

Afstudeerverslag

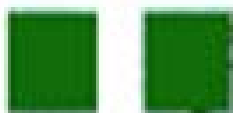
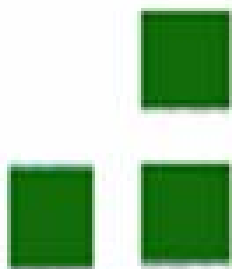
TECHNICO



Automatisering



Ontwikkeling incident managementsysteem



Auteur:	Y. Gruppe
Studentnr:	20006670
Examinatoren:	Dhr. T.W.M. van Gerwen Dhr. J.M.M. van der Zwet
Datum:	05-10-04
Plaats:	Leiden
Afstudeerbedrijf:	Technico Automatisering
Versie:	3.0 (final)

Referaat

Aan:

Dhr. T.W.M. van Gerwen
Dhr. J.M.M. van der Zwet
Dhr. H. Houwaard

Examinator
Examinator
Bedrijfsmentor/opdrachtgever

Afstudeerbedrijf:

Technico Automatisering
Willem Barentszstraat 41
2315 TZ Leiden
Telefoon: 071-5219506
Fax: 071-5219303

Auteur:

Yoran Gruppe
20006670

Studie:

Afdeling I&I: Informatica en Informatiekunde
Studievariant IVIT-4: Informatievoorziening & Informatietechnologie
Afstudeerrichting: Informatievoorziening

Datum:

8 oktober 2004

Trefwoorden:

- Incident management
- Systeemontwikkeling
- IAD ontwikkelmethode
- Veranderingsanalyse
- Prototyping

Voorwoord

Dit afstudeerrapport is geschreven in het kader van het afstuderen aan de Haagse Hogeschool, opleiding Informatica en Informatiekunde. Ik heb mijn afstudeeropdracht uitgevoerd bij het bedrijf Technico Automatisering. De afstudeeropdracht behelsde het ontwikkelen van een incident managementsysteem.

Het primaire lezerspubliek van dit afstudeerrapport wordt gevormd door mijn examinatoren en de toegewezen gecommiteerde, die mijn afstudeerstage onder andere aan de hand van dit document gaan beoordelen. Uiteraard is iedereen die enige affectie heeft met incident management van harte uitgenodigd dit rapport eens een keer door te bladeren. Hiervoor is enige voorkennis op het gebied van informatica wel een vereiste.

Voor het uitvoeren van deze opdracht gaat mijn dank in het bijzonder uit naar de opdrachtgever en bedrijfsbegeleider in de persoon van Henk Houwaard voor het verlenen van de opdracht en voor de tijd en energie die hij in mij heeft gestoken. Daarnaast wil ik alle collega's bedanken die mijn afstuderen tot een succes hebben gemaakt. Naast het feit dat ik altijd met vragen bij hen terecht kon hebben ze ook gezorgd voor een goede werksfeer. Tot slot wil ik Theo van Gerwen en Co van der Zwet bedanken voor hun tijd en begeleiding tijdens de afstudeerstage.

Yoran Gruppe
Leiden, 2 oktober 2004

Inhoudsopgave

1. Inleiding	6
2. Technico Automatisering	7
2.1 Technico automatisering en haar werkzaamheden	7
2.2 Het aanbod van Technico Automatisering	7
2.3 Plek binnen Technico Automatisering	8
3. De opdracht	9
3.1 De omschrijving van de opdracht	9
3.2 De probleemstelling van de opdracht	9
3.3 De doelstelling van de opdracht	9
4. De aanpak van het project	10
4.1 De gebruikte middelen	10
4.1.1 De gebruikte ontwikkelomgevingen	10
4.1.2 De gebruikte methoden	11
4.1.3 De gebruikte ontwikkeltechnieken	12
4.2 De planning	13
5. De veranderingsanalyse	17
5.1 De verkenningsfase	17
5.1.1 Probleeminventarisatie en analyse	17
5.1.2 Vaststellen huidige activiteiten	18
5.1.3 Bepalen doelstellingen	19
5.2 De ontwerpfase	20
5.3 De keuzefase	20
6. De definitiestudie	22
6.1 Specificeren plan van aanpak definitiestudie	22
6.2 Definieren ontwikkelscenario	22
6.3 Definieren systeemeisen	23
6.3.1 Ontwikkelen prototype	23
6.3.2 Bepalen functionele systeemeisen	25
6.4 Bepalen systeemconcept	25
6.4.1 Ontwikkelen use cases	25
6.4.2 Opstellen klassediagram	27
6.5 Beschouwen technische structuur	29
6.6 Beschouwen organisatorische inrichting	29
6.7 Opstellen pilotplan	30
6.8 Verificatie en validatie	31
7. Pilotontwikkeling	32
7.1 Pilot 1: Database	32
7.1.1 Specificeren plan van aanpak database	32
7.1.2 Specificeren globaal functionele structuur database	32
7.1.3 Opstellen pilotontwikkelplan database	35
7.1.4 Verificatie en validatie database	36
7.2 Pilot 2: Meldingsysteem	36
7.2.1 Specificeren plan van aanpak meldingsysteem	37
7.2.2 Specificeren globaal functionele structuur meldingsysteem	37
7.2.3 Opstellen pilotontwikkelplan meldingsysteem	45
7.2.4 Verificatie en validatie meldingsysteem	45
7.3 Pilot 3: Gegevensbeheer	46
8. Testen van het systeem	50
9. Evaluatie	52
9.1 De evaluatie van het proces	52
9.2 De evaluatie van het product	53
9.3 De behaalde leerwinsten	54
9.4 De conclusie	54
Figurenlijst	55
Verklarende woordenlijst	56
Index	59

Bijgevoegde documenten:

Interne bijlagen	60
Interne bijlage 1: De Opdrachtschrijving	60
Interne bijlage 2: Het plan van aanpak.....	64
Interne bijlage 3: Interview incident management	68
Interne bijlage 4: PHP code 'oplossing wijzigen'	70
Externe bijlagen	
Externe bijlage 1: Definitiestudie	
Externe bijlage 2: Pilotontwikkeldocument 1: Database	
Externe bijlage 3: Pilotontwikkeldocument 2: Meldingssysteem	
Externe bijlage 4: Pilotontwikkeldocument 3: Gegevensbeheer	
Externe bijlage 5: Veranderingsanalyse	
Externe bijlage 6: Testplan	

1. Inleiding

De opleiding IVIT4 (InformatieTechnologie en InformatieVoorziening) schrijft voor, dat de studenten van de opleiding afstuderen in een organisatie. Binnen de organisatie dient zelfstandig een project te worden uitgevoerd om te bewijzen dat de student het HBO werkniveau bereikt heeft en de opleiding goed heeft doorlopen. In dit kader heb ik een gewerkt aan de ontwikkeling van een incident management systeem bij Technico Automatisering te Leiden.

Het uitgevoerde project had ten doel incident management mogelijk te maken binnen Technico Automatisering. Om dit doel te realiseren is er enerzijds gekeken naar de introductie van begeleidende procedures bij het beheren van incidenten en anderzijds naar de invoering van een informatiesysteem dat ondersteuning zou bieden bij het registreren van incidenten. Dit afstudeerverslag zal antwoord geven op de vraag hoe ik er in geslaagd ben, incident management binnen Technico Automatisering te introduceren.

Al voor aanvang van de afstudeeropdracht, was er door Technico Automatisering onderzoek gedaan naar de mogelijke aankoop van een commercieel incident management pakket. De organisatie kwam tijdens dit onderzoek tot de conclusie, dat de beschikbare pakketten enerzijds buiten het beschikbare budget lagen. Anderzijds boden deze pakketten extreem veel overbodige functionaliteiten. Zo is de organisatie dan ook dat het besluit gekomen een incident management-systeem op maat te laten ontwikkelen door een afstudeerder.

Dit document is gestructureerd naar de gevolgde stappen volgens de methodiek IAD (Iterative Application Design) met uitzondering van de veranderingsanalyse. De veranderingsanalyse is geen onderdeel uit IAD maar werd wel noodzakelijk geacht om tot een correcte en concrete oplossing te komen alvorens het incident management systeem daadwerkelijk ontwikkeld kon worden. Er is voor deze structurering gekozen omdat het doorlopen proces zo op een chronologische en gestructureerde manier kan worden beschreven, wat niet alleen de overzichtelijkheid van dit document bevordert, maar ook de leesbaarheid en begrijpbaarheid.

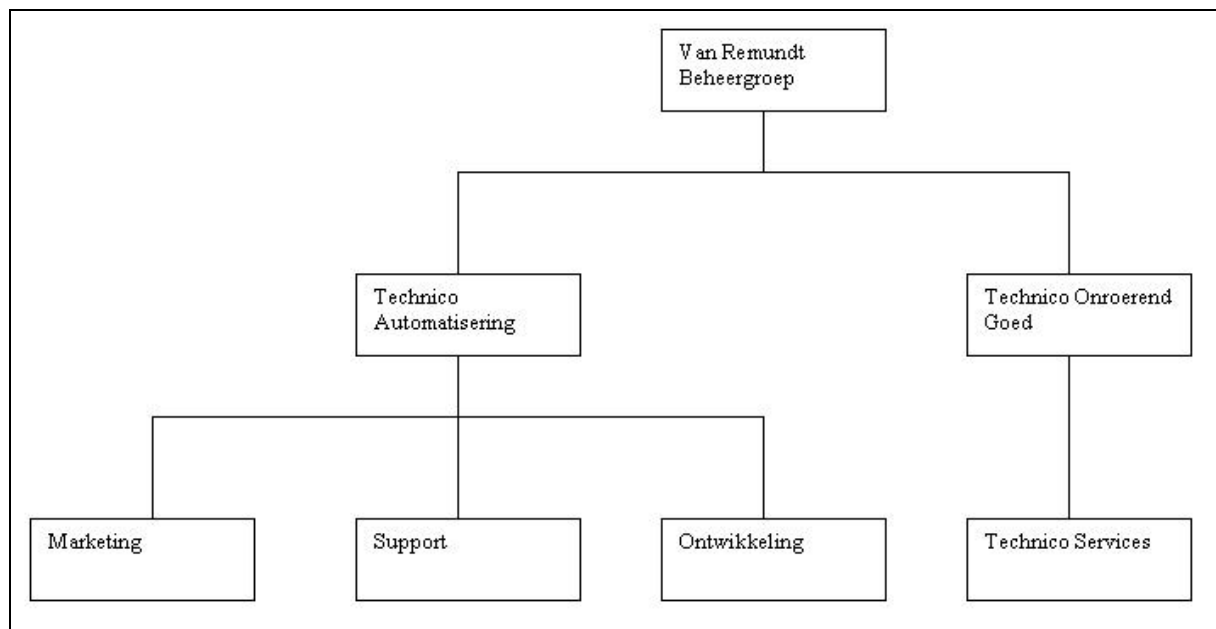
In hoofdstuk twee zal er iets verteld worden over het afstudeerbedrijf Technico Automatisering en over de producten die zij ontwikkeld. In het hierop volgende hoofdstuk zal de opdracht globaal beschreven worden. In hoofdstuk vier zullen de middelen die voor dit project zijn gebruikt nader worden toegelicht. De veranderingsanalyse en de oplossingen die hieruit voortgevloeid zijn zullen in hoofdstuk vijf aan bod komen. Logischerwijs zal in hoofdstuk zes de definitiestudie beschreven worden. In dit hoofdstuk wordt de nadruk gelegd op de systeemeisen, het ontwikkelde prototype en het systeemconcept omdat dit in mijn ogen de meest essentiële onderdelen waren van de definitiestudie. Na de definitiestudie zullen in hoofdstuk zeven de verschillende pilots waaruit het incident managementsysteem is opgebouwd beschreven worden. In dit hoofdstuk zullen per pilot de uitgevoerde activiteiten aan het licht worden gebracht. In hoofdstuk acht word beschreven hoe ik het testplan voor het incident managementsysteem heb opgesteld en hoe ik daarmee het systeem heb getest. Tot slot zal in hoofdstuk negen zowel het doorlopen proces als het uiteindelijk opgeleverde product worden geëvalueerd. Tevens zijn er in dit document diverse bijlagen opgenomen. In de interne bijlagen zijn de opdrachtschrijving en het plan van aanpak te vinden. In de externe bijlagen zijn de veranderingsanalyse, de definitiestudie, de drie pilotontwikkelplannen en het testplan te vinden.

2. Technico Automatisering

Dit hoofdstuk is bedoeld om de lezer inzicht te geven in de omgeving waarin de afstudeeropdracht heeft plaatsgevonden. De afstudeerder heeft zijn afstudeerstage gelopen bij Technico Automatisering. In dit gedeelte wordt een omschrijving gegeven van het bedrijf, de producten die het bedrijf aanbiedt en de plek van de afstudeerder binnen het bedrijf.

2.1 Technico automatisering en haar werkzaamheden

Technico Automatisering B.V. is een dochteronderneming van de Van Remundt Beheergroep. Binnen deze groep richten Technico Services B.V. en van Remundt Consultants hun activiteiten sinds 1985 op de industriële wasindustrie en zijn opgericht om invulling te geven aan de unieke probleemstellingen binnen deze industrie. Dit loopt uiteen van advies in de logistieke procesvoering tot klantgerichte aanpassing op industriële toepassingen binnen het totale reinigingsproces. Daarnaast worden er onderhouds- en installatiewerkzaamheden uitgevoerd. In 2003 is Technico Automatisering B.V. opgericht om een duidelijke scheiding van activiteiten binnen de Van Remundt Beheergroep te creëren. In figuur 2.1 is een organigram te zien van de Van Remundt Beheergroep.



Figuur 2.1 Organigram Van Remundt Beheergroep

Al in 1985 werd duidelijk dat binnen de industriële wasindustrie er grote vraag was naar informatievoorzienende software en het automatiseren van diverse processen. Door het inspringen op deze behoefte, werd de eerste stap gezet op weg naar een uitgebreid softwarepakket gericht op de afhandeling van het persoonsgebonden wasgoed. De samenwerking tussen de Van Remundt Beheergroep en de wasindustrie heeft ervoor gezorgd dat het softwarepakket is uitgegroeid tot een uitgebreid pakket aan softwareprogramma's, welke de gebruiker voorziet in de informatie-behoefte bij de afhandeling van persoonsgebonden goed.

2.2 Het aanbod van Technico Automatisering

Voor de afhandeling van persoonsgebondengoed (PGG) heeft Technico Automatisering systemen ontwikkeld die de integrale logistieke procesgang optimaliseren. De integrale, logistieke procesgang omvat het totale traject van reiniging, afwerking en retournering. De bewaking is

gericht op de mogelijkheid om op ieder gewenst moment de positie van het artikel in de procesgang te bepalen. Verder is de software specifiek ontwikkeld voor de bewerking en verwerking van het persoonsgebondengoed, het opbouwen en verwerken van data en het afdrukken van afleverbonnen, cumulatieve staten en facturen. Met het Kleding Informatie Systeem (TechniKIS) kan door middel van de sorteertoepassingen de sortering van persoonsgebondengoed vereenvoudigd en versneld worden, waardoor het voor minder ervaren personeel mogelijk is, de genormeerde productiesnelheid te halen. Daarnaast is er de mogelijkheid klanten een dienstenpakket op het gebied van sorteermogelijkheden en de wijze van doorbelasting van de kosten aan te bieden. Binnen TechniKIS worden de aan de bewoner en kledingstuk gekoppelde gegevens opgeslagen. Deze gegevens kunnen vervolgens worden opgebouwd, aangepast en uitgebreid. Hierdoor is er de mogelijkheid een geschiedenis op te bouwen, waarin alle noodzakelijke gegevens zijn opgeslagen.

Binnen het systeem TechniKIS zorgen de sorteersystemen voor de afhandeling en sortering van het PGG. Door het gebruik van deze machines worden sorteerfouten geminimaliseerd en kunnen de verwerkingsaantallen per uur worden verhoogd. Door de koppeling van de machines aan sorteerapplicaties wordt voor de gebruiker inzichtelijk welke handeling er uitgevoerd dient te worden. Door het gebruik van verschillende sorteerapplicaties wordt het mogelijk artikelen, voorzien van een label met barcode of met numerieke/ alfanumerieke identificatie, te sorteren.

2.3 Plek binnen Technico Automatisering

Zoals in figuur 2.1 te zien is bestaat Technico Automatisering uit drie afdelingen, te weten marketing, support en ontwikkeling. De daadwerkelijke ontwikkeling van de producten die Technico Automatisering verkoopt, wordt al sinds enkele jaren door een extern softwarebedrijf gedaan. Technico Automatisering speelt in dit proces een sturende en ondersteunde rol. De afdeling 'marketing' is verantwoordelijk voor het volledige commerciële proces. De afstudeeropdracht heeft zich afgespeeld op de afdeling 'support'. Deze afdeling heeft een breed pakket aan taken zoals het installeren van producten bij klanten en het uitvoeren van onderhoud. Het proces waar de opdracht betrekking op had was het hele gebeuren rondom het afhandelen van gemelde incidenten. Op de afdeling 'support' zijn in totaal vier mensen werkzaam. Al deze werknemers zouden in de toekomst met het te ontwikkelen incident managementsysteem te maken krijgen. Bij het vaststellen van het prototype, de systeemeisen en het systeemconcept speelde hierom hun input een belangrijke rol. Om deze input op papier te krijgen heeft de afstudeerder gedurende het project verschillende gesprekken en interviews met deze groep van belanghebbende gevoerd.

3. De opdracht

Dit hoofdstuk betreft de opdracht. In dit hoofdstuk zal een korte omschrijving van de opdracht worden gegeven. Tevens zullen de aanleiding, probleemstelling en doelstelling van de opdracht worden toegelicht. Dit hoofdstuk is in dit document opgenomen om de lezer inzicht te geven in de context waarbinnen dit project zich heeft afgespeeld. Dit hoofdstuk is gebaseerd op de opdrachtomschrijving welke is terug te vinden in interne bijlage één, de opdrachtomschrijving, van dit document.

3.1 De omschrijving van de opdracht

Voor deze opdracht moet er een web-based informatiesysteem ontwikkeld worden. Met dit informatiesysteem moet het mogelijk zijn voor klanten maar ook voor medewerkers zelf om een incident te melden via het systeem. Bij een incident kan het gaan om een storing, klacht of vraag en hij heeft doorgaans betrekking op de door Technico Automatisering geleverde producten. De incidentmelding moet vervolgens bij de juiste medewerker van het bedrijf terechtkomen. Als een medewerker een incidentmelding binnenkrijgt van een klant, moet hij precies kunnen zien welke softwareversie de klant gebruikt, wat de 'known errors' van deze versie zijn en wat voor licentie de klant ervan heeft. Het systeem moet in de toekomst ook andere soorten informatie kunnen gaan beheren voor de medewerkers zoals planningen, urenregistraties en relatiebeheer. Tevens dient er een beheerfunctie ontwikkeld te worden die het beheren van onder andere gebruikers en stamgegevens mogelijk maakt.

3.2 De probleemstelling van de opdracht

De probleemstelling van deze opdracht is tweeledig. Enerzijds heerst binnen het bedrijf het gevoel dat de meldingen van incidenten niet snel en gestructureerd genoeg kunnen worden opgelost. Anderzijds wordt de gedecentraliseerde opslag van gegevens binnen het bedrijf als probleem ervaren. Binnen Technico Automatisering BV wordt op verschillende plaatsen informatie opgeslagen en beheerd. Het betreft hier informatie in de meest brede zin van het woord. Deze informatie loopt uiteen van klantgegevens, vastgelegde storingsgegevens, gegevens inzake het uitvoeren van versiebeheer, prijslijsten enz. Binnen Technico Automatisering leeft het gevoel dat het centraal beheren van zoveel mogelijk informatie grote voordelen heeft met het oog op het delen van kennis en efficiency.

Meldingen van klanten verlopen momenteel telefonisch of via e-mail. Tevens worden vrijwel alle gegevens die verspreid liggen binnen de organisatie, individueel beheerd in 'Excelsheets'. Er is een eerste aanzet gemaakt tot een centraal informatiesysteem in de vorm van een opdrachtbeschrijving. In deze opdrachtbeschrijving staan alle informatiesoorten die in het uiteindelijke systeem moeten worden opgenomen.

3.3 De doelstelling van de opdracht

Het doel van deze opdracht is een web-based informatiesysteem te ontwikkelen dat het voor klanten en medewerkers mogelijk maakt incidenten te melden. Het informatiesysteem moet er vervolgens voor zorgen dat deze meldingen bij de juiste medewerkers terecht komen. Een medewerker die een melding in behandeling neemt moet in het informatiesysteem kunnen aangeven wat de status van deze melding is en welke stappen hij heeft ondernomen om de melding op te lossen. Het informatiesysteem moet het ook mogelijk maken verschillende soorten gegevens te beheren. Op lange termijn dient alle informatie, die momenteel nog verspreid ligt binnen de organisatie, in dit systeem te worden geïntegreerd. Medewerkers van de organisatie moeten via het internet toegang krijgen tot dit informatiesysteem zodat deze op elke locatie met een pc en internetverbinding voor hun beschikbaar is.

4. De aanpak van het project

De aanpak van dit project is beschreven doormiddel van een plan van aanpak. In dit plan van aanpak staat beschreven welke activiteiten uitgevoerd moesten worden, hoeveel tijd er voor deze activiteiten beschikbaar was en welke middelen er nodig waren om de activiteiten uit te kunnen voeren. Het plan van aanpak is opgesteld om voor de afstudeerder zelf inzicht te verschaffen in het uit te voeren project en om een leidraad te creëren voor het uitvoeren van het project. In dit hoofdstuk zal verteld worden welke middelen er gebruikt zijn bij het ontwikkelen van het incident managementsysteem en hoe de afstudeerder tot de keuze voor deze middelen is gekomen. Tevens zal in dit hoofdstuk de procesgang van de planning gedurende het project besproken worden.

4.1 De gebruikte middelen

Tijdens dit project zijn er van een aantal middelen gebruik gemaakt om de benodigde activiteiten uit te kunnen voeren die uiteindelijk tot de ontwikkeling het incident managementsysteem hebben geleid. Deze middelen zijn in dit document opgenomen om de lezer inzicht te verschaffen in de context waarbinnen dit project zich heeft afgespeeld. De gebruikte middelen die in deze paragraaf worden behandeld staan hieronder opgesomd:

- Ontwikkelomgevingen: PHP, HTML, MySQL, PHPMyAdmin
- (Ontwikkel)methoden: IAD, ITIL
- Ontwikkeltechnieken: UML, interviews

4.1.1 De gebruikte ontwikkelomgevingen

Voor het daadwerkelijk coderen van het systeem is gebruik gemaakt van PHP en HTML. PHP (PHP: Hypertext Preprocessor) is een taal waarmee een webbased systeem ontwikkeld kan worden en waarmee tussen bijvoorbeeld een website en een database kan worden gecommuniceerd. HTML (Hyper Text Markup Language) is ook een taal waarmee websites kunnen worden gebouwd, alleen een stuk meer gericht op de lay-out en niet zo zeer op de communicatie met de database. De keuze voor de scripttaal PHP was vrij snel gemaakt. De afstudeerder had tijdens eerdere projecten al enige kennis en ervaring opgedaan van PHP. Daarbij kwam dat PHP een gratis scriptaal is met een ruime technische ondersteuning via internet. Tevens moet gezegd worden dat het tijdens het project al vrij snel duidelijk was dat PHP de benodigde functionaliteit bood voor het ontwikkelen van het beoogde systeem. HTML was voor de afstudeerder de taal bij uitstek voor het ontwikkelen van de schetsen van de schermen, de lay-out en de navigatie van het systeem. Hij had hier immers al enige ervaring mee. Tevens wist de afstudeerder in dit stadium van het project al, dat met HTML alle grafische onderdelen van het systeem te ontwikkelen waren en was er de beschikking over een licentie van een HTML editor genaamd 'Dreamweaver'.

MySQL is een database die doormiddel van SQL queries gekoppeld kan worden aan een PHPscript. De keuze voor MySQL is wederom het feit dat de afstudeerder al eerder met deze database had gewerkt, hij gratis is en via internet technisch wordt ondersteund. PHPMyAdmin is een tool om de MySQL database mee te beheren. De reden dat er voor deze onderhoudstool was gekozen was omdat hij goed aansloot op de MySQL database en dat hij gratis was.

4.1.2 De gebruikte methoden

Iterative Application Design

Voor het uitvoeren van dit project is er gebruik gemaakt van een ontwikkelmethode. Als ontwikkelmethode is er gekozen voor IAD (Iterative Application Design). IAD is een methode die uit drie fases bestaat. Deze fases zijn:

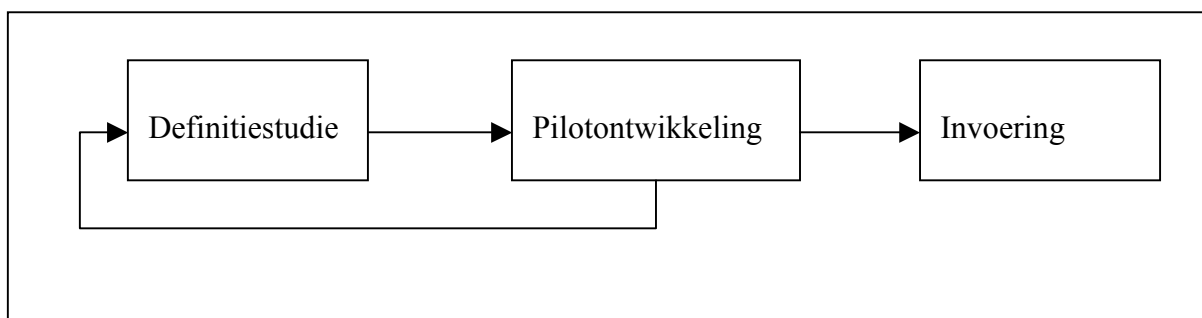
- Definitiestudie
- Pilotontwikkeling
- Invoering

In dit project is er vanwege een van tevoren ingecalculeerd gebrek aan tijd, besloten het systeem niet in te voeren. Dit was besloten in overeenstemming met de opdrachtgever zodat er meer functionaliteiten van het systeem ontwikkeld konden worden. De invoering zou eventueel buiten de formele afstudeerperiode worden uitgevoerd door de afstudeerder. Wel zou het systeem uitgebreid getest gaan worden voordat het aan de opdrachtgever zou worden overgedragen. Dit was belangrijk omdat er klanten met het systeem in aanraking zouden komen en er hierdoor grote waarde aan de correctheid en betrouwbaarheid van het systeem werd gehecht.

Er is voor dit project om verschillende redenen gekozen voor de ontwikkelmethode IAD. Deze redenen worden hieronder opgesomd:

- Het gebruik van IAD stelt de ontwikkelaar in staat de gebruikers en opdrachtgever nadrukkelijker bij het ontwikkelproces te betrekken. Dit gegeven zal er toe leiden dat er uiteindelijk een systeem zal worden opgeleverd dat nauwer aansluit op de wensen van de opdrachtgever. Aangezien alle toekomstige gebruikers regelmatig beschikbaar waren voor het geven van feedback op de ontwikkelde delen en zij dit zelfs wenselijk vonden, was deze manier van ontwikkelen zeer geschikt voor dit project.
- Bij de eerste gesprekken die de afstudeerder met de opdrachtgever had, bleek al snel dat concrete tussenresultaten (in de vorm van werken onderdelen van het systeem) in een relatief vroeg stadium van het project op prijs werden gesteld. Doordat IAD prototyping technieken bevat en het te ontwikkelen systeem in relatief snel te ontwikkelen pilots wordt opgedeeld, kon er aan deze wens worden voldaan.
- De afstudeerder had al behoorlijk wat ervaring met de toepassing van IAD door reeds eerder uitgevoerde projecten. Het toepassen van IAD zou hierdoor niet alleen een kwalitatief beter resultaat opleveren, in tegenstelling tot het toepassen van een vergelijkbare maar voor de afstudeerder onbekende ontwikkelstrategie, maar tevens behoorlijk wat tijd besparen omdat de afstudeerder zich dan geen nieuwe ontwikkelstrategie eigen hoefde te maken.

IAD kan volgens verschillende scenario's worden toegepast. Het scenario bepaalt hoe de verschillende fases worden doorlopen. Voor dit project was er gekozen om incrementeel te gaan ontwikkelen volgens het 'big bang scenario' (zie figuur 4.1).



Figuur 4.1 Iteratie strategie Big-Bang-Invoeren

Big-bang-invoeren is een ontwikkelstrategie waarin de eerste twee fases (definitiestudie en pilotontwikkeling) iteratief worden uitgevoerd, maar de laatste (invoering) niet. Er worden dus weliswaar diverse pilots na elkaar ontwikkeld, maar geen van die pilots wordt ingevoerd binnen de organisatie, dat gebeurt pas met de definitieve pilots. De fase invoering wordt dus slechts eenmalig uitgevoerd. De rede dat er juist voor dit scenario was gekozen, is het feit dat snelle ingevoerde resultaten niet nodig waren in dit project. Het te ontwikkelen systeem zou pas zinvol gebruikt kunnen worden, wanneer alle delen waren ingevoerd. Al in een vrij vroeg stadium van dit project werd in samenspraak met de opdrachtgever echter besloten dat de fase invoering niet zou worden uitgevoerd vanwege een gebrek aan tijd. Wel zou het systeem uitgebreid getest worden voordat het aan de opdrachtgever zou worden opgeleverd. Dit feit veroorzaakt geen drastische wijzigingen in ontwikkelscenario. De fases definitiestudie en pilotontwikkeling zouden gewoon iteratief doorlopen worden.

Information Technology Infrastructure Library

Een andere methode die voor dit project is gebruikt is ITIL. ITIL staat voor Information Technology Infrastructure Library. Het is een reeks publicaties van beste praktijkoplossingen op het gebied van beheer van IT. Met andere woorden: richtlijnen hoe de IT dienstverlener het best kan garanderen dat zijn klanten de producten en diensten krijgt die zij verlangen. Uitgangspunten zijn een systematische aanpak van het beheer en een procesmatige benadering van de bedrijfsorganisatie. ITIL is ontwikkeld door verschillende organisaties om het IT Service Management te analyseren en vervolgens de succesvolle aspecten eruit te halen die generiek bruikbaar zijn voor elke IT Service organisatie. Voor dit project zal er slechts één van de onderdelen uit ITIL gebruikt worden, namelijk incident management. Incident management heeft als doel de dreigende storingen in IT-diensten zo snel mogelijk op te lossen en ervoor te zorgen dat de gebruikers zo snel mogelijk weer aan het werk kunnen. Daarnaast heeft incident management de taak incidenten optimaal te registreren om het proces te kunnen meten en verbeteren en om er over te kunnen rapporteren aan de andere processen. De rede dat er voor gekozen is om incident management binnen Technico Automatisering te implementeren, was omdat dit de basis van ITIL vormt en het de weg openende voor de uiteindelijke implementatie van het volledige ITIL proces.

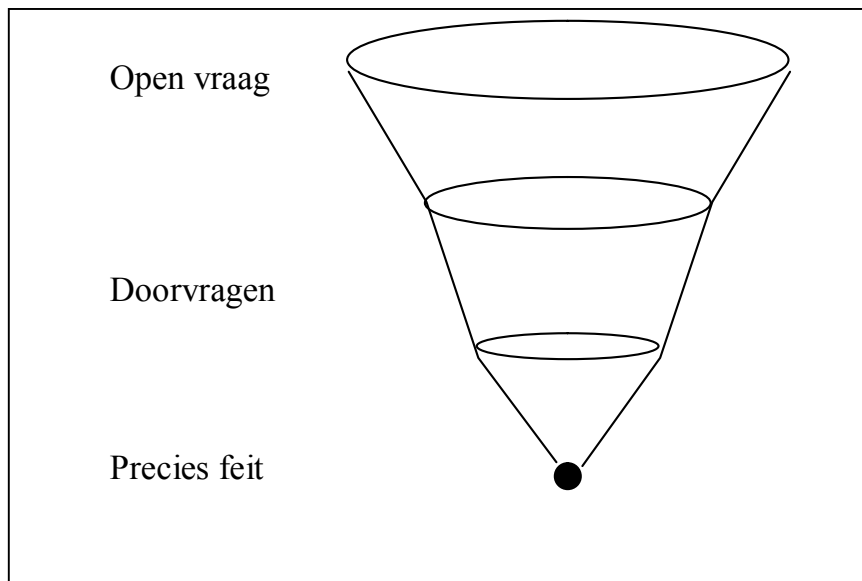
4.1.3 De gebruikte ontwikkeltechnieken

Unified Modeling Language

Als modelleringstechniek voor dit project is gekozen voor UML (Unified Modeling Language). UML bestaat uit een aantal modelleertechnieken zoals use cases, activiteitsdiagrammen, sequence diagrammen en klassediagrammen. Deze modelleertechnieken worden gebruikt tijdens de definitiestudie bij het opstellen van het systeemconcept. De reden dat de afstudeerder voor de ontwikkeltechniek heeft gekozen, is dat hij er al behoorlijk wat ervaring mee had opgedaan in eerdere projecten. Tevens was de afstudeerder van mening, dat alle beoogde conceptuele modellen met behulp van UML ontwikkeld konden worden.

Interviewtechniek

Om de opdracht duidelijk te krijgen, leek het interviewen van de opdrachtgever een geschikte techniek. Gedurende de opleiding is er een module geweest die interviewtechnieken heeft behandeld. Aan de hand van deze module is de interviewstrategie bepaald. Middels de 'fuikprocedure' zal er getracht worden gedetailleerde en concrete antwoorden te krijgen op de gestelde vragen. In de 'fuikprocedure' begint de interviewer met een aantal algemene open vragen waarop vervolgens wordt doorgevraagd totdat de opdrachtgever gedetailleerd genoeg heeft geantwoord. In figuur 4.2 is een schematische weergave te zien van de 'fuikprocedure'.



Figuur 4.2 Fuijck procedure

Een interview waarin de fuijckprocedure zijn nut bewees, was een interview betreft de systeemeisen. Het was in dit geval belangrijk de wensen van de opdrachtgever zeer concreet en meetbaar op papier te krijgen. De fuijckprocedure maakte dit mogelijk. In dit interview stelde de afstudeerder de vraag welke gegevens er aan een melding van een incident gekoppeld moesten kunnen worden. Door het stellen van deze vraag werd onder andere duidelijk dat de naam van een klant aan een incidentmelding gekoppeld moesten kunnen worden. Toen de afstudeerder vervolgens doorvroeg naar eventuele klantgegevens, werd duidelijk dat het mogelijk moest zijn verschillende gegevens van een klant met het systeem te kunnen registreren. Hierna was al vrij snel vastgesteld welke klantgegevens precies geregistreerd moesten kunnen worden.

4.2 De planning

Voordat er daadwerkelijk aan het project begonnen kon worden, diende er eerst een planning te worden opgesteld. Om een globaal overzicht te creëren van de werkzaamheden die voor dit project moesten worden verricht, is er eerst een globale planning opgesteld. De planning was opgesteld op basis van de opdrachtschrijving en een aantal oriënterende gesprekken die met de opdrachtgever waren gehouden. Een afbeelding van de laatste versie van deze globale planning is te zien in figuur 4.3.

Activiteit	Dagen
Opstellen plan van aanpak	5
Uitvoeren veranderingsanalyse	5
Definitiestudie	18
Specificeren plan van aanpak	1
Opstellen ontwikkelscenario	2
Bepalen systeemeisen	3
Opstellen systeemconcept	5
Opstellen technische structuur	2
Opstellen pilotplan	4
Verificatie en validatie	1
Pilotontwikkeling	42
Specificeren plan van aanpak	2
Opzetten testomgeving	2
Bepalen globaal functionele structuur	18
Bepalen globaal organisatorische inrichting	1
Opstellen pilotontwikkelplan	3
Bouwen bouweenheden	15
Verificatie en validatie	1
Testen	5

Figuur 4.3 Globale planning

Het was voor dit project om een aantal reden mogelijk realistische en concrete planningen op te stellen:

- De afstudeerder had al ervaring met het plannen van projecten door eerder uitgevoerde projecten.
- De afstudeerder had al eerder gewerkt met de ontwikkelmethodiek IAD en kon hier door bepalen welke onderdelen hij uit de methodiek kon gebruiken voor dit project en inschatten hoeveel tijd hij per onderdeel kwijt zou zijn.
- De afstudeerder had de globale planning met diverse mensen besproken onder andere met zijn examinerator en een aantal medestudenten.
- Door de planning regelmatig aan te passen en bij te werken, bleef deze constant actueel en realistisch.

Nadat de eerste versie van het globale plan van aanpak naar tevredenheid was opgesteld, kon er worden begonnen met het opstellen van de detailplanning. De eerste detailplanning die was opgesteld bestond uit een gedetailleerd overzicht van de activiteiten die per dag zouden worden uitgevoerd over de gehele afstudeerperiode. Na een aantal weken werd het duidelijk dat deze manier van plannen vrij onpraktisch was. De afstudeerder ervaaarde dat het lastig was om voor een periode van 75 dagen per dag de uit te voeren activiteiten te bepalen. Dit zou betekenen dat wanneer hij op een bepaald moment een activiteit niet kon uitvoeren, de hele planning aangepast moest worden, door alle activiteiten en dag opzij te schuiven. In de ogen van de afstudeerder waren er twee mogelijke alternatieven:

- Op basis van de globale planning kon een gedetailleerdere weekplanning worden opgesteld. Het voordeel hiervan was dat het ruimte bood voor flexibiliteit. Het nadeel was daarentegen het verlies aan gedetailleerde structuur.
- Op basis van de globale planning kon een zeer gedetailleerde dagplanning worden opgesteld maar dan per onderdeel (per fase en iteratie) en pas aan het begin van elk onderdeel. Het voordeel hiervan was dat het zowel veel structuur als flexibiliteit bood. Het nadeel daarentegen was echter dat de gedetailleerde structuur pas beschikbaar was bij de start van elk onderdeel.

Vanwege de omvang van dit project hechtte de afstudeerder veel waarde aan een concrete structuur in de werkzaamheden. Om deze rede koos hij dan ook om per onderdeel (het tweede

alternatief) te gaan plannen. Om toch al een concreet overzicht van het project te creëren, werd er een tussenoplossing bedacht. Op basis van de globale planning zijn er per iteratie en per onderdeel, de uit te voeren activiteiten bepaald. Tevens is per onderdeel het aantal dagen bepaald. In figuur 4.4 is een fragment uit de planning te zien die deze tussenoplossing heeft opgeleverd.

Eerste iteratie	
• Definitiestudie (8 dagen)	
o Plan van aanpak	
o Ontwikkelscenario	
o Opzetten webserver	
o Prototype	
o Systeemeisen	
o Systemconcept	
o Technische structuur	
o Organisatorische inrichting	
o Pilotplan	
o Verificatie en validatie	
• Pilot 1: database (7 dagen)	
o Plan van aanpak	
o Functionele structuur	
o Pilotontwikkeling	
o Verificatie en validatie	
• Bouw database	

Figuur 4.4 Tussenoplossing planning

De volledig planning uit figuur 4.4. is te zien in interne bijlage twee, het plan van aanpak. Op basis van deze planning kon bij de start van elk onderdeel, een gedetailleerde planning voor dat onderdeel worden opgesteld. In figuur 4.5 is de detailplanning van de definitiestudie te zien in de eerste iteratie. Zoals in het figuur te zien is, worden per dag de uit te voeren activiteiten beschreven, het aantal uur dat per activiteit beschikbaar is en de betreffende iteratie waarin het onderdeel zich bevindt. De detailplanningen zijn te vinden in de rapporten van de verschillende onderdelen welke als externe bijlagen aan dit document zijn toegevoegd.

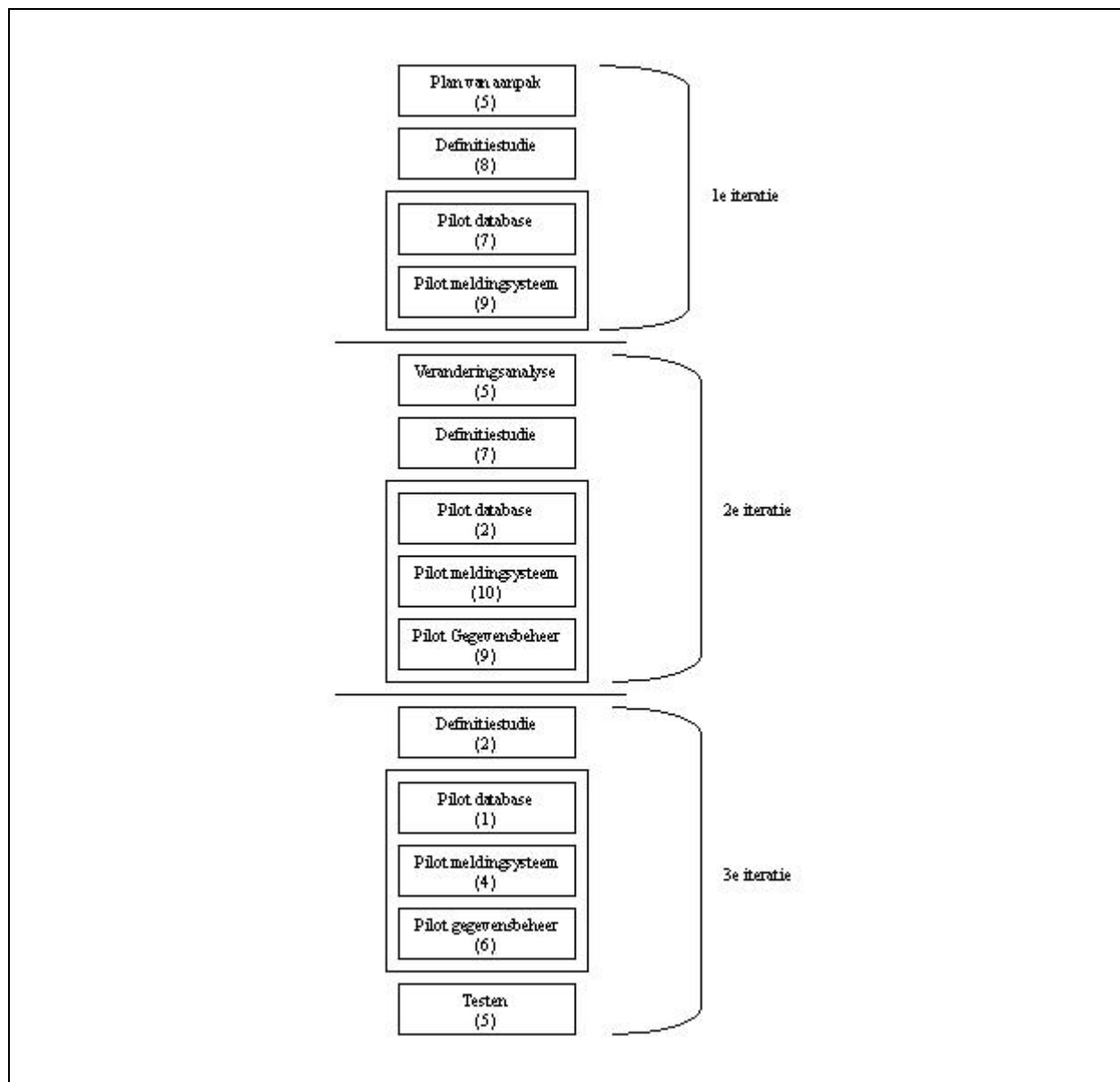
Week 2	Onderdeel	Activiteit	# uur	Iteratie
24-05-04	PVA/ ontwikkelscenario	<ul style="list-style-type: none"> • Specificeren plan van aanpak • Opstellen ontwikkelscenario 	2 6	1
25-05-04	Testomgeving	<ul style="list-style-type: none"> • Bestuderen documentatie • Installeren software 	2 6	1
26-05-04	Testomgeving/ prototype	<ul style="list-style-type: none"> • Operationeel maken testomgeving • Ontwikkelen prototype 	8	1
27-05-04	Prototype/ systeemeisen	<ul style="list-style-type: none"> • Ontwikkelen prototype • Bepalen systeemeisen 	8	1
28-05-04	Systeemconcept	<ul style="list-style-type: none"> • Opstellen systeemconcept • Vast contactmoment 	7 1	1
Week 3				
31-05-04	Systeemconcept	<ul style="list-style-type: none"> • Opstellen use cases • Opstellen use case diagram 	6 2	1
01-06-04	Tweede pinksterdag	-		-
02-06-04	Systeemconcept/ organisatorische inrichting/ technische structuur	<ul style="list-style-type: none"> • Opstellen klassediagram • Opstellen organisatorische inrichting • Opstellen technische structuur 	5 2 1	1
03-06-04	pilotplan/ verificatie/ validatie	<ul style="list-style-type: none"> • Opstellen pilotplan • Verificatie en validatie 	6 2	1

Figuur 4.5 Detailplanning

In de beginfase van het project, was het al bekend dat het systeem doormiddel van twee iteratieslagen ontwikkeld zou gaan worden. De verantwoording van deze keuze is te vinden in paragraaf 6.2 ontwikkelscenario. In de achtste week van het project kreeg de afstudeerder bezoek van zijn examinatoren. Tijdens dit gesprek werd duidelijk dat er een aantal zaken beter anders aangepakt konden worden. Deze zaken waren:

- Er diende een veranderingsanalyse opgesteld te worden om op een gestructureerde wijze, de beste mogelijke oplossing voor het aanwezige probleem te achterhalen
- ITIL diende in het ontwerp van het systeem opgenomen te worden om een logische structuur in het te ontwikkelen incident management systeem aan te kunnen brengen.

Deze twee besluiten leidde ertoe dat de planning moest worden herzien. Op het moment van dit gesprek was de afstudeerder al bezig met het functionele ontwerp van de tweede pilot, het meldingsysteem. Er werd besloten dat alle verrichtte werkzaamheden tot aan het gesprek als eerste iteratie beschouwd zouden worden. In deze iteratie waren er een aantal onderdelen gerealiseerd, dit waren de definitiestudie, pilot één de database en pilot twee het meldingsysteem (tot en met het functioneel ontwerp). Omdat deze eerste iteratie niet volledig was afgerond, werd tevens besloten dat dit eigenlijk maar een halve iteratie was. Na het gesprek werden twee nieuwe iteraties ingepland. De eerste hiervan zou vooraf worden gegaan door een veranderingsanalyse. Een afbeelding van de voorgaande geschetste situatie is te zien in figuur 4.6.



Figuur 4.6 Doorlopen iteraties

5. De veranderingsanalyse

In dit hoofdstuk wordt de procesgang van de veranderingsanalyse beschreven. Het doel van de veranderingsanalyse is het achterhalen van de best mogelijke oplossing voor een probleem dat binnen de organisatie aanwezig is. Doormiddel van het vaststellen van de huidige situatie en het bepalen van een aantal kenmerken van de organisatie wordt er een helder beeld gevormd van hoe het er binnen de organisatie aan toegaat op het gebied van incident management. Vervolgens moet er worden bepaald wat nu precies het probleem is en welke oplossingen er op het probleem kunnen worden toegepast. Tot slot zal er doormiddel van een selectieprocedure één oplossing worden uitkozen en worden ontwikkeld. Omdat bij de aanvang van het afstuderen het op te lossen probleem nog niet geheel duidelijk was in de ogen van zowel de opdrachtgever als de afstudeerder, werd er besloten een veranderingsanalyse uit te voeren. Om de veranderingsanalyse te kunnen opstellen, waren er verschillende interviews met de opdrachtgever en toekomstige gebruikers vereist. De afstudeerder was van mening dat de beschikbare documentatie vanuit de opleiding betreft het opstellen van een veranderingsanalyse hem weinig houvast bood wat betreft het opstellen ervan. Tevens had de afstudeerder nog weinig praktische ervaring met het opstellen van een veranderingsanalyse. Dit leidde tot het besluit een indeling te gebruiken die wordt voorgeschreven in het boek Organisatie, besturing en informatie door Eleanor Pascoe-Samson. Zij beschrijft in haar boek dat een veranderingsanalyse bestaat uit een verkenningsfase, ontwerpfase en keuzefase. In de volgende paragrafen zal per fase van de veranderingsanalyse beschreven worden hoe het onderdeel precies tot stand is gekomen.

5.1 De verkenningsfase

In de verkenningsfase wordt er getracht een duidelijk beeld te krijgen van de 'echte' problemen die zich binnen de organisatie afspelen. Tevens worden in deze fase de technische, culturele en machts- en politieke aspecten vastgesteld. Tot slot worden de doelen, wensen en eisen die mensen hebben ten aanzien van het probleem geïnventariseerd. Nadat de voorgaande zaken waren vastgesteld, kon de huidige situatie op basis van problemen en doelstellingen beoordeeld worden en kon tevens de veranderingsbehoefte worden vastgesteld.

De verkenningsfase omvat vier stappen:

- Probleeminventarisatie en analyse
- Beschrijving van de huidige activiteiten
- Bepaling van de doelstellingen
- Beoordeling van de huidige situatie

In de volgende subparagrafen zal per stap de procesgang beschreven worden.

5.1.1 Probleeminventarisatie en analyse

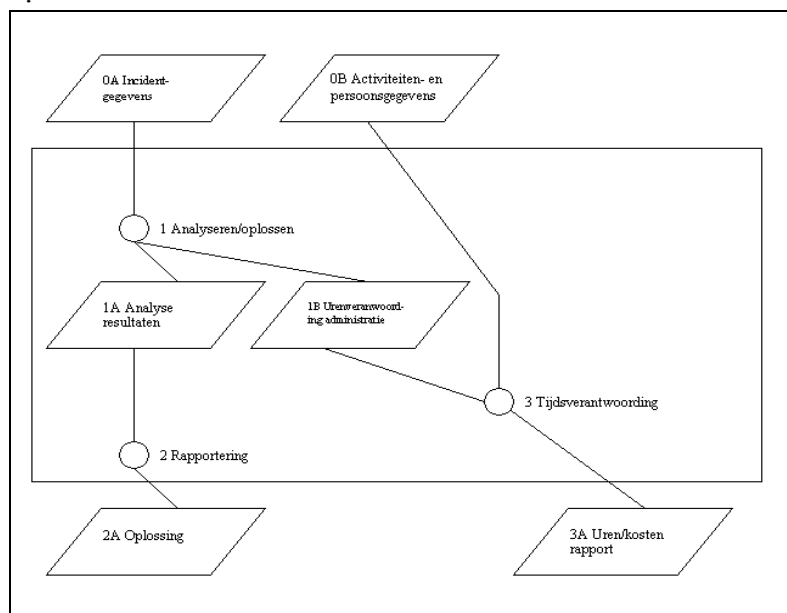
Voorafgaand aan het opstellen van de veranderingsanalyse was er een interview met de opdrachtgever gehouden. Het doel van dit interview was te achterhalen hoe de huidige manier van 'incident management' binnen de organisatie werd ervaren. De uitwerking van dit interview is te vinden in interne bijlage drie van dit document. Aangezien de opdrachtgever werkzaam was in dit proces en het meest toegankelijk was voor de afstudeerder, was hij de juiste persoon om te interviewen. Uit het interview bleek dat het hoofdprobleem was, dat de manier waarop incidentmeldingen werden afgehandeld, door de opdrachtgever en diverse belanghebbenden als onvoldoende werd beschouwd. Deze ontdekking leidde tot de volgende deelproblemen:

- Meldingen van incidenten worden niet structureel geregistreerd
- Meldingen van incidenten worden niet structureel opgelost
- De organisatie beschikt over een slecht overzicht van de frequentie, aard en andere kenmerken van de incidentmeldingen

Uit de bovengenoemde problemen werd de conclusie getrokken dat er binnen de organisatie geen procedure aanwezig was voor het registreren en afhandelen van incidentmeldingen. Tevens werd het duidelijk, dat er binnen de organisatie hiervoor geen ondersteunend informatiesysteem aanwezig was.

5.1.2 Vaststellen huidige activiteiten

Uit de gesprekken en interviews die gedurende deze fase met de opdrachtgever en werknemers die zich bezighielden met het afhandelen van incidenten waren gehouden, beschikte de afstudeerder al vrij snel over een helder beeld van het incident afhandelingproces zoals zij op de huidige manier werd uitgevoerd. Door het opstellen van een activiteitendiagram, konden de huidige activiteiten betreft dit proces eenduidig in overstemming met de opdrachtgever en overige belanghebbende, worden vastgesteld. In figuur 5.1 is het activiteitendiagram te zien.



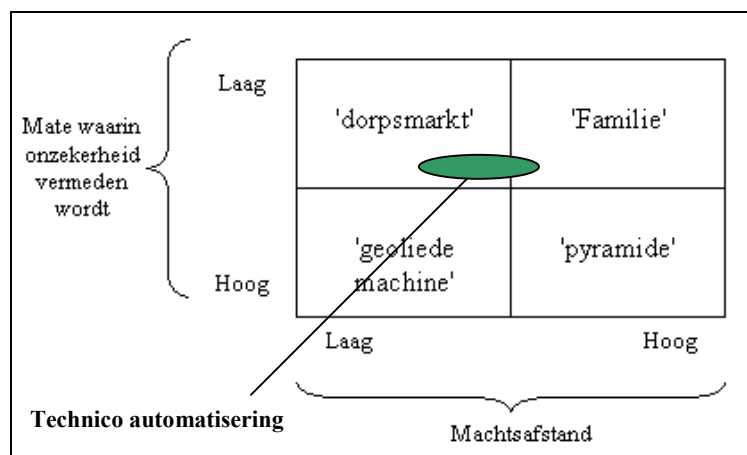
Figuur 5.1 Activiteitendiagram incident afhandelingproces

In dit onderdeel zijn tevens cultuurdimensies en de organisatiestructuur vastgesteld. De cultuurdimensies zijn opgesteld aan de hand van de cultuurdimensies van Hofstede. Voor het welslagen van de uit te voeren opdracht was het belangrijk om te weten welke cultuurdimensies er op deze organisatie van toepassing waren. Om dit te illustreren met een voorbeeld, het volgende: *Het is zinloos zeer strakke procedures in een organisatie te implementeren wanneer de dimensie 'klantoriëntatie', en dus de klantenwensen belangrijker zijn dan procedures, van toepassing is op de organisatie.* Het was voor de afstudeerder belangrijk om te weten wat de organisatiecultuur van Technico Automatisering was om zo te kunnen bepalen hoe incident management het best binnen de organisatie geïmplementeerd kon worden. De cultuurdimensies van Hofstede boden hierbij uitkomst. Het feit dat de afstudeerder al in eerdere projecten kennis had gemaakt met de cultuurdimensies van Hofstede, droeg ook bij aan deze keuze. Het resultaat van deze activiteit was enerzijds behaald door de inzichten die afstudeerder zelf gedurende het project van de organisatie heeft opgedaan. Anderzijds heeft de input van de opdrachtgever enorm bijgedragen bij vaststellen van de organisatiecultuur. Aangezien de opdrachtgever pas een korte tijd actief was bij de organisatie, kon hij hier een objectief oordeel over vellen. Twee van de dimensies die van toepassing zijn op Technico Automatisering staan hieronder opgesomd:

- Resultaatgericht; de organisatie voelt zich op zijn gemak in onbekende situaties, iedereen doet zijn uiterste best en elke dag wordt als een nieuwe uitdaging beschouwd.
- Klantoriëntatie; de organisatie komt te gemoed aan klantenwensen en vindt resultaten belangrijker dan procedures. De organisatie heeft een pragmatische houding tegenover ethiek.

Voor een complete lijst van de betreffende cultuurdimensies wordt doorverwezen naar externe bijlage vijf, de veranderingsanalyse. Uit het voorgaande kon de afstudeerder concluderen dat strakke procedures omtrent het hele proces van incidentmeldingen, niet binnen de organisatie geaccepteerd zouden worden. Dit bleek onder andere uit het feit dat de organisatie zeer resultaatgericht en klantgeoriënteerd te werk ging. Veel beter zou een geautomatiseerd systeem zijn dat de medewerkers ondersteunde en begeleidde bij het registreren en afhandelen van incidentmeldingen, gebaseerd op een logisch proces.

Voor het vaststellen van de organisatiestructuur is wederom dankbaar gebruik gemaakt van de manier die Hofstede voorschrijft. Hofstede onderscheidt vier organisatietype, gebaseerd op de culturele waarden van de mensen. De machtsafstand en de mate waarin onzekerheid wordt vermeden spelen in deze verdeling een belangrijke rol. Zoals in figuur 5.2 te zien is maakt Hofstede onderscheid tussen vier organisatietypen: dorpsmarkt, familie, geoliede machine en piramide.



Figuur 5.2 Organisatietypen van Hofstede

Op deze organisatie zijn twee organisatietype van toepassing, de dorpsmarkt en de familie. Omdat regels en hiërarchie binnen deze organisatie geen dominerende rol spelen, kan hij geclassificeerd worden als een dorpsmarkt. Echter, de directeur dient wel geraadpleegd te worden wanneer er belangrijke besluiten dienen te worden genomen. Tevens is de directeur verantwoordelijk voor het bepalen van het tactisch en strategisch beleid. Deze kenmerken tonen duidelijke overeenkomsten met een familie. Uit het voorgaande kon de afstudeerder wederom weer concluderen dat strakke procedures geen onderdeel zouden gaan vormen van de oplossing (dorpsmarkt). Tevens kon geconcludeerd worden dat wanneer de oplossing grote organisatorische gevolgen zou hebben, de directeur hiervoor geraadpleegd diende te worden (familie). Voor kleinere besluiten hoefde de directeur echter niet geraadpleegd te worden, wat de afstudeerder en opdrachtgever enige vrijheid gaf wat betreft de invulling en uitvoering van de opdracht.

5.1.3 Bepalen doelstellingen

Nadat de afstudeerder zich een duidelijk beeld had gevormd van de organisatie en haar op verschillende gebieden had geclassificeerd, kon er begonnen worden met het bepalen van de doelstellingen. Uit de in paragraaf 5.1.1 geïnventariseerde problemen kon een probleemgebied worden bepaald. Het probleemgebied bestond uit het gehele proces omtrent het registreren en afhandelen van incidentmelding. Voor achterhalen van de doelstellingen die gesteld diende te worden om de problemen op te lossen waren de volgende vragen aan verschillende belanghebbende gesteld:

- Welke rol zou het probleemgebied moeten spelen binnen de organisatie?
- Welke belangrijke veranderingen worden de komende twee, vijf of misschien tien jaar verwacht die van invloed zijn op het probleemgebied?
- Welke van deze veranderingen zullen de meeste invloed hebben op het probleemgebied?

- Met welke aspecten moet rekening worden gehouden bij veranderingen in het probleemgebied?

De antwoorden op de bovengenoemde vragen leidde tot de volgende concrete doelstellingen:

- Incidentmeldingen moeten structureert geregistreerd worden
- Oplossingen moeten geregistreerd worden
- Incidentmeldingen moet op een gestructureerde manier afgehandeld worden
- Gegevens betreffende incidenten moeten binnen de organisatie gedeeld worden

5.2 De ontwerpfase

In de ontwerpfase zijn op basis van de in de verkenningsfase opgestelde doelstellingen, oplossingen geformuleerd. De gevonden oplossingen zijn als volgt:

- Er moet een informatiesysteem worden ontwikkeld ten behoeve van incident management binnen de organisatie
- Het informatiesysteem moet het mogelijk maken incidentmeldingen te registreren
- Het informatiesysteem moet op een gestructureerde manier ondersteuning bieden bij het afhandelen van een incidentmelding
- Het informatiesysteem moet inzichten geven in zaken zoals de frequentie en de aard van incidentmeldingen
- De afdeling 'support' blijft verantwoordelijk voor het afhandelen van de incidentmeldingen. Zij zullen zowel de eerste als de tweede lijn vormen binnen het incidentmanagement proces.
- Het incident managementproces zal gebaseerd worden op het incident management zoals door ITIL wordt voorgeschreven.

De bovenstaande oplossingen moeten gezien worden als één grote oplossing verdeeld in meerdere deeloplossingen. Kortom kan gezegd worden, dat er een informatiesysteem ontwikkeld moest worden, dat incidentmanagement binnen de organisatie mogelijk zou maken.

Al voor de aanvang van deze fase waren er al een aantal alternatieven besproken. Eén van die alternatieven was de eerste lijn van het incident managementproces naar te leggen bij de administratief medewerkers. De supportafdeling zou zich in deze situatie beter kunnen richten op de echte relevante problemen. Onder andere vanwege sterke klantenbindingen van de support-medewerkers, werd dit idee verworpen.

5.3 De keuzefase

In de keuzefase zal de gekozen oplossing op basis van de in de verkenningsfase opgestelde doelstellingen worden geformuleerd.

Zoals uit de vorige paragraaf blijkt, heeft de ontwerpfase tot één oplossing geleid. In overeenstemming met de opdrachtgever is bepaald dat dit niet alleen de beste oplossing is, maar ook vastgesteld dat hij aan alle 'stoepfactoren' voldoet. Voorbeelden van stoepfactoren waaraan de oplossing moet voldoen zijn:

- Sociaal acceptabel; de huidige werknemers die te maken zullen krijgen met de toekomstige situatie, zijn akkoord gegaan met het oplossingsvoorstel.
- Technisch uitvoer; enerzijds beschikt de afstudeerder over de kennis en vaardigheden voor het realiseren van deze oplossing. Anderzijds beschikt de organisatie over de tijd en middelen voor het realiseren van deze oplossing.

Voor een complete lijst van stoepfactoren wordt doorverwezen naar externe bijlage vijf, de veranderingsanalyse.

Tot slot dient er in de keuzefase te worden nagedacht over de voor- en nadelen van de gekozen oplossing. In overeenstemming met de opdrachtgever zijn de voor- en nadelen bepaald. Hieronder volgt een voorbeeld van zowel een voordeel als een nadeel:

- Voordeel: Incidentmeldingen kunnen gestructureerder worden opgelost waardoor ze ook sneller kunnen worden opgelost en efficiënter verdeeld kunnen worden over de support-medewerkers.
- Nadeel: De service naar de klanten toe zal door de implementatie van deze oplossing enigszins veranderen. Het is mogelijk dat hierdoor een bepaalde weerstand ontstaat tegen de nieuwe situatie.

Nadat was besloten dan de nadelen niet opwogen tegen de voordelen die deze oplossing met zich mee bracht, kon worden begonnen aan de definitiestudie.

6. De definitiestudie

In dit hoofdstuk wordt de procesgang van de definitiestudie beschreven. In de definitiestudie worden de doelen van het systeem geanalyseerd evenals de beperkingen. Dit hoofdstuk is van belang omdat het de lezer inzicht biedt in het proces tijdens deze fase van het proces. De definitiestudie is van belang voor het project om te kunnen specificeren wat het te ontwikkelen systeem moet kunnen. Dit is erg belangrijk omdat het inzicht biedt in de aanpak voor de rest van het project. De procesgang van de definitiestudie wordt beschreven aan de hand van het plan van aanpak, het ontwikkelscenario, de systeemeisen, het systeemconcept, de technische structuur, de organisatorische inrichting, het pilotplan en tot slot de verificatie en validatie. Tevens is er gedurende deze fase een prototype ontwikkeld om samen met de opdrachtgever en overige belanghebbende de systeemeisen helder te krijgen. Per activiteit wordt er beschreven wat er gedaan moest worden, waarom dit gedaan moest worden, wat de aanpak was, de problemen die daarbij ontstonden en de oplossingen die daarvoor werden bedacht.

6.1 Specificeren plan van aanpak definitiestudie

In het plan van aanpak staat gespecificeerd welke activiteiten er in deze fase moesten worden uitgevoerd, uit welke onderdelen de definitiestudie zou gaan bestaan en hoeveel tijd deze onderdelen in beslag zouden gaan nemen. Het plan van aanpak voor de definitiestudie bepaalde de aanpak gedurende deze fase. Het belangrijkste onderdeel uit het plan van aanpak is de planning. De planning hielp structuur te geven aan de werkzaamheden en sloot vertraging en het niet halen van deadlines zo veel mogelijk uit. Deze planning is opgesteld aan de hand van de in paragraaf 4.2 genoemde tussenoplossing.

6.2 Definiëren ontwikkelscenario

In dit onderdeel wordt de doorlopen procesgang van het ontwikkelscenario beschreven. Het ontwikkelscenario is een adequate manier om na te kunnen denken over de pilots waaruit het systeem moet gaan bestaan. Het hoofddoel van deze activiteit is de reikwijdte en de context van het ontwikkelproject vast te stellen. In het ontwikkelscenario wordt bepaald met welke strategie het informatiesysteem ontwikkeld zal gaan worden. Aan de hand van deze strategie zal bepaald worden uit welke pilots het systeem zal gaan bestaan en hoe de pilots verder gepland kunnen worden.

Na het opstellen van de veranderingsanalyse had de afstudeerder al een behoorlijk goed beeld van hoe het systeem eruit moest komen te zien. Er werd bepaald dat de pilotverdeling er grofweg als volgt zou uit te komen zien:

- Pilot 1: Database
- Pilot 2: Meldingssysteem
- Pilot 3: Gegevensbeheer

De eerste pilot zou gaan bestaan uit de database. De reden dat database als eerste pilot werd gekozen, was het feit dat de database de basis zou gaan vormen voor het gehele te ontwikkelen systeem. Omdat de pilot database geen functionele subset was van het systeem, werd hij als een 'kernelpilot' beschouwd, of te wel een niet functionele subset van het systeem. De tweede pilot zou 'meldingssysteem' gaan heten. In deze pilot zouden alle functionaliteiten voor het registreren, wijzigen en verwijderen van incidentmeldingen ondergebracht worden. De reden dat deze pilot als tweede pilot is gekozen, was omdat deze functionaliteiten als de primaire functionaliteiten van het systeem werden beschouwd. Tot slot de laatste pilot, 'gegevensbeheer'. Hier werden alle functionaliteiten voor het onderhouden van de gegevens die aan een incidentmelding konden worden gekoppeld in ondergebracht.

De strategie die voor dit project is gebruikt, is 'big-bang-invoeren'. Big-bang-invoeren houdt in dat de eerste twee fasen, de definitiestudie en pilotontwikkeling, iteratief worden uitgevoerd. De fase

invoering wordt echter maar één keer uitgevoerd. De drie pilots worden binnen de pilot-ontwikkeling achtereenvolgens ontwikkeld. De reden dat er juist voor deze strategie is gekozen, was het feit dat snelle ingevoerde resultaten niet nodig waren in dit project. Het te ontwikkelen systeem zou pas zinvol gebruikt kunnen worden, wanneer alle delen waren ingevoerd. Al in een vrij vroeg stadium van dit project werd in samenspraak met de opdrachtgever echter besloten dat de fase invoering niet zou worden uitgevoerd vanwege een gebrek aan tijd.

De iteratiestrategie brengt een aantal iteraties met zich mee. Voor dit project is er gekozen voor twee iteratieslagen. De keuze voor twee iteratieslagen had te maken met het de beschikbare tijd voor dit project (te weten 75 dagen). Tevens was de afstudeerder van mening dat het systeem middels twee iteraties gerealiseerd kon worden. Alle drie de pilots zouden in totaal twee iteraties gaan doorlopen.

6.3 Definiëren systeemeisen

In dit onderdeel zal de gevolgde procesgang bij het definiëren van de systeemeisen worden beschreven. Er worden twee soorten systeemeisen onderkend, te weten: Functionele en non-functionele systeemeisen. Het doel van de systeemeisen is een eenduidige overeenkomst te bereiken met de opdrachtgever wat betreft de functionele en non-functionele systeemeisen. De functionele systeemeisen zijn vastgesteld aan de hand van interviews met de opdrachtgever/bedrijfsmentor, de gevonden oplossingen in de veranderingsanalyse en aan de hand van een ontwikkeld prototype. De non-functionele systeemeisen zijn uit de functionele systeemeisen voortgevloeid en bestaan uit interface-eisen, integriteiteisen, performance eisen en operationele eisen.

6.3.1 Ontwikkelen prototype

Om op een relatief snelle manier overeenstemming te bereiken met de opdrachtgever wat betreft de systeemeisen, was er op basis van de opdrachtoomschrijving en een aantal gevoerde gesprekken en interviews, een prototype ontwikkeld. Dit prototype bestond uit een aantal schermen met daarin een aantal voorbeeldwaardes en ertussen verschillende koppelingen. Telkens wanneer in het beginstadium van het project de systeemeisen ter sprake kwamen, werd het prototype erbij gepakt. De ontvangen feedback werd vervolgens zowel in de systeemeisen als in het prototype verwerkt. Het voordeel van deze werkwijze was dat de functionaliteiten die het uiteindelijke systeem moest gaan bieden al in een vrij vroeg stadium duidelijk waren voor zowel de opdrachtgever als de afstudeerder. In figuur 6.1 is een screenshot te zien van één van de schermen van de laatste versie van het prototype.

Technico b.v. meldingssysteem										
Nieuwe meldingen:										
	Soort melding	Meldings nummer	Prioriteit	Datum	Relatie	Applicatie	Versienr	Screenshot	Omschrijving	Status
Bekijk Verwijder Naar hist.	Storing	123	Hoog	19/05/04	Snavelenburg	Technicus	1.2	Afbeelding1	de applicatie start niet meer op	
Bekijk Verwijder Naar hist.	Klacht	122	Gemiddeld							
Bekijk Verwijder Naar hist.										
Oude meldingen:										
	Soort melding	Meldings nummer	Prioriteit	Datum	Relatie	Applicatie	Versienr	Screenshot	Omschrijving	Status
Bekijk Verwijder Naar hist.	Storing	123	Hoog	19/05/04	Snavelenburg	Technicus	1.2	Afbeelding1	de applicatie start niet meer op	
Bekijk Verwijder Naar hist.	Klacht	122	Gemiddeld							
Bekijk Verwijder Naar hist.										

Figuur 6.1 Scherm uit prototype

In dit screenshot is een tabel te zien met daarin de gegevens van alle nieuwe en oude incidentmeldingen. Vanuit dit scherm moest het mogelijk zijn een incidentmelding in detail te bekijken, te verwijderen en naar historie te verplaatsen. Al hoewel het uiteindelijke systeem er behoorlijk anders is uit komen te zien, heeft het prototype zijn nut bij het opstellen van de systeemeisen wel degelijk bewezen. Dankzij het opstellen van het prototype, zijn de systeemeisen gedurende het project geen enkele keer echt meer drastisch gewijzigd. Een aantal onderdelen van de lay-out zijn zelfs gebruikt in het uiteindelijk ontwikkelde systeem. Hoewel de screenshots van het ontwikkelde systeem pas in hoofdstuk 7 (de pilotontwikkeling) aan bod komen, volgt in figuur 6.2 alvast een voorproefje, om het verschil met het prototype te demonstreren

TECHNICO ■ ■ ■ Incident management systeem Automatisering B.V. ■ ■ ■									
Alle meldingen									
Dit zijn alle meldingen.									
Status	Maak selectie	Ok	Relatie	Maak selectie	Ok	Verantw.	Maak selectie	Ok	
Nummer	Datum	Relatie	Categorie	Soort	Prioriteit	Verantw.	Iijn	Status	
30	240904	test	Software-fout	Klacht	Middel	eelke	Tweede	In_behandeling	
29	240904	test	Hardware-fout	Klacht	Laag	ronald	Eerste	Niet_in_behandeling	
28	240904	test	Hardware-fout	Klacht	Laag	ronald	Tweede	Niet_in_behandeling	
27	240904	test	Software-fout	Klacht	Laag	ronald	Derde	Niet_in_behandeling	
26	240904	-	-	-	-	-	-	-	
25	240904	-	-	-	-	-	-	-	
24	240904	-	-	-	-	-	-	-	
23	240904	-	-	-	-	-	-	-	
22	240904	-	-	-	-	-	-	-	
21	240904	-	-	-	-	-	-	-	
20	240904	-	-	-	-	-	-	-	
19	240904	-	-	-	-	-	-	-	
18	240904	-	-	-	-	-	-	-	
17	240904	-	-	-	-	-	-	-	

Figuur 6.2 Globaal overzicht incidentmeldingen

6.3.2 Bepalen functionele systeemeisen

Zoals in de vorige paragraaf al verteld was, waren de functionele systeemeisen voor een groot deel onttrokken aan het in de definitiestudie ontwikkelde prototype. Hieronder volgt een fragment uit de systeemeisen. De hele lijst van systeemeisen kan worden ingezien in de externe bijlage één, de definitiestudie.

1. Medewerker moet een incidentmelding kunnen maken of afhandelen door:
 - De externe status van de incidentmelding te kunnen wijzigen
 - De interne status van een incidentmelding te kunnen wijzigen
 - Een incidentmelding aan een categorie kunnen toekennen
 - Aan te kunnen geven welke acties hij heeft ondernomen om de incidentmelding op te lossen.
 - Één of meerdere werknemers aan de incidentmelding te kunnen koppelen
 - De gebruikte middelen te kunnen registreren
 - Te kunnen registreren of het incidentmelding in overeenstemming is met de SLA
 - Een oplossing te kunnen registreren
 - De oplostijd te kunnen registreren
 - Een incidentmelding aan een probleem te koppelen
 - Een algemene notitie aan een meldingmelding te kunnen koppelen
 - De omschrijving te kunnen wijzigen
 - De betrokken randapparatuur te kunnen wijzigen
 - De betrokken applicatie te kunnen wijzigen
2. Medewerker moet van zowel een globaal overzicht van alle incidentmeldingen kunnen opvragen als een enkele melding in detail bekijken
3. Beheerder moet de volgende gegevensbronnen volledig kunnen beheeren (toevoegen, verwijderen, wijzigen)
 1. Relaties
 2. Randapparatuur
 3. Software
 4. Te gebruiken middelen
 5. Categorieën
 6. Oplossingen
 7. Systeemgebruikers

De systeemeisen zijn gedurende het project regelmatig gewijzigd. Gedurende het ontwikkelen van de drie pilots werd er een lijst bijgehouden van functionaliteiten die in de opvolgende iteratie gewijzigd of juist toegevoegd moesten worden. De systeemeisen werden enerzijds veroorzaakt door logische ontwerpfouten (bijvoorbeeld oplossingen die je in een database invoert, dienen ook in een overzicht aan de gebruiker gepresenteerd te kunnen worden). Anderzijds werden de wensen en eisen van de opdrachtgever ook steeds concreter en specifiek. Dit bleek onder andere uit het feit dat de opdrachtgever het voorstel deed, de gebruikersrechten in een aparte tabel in de database onder te brengen.

6.4 Bepalen systeemconcept

In dit onderdeel wordt de doorlopen procesgang van het systeemconcept beschreven. Het systeemconcept vormt een eenduidige basis voor het verder ontwikkelen van het systeem en is tevens conceptueel genoeg om aan de opdrachtgever en gebruikers te kunnen rapporteren. Het systeemconcept wordt beschreven aan de hand van use cases, een use case diagram en een klassediagram. De use cases en het use case diagram zijn ontwikkeld ten behoeve van de pilots 'meldingsysteem' en 'gegevensbeheer'. Het klassediagram is opgesteld ten behoeve van de pilot 'database'.

6.4.1 Ontwikkelen use cases

Use cases zijn een middel om de functionele eisen die gesteld worden aan het informatiesysteem weer te geven. Ze gaan uit van de gebruiker van het systeem en beschrijven hoe hij met het systeem om wil gaan. De use cases zijn opgesteld aan de hand van de systeemeisen. Sommige use cases bestaan uit een enkele systeemeisen, andere uit een cluster van systeemeisen. Hieronder

volgt een representatief fragment uit de lijst van alle use cases, gebaseerd op de lijst van functionele systeemeisen in paragraaf 6.3.2.

- Incidentmelding registreren
- Incidentmelding verwijderen
- Oplossingsbeheer
 - Oplossing registreren
 - Oplossing verwijderen
 - Oplossing wijzigen

De hele lijst van use cases kan worden ingezien in de externe bijlage definitiestudie. Om de lezer een concreet beeld te geven van de ontwikkelde use cases, zullen de use cases 'incidentmelding registreren' en 'incidentmelding verwijderen' behandeld worden.

Al in een relatief vroeg stadium van het project werd duidelijk dat het registreren van incident meldingen een van de belangrijkste functionaliteiten van het systeem zou gaan worden. Uit dit gegeven ontstond dan ook de eerste systeemeis genoemd in paragraaf 6.3.2. In figuur 6.3 is de hieruit voortgevloeide use case te zien.

Naam	Incidentmelding registreren
Nummer	3
Samenvatting	De Gebruiker registreert een incident in de vorm van een melding
Actoren	Klant, medewerker, beheerder
Aannamen	De gebruiker is ingelogd en bevindt zich in het incident registreren scherm
Beschrijving	<p>(1) De gebruiker maakt aan het systeem bekend dat hij een incident wilt registreren</p> <p>(2) Als de gebruiker een medewerker is, voert hij de volgende gegevens in: soort, omschrijving, interne prioriteit, externe prioriteit, status, lijn, categorie, betrokken software, betrokken hardware, betreffende relatie, oplossing, de verantwoordelijke medewerker, de oplostijd, een algemene notitie en of de incidentmelding in overeenkomst is met de SLA. Als de gebruiker een klant is, voert hij de volgende gegevens in: Omschrijving, prioriteit, soort, betrokken software, betrokken hardware, relatienaam.(4) De gebruiker verzendt de melding.</p> <p>(5). Het systeem geeft de gebruiker een bevestiging van de ingevoerde gegevens en genereert een meldingsnummer voor de gebruiker</p>
Uitzonderingen	-
Resultaat	De gebruiker heeft een melding geregistreerd en is in het bezit van het bijbehorende meldingsnummer

Figuur 6.3 Incidentmelding registreren

De use case incidentmelding registreren maakt het voor de gebruiker mogelijk een incidentmelding in het systeem op te slaan en hier tevens een aantal attributen aan toe te voegen. Zo kan de gebruiker onder andere aangeven om wat voor soort incident het gaat, wat de status ervan is en welke hardware erbij betrokken is. Opvallend aan deze use case is dat de in te voeren gegevens afhankelijk zijn van het actortype (klant, medewerker of beheerder). Dit is voortgekomen uit de wens van de opdrachtgever, dat klanten ook direct hun eigen incidentmelding kunnen maken via internet. Gegevens die klanten bijvoorbeeld niet kunnen koppelen aan een melding zijn lijn, categorie en oplossing. De in te voeren gegevens wat betreft de actor medewerker en beheerder zijn in deze use case gelijk.

Tijdens het opstellen van de use cases werd duidelijk dat een incidentmelding ook gewijzigd moest kunnen worden. Dit bood de gebruiker de mogelijkheid later gegevens toe te voegen aan een incidentmelding, maar ook veel veranderende eigenschappen van een melding, zoals status en lijn, te kunnen wijzigen. In figuur 6.4 is de use case die dit mogelijk maakt te zien.

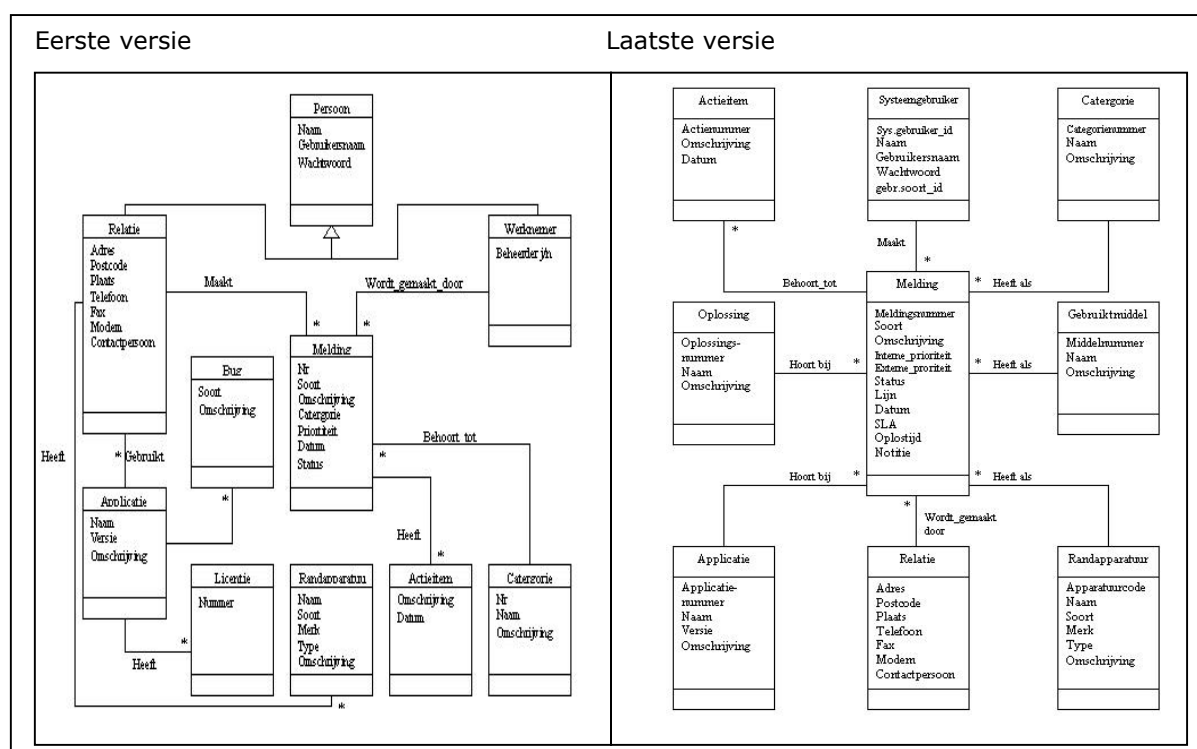
Naam	Incidentmelding wijzigen
Nummer	4
Samenvatting	De Gebruiker wijzigt een incidentmelding
Actoren	Medewerker, beheerder
Aannamen	De gebruiker is ingelogd en bevindt zich in het incidentmelding detail scherm
Beschrijving	(1) De gebruiker maakt aan het systeem bekend dat hij een incidentmelding wilt wijzigen (2) Vervolgens wijzigt de gebruiker de gewenste gegevens. De volgende gegevens kunnen door de gebruiker gewijzigd worden: soort, omschrijving, interne prioriteit, externe prioriteit, status, lijn, categorie, betrokken software, betrokken hardware, betreffende relatie, oplossing, de verantwoordelijke medewerker, de oplostijd, een algemene notitie en of de incidentmelding in overeenkomst is met de SLA (4) De gebruiker klikt op wijzigen. (5). Het systeem geeft de gebruiker een bevestiging van de gewijzigde gegevens
Uitzonderingen	-
Resultaat	De gebruiker heeft een incidentmelding gewijzigd

Figuur 6.4 Incidentmelding registreren

Zoals in deze use case te zien is, kunnen vrijwel alle gegevens van een incidentmelding gewijzigd worden, op de datum en nummer van een melding na. De datum dient verplicht geregistreerd te worden bij het registreren van een nieuwe melding en kan later niet gewijzigd worden om eventuele fraude te voorkomen. Omdat het bij het opstellen van deze use case al bekend was dat het nummer van de melding de primaire sleutel zou gaan vormen, mocht deze ook niet gewijzigd worden.

6.4.2 Opstellen klassediagram

In het systeemconcept is er een klassediagram opgesteld ter modellering van de te ontwikkelen database. Een klassediagram is een model dat alle klassen en attributen van de te ontwikkelen database bevat en de onderlinge relaties tussen de verschillende klassen laat zien. Dit klassediagram is opgesteld op basis van de systeemeisen en de use cases in de definitiestudie. Het klassediagram was een belangrijk onderdeel van dit project omdat het de basis vormde voor de database en de gehele verdere ontwikkeling van het incident managementsysteem. In figuur 6.5 is zowel het allereerste als het allerlaatste klassediagram dat tijdens dit project is opgesteld te zien.



Figuur 6.5 Klassediagrammen

Allereerst zal er iets verteld worden over de wijzigingen die het klassediagram gedurende het project heeft ondergaan. Hierna zal er verteld worden hoe de beschreven systeemeisen in paragraaf 6.3.2, in het klassediagram zijn verwerkt.

De eerste versie van het klassediagram was uiteraard gebaseerd op de eerste versie van de systeemeisen en use cases. Om de klassen in het klassediagram te kunnen vaststellen, was er samen met de opdrachtgever een brainstormsessie gehouden. Binnen deze brainstormsessie kon vrij nagedacht worden over de klassen en de bijbehorende associaties die een plek moesten krijgen in het klassendiagram, zolang ze maar wel binnen de context van de use cases vielen. Er zijn in deze brainstormsessie dus geen nieuwe functionaliteiten naar voren gekomen. Omdat gedurende het project de systeemeisen regelmatig bijgesteld moesten worden, had dit uiteraard ook gevolgen voor het klassediagram. Een opvallend verschil tussen de twee klassediagrammen in figuur 6.5 is het ontbreken van de tabel 'oplossing' in de eerste versie. Dit was een logische ontwerpfout, die al vrij vroeg in het project was gecorrigeerd. Wat ook opvalt tussen beide diagrammen, zijn de attributen die aan een melding kunnen worden gekoppeld. Gedurende het project zijn deze regelmatig veranderd en tot slot geprioriteerd, tot de te koppelen gegevens in de laatste versie van het klassediagram.

Als we even terugkijken naar de lijst van functionele systeemeisen in paragraaf 6.3.2 zult u, als lezer, deze waarschijnlijk wel herkennen in de laatste versie van het klassediagram. Als we naar de eerst genoemde functionele systeemeis kijken, zien we dat een medewerker een incidentmelding moet kunnen registreren of afhandelen door een er een aantal gegevens aan te kunnen koppelen of bepaalde gegevens te kunnen wijzigen. Zoals in de laatste versie van het klassediagram te zien is, staan een aantal van deze gegevens lokaal in de tabel 'melding'. De overige gegevens staan in de verbonden vreemde tabellen. Voor de tweede systeemeis uit de lijst geldt het zelfde principe. Voor het creëren van een globaal en gedetailleerd overzicht zullen er zowel een aantal gegevens lokaal uit de tabel melding worden gehaald, als een aantal gegevens uit de omliggende tabellen. In het gedetailleerde overzicht zullen alle beschikbare gegevens van een incidentmelding gepresenteerd worden aan de gebruiker. De gegevens die in het globale overzicht zullen komen te staan, zullen in het pilotontwikkelplan van de pilot 'meldingsysteem' worden bepaald. De derde systeemeis uit de lijst, richt zich specifiek op de aan de tabel 'melding' gekoppelde tabellen. Alle gegevens uit deze tabellen, dienen in het incident managementsysteem door de beheerder geregistreerd, verwijderd en gewijzigd te kunnen worden.

6.5 Beschouwen technische structuur

In dit onderdeel is de technische structuur waarop het incident management operationeel zou worden gemaakt beschreven. Het doel van de technische structuur is het specificeren van de technische aspecten rond de te ontwikkelen pilots. Dit zal worden gedaan door enerzijds te beschrijven welke hard- en software er nodig is voor de implementatie van het te ontwikkelen systeem en anderzijds door te kijken welke hard- en software er nodig is voor het ontwikkelen van het systeem.

Omdat er in het plan van aanpak van de definitiestudie al grotendeels bepaald was welke hard- en software er tijdens de ontwikkeling van het systeem zou worden gebruikt, was door de afstudeerder besloten, zich te concentreren op de webserver waarop het systeem zou komen te draaien. Voor de aanvang van de technische structuur was er al wel bekend, dat het systeem lokaal ontwikkeld zou gaan worden op een testserver. De organisatie beschikte over een externe webserver die werd onderhouden door een bedrijf genaamd Solcon. Door enerzijds kritisch te kijken naar de specificatie van de webserver en anderzijds naar de benodigde specificatie van het te ontwikkelen systeem, kon de conclusie worden getrokken dat de webserver van Solcon in ruime mate voldeed. In de onderstaande tabel zijn de specificaties van de externe webserver te zien.

Omvang website (MB)	100
Vrij dataverkeer website (MB/maand)	20000
Real audio/video	Ja
Scripts toegestaan	Ja, CGI/PHP/ASP
Database toegestaan	Ja, MySQL
Statistieken website bezoek en dataverkeer	Ja
Webserver op locatie	Nee

Bron: www.solcon.nl

6.6 Beschouwen organisatorische inrichting

In dit onderdeel is er naar de organisatorische inrichting van Technico Automatisering gekeken. Het doel van het bepalen van de organisatorische inrichting is het specificeren van de organisatieaspecten rond de pilot in ontwikkeling. Dit is belangrijk om een goed overzicht te creëren van de veranderingen die de uiteindelijke invoering van het systeem met zich mee zullen brengen. Door kritisch naar de huidige procesgang van het afhandelen van incidentmeldingen te kijken, was de afstudeerder in staat de organisatorische inrichting op een correcte en realistische manier te beschrijven. Tevens is het eindresultaat van dit onderdeel met de opdrachtgever besproken en is zijn feedback erin verwerkt. In organisatorische inrichting is er naar een volgende zaken gekeken:

- De manier waarop incidentmeldingen in de huidige situatie worden geregistreerd en afgehandeld
- De manier waarop de service die aan de klant wordt verleend zal gaan veranderen
- Waaruit het globaal opleidingsplan zal gaan bestaan
- Waaruit de gebruikersdocumentatie zal gaan bestaan

Door deze zaken in een vroegtijdig stadium te beschrijven, waren de gevolgen voor de organisatorische inrichting met de invoering van het incident managementsysteem voor zowel de afstudeerder als de opdrachtgever bekend en kon er rekening mee gehouden worden. Op basis van de organisatorische inrichting kon de conclusie worden getrokken dat het incident management-systeem een bepaalde aanlooptijd had en eerst zijn nut naar de klanten toe moest bewijzen, eer het door hen gebruikt zou worden. Dit nut was voor de klant het feit dat zijn incidentmelding sneller afgehandeld zou worden en hij de status van zijn incidentmelding beter kon volgen. Tevens kon er worden geconcludeerd op basis van het kennisniveau van de toekomstige gebruikers (medewerkers en beheerders), dat het opleidingsplan in de vorm van een korte demonstratie of beknopte handleiding kon worden achtergelaten bij de organisatie.

6.7 Opstellen pilotplan

In dit onderdeel is het pilotplan opgesteld. Het pilotplan is één van de belangrijkste producten waarop een beslissing over verdere ontwikkeling van het systeem wordt gebaseerd. Elke pilot heeft betrekking op een bepaald deel van het systeemconcept. In het pilotplan zijn de verschillende pilots nader toegelicht en word aangeven hoe het ontwerp van deze pilot eruit zal zien. Daarnaast word wederom een verder detaillering van het planning gemaakt. Het pilotplan is opgenomen in dit verslag om de lezer inzicht te verschaffen in de denkwijze die is gevolgd bij het plannen van de pilots

In het pilotplan werd de verdeling van de pilots pas definitief besloten. Het systeem zou uit de volgende pilots gaan bestaan:

1. Database
2. Meldingssysteem
3. Gegevensbeheer

Nadat dit definitief was vastgesteld werden de drie pilots ingepland met behulp van een timebox. In een time box staat de tijd die er voor het ontwikkelen van een pilot beschikbaar is en de prioriteit van de pilot ten opzichte van de andere pilots. De rede voor het onderbrengen van de pilots in timeboxes was om een strakke scheiding qua tijd te krijgen, in de drie onderdelen waaruit het systeem zou gaan bestaan. Bij het toekennen van de beschikbare tijd per pilot, is er teruggegrepen op de ervaring die de afstudeerder in eerdere projecten heeft opgedaan. Tevens had de afstudeerder de planning besproken met de opdrachtgever en medestudenten en hun expertise meegenomen in het bepalen van de beschikbare tijd per pilot.

Nadat alle pilots in de timebox waren verwerkt, werd er bepaald uit welke onderdelen de drie pilots zouden gaan bestaan. Hierbij werd onder andere gekeken naar welke onderdelen uit de methodiek IAD van toepassing waren op de pilot. Hieronder staan de verschillende onderdelen die voor de pilot 'meldingssysteem' en 'gegevensbeheer' zouden worden opgesteld. Tevens staat er bij elk onderdeel een verklaring waarom het onderdeel voor deze pilots is opgesteld.

- Plan van aanpak; het plan van aanpak is absoluut noodzakelijk voor alle pilots die gedurende dit project ontwikkeld gaan worden. Het plan van aanpak helpt structuur te geven aan de werkzaamheden en sluit vertraging en het niet halen van deadlines zo veel mogelijk uit.
- Functionele structuur pilot; de functionele structuur is belangrijk omdat in dit onderdeel de functionaliteiten van het incident managementsysteem worden ontwikkeld. Dit wordt gedaan doormiddel van schetsen van de te ontwikkelen schermen, een menustructuur en een navigatieschema.
- Organisatorische structuur pilot; Het is voor de ontwikkelaar belangrijk om te weten of het organisatorisch mogelijk is zijn systeem binnen de organisatie te implementeren en wat de gevolgen voor de organisatie hiervan zijn. Het is in dit geval vooral belangrijk, omdat het te ontwikkelen systeem ondersteuning moet gaan bieden bij een primair proces, incident management.
- Pilotontwikkelplan; in dit onderdeel wordt de pilot daadwerkelijk ontworpen. Zonder hier te veel over te zeggen, kan gesteld worden dat dit het meest belangrijke onderdeel is van het pilotplan.
- Verificatie en validatie; Het is belangrijk dat het conceptueel ontwerp uit de definitiestudie ook daadwerkelijk een plek krijgt in het uiteindelijke incident managementsysteem. In dit onderdeel zal worden gekeken of het toegewezen conceptuele gedeelte ook daadwerkelijk door de pilot wordt gedekt.

Deze onderdelen waren ook opgesteld voor de pilot 'database', met uitzondering van de organisatorische inrichting. Omdat deze pilot zelf geen functionaliteiten bevatte, had hij ook geen directe invloed op aanwezige processen binnen de organisatie.

6.8 Verificatie en validatie

In dit onderdeel is verificatie en validatie van de definitiestudie uitgevoerd. Activiteiten met betrekking tot verificatie en validatie zijn erop gericht te verzekeren dat de systeemontwikkeling op de juiste manier wordt uitgevoerd met de juiste resultaten. In het kader van de verificatie en validatie van de definitiestudie zijn functionele systeemeisen tegen de use cases en de pilotdelen afgezet. In figuur 6.6 wordt er in de linker afbeelding geverifieerd dat de verschillende systeemeisen ondervangen zijn door de opgestelde use cases. In de rechter afbeelding wordt geverifieerd dat de verschillende systeemeisen ondervangen zijn door de drie pilots. De conclusie die na de verificatie en validatie kon worden getrokken, was dat alle systeemeisen waren ondervangen door de use cases en alle use cases door de drie pilots. Pas nadat dit bevestigd was, kon er begonnen worden met de ontwikkeling van de eerste pilot, de database.

	1	2	3i	3ii	3iii	3iv	3v	3vi	3vii	3viii	3ix	4i	4ii	4iii
1	x	x										x	x	x
2	x	x										x	x	x
3			x			x						x	x	x
4					x							x	x	x
5										x		x	x	x
6				x								x	x	x
7						x						x	x	x
8							x					x	x	x
9						x						x	x	x
10										x		x	x	x
11												x	x	x
12												x	x	x
13										x		x	x	x
14												x	x	x
15										x		x	x	x
16										x		x	x	x
17										x		x	x	x
18										x		x	x	x
19										x		x	x	x
20										x		x	x	x
21										x		x	x	x
22										x		x	x	x
23										x		x	x	x
24										x		x	x	x
25										x		x	x	x
26										x		x	x	x
27										x		x	x	x
28											x	x	x	x
29												x	x	x
30												x	x	x
31												x	x	x
32												x	x	x
33												x	x	x
34												x	x	x

	1	2	3i	3ii	3iii	3iv	3v	3vi	3vii	3viii	3ix	4i	4ii	4iii
1	x	x										x	x	x
2	x	x	x	x	x	x	x	x	x			x	x	x
3	x	x							x	x	x	x	x	x

N.b. in de rijen staan de nummers van de pilots en in de kolommen de nummers van de systeemeisen.

N.b. in de rijen staan de nummers van de use cases en in de kolommen de nummers van de systeemeisen.

Figuur 6.6 Functionele systeemeisen v.s. use cases en pilotdelen

7. Pilotontwikkeling

In dit hoofdstuk zal de doorlopen procesgang van de pilotontwikkeling worden beschreven. In het ontwikkelscenario van de definitiestudie is besloten het te ontwikkelen systeem op te delen in drie pilots, namelijk de database, het meldingsysteem en het gegevensbeheer. Per pilot zal er worden beschreven welke activiteiten er moesten worden uitgevoerd, hoe dit werd aangepakt, de problemen die de afstudeerder hierbij tegenkwam en de besluiten c.q. oplossingen die hij heeft genomen c.q. gemaakt

7.1 Pilot 1: Database

In dit onderdeel is de database van het systeem ontwikkeld. De database moest aan verschillende eisen voldoen welke zijn terug te vinden in de systeemeisen van de externe bijlage definitiestudie. De geplande tijd voor deze pilot was geschat op tien dagen, conform het pilotplan in de definitiestudie. In de pilot database, zijn de volgende onderdelen gerealiseerd:

- Pilotontwikkeldocument database
 - Plan van aanpak
 - Functionele structuur
 - Pilotontwikkeldocument
 - Verificatie en validatie
- De daadwerkelijke database

7.1.1 Specificeren plan van aanpak database

In het plan van aanpak staat gespecificeerd welke activiteiten er in deze fase zullen worden uitgevoerd, uit welke onderdelen het pilotontwikkeldocument van de pilot database zal gaan bestaan en hoeveel tijd deze onderdelen in beslag zullen gaan nemen. Het plan van aanpak voor de pilotontwikkeling van de database bepaald de aanpak gedurende dit onderdeel. Het belangrijkste onderdeel uit het plan van aanpak is de planning. In deze planning staat nauwkeurig vermeld welke activiteiten wanneer diende te worden uitgevoerd en hoeveel tijd hiervoor beschikbaar was. Het plan van aanpak hielp structuur te geven aan de werkzaamheden en sloot vertraging en het niet halen van deadlines zo veel mogelijk uit. Het plan van aanpak van de pilotontwikkeling van de database is afgeleid van het globale plan van aanpak dat aan het begin van het project is opgesteld. In het plan van aanpak voor deze pilot zijn achtereenvolgens de reikwijdte, de resultaten en de planning beschreven. De planning was het belangrijkste onderdeel van dit plan van aanpak. Door de eerder beschreven manier van plannen, moest via deze manier de overkoepelende planning verder gedetailleerd worden. Voor een gedetailleerdere beschrijving van veranderingen en opbouw van de planning die gedurende dit project is gebruikt wordt doorverwezen naar paragraaf 4.2 van dit document.

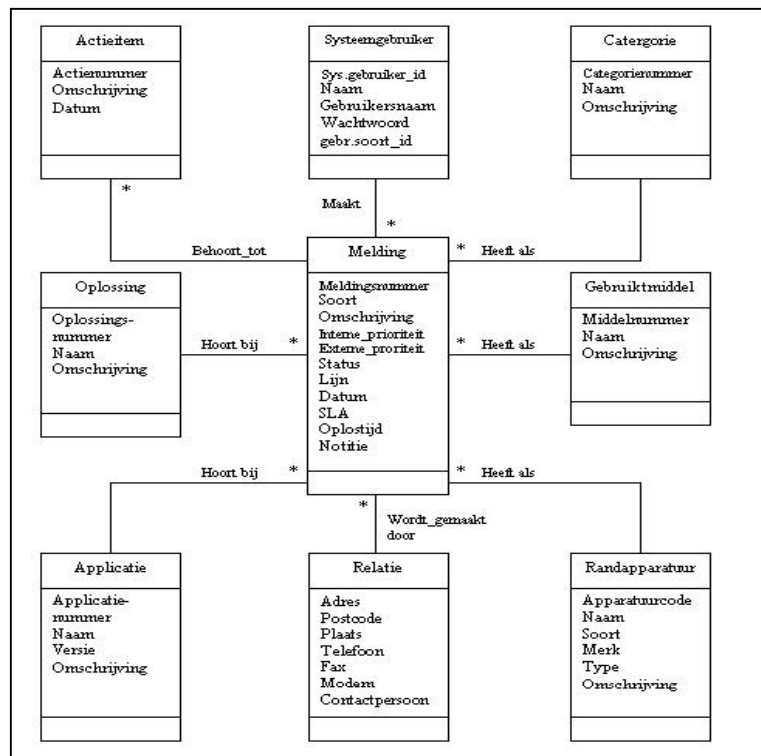
7.1.2 Specificeren globaal functionele structuur database

In de functionele structuur van de pilot 'database' zijn de volgende onderdelen gerealiseerd:

- Klassediagram
- Relationeel representatiemodel
- Relationeel implementatiemodel

De methodiek IAD biedt in deze fase van het ontwikkeltraject de gelegenheid nog eens kritisch te kijken naar het klassediagram waarna hij vervolgens definitief kan worden gemaakt voor de betreffende iteratie. Hoewel deze procesgang wel om deze manier is doorlopen, is het klassediagram dat in het systeemconcept van de definitiestudie wordt beschreven (paragraaf 6.4.2) de definitieve. Daarom zal hier verder niet meer worden ingegaan op het klassediagram.

Wel is in figuur 7.1 nogmaals de laatste versie van het klassediagram te zien, dit ten behoeve van de overzichtelijkheid van dit document.



Figuur 7.1 Laatste versie klassediagram

In de functionele structuur van de database zijn ook een relationeel representatiemodel en relationeel implementatiemodel opgesteld. De rede voor het kiezen van deze twee modellen is tweeledig:

- Het zijn universele modellen die voor het ontwikkelen van vrijwel elk soort relationele database te gebruiken zijn.
- Deze twee modellen zijn de afstudeerder in de opleiding aangeleerd, het zou zonde van de tijd zijn en onnodige risico's met zich meebrengen als de afstudeerder voor het gebruik van andere modellen zou kiezen.

Het relationele representatiemodel is een tussenstap om van een klassediagram een uiteindelijk implementatiemodel te maken. In het relationeel representatiemodel zijn de primaire sleutels en vreemde sleutels aan de verschillende tabellen uit het klassediagram toegewezen. Doormiddel van de relaties tussen de verschillende tabellen in het klassediagram was het bij het opstellen van het relationeel representatiemodel al wel duidelijk in welke tabellen de vreemde sleutels zouden komen te staan. Hierna restte alleen nog het bepalen van de primaire sleutels van elke tabel. Het was belangrijk dat deze een logische naam kregen die paste binnen de tabel. Een voorbeeld hiervan is de tabel 'melding'. Deze kreeg als primaire sleutel 'meldingsnummer'. Hieronder volgt een fragment uit het relationeel representatiemodel van de klasse melding:

Melding (Meldingsnummer, Soort, Omschrijving, Datum, Status, Lijn, Interne_prioriteit, Externe_prioriteit, SLA, Oplostijd, Notitie, *Persoon_id*, *Randapparatuurnummer*, *Categorinummer*, *Systeem_gebruiker_id*, *Oplossingsnummer*, *Middelnummer*)

Vreemde sleutel *Persoon_id* refereert aan *Persoon_id* in Relatie (null)

Vreemde sleutel *Applicatienummer* refereert aan *Applicatienummer* in Applicatie (null)

Vreemde sleutel *Randapparatuurnummer* refereert aan *Randapparatuurnummer* in Randapparatuur (null)

Vreemde sleutel *Categorinummer* refereert aan *Categorinummer* in Categorie (null)

Vreemde sleutel *Systeem_gebruiker_id* refereert aan *Systeem_gebruiker_id* in Systeem_gebruiker (null)

Vreemde sleutel *Oplossingsnummer* refereert aan *Oplossingsnummer* in Oplossing (null)

Vreemde sleutel *Middelnummer* refereert aan *Middelnummer* in Gebruikt_middel (null)

Voor het volledige relationele representatiemodel van de database wordt doorverwezen naar externe bijlage twee, pilotontwikkelplan database. In het bovenstaande fragment zijn de primaire sleutels onderstreept en de vreemde sleutels *schuin* gedrukt. Nadat het relationeel representatiemodel naar bevrediging was opgesteld diende er een relationeel implementatiemodel te worden opgesteld. In het relationeel implementatiemodel werden aan de verschillende attributen, attribuuttypen toegekend. Als een attribuut bijvoorbeeld bestond uit alleen maar hele cijfers, dan kreeg hij als attribuut typen 'integer'. Als een attribuut uit een tekst bestond, dan kreeg hij als attribuut typen 'text'. Tevens moest er bij elk attribuut worden bepaald of hij leeg mocht zijn (null) of dat hij altijd een waarde moest hebben (not null). Tot slot werd in het relationeel implementatie model net zoals in het relationeel representatiemodel, de primaire en vreemde sleutels aangegeven. Hieronder volgt een fragment uit het relationeel implementatiemodel van de klasse melding:

```
CREATE TABLE melding (
  Meldingsnummer      int(4)          NOT NULL      auto_increment,
  Persoon_id           int(10)         default NULL,
  Soort                varchar(13)     default NULL,
  Omschrijving         text,
  Datum               date             default NULL,
  Status              varchar(20)     default '',
  Lijn                varchar(13)     default NULL,
  Interne_prioriteit   varchar(13)    default NULL,
  Externe_prioriteit   varchar(13)    default NULL,
  SLA                 int(1)          default NULL,
  Oplostijd           int(3)          default NULL,
  Notitie             text,
  Applicatienummer     int(10)         default NULL,
  Randapparatuurnummer int(5)         default NULL,
  Catergorienummer     int(2)         default NULL,
  Systeem_gebruiker_id int(10)         default NULL,
  Oplossingsnummer     int(3)         default NULL,
  Middelnummer        int(2)         default NULL,

  FOREIGN KEY (Applicatienummer) REFERENCES Applicatie
  FOREIGN KEY (Randapparatuurnummer) REFERENCES Randapparaatuur
  FOREIGN KEY (Catergorienummer) REFERENCES Catergorie
  FOREIGN KEY (Systeem_gebruiker_id) REFERENCES Systeem_gebruiker
  FOREIGN KEY (Oplossingsnummer) REFERENCES Oplossing (Oplossingsnummer),
  FOREIGN KEY (Middelnummer) REFERENCES Gebruikt middel (Middelnummer),
  FOREIGN KEY (Persoon_id) REFERENCES Relatie (Persoon_id),
  PRIMARY KEY (Meldingsnummer)
)
```

Voor het volledige relationele implementatiemodel van de database wordt doorverwezen naar externe bijlage twee, pilotontwikkelplan database. Nadat het relationeel volledig was opgesteld, moest deze worden geïmporteerd in de MySQL database. MySQL is een relationeel database platform dat gebruikt is voor dit project. Met behulp van PHPMyAdmin, een beheertools voor de MySQL database, kon het relationeel implementatiemodel in MySQL worden geïmporteerd. Een screenshot van het invoerveld van PHPMyAdmin is te zien in figuur 7.2.



Figuur 7.2 PHP Admin invoerveld

Nadat de database was ingevoerd in de MySQL database, was hij klaar voor gebruik. Nu kon er begonnen worden aan het specificeren van de bijbehorende functionele documentatie het opstellen van een testprocedure.

7.1.3 Opstellen pilotontwikkelplan database

In het pilotontwikkelplan van de pilot 'database' zijn de volgende onderdelen gerealiseerd:

- Specificatie klassen
- Specificatie attributen
- Planning pilot
- Testprocedure
- Invoering pilot

Om zowel de afstudeerder, de opdrachtgever en eventuele toekomstige ontwikkelaars inzicht te verschaffen in de gedachtegang achter de verschillende klassen en bijbehorende attributen, zijn deze per stuk gedetailleerd beschreven. Een fragment van deze beschrijving is in de onderstaande tabel te zien. Het gaat hier om drie attributen van de klasse melding.

Melding	Meldingsnummer	Dit is de primaire sleutel die elke melding uniek maakt binnen het systeem
	Soort	Dit geeft aan om wat voor soort melding het gaat
	Omschrijving	Hier kan een omschrijving van de melding worden gegeven

Nadat alle klassen en attributen beschreven waren, was het tijd om na te denken over hoe de database getest kon gaan worden. Na een kort onderzoek, waarin verschillende PHP testscrips op bruikbaarheid werden bestudeerd, werd besloten het onderstaande testscript te gebruiken vanwege zijn simpele en overzichtelijk karakter.

```
?php

$hostname = "localhost";
$username = "root";
$password = "";
$db = "Meldingsysteem";

$global_dbh = mysql_connect($hostname, $username, $password);
mysql_select_db($db, $global_dbh);

function display_db_table($tablename, $connection)
{
    $query_string = "select * from $tablename";
    $result_id = mysql_query($query_string, $connection);
    $column_count = mysql_num_fields($result_id);

    print("<TABLE BORDER=1>\n");
    while ($row = mysql_fetch_row($result_id))
    {
        print("<TR ALIGN=LEFT VALIGN=TOP>");
        for ($column_num = 0;
            $column_num < $column_count;
            $column_num++)
        {
            print("<TD>$row[$column_num]</TD>\n");
        }
        print("</TR>\n");
    }
    print("</TABLE>\n");
}

?>

<HTML>
<HEAD>
<TITLE>Database overzicht</TITLE>
</HEAD>
<BODY>

<TABLE><TR><TD>

</TD><TD>
```

```

<?php display_db_table("Melding", $global_dbh); ?>
</TD><TD>
<?php display_db_table("Oplossing", $global_dbh); ?>
</TD><TD>

<?php display_db_table("Randapparatuur", $global_dbh); ?>
</TD><TD>
<?php display_db_table("Relatie", $global_dbh); ?>
</TD><TD>
<?php display_db_table("Systeem_gebruiker", $global_dbh); ?>
</TD></TR></TABLE></BODY></HTML>

```

Voordat het script gebruikt kon gaan worden, diende er eerst een aantal zaken aangepast te worden. De inloggegevens van de database moesten gewijzigd worden met de juiste 'hostname', 'username', 'password' en 'db' (de naam van de database). Vervolgens moest de *display_db_table* functie, welke er voor zorgde dat alle tabellen uit de database op het scherm werden getoond, voorzien worden met de namen van de tabellen uit de database. Tot slot diende de database gevuld te worden met een aantal testgegevens. Na het uitvoeren van het testscript werden alle tabellen op het scherm getoond. Dit leidde tot de conclusie dat de implementatie van de database een geslaagde onderneming was.

7.1.4 Verificatie en validatie database

In dit onderdeel wordt de verificatie en validatie van de pilot 'database' beschreven. Activiteiten met betrekking tot verificatie en validatie zijn erop gericht te verzekeren dat de systeemontwikkeling op de juiste manier wordt uitgevoerd met de juiste resultaten. In het kader van de verificatie en validatie van de pilot 'database', zijn de attributen uit het klassediagram, tegen het relationeel representatie model en relationeel implementatiemodel afgezet. De conclusie die uit deze verificatie kon worden getrokken, was dat alle attributen uit het klassediagram uiteindelijk ook in de database waren opgenomen. Vanwege de grote van de verificatietabel, is er in de onderstaande tabel slechts een fragment van te zien. Voor de volledig tabel wordt doorverwezen naar externe bijlage twee, pilotontwikkelplan database.

Klasse	Attribuut	Klasse-diagram	RRM	RIM
Melding	Meldingsnummer	x	x	x
	Soort	x	x	x
	Omschrijving	x	x	x
	Datum	x	x	x
	Status	x	x	x
	Lijn	x	x	x
	Interne prioriteit	x	x	x
	Externe prioriteit	x	x	x
	SLA	x	x	x
	Oplostijd	x	x	x
	Notitie	x	x	x
	Persoon_id	x	x	x
	Randapparatuur-nummer	x	x	x
	Catergorienummer	x	x	x
	Systeem_gebruiker_id	x	x	x
	Oplossingsnummer	x	x	x
	Middelnummer	x	x	x

Telkens nadat de pilot database door de iteraties heen was geversificeerd en gevalideerd, kon er verder worden gegaan met de volgende pilot, het meldingsstelsysteem.

7.2 Pilot 2: Meldingsstelsysteem

In dit onderdeel is de pilot 'meldingsstelsysteem' ontwikkeld. De pilot 'meldingsstelsysteem' moest aan verschillende eisen voldoen welke zijn terug te vinden in de systeemeisen van de definitiestudie. De geplande tijd voor deze pilot was geschat op 23 dagen, conform het pilotplan in de definitiestudie. Deze pilot omvatte alle functionaliteiten omtrent het beheren van een incident-melding. In de pilot 'meldingsstelsysteem', zijn de volgende onderdelen gerealiseerd:

- Pilotontwikkeldrapport meldingsysteem
 - Plan van aanpak
 - Functionele structuur
 - Organisatorische inrichting
 - Pilotontwikkeldplan
 - Verificatie en validatie
- Pilot 2 'meldingsysteem'

Ten eerste zal er in deze paragraaf verteld worden hoe het plan van aanpak van deze pilot tot stand is gekomen. Vervolgens zal er worden beschreven hoe de afstudeerder tot de functionele structuur van deze pilot is gekomen. Tot slot zal er iets verteld worden over de procesgang van de organisatorische inrichting, het pilotontwikkeldplan en de verificatie en validatie.

7.2.1 Specificeren plan van aanpak meldingsysteem

In het plan van aanpak is gespecificeerd welke activiteiten er in deze fase zouden worden uitgevoerd, uit welke onderdelen het pilotontwikkeldrapport van de pilot meldingsysteem zou gaan bestaan en hoeveel tijd deze onderdelen in beslag zouden gaan nemen. Voor een gedetailleerde beschrijving van veranderingen en opbouw van de planning die gedurende dit project is gebruikt wordt doorverwezen naar paragraaf 4.2 van dit document.

7.2.2 Specificeren globaal functionele structuur meldingsysteem

In deze paragraaf is de functionele inhoud van de pilot 'meldingsysteem' bepaald. De afstudeerder streefde ernaar, de functionaliteit in dit onderdeel op een zodanig detailniveau te beschrijven, dat het als een duidelijke en ondubbelzinnige basis gebruikt kon worden voor de bouw van de pilot 'meldingsysteem'.

De globaal functionele structuur van het meldingsysteem bestaat uit:

- Gebruikte use cases uit de definitiestudie
- Menustructuur
- Navigatieschema
- Schets van de schermen

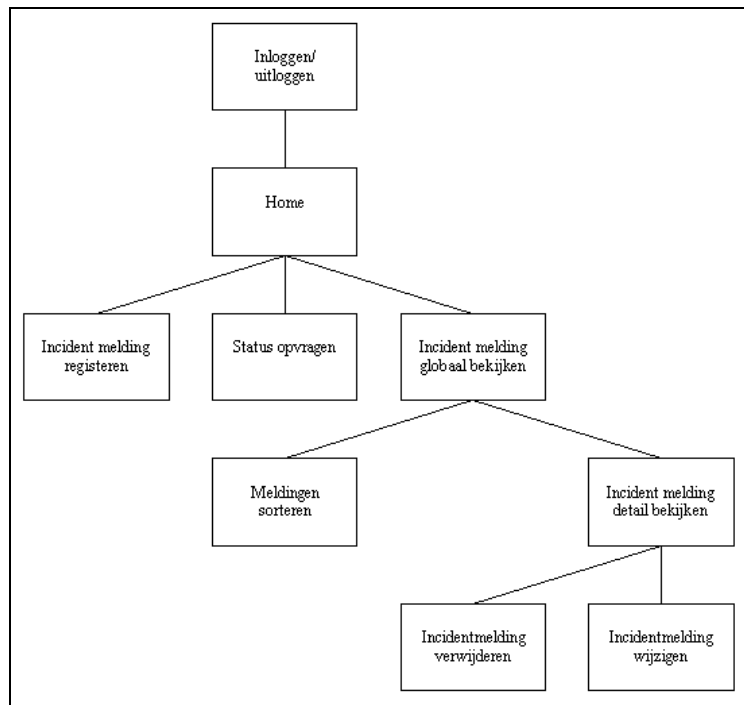
Gebruikte use cases uit de definitiestudie meldingsysteem

In het systeemconcept van de definitiestudie zijn de use cases opgesteld die uiteindelijk gezamenlijk de functionaliteiten van het systeem moesten gaan vormen. De pilot 'meldingsysteem' is gevormd uit een cluster van use cases. De gedachten hierachter was alle use case die een directe betrekking hadden op de primaire meldingfunctionaliteiten in één pilot onder te brengen. In deze pilot is voor elke use case een scherm ontwikkeld omdat het in de ogen van de afstudeerder in het kader van de overzichtelijkheid en gebruiksvriendelijkheid van het systeem, onverstandig was meerdere use cases samen te voegen tot één scherm. Voor een volledige lijst van de use cases die in deze pilot gebruikt zijn, wordt u doorverwezen naar het systeemconcept in de definitiestudie, externe bijlage één.

Menu structuur meldingsysteem

Voor dat er aan de schetsen van de schermen kon worden begonnen, diende er eerst een menustructuur te worden opgesteld. De menustructuur was ter verduidelijking van de hiërarchie van de schermen van het meldingsysteem. In dit schema is te zien waar de verschillende schermen zich in de hiërarchie van het totale systeem moesten gaan bevinden. De menustructuur is opgesteld aan de hand van de toewezende use cases. Daarbij kwam de ervaring die de afstudeerder in eerdere projecten had opgedaan op het gebied van menustructuren goed van pas. De hiërarchie is bepaald na een brainstormsessie samen met de opdrachtgever. In deze brainstormsessie is gesimuleerd hoe een gebruiker de verschillende functionaliteiten van de schermen moest kunnen gaan gebruiken.

In figuur 7.3 is de relatie te zien tussen de diverse schermen van de pilot 'meldingsysteem' in de opgestelde menustructuur.



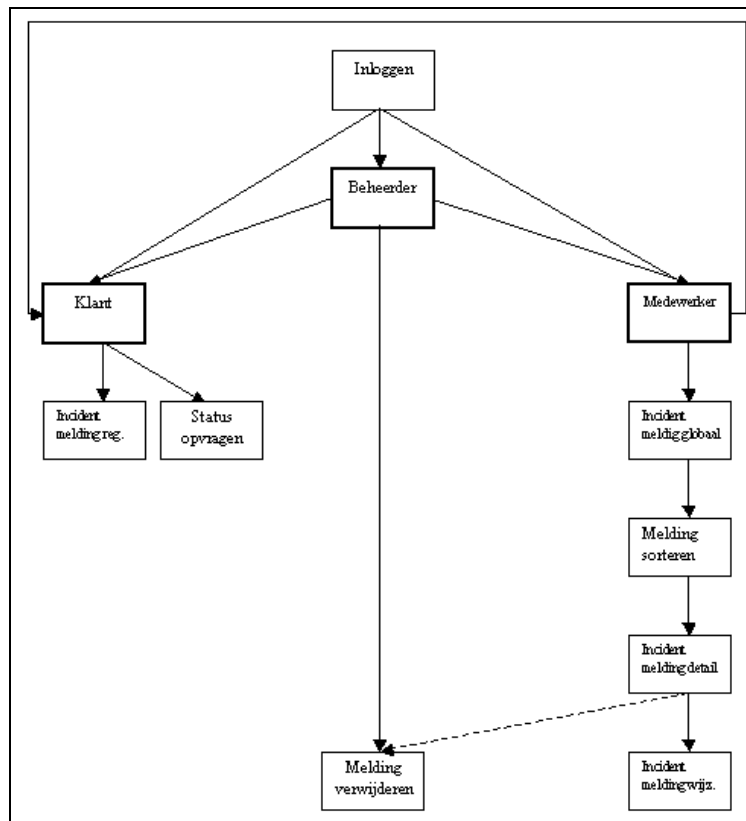
Figuur 7.3 Menustructuur pilot 'meldingsysteem'

Nadat de menustructuur volledig was afgerond kon er begonnen worden met het opstellen van het navigatieschema.

Navigatieschema meldingsysteem

Het navigatieschema is ter verduidelijking van de schermen van het meldingsysteem. De onderlinge relaties tussen de schermen staan erin weergegeven. Het opstellen van een navigatieschema werd in dit project belangrijk geacht, omdat er in zo'n schema eenduidig kan worden weergegeven, welke koppelingen er tussen de schermen moesten worden ontwikkeld. Tevens bood het de gelegenheid nog eens goed te kijken naar de verdeling van de verschillende actortypen. Zoals in figuur 7.4 te zien is, worden er in het navigatieschema drie actortypen onderscheiden. Deze zijn de klant, medewerker en beheerder. Elk hebben ze toegang tot bepaalde functionaliteiten van het meldingsysteem.

In het navigatieschema in figuur 7.4 is een stippellijn te zien die van 'incidentmeldingen detail' naar 'melding verwijderen' loopt. De rede van deze stippellijn is het feit dat de functionaliteit 'melding verwijderen' toegankelijk is vanaf 'incidentmeldingen detail'. Het is echter geen functionaliteit waar de actor medewerker toegang toe heeft. Dit was het resultaat van het besluit van de opdrachtgever, dat werknemers niet bevoegd mochten zijn meldingen van incidenten uit het systeem te verwijderen. Dit besluit was genomen om fraude of eventuele misverstanden zo veel mogelijk uit te sluiten.



Figuur 7.4 Navigatieschema pilot 'meldingsysteem'

Nadat de hiërarchie van de schermen en de navigatie ertussen vastgesteld was, kon er worden begonnen met het ontwerpen van de schermen.

Schetsen van de schermen meldingsysteem

In deze subparagraaf zullen de schetsen van de schermen behandeld worden. De schetsen van de schermen zijn als het ware bouwplannen die de ontwikkelaar maakt om zich een concreet beeld te vormen van hoe de diverse schermen van het uiteindelijke systeem eruit moeten komen te zien. Uiteraard is dit in dit project in regelmatige overstemming met de opdrachtgever gedaan. Door de schetsen van de schermen regelmatig met hem te bespreken en rekening te houden met alle voorgaande stappen, zoals de use cases, menustructuur en navigatieschema, heeft de afstudeerder schetsen kunnen ontwerpen, die nauw aansloten op de wensen van de toekomstige gebruikers.

De pilot 'meldingsysteem' was na een clustering van de use cases, opgedeeld in de volgende pilotdelen:

1. Inloggen/uitloggen: dit deel betreft de inlog en uitlog functionaliteiten van het systeem.
2. Incidentmelding registreren: dit deel betreft de functionaliteiten die nodig zijn voor het registreren van een incidentmelding.
3. Incidentmelding wijzigen: dit deel betreft de functionaliteiten die nodig zijn voor het wijzigen van een incidentmelding.
4. Incidentmelding verwijderen: dit deel betreft de functionaliteiten die nodig zijn voor het verwijderen van een incidentmelding.
5. Status opvragen: dit deel betreft de functionaliteiten die nodig zijn voor het opvragen van de status van een incidentmelding.
6. Incidentmelding globaal bekijken: dit deel betreft de functionaliteiten die nodig zijn om een globaal overzicht van alle incidentmeldingen te creëren.
7. Meldingen sorteren: dit deel betreft de functionaliteiten die nodig zijn voor het uitvoeren van selectiecriteria op de incidentmeldingen.
8. Incidentmeldingen detail bekijken: dit deel betreft de functionaliteiten die nodig zijn om een gedetailleerd overzicht te creëren van één enkele incidentmelding.

De pilot meldingsysteem is in pilotdelen opgedeeld, omdat hij te groot was om in één keer te ontwikkelen. De delen zijn zo gekozen, dat ze één functionaliteit of een groep van verwante functionaliteiten omvatten.

Bij het inloggen op het systeem wordt de gebruiker afhankelijk van de ingevoerde combinatie van de gebruikersnaam en wachtwoord, doorgestuurd naar zijn startpagina. Met het onderstaande stukje code wordt gecontroleerd om wat voor actortype het gaat, alvorens de gebruiker naar de juiste startpagina wordt doorverwezen:

```
//als de gebruiker een beheerder is
if (mysql_num_rows($res_db) >= 1)
{
    // zet de values van database in een array
    $row = mysql_fetch_array($res_db);

    // zet de sessions
    $sess_id = $row['Systeem_gebruiker_id'];
    $sess_loginnaam = $row['Username'];

    // registreer ze
    session_register('sess_id');
    session_register('sess_loginnaam');

    echo "<body bgcolor=\"blue\"><strong><center><font color=\"#FFFFFF\">Je bent nu ingelogd. Klik <a
href=$nextpage_beheerder>hier</a> om verder te gaan.</font></center></strong></body>";
}
```

Wat er vervolgens ontwikkeld diende te worden was het hoofdframe. Er moest dus nadacht worden over zaken zoals, waar worden de diverse schermen geopend, hoe komt het titelframe eruit te zien, waar komen de navigatie-elementen te staan en welke kleuren zullen er gebruikt worden. Het nadenken over en bespreken van deze vragen leiden uiteindelijk tot het hoofdframe in figuur 7.5.



Figuur 7.5 Hoofdscherm incident managementsysteem

Het voordeel van het gebruik van de indeling in figuur 7.5, is dat het door het hele programma mogelijk blijft om naar vaste functionaliteiten te kunnen navigeren. Zo blijft het overzicht binnen het systeem behouden. De omcirkelde navigatieknoppen in figuur 7.5 zijn specifiek voor de pilot

meldingsysteem. Wanneer de gebruiker op de knop 'meldingen' klikt, zal hij terecht komen in het 'incidentmelding globaal scherm'. Een schets van dit scherm is te zien in figuur 7.6.

Nummer	Datum	Relatie	Categorie	Soort	Prioriteit	Werknemer	Lijn	Status
<i>Hyperlink</i>	<i>Tekstveld</i>	<i>Tekstveld</i>	<i>Tekstveld</i>	<i>Tekstveld</i>	<i>Tekstveld</i>	<i>Tekstveld</i>	<i>Tekstveld</i>	<i>Tekstveld</i>

Figuur 7.6 Schets 'incidentmelding globaal' scherm

In de eerste instantie waren alle schetsen van schermen gemaakt met behulp van de HTML editor 'Dreamweaver'. Het gebruik van deze editor had voordelen en nadelen. Het voordeel was dat de schetsen van de schermen er zeer realistisch en begrijpelijk uitzagen, wat de communicatie naar de opdrachtgever en de toekomstige gebruikers toe enorm vereenvoudigde. Een bijkomend voordeel was dat de schetsen gedeeltelijk in hun huidige vorm, geïmplementeerd kon worden in het uiteindelijke systeem. Deze twee voordelen wogen echter niet op tegen één groot nadeel. Vanwege het dynamische karakter van schetsen van de schermen, dienden ze regelmatig aangepast te worden. Dit was een nogal tijdrovende activiteit omdat de schetsen in HTML waren opgebouwd. Dit leidde tot het besluit de schetsen met behulp van Microsoft Word te maken wat als voordeel had dat je schetsen gemakkelijk aanpasbaar werden.

In de schets in figuur 7.6 ziet de gebruiker een globaal overzicht van alle incidentmeldingen. Het scherm laat slechts een deel van de bijhorende gegevens zien. De keuze voor deze gegevens zijn in overeenstemming met de opdrachtgever en overige belanghebbende gemaakt. In figuur 7.7 is te zien hoe deze schets uiteindelijk is ontwikkeld tot een functioneel scherm.

Nummer	Datum	Relatie	Categorie	Soort	Prioriteit	Verantw.	Lijn	Status
30	240904	test	Software-fout	Klacht	Middel	eelke	Tweede	In behandeling
29	240904	test	Hardware-fout	Klacht	Laag	ronald	Eerste	Niet in behandeling
28	240904	test	Hardware-fout	Klacht	Laag	ronald	Tweede	Niet in behandeling
27	240904	test	Software-fout	Klacht	Laag	ronald	Derde	Niet in behandeling
26	240904	-	-	-	-	-	-	-
25	240904	-	-	-	-	-	-	-
24	240904	-	-	-	-	-	-	-
23	240904	-	-	-	-	-	-	-
22	240904	-	-	-	-	-	-	-
21	240904	-	-	-	-	-	-	-
20	240904	-	-	-	-	-	-	-
19	240904	-	-	-	-	-	-	-
18	240904	-	-	-	-	-	-	-
17	240904	-	-	-	-	-	-	-

Figuur 7.7 Screenshot 'incidentmelding globaal' scherm

In figuur 7.7 is te zien dat het uiteindelijke scherm, behoorlijk overeenkomt met de eerder gemaakte schets. Tevens is in dit scherm de use case 'incidentmelding sorteren' (zie bijlage definitiestudie, systeemconcept opgenomen. Dit is gedaan omdat tijdens het ontwikkelen van de

schermen, erg onlogisch bleek deze use case in een apart scherm op te nemen. Dit zou ten kosten gaan van de overzichtelijkheid en gebruiksvriendelijkheid van het systeem. Deze functionaliteit maakt het mogelijk incidentmelding te sorteren op basis van hun status, de betreffende relatie en de verantwoordelijke werknemer. Wanneer de gebruiker de incidentmeldingen wilt sorteren op status, doet hij dit door een status te selecteren uit de listbox en vervolgens op 'ok' te klikken. Nu worden alle incidentmeldingen met de betreffende status getoond. Rechts bovenin het scherm bevindt zich een knop met daarop het Microsoft Excel icoon. Als de gebruiker op deze knop druk wordt het globale incidentmelding overzicht geëxporteerd naar een Excel bestand. Op dit Excel bestand kunnen dan vervolgens weer bewerkingen worden uitgevoerd, zoals het generen van grafieken en het uitvoeren van berekeningen. Hieronder is de code van deze functie te zien:

```
<?php
header("Content-type: application/octet-stream");
header("Content-Disposition: attachment; filename=alle meldingen.xls");
header("Pragma: no-cache");
header("Expires: 0");
include_once("../inc/dblogin.php");
$rDbConn = connectdb();
$result = mysql_query('select * from melding ORDER BY Meldingsnummer');
$count = mysql_num_fields($result);
for ($i = 0; $i < $count; $i++){
    $header .= mysql_field_name($result, $i)."\t";
}
while($row = mysql_fetch_row($result)){
    $line = "";
    foreach($row as $value){
        if(!isset($value) || $value == ""){
            $value = "\t";
        }else{
            $value = str_replace("'", "''", $value);
            $value = "'" . $value . "'\t";
        }
        $line .= $value;
    }
    $data .= trim($line)."\n";
}
$data = str_replace("\r", "", $data);
if ($data == "") {
    $data = "\nno matching records found\n";
}
echo $header."\n".$data;
?>
```

De uitvoer die deze code oplevert, is hieronder te zien in figuur 7.8.

Meldingsnummer	Persoon_id	Soort	Omschrijving	Datum	Status	Lijn	Interne_prioriteit	Externe_prioriteit
1	1	Maak_selectie		12-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
2	9	1	Maak_selectie	12-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
3	10	1	Maak_selectie	12-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
4	9	1	Maak_selectie	12-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
5	10	1	Maak_selectie	12-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
6	11	1	Maak_selectie	12-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
7	12	1	Maak_selectie	0000-00-00	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
8	13	1	Maak_selectie	0000-00-00	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
9	14	1	Maak_selectie	0000-00-00	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
10	15	3	Maak_selectie	0000-00-00	Maak_selectie	Maak_selectie	Maak_selectie	Hoog
11	16	2	Klacht	17-8-2004	Nog_niet_in_behandel	Tweede	Middel	Laag
12	17	2	Storing	17-8-2004	In_behandeling	Tweede	Laag	Laag
13	18	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
14	19	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
15	20	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
16	21	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
17	22	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
18	23	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
19	24	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
20	25	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
21	26	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
22	27	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
23	28	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
24	29	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
25	30	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
26	31	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
27	32	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
28	33	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
29	34	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
30	35	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
31	36	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
32	37	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
33	38	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
34	39	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
35	40	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
36	41	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
37	42	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
38	43	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
39	44	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
40	45	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
41	46	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie
42	47	1	Maak_selectie	17-8-2004	Maak_selectie	Maak_selectie	Maak_selectie	Maak_selectie

Figuur 7.8 Uitvoer incidentmeldingen in Excel

Achter het scherm uit figuur 7.7 zit een PHP code. Deze code bestaat uit verschillende onderdelen. Een deel van de code die ervoor zorgt dat de tabel met incidentmeldingen wordt opgebouwd en gevuld, zal nu verder uitgediept worden. Ten eerste moeten alle benodigde gegevens uit de tabel melding zelf worden gehaald en in de variabele \$tabel worden gezet. Dit wordt gedaan met behulp van de volgende SQL query:

```
$query = mysql_query("SELECT * FROM melding ORDER BY meldingsnummer desc")
while($tabel = mysql_fetch_array($query)){}
```

Vervolgens moeten alle waardes uit de vreemde tabellen worden opgehaald, die aan de tabel melding zijn gekoppeld doormiddel van vreemde sleutels. Voor de waarde uit de tabel categorie, wordt dit gedaan met behulp van de volgende SQL query:

```
$sql1 = mysql_query("SELECT Categorienummer, Naam FROM categorie WHERE Categorienummer = '$tabel[Categorienummer]'");
$fetch1 = mysql_fetch_assoc($sql1);
```

Hierna moeten de verschillende tabelkoppen worden gegenereerd. Ter illustratie is hieronder de code te zien, die de tabelkoppen van het 'meldingsnummer' en 'categorie' genereert:

```
echo "<td bgcolor=#0000CC><font color=\"#FFFFFF\" ><b> Numme r</b> </td> <td bgcolor=#0000CC> <font color=\"#FFFFFF\"><b>Categorie</b></td>
```

Tot slot dient de tabel gevuld te worden met de opgehaalde gegevens. Dit wordt voor de waardes 'meldingsnummer' en 'categorie' met behulp van de onderstaande code gedaan:

```
echo "<tr><td bgcolor=#0000CC <font color= orange >$tabel[Meldingsnummer] </font > </strong> </center> </a> </td> <td bgcolor=#0000CC> <font color = #FFFFFF bgcolor=#0000CC >fetch1[Naam] </font> </td></tr>
```

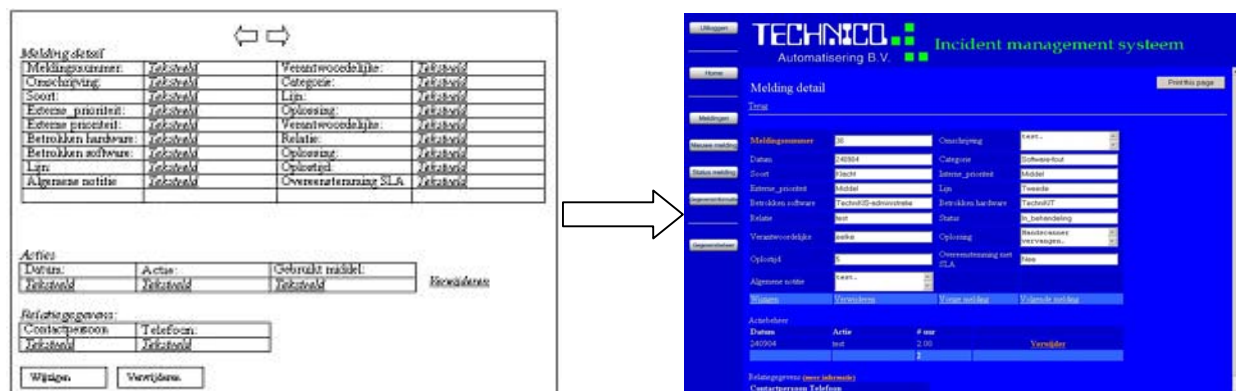
Het oranje gedrukte meldingsnummer in de tabel in figuur 7.7, dient als een koppeling naar het 'incidentmelding detail scherm'. In dit scherm krijgt de gebruiker de detailgegevens van de betreffende incidentmelding te zien. Deze koppeling is gerealiseerd door het volgende stukje code:

```
<ahref="Melding_detail.php?meldingX=$tabel[Meldingsnummer]"> <center> <strong> <font color= orange>
>$tabel[Meldingsnummer] </font></strong></center>
```

In dit stukje code, wordt het meldingsnummer waarop de gebruiker klikt, opgeslagen in de variabele 'MeldingX'. In het 'incidentmelding detail scherm' wordt deze variabele vervolgens weer aangeroepen met behulp van het onderstaande stukje code:

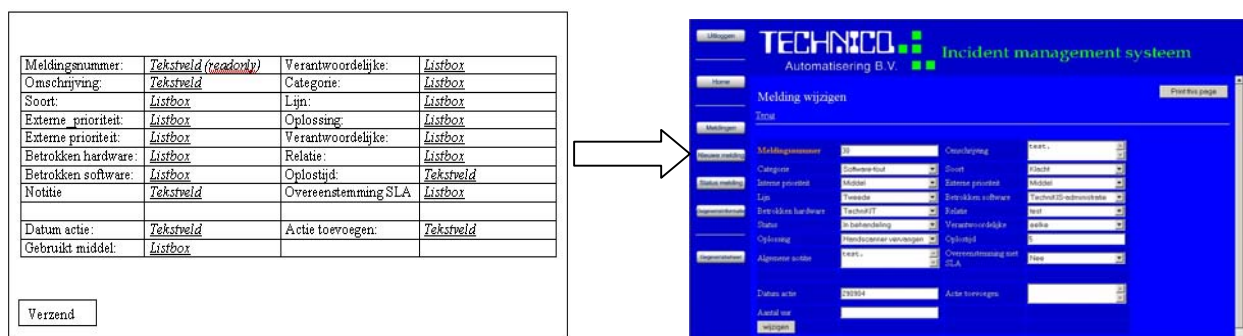
```
$query = mysql_query("SELECT * FROM Melding WHERE Meldingsnummer ='$meldingX'");
```

De schets en het uiteindelijke scherm van het 'incidentmelding detailscherm' en de bijbehorende functionaliteiten, zijn te zien in figuur 7.9.



Figuur 7.9 Incidentmelding detailscherm, van schets naar scherm

In dit scherm worden alle gegevens van een incidentmelding weergegeven. In de tabel 'melding detail' staan alle gegevens die in de klasse melding zelf zijn opgeslagen en de gegevens uit een aantal gekoppelde tabellen. De schets en het uiteindelijke scherm van het 'incidentmelding wijzigen scherm' en de bijbehorende functionaliteiten, zijn te zien in figuur 7.10.



Figuur 7.10 Incidentmelding wijzigen scherm, van schets naar scherm

In dit scherm kunnen alle geregistreerde gegevens van een melding worden gewijzigd. Tevens kunnen in dit scherm acties die werknemers hebben ondernomen voor het oplossen van het incident worden geregistreerd. Voor het registreren van een actie dienen de datum (default is de datum van de betreffende dag), een omschrijving en het aantal uur te worden ingevoerd. De opdrachtgever vond het registreren van acties een belangrijk onderdeel van het systeem omdat het in de toekomst onder andere de deur opende naar automatische facturering van de besteedde uren.

Het ontwikkelen van de functionaliteit voor het wijzigen van een incidentmelding is niet in één keer geslaagd. De afstudeerder zat met het probleem dat wanneer een gebruiker een incidentmelding wilde wijzigen, alle listboxen automatisch op hun standaardwaarden sprongen. Dit hield dus in dat de gebruiker alle listboxen telkens weer op hun correcte waarden moest zetten, ook al wilde hij er maar eentje wijzigen. De kennis van de afstudeerder op het gebied van PHP schoot hier tekort. Nadat hij het probleem ook niet met behulp van internet kon oplossen, heeft hij de hulp van een aantal studiegenoten ingeschakeld. Alhoewel zij geen directe oplossing konden aandragen voor het betreffende probleem, heeft hun expertise zeker wel bijgedragen bij het vinden van de uiteindelijke oplossing. Pas in de derde iteratie van deze pilot, is het probleem daadwerkelijk verholpen. Omdat de afstudeerder toch best wel trots is op het proces dat hij hier heeft doorlopen en de oplossing waartoe hij uiteindelijk is gekomen, is de PHP code voor het wijzigen van het attribuut oplossing, in interne bijlage 4 van dit document opgenomen. Voor de degene met weinig ervaring op het gebied van PHP, is de code voorzien van commentaar.

Nadat de schetsen van de schermen waren afgerond, kon worden begonnen aan het pilotontwikkelplan van de pilot 'meldingsysteem'.

7.2.3 Opstellen pilotontwikkelplan meldingsysteem

In dit onderdeel is het pilotontwikkelplan van de pilot meldingsysteem opgesteld. Dit pilotontwikkelplan is opgesteld op basis van de eerdere gemaakte globale functionele en organisatorische specificaties. Vanwege rapporttechnische zaken, zijn een aantal onderdelen van het pilotontwikkelplan al eerder besproken in deze paragraaf. Tijdens het opstellen van dit onderdeel moest in de eerste instantie worden nagedacht over de opdeling van de pilot in verschillende pilotdelen. Hierna moesten deze worden ingepland en vervolgens worden geprioriteerd. De afstudeerder zelf, maar ook de opdrachtgever waren van mening dat alle onderdelen van deze pilot even belangrijk waren en sowieso ontwikkeld moesten worden. Als één van de onderdelen zou ontbreken, dan kon het incident management proces niet in zijn volledigheid worden uitgevoerd. Om deze reden kregen alle pilotdelen de prioriteit 'basis'. Tevens bood het pilotontwikkelplan de afstudeerder de gelegenheid, na te denken over de te volgen testprocedure en de invoering. Al in de aanpak van dit project was besloten dat het uiteindelijk ontwikkelde systeem niet zou worden ingevoerd binnen de afstudeerperiode. Wel zou het, wanneer het volledig ontwikkeld was worden getest. Om toch tussentijds de kwaliteit van deze pilot te kunnen garanderen, werd besloten deze pilot te testen doormiddel van een 'vrije test'. In een vrije test wordt een toekomstige eindgebruiker uitgedaagd fouten te ontdekken in de ontwikkelde pilot door als het ware met de functionaliteiten te spelen. De gebruikers die als tester fungeerde, kregen van de afstudeerder wel schriftelijke aanwijzingen mee waarbij ze tijdens het testen van deze pilot op moesten letten. In elke iteratie, werd deze test minimaal door twee gebruikers uitgevoerd. Dit leverde niet alleen een lijst op met allerlei mankementen aan de pilot, maar leverde ook input voor de volgende iteratie. De afstudeerder was op het idee van het vrije testen gekomen door een medestudent die parttime werkte als softwaretester en deze manier van testen regelmatig gebruikte in zijn dagelijkse werkzaamheden.

7.2.4 Verificatie en validatie meldingsysteem

In dit onderdeel wordt de verificatie en validatie van de pilot 'meldingsysteem' beschreven. Activiteiten met betrekking tot verificatie en validatie zijn erop gericht te verzekeren dat de systeemontwikkeling op de juiste manier wordt uitgevoerd met de juiste resultaten. In het kader van de verificatie en validatie van de pilot 'meldingsysteem', zijn de use cases die van toepassing waren op deze pilot, tegen de betreffende systeemeisen en schermschetsen afgezet. De conclusie die uit deze verificatie kon worden getrokken, was dat alle functionele systeemeisen uit de definitiestudie uiteindelijk ook in de pilot meldingsysteem waren verwerkt. De verificatie en validatie tabellen van deze pilot, zijn terug te vinden in externe bijlage drie, het pilotontwikkelrapport meldingsysteem. Telkens nadat de pilot meldingsysteem door de iteraties heen was geversificeerd en gevalideerd, kon er verder worden gegaan met de volgende pilot, het gegevensbeheer.

7.3 Pilot 3: Gegevensbeheer

Tijdens dit onderdeel is de pilot 'gegevensbeheer' ontwikkeld. De pilot 'gegevensbeheer' moest aan verschillende eisen voldoen welke zijn terug te vinden in de systeemeisen van de definitiestudie. De geplande tijd voor deze pilot was geschat op 15 dagen, conform het pilotplan in de definitiestudie. Deze pilot omvatte alle functionaliteiten omtrent het beheren van de gegevens die gekoppeld konden worden aan een incidentmelding. Overigens waren de functionaliteit uit deze pilot grotendeels alleen toegankelijk voor de actor beheerder. Alleen enkele gegevensoverzichten waren ook beschikbaar voor de actor medewerker. In de pilot 'gegevensbeheer', zijn de volgende onderdelen gerealiseerd:

- Pilotontwikkeldrapport gegevensbeheer
 - Plan van aanpak
 - Functionele structuur
 - Organisatorische inrichting
 - Pilotontwikkelpun
 - Verificatie en validatie
- Pilot 3 'gegevensbeheer'

Omdat de pilot meldingsstelsam in de vorige paragraaf uitgebreid is behandeld, zal in deze paragraaf de laatste pilot, gegevensbeheer, een stuk beknopter behandeld worden. Er zullen een aantal relevante zaken uit de globaal functionele structuur worden behandeld. Tevens zal er beknopt worden ingegaan op de overige activiteiten die voor deze pilot zijn uitgevoerd.

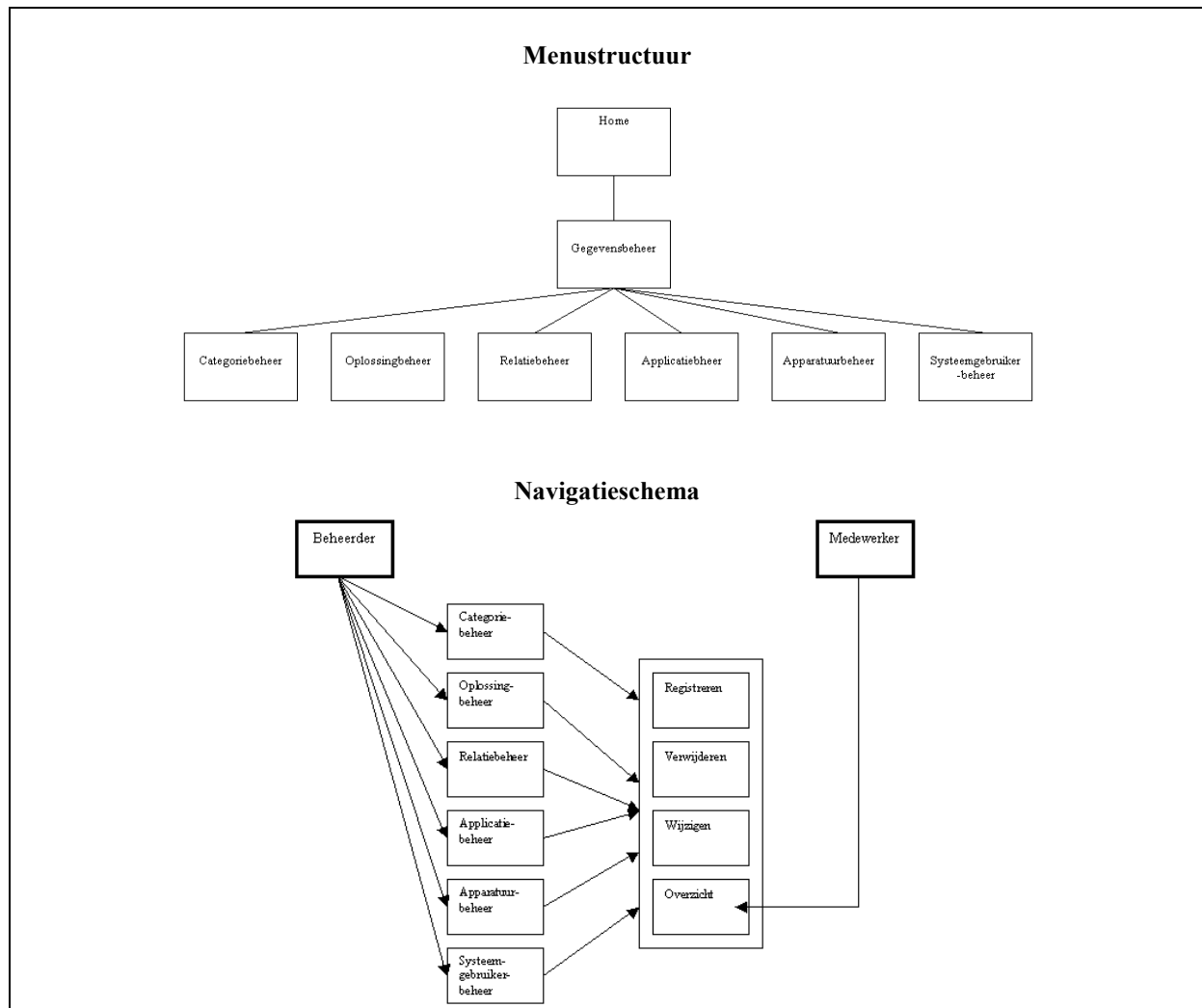
In het plan van aanpak is gespecificeerd welke activiteiten er in deze fase zouden worden uitgevoerd, uit welke onderdelen het pilotontwikkeldrapport van de pilot gegevensbeheer zou gaan bestaan en hoeveel tijd deze onderdelen in beslag zouden gaan nemen. Voor een gedetailleerde beschrijving van veranderingen en opbouw van de planning die gedurende dit project is gebruikt wordt doorverwezen naar paragraaf 4.2 van dit document.

Het bepalen van de te gebruiken use cases is deze pilot was een eenvoudige taak, dit waren namelijk alle use cases die waren overgebleven na de ontwikkeling van de pilot meldingsstelsam. Al deze use case betroffen functionaliteiten voor het beheren van de gegevens die aan een incidentmelding konden worden gekoppeld. Wederom waren net zoals in de voorgaande pilot, de verschillende use cases samengevoegd tot functionele clusters. Deze clusters zouden later de verschillende pilotdelen vormen. Bij het clusteren werd gelet op de gelijkenis in functionaliteit van de use cases. De use case oplossing registreren, oplossing wijzigen en oplossing verwijderen werden bijvoorbeeld samengevoegd tot de cluster 'oplossingsbeheer'. Hieronder volgt de lijst van clusters die het samenvoegen van de betreffende use cases heeft opgeleverd:

- Categoriebeheer: in dit onderdeel kan binnen het stelsam een nieuwe categorie geregistreerd, een bestaande gewijzigd of verwijderd worden en een overzicht worden bekeken.
- Oplossingbeheer: in dit onderdeel kan binnen het stelsam een nieuwe oplossing geregistreerd, een bestaande gewijzigd of verwijderd worden en een overzicht worden bekeken.
- Relatiebeheer: in dit onderdeel kan binnen het stelsam een nieuwe relatie geregistreerd, een bestaande gewijzigd of verwijderd worden en een overzicht worden bekeken.
- Applicatiebeheer: in dit onderdeel kan binnen het stelsam een nieuwe applicatie geregistreerd, een bestaande gewijzigd of verwijderd worden en een overzicht worden bekeken.
- Apparatuurbeheer: in dit onderdeel kan binnen het stelsam een nieuw apparaat geregistreerd, een bestaande gewijzigd of verwijderd worden en een overzicht worden bekeken.
- Stelsamgebruiker beheer in dit onderdeel kan binnen het stelsam een nieuwe stelsamgebruiker geregistreerd, een bestaande gewijzigd of verwijderd worden en een overzicht worden bekeken.

Nadat alle use cases waren geclusterd, kon er met het opstellen van de menustructuur en het navigatieschema begonnen worden. De procesgang die zich bij het opstellen van deze twee modellen heeft afgespeeld vertoond in grote mate gelijkenis met de procesgang die zich in de pilot

meldingsysteem heeft afgespeeld en zal hier daarom niet verder worden besproken. In figuur 7.11 zijn respectievelijk de menustructuur en het navigatieschema van de pilot gegevensbeheer te zien.



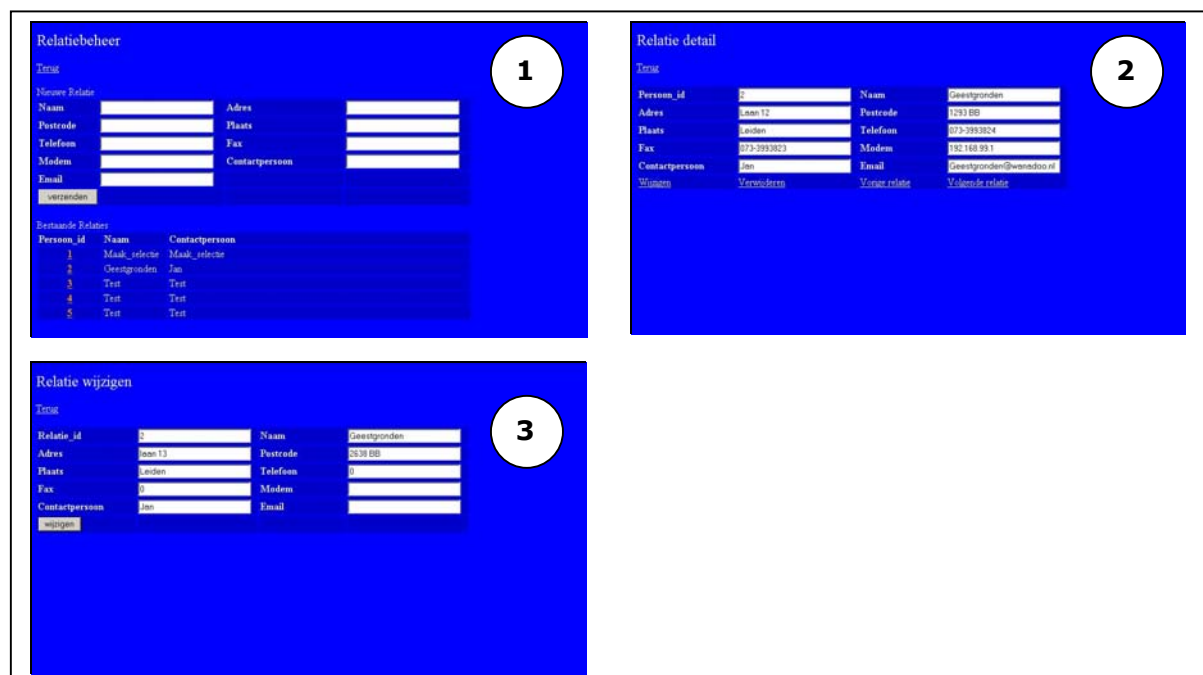
Figuur 7.11 Menustructuur en navigatieschema gegevensbeheer

Nadat de hiërarchie van de schermen en de navigatie ertussen vastgesteld was, kon er worden begonnen met het ontwerpen van de schermen. Concrete use cases en een duidelijke hiërarchie en navigatiestructuur, maakte het voor de afstudeerder mogelijk, de eerste schetsen van de schermen letterlijk op papier te zetten. Om doormiddel van deze schetsen de lezer een representatief beeld van deze pilot te geven, is er gekozen de cluster 'relatiebeheer' te behandelen. Deze cluster kan als representatief voor deze pilot worden beschouwd aangezien de rest van de clusters soortgelijke functionaliteiten bevat. Allereerst is in figuur 7.12 in een screenshot van het hoofdmenu van gegevensbeheer te zien.



Figuur 7.12 Screenshot hoofdmenu gegevensbeheer

Dit screenshot is in dit document opgenomen om de lezer een indruk te geven van de omvang van deze pilot. In het screenshot is te zien dat er onder het kopje relatiebeheer twee opties staan. De knop 'alle relaties' leidt naar een lijst met alle relaties die in de database opgeslagen staan. Dit is de optie waar ook de medewerker toegang toe heeft. De optie 'relatiebeheer' leidt tot de mogelijkheden voor het beheeren van de relaties. Wanneer de gebruiker hierop klikt, komt hij terecht in de schermen waarin de relaties daadwerkelijk beheerd kunnen worden. Deze schermen zijn te zien in figuur 7.13.



Figuur 7.13 Screenshots relatiebeheer

In het eerste scherm uit figuur 7.13 is een screenshot te zien met daarin de functionaliteit voor het registreren van een nieuwe relatie. Tevens is er een lijst in opgenomen met daarin een deel van de

gegevens van de geregistreerde relaties. De oranje relatienummers in deze lijst, zijn de primaire sleutels van de verschillende relaties. Door hierop te klikken, komt de gebruiker terecht in het detailscherm van de betreffende relaties. Dit detailscherm is in het tweede screenshot van figuur 7.13 te zien. In dit scherm staan alle geregistreerde gegevens van de relatie weergegeven. Vanuit dit scherm beschikt de gebruiker over verschillende opties. Hij kan de relatie wijzigen en verwijderen. Tevens kan de gebruiker hier tussen de verschillende relaties bladeren. Wanneer de gebruiker voor de optie wijzigen kiest, komt hij terecht in het derde screenshot in figuur 7.13. Hier kunnen alle reeds eerder geregistreerde gegevens gewijzigd worden, met uitzondering van `persoon_id`. Aangezien deze primaire sleutel uniek dient te zijn binnen het systeem, mag hij niet gewijzigd worden.

Nadat alle schetsen van de schermen voor deze pilot voltooid waren, kon verder worden gegaan met het opstellen van het pilotontwikkelpun en het versificeren en valideren van deze pilot. Het proces dat hierbij is gevolgd, is vergelijkbaar met die van de procesgang die bij de het ontwikkelen van dit onderdeel voor de pilot meldingsysteem is gevolgd. Om deze reden zal er hier verder niet meer op in worden gegaan. Nadat deze pilot twee iteraties had doorlopen, was de ontwikkeling van het systeem afgerond en kon er begonnen worden met het testen van het incident management-systeem.

8. Testen van het systeem

In dit hoofdstuk zal de procesgang van de testen van het incident managementsysteem worden beschreven. Om na het testen eventuele naar bovengekomen mankementen aan het systeem te kunnen verhelpen, waren hier in dit onderdeel twee dagen voor ingepland. Drie volle dagen waren gereserveerd voor het testen zelf.

Voor er aan het daadwerkelijke testen van het systeem kon worden begonnen, was het belangrijk te weten hoe er getest ging worden, wat er getest ging worden en of het hele systeem met deze test werd afgedekt. Er werd door de afstudeerder ervoor gekozen het systeem 'black box' te gaan testen. Black box testen wordt ook wel functioneel testen genoemd. Bij deze vorm van testen wordt er gekeken naar de invoer in het systeem en de gerelateerde uitvoer. Als de daadwerkelijk uitvoer anders is, dan is er een fout gevonden in het systeem. Omdat het incident managementsysteem opgebouwd was uit een aantal zeer concrete functionaliteiten, was black box testen de meest geschikte methode.

Om zeker te weten dat alle onderdelen van het systeem aan bod kwamen in de test, werd het systeem opgedeeld in verschillende deelprocessen.

- Het registreren van een incidentmelding
- Het wijzigen van een incidentmelding
- Het verwijderen van een incidentmelding
- Het opvragen van de status van een incidentmelding
- Registreren van een actie
- Het toevoegen, wijzigen en verwijderen van de volgende gegevens:
 - Categorie
 - Oplossing
 - Software
 - Randapparatuur
 - Relatie
 - Systeemgebruiker

Nadat deze processen waren vastgesteld, was bekend hoe er getest ging worden en wat er getest ging worden. Nu was het tijd op de processen te beschrijven in de vorm van testcases. In deze testcases stond beschreven, wat de tester diende in te voeren en wat de verwachte uitvoer van het systeem was. Het was de taak van de tester om te controleren of de daadwerkelijke uitvoer gelijk was aan de verwachte uitvoer. In de onderstaande tabel is het resultaat van de testcase 'incidentmelding registreren' te zien.

Naam invoerveld	Soort invoerveld	Invoer	Verwachte uitvoer	Daadwerkelijke uitvoer
Omschrijving	Textarea	Handscanner heeft kuren.	Handscanner heeft kuren.	Handscanner heeft kuren.
Soort	Listbox	Storing	Storing	Storing
Externe prioriteit	Listbox	Hoog	Hoog	Hoog
Betrokken software	Listbox	Technikis Relatie	Technikis Relatie	Technikis Relatie
Relatie	Listbox	De Geestgronden	De Geestgronden	De Geestgronden
Verantwoordelijke	Listbox	Henk	Henk	Henk
Oplostijd	Textveld	5	5	5
Overeenstemming met SLA	Listbox	Ja	Ja	Ja
Categorie	Listbox	Gebruikersfout	Gebruikersfout	Gebruikersfout
Interne prioriteit	Listbox	Laag	Laag	Laag
Lijn	Listbox	Derde	Derde	Derde
Betrokken hardware	Listbox	Handscanner	Handscanner	Handscanner
Status	Listbox	In behandeling	In behandeling	In behandeling
Oplossing	Listbox	Handscanner opnieuw installeren	Handscanner opnieuw installeren	Handscanner opnieuw installeren
Algemene notitie	Textarea	Geen.	Geen.	Geen.

De conclusie die in het geval van deze testcase kon worden getrokken, was dat het registreren van een incidentmelding volledig naar behoren werkten.

Om een zo goed mogelijk testresultaat te krijgen, was besloten het systeem te laten testen door twee toekomstige gebruikers. Na dat zij de het testplan de eerste keren waren doorlopen, hadden

ze beide een lijst opgesteld van een aantal onderdelen van het systeem die een onverwachte uitvoer opleverde. De lijst was bij beide testers overigens identiek. Dit leidde tot de conclusie dat het testplan correct was uitgevoerd door de testers. Nadat alle geconstateerde problemen in het systeem waren opgelost, werd het testplan voor de laatste keer door de afstudeerder zelf uitgevoerd. Hierbij werden geen fouten meer geconstateerd in het systeem. Tot slot is er door de afstudeerder nog gecontroleerd, of het incident managementsysteem voldeed aan de non-functionele systeemeisen uit de definitiestudie. Er werd hier onder andere naar de compatibiliteit en responsetijd van het systeem gekeken. Nadat bevestigd was dat het systeem aan alle gestelde eisen voldeed, kon het als afgerond worden beschouwd.

9. Evaluatie

Dit hoofdstuk betreft de evaluatie van het doorlopen ontwikkeltraject. De evaluatie is belangrijk om te kunnen reflecteren op de uitgevoerde activiteiten tijdens dit project. Waarom deze zijn gedaan, hoe deze zijn gedaan en hadden ze misschien anders gekund? In dit hoofdstuk zullen zowel het proces als het opgeleverde product worden geëvalueerd. Tevens zullen de behaalde leerwinsten van de afstudeerder worden beschreven. Tot slot zal worden beschreven of deze afstudeerstage een succes was in de vorm van een conclusie. Om de evaluatie een wat persoonlijker tint te geven, is het in tegenstelling tot de rest van dit rapport, in de ikvorm geschreven.

9.1 De evaluatie van het proces

In deze paragraaf zal er naar een aantal aspecten van het doorlopen proces gekeken worden. Allereerst zal er worden teruggeblikt over hoe de afstudeerder het contact met zijn opdrachtgevers en andere collega's heeft ervaren. Vervolgens zal de manier van plannen en het werken met IAD geëvalueerd worden. Tot slot zal er worden teruggekeken naar de doorlopen procesgang bij het opstellen van de veranderingsanalyse en de definitiestudie, het ontwikkelen van de drie pilots en het testen van het incident managementsysteem.

Tijdens deze afstudeerperiode heb ik de beschikking gehad over een bedrijfsmentor die tevens als mijn opdrachtgever fungeerde. De wensen en eisen van mijn bedrijfsmentor, hebben gedurende het gehele project centraal gestaan. Deze wensen en eisen waren echter niet de enigen waar ik rekening mee moest houden. Er waren ook wensen van een aantal andere toekomstige gebruikers. Het was mijn taak alle belanghebbende van elkaars eisen en wensen constant op de hoogte te houden. De lukt altijd vrij goed. De relatie die ik had met mijn bedrijfsmentor en overige collega's, was vrij informeel en dat beviel me goed. Hoewel ik wekelijks een formeel contactmoment had met mijn bedrijfsmentor, zijn het toch de vele informele contacten met de opdrachtgever en collega's, die het meest hebben bijgedragen aan het welslagen van dit project. Wat mij ook erg beviel aan de organisatie, was het vrije en informele karakter dat veel ruimte bood voor persoonlijke ontwikkeling en het nemen van initiatief. Dit tweede werd dan ook erg gewaardeerd door mijn bedrijfsmentor.

Het hele gebeuren rondom het plannen van dit project is mij in tegenstelling tot eerdere projecten die ik heb uitgevoerd, goed vergaan. In het beginstadium van het project had ik in de eerste instantie op basis van de globale planning in het algemene plan van aanpak, een detailplanning voor de gehele afstudeerperiode opgesteld. Ik ondervond al snel dat het moeilijk was een project van deze omvang zover vooruit te plannen. Dit merkte ik aan het feit dat ik deze bijna dagelijks moest bijstellen. De in paragraaf 4.2 genoemde tussenoplossing, beschouwde ik dan ook als een gouden vondst, die voor eens en voor altijd een eind had gemaakt aan mijn problemen met het plannen. De tussenoplossing bood zeker wel structuur aan de werkzaamheden maar liet tevens ruimte open voor ongeziene gevallen. Na de introductie van de tussenoplossing was het onderhouden van de planning een stuk minder arbeidsintensief en de planning zelf een stuk realistischer.

Gedurende dit project heb ik gebruik gemaakt van de ontwikkelmethodiek IAD. Doordat ik IAD al in een aantal eerdere projecten als ontwikkelmethode had gebruikt, voelde het gebruik ervan tijdens dit project erg vertrouwd en natuurlijk aan. Ik ben van mening dat ik er goed aan heb gedaan IAD voor dit project te gebruiken. Niet alleen vanwege de ervaring die ik er mee had, maar ook omdat het me veel meer tijd had gekost als ik een nieuwe ontwikkelstrategie had moeten aanleren. Tevens heeft IAD het plannen van het project een stuk eenvoudiger gemaakt, omdat ik al bekend was met de omvang en essentie van de onderdelen uit de ontwikkelmethode. Toch zal ik me de volgende keer bij een soortgelijke opdracht beter oriënteren op de markt van ontwikkelstrategieën. Ik ben namelijk van mening, dat het gebruik van IAD het best werkt bij projecten waaraan door een projectgroep wordt gewerkt. De grote hoeveelheid documentatie die met het gebruik van IAD wordt gegenereerd, leek soms een beetje inefficiënt.

Na het gesprek met de examinatoren (zie paragraaf 4.2), was ik tot de conclusie gekomen, dat ik om tot de beste mogelijke oplossing te kunnen komen voor het aanwezige probleem binnen de organisatie op het gebied van incident management, er een veranderingsanalyse moest worden uitgevoerd. Tevens werd besloten dat de structuur van incident management, zoals ITIL die

voorschrijft, een grotere rol moest spelen bij de ontwikkeling van het incident managementsysteem. Ik besloot de eerste iteratie, waar ik me op dat moment in bevond, af te breken en een direct de tweede iteratie in te gaan om vervolgens als eerste de veranderingsanalyse uit te voeren. Tegen de tijd dat ik met de veranderingsanalyse begon, was ik al zo'n acht weken bezig met de afstudeeropdracht. Ik had de uit te voeren opdracht al redelijk voor ogen. De veranderingsanalyse was eigenlijk een activiteit, waarover ik in de eerste iteratie eigenlijk al min of meer informeel over had nagedacht. De oplossing die centraal stond voor dit project, was immers niet uit de lucht komen vallen. Wel bood de veranderingsanalyse mij de gelegenheid, kritisch naar de organisatie te kijken en een op concrete feiten gebaseerde oplossing te formuleren. Op dit vlak heeft de veranderingsanalyse dan ook zijn waarde bewezen. In toekomstige projecten zal ik wanneer nodig, de veranderingsanalyse zeer zeker direct bij de aanvang van het project uitvoeren. Je beschikt dan nog over een objectief blik op de opdracht en bent dan veel beter in staat een bruikbare oplossing op te stellen.

De definitiestudie met daarin de systeemeisen en het systeemconcept, heeft een centrale rol gespeeld tijdens dit project. Door het gebruik van het paragraaf 6.3.1 genoemde prototype, stonden er al snel concrete systeemeisen op papier wat vervolgens al snel leidde tot een realistisch systeemconcept. Het ontwikkelen van een prototype voor het achterhalen van de systeemeisen, niet alleen bij de opdrachtgever maar ook bij de toekomstige eindgebruikers, was mij tweede gouden vondst tijdens dit project. Ik had het prototype opgesteld omdat de wensen en eisen betreffende het systeem in de eerste instantie nog erg onduidelijk waren. Het prototype heeft in dit project zeer zeker zijn nut bewezen. Bij een volgend vergelijkbaar project, zal ik dan ook weer dankbaar gebruik maken van deze techniek.

Bij het ontwikkelen van het incident managementsysteem zijn de pilotontwikkeldrapporten van essentieel belang geweest. Als ik nu terug kijk op de pilots waarin het systeem was verdeeld, denk ik dat ik hierin in een goede keuze heb gemaakt. De drie pilots waren in omvang en complexiteit, elk goed te ontwikkelen en volgde elkaar op een logische wijze op. De pilots database en meldingsstelsysteem hebben elk drie iteraties, de pilot gegevensbeheer maar twee. Achteraf bleken de twee iteraties, voldoende te zijn voor het ontwikkelen van de pilot gegevensbeheer. Dit werd mede veroorzaakt door het feit dat ik bij de start van deze pilot al een behoorlijke kennis had opgebouwd van PHP. Tevens bleek de complexiteit van deze pilot mee te vallen in verhouding tot de pilot meldingsstelsysteem. Tot slot kan gesteld worden dat deze fase zonder al te grote problemen is verlopen.

9.2 De evaluatie van het product

In deze paragraaf zal het ontwikkelde incident managementsysteem geëvalueerd worden. Er zal worden of het systeem dat Technico Automatisering vijf maanden geleden voor ogen had, ook daadwerkelijk ontwikkeld is.

Ik denk dat het incident managementsysteem dat ik ontwikkeld heb, ik grote mate overeenkomt met de systeem dat Technico Automatisering vijf maanden geleden voor ogen had. Dankzij het systeem, zullen nu niet alleen alle incidenten geregistreerd worden, maar zal er tevens structuur worden geboden aan deze activiteit. Dankzij het systeem kan nu begonnen worden met het opbouwen van een historisch overzicht van incidentmeldingen.

Toen ik ongeveer vier maanden geleden mijn eerste PHP code invoerde, had ik niet gedacht in staat te zijn het systeem te bouwen dat ik nu heb gebouwd. Programmeren is namelijk nooit mijn sterkste kant geweest. Door thuis in de avondurtjes veel te oefenen met PHP, ben ik er toch ik geslaagd de programmeertaal relatief snel onder de knie te krijgen. Toen mijn PHP code geleidelijk in functionaliteiten begon te transformeren, begon ik zelfs enige affectie voor de programmeertaal te ontwikkelen, en ja ik begon programmeren zelfs leuk te vinden.

Als ik mezelf de vraag stel, kan het incident management nu wat het moest gaan doen, kan ik hier als antwoord ongetwijfeld een ja op geven. Als ik mezelf de vraag stel, gaat het systeem ook doen wat het kan doen, kan ik hier als antwoord alleen maar opgeven dat ik dit van harte hoop. Al in de veranderingsanalyse had ik geconcludeerd dat Technico Automatisering een zeer jonge en informele organisatie is waar de werknemers zeer 'vrij van geest' zijn. Ik heb dan ook geprobeerd, de drempel naar een beter en gestructureerder incident managementproces zo laag mogelijk te maken. Ik denk zelf dat er hier behoorlijk in geslaagd ben en een goede basis heb gelegd voor incident management binnen Technico Automatisering.

9.3 De behaalde leerwinsten

Tijdens het afstuderen heb ik veel geleerd. Uiteraard heb ik veel ervaring opgedaan op het gebied van projectmatig werken en het werken met PHP en HTML. Het feit dat ik veel initiatief heb moeten nemen door de opdrachtgever en werknemers zo intensief mogelijk bij het proces te betrekken, is voor mij een goed leermoment geweest. Nog nooit eerder heb ik een project, deze mate van initiatief hoeven te tonen. Ook op het gebied van effectieve communicatie heb ik een hoop geleerd en heb ik gemerkt dat informele communicatie een zeer belangrijke rol speelde in de communicatie tussen mij en de opdrachtgever. Ik merk nu dat ik door mijn afstudeerstage ontwikkeld ben tot een beter persoon. Ik durf in mijn professionele en persoonlijk leven nu veel meer initiatief te nemen, en ga uitdagingen minder snel uit de weg.

9.4 De conclusie

Ik heb het de afgelopen vijf maanden erg naar mijn zin gehad bij Technico Automatisering. De opdracht die ik voor hun in het kader van het afstuderen heb uitgevoerd, vond ik leuk en interessant om te doen. Niet alleen ik ben tevreden over de uitkomst van dit project, maar ook de opdrachtgever en de toekomstige gebruikers. De enige conclusie die ik kan trekken is dat het afstuderen een geslaagde onderneming is geweest waar ik met heel veel plezier aan heb gewerkt.

Figurenlijst

Figuur 2.1 Organigram Van Remundt Beheergroep	7
Figuur 4.1 Iteratie strategie Big-Bang-Invoeren	11
Figuur 4.2 Fuik procedure.....	13
Figuur 4.3 Globale planning	14
Figuur 4.4 Tussenoplossing planning.....	15
Figuur 4.5 Detailplanning	15
Figuur 4.6 Doorlopen iteraties.....	16
Figuur 5.1 Activiteitendiagram incident afhandelingproces	18
Figuur 5.2 Organisatietypen van Hofstede	19
Figuur 6.1 Scherm uit prototype.....	24
Figuur 6.2 Globaal overzicht incidentmeldingen	24
Figuur 6.3 Incidentmelding registreren.....	26
Figuur 6.4 Incidentmelding registreren.....	27
Figuur 6.5 Klassediagrammen	28
Figuur 6.6 Functionele systeemeisen v.s. use cases en pilotdelen	31
Figuur 7.1 Laatste versie klassediagram.....	33
Figuur 7.2 PHP Admin invoerveld.....	34
Figuur 7.3 Menustructuur pilot 'meldingsysteem'	38
Figuur 7.4 Navigatieschema pilot 'meldingsysteem'	39
Figuur 7.5 Hoofdscherm incident managementsysteem	40
Figuur 7.6 Schets 'incidentmelding globaal' scherm	41
Figuur 7.7 Screenshot 'incidentmelding globaal' scherm	41
Figuur 7.8 Uitvoer incidentmeldingen in Excel	43
Figuur 7.9 Incidentmelding detailscherm, van schets naar scherm.....	44
Figuur 7.10 Incidentmelding wijzigen scherm, van schets naar scherm	44
Figuur 7.11 Menustructuur en navigatieschema gegevensbeheer.....	47
Figuur 7.12 Screenshot hoofdmenu gegevensbeheer	48
Figuur 7.13 Screenshots relatiebeheer	48

Verklarende woordenlijst

Actor: Een actor is een entiteit die buiten het systeem staat en die direct communiceert met het systeem.

Black box testen: Een testtechniek waarbij er geen kennis van de code van een informatiesysteem wordt vereist. Er wordt alleen gekeken hoe, bij een bepaalde invoer, het product zich gedraagt en wat de uitvoer is.

Definitiestudie: De eerste fase uit de ontwikkelmethode IAD waarin de doelen en beperkingen van het te ontwikkelen systeem worden geanalyseerd.

Dreamweaver: Een 'What You See Is what You Get' editor voor het ontwikkelen van websites.

HTML: Hyper Text Markup Language, een opmaaktaal voor het ontwikkelen van websites.

IAD: Iterative Application Design, een ontwikkelmethode waarin het informatiesysteem tot stand komt via een aantal snel uitgevoerde iteraties van de complete ontwikkelcyclus.

Iteratie: Er is sprake van een iteratie, wanneer een fase uit de ontwikkelmethode IAD meerdere malen wordt uitgevoerd.

ITIL: Information Technology Infrastructure Library, een reeks publicaties van beste praktijkoplossingen op het gebied van beheer van IT.

Incident: Een gebeurtenis die niet tot de standaardoperatie van een service behoort en die een interruptie of een vermindering van de kwaliteit van die service kan veroorzaken.

Incidentmanagement: Het proces dat ervoor moet zorgdragen dat dreigende storingen in IT-diensten zo snel mogelijk worden opgelost zodat gebruikers zo snel mogelijk weer aan het werk kunnen.

Kernelpilot: Een niet functionele subset van het totale systeem.

Klassediagram: Een model dat alle klassen en attributen van de te ontwikkelen database bevat en de onderlinge relaties tussen de verschillende klassen laat zien.

Menustructuur: Een model ter verduidelijking van de hiërarchie van de schermen waarin te zien is waar de verschillende schermen zich in de hiërarchie van het totale systeem bevinden.

MySQL: Een relationeel 'open source' database managementsysteem.

Navigatieschema: Een schema ter verduidelijking van de schermen waarin onderlinge relaties en navigatiemogelijkheden tussen de schermen staan weergegeven.

PHP: (Hypertext PreProcessor), een open source in HTML ingebedde server-side scripttaal en engine.

Pilot: Een functioneel operationele subset van het totale systeem.

Pilotontwikkeling: De tweede fase uit de ontwikkelmethode IAD waarin de verschillende systeemonderdelen (pilots) daadwerkelijk worden ontwikkelt.

RIM: (Relationeel Implementatie Model), een model waarmee een relationele database kan worden geïmplementeerd.

RRM: (Relationeel Representatie Model), een model waarmee een relationele database wordt gerepresenteerd.

UML: Unified Modeling Language, een objectgeoriënteerde taal die een set technieken aanlevert om het beoogde systeem te kunnen modelleren.

Use case: Een beschrijving van een reeks interacties tussen één of meer actoren en het systeem. Het gezichtspunt hierbij is het externe gedrag van het systeem, vanuit het gezichtspunt van de gebruiker.

Veranderingsanalyse: Een activiteit waarin getracht wordt, de best mogelijke oplossing voor een probleem te achterhalen.

Webserver: Een locatie die online benaderbaar is en verantwoordelijk is voor het beantwoorden van verzoeken om webpagina's.

Literatuurlijst

Boeken:

- J. van Bon, IT service management, een introductie, van Haren publishing, september 2002
- Tim Converse & Joyce Park, Het complete handboek PHP 4, Den Haag: Academic Services, december 2003
- Eleanor Pascoe-Samson, Organisatie, besturing en informatie, Deventer: Kluwer Bedrijfs-informatie, september 1999
- R.J.H. Tolido, IAD Het evolutionair ontwikkelen van informatiesystemen, Utrecht: Academic Services, oktober 2001
- Jos Warmer & Anneke Kleppe, Praktisch UML, Amsterdam: Addison Wesley Longman Nederland BV., 1999

Handleidingen:

- Moduleboek inleiding systeemontwikkeling, Haagse Hogeschool, sector informatica
- Moduleboek Rapportagetechnieken, Haagse hogeschool
- Modulehandleiding SO-deel IP-841, Haagse Hogeschool, sector informatica
- Moduleboek inleiding databases, Haagse hogeschool, sector informatica

Websites:

- PHP informatieve website, www.phpfreakz.nl
- PHP homepage, www.php.net

Index

A

Actortype · 26; 40

B

Big bang scenario · 11
 Black box testen · 50; 56

C

Cultuurdimensies · 18; 19

D

Definitiestudie · 11; 56
 Dreamweaver · 10; 41; 56

F

Fuikprocedure · 12; 13
 Functionele structuur · 30; 32; 33; 37; 46

H

HTML · 10; 35; 41; 54; 56

I

IAD · 11; 14; 30; 32; 52; 56
 Incidentmanagement · 56
 Iteratie · 11; 56
 ITIL · 12; 16; 52; 56

K

Klassediagram · 32; 56

M

Menustructuur · 37; 38; 47; 56
 MySQL · 10; 29; 34; 35; 56

N

Navigatieschema · 37; 38; 39; 47; 56

O

Ontwikkelscenario · 32
 Organisatiestructuur · 18; 19
 Organisatorische inrichting · 29; 30; 37

P

PHP · 10; 29; 34; 35; 53; 54; 56
 PHPMyAdmin · 10; 34
 Pilot · 22; 32; 36; 37; 56
 Pilotontwikkeling · 11; 24; 32; 56
 Pilotplan · 30; 32; 36; 46
 Prototype · 23; 24; 25; 53

R

Relationeel Implementatie Model · 56
 Relationeel Representatie Model · 56

S

Stoepfactoren · 20
 systeemconcept · 12; 25; 30; 53
 Systeemconcept · 27; 32; 37
 Systeemeisen · 13; 23; 25; 26; 27; 28; 31;
 36; 46; 53

T

Technische structuur · 29

U

UML · 12; 56

V

Veranderingsanalyse · 16; 17; 19; 52; 53; 57
 Verificatie · 30; 31; 32; 36; 37
 Vrije test · 45

W

Webserver · 29; 57

Interne bijlagen

Interne bijlage 1: De Opdrachtomschrijving

1. Omschrijving afstudeeropdracht

Voor deze opdracht moet er een web-based informatiesysteem ontwikkeld worden. Met dit informatiesysteem moet het mogelijk zijn voor klanten maar ook voor medewerkers zelf om een incident te melden via het systeem. De melding moet vervolgens bij de juiste medewerker van het bedrijf terechtkomen. Als een medewerker een melding binnenkrijgt van een klant, moet hij precies kunnen zien welke softwareversie de klant gebruikt, wat de 'known errors' van deze versie zijn en wat voor licentie de klant er van heeft. Het systeem moet in de toekomst ook andere soorten informatie kunnen gaan beheren voor de medewerkers zoals planningen, urenregistraties en relatiebeheer. Een beheerfunctie die het beheren van onder andere gebruikers en stamgegevens mogelijk maakt, is wenselijk.

2. Inleiding

Technico automatisering is een onderneming die deel uit maakt van de Van Remundt Beheer-groep. Het bedrijf richt zich op activiteiten in de industriële wasindustrie. Vanaf 1985 houdt het zich bezig met:

- service en onderhoud binnen wasserijen;
- levering, installatie en implementatie van diverse hardware voor het reinigingsproces;
- advies voor de inrichting en bedrijfsvoering op productie- en managementniveau binnen de wasserij;
- ontwikkeling, levering, installatie en implementatie van kledingsorteersystemen en managementsoftware voor beheersing van persoonsgebonden kleding.

In nauwe samenwerking met Technico Services zorgt Technico Automatisering voor de ontwikkeling van kledingsorteersystemen en software

3. Probleemstelling

De probleemstelling van deze opdracht is tweeledig. Enerzijds heerst binnen het bedrijf het gevoel dat de meldingen van incidenten niet snel en gestructureerd genoeg kunnen worden opgelost. Anderzijds wordt de gedecentraliseerde opslag van gegevens binnen het bedrijf als probleem ervaren. Binnen Technico Automatisering wordt op verschillende plaatsen informatie opgeslagen en beheerd. Het betreft hier informatie in de meest brede zin van het woord. Deze informatie loopt uiteen van klantgegevens, vastgelegde storingsgegevens, gegevens inzake het uitvoeren van versiebeheer, prijslijsten enz. Binnen Technico Automatisering leeft het gevoel dat het centraal beheren van zoveel mogelijk informatie grote voordelen heeft met het oog op het delen van kennis en efficiency.

4. Doelstelling van de opdracht

Het doel van deze opdracht is een web-based informatiesysteem te ontwikkelen dat het voor klanten en medewerkers mogelijk maakt incidenten te melden. Het informatiesysteem moet er vervolgens voor zorgen dat deze meldingen bij de juiste medewerkers terecht komen. Een medewerker die een melding in behandeling neemt moet in het informatiesysteem kunnen aangeven wat de status van deze melding is en welke stappen hij heeft ondernomen om de melding op te lossen. Het informatiesysteem moet het ook mogelijk maken verschillende soorten gegevens te beheren. Op lange termijn dient alle informatie die momenteel nog verspreidt ligt binnen de organisatie in dit systeem te worden geïntegreerd. Medewerkers van de organisatie moeten via het internet toegang krijgen tot dit informatiesysteem zodat deze op elke locatie met een pc en internetverbinding voor hun beschikbaar is.

5. Uitgangssituatie

Meldingen van klanten verlopen momenteel telefonisch of via e-mail. Tevens worden vrijwel alle gegevens die verspreidt liggen binnen de organisatie, individueel beheerd in exelsheets. Er is een eerste aanzet gemaakt tot een centraal informatiesysteem in de vorm van een opdracht-beschrijving. In deze opdrachtbeschrijving staan alle informatiesoorten die in het uiteindelijke systeem moeten worden opgenomen.

6. Benodigde software

Voor het vervullen van deze afstudeeropdracht zal er gebruik worden gemaakt van verschillende applicaties. Deze zijn:

- PHP
- MySQL
- PHP MyAdmin
- PHP server
- Lokale webserver
- Dreamweaver
- FTP client

7. Benodigde hardware

Voor het vervullen van deze afstudeeropdracht zal er gebruik worden gemaakt van verschillende soorten hardware. Deze zijn:

- Complete werkplek met internetverbinding
- Webserver

8. Aanwezige ideeën

Er is een idee om meldingen van klanten via een 'SMS server' bij de juiste medewerker terecht te laten komen. Mocht de tijd dit toelaten, dan zal dit onderzocht worden.

9. Uit te voeren activiteiten

- Opstellen plan van aanpak
- Definitiestudie
 - Specificeren plan van aanpak
 - Uitvoeren veranderingsanalyse
 - Opstellen ontwikkelscenario
 - Bepalen systeemeisen
 - Opstellen systeemconcept
 - Opstellen technische structuur
 - Opstellen pilotplan
 - Verificatie en validatie
- Pilotontwikkeling
 - Specificeren plan van aanpak
 - Opzetten testomgeving
 - Bepalen globaal functionele structuur
 - Bepalen globaal technische structuur
 - Bepalen globaal organisatorische inrichting
 - Opstellen pilotontwikkelplan
 - Ontwerpen bouweenheden
 - Bouwen bouweenheden
 - Opstellen handmatige procedures
 - Beoordelen en testen pilot
 - Verificatie en validatie

De invoering zal alleen worden uitgevoerd wanneer hier aan het einde van het project tijd voor over is. Vanwege dit feit is de invoering niet opgenomen in de globale planning en in de op te leveren producten. Mocht de tijd het daadwerkelijk toelaten de fase invoering uit te voeren, dan zal er tegen die tijd een prioritering en planning worden gemaakt van de verschillende onderdelen van de invoering. De fase invoering bestaat uit de volgende onderdelen:

- Invoering
 - Specificeren plan van aanpak
 - Opstellen handleiding
 - Opstellen invoerprocedure
 - Invoering systeem
 - Systeemacceptatie
 - Overdracht

10. Te hanteren methodieken

- IAD (Iterative Application Development), een ontwikkelstrategie waarin het informatiesysteem tot stand komt via een aantal snel uitgevoerde iteraties van de complete ontwikkelcyclus.
- ITIL (Information Technology Infrastructure Library), een verzameling 'Best Practices' die in het werkveld van de IT-dienstverlening is aangetroffen voor de omschrijving van een serie fundamentele processen in de IT-beheer.
- Er zal een veranderingsanalyse worden uitgevoerd tijdens het voortraject.

11. Te gebruiken technieken

- UML (Unified Modeling Language), een modelleringstaal die bestaat uit een aantal modelleertechnieken als use cases, activity diagrammen, sequence diagrammen en klassendiagrammen.
- Interviewtechnieken

12. Planning

De afstudeerperiode beslaat 18 weken. 3 weken moeten worden gebruikt voor het opstellen van het eindverslag. Dit komt er op neer dat er in totaal 15 weken zijn voor het ontwikkelen van de verschillende systeemonderdelen. In de planning zijn de verschillende onderdelen van IAD, die van toepassing zijn op deze opdracht opgenomen.

Activiteit	Dagen
Opstellen plan van aanpak	5
Uitvoeren veranderingsanalyse	5
Definitiestudie	18
Specificeren plan van aanpak	1
Opstellen ontwikkelscenario	2
Bepalen systeemeisen	3
Opstellen systeemconcept	5
Opstellen technische structuur	2
Opstellen pilotplan	4
Verificatie en validatie	1
Pilontonwikkeling	42
Specificeren plan van aanpak	2
Opzetten testomgeving	2
Bepalen globaal functionele structuur	18
Bepalen globaal organisatorische inrichting	1
Opstellen pilotontwikkelplan	3
Bouwen bouweenheden	15

Verificatie en validatie	1
Testen	5

13. Te vermelden nadrukken

Technico automatisering heeft te kennen gegeven dat ze het erg belangrijk vindt dat de afstudeerder zelf voorstellen doet ten aanzien van het te ontwikkelen informatiesysteem, om zo tot een wederzijdse overstemming te komen betreft de functionaliteiten die het moet gaan bieden.

14. Resultaten voor de opdrachtgever

- Plan van aanpak
- Advies procesverbetering
- Definitiestudierapport
- Pilotontwikkeldrapporten
- Informatiesysteem*:
 - Database
 - Meldingsysteem
 - Gegevensbeheer

* Het op te leveren informatiesysteem zal volledig worden getest. De invoering van het systeem vindt echter alleen plaats wanneer hier aan het einde van het project tijd voor over is (zie uit te voeren activiteiten).

Interne bijlage 2: Het plan van aanpak

1. Afbakening opdracht

Gedurende dit project zal er een informatiesysteem worden ontwikkeld dat een koppeling realiseert tussen Technico Automatisering en hun klanten. Klanten kunnen via dit web-based informatiesysteem een melding (storing, klacht of vraag) maken welke vervolgens bij Technico Automatisering terechtkomt. Een medewerker van Technico Automatisering kan in het informatiesysteem aangeven welke stappen hij heeft ondernomen om de melding op te lossen. Bij het oplossen van de melding moet de medewerker over verschillende geïntegreerde informatiebronnen beschikken. Tevens moet deze gegevens met het informatiesysteem onderhouden kunnen worden en moeten er met systeem stamgegevens en gebruikersrechten beheerd kunnen worden.

2. Randvoorwaarden

- De bedrijfsmentor dient met enige regelmaat beschikbaar te zijn voor begeleiding en het voorzien van feedback.
- Er dient voor dit project een complete pc configuratie ter beschikking te staan (zie benodigde mensen/middelen).
- Er dient voor dit project een complete testomgeving (zie benodigde mensen/middelen) ter beschikking te staan.

3. Risicofactoren

- PHP is een taal waar afstudeerder al enige ervaring mee heeft. Het risico bestaat dat deze kennis echter onvoldoende is voor het ontwikkelen van het beoogde informatiesysteem. Aan de technische kennis van PHP zal gedurende dit project extra aandacht worden besteed.
- Een altijd aanwezig gevaar is het verlies van essentiële documentatie of programmatuur door systeemfalen. Dit risico zal tot een minimum beperkt worden door goede back-up procedures te volgen.
- Een deadline niet halen kan ongewenste gevolgen hebben voor het verloop van het project. De vertraging zou er toe kunnen leiden dat bepaalde delen van het project geschrapt moeten worden. Dit risico zal tot een minimum beperkt worden door een realistische planning op te stellen en deze zo strak mogelijk na te volgen.
- Scopecreep is een risico dat optreedt als het doel van het project uit het oog raakt verloren. Hoewel er tijdens dit project wel sprake is van een aantal incidentiele nevenactiviteiten, zal de focus gericht blijven op het oorspronkelijk geplande doel.
- Beperkte betrokkenheid van de bedrijfsmentor kan ertoe leiden dat de uiteindelijke functionaliteiten van het systeem niet aansluiten op de oorspronkelijke verwachtingen van het bedrijf. Door de noodzaak van zinvolle feedback van de bedrijfsmentor te benadrukken zal dit risico zoveel mogelijk vermeden worden.

4. Projectorganisatie

Gedurende dit project is er geen sprake van een projectorganisatie aangezien het gehele project door één persoon zal worden uitgevoerd. Deze persoon is verantwoordelijk voor alle aspecten van het project.

5. Wijze van rapporteren

Rapportage zal plaatsvinden in de vorm van dagelijkse contactmomenten met mijn bedrijfsmentor. Voorafgaand aan het contactmoment zal er mondeling een afspraak worden gemaakt wat betreft de tijd en aard van het contactmoment.

6. Benodigde mensen/middelen

Mensen:

- Bedrijfsmentor 1: Henk Houwaard
- Toekomstige systeemgebruikers

Middelen:

- Complete PC configuratie met internetverbinding (MS Windows XP, MS office)
- Ontwikkelomgevingen (PHP, MySQL, HTML)
- Ontwikkelstrategie (IAD)
- Literatuur (IAD, UML, PHP, MySQL, HTML, ITIL)
- Apache webserver

7. Globale planning

Activiteit	Dagen
Opstellen plan van aanpak	5
Uitvoeren veranderingsanalyse	5
Definitiestudie	18
Specificeren plan van aanpak	1
Opstellen ontwikkelscenario	2
Bepalen systeemeisen	3
Opstellen systeemconcept	5
Opstellen technische structuur	2
Opstellen pilotplan	4
Verificatie en validatie	1
Pilontonwikkeling	42
Specificeren plan van aanpak	2
Opzetten testomgeving	2
Bepalen globaal functionele structuur	18
Bepalen globaal organisatorische inrichting	1
Opstellen pilotontwikkelplan	3
Bouwen bouweenheden	15
Verificatie en validatie	1
Testen	5

8. Detailplanning

Eerste iteratie

- Definitiestudie (8 dagen)
 - Plan van aanpak
 - Ontwikkelscenario
 - Opzetten webserver
 - Prototype
 - Systeemeisen
 - Systeemconcept
 - Technische structuur
 - Organisatorische inrichting
 - Pilotplan
 - Verificatie en validatie
- Pilot 1: database (7 dagen)
 - Plan van aanpak

- Functionele structuur
 - Pilotontwikkeling
 - Verificatie en validatie
- Bouw database
- Pilot 2: Meldingsysteem (9 dagen)
 - Plan van aanpak
 - Functionele structuur
 - Organisatorische inrichting
 - Pilotontwikkelplan
 - Verificatie en validatie
- Bouw meldingsysteem

Tweede iteratie

- Veranderingsanalyse (5 dagen)
 - Verkenningfase
 - Ontwerpfase
 - Keuzefase
- Definitiestudie (7 dagen)
 - Plan van aanpak
 - Ontwikkelscenario
 - Systeemeisen
 - Systeemconcept
 - Technische structuur
 - Organisatorische inrichting
 - Pilotplan
 - Verificatie en validatie
- Pilot 1: database (2 dagen)
 - Plan van aanpak
 - Functionele structuur
 - Pilotontwikkeling
 - Verificatie en validatie
 - Bouw database
- Pilot 2: Meldingsysteem (10 dagen)
 - Plan van aanpak
 - Functionele structuur
 - Organisatorische inrichting
 - Pilotontwikkelplan
 - Verificatie en validatie
- Bouw meldingsysteem
- Pilot 3: Gegevensbeheer (9 dagen)
 - Plan van aanpak
 - Functionele structuur
 - Organisatorische inrichting
 - Pilotontwikkelplan
 - Verificatie en validatie
- Bouw gegevensbeheer

Derde iteratie

- Definitiestudie (2 dagen)
 - Plan van aanpak
 - Ontwikkelscenario
 - Systeemeisen
 - Systeemconcept
 - Technische structuur
 - Organisatorische inrichting
 - Pilotplan
 - Verificatie en validatie
- Pilot 1: database (1 dag)
 - Plan van aanpak
 - Functionele structuur
 - Pilotontwikkeling
 - Verificatie en validatie

- Bouw database
- Pilot 2: Meldingsysteem (4 dagen)
 - Plan van aanpak
 - Functionele structuur
 - Organisatorische inrichting
 - Pilotontwikkelplan
 - Verificatie en validatie
- Bouw meldingsysteem
- Pilot 3: Gegevensbeheer (6 dagen)
 - Plan van aanpak
 - Functionele structuur
 - Organisatorische inrichting
 - Pilotontwikkelplan
 - Verificatie en validatie
- Bouw gegevensbeheer

Invoering

- Testen (5 dagen)

9. Beschrijving mijlpaalproducten

- Opdrachtschrijving + plan van aanpak
- Advies procesverbetering
- Individueel voortgangsverslag
- Eindverslag
- Definitiestudierapport
- Pilotontwikkelrapporten
- Invoeringsrapport
- Informatiesysteem
 - Database
 - Meldingsysteem
 - Gegevensbeheer

10. Kosten/batenanalyse

De kosten van dit project zijn tweeledig. Enerzijds is het de stagevergoeding die wordt uitgekeerd in het kader van het project. Anderzijds is het de tijd die de bedrijfsmentor in het project steekt.

De baten van dit project drukken zich uit in het informatiesysteem inclusief documentatie dat aan het einde van dit project zal worden opgeleverd.

Interne bijlage 3: Interview incident management

Datum: 07/07/04

Interviewer: Yoran Gruppe

Geïnterviewde: Henk Houwaard

Betreft: incident management bij Technico Automatisering

1. Hoe lang duurt het oplossen van een telefonisch oplosbaar incident gemiddeld?

- Het oplossen van een incident duurt gemiddeld tien minuten. Dit is een zeer globale schatting omdat een telefonisch incident soms binnen enkele minuten is opgelost als het bijvoorbeeld gaat om een algemene vraag. Het kan ook voorkomen dat de gebruiker een verkeerde handeling met het systeem heeft uitgevoerd, in dit soort gevallen kan het soms wel een uur duren eer zo een melding is opgelost.

2. Hoeveel incidenten worden er gemiddeld dagelijks, wekelijks of maandelijks gemeld?

- Er worden dagelijks zo tussen de vijf a acht incidenten gemeld. Ook dit is een geschat gemiddelde.

3. Wat is doorgaans de moeilijkheidsgraad van een call?

- De meeste meldingen die binnenkomen betreffen een lage moeilijkheidsgraad, of te wel zijn eerstelijns.

4. Worden er prioriteiten aan calls toegekend? Zo ja, op basis waarvan?

- Formeel worden er geen prioriteiten toegekend aan incidentmeldingen. Informeel gebeurt dit wel. Er wordt vaak ter plekken bepaald welke meldingen kunnen wachten en welke niet.

5. Worden incidenten geregistreerd?

- Incidenten worden per onderwerp geregistreerd. Dit houdt in dat wanneer de specialist regelmatig een melding ontvangt over een fout in de software, dan wordt dit opgenomen in de 'buglist'.

6. Er is globaal gezien een verdeling te maken in soorten calls?

- Ja, even kort uit mijn hoofd zijn dat: gebruikersvragen, gebruikersfouten en storingen. Storingen zijn onderverdeeld in eigen software, externe software en apparatuur.

7. Als een incident niet direct via de telefoon kan worden opgelost, wat gebeurt er dan precies?

- In dit geval wordt er met de klant een termijn afgesproken waarbinnen hij wordt teruggebeld (bijvoorbeeld binnen 10 minuten). Vervolgens wordt het probleem opgelost en wordt de klant teruggebeld met de oplossing of gaat de specialist naar de klant toe om het probleem op te lossen. Als het probleem niet door de specialist kan worden opgelost dan wordt het probleem doorgegeven aan de leverancier en probeert de specialist een 'work around' te bedenken voor de klant. De specialist is verantwoordelijk voor het op de hoogte houden van de klant over de vorderingen van de leverancier.

8. Wat verdient doorgaans de voorkeur? Een incident zo snel mogelijk of zo goed mogelijk op te lossen?

- Uiteraard verdient een incident zo goed mogelijk op lossen de voorkeur. Vaak komt het voor dat de klant de beslissing neemt. Als de klant bereidt is te wachten, kan er gezocht worden naar de best mogelijke oplossing. Als hij niet bereidt is te wachten dan zal er in overleg met de klant voor een noodoplossing gekozen moeten worden.

9. Wie vormen binnen Technico Automatisering de eerste, tweede en derde lijn?

- De algemeen medewerkers, welke tevens de specialisten zijn vormen zowel de eerste als de tweede lijn. De derde lijn wordt gevormd door de verschillende leveranciers van de geleverde software.

10. Zijn er tussen Technico Automatisering en klanten SLA's gemaakt?

- Ja.

11. Verleend Technico Automatisering alleen maar diensten die in de SLA zijn opgenomen?

- Nee, het komt regelmatig voor dat Technico Automatisering diensten levert die buiten de SLA vallen.

Interne bijlage 4: PHP code 'oplossing wijzigen'

// Ophalen van oplossingen voor in de listbox

```
$sql8 = "SELECT Oplossingsnummer, Naam FROM oplossing ORDER BY Oplossingsnummer";
$result8 = mysql_query($sql8);
```

// Dit moet er worden uitgevoerd wanneer er op de knop 'wijzigen' word gedrukt

```
if($action == "toevoegen") {
```

// Hier wordt gecontroleerd of de waarde in \$oplossing een naam bevat. Als dit het geval is wordt het bijbehorende oplossingsnummer uit de tabel oplossing opgehaald en in \$oplossing gezet

```
$Oplossingsel = mysql_query("SELECT Naam, Oplossingsnummer FROM oplossing WHERE
Oplossingsnummer = '$tabel[Oplossingsnummer]'");
$Oplossingselect = mysql_fetch_array($Oplossingsel);
```

```
if ($oplossing == $Oplossingselect[Naam]) {
```

```
$oplossing = "$Oplossingselect[Oplossingsnummer]";
```

```
}
```

```
$sql = "UPDATE `melding` SET `Meldingsnummer` = \"\$meldingsnummer`Oplossingsnummer` =
\"$oplossing `WHERE `Meldingsnummer` = \"\$tabel[Meldingsnummer]\"";
```

```
}
```

// Listbox voor oplossing met 'preselect functie'

```
$Oplossingsel = mysql_query("SELECT Naam, Oplossingsnummer FROM oplossing WHERE
Oplossingsnummer= '$tabel[Oplossingsnummer]'");
$Oplossingselect = mysql_fetch_array($Oplossingsel);
$Oplossingselected = $Oplossingselect[Naam];
$Oplossingselected2 = $Oplossingselect[Oplossingsnummer];
```

```
echo " <td bgcolor=#0000CC width=\"200\"> <font color=\"#FFFFFF\" > Oplossing
</font></td>";
```

```
echo " <td bgcolor=#0000CC width=\"200\"><select name=\"oplossing\"style=width:180>";
while($oplossing = mysql_fetch_array($result8))
```

```
{
    print("<OPTION
VALUE=\"\$oplossing[0]\">$oplossing[1]\n");
}
```

```
?><option SELECTED> <?php echo $Oplossingselected ?></option> <?
echo " </select>";
```