatie. Tijdens het maken van deze applicatie heb ik intensief samengewerkt met Willem. Hij heeft al het programmeerwerk voor mij gedaan. Ik vond het wel jammer dat ik hem niet met alles kon helpen aangezien mijn programmeerkennis niet groot is. Toch heb ik veel geleerd van zijn uitleg en heb ik gezien hoeveel werk zo'n relatief kleine applicatie is. Waar ik wel mee kon helpen is het schrijven van SQL query's. Dit is de eerste keer dat ik mijn kennis hiervan in de praktijk heb kunnen gebruiken en het is me goed bevallen.

## 8 FASE 3: AFRONDING

In dit hoofdstuk wordt besproken hoe de afronding van dit project is verlopen en hoe het project is overgedragen aan Infodis.

### 8.1 INTRODUCEREN CONCEPT SYSTEEM

Het in de Elaboration fase gebouwde systeem was compleet genoeg om aan alle medewerkers te getoond te worden. Na het tonen was het de bedoeling dat zij aan de hand van het front-end concept hun mening konden geven. Voordat het zover was moest er eerst nog geregeld worden dat het Infodis CRM voor iedereen beschikbaar was. En ook moest het nog officieel gepresenteerd worden.

Het distribueren van de applicatie was niet veel werk omdat er regelmatig tools e.d. beschikbaar worden gemaakt voor alle medewerkers. De rechten op de test database moesten wel worden aangepast zodat iedereen de gegevens kon uitlezen. Toen dit beide gedaan was konden medewerkers zelf het Infodis CRM bekijken. De eerste volgende stap was het bekendmaken en tonen van de applicatie in een presentatie.

Ik heb de presentatie van dit project gehouden voor alle Infodis medewerkers. In de presentatie heb ik niet alleen de applicatie gepresenteerd maar ook verteld over het verloop van het project. Allereerst natuurlijk de aanleiding van het project, de oplossing, de opbouw van het Infodis CRM en over het vervolg van dit project. Vervolgens heb ik in een demonstratie laten zien hoever we zijn gekomen. En ook uitgelegd hoe medewerkers vanaf hun eigen werkplek de applicatie kunnen opstarten. Tenslotte heb ik gevraagd of de medewerkers feedback konden geven zodra ze iets tegenkwamen wat niet duidelijk was of waarvan ze dachten dat het beter kon.

### 8.2 OPSTELLEN OVERDRACHT DOCUMENT

Ik heb in deze fase ook een document geschreven waarin alles rondom de overdracht van dit project beschreven wordt. Dit heb ik gedaan in plaats van het schrijven van een handleiding voor het Infodis CRM. De applicatie was immers nog lang niet af en een handleiding ervoor schrijven zou niet nuttig zijn. Deze zou al snel niet meer aansluiten op de werkelijkheid.

In het overdracht document heb ik alle documentatie rondom het project nog eens opgesomd. Ook heb ik ervoor gezorgd dat alle documentatie compleet op de netwerkschijf van Infodis was te vinden. Hier heb ik alle definitieve versies van de documenten geplaatst. Oudere versies zijn hierbij gearchiveerd.

Daarna heb ik beschreven hoe dit project vervolgt moet worden. Dit heb ik gedaan in vorm van een checklist waarop alle nog te nemen stappen zijn beschreven. Het zijn vooral nog lopende items waaraan gewerkt moet worden in de periode na mijn afstudeerperiode. Deze checklist heb ik met input van mijn begeleider en Willem Kosmeijer gemaakt.

Als de checklist is afgerond betekend dit uiteraard nog niet dat het project is afgelopen. Er zullen dan wel weer nieuwe punten op de planning komen waaraan gewerkt gaat worden. Tenslotte heb ik nog voorbeelden/informatie gegeven waarmee items op de checklist afgehandeld kunnen worden.

## 8.3 ITERATION ASSESSMENT

Dit was een korte iteratie en deze is binnen de geplande tijd afgerond. Met het afsluiten van deze iteratie is ook een einde gekomen in mijn aandeel aan het project. Het project is inmiddels overgedragen aan Infodis en zij zullen eraan verder werken.

Tijdens de presentatie en demonstratie van het Infodis CRM hebben medewerkers kunnen zien waaraan is gewerkt in de periode dat ik bij Infodis aan het werk was. Zij hebben gezien wat er met hun input is gebeurd en er is uitgelegd hoe het project nu verder gaat. De reacties na de presentatie waren zeer positief en medewerkers wilde graag met het systeem aan de gang gaan. Ik verwacht daarom dat er veel feedback zal komen op de reeds gebouwde onderdelen van het systeem.

De voorbereiding van de presentatie had beter gekund achteraf. Ik kwam er pas laat achter dat de applicatie niet wilde openen op de pc in de vergaderzaal. Hierdoor moest ik op het laatste moment nog ervoor zorgen dat dit wel kon. Gelukkig heeft dit niet voor vertraging gezorgd. Wat wel voor vertraging heeft gezorgd is dat de fileserver opnieuw werd opgestart precies op het moment dat ik wilde beginnen met de presentatie. Gelukkig kon iedereen hier wel om lachen.

Mijn begeleider heeft aangegeven dat ik voor de overdracht van het project kort op papier moest zetten aan welke zaken nog gewerkt moet worden en waar we nog mee bezig waren. Deze zaken heb ik daarom opgesomd in het overdracht document. Dit was een nieuw iets voor mij want het is de eerste keer dat ik een nog lopend project verlaat. Ik ben door school en eerdere stages gewend een project af te sluiten na een periode. Mijn begeleider heeft mij echter verzekerd dat het opsommen van deze zaken voldoende zou zijn voor hun om eraan verder te werken. We hebben afgesproken dat er bij vragen contact met mij opgenomen kon worden.

Ik heb ook met hem besproken in hoeverre het doel van mijn afstudeerproject is bereikt. Naar zijn mening zijn we verder gekomen dan het oorspronkelijke doel. In eerste instantie lag de focus op het opleveren van een ontwerp. Uiteindelijk is er een ontwerp opgeleverd waarvan een groot stuk gebouwd is.

Voordat mijn afstudeerperiode definitief voorbij was heb ik nog kunnen genieten van een lekker gebakje en een leuke afscheidsspeech van Kees. En na iedereen nog even gedag te hebben gezegd trok ik de deur voor de "laatste" keer achter me dicht.

## 9 AFWIJKINGEN MET HET AFSTUDEERPLAN

In dit hoofdstuk worden afwijkingen ten opzichte van het originele afstudeerplan beschreven.

### 9.1 PROOF OF CONCEPT

Was gepland voor de Inception fase maar is verplaatst naar de Elaboration fase.

Zie hoofdstuk 5.6 voor een uitleg.

### 9.2 SYSTEM SEQUENCE DIAGRAM

Deze diagrammen had ik opgenomen in mijn afstudeerplan onder methoden en technieken. Ik heb deze alleen niet gebruikt tijdens mijn afstuderen. Willem Kosmeijer heeft tegen mij gezet dat dit hele handige diagrammen kunnen zijn maar dat hij deze in dit project niet nodig had. De interactie tussen systeem en gebruiker is ook niet dermate ingewikkeld. Hij kon het systeem ook aan de hand van de use case scenario's en het datamodel maken.

Gevolg hiervan is wel dat er minder documentatie is omtrent de gegevens uitwisseling tussen gebruiker en systeem is. Ik voorzie in de toekomst geen grote problemen aangezien het vrij logisch is wat voor interactie er is in een dergelijk systeem. Dit is onder andere te danken aan de duidelijke scheiding tussen verschillende soorten klantgegevens in de database.

### 9.3 ETL

In het afstudeerplan is niet opgenomen dat ik gebruik zou maken van ETL technieken. ETL staat voor Extract, Transform en Load. ETL is een techniek die gebruikt wordt om data uit verschillende bronnen op te halen en deze te importeren in een andere bron. Bronnen zijn hierbij meestal databases. Ik heb besloten dat met deze techniek het vullen van de CRM database goed werd gedocumenteerd. Ook konden developers op basis van ETL schema's scripts schrijven om data uit de ene database naar de andere te exporteren.

*"ETL is de afkorting voor Extraction, Transformation and Load. De term ETL staat voor een groep technologieën die veelal gebruikt worden bij de koppeling tussen systemen, waarbij er gestreefd wordt naar een minimale technische en semantische koppeling tussen de systemen. Het is een batchproces dat regelmatig gebruikt wordt."*

- *Extract = leest data van een bron en pakt het gewenste pakket uit*
- *Transform = Zet opgenomen data om, gebruik makende van regels, opzoektabellen of maakt combinaties van data van verschillende bronnen*
- *Load = schrijft de data naar een gewenst doel"*

Bron: <http://nl.wikipedia.org/wiki/ETL>

### 9.4 SQL (STRUCTURED QUERY LANGUAGE)

Een techniek die ik niet heb vermeld in het afstudeerplan maar wel enorm veel gebruikt heb is SQL. Deze techniek wordt gebruikt om gegevens uit een database te halen door middel van query's. Ook kan je met SQL databases opbouwen en beheren. Een zeer bruikbare techniek voor een project als deze.

Ik had al eerder gewerkt met SQL en daarom had ik hier al wat kennis van. Hier had ik wel een voordeel mee. Als ik SQL helemaal vanaf niets had moeten leren beheersen midden in een afstudeerperiode had ik veel tijd verspild. Alsnog heeft het me een aantal dagen geduurd voordat ik de taal weer helemaal onder de knieën had.

## **9.5 DOCUMENTATIE IPV HANDLEIDING**

In het afstudeerplan is gepland dat er een handleiding gemaakt zou worden. Er is geen alleen geen handleiding gemaakt voor de applicatie omdat deze pas in de beginfase van ontwikkeling was. Het zou daarom niet handig zijn om er een handleiding voor de te schrijven. Deze handleiding zou vrijwel zeker binnen enkele weken weer achterhaald zijn. In de plaats daarvan is uitgebreide documentatie over de database geschreven. Deze was immers al wel volledig afgebouwd.

## 10 COMPETENTIEVERANTWOORDING

In dit hoofdstuk beschrijf ik hoe ik heb gewerkt aan mijn competenties. Het beschrijven heb ik gedaan met behulp van de STARR methode.

### 10.1 BUSINESS ANALYSE

- Modelleren business activity diagram.
- Relevante bedrijfsprocessen modelleren, daarmee inzicht krijgen in de werkwijze bij Infodis.
- Afbakenen beschouwingsgebied.
- Glossary opstellen om vaktermen vast te leggen.

#### Situatie

Ik was net begonnen met afstuderen bij Infodis en ik wilde inzicht krijgen in de organisatie, het vakgebied en ook in de bedrijfsprocessen. De afstudeeropdracht was mij duidelijk en ik begon met het uitzoeken van hoe informatie over klantgegevens in de huidige situatie werd ingevoerd / bijgehouden. Ik wilde weten hoe en wanneer de klantgegevens ingevoerd werden.

#### Taak

De taak die ik had was om op een relatief hoog niveau de bedrijfsprocessen te modelleren waar mijn opdracht op van toepassing was. Uiteraard moest ik eerst ontdekken welke bedrijfsprocessen dit waren. Daarom moest ik eerst uitzoeken bij welke processen klantgegevens werden gebruikt. Hierbij moest ik rekening houden met alle medewerkers binnen Infodis en de bij hun horende functies. Dit moest omdat ik een accuraat model wilde maken waarin ik geen “actoren” wilde missen.

Vervolgens zou ik met mijn begeleider afspreken wat er binnen het beschouwingsgebied van mijn opdracht zou vallen en wat niet.

Ik heb ook, voornamelijk voor mijzelf, een glossary opgesteld om alle relevante vaktermen vast te leggen die gebruikt worden in de organisatie. Dit was nodig om volledig te begrijpen wat er allemaal gedaan werd. En ook zodat iedereen die bij het project betrokken was hetzelfde begrip over deze vaktermen zou hebben.

#### Aanpak

Ik heb deze taak aangepakt door de volgende werkzaamheden uit te voeren:

- Oriëntatie binnen Infodis
  - a. De website doorlezen
  - b. Alle medewerkers een keer spreken
  - c. Documentatie op de netwerkschijven doorlezen
  - d. De verschillende afdelingen benoemen / de functies onderscheiden
  - e. Organogram opstellen
- Interviews met medewerkers
- Bestaande procesbeschrijvingen bestudeerd
- Modelleren bedrijfsproces “Klantgegevens bijhouden”
- Knelpunten inventariseren



## Resultaat

- Duidelijk beeld van de organisatie
- Organogram
- Business Activity Diagram van de huidige en toekomstige situatie
- Knelpunten aangetoond
- Stakeholders ontdekt en beschreven

## Reflectie

Iets wat ik een volgende keer anders zou doen is met meer mensen specifiek de huidige gang van zaken rond de bedrijfsprocessen bespreken. Nu heb ik bijvoorbeeld met drie van de vijf accountmanagers gesproken over de bedrijfsprocessen. Wellicht had ik van de overige twee nog meer waardevolle informatie kunnen krijgen.

Ik had hun kunnen vragen of zij het bedrijfsproces model konden verifiëren voor mij. Op die manier zou het ook gelijk duidelijk geweest zijn of het model klopte met de werkelijkheid door een bevestiging van hun kant.

Ik vond het goed dat ik het modelleren op een hoog niveau heb gedaan. Dit was een goede keus voor de omvang en complexiteit van dit project. Het was namelijk niet nodig om de bedrijfsprocessen tot in detail te modelleren. Het probleem werd op een hoog niveau ook al duidelijk zichtbaar.

## 10.2 INFORMATIE ANALYSE

- Modelleren bedrijfsgegevensmodel.
- Modelleren van de informatie objecten rondom de relevante bedrijfsprocessen.
- Benoemen welke informatie objecten relevant zijn voor dit project.
- Pragmatische kwaliteit van de informatie objecten bepalen.

### Situatie

Het systeem wat ik moest ontwerpen was een klantgegevens systeem. Er was aan de start van het project nog niet veel duidelijkheid over hoe de oplossing, dit systeem, vormgegeven moest worden. Tevens was nog niet duidelijk waar alle klantgegevens nu waren vastgelegd. Om hierin duidelijkheid te krijgen ben ik gaan onderzoeken wat voor klantgegevens er allemaal werden vastgelegd.

### Taak

Om te weten wat voor oplossing er gemaakt diende te worden moest ik gaan uitzoeken wat voor klantgegevens er allemaal werden bijgehouden. En daarnaast ook gelijk vastleggen waar deze werden vastgelegd, op welke locatie, welk medium, etc... Ook wilde ik kijken of de verschillende afdelingen erg van elkaar afwijkende klantgegevens nodig hadden om hun werk goed uit te voeren.

### Aanpak

Om dit allemaal uit te zoeken ben ik medewerkers gaan interviewen. In de interviews vroeg ik wat voor klantgegevens er allemaal gebruikt werden. Een vraag was bijvoorbeeld: "Als je een contactpersoon van bedrijf X wilt benaderen, waar zoek je dan de gegevens van die contactpersoon op?". Bij andere medewerkers heb ik ook meegekeken met de dagelijkse werkzaamheden. Als ik dan dingen zag die ik niet begreep of waarvan ik dacht dat ze mogelijk interessant waren ben ik daar vragen over gaan stellen. Aan de hand van deze twee methodes kon ik al veel bronnen van klantgegevens benoemen.

Ook heb ik onderzocht of sommige klantgegevens wellicht alleen voor bepaalde klanten worden opgeslagen. Bijvoorbeeld speciale instellingen die eigenlijk maar voor één klant geleiden. Van dit soort gegevens wilde ik eerst weten of ze binnen het beschouwingsgebied van mijn opdracht vielen.

Het bekijken van netwerkschijven en bestaande databases was ook een onderdeel van de informatie analyse. Ik heb bekeken wat er op de netwerkschijven aan informatie werd bijgehouden en wat daarvan klantgegevens betrof. In de databases heb ik gekeken wat voor klantgegevens daarin werden bijgehouden. Ook heb ik daarvan alle velden benoemd die klantgegevens bevatte. Daarbij heb ik geïnventariseerd in hoeveel gevallen de informatie hier ook echt compleet was ingevuld.

### **Resultaat**

Medewerkers hebben me laten zien in welke programma's en op welke plekken ze gegevens opsloegen. Er werd duidelijk welke informatie belangrijk was en ook voor welke stakeholders. Ook werd duidelijk welke informatie eenvoudig toegankelijk was en welke minder eenvoudig.

Alle informatie objecten zijn geïnventariseerd, in eerste instantie alleen globaal maar naarmate het project vorderde steeds gedetailleerder. Van alle informatie objecten is bepaald wat de pragmatische kwaliteit van de inhoud was. De bruikbare informatie in de informatie objecten zijn uiteindelijk gebruikt om een database ontwerp te maken.

### **Reflectie**

In de toekomst zou ik eerder gaan kijken wat de kwaliteit van de informatie objecten is. Hiermee ben ik nu laat begonnen en daardoor kwam ik er laat achter dat niet alle informatie van hoge kwaliteit was. In het geval van dit project was dit niet zo'n groot probleem omdat van te voren al is gezegd dat veel informatie incompleet is. Dat was ook de aanleiding om dit project te starten. Ik begrijp echter wel dat dit in andere projecten een hele grote impact kan hebben. En daarom zou ik voortaan een hogere prioriteit geven aan het beoordelen van de kwaliteit van informatie objecten.

Ik ben wel tevreden met mijn aanpak van de informatie analyse in dit project. Ik ben niet gelijk begonnen met het in detail analyseren van de geïnventariseerde gegevens. De prioriteit lag in eerste instantie op het uitzoeken van welke gegevens echt vastgelegd moesten gaan worden. Later ben ik gaan kijken uit welke informatie objecten ik deze informatie kon vergaren. Daardoor ben ik er ook achter gekomen dat veel gewenste gegevens nog helemaal niet waren vastgelegd. Deze gegevens zou ik nooit ontdekt hebben als ik me alleen maar op bestaande informatie objecten had gefocust.

## 10.3 SYSTEEM ARCHITECTUUR

- Ontwerpen gebruikersfuncties
  - Wensen en eisen (requirements) van gebruikers aan het te ontwerpen systeem vastleggen.
  - Use cases opstellen
  - Functionaliteit van het systeem ontwerpen. Tonen hoe deze functionaliteit de gebruikers ondersteunt.
- Valideren ontwerp gebruikersfuncties
  - Het ontwerp valideren met de wensen en eisen.
- Bouwen functioneel prototype
  - Bouwen van een "Proof of Concept" van de belangrijkste use cases.
  - Implementeren belangrijkste use cases.

### Situatie

Ik moest een systeem ontwerpen waarvan ik nog niet wist wat de functionaliteiten ervan zouden worden. Ook wist ik nog niet wat de beste oplossing voor het systeem zou worden. Het aanpassen van een bestaand systeem, een stand-alone applicatie of een website. Het behoorde allemaal tot de mogelijkheden. Bij het ontwerpen en bouwen zou ik samenwerken met een developer.

### Taak

Ik moest uitzoeken welke functionaliteiten gewenst waren om de knelpunten op te lossen. De bedoeling was dat medewerkers hierbij wensen en eisen zouden aanleveren. De functionaliteiten moest ik aan de hand van deze wensen en eisen beschrijven en ontwerpen. Vervolgens moest ik bepalen of deze functionaliteiten in een bestaand systeem te implementeren waren of dat er een nieuw systeem gebouwd moest worden.

Op basis van de wensen en eisen moest ik een POC maken waarin de functionaliteiten terug waren te vinden. Hiermee kon ik aantonen of de ontworpen functionaliteiten daadwerkelijk een oplossing waren voor de knelpunten.

Tenslotte moest ik op basis van mijn ontwerp een developer aan het werk zetten om de belangrijkste use cases te bouwen.

### Aanpak

Ik ben begonnen met het interviewen van medewerkers om zo hun wensen en eisen aan de oplossing te verzamelen. Deze ben ik vervolgens tot requirements gaan vormen. Om de requirements op een geordende manier vast te leggen heb ik een tool gemaakt in Excel. Hierin kon ik de requirements allerlei extra informatie meegeven waardoor deze heel specifiek werden.

Van de gespecificeerde requirements zijn weer functionaliteiten afgeleid. Ook heb ik zelf nagedacht over functionaliteiten om de knelpunten op te lossen. Al deze functionaliteiten heb ik genoteerd en daarna bepaald of deze in een bestaand systeem gerealiseerd konden worden. De functionaliteiten heb ik vastgelegd in use cases en een use case model. In elke use case heb ik op basis van de requirements beschreven hoe het systeem met de gebruiker interacteert. Na deze stap heb ik bepaald of er een bestaand systeem was waarin deze functionaliteiten verwerkt konden worden.

Om een Proof of Concept te maken heb ik schetsen van interfaces gemaakt op een eenvoudig doch duidelijke manier. Hierbij heb ik gebruik gemaakt van een template voor een soort rapid prototyping. Deze interfaces konden ook gebruikt worden om medewerkers te laten zien hoe hun requirements verwerkt waren.

Uiteindelijk moest ik de ontworpen functionaliteiten laten bouwen door een developer. Hierbij moest ik hem ondersteunen en controleren of alles correct uit het ontwerp werd overgenomen. Soms moest ik hierbij de developer overtuigen van beslissingen in het ontwerp.

## **Resultaat**

Er is een tool gemaakt waarin alle requirements opgeslagen kunnen worden. De medewerkers hebben heel veel wensen en eisen aangeleverd. Er is vervolgens een lange lijst met requirements opgesteld die van toepassing was op de oplossing. Aan de hand daarvan kon functionaliteiten omschreven worden en gewerkt worden aan use cases.

In de use cases en het use case model worden alle functionaliteiten beschreven waaraan de oplossing moet voldoen. Hiermee is duidelijk gedocumenteerd hoe de functionaliteiten de gebruikers ondersteunen en hoe de interactie met het systeem is.

Op basis van alle beschreven functionaliteiten is bekeken of deze in een al bestaand systeem van Infodis geïmplementeerd konden worden. Dit bleek niet het geval. Daarom is in overleg met de opdrachtgever bepaald dat er een nieuw systeem gebouwd zou gaan worden.

Om te bewijzen dat het ontworpen systeem werkte is een Proof of Concept gemaakt. Medewerkers hebben hier positief of gereageerd. Daarna is aangetoond dat de functionaliteiten compleet waren en dat het systeem gebouwd mocht gaan worden.

Op basis van het ontwerp heeft de developer een concept front-end applicatie gemaakt. Hierin waren de hoogst geprioriteerde use cases gerealiseerd. Deze front-end applicatie is gebruikt om het systeem te presenteren/demonstreren aan de medewerkers van Infodis.

## **Reflectie**

Het interviewen en verzamelen van wensen en eisen was veel werk. Dit heeft mij veel energie gekost. Ik ben wel blij dat ik dit gedaan heb omdat het ontwerp erdoor goed aansluit op de verwachting. Deze ervaring kan ik daarnaast goed gebruiken in toekomstige projecten. Ook het maken van een eigen requirements tool is een verstandige beslissing geweest. Zonder dit was het managen van de requirements een rommeltje geworden.

Het gebruik van use cases is mij goed bevallen. Dit omdat je ze op maat kan maken voor het project waar ze op van toepassing zijn. En je hoeft ermee niet direct tot in detail de functionaliteiten te beschrijven. Als je use cases kan je ook heel goed een totaal plaatje van het systeem in de gaten houden. Daarnaast is het makkelijk om de verschillende onderdelen te prioriteren.

Aan de beslissing om een nieuw systeem zelf te ontwikkelen had meer onderzoek aan vooraf kunnen gaan. Dat de functionaliteiten niet in een bestaand systeem geïmplementeerd konden worden was duidelijk. Misschien had er wel een systeem aangeschaft kunnen worden waarin deze functionaliteiten wel zaten. Er werd echter al snel besloten dat er niet teveel tijd verspild moest worden aan het zoeken naar zo'n systeem. Dit is achteraf ook wel goed te praten omdat het zelf ontwikkelen van een nieuw systeem snel is gegaan. En ook omdat dit systeem volledig op maat gemaakt is.

Het Proof of Concept is positief ontvangen door de medewerkers. Dat is goed omdat deze is gemaakt op basis van hun wensen en eisen. Het maken hiervan is mij bevallen omdat je ermee snel kan bepalen of requirements realistisch zijn. Ook kan je zonder heel veel werk te verrichten aan medewerkers laten zien wat je wilt gaan bouwen. Het laten zien van een interface ontwerp heeft ook een heel ander effect dan het laten lezen van een requirement. Vaak kreeg ik veel meer feedback op een ontwerp.

De samenwerking met Willem Kosmeijer van development is mij zeer bevallen. Omdat we op dezelfde afdeling zaten konden we snel communiceren en onduidelijkheden vermijden. Ik heb veel geleerd van zijn uitleg over de code en programmeren in het algemeen. Dit is de eerste keer dat ik het programmeer proces heb meegemaakt en ik begrijp nu beter wat programmeurs nodig hebben om hun taak uit te voeren. Daarnaast weet ik nu hoe belangrijk het is om regelmatig overleg te hebben tussen ontwerper en bouwer. Als je dit niet doet dan moet er waarschijnlijk vaak werk aangepast worden.

## 10.4 INFORMATIE ARCHITECTUUR

- Ontwerpen database
  - Modelleren van een database voor het te realiseren informatiesysteem.
- Bouwen prototype database
  - Implementeren van een beta versie aan de hand van een databaseontwerp.
  - Valideren databaseontwerp.
- Bouwen database
- Vullen database

### Situatie

Er was besloten dat er een nieuw systeem ontwikkeld moest worden. Voor dat systeem moest ook een database ontwerp gemaakt worden. Dit moest gebeuren op basis van de verzamelde informatie objecten. Van deze objecten was al bepaald in hoeverre de informatie van hoge kwaliteit was. Bij het ontwerpen kreeg ik ondersteuning van een developer.

### Taak

Ik moest een database ontwerp maken waarmee de oplossing ondersteund zou worden. Het ontwerp moest gebaseerd zijn op de verzamelde informatie objecten. Ook moesten de klantgegevens in een bepaalde structuur worden opgeslagen die we bedacht hadden.

Op basis van het ontwerp moest ik vervolgens een prototype database bouwen. Hiermee kon ik gaan testen of de database aansloot op de functionaliteiten in de use cases. Het bouwen moest gebeuren op een SQL Server platform. Als het prototype was goedgekeurd kon er begonnen worden aan het bouwen van de volledige database.

Tenslotte was het de bedoeling dat de gebouwde database gevuld zou worden vanuit bestaande databases bij Infodis.

### Aanpak

Ik ben het ontwerp begonnen met het vormen van entiteiten op basis van de informatie objecten die van hoge kwaliteit waren. Deze entiteiten heb ik daarna aan elkaar gekoppeld door relaties te bepalen. Hierbij heb ik constant rekening gehouden met de use cases en met de structuur die afgesproken was met de opdrachtgever. Use cases zijn af en toe aangepast om beter te matchen met het databaseontwerp. Toen ik alle informatie objecten had opgenomen in het ontwerp heb ik deze laten controleren door meerdere developers.

Van het ontwerp is vervolgens een prototype gebouwd om te controleren of het ontwerp valide was. Dit heb ik getest door de database met testgegevens te vullen en er vervolgens een aantal query's op los te laten. Toen bleek dat deze query's geen problemen veroorzaakte ben ik begonnen met het volledig bouwen van de database. De database is daarna ook nog getest door te controleren of deze kon voldoen aan de use case scenario's en de requirements.

Toen de database alle tests had doorstaan kon begonnen worden met het vullen ervan. Omdat het databaseontwerp gebaseerd is op bestaande informatie objecten was eenvoudig te bepalen waar de gegevens vandaan moesten komen. Om het proces goed te documenteren heb ik besloten om ETL schema's te maken. Aan de hand van deze schema's konden vervolgens conversie scripts en tools gemaakt worden.

## **Resultaat**

Er is een valide ontwerp gemaakt van de database. Deze ondersteunt de use cases volledig omdat ze volledig consistent aan elkaar zijn gemaakt. Het database ontwerp is eerst als prototype gebouwd in SQL Server. Hierop zijn verschillende query's getest met een positief resultaat. Het prototype is daarom goedgekeurd waardoor er groen licht was voor het volledig bouwen van de database. Wat daarna dus ook gebeurd is. Tenslotte moest de database nog gevuld worden. Dankzij de ETL schema's konden er scripts geschreven worden waardoor er al snel gegevens geïmporteerd werden. Hiervoor moesten in sommige gevallen tijdelijke tabellen worden toegevoegd. Het vullen vanuit diverse databases heeft veel tijd in beslag genomen omdat ze erg verschillend van opbouw waren. Uiteindelijk is het vullen we succesvol geweest omdat ermee de basis is gelegd voor een volledig gevulde database.

## **Reflectie**

Het ontwerpen van de database heeft in totaal veel tijd gekost. Dit kwam omdat het ontwerp meerdere malen gecontroleerd en aangepast is. Dit was nodig omdat het ontwerp 100% moest matchen met het systeem wat gebouwd moest worden. Ook moest 100% zeker zijn dat alle relevante gegevens erin opgeslagen konden worden.

Het bouwen van het prototype ging me goed af ondanks dat ik nog nooit met SQL Server Management Studio had gewerkt. Het opnieuw oppakken van de SQL taal heeft wel een aantal dagen geduurd. Dit had ik wel verwacht omdat ik het bijna nooit gebruik. Testen met behulp van query's was nieuw voor mij. Ik vind het wel een goede methode omdat je er eigenlijk een hele use case mee kan testen door de scenario regels om te zetten in query's.

Het volledig bouwen van de database was meer werk dan ik van te voren had gedacht. Je moet met veel dingen rekening houden en je moet heel erg precies werken. Als ik even niet oplette had ik bijvoorbeeld al een fout in naamgeving gemaakt of een sleutel verkeerd ingesteld. Ik denk dat het daarom ook belangrijk is geweest dat ik Willem had die mijn werk controleerde. Met een frisse blik zie je dat soort foutjes direct en kunnen ze gelijk opgelost worden.

Het vullen van een database vanuit bestaande databases was ook nieuw voor mij. Ik zal door wat ik hiervan geleerd heb nooit onderschatten hoeveel werk dit kan zijn. Ook hierbij komen veel kleine zaken voorbij waar rekening mee gehouden moet worden. Bijvoorbeeld welke records je wel importeert en welke niet. Vanaf welk punt in de tijd zijn records relevant, zijn de gegevens compleet, kunnen er gegevens samengevoegd worden om ze compleet te maken, etc. Dit is wel heel erg leuk werk om te doen. Ik denk dat het komt omdat je er heel veel handwerk mee kan besparen als je naar creatieve oplossingen zoekt.

## 11 EVALUATIE

In dit hoofdstuk de eindevaluatie van het gehele proces en het product. Omdat ik elke iteratie ook een evaluatie gemaakt heb zal ik het in dit hoofdstuk globaler doen.

### 11.1 PROCES

Mijn draai vinden binnen Infodis is snel gegaan. Ik werd overal en door iedereen vriendelijk ontvangen en kon daardoor makkelijk vragen stellen. Het verzamelen wensen en eisen is goed verlopen dankzij de behulpzaamheid van de medewerkers bij Infodis. De interviews die ik heb gehouden hebben veel informatie opgeleverd en ik ben daarbij ook niet tegen weerstand aangelopen.

Een verbeterpunt voor een volgend project zou zijn dat ik alle medewerkers van een afdeling zou interviewen, mits dit mogelijk is. In dit project heb ik van sommige afdelingen niet iedereen geïnterviewd. Daarmee is altijd een risico dat iemands belangen niet worden behartigd. Ik hoop dat ik dat genoeg heb opgevangen door regelmatig langs te lopen bij verschillende medewerkers om mijn werk te laten controleren.

Het correct specificeren van requirements heeft me af en toe moeite gekost omdat ik soms niet goed kon bepalen wat voor soort requirement iets was. Na veel oefening en goed het boek “Succes met de requirements” gelezen te hebben ging dit veel beter. Aan de hand van dat boek kon ik ook mijn eigen requirements tool bouwen. Iets wat een essentieel hulpstuk is bij het managen van requirements.

Het werken met OpenUP is mij goed bevallen. De methode sloot goed aan op dit project en daardoor had ik meer tijd om me op de werkzaamheden te richten. Door elke iteratie apart te plannen kan je goed omgaan met de tijd die je hebt gekregen. Ik heb wel af en toe moeite gehad met de uitleg op de website van OpenUP, deze is niet altijd even duidelijk. Verder vind ik het heel fijn dat OpenUP heel flexibel is. Daardoor kan je zelf afstemmen welke artefacten je nodig hebt en welke je kunt laten vallen.

Het gebruik van ETL schema's kan als een rare zet gezien worden omdat deze meestal gebruikt worden voor het vullen van een data warehouse. De schema's zijn echter ook zeer effectief gebleken voor het vullen van andere databases. Hiervoor moest ik wel creatief omgaan met de regels die voor ETL schema's gelden. Deze regels zouden heel erg beperkend zijn geweest in dit project.

Mijn samenwerking met Willem Kosmeijer van development is doorslaggevend geweest voor dit project. Zonder hem had ik wel verder gekund maar dan had Infodis vooral veel documentatie gekregen en geen product waar ze mee aan de slag konden gaan. Hij heeft me veel geleerd over het ontwerpen van databases en programmeren in het algemeen. Leuk was dat we elkaar af en toe moesten overtuigen van beslissingen rondom het ontwerp. Ik ben blij dat ik deze ervaring heb opgedaan omdat ik deze zeker kan gebruiken in toekomstige projecten.

## 11.2 PRODUCT

Ik ben erg tevreden over de opgeleverde tussenproducten en het eindproduct.

De tussenproducten waren allemaal erg nuttig in het tot stand komen van het eindproduct. Ook al zijn sommige tussenproducten soms erg vervelend om bij te houden. Uiteindelijk pluk je toch de vruchten van dat bijhouden. Waarschijnlijk heb ik wel meer tussenproducten opgeleverd dan ze bij Infodis gewend zijn. Dit komt vooral omdat ik met een andere methode werk als hun. Toch vond ik het voor mezelf nodig om deze producten op te leveren. Ik heb niet zoveel ervaring als hun en gebruik de tussenproducten daarom als handvatten om het project de goede richting op te sturen.

Ik had met de start van het project niet verwacht dat er zoveel gebouwd had kunnen worden. Hiermee ben ik voorbij het originele doel van het project gegaan. Dat was namelijk alleen het ontwerpen van een systeem en wellicht een klein stukje bouwen. Nu heb ik een volledig werkende database en een concept front-end applicatie opgeleverd. Dit was alleen mogelijk omdat ik gekoppeld ben aan een developer naarmate het project vorderde. Ik ben ook blij dat het eindproduct zo positief is ontvangen door mijn opdrachtgever en de medewerkers. Mijn hoop is dat het Infodis CRM een effectief stuk gereedschap wordt waarmee Infodis klaar is voor de toekomst.

Ik heb na de afstudeerperiode nog een aantal keer contact gehad om te vragen hoe het nu ging met het Infodis CRM en ze zijn deze nog steeds verder aan het bouwen.



## 12 LITERATUURLIJST

**A better fit, characterising the stakeholders** – Ian F. Alexander

**Business-alignment, van business architecture naar software architecture (2012)** – Berry Pieters, Paul de Vries, Jeroen Vuurens.

**Customer relationship management** –  
([http://nl.wikipedia.org/wiki/Customer\\_relationship\\_management](http://nl.wikipedia.org/wiki/Customer_relationship_management))

**Databases, deel 1: de grondbeginselen (1998)** – D.T.G.P. Vogelaars

**Database design with UML and SQL** - Tom Jewett

**Reader: Datawarehouse ontwerp (2009)** – Jeroen Vuurens

**Flowstate** – (<http://miksofsky.blogs.com/flowstate/>)

**GUUUI** – (<http://www.guuui.com>)

**How To Visualize Stakeholder Analysis** – (<http://tynerblain.com/blog/2007/03/13/visualize-stakeholder-analysis/>)

**Improving Data Warehouse and Business Information Quality** - Larry P. English

**Introduction to Transactions (25 Januari 2004)** - Bill Graziano

**Managing software requirements (2003)** – Dean Leffingwell, Don Widrig

**E.123 Notation for national and international telephone numbers, e-mail addresses and Web addresses (2001)** - International Telecommunication Union

**OpenUP** - (<http://epf.eclipse.org/wikis/openup/>)

**Reflectie met de STARR-methode** – Uit het boek **Kwaliteit met beleid**

**Succes met de requirements! (2010)** – Martin Arendsen, Jan Jaap Cannegieter, Arno Grund, Petra Heck, Serge de Klerk, Johan Zandhuis

**The Rational Unified Process, an introduction (2000)** – Philippe Kruchten

**UX Newsfeed** – (<http://www.uxnewsfeed.com/>)

**Wat is Onderzoek? (2007)** – Nel Verhoeven

**Writing Effective Use Cases (2001)** - Alistair Cockburn