



**algemene
woningbouw
vereniging**

Afstudeerverslag

**Incident Management, Problem Management en Knowledge Base binnen de
Algemene Woningbouw Vereniging**

**Dhr. D.G. Evenwel (20004838)
Amsterdam, oktober 2004**

**Vormgeving en Ontwerp van Interactie
Sector Informatica, Haagse Hogeschool
Examinatoren: ir. K.B. de Jong & ir. A.D. van Duuren**



Colofon

Auteur

Evenwel, D. G. (studentnummer 20004838)

Titel

Incident Management, Problem Management en Knowledge Base binnen de AWV

Afstudeerperiode

mei t/m oktober 2004

Datum

Oktober 2004

Plaats

Amsterdam

Afstudeeromgeving

Algemene Woningbouw Vereniging te Amsterdam

Bedrijfsmentor

Dhr. G. Verstoep

Instituut

Vormgeving en Ontwerp van Interactie (VIA) - Sector Informatica

Haagse Hogeschool, 's-Gravenhage

Examinatoren

Ir. K. B. de Jong

Ir. A. D. van Duuren

Contactgegevens Haagse Hogeschool

Bezoekadres: Johanna Westerdijkplein 75
2521 EN Den Haag

Postadres: Postbus 19
2520 CB Den Haag

Telefoon: 070-4558400

Fax: 070-4558405

Internet: <http://www.hhs.nl>

Contactgegevens Algemene Woningbouw Vereniging

Bezoekadres: Poeldijkstraat 10
1059 VM Amsterdam
Postbus 9252
1006 AG Amsterdam

Telefoon: 020-5118000

Fax: 020-5118233

Internet: <http://www.awv.nl>



Referaat

Evenwel, D.G.

Incident Management, Problem Management en Knowledge Base binnen de AWV

Eindverslag van het afstuderen van Danny Evenwel, in het kader van het afstuderen voor de opleiding Vormgeving en Ontwerp van Interactie (VIA) aan de Haagse Hogeschool te 's-Gravenhage.

Dit verslag beschrijft chronologisch de afstudeerperiode van Danny Evenwel bij de Algemene Woningbouw Vereniging te Amsterdam in de periode van mei tot en met oktober 2004.

De afstudeeropdracht omvatte het ontwerpen, bouwen en invoeren van een op ITIL gebaseerd systeem voor incident- en probleemregistratie, compleet met een Knowledge Base.

Descriptor

- Algemene Woningbouw Vereniging (AWV)
- Helpdesk
- ITIL
- Microsoft .NET
- Visual Basic
- IAD
- UML
- GUIDE
- Thinking Aloud



Voorwoord

Dit verslag vormt een beschrijving van mijn afstudeerperiode, welke van mei tot en met oktober 2004 bij de Algemene Woningbouw Vereniging te Amsterdam op de afdeling ICT is doorlopen. Tijdens de afstudeerperiode is gewerkt aan het ontwerp en de bouw van een incidentregistratiesysteem dat, eenmaal ingevoerd, een deel van de lopende problemen binnen de organisatie weg moet nemen.

Dit verslag moet de lezer een indruk geven van de gebruikte methoden en technieken, de uitgevoerde activiteiten en de resultaten daarvan. Daarnaast geeft het een indruk van de omvang en diepgang van de afstudeeropdracht.

Graag wil ik van deze gelegenheid gebruik maken om een dankwoord te uiten naar iedereen die binnen de organisatie in grote of kleine mate betrokken is geweest bij dit project. Met name dhr. Overdeest voor het verlenen van de afstudeerplek. Binnen de afdeling ICT gaat mijn dank uit naar mijn bedrijfsmentor dhr. Verstoep, dhr. Repko en dhr. Tjoa. Ook ben ik dank verschuldigd aan de afdeling Facilitaire Dienst, met in het bijzonder dhr. van 't Klooster voor het geven van waardevolle adviezen.

Als laatste wil ik vanaf deze plek dhr. K. De Jong en mevr. A. van Duuren bedanken voor het bieden van begeleiding en ondersteuning vanuit de Haagse Hogeschool.

Ik wens u veel plezier toe bij het lezen van dit verslag.

Danny Evenwel
Amsterdam, oktober 2004

Inhoudsopgave

I Organisatie	3
1.1 De organisatie	3
1.1.1 Geschiedenis	3
1.1.2 Missie / doelstellingen	3
1.2 Ondernemingsplan	4
1.3 Organisatiestructuur	4
1.4 Plaats binnen de organisatie	5
2 Afstudeeropdracht	6
2.1 Context van de opdracht	6
2.2 Probleemstelling	6
2.3 Doelstelling	7
2.4 Uitgangssituatie	8
2.4.1 Aanwezige ideeën	8
2.4.2 Kennis en ervaring	8
2.4.3 Aanwezige middelen	8
2.5 Opdrachtformulering	9
2.6 Aanpak (methoden en technieken)	10
2.6.1 Ontwikkelmethode en iteratiestrategie	10
2.6.2 Technieken	12
2.7 Uit te voeren werkzaamheden	12
2.8 Op te leveren producten	13
2.9 Planning	14
3 Fase Definitiestudie	15
3.1 Definieren gebruikersgroepen	15
3.2 Uitvoeren probleemanalyse	16
3.3 ITIL en helpdesks binnen de organisatie	17
3.3.1 Bedrijfscultuur en visie	17
3.3.2 Helpdesks binnen de organisatie	18
3.3.3 Invoering ITIL en helpdesksysteem	19
3.4 Definieren systeemeisen	22
3.4.1 Basissysteemeis A2: Wachtrij en notificatie	23
3.4.2 Basissysteemeis A6: Doorverwijzen incidenten	23
3.4.3 Basissysteemeis A9: Factuuroverzichten	24
3.4.4 Performance-eis G1: Verstoring primaire werkprocessen	24
3.4.5 Operationele eis H1: Gebruikersauthenticatie	25
3.5 Vervaardigen activiteitendiagrammen	25
3.6 Opstellen Use-cases	26
3.7 Vervaardigen toestandsdiagrammen	28
3.8 Bepalen objectklassen en database(model)	28
3.8.1 Samenstelling klassen	28
3.8.2 Opstellen relationeel databasemodel	31
3.8.3 Kiezen applicatie-instellingen	32
3.8.4 Besluitvorming Stored Procedures versus embedded T-SQL	32
3.8.5 Bepalen handmatig onderhoud	33
3.9 Vaststellen technische aspecten	33

3.10	Vaststellen organisatorische aspecten	33
3.11	Opstellen pilotplan	34
4	Pilotontwikkeling.....	37
4.1	Opstellen Application Style Guide	37
4.1.1	Bepalen Style Guide richtlijnen en invloeden	38
4.1.2	Bepalen iconen en grafische elementen	38
4.1.3	Vaststellen standaarden	39
4.2	Vervaardigen Prototypes gebruikersinterface en vaststellen functionele structuur	41
4.2.1	Pilot 1 - Incidentbeheer	42
4.2.2	Wachtrij en openstaande incidenten	43
4.2.3	Pilot 2 - Probleembeheer	45
4.2.4	Pilot 3 – Incidentinvoer- en status, kennissysteem en historie	45
4.2.5	Pilot 4 – Beheer, rapportage en voortgangscontrole	46
4.3	Vervaardigen navigatiediagrammen	48
4.4	Pilotontwikkelplan	48
4.5	Bouwen pilots	49
4.5.1	Importeren grafische elementen en iconen	50
4.5.2	Omzetten prototypes naar definitieve schermen	50
4.5.3	Aanmaken classes	50
4.5.4	Implementeren systeemlogica en werking	50
4.5.5	Vervaardigen Stored Procedures	51
4.5.6	Implementeren applicatie-instellingen	52
4.5.7	Implementeren periodieke statuscontrole	52
4.5.8	Implementeren rapportagemogelijkheden	52
4.6	Uitvoeren gebruikerstests	53
4.6.1	Kiezen testmethode	53
4.6.2	Opstellen testscenario's	54
4.6.3	Uitvoeren tests en vastleggen resultaten	55
5	Invoering en afronding.....	56
5.1	Invoering	56
5.2	Opstellen helpfiles	56
6	Evaluatie	58
6.1	Productevaluatie	58
6.1.1	Plan van Aanpak	58
6.1.2	Definitiestudie	58
6.1.3	Testplan en testresultaten	59
6.1.4	Pilotontwikkelplannen en ontwikkelde pilots	60
6.1.5	Ontwikkelde pilots	60
6.2	Procesevaluatie	61
6.2.1	Gekozen ontwikkelmethode en iteratiestrategie	61
6.2.2	Gekozen pilotstrategie en bepaling pilots	61
6.2.3	Gebruikte technieken	62
6.2.4	Planning en de fase Invoering	62
6.3	Doelrealisatie en gerealiseerde eisen	63
6.4	Opgeloste problemen	64
6.5	Invoering van het systeem en de toekomst	66
6.6	Conclusie	66
	Literatuurlijst	67
	Verklarende woordenlijst	68
	Bijlagen	70
	Bijlage A Opdrachtoomschrijving	71



Inleiding

De afstudeerperiode binnen de opleiding 'Vormgeving en Ontwerp van Interactie' (VIA) aan de Haagse Hogeschool vormt de afsluiting van de opleiding. Binnen de afstudeerperiode voert de student zelfstandig een opdracht uit, om te bewijzen dat de student het HBO niveau heeft bereikt en de opleiding met succes heeft doorlopen. De afstudeeropdracht is uitgevoerd binnen de Algemene Woningbouw Vereniging te Amsterdam.

Het doel van de opdracht omvatte het ontwerpen, ontwikkelen en implementeren van een compleet incident- en probleembehandelingspakket op basis van het ITIL raamwerk. Naast de afhandeling van incidenten en problemen vormen een Kennissysteem, management-rapportage en het opbouwen van een historie onderdeel van de opdracht.

Tijdens de afstudeerperiode is onder meer kennis en ervaring opgedaan met onder meer het onlangs door Microsoft gelanceerde .NET platform, de Visual Studio .NET ontwikkel-omgeving, Microsoft SQL Server 2000 in combinatie met T-SQL, Stored Procedures en de Active Directory Services. Daarnaast is een belangrijk deel van de opdracht organisatorisch van aard. Er is gekeken naar de mogelijkheid van het invoeren van delen van het ITIL raamwerk, en er is onderzoek gedaan naar de gevolgen van het invoeren van een ge-automatiseerd helpdeskpakket. Centraal hierin blijven de gevolgen in de werkwijzen voor alle betrokken gebruikersgroepen.

Binnen dit verslag zal de ontwikkeling van de producten in tijd als rode draad worden aangehouden. Hoewel een deel iteratief is ontwikkeld, kan op deze wijze een goede indruk worden gevormd van het gehele ontwikkeltraject, tot aan de daadwerkelijke invoering aan toe.

Dit verslag begint in hoofdstuk 2 met een overzicht van de organisatie waarin de afstudeeropdracht is uitgevoerd. Hoofdstuk 3 omvat een beschrijving van de afstudeeropdracht, met daarin onderwerpen als de doelstelling, probleemstelling en de gekozen aanpak.

De basis van dit verslag wordt gevormd door een bespreking van de drie hoofdfasen die gedurende het project zijn doorlopen: de Definitiestudie, Pilotontwikkeling en Invoering. Deze vormen respectievelijk hoofdstuk 4, 5 en 6. Centraal hierin staan de uitgevoerde werkzaamheden, belangrijke genomen beslissingen en interessante knelpunten. Belangrijk om in gedachten te houden is dat er in de voorgenoemde hoofdstukken geen oordeel zal worden gegeven over het doorlopen proces of de opgeleverde producten; dit zal worden behandeld in hoofdstuk 7.

Hoofdstuk 7 bevat een proces- en productevaluatie, waarin ingegaan wordt op het doorlopen proces en de opgeleverde producten. Er zal worden vastgesteld op welke punten er - achteraf gezien - ruimte voor verbetering is, of de gestelde doelstellingen zijn behaald en of de problemen uit de probleemstelling zijn opgelost of afgenomen. Verder zal aangegeven worden of aan de systeemeisen is voldaan.



Tot slot bevat dit verslag een literatuurlijst en een verklarende woordenlijst. Naast dit document bestaat het complete verslag uit een serie bijlagen. Achterin dit document is een verwijzing opgenomen naar deze documenten.

Verwijzingen naar literatuur of websites worden aangeduid met een nummer tussen rechte haken [...]. Deze referenties zijn terug te vinden in de bronvermelding, eveneens te vinden achterin dit verslag.

Organisatie

In dit hoofdstuk staat de organisatie, waarin de afstudeeropdracht is uitgevoerd, centraal. Er zal worden gestart met een overzicht van de organisatie, waarin wordt ingegaan op de geschiedenis en het ontstaan van de organisatie en de doelstellingen. Daarna is er aandacht voor het nieuwe ondernemingsplan, de organisatiestructuur en de plaats van de afstudeerder binnen de organisatie.

I.1 De organisatie

I.1.1 Geschiedenis

De Algemene Woningbouw Vereniging (AWV) is met bijna 18.000 woningen en bedrijfsruimten een middelgrote woningcorporatie in Amsterdam. De AWV, opgericht in 1910 vanuit de socialistische arbeidersbeweging, kan zich beroepen op een lange traditie op het gebied van wonen in Amsterdam. De AWV is een particuliere non-profit organisatie.

Vanaf haar oprichting stond “de arbeider” centraal. In het Beleidsplan van 1998 is te lezen dat de AWV zich met name richt op “mensen die door hun inkomen of door andere omstandigheden moeilijkheden ondervinden bij het vinden van voor hen passende huisvesting”. Direct werd de koppeling gelegd naar “de primaire doelgroep”, oftewel huishoudens met lage inkomens. Jarenlang is de aandacht van de AWV (en met haar de meeste corporaties) gefixeerd geweest op deze doelgroep. Naar buiten toe overheerste het beeld: sociaal-democratie, vakbond, lage inkomens, lage huren en sober en doelmatig.

I.1.2 Missie / doelstellingen

De primaire doelstelling van de AWV was oorspronkelijk "het streven naar goede huisvesting voor arbeiders". Door de maatschappelijke ontwikkelingen in de afgelopen jaren is deze veranderd in "uitsluitend werkzaam zijn in het belang van de volkshuisvesting". De belangrijkste taken en activiteiten die hieruit voortvloeien zijn het bouwen, renoveren, onderhouden en beheren van woningen in de sociale sector.

Voor de periode 2004 – 2008 heeft de AWV zich in het ondernemingsplan een tweeledige missie gesteld. De belangrijkste is 'het bieden van kwalitatief goede woondiensten aan al die groepen die daar onder de Amsterdamse marktomstandigheden niet zelfstandig in kunnen voorzien'. De AWV biedt een breed pakket voor een brede doelgroep, verspreid over Amsterdam. Daarnaast heeft de AWV ook een maatschappelijke verantwoordelijkheid, die verder gaat dan het verhuren van woningen. Ook op het gebied van veiligheid, leefbaarheid en bewonersparticipatie wil de AWV een bijdrage leveren.

De organisatie richt zich op een aantal doelgroepen:

- Mensen met lage(re) inkomens: de primaire doelgroep. De AWV is van mening dat er voor deze groep bijna nergens in Amsterdam een marktconforme en kostendekkende huur zal kunnen worden gevraagd. Daarom is deze doelgroep aangewezen op woningen waarvoor huursubsidie kan worden verkregen.
- Middeninkomens. Ook deze groep is niet in staat een marktconforme prijs te betalen
- Zorgbehoevende groepen. Bij deze groep gaat het om de combinatie van een geschikte woonvorm, een passend zorgaanbod en betaalbaarheid.
- Starters, nieuwkomers en passanten. Bij deze groep, waar het meestal gaat om tijdelijke huisvesting, is locatie en snelle beschikbaarheid de primaire insteek.
- Overige klanten.

De AWV wil in de komende vijf jaar gericht inspelen op de vraag van de verschillende doelgroepen. Dit betekent een algemene kwaliteitsverbetering van het bezit, meer duurdere huurwoningen, meer woningen die geschikt zijn voor ouderen en mensen met een handicap en meer woonegelegenheden die snel beschikbaar zijn voor nieuwkomers in de stad.

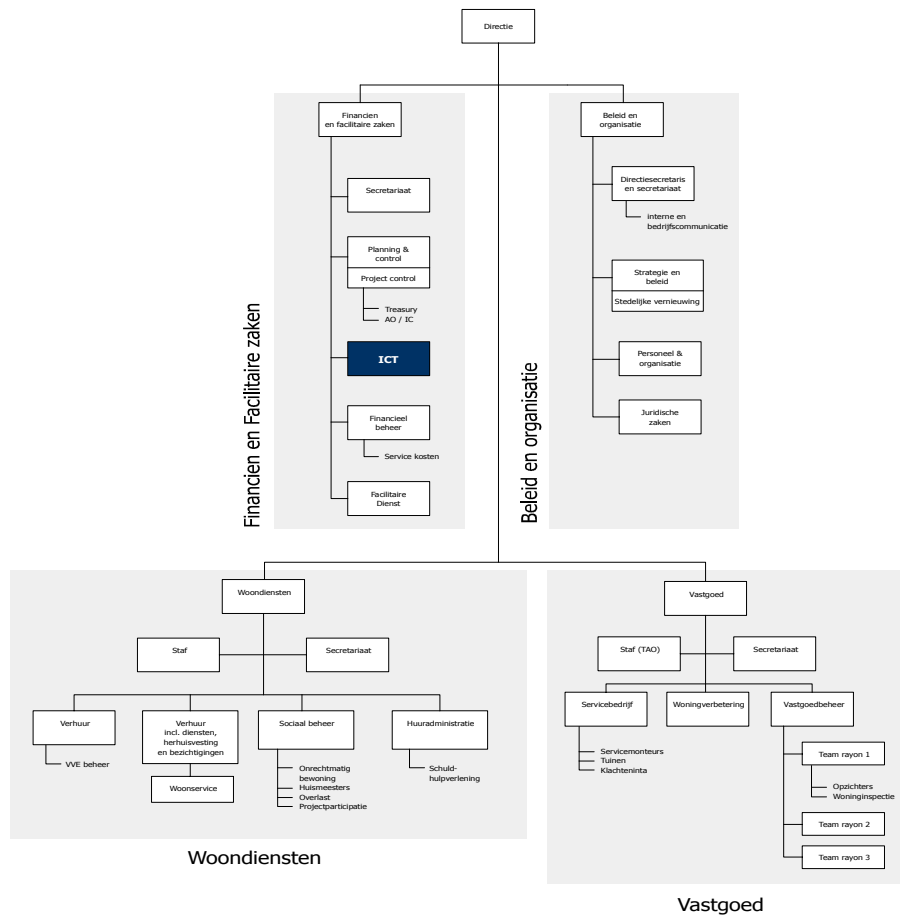
1.2 Ondernemingsplan

Het nieuwe ondernemingsplan heeft ook gevolgen voor de interne organisatie. Zo zal er ondermeer gekeken gaan worden waar activiteiten van verschillende afdelingen elkaar overlappen. Vervolgens worden er - per afdeling - nieuwe werkplannen ontwikkeld, waarin de raakvlakken met andere afdelingen worden beschreven en er duidelijke afspraken worden gemaakt om zo efficiënt mogelijk met de beschikbare resources om te gaan.

Een voorbeeld van overlappende werkzaamheden, waarvoor betere afspraken gemaakt kunnen worden, betreft de afdelingen ICT en Facilitaire Dienst. Beide afdelingen beschikken over een eigen eerstelijns helpdesk. In het nieuwe ondernemingsplan zal worden vastgelegd dat deze afdelingen nauwer zullen gaan samenwerken om overlappings in de werkzaamheden zoveel mogelijk te zullen gaan voorkomen.

1.3 Organisatiestructuur

Het onderstaande organigram geeft een beeld van de interne organisatie van de AWV. De organisatie is onderverdeeld in vier hoofdgebieden (portefeuilles): Financiën & Facilitaire Zaken, Beleid & Organisatie, Woondiensten en Vastgoed. Elk van deze gebieden heeft een eigen directeur, die eindverantwoordelijk is voor de betrokken afdelingen.



Figuur 1: Organigram Algemene Woningbouw Vereniging

1.4 Plaats binnen de organisatie

De opdracht is uitgevoerd binnen de afdeling ICT binnen de portefeuille Financiën & Facilitaire Zaken. Dit is aangegeven in het organigram in de vorige paragraaf. De afdeling ICT heeft als kerndoelstelling (missie): 'het waarborgen van de continuïteit en integriteit van de informatievoorziening'. Concreet houdt dit in dat de medewerkers van de afdeling zijn belast met het verlenen van ondersteuning op het gebied van de aanwezige hardware en software. Hieronder vallen zaken als het beheer van het netwerk, telefonie, bedrijfskritische applicaties en de computerapparatuur.

De plaatsing van de afstudeerder op deze afdeling is een logische: de medewerkers van de afdeling ICT vormen de gebruikersgroep die, naar verwachting, het meeste gebruik zullen gaan maken van de resultaten van het project. De verwachting is dat de medewerkers hun wensen, eisen en verwachtingen met betrekking tot het project duidelijk kenbaar kunnen maken. Daarnaast is specifieke kennis aanwezig, zoals kennis van de aanwezige hard- en software, die nodig is om het project uit te kunnen voeren. Als laatste is er via deze weg gemakkelijk contact te leggen met de externe medewerkers van softwareontwikkelaar DataBalk, die ingeschakeld kunnen worden bij het oplossen van specifieke problemen.

2

Afstudeeropdracht

Op basis van de opdrachtschrijving (zie bijlage A), is bij de start van de afstudeerperiode een Plan van Aanpak opgesteld. Dit document diende als basis voor de uitvoering.

Dit hoofdstuk is een afgeleide van het Plan van Aanpak, dat terug te vinden is in bijlage B. Het bevat onder andere een gedetailleerde planning. Waar nodig zal worden aangegeven hoe de relevante gegevens zijn verkregen.

2.1 Context van de opdracht

Sinds enige tijd is de behoefte binnen de organisatie aanwezig om structuur aan te brengen in het proces van het melden en afhandelen van incidenten op ICT-gebied. Incidenten beslaan het gebied van simpele vragen of verzoeken tot serieuze en ingrijpende klachten of problemen.

Met de komst van de beschreven organisatorische veranderingen wordt dit gezien als een goede gelegenheid de betrokken processen in kaart te brengen, deze zonodig aan te passen of nieuwe processen te introduceren. Tevens wordt dit gezien als een gelegenheid om de Facilitaire Dienst (specifiek: de afdeling post/repro) hierin mee te nemen omdat beide afdelingen incidenten ontvangen en afhandelen. Hetzelfde geldt voor de aanwezige applicatiebeheerders die binnen de organisatie zijn aangesteld om problemen af te handelen en vragen te beantwoorden met betrekking tot de primaire bedrijfsapplicatie Wocas/X.

2.2 Probleemstelling

Binnen de AWW ontbreekt een gestructureerde wijze van afhandelen van incidenten en service requests voor de betrokken groepen. Hierdoor kan het voorkomen dat deze op een ad hoc wijze worden afgehandeld wat de efficiency en overzichtelijkheid van de werkzaamheden niet ten goede komt. Eindgebruikers nemen via de telefoon, e-mail of in eigen persoon contact op met één van de medewerkers waarna deze, vaak op eigen wijze, de zaak in behandeling neemt.

Door deze werkwijze wordt er niet altijd vastgelegd wat de oplossingen voor problemen zijn geweest. Nieuwe medewerkers moeten hierdoor langer ingewerkt worden. Ook komen bepaalde problemen vaak terug, zoals problemen met het in- en uitloggen en problemen met printers. Deze zijn in veel gevallen door de gebruikers zelf op te lossen; er is echter geen centrale plek waar gebruikers naar verwezen kunnen worden voor veel voorkomende problemen. Wel zijn er handleidingen beschikbaar op het intranet en worden de gebruikers per e-mail op de hoogte gesteld van oplossingen.

Tevens is het vaak niet duidelijk welke knel- of verbeterpunten er binnen de ICT infrastructuur bestaan. Door de wijze van het oplossen van klachten en problemen komen

achterliggende problemen soms niet naar voren en wordt hier niets mee gedaan. Er wordt binnen de afdeling ICT wel gestructureerd aan oplossingen gewerkt, echter de oplossingswijze is niet altijd geformaliseerd op één plaats vastgelegd.

Op basis van de bovenstaande paragrafen is in de fase Definitiestudie (hoofdstuk 4) een concrete lijst met probleempunten opgesteld. Deze bevatten aanvullingen op de initiële probleemstelling, die hoofdzakelijk zijn voortgekomen uit gesprekken met medewerkers van de afdeling ICT, het hoofd van de afdeling Facilitaire Dienst, en nieuw verkregen inzichten.

2.3 Doelstelling

De doelstelling van de opdracht is het ontwerpen, ontwikkelen en implementeren van een softwarepakket dat het onder andere mogelijk maakt het complete traject voor het afhandelen van incidenten te automatiseren en te stroomlijnen. Het doel is het aanbrengen van structuur in de betrokken processen en de informatievoorziening naar de gebruiker te vergroten. De voornaamste redenen voor het ontwikkelen en implementeren van een dergelijk softwarepakket zijn:

- De voortgang van openstaande incidenten kan beter bewaakt worden waardoor helpdeskmedewerkers meer gestructureerd kunnen werken aan het oplossen van incidenten.
- Eindgebruikers kunnen, al dan niet automatisch, op de hoogte worden gesteld van de status van hun lopende incidenten.
- Geregistreerde incidenten kunnen later opnieuw worden gebruikt om diagnoses te stellen en oorzaken te achterhalen. Ook zal het mondeling overdragen van kennis aan nieuwe medewerkers verminderd kunnen worden wanneer deze informatie bewaard blijft en opvraagbaar is.
- De helpdesks kunnen worden ontlast (vermindering van het aantal incidenten) doordat het antwoord op veelvoorkomende vragen door de gebruikers zelf kan worden opgezocht.
- Informatie uit het systeem kan worden gebruikt voor het opstellen van managementrapportages, die onder andere kunnen bijdragen aan het nemen van beslissingen. Denk hierbij bijvoorbeeld aan het aantal incidenten per categorie of het aantal opgeloste problemen.

De bovenstaande punten worden in hoofdstuk 7 gebruikt, waarin het evalueren van de doelstellingen een belangrijke plaats inneemt. Wanneer de doelstellingen zijn behaald, de problemen zijn opgelost en aan de eisen is voldaan kan een oordeel worden gevormd over het succes van het project.

2.4 Uitgangssituatie

2.4.1 Aanwezige ideeën

Bij aanvang van de projectperiode waren de volgende ideeën aanwezig:

- Het bestaande Windows 2000 domein kan gebruikt worden als basis voor het instellen van gebruikersrechten.
- Ontwikkelde programmatuur kan op de centrale applicatieserver van de organisatie worden geplaatst, waarna gebruikers deze door middel van een snelkoppeling in het gebruikersprofiel kunnen opstarten.

2.4.2 Kennis en ervaring

Omdat de opdracht binnen de afdeling ICT is uitgevoerd was er - via de medewerkers - toegang tot kennis over het interne netwerk, de bedrijfservers, databases en organisatorische inrichting.

De afstudeerder had, naast de kennis uit afgesloten modules uit de opleiding, kennis van 4^e generatie programmeertalen, web-scriptingtalen, ervaring met databasebeheer- en databasereval en ervaring en affiniteit met grafisch ontwerp.

2.4.3 Aanwezige middelen

Een deel van de te gebruiken software is voor aanvang van de projectperiode opgenomen in de opdrachtschrijving. Aanvullingen hierop zijn - samen met de originele punten - terug te vinden in de onderstaande lijst:

- Microsoft Windows besturingssysteem
- Microsoft Office (documentatie en rapportage)
- Adobe Photoshop en Illustrator (grafische elementen)
- Microangelo 5.5 (maken en verkennen van icoon libraries)
- Microsoft Visual Basic .NET 2002, i.c.m. .NET framework 1.0 (programmatuurontwikkeling)
- SQL Server 2000 tools: Enterprise Manager, Query Analyzer en Performance Monitor (database toegang en beheer)
- Softerra LDAP Administrator (Active Directory Services beheer en browser)
- CVS software (versiebeheer)
- Microsoft HTML Help Workshop (vervaardigen helpfiles)
- Adobe Type Manager en Typograf (verkennen en invoegen lettertypen)
- Externe verbindingsoftware voor RDP en Citrix (serverbeheer op afstand)
- Evaluatieversies van o.a. TopDesk en Track-it

Middelen in de vorm van boeken en andersoortige documentatie, los van de voorgeschreven literatuur vanuit de opleiding:

- Een aantal boeken en websites, specifiek gericht op het .NET platform en Visual Studio .NET
- Documentatie m.b.t. Microsoft SQL Server 2000 (Online Books)
- Informatie over ITIL processen
- Microsoft richtlijnen m.b.t. interactieontwerp
- Verschillende websites en een boek over specifieke vormen van gebruikerstests

Naast de verzameling software en geschreven materiaal valt onder de aanwezige middelen ook de computerhardware als een PC met internetaansluiting, monitor, printers en andere fysieke zaken als een werkplek.

2.5 Opdrachtformulering

De opdrachtformulering is afgeleid uit de probleemstelling, initiële wensen en eisen vanuit de organisatie en een idee over de afbakening van de opdracht en vormt een concreet beeld – in globale vorm - van de gehele omvang van de opdracht.

De opdracht omvat het ontwerpen, ontwikkelen en implementeren van een systeem, dat de voorgenoemde doelstellingen ondersteunt. Het systeem bevat de volgende mogelijkheden en kenmerken:

- Incidentinvoer en Incidentstatus. Een eindgebruiker moet hiermee een incident kunnen melden vanaf iedere werkplek. Ook moet de eindgebruiker de status van zijn of haar lopende incidenten kunnen opvragen.
- Knowledge Base (Kennissysteem). De Knowledge Base moet hoofdzakelijk gaan dienen als kennissysteem waarnaar eindgebruikers verwezen kunnen worden voor oplossingen van veelvoorkomende problemen. De Knowledge Base moet aangevuld en aangepast kunnen worden door de (helpdesk)medewerkers, welke aan hoofd- en subcategorieën onderwerpen en items kunnen toevoegen.
- Incident Management (Incidentbeheer). Na binnenkomst van een incident wordt deze in behandeling genomen door één van de helpdeskmedewerkers, afhankelijk van het type melding. Het gaat hier dus om het complete afhandelingstraject van een incident. Het moet mogelijk worden incidenten permanent op te slaan zodat Problem Management op basis hiervan problemen kan definiëren.
- Problem Management (Probleembeheer). Al dan niet op basis van incidenten moeten problemen gedefinieerd kunnen worden. Hierna moet het mogelijk zijn een probleembeheerder toe te kennen en oplossing aan het probleem te koppelen zodat een 'Known Error' (een bekend probleem) ontstaat. Ook hier gaat het om het complete traject van de afhandeling van problemen.

Het systeem zal zoveel mogelijk ontworpen worden volgens incidentafhandeling op basis van de 'IT Infrastructure Library' (ITIL) [5]. Wel zal het systeem waar nodig worden aangepast aan de wensen en eisen van de organisatie. Andere punten van belang zijn:

- De drempel voor het gebruik van de invoer van een incident moet laag zijn, om te voorkomen dat medewerkers alsnog de telefoon pakken om een melding te doen.
- Vanaf iedere werkplek moet het systeem toegankelijk zijn voor iedere medewerker van de organisatie. De meldingen moeten vervolgens bij de juiste helpdesk binnen komen.
- Door gebruik te maken van gebruikersrechten kan de toegang tot delen van het systeem worden bepaald.

2.6 Aanpak (methoden en technieken)

Deze paragraaf gaat in op de gekozen aanpak. Centraal staan de gekozen ontwikkelmethode, iteratiestrategie en te gebruiken technieken.

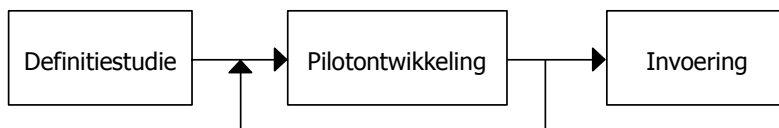
2.6.1 Ontwikkelmethode en iteratiestrategie

Als ontwikkelmethode is voor 'Iterative Application Development' (IAD) [1] gekozen. Deze ontwikkelmethode kenmerkt zich in het feit dat één of meerdere fasen van het ontwikkelproces iteratief (meerdere malen) wordt doorlopen. Dit in tegenstelling tot een meer klassieke 'waterval' methode, waar de fasen elkaar in lineaire vorm opvolgen.

Als iteratiestrategie is gekozen voor het incrementeel ontwikkelen. Dit houdt in dat in een Definitiestudie het systeem volledig wordt beschreven en alle eisen worden vastgelegd. Vervolgens wordt in onderdelen (pilots) het systeem ontwikkeld.

Na ontwikkeling van de pilots wordt het systeem in zijn geheel ingevoerd. Deze strategie wordt ook wel vergeleken met de ontwikkelmethode Rapid Application Development (RAD), waarin het vaststellen van eisen samen met de eindgebruikers, het gebruik van prototyping en het hergebruik van softwarecomponenten centraal staat.

Schematisch kan deze iteratiestrategie als volgt worden weergegeven:



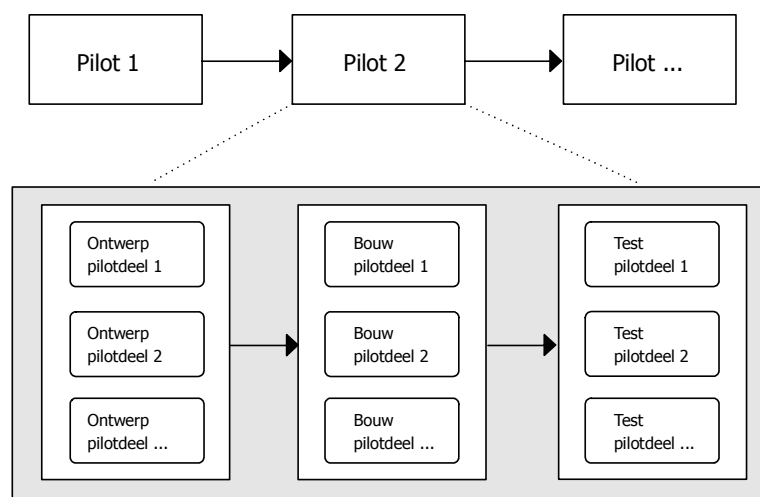
Figuur 2: Hoofd fasen in de IAD ontwikkelmethode

De redenen voor het kiezen van deze strategie zijn:

- Het is mogelijk van tevoren het systeem en de systeemeisen te specificeren. Er kan vanuit worden gegaan dat de betrokken afdelingen binnen het ontwikkeltraject geen ingrijpende veranderingen zullen ondergaan, afgezien van de al geplande organisatorische veranderingen. Ook kan worden verwacht dat de wens voor een dergelijk systeem zal blijven bestaan.
- Wanneer de fase Definitiestudie in iteraties zou plaatsvinden, zou 'scope creep' op kunnen treden. Dit houdt in dat het systeem nooit 'af' is en de systeemeisen steeds weer worden bijgesteld. Dit is natuurlijk niet wenselijk in een project waar een vast tijds kader voor staat. Tevens dient het vaststellen van systeemeisen en systeemfunctionaliteit als een houvast voor de ontwikkelaar: er kan – aan het einde van de fase Definitiestudie – een concreet beeld worden gevormd van de inhoud van het project.
- De omvang van het systeem is te overzien. Het totale systeem kan goed worden opgedeeld in subsystemen, waarvan vooraf een duidelijke specificatie gemaakt kan worden.
- Een stapsgewijze invoering is niet gewenst. De voorkeur wordt gegeven aan het in één stap invoeren van het gehele systeem.
- Er is geen dringende behoefte om bepaalde subsystemen zo snel mogelijk in te voeren.

Door de gekozen iteratiestrategie in ogenschouw te nemen, kan gesteld worden dat de ontwikkelwijze een tussenvorm is tussen het puur evolutionair ontwikkelen van een systeem en de waterval methode. De functionele structuur en eisen worden vastgelegd (en zijn dus naderhand niet meer aan te passen), terwijl de daadwerkelijke invulling en ontwikkeling van het systeem in pilots iteratief tot stand kan komen.

Binnen de ontwikkeling van iedere pilot zal een scenario worden aangehouden waarin de globale specificatie in één keer voor de hele pilot wordt opgesteld, waarna de pilot in een aantal pilotdelen wordt ontwikkeld. Deze kan later worden opgedeeld in bouweenheden. Na de bouw kan eventueel getest worden. Dit is schematisch als volgt weer te geven:



Figuur 3: Activiteiten binnen de fase Pilotontwikkeling

2.6.2 Technieken

De volgende technieken zullen gebruikt worden gedurende het project:

- Diverse technieken uit GUIDE [2], zoals het opstellen van navigatieschema's, GUI prototypes en een Application Style Guide.
- UML [3] modelleringstechnieken, zoals Use-cases en activiteitendiagrammen voor het in kaart brengen van activiteiten in de nieuwe situatie.
- Testtechnieken [4], voor het evalueren van opgeleverde producten.

Verder zal ITIL voornamelijk dienen als basis voor het ontwerpen van de processen voor incidentafhandeling (Incident Management) en problemen (Problem Management). Waar nodig zal hiervan worden afgeweken om op de wensen en eisen van de organisatie in te spelen.

2.7 Uit te voeren werkzaamheden

In grote lijnen zal de ontwikkeling van het project in de onderstaande stappen uitgevoerd worden. Deze punten zijn grotendeels opgesteld aan de hand van de gekozen ontwikkelmethode.

1. Het opstellen van het Plan van Aanpak, waarin de opdracht en de doelstelling wordt geformuleerd.
2. Het opstellen van een definitiestudie, waarin de volgende onderwerpen worden beschreven:
 - De huidige situatie bij de AWV met betrekking tot de betrokken afdelingen en taken.
 - De eisen waaraan het nieuwe systeem moet voldoen (de systeemeisen).
 - Het systeemconcept, waarin op globaal niveau een oplossing voor de gestelde problemen wordt geformuleerd.
 - Een globale beschrijving van de technische structuur waarin het systeem zal worden geplaatst. Denk hierbij aan de ICT infrastructuur en aanwezige systeemconfiguraties.
 - Een beschrijving van de organisatorische inrichting over de betrokken afdelingen, welke veranderingen daar zullen optreden als gevolg van het invoeren van een nieuw systeem.
 - Een pilotplan, waarin uiteengezet wordt in welke pilots het systeem zal worden opgedeeld, wat deze inhouden en met welke prioriteit deze zullen worden ontwikkeld.

3. Het opstellen van pilotontwikkelpannen voor alle pilots afzonderlijk, waarin per pilot het onderstaande naar voren komt:
 - De functionele structuur, waarin onder andere de gebruikersinterface aan bod komt samen met een (conceptueel) gegevensmodel. Ook zal hier aandacht worden besteed aan de Style Guide die zal gelden voor het gehele systeem.
 - De technische structuur, waarin de technische kant van de pilot wordt behandeld.
 - De organisatorische inrichting, waarin zal worden ingegaan op de gebruikersrollen en handmatige procedures die gelden voor de pilot.
 - Een pilotontwikkelpan waarin de pilot wordt opgedeeld in pilotdelen en bouweenheden.
4. Het ontwikkelen van de pilots, welke samen het complete systeem vormen. In deze fase vindt de daadwerkelijke bouw van het systeem plaats.
5. De invoering van het systeem, waarbij het systeem beschikbaar zal worden gemaakt op de werkplekken van de eindgebruikers.
6. Het ontwikkelen van de benodigde helpfiles voor alle pilots.
7. Het voorbereiden en uitvoeren van een aantal gebruikerstests en de resultaten hiervan vastleggen.

Ter ondersteuning van het bovenstaande is het noodzakelijk dat er onderzoek gedaan wordt naar de bruikbaarheid van de ITIL standaarden en welke werkprocessen aangepast dienen te worden voor de nieuwe situatie.

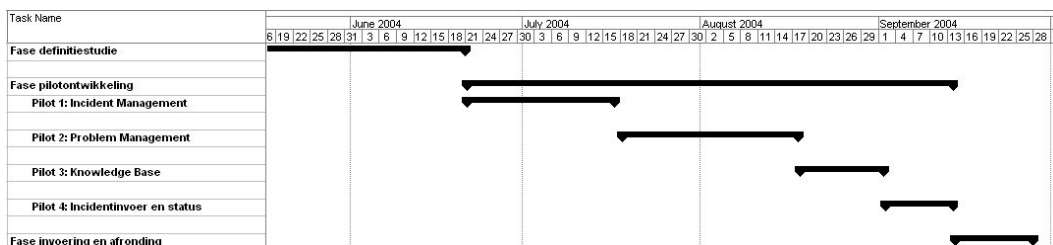
2.8 Op te leveren producten

De volgende producten en documentatie zullen worden opgeleverd:

1. Een Plan van Aanpak.
2. Een Definitiestudie.
3. Een aantal Pilotontwikkelpannen.
4. De pilots zelf, zoals beschreven in de voorgenoemde projectdoelstellingen, als losse onderdelen die uiteindelijk een compleet systeem vormen wanneer deze geïmplementeerd worden.
5. Helpfiles voor alle opgeleverde producten.
6. Een testplan en de uitwerking van de testresultaten, waarin duidelijk wordt hoe er getest zal worden en wat de resultaten van de tests zijn.

2.9 Planning

De onderstaande afbeelding laat in globale vorm de planning zien, die aan het begin van de projectperiode is opgesteld.



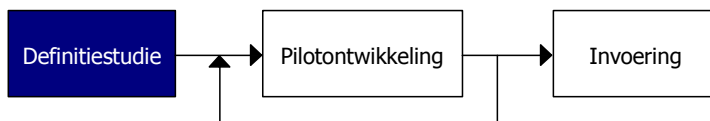
Figuur 4: Initiële planning in globale vorm

De afronding van de Definitiestudie, de afronding van individuele pilots en het afronden van de invoering worden gezien als mijlpalen.

3

Fase Definitiestudie

Het ontwikkeltraject van het project is van start gegaan met de eerste fase, de fase Definitiestudie. De onderstaande figuur geeft de plaats van deze fase in het totale ontwikkeltraject weer:



Figuur 5: De fase Definitiestudie

In dit hoofdstuk worden de belangrijkste activiteiten besproken die binnen deze fase zijn uitgevoerd. Dit zijn onder andere het bepalen van de inrichting van de helpdesks binnen de organisatie, het vaststellen van de opslagstructuur van het beoogde systeem en het bepalen van taken die uitgevoerd moeten gaan worden in de toekomstige situatie.

Het resultaat van deze fase is opgenomen als bijlage D.

3.1 Definiëren gebruikersgroepen

Als één van de eerste activiteiten binnen de fase Definitiestudie is het definiëren van de gebruikersgroepen uitgevoerd. Hierbij is vastgesteld welke verschillende groepen er bestaan binnen het project, en welke karakteristieken zij bezitten. Deze activiteit is te vergelijken met het definiëren van de aanwezige ‘User-classes’ binnen de GUIDE [2] methodiek.

Dit is gedaan op basis van de opdrachtomschrijving en een globaal idee van het nieuwe systeem. De volgende gebruikersgroepen zijn naar voren gekomen:

- Medewerkers van de afdeling ICT
- Medewerkers van de afdeling Facilitaire Dienst
- Applicatiebeheerders van het softwarepakket Wocas/X
- Alle ‘gewone’ medewerkers van de AWV

Een beschrijving van de bovengenoemde groepen is opgenomen in paragraaf 1.1 van de Definitiestudie. Belangrijk om op te merken is dat alleen de groepen die direct interactie met het nieuwe systeem zullen hebben zijn aangemerkt als gebruikersgroep. Managers die rapportages ontvangen vallen hier dus buiten. Ook is er een verschil in benodigde kennis en ervaring te onderkennen. Van de medewerkers die veel gebruik zullen gaan maken van het systeem (de eerste drie genoemde groepen) mag verwacht worden dat zij tijd en moeite steken in een systeem waar zij regelmatig gebruik van zullen gaan maken. Daarentegen is het niet redelijk van de gewone medewerkers te verwachten dat zij veel tijd spenderen aan het onder de knie krijgen van een systeem dat zij niet regelmatig zullen gebruiken.

Verder is vastgesteld dat het binnenkomen van incidenten bij de applicatiebeheerders als een golfbeweging gezien kan worden. De ervaring is dat bij een nieuwe release van het pakket meldingen binnenkomen met betrekking tot problemen en vragen. Na verloop van tijd zijn

deze opgelost en komen er slechts sporadisch meldingen binnen, totdat er weer een nieuwe release wordt ingevoerd.

De bovengenoemde groepen zijn het uitgangspunt geweest bij de het ontwerpen van het systeem. Deze kunnen desgewenst worden uitgebreid.

3.2 Uitvoeren probleemanalyse

In het Plan van Aanpak, dat voor een deel is gebaseerd op de opdrachtomschrijving, is de probleemstelling van het project te vinden. Aangezien er bij aanvang van de Definitiestudie meerdere gebruikersgroepen zijn gedefinieerd, leek het nuttig na te gaan wat de visie van andere betrokkenen op de probleemstelling is. Door middel van gesprekken met een aantal medewerkers en afdelingshoofden is deze informatie verzameld.

Vooralsnog bleken de inzichten van het hoofd van de afdeling Facilitaire Dienst. Zo werd bijvoorbeeld het probleem van verkeerd binnenkomende telefoontjes aangekaart. Omdat er soms sprake is van overlappende werkzaamheden (bijvoorbeeld in het geval van printers) en het niet altijd duidelijk is bij welke afdeling een gebruiker moet zijn, kan het gebeuren dat telefoontjes bij de verkeerde afdeling terecht komen. De gebruiker moet dan doorgestuurd of doorverbonden worden. Hoewel dit gezien kan worden als een vorm van werkverschaffing, is dit punt toch meegenomen in de probleemanalyse. Het oplossen van dit probleem kan immers de efficiëntie verhogen.

De definitieve lijst met concrete punten:

- Er ontbreekt een duidelijke, uniforme wijze van het afhandelen van incidenten. Helpdeskmedewerkers hebben geen overzicht van de openstaande incidenten, wie met welke taken bezig is en hoeveel tijd hiervoor staat.
- De informatievoorziening aan de eindgebruikers (de klant) verdient verbetering. Soms is niet duidelijk voor de gebruiker wat de status van zijn of haar incident is dan wel wanneer deze in behandeling wordt genomen.
- Door het niet vastleggen van binnengekomen incidenten wordt het niet altijd duidelijk dat er achterliggende problemen bestaan. Wanneer problemen wel geïdentificeerd en opgelost zijn, wordt soms niet vastgelegd wat de oplossing is geweest.
- De helpdesks worden belast met veelvoorkomende problemen die de gebruikers gemakkelijk zelf op kunnen lossen. Dit veroorzaakt een onnodige belasting van de helpdesks. Een aanname hierbij is dat de gebruiker het probleem zelf op wil lossen. Uit de praktijk is gebleken dat het sommige gebruikers niet interesseert wat het probleem is en hoe deze opgelost kan worden; zij willen slechts zo snel mogelijk geholpen worden.
- Het management heeft geen inzicht in de resultaten van de afdelingen met betrekking tot de opgeloste problemen en afgehandelde incidenten.
- Veel kennis moet mondeling worden overgedragen aan nieuwe medewerkers.
- Door gebrek aan kennis van het onderwerp kan het voorkomen dat een incident of aanvraag bij de verkeerde afdeling terecht komt.

- Het komt geregeld voor dat gebruikers een e-mail sturen naar een specifieke medewerker van één van de afdelingen met vragen of verzoeken. Wanneer deze medewerker niet aanwezig is, kan een kwestie onnodig lang blijven liggen.

De bovenstaande lijst diende als één van de bronnen bij het definiëren van de systeemeisen, welke in paragraaf 3.4 worden besproken.

3.3 ITIL en helpdesks binnen de organisatie

De invoering van een geautomatiseerd helpdesksysteem kan veel gevolgen hebben voor een organisatie, zeker wanneer de bedrijfscultuur in ogenschouw wordt genomen. Men krijgt te maken met een verandering van werkwijze en nieuwe taken en verantwoordelijkheden worden geïntroduceerd.

In deze paragraaf wordt besproken op welke wijze rekening is gehouden met de heersende bedrijfscultuur, hoe de besluitvorming is geweest met betrekking tot de helpdesks in de toekomstige situatie en op elke wijze het ITIL raamwerk is aangehouden.

3.3.1 Bedrijfscultuur en visie

Het afdelingsplan van de afdeling Facilitaire Dienst geeft als één van de cultuurcompetenties waaraan de afdeling extra aandacht wil besteden het volgende:

Klantgerichtheid: De FD heeft voornamelijk de Interne Klant = al onze AWV medewerkers. Onze collega verwacht ongevraagd een prettige, efficiënte en respectvolle behandeling. Wij als FD verwachten bij voorkeur een dito verzoek. Wij staan graag tot uw dienst.

Uit deze cultuurcompetentie wordt duidelijk dat de medewerkers gezien kunnen worden als interne klant. Hierbij is het nodig te verplaatsen in de rol van de gewone gebruiker (de interne klant) binnen dit project. Wanneer deze een probleem of vraag heeft, interesseert het hem of haar niet welke afdeling verantwoordelijk is voor de afhandeling of wat de oplossing is. De gebruiker wil zo snel en efficiënt mogelijk geholpen worden, en staat altijd centraal.

De bedrijfscultuur binnen de AWV kan gekenmerkt worden als een vrij losse en informele cultuur. Men is gewend om bij elkaar binnen te lopen en snel een vraag te stellen of een probleem neer te leggen. De afweging moet gemaakt worden tussen het behouden van de huidige bedrijfscultuur (met betrekking tot het melden van incidenten) en het invoeren van een strenger 'slot op de deur' beleid, waarbij incidenten alleen nog via het nieuwe systeem gemeld kunnen worden. Het voordeel van deze laatste aanpak is dat men op de betrokken afdelingen meer structuur kan aanbrengen en niet 'lastig gevallen' wordt met telefoontjes en e-mails: alle meldingen komen immers op één centrale plek binnen.

Hier kleeft een aantal nadelen aan:

- De gebruiker kan moeite hebben met het onder woorden brengen van het probleem of weet niet precies wat het probleem inhoudt. Het gevolg is een onvolledige of foutieve omschrijving van het incident.
- Het incident is van dusdanige omvang of complexiteit dat dit beter persoonlijk kan worden besproken.
- Het persoonlijke contact tussen de medewerkers onderling gaat voor een deel verloren
- De afwisseling in het werk voor de medewerkers kan verminderd worden

Besloten is dat de ‘vrije’ bedrijfscultuur zoveel mogelijk behouden moet blijven. De gebruiker moet de mogelijkheid houden om bij de afdeling binnen te lopen of een e-mail te sturen om een incident aan te melden. Hiermee wordt de mogelijke weerstand tot een minimum beperkt, omdat er voor de gewone gebruiker geen negatieve veranderingen optreden. De gebruiker heeft immers méér mogelijkheden dan vóór de invoering van het nieuwe systeem: het snel aanmelden van een incident vanaf de eigen werkplek en de status hiervan opvragen, en toegang tot gestructureerde kennis via het Kennissysteem. De hoop en verwachting is dat gebruikers zelf de voordelen hiervan ontdekken en steeds meer gebruik gaan maken van het systeem.

3.3.2 Helpdesks binnen de organisatie

Binnen de AWW fungeren de afdelingen ICT en Facilitaire Dienst als helpdesk, en zijn beschikbaar om vragen te beantwoorden en problemen op te lossen. De applicatie-beheerders, die ieder een deel van het primaire bedrijfssysteem Wocas/X onder de hoede hebben, zijn het aanspreekpunt voor de gebruikers op de desbetreffende afdeling en fungeren als een ‘eenpersoonshelpdesk’. Deze medewerkers geven ondersteuning op het gebied van technische dienst, financiën, verhuurzaken en huuradministratie. Hoewel de term ‘helpdesk’ niet binnen de organisatie wordt gebruikt, kan deze opstelling toch geclassificeerd worden als zodanig. Er is hier sprake van afdelingen of personen die specialistische kennis bezitten over een bepaald vakgebied en de meeste incidenten zelf op kunnen lossen. Binnen ITIL (zie de volgende paragraaf) wordt dit aangeduid als een ‘Expert Service Desk’.

Bij aanvang van het project speelde het vraagstuk of het nuttig was om, naast een geautomatiseerd meldpunt, één centraal telefoonnummer in te stellen. Gebruikers kunnen dan bellen naar dit telefoonnummer, waar zij hun vraag of probleem kunnen afleveren. De medewerker die dit telefoonnummer bemant, voert dit vervolgens in het systeem in en zorgt ervoor dat het incident bij de juiste afdeling of persoon terecht komt. Bij een dergelijke opstelling zou een vorm van een ‘Callcenter’ [5] ontstaan. Een Callcenter (zoals gedefinieerd binnen het ITIL raamwerk) registreert alleen incidenten en doet niets aan de oplossing. Dit heeft als voordeel dat de klantvriendelijkheid wordt vergroot. Gebruikers hoeven, wanneer zij telefonisch hulp nodig hebben, maar één nummer te bellen. Nadelen hiervan zijn dat er te allen tijde een medewerker beschikbaar moet zijn om telefoontjes aan te nemen, dat deze medewerker genoeg kennis van het onderwerp moet hebben om zorg te dragen voor de juiste routing van de melding en dat het gevaar bestaat dat medewerkers van verschillende afdelingen ongewenst taken van elkaar over gaan nemen.

Hoewel dit goed zou aansluiten bij de cultuurcompetentie van Facilitaire Dienst genoemd in de vorige paragraaf, is hierover nog geen definitief besluit genomen. Voorlopig blijft de situatie dus ongewijzigd. Wel wordt er mogelijk een proef van 3 maanden opgestart waarin een centraal telefoonnummer in gebruik wordt genomen, waarna de resultaten geëvalueerd kunnen worden. Bij de ontwikkeling van het systeem is rekening gehouden met de mogelijkheid van de invoer van een Callcenter.

3.3.3 Invoering ITIL en helpdesksysteem

De IT Infrastructure Library (ITIL) [5] kan in het kort worden samengevat als een verzameling 'best practices', waarin het leveren en ondersteunen van IT-diensten centraal staat. Hierbij geeft ITIL richtlijnen met betrekking tot het ontwerpen en afstemmen van processen en het definiëren van de activiteiten hieromtrent. Het is aan de organisatie om te bepalen welke procedures nuttig en toepasbaar zijn om in te voeren. ITIL is in dit project als houvast of basis gebruikt om de centrale delen van het systeem (het beheren van incidenten en problemen) in te richten.

De ITIL richtlijnen zijn onder te verdelen in zes hoofdprocessen. Deze zijn in de onderstaande figuur weergegeven.



Figuur 6: ITIL hoofdprocessen

Bij de Service Desk wordt een probleem aangemeld, dat door de Service Desk aan Incident Management wordt doorgegeven. Dit proces houdt zich bezig met het zo snel mogelijk verhelpen van incidenten. Bij het uitvoeren van Incident Management wordt geprobeerd om structurele fouten op te sporen welke worden doorgegeven aan Problem Management. Binnen Problem Management wordt vastgesteld dat er een verandering in de IT infrastructuur moet plaatsvinden, en wordt dit aan het Change Management proces kenbaar gemaakt. Change Management zorgt voor het juiste afhandeling en doorvoer van wijzigingen. Hierin wordt door middel van Configuration Management proces de configuratie van de infrastructuur (zowel hardware als software) in betrokken. Daarnaast is het denkbaar dat nieuwe releases van software moeten worden ingevoerd onder leiding van het Release Management proces. Tussen de processen kan onderling uitwisseling van informatie plaatsvinden.

In de volgende paragrafen volgt een beschrijving van de genoemde ITIL processen, en in welke mate zij zijn doorgevoerd binnen het project. Meer informatie rond de besluitvorming over de ITIL processen is terug te vinden in paragraaf 3.2 van de Definitiestudie (bijlage D). De relevante processen binnen de hoofdonderwerpen zijn met behulp van de UML modelleringstechniek omgezet naar activiteitendiagrammen. Deze worden besproken in paragraaf 4.5.

Service Desk

De Service Desk is binnen ITIL het (centrale) aanspreekpunt van de IT-organisatie voor de gebruikers. Het doel van de Service Desk is het verlenen van ondersteuning in een organisatie door de bereikbaarheid en toegankelijkheid van de automatisering te garanderen. De Service Desk onderhoudt contacten met de gebruikers, biedt ondersteuning van de applicaties en voert overige werkzaamheden en beheertaken uit. De Service Desk wordt in dit verband meestal als een (aparte) afdeling gezien. De Service Desk kan in contact staan met alle andere hoofdonderwerpen.

Zoals al eerder is aangegeven, beschikt de AWV niet over een centrale Service- of Helpdesk waar alle incidenten en verzoeken binnenkomen. Alle incidenten komen binnen bij de desbetreffende afdelingen. Dit zal in de toekomst mogelijk gaan veranderen na evaluatie van de proef van 3 maanden. Er zal dan geen strikte Service Desk (volgens het ITIL raamwerk) ontstaan, maar er ontstaat - zoals gezegd - een centraal telefonisch meldpunt. De mogelijke Service Desk zal alleen incidenten aannemen via de telefoon, deze in het systeem invoeren en zorgen dat deze aan de juiste probleemoplosser worden toegekend.

Op basis van de genomen besluiten is vastgesteld dat een Service Desk binnen de AWV in de volgende vorm geïntroduceerd zal worden:

- Het doel van een centraal meldpunt wordt bereikt door de eindgebruikers de mogelijkheid te geven via een invulformulier een incident te registreren. De aard van het incident is niet belangrijk: alle incidenten kunnen centraal worden gemeld. Dit is enigszins vergelijkbaar met een Service Desk zoals deze binnen ITIL wordt beschreven.
- Een medewerker van de Service Desk (afdeling of individuele medewerker) neemt de incidenten die niet via het systeem kunnen worden ingevoerd telefonisch aan.
- Informatievoorziening naar de gebruikers toe vindt plaats in de vorm van een Kennissysteem en door de mogelijkheid te bieden de status van incidenten op te vragen. Het Kennissysteem kan gezien worden als de 1^e lijn, terwijl de medewerkers van de helpdesks gezien kunnen worden als de 2^e lijn.

De invoering van een Service Desk beperkt zich in eerste instantie tot een centraal elektronisch meld- en informatiepunt.

Incident Management (Incidentbeheer)

Incident Management (Incidentbeheer) kan gezien worden als het belangrijkste of centrale proces binnen ITIL. Het doel van dit proces is om (dreigende) storingen van de IT-diensten te verhelpen en ervoor zorgen dat de gebruikers zo snel mogelijk weer aan het werk kunnen. Daarnaast heeft Incidentbeheer de taak om storingen zodanig te registreren dat er later onderzoek gedaan kan worden naar de (structurele) oorzaak. Dit hoofdonderwerp staat doorgaans in nauw contact met Problem Management (Probleembeheer), omdat incidenten kunnen leiden tot het onderkennen van een structureel probleem.

Binnen het ITIL raamwerk worden binnen dit hoofdonderwerp de volgende activiteiten onderkend:

- **Aannemen & registreren:** het incident wordt aangenomen en het incident wordt geregistreerd, waarbij onder andere basisgegevens en de prioriteit worden vastgelegd.
- **Classificeren:** het incident krijgt een classificatie van soort en wordt ingedeeld in een categorie.
- **Matchen:** er wordt gekeken of het incident overeenkomt met bekende (gedocumenteerde) problemen waar al een oplossing voor bestaat. Is dit het geval dan wordt het incident afgehandeld door de betreffende oplossing toe te passen en wordt het incident gekoppeld aan het bekende probleem.
- **Analyse & diagnose:** als er geen oplossing voorhanden is dan wordt hier onderzoek naar gedaan.
- **Oplossen & herstellen:** is de oplossing van een incident gevonden, dan wordt deze toegepast. Wordt een oplossing binnen Incidentbeheer gevonden voor een incident waaraan een Probleem is gekoppeld, dan wordt dit bekend gemaakt aan Probleembeheer. Andersom kan hetzelfde gebeuren: wordt een oplossing gevonden voor een Probleem, kan Incidentbeheer genotificeerd worden en kunnen gekoppelde incidenten gesloten worden.
- **Afsluiten:** met de gebruiker wordt afgestemd dat de oplossing naar tevredenheid is uitgevoerd, waarna het incident kan worden afgesloten.
- **Voortgang volgen & bewaken:** de volledige cyclus wordt gevolgd; wanneer nodig kan escalatie plaatsvinden. Bij escalatie kan besloten worden meer menskracht in te zetten, de deadline te verplaatsen of het incident aan een meer gespecialiseerde medewerker toe te wijzen.

Besloten is om Incidentbeheer voor het grootste gedeelte volgens de ITIL richtlijnen over te nemen in het nieuwe systeem. Er zal worden afgeweken op een aantal punten in de procesgang.

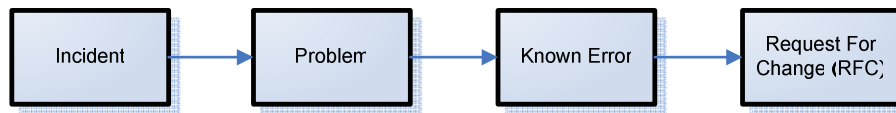
Deze wijzigingen hebben voornamelijk betrekking op de beslissing om gebruikers de mogelijkheid te geven zelf een incident in te voeren, waarmee een onderscheid wordt gemaakt tussen de aanmelder van het incident en de acties die daarop betrekking hebben. Daarnaast is de activiteit van het matchen van incidenten aan bekende problemen optioneel gemaakt. De helpdeskmedewerker kan zelf beslissen - aan de hand van inzicht en ervaring - of er gezocht moet worden in de historie van afgehandelde problemen. Omdat er onderscheid wordt gemaakt tussen incidenten die (mogelijk) problemen veroorzaken en incidenten die betrekking hebben op verzoeken van gebruikers is de helpdeskmedewerker hierin vrij gelaten.

De volledige lijst met wijzigingen is terug te vinden in paragraaf 3.2 van de Definitiestudie. Op basis van de genoemde activiteiten en het ITIL procesmodel is een activiteitendiagram opgesteld, om aan te geven hoe het aanmelden en afhandelen van incidenten zal gaan verlopen.

Problem Management (Probleembeheer)

Problem Management is het proces dat zich richt op het opsporen en wegnemen van structurele fouten in de IT-infrastructuur. Om dit doel te bereiken kan de historie van geregistreerde incidenten geraadpleegd worden om de impact en frequentie te achterhalen.

Is er eenmaal vastgesteld dat een probleem bestaat, dan wordt deze gedefinieerd en kan er een oplossing voor het probleem worden gezocht. Problemen komen meestal voort uit één of meer incidenten. Is de oorzaak van het probleem gevonden, dan ontstaat een Known Error: een bekende fout. Hieruit kan eventueel een Request For Change (een wijzigingsverzoek) volgen. De onderstaande figuur geeft dit weer:



Figuur 7: Escalatie van een incident volgens ITIL

Binnen het ITIL raamwerk bestaat Probleembeheer uit een vijftal activiteiten:

- **Problem Control:** het uitvoeren van probleemdefinities en diagnoses. Wanneer van toepassing kunnen geregistreerde incidenten worden gekoppeld aan het probleem.
- **Error Control:** het vastleggen van Known Errors (Bekende Problemen) en bijbehorende RFC's.
- **Volgen en bewaken van Problems en Known Errors:** het bewaken van de (naderende) deadline en de medewerker hierover alarmeren.
- **Pro-active Problem Management:** het uitvoeren van trendanalyses om voorspellingen te doen.
- **Managementrapportage:** het genereren van rapportage aan de hand van de aanwezige informatie.

Besloten is Probleembeheer gedeeltelijk volgens het ITIL raamwerk over te nemen. Deze beslissing is gebaseerd op twee factoren. Omdat Change Management (zie volgende paragraaf) buiten het bereik van het project valt, zal het niet mogelijk zijn RFC's bij te houden. De activiteit Error Control zal dus gedeeltelijk worden ingevoerd. Het ontwikkelen van functionaliteit die de activiteit Pro-active Problem Management mogelijk moet maken (het analyseren van trends) valt eveneens buiten het bereik van het project.

De volledige besluitvorming rond Probleembeheer is terug te vinden in paragraaf 3.2.3 van de Definitiestudie (bijlage D).

Change-, Configuration- en Releasemanagement

Deze hoofdonderwerpen houden zich respectievelijk bezig met het registreren en afhandelen van RFC's, het beheren van informatie over de infrastructuur en het waarborgen van de kwaliteit van de productieomgeving. Deze hoofdonderwerpen vallen allen buiten het bereik van het project, zoals dit is afgebakend in de opdrachtomschrijving.

3.4 Definiëren systeemeisen

Het definiëren van de systeemeisen en het bepalen van de inrichting van de toekomstige helpdesks is in nauw verband gedaan. De eisen die aan het systeem worden gesteld hangen immers samen met de keuzes die gemaakt zijn met betrekking tot de inrichting van de helpdesks. De systeemeisen zijn niet uitsluitend voortgekomen uit de (toekomstige) inrichting van de helpdesks, maar ook uit ideeën van medewerkers, mogelijkheden die vergelijkbare helpdeskpakketten bieden en de probleemanalyse.

Er is, aan de hand van de richtlijnen gegeven door IAD, een onderscheid gemaakt tussen basissysteemeisen, integriteitseisen, performance-eisen en operationele eisen. De systeemeisen zijn op deze plek niet geprioriteerd. Wanneer aan het einde van het project alle systeemeisen zijn ingewilligd is een systeem operationeel dat men in de fase Definitiestudie voor ogen had. Dit wil echter niet zeggen dat aan alle systeemeisen moet zijn voldaan voordat het systeem in gebruik kan worden genomen. De systeemeisen zijn bij de evaluatie van het project gebruikt om een oordeel te vellen over de mate van compleetheid van het beoogde systeem.

In de volgende paragrafen zullen een aantal centrale en belangrijke systeemeisen worden besproken. Het volledige overzicht met systeemeisen is terug te vinden in hoofdstuk 2 van de Definitiestudie (bijlage D).

3.4.1 Basissysteemeis A2: Wachtrij en notificatie

Alle openstaande incidenten (calls) moeten in eerste instantie in een wachtrij binnenkomen. Medewerkers van de helpdesks moeten eventueel genotificeerd worden van binnenkomende incidenten.

Voor aanvang van het project is gekeken naar een aantal bestaande helpdeskpakketten (waarvan een deel gebaseerd is op het ITIL raamwerk) zoals TopDesk, Track-IT en SMILE Complaint Management. Een kenmerk van deze pakketten is dat zij de gebruikers van de (IT) diensten, niet de mogelijkheid geven zelf een melding direct in het systeem in te voeren. Dit kan alleen gedaan worden door medewerkers van het Callcenter of de Service Desk. Aangezien gebruikers bij de AWV over het algemeen wel weten wat het probleem of verzoek is en deze kunnen omschrijven, is het niet meer dan logisch de gebruikers de mogelijkheid te geven zelf een incident - direct in het systeem - aan te kunnen melden. De stap van het telefonisch, per e-mail of op andere wijze doorgeven van een omschrijving kan dan vervallen. Dit roept wel de vraag op: "bij welke helpdeskmedewerker moet het incident binnenkomen?". De oplossing is het introduceren van een gezamenlijke wachtrij, waar alle binnenkomende incidenten in terecht komen. De helpdeskmedewerker kan de incidenten oppakken. Om ervoor te zorgen dat incidenten niet in de wachtrij blijven staan moet er gezorgd worden voor een wijze van notificatie naar de helpdeskmedewerker toe. Hiervoor is systeemeis D1 opgenomen.

3.4.2 Basissysteemeis A6: Doorverwijzen incidenten

De mogelijkheid moet bestaan om incidenten door te verwijzen naar een andere helpdeskmedewerker of naar een bepaalde oplosgroep. In het laatste geval kan een medewerker van de desbetreffende oplosgroep vervolgens vanuit de wachtrij een behandelaar aan het incident koppelen.

Door deze systeemeis wordt het escaleren van incidenten mogelijk gemaakt. Het kan voorkomen dat een medewerker aan het oplossen van een incident werkt, en zelf het incident niet succesvol kan afsluiten. Ook is het denkbaar dat een incident tijdens de behandelperiode aandacht vereist van meerdere medewerkers, binnen één of meerdere oplosgroepen. Binnen dit project wordt een oplosgroep gedefinieerd als een groep (helpdesk)medewerkers die samen verantwoordelijk zijn voor een deel van de (IT) dienstverlening. In het geval van de ICT en Facilitaire Dienst uit zich dit in aparte afdelingen, terwijl de applicatiebeheerders verdeeld zijn over verschillende afdelingen.

Deze eis hangt samen met systeemeis A7, die onder meer stelt dat het mogelijk moet zijn incidenten van andere helpdeskmedewerkers in dezelfde oplosgroep over te nemen. Dit lost het punt uit de probleemstelling op welke stelt dat zaken die via e-mail zijn aangemeld mogelijk blijven liggen door ziekte of afwezigheid. Wanneer een medewerker niet aanwezig is kunnen de incidenten die zijn toegewezen aan de medewerker overgenomen worden door een andere medewerker in dezelfde oplosgroep. Wel vereist dit een zekere mate van discipline van de helpdeskmedewerkers om niet zonder reden incidenten van elkaar over te gaan nemen.

3.4.3 Basissysteemeis A9: Factuuroverzichten

Het moet mogelijk worden een (factuur) overzicht te genereren voor afgehandelde incidenten, om zo een doorbelasting te kunnen maken. Door een periode op te geven moet een overzicht worden gegenereerd van alle incidenten die binnen die periode zijn afgehandeld voor een externe klant. Dit in verband met een mogelijke samenwerking met Rochdale, waarvoor in de toekomst eventueel werkzaamheden uitgevoerd gaan worden.

Een groot voordeel van een geautomatiseerd helpdesksysteem is dat gegevens permanent opgeslagen worden en op elk gewenst tijdstip weer opgevraagd kunnen worden. Hoe langer er met een dergelijk systeem wordt gewerkt, hoe meer informatie er beschikbaar is. Niet alleen kan het voor de helpdeskmedewerkers eenvoudiger worden incidenten op te lossen aan de hand van eerder geconstateerde problemen, het genereren van managementrapporten behoort ook tot de mogelijkheden. Binnen ITIL wordt dit bij nagenoeg alle hoofdonderwerpen genoemd.

Bij het definiëren van de systeemeisen werd door één van de medewerkers van de afdeling ICT aangekaart dat de verwachting is dat bij het ingaan van het samenwerkingsverband met woonstichting Rochdale er diensten zullen worden verleend voor externe partijen of opdrachtgevers, waaronder Rochdale. Dit gebeurt in de huidige situatie al in beperkte mate. Op basis van de gewerkte uren en het uurtarief worden deze doorberekend binnen de administratie.

3.4.4 Performance-eis G1: Verstoring primaire werkprocessen

Het systeem mag geen verstoring of vertraging (door middel van het aantal database-transacties op de Microsoft SQL 2000 server) veroorzaken van de primaire werkprocessen, zoals het DisKis systeem.

Binnen de fase Definitiestudie is vastgesteld dat de centrale database van het te implementeren systeem geplaatst moet gaan worden op één van de servers waar ook de primaire bedrijfsapplicatie DisKis op draait. Hierdoor is het van belang dat het nieuwe systeem geen negatieve invloed heeft op het DisKis systeem. Dit kan gemeten worden met SQL Server 2000 hulpprogramma's die beschikbaar zijn. Denk hierbij aan de belasting op de processor, het beschikbare interne geheugen en het netwerkverkeer dat wordt gegenereerd.

3.4.5 Operationele eis H1: Gebruikersauthenticatie

Het systeem moet vanaf iedere werkplek binnen de organisatie, waarop een gebruiker op een Citrix server is ingelogd, op te starten en te gebruiken zijn. Gebruikers hoeven dus niet apart in het systeem in te loggen.

Gebruikers binnen de AWV werken via een ICA Client (Citrix verbindingsssoftware) op één van de 7 Citrix Servers waarover de AWV beschikt. Op deze wijze kunnen vrijwel alle benodigde applicaties worden gestart. Door middel van load balancing wordt de gebruiker toegewezen aan één van de servers. De beeldschermuitvoer wordt over het netwerk naar de cliënts toegestuurd.

Een gebruiker dient bij het starten van de cliëntmachine in te loggen op de Citrix server via het Windows 2000 domein door middel van een gebruikersnaam en wachtwoord. Na het succesvol inloggen is de gebruikersnaam van de gebruiker bekend en kan er vanuit worden gegaan dat de persoon die de machine gebruikt ook daadwerkelijk de gebruiker is die aan het werk is. Het is dus overbodig dat een gebruiker apart in het beoogde systeem inlogt, omdat er bij het inloggen op de server al authenticatie heeft plaatsgevonden.

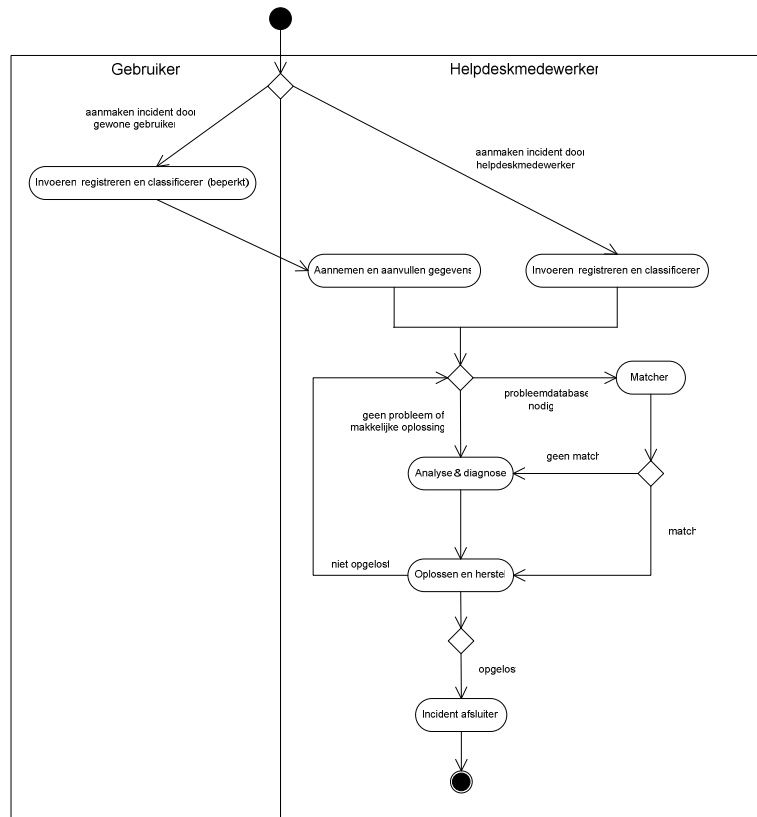
Paragraaf 4.9 gaat dieper in op de technische inrichting binnen de organisatie.

3.5 Vervaardigen activiteitendiagrammen

Bij het vaststellen van de wijze van het doorvoeren van de ITIL principes, zijn de relevante processen omgezet naar UML activiteitendiagrammen [3]. Deze activiteitendiagrammen zijn gebaseerd op de voorgedefinieerde processchema's uit ITIL [5]. Deze geven aan op welke wijze de activiteiten per ITIL-hoofdonderdeel worden uitgevoerd. Er is binnen het project zoveel mogelijk vastgehouden aan de UML modelleringstechniek, om de uniformiteit tussen de verschillende soorten modellen zo groot mogelijk te houden.

Daarnaast is het mogelijk om in activiteitendiagrammen zogenaamde 'swimlanes' op te nemen, waardoor het diagram kan worden opgesplitst in verschillende banen. Deze methode is gebruikt om een onderscheid te maken tussen activiteiten die gewone gebruikers en behandelende helpdeskmedewerkers uitvoeren. Ook is bij een activiteitendiagram duidelijk te zien waar een object binnen de activiteit begint en eindigt.

De onderstaande figuur geeft een voorbeeld van de vervaardigde activiteitendiagrammen, en laat zien op welke wijze Incidentbeheer zal worden doorgevoerd.



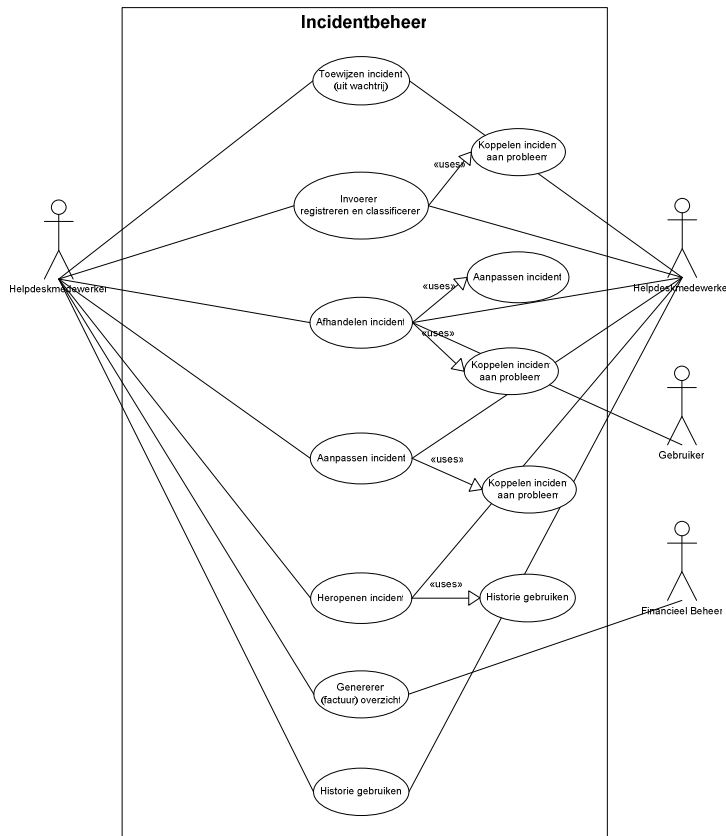
Figuur 8: Activiteitendiagram Incidentbeheer, op basis van het ITIL proces

3.6 Opstellen Use-cases

Centraal binnen het definiëren van het systeemconcept staat de interactie van de gebruiker met het systeem. Om deze op een uniforme wijze weer te geven is gekozen om de interactie op te stellen aan de hand van UML Use-cases. Een bekend alternatief hiervoor is het opstellen van taakscenario's uit GUIDE. Beide manieren bereiken hetzelfde doel. Omdat gekozen is zoveel mogelijk de UML modelleringstechniek te hanteren, zijn UML Use-cases gemaakt. Belangrijk om op te merken is dat de Use-cases alleen aangeven welke interactie de gebruiker met het systeem heeft, en niet hoe deze interactie specifiek plaatsvindt. Dit wordt uitgewerkt in de pilotontwikkelplannen, die in hoofdstuk 5 worden behandeld.

De Use-cases zijn voor de leesbaar- en duidelijkheid opgesplitst in de gebieden Incidentbeheer, Probleembeheer, Kennissysteem, Incidentinvoer- en Incidentstatus en Algemeen.

Een complete Use-case binnen UML bestaat uit één of meerdere Use-cases en een bijbehorend Use-case diagram. Een Use-case geeft een individueel uit te voeren taak aan, die uit meerdere acties kan bestaan. Het Use-case diagram geeft de verhouding tussen Use-cases en de betrokken personen aan. De onderstaande figuur geeft een voorbeeld van een Use-case diagram:



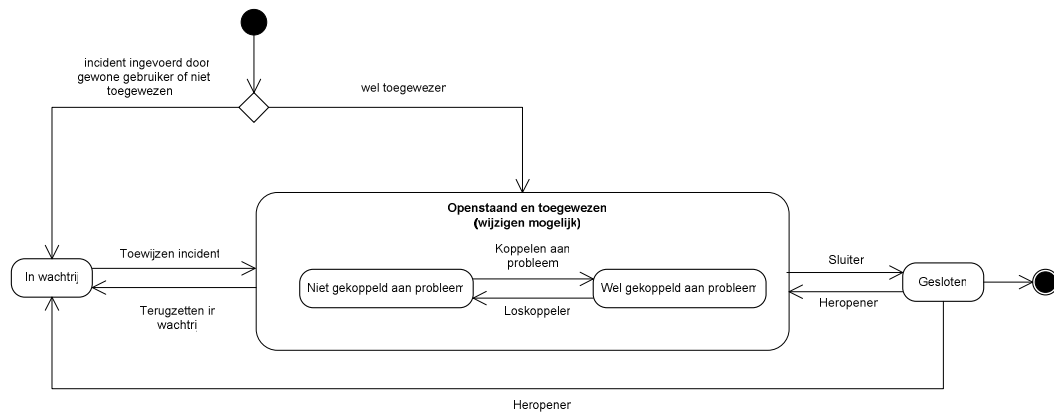
Figuur 9: Use-case diagram hoofdonderdeel Incidentbeheer

De individuele Use-cases zijn opgesteld aan de hand van het beeld van het systeem dat op dat moment aanwezig was. Een Use-case wordt gevormd door een beschrijving van de taak, aannamen die worden gedaan, pre- en postcondities, een lijst van stappen die uitgevoerd wordt en de profiterende 'acteur'. De profiterende acteur is de persoon of groep die profijt of voordeel heeft van het uitvoeren van de taak. Om de complete interactie met het systeem weer te geven zijn in totaal 23 Use-cases opgesteld, die terug te vinden zijn in paragraaf 3.3 van de Definitiestudie (bijlage D).

3.7 Vervaardigen toestandsdiagrammen

Binnen de fase Definitiestudie is een tweetal UML toestandsdiagrammen vervaardigd. Deze geven aan in welke toestand incidenten en problemen zich kunnen bevinden.

Als voorbeeld wordt hier het toestandsdiagram genomen dat aangeeft in welke toestanden incidenten zich kunnen bevinden:



Figuur 10: Toestandsdiagram Incidenten

Te zien is dat een incident zich in drie verschillende hoofdtoestanden kan bevinden: in de wachtrij, openstaand (en toegewezen, dus niet in de wachtrij) en gesloten. Een incident kan alleen worden gewijzigd wanneer het zich in de tweede toestand bevindt; in de andere twee toestanden kunnen de gegevens alleen worden opgevraagd. Dit besluit is genomen om het scenario te beperken waarin meerdere helpdeskmedewerkers tegelijkertijd aan één incident werken.

Binnen de toestand 'openstaand en toegewezen' zijn twee subtoestanden opgenomen: 'niet gekoppeld aan probleem' en 'wel gekoppeld aan probleem'.

3.8 Bepalen objectklassen en database(model)

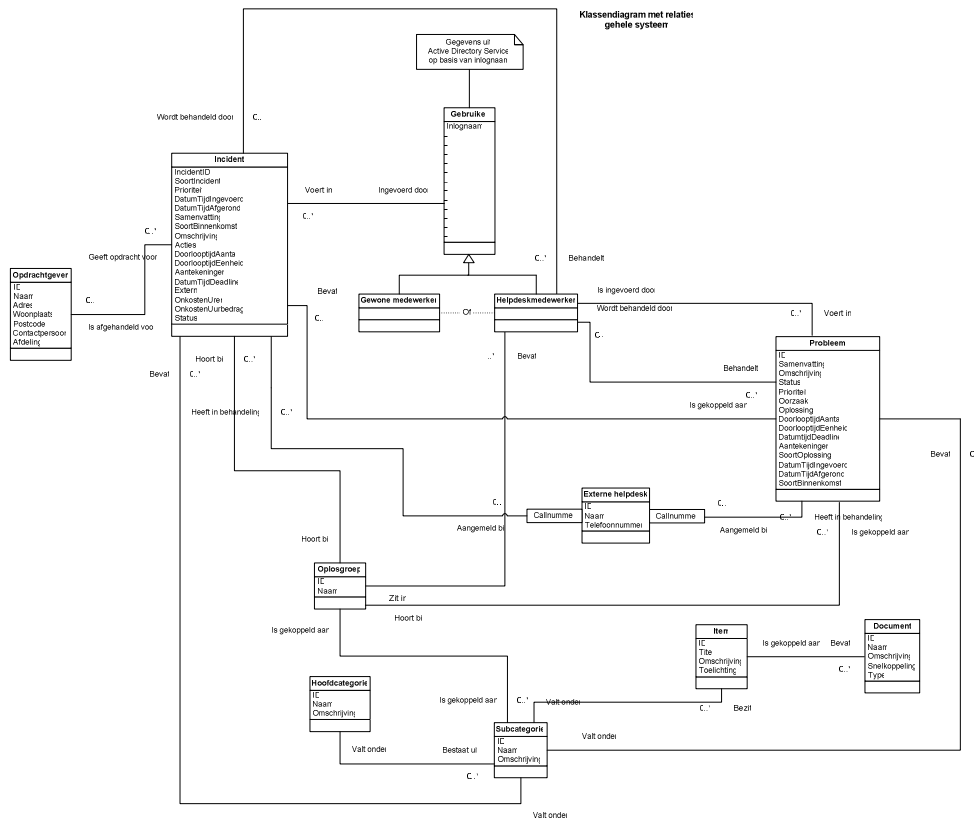
Op basis van het beeld dat is ontstaan van het systeem is in deze fase een relationeel data-model opgesteld. In deze paragraaf zal behandeld worden hoe deze tot stand is gekomen aan de hand van een UML klassendiagram, en welke relevante beslissingen zijn genomen met betrekking tot de gegevensstructuur.

3.8.1 Samenstelling klassen

Om een gegevensmodel te realiseren - welke uiteindelijk zal leiden tot een relationeel databasemodel - is gestart met het opstellen van de individuele klassen aan de hand van de UML modelleringstechniek. De klassen zijn ontstaan door het categoriseren van de benodigde gegevens, afgeleid uit de Use-cases. Bij elke klasse is aangegeven uit welke attributen (gegevens) deze bestaat. Een alternatief voor het opstellen van een klassendiagram is het vervaardigen van een Entity-Relationship-model (ER-model) [6].

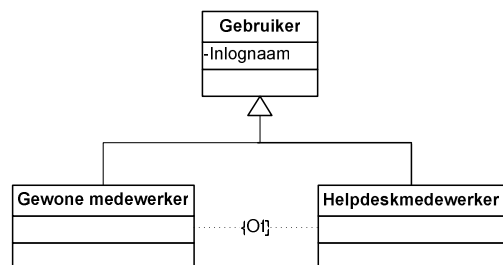
In een dergelijk model worden de attributen bij de klassen als losse ballonnen bij de entiteiten weergegeven. Omwille van de leesbaarheid en de beslissing om zoveel mogelijk met de UML modelleringstechniek te werken is gekozen om een klassendiagram op te stellen.

Door de associaties tussen de individuele klassen aan te geven ontstaat een klassendiagram voor het gehele systeem:



Figuur 11: Klassendiagram voor het gehele systeem

Een opvallende klasse is de klasse 'Gebruiker'. Deze klasse, die uit 2 subklassen bestaat, heeft slechts één attribuut, namelijk 'inlognaam'. De klasse is opgedeeld in subklassen omdat er een verschil bestaat tussen gewone medewerkers en helpdeskmedewerkers met betrekking tot de associaties met andere klassen. Figuur 12 geeft dit weer:



Figuur 12: De klasse 'Gebruiker' in UML notatie

Met de realisatie dat er meer gegevens over een gebruiker bestaan (zoals de voor- en achternaam, de afdeling, het e-mail adres, het telefoonnummer etc.) dan alleen de inlognaam, waarom is dan toch gekozen voor deze inrichting van de klasse?

In eerste instantie was een oplossing bedacht waarbij de gegevens met betrekking tot de aanwezige gebruikers uit het softwarepakket DisKis werden gehaald. De centrale database van dit pakket draait immers op één van de SQL Server 2000 machines van de organisatie; het zou technische mogelijk zijn om de database van het beoogde systeem te synchroniseren met de DisKis database, of rechtstreeks een koppeling met de juiste tabel te maken. Eén van de medewerkers van de afdeling ICT kwam vervolgens met de suggestie rechtstreeks de Active Directory van de organisatie te benaderen om zo gegevens m.b.t. de medewerkers op te halen.

Een Active Directory kan gezien worden als een database waarin een hiërarchische structuur kan worden aangemaakt met daarin groepen (Organisational Units) en gebruikers. De Active Directory wordt onder andere gebruikt om rechten te verlenen voor het inloggen binnen het Windows 2000 domein en de Citrix omgeving en het bepalen van de rechten om applicaties te starten. Na het uitvoeren van enig onderzoek bleek dat het Microsoft .NET framework beschikt over een aantal classes die specifiek zijn ingericht voor het benaderen van een Active Directory. Deze classes werken op basis van het Lightweight Directory Access Protocol (LDAP) – een door het Internet Engineer Task Force (IETF) geïntroduceerde manier om over het TCP/IP protocol toegang te krijgen tot een Active Directory. Op dit punt is vastgesteld dat wanneer de inlognaam van de gebruiker bekend is, de rest van de gegevens uit de Active Directory gehaald kunnen worden. Deze gegevens hoeven dus niet (apart) permanent in een database opgeslagen te worden. Er kan van uit worden gegaan dat de server waarop de Active Directory draait, altijd actief is aangezien de complete IT dienstverlening hierop steunt.

Deze beslissing introduceert wel een andere kwestie. Wanneer medewerkers uit dienst treden, worden zij na verloop van tijd uit de Active Directory verwijderd en zijn de gegevens m.b.t de medewerker niet meer opvraagbaar. Wanneer we met deze kennis vooruitdenken is de volgende situatie denkbaar. Een gewone medewerker voert op een gegeven ogenblik een incident in het systeem in. Door middel van de associatie tussen de (hoofd) klasse ‘gebruiker’ en de klasse ‘incident’ wordt vastgesteld dat het systeem bijhoudt door wie het incident is aangemeld. Het incident wordt succesvol opgelost en door de behandelende helpdesk-medewerker afgesloten. Na verloop van tijd treedt de medewerker die het incident heeft aangemeld uit dienst en wordt uit de Active Directory verwijderd.

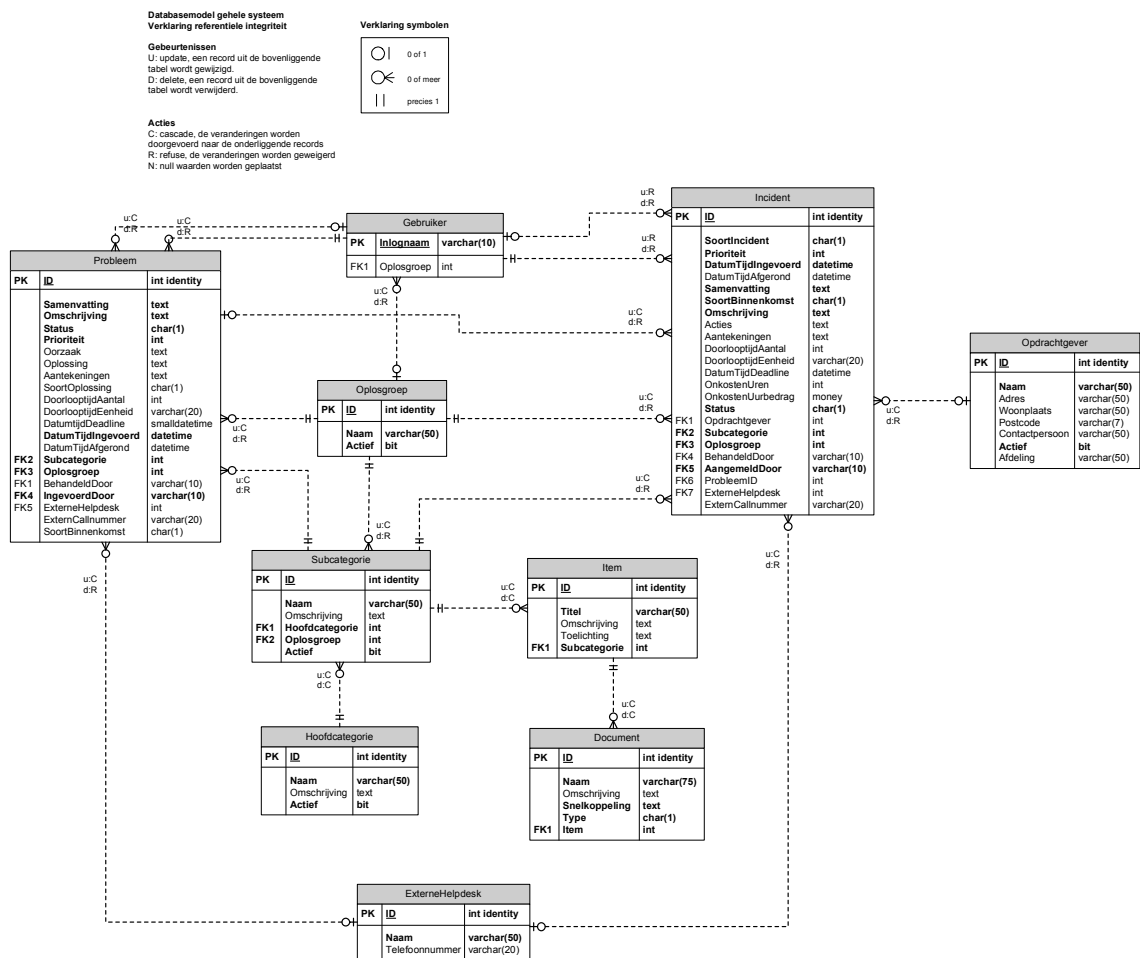
Wanneer nu in de toekomst het reeds gesloten incident wordt opgevraagd is alleen de inlognaam van de aanmelder binnen het systeem bekend. De vraag is dan: moeten alle gegevens met betrekking tot oud-medewerkers beschikbaar blijven? Het antwoord is: nee. Helpdesk-medewerkers hebben de gegevens van medewerkers alleen nodig bij het behandelen van incidenten en problemen, om snel te kunnen zien wat het telefoonnummer van de aanmelder is of om even snel een e-mail bericht te kunnen sturen. Het is niet interessant om bijvoorbeeld te weten wat het telefoonnummer is geweest van een medewerker die jaren geleden in dienst was. Daarnaast kunnen deze gegevens, in het geval dat het onverhoopt toch nodig is, in de administratie worden opgezocht.

Verder is de associatie tussen de klassen ‘Oplosgroep’ en ‘Subcategorie’ van belang op te merken. Deze associatie verzorgt de koppeling tussen een oplosgroep en een subcategorie waaronder een incident kan vallen. Door middel van de associatie kan aan iedere subcategorie een ‘standaard’-oplosgroep worden gekoppeld. Bij het binnenkomen van incidenten via het systeem kan aan de hand van de opgegeven subcategorie het incident automatisch terecht komen bij de standaard oplosgroep. Voor een correcte werking hiervan is het wel nodig dat de gebruiker de juiste subcategorie selecteert bij het invoeren van het incident. Mocht een incident verkeerd worden ingedeeld, zal dit door één van de helpdeskmedewer-

kers snel genoeg worden opgemerkt en zal het incident naar de juiste oplosgroep worden doorgestuurd.

3.8.2 Opstellen relationeel databasemodel

Aan de hand van het klassendiagram, besproken in de vorige paragraaf, is een relationeel databasemodel opgezet, aan de hand van de regels uiteengezet in ‘Databasesystemen in de praktijk’ [6]. Het hieruit voortgekomen databasemodel is in figuur 13 te zien.



Figuur 13: Het relationele databasemodel, afgeleid van het klassendiagram

Bij het opstellen van de tabel 'Gebruiker' is ietwat afgeweken van de vooropgestelde regels. Omdat er sprake is van specialisatie, zoals bij de klasse 'Gebruiker', moet er eigenlijk een extra attribuut opgenomen worden welke aangeeft van welke type de specialisatie is. In dit geval dus een gewone medewerker of een helpdeskmedewerker. Toch is gekozen om dit niet te doen. De reden hiervoor is dat een gebruiker gekenmerkt kan worden door de aanwezigheid in een oplosgroep. Zit deze in een oplosgroep, dan is de gebruiker een helpdeskmedewerker. Zit deze niet in een oplosgroep, dan is deze een gewone gebruiker. Het attribuut 'Oplosgroep' in de tabel 'Gebruiker' bepaalt dus bij welk type een gebruiker hoort.

Naast de tabellen zijn in dit model ook de constraints die voor de tabellen gelden weergegeven. Zo is het bijvoorbeeld toegestaan dat alle gekoppelde documenten mogen worden ver-

wijderd wanneer een Kennissysteem-item uit het systeem wordt gehaald. Aan de hand van het databasemodel is in de SQL Server 2000 Enterprise Manager applicatie een fysieke database aangemaakt en geplaatst op een testserver. Hierbij is dus geen implementatiemodel gemaakt.

3.8.3 Kiezen applicatie-instellingen

In eerste instantie is gekozen om de applicatie-instellingen ook in de database op te nemen, als aparte tabel. Hierbij gaat het om instellingen als de naam en locatie van de database en de naam van de Active Directory server. Na enig onderzoek op een aantal internetfora bleek dat het .NET framework beschikt over een class die speciaal is ingericht voor het bijhouden en ophalen van applicatie-instellingen, deze werkt op basis van Extensible Markup Language (XML). In een XML bestand kunnen de instellingen dan door de beheerder worden bijgehouden. Dit wordt verder besproken in paragraaf 5.5.6.

3.8.4 Besluitvorming Stored Procedures versus embedded T-SQL

Stored Procedures kunnen gedefinieerd worden als procedures of opdrachten, gewoonlijk Structured Query Language (SQL) statements, welke opgeslagen kunnen worden in een database. Transact-SQL (T-SQL) is een uitbreiding op SQL en voegt enkele mogelijkheden als transacties en de mogelijkheid variabelen te declareren in een statement toe. Wanneer een Stored Procedure vanuit een applicatie wordt aangeroepen wordt deze direct op de database-engine uitgevoerd. Een alternatief hiervoor is het opslaan van de statements in de programmacode (embedded), en deze los te laten op de database.

Het gebruik van Stored Procedures heeft een aantal voordelen [7] ten opzichte van embedded T-SQL:

- Er kan een scheiding worden aangebracht tussen de applicatie en de bewerkingen en opvragingen op de database. Dit heeft als voordeel dat wanneer een wijziging moet worden aangebracht, zoals de sorteervolgorde in een T-SQL statement, de applicatie niet opnieuw gecompileerd en ingevoerd hoeft te worden.
- Stored Procedures worden in de database gecompileerd en geoptimaliseerd en werken over het algemeen sneller wanneer grote statements moeten worden uitgevoerd of wanneer statements vaak worden aangeroepen.
- Netwerkverkeer kan tot een minimum worden beperkt. Bij het uitvoeren van een Stored Procedure hoeft alleen de naam en eventuele parameters over het netwerk worden gestuurd, terwijl bij embedded T-SQL statements het complete statement moet worden verstuurd.

Natuurlijk zijn er niet alleen maar voordelen bij het gebruik van Stored Procedures. Wanneer een Stored Procedure op meerdere plaatsen binnen één of vanuit meerdere applicaties wordt aangeroepen en de Stored Procedure wordt veranderd - het kan bijvoorbeeld nodig zijn een extra parameter toe te voegen - dan heeft dit gevolgen voor alle delen van de applicatie waar de Stored Procedure wordt aangeroepen. Toch is de keuze gemaakt om gebruik te maken van Stored Procedures. De voordelen wegen, mits de Stored Procedures zorgvuldig worden geïmplementeerd en gebruikt, op tegen de nadelen en de mogelijkheid is aangegrepen om ervaring op te doen met Stored Procedures.

3.8.5 Bepalen handmatig onderhoud

Handmatig onderhoud aan de database zal moeten plaatsvinden in twee situaties aan de tabellen 'Externe helpdesk' en 'Opdrachtgever'. Er is gekozen een grafische front-end voor deze tabellen niet mee te nemen in dit ontwikkeltraject omdat de verwachting is dat deze twee tabellen zeer zelden gewijzigd moeten worden. Wanneer dit toch nodig is, kan onderhoud aan deze tabellen worden gepleegd door één van de medewerkers van de afdeling ICT. Op deze afdeling is voldoende kennis van databasebeheer aanwezig, en in combinatie met de documentatie over de database in de Definitiestudie mag dit geen problemen opleveren.

3.9 Vaststellen technische aspecten

Om een beeld te vormen van de technische infrastructuur binnen de organisatie, is een inventarisatie opgesteld van de beschikbare servers, de netwerkstructuur, en systeemconfiguraties van de aanwezige cliëntmachines. Deze inventarisatie is terug te vinden in de hoofdstuk 4 van de Definitiestudie (bijlage D).

Naast de inventarisatie is een lijst opgesteld met veranderingen of aanpassingen die moeten plaatsvinden voor de invoering van het nieuwe systeem. Gebleken is dat de aanpassingen beperkt blijven tot het installeren van software of het aanpassen van bestaande software. Hardwarematige aanpassingen hoeven niet plaats te vinden. Het volgende overzicht laat dit zien:

- De database van het nieuwe systeem zal geplaatst moeten worden op één van de MS SQL Server 2000 servers. Naar alle waarschijnlijkheid zal dit één van de servers worden waar het DisKis systeem op draait. Deze server beschikt over een gigabyte aan RAM op een dual 1,4 Ghz machine, en heeft voldoende capaciteit om zowel DisKis als het nieuwe systeem te bedienen.
- De .NET Framework redistributable versie 1.0 of hoger moet worden geïnstalleerd op alle Citrix servers, als dit nog niet is gedaan.
- Een snelkoppeling naar de applicatie van het nieuwe systeem zal in het profiel van alle gebruikers opgenomen moeten worden zodat deze vanaf het bureaublad of het startmenu te benaderen is.
- De applicatie zelf moet worden geplaatst op de uiteindelijke productieserver, en voorzieningen moeten worden getroffen om ervoor te zorgen dat gebruikers toegang tot de applicatie hebben. Benodigde DLL's en executables moeten worden geregistreerd op alle betrokken servers.

3.10 Vaststellen organisatorische aspecten

Naast de beschouwing van de technische aspecten zijn ook de organisatorische veranderingen die plaats zullen vinden bij de invoering van het systeem op een rij gezet. Voor elke gebruikersgroep is aangegeven welke veranderingen er zullen (moeten) plaatsvinden, en wat de positieve en negatieve gevolgen hiervan zijn.

Voor de gewone medewerkers zal de impact van de invoering van het nieuwe systeem relatief laag zijn. De medewerkers krijgen te maken met een nieuwe manier van het melden van incidenten en verzoeken en zij worden voortaan aangespoord eerst in het Kennissysteem te kijken voordat zij een incident melden. Wanneer de positieve aspecten tegenover de negatieve aspecten worden gezet kan geconcludeerd worden dat de negatieve aspecten voornamelijk te maken hebben met gewenning en de ‘opvoeding’ van de gebruikers. De verwachting is dan ook dat na verloop van tijd (na de invoering) de gebruikers minder snel naar de telefoon zullen grijpen en het systeem gebruiken voor de aanmelding van incidenten.

De impact voor de helpdeskmedewerkers zal groter zijn. Bij het uitvoeren van de dagelijkse werkzaamheden zal het nieuwe systeem een rol spelen. Niet alleen hebben de helpdeskmedewerkers de mogelijkheid (niet de verplichting!) om gegevens m.b.t. hun werk vast te leggen, het Kennissysteem zal ook up-to-date gehouden moeten worden. Ook hier is de verwachting dat dit een kwestie van wennen is, en dat na verloop van tijd de voordelen duidelijk zullen worden.

Het complete overzicht met organisatorische veranderingen met de uitgewerkte voor- en nadelen is te vinden in hoofdstuk 5 van de Definitiestudie.

Ook zijn opleidingsdoelstellingen geformuleerd. Zo is bepaald dat middels een e-mail de gewone gebruikers op de hoogte worden gesteld van de ingebruikname van het systeem, en zal er een kort overzicht worden gegeven van de mogelijkheden van het deel van het systeem dat is bedoeld voor de gewone gebruiker. De helpdeskmedewerkers zullen een instructie krijgen om bekend te raken met het systeem en de beschikbare functies vertrouwd te raken. Verder kan altijd teruggevallen worden op de helpfiles.

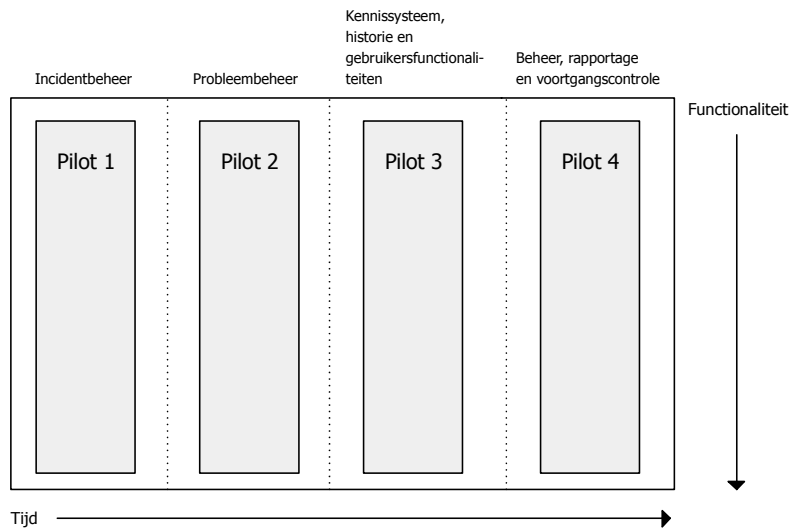
3.1.1 Opstellen pilotplan

Als laatste activiteit in de fase Definitiestudie is het pilotplan voor de gehele ontwikkeling van het systeem vastgesteld. Deze paragraaf gaat in op de pilotstrategie en de besluitvorming rond het opdelen van het systeem in verschillende pilots. Een pilot wordt binnen IAD [1] gezien als “een operationele subset van de functionaliteiten van het uiteindelijk beoogde systeem”.

De pilotstrategie bepaalt de functionele inhoud en onderlinge volgorde van pilots die worden aangehouden bij het ontwikkelen (bouwen) van de pilots. Bij het opstellen van het pilotplan is een vorm van het principe van ‘Juicy Bits First’ [1] aangehouden. Dit principe houdt in dat de meest aansprekende delen van het systeem als eerste zullen worden ontwikkeld, waardoor draagkracht binnen de organisatie kan worden verkregen. Een variant hierop is gehanteerd, waarbij er van uit is uitgegaan dat de meest essentiële of belangrijke delen van het totale systeem voorrang hebben. Hieronder vallen ook de ‘kernfunctionaliteiten’: functionaliteiten die moeten worden ontwikkeld omdat zij als basis dienen voor andere functionaliteiten. Denk hierbij aan bepaalde interfacedelen en toegang naar de database. Secundaire en minder essentiële functionaliteiten, zoals het bijhouden van gebruikers en oplosgroepen zullen dan later in het traject worden ontwikkeld. Dit heeft als voordeel dat al in een vroeg stadium feedback kan worden verkregen over de belangrijkste delen van het systeem en dat aan het einde van het traject in ieder geval het systeem gedeeltelijk, met de meest essentiële functionaliteiten, ingevoerd kan worden.

Met het bovenstaande in gedachten is een pilotstrategie gekozen. De keuze is gevallen op het ontwikkelen van de pilots met een diepe functionaliteit. Dit houdt in dat het gehele systeem wordt opgedeeld in pilots die achtereenvolgens compleet worden ontwikkeld. Daarnaast is rekening gehouden met de herbruikbaarheid van componenten (de pilotstrategie 'herbruikbare componenten').

Dit is weergegeven in de volgende figuur:



Figuur 14: Pilotstrategie 'diepe functionaliteit'

De eerste pilot richt zich op het ontwikkelen van Incidentbeheer, de tweede pilot op Probleembeheer, de derde pilot op het Kennissysteem, de historie en het deel van het systeem dat voor de gewone gebruikers bestemd is en de vierde en laatste pilot op beheer- rapportage en voortgangscntrolefuncties. De pilots zijn op deze manier samengesteld en geprioriteerd op basis van belangrijkheid (Incidentbeheer heeft een hogere prioriteit dan de beheerfuncties). De gedefinieerde pilots kunnen zelfstandig worden ontwikkeld en getest.

Alternatieve pilotstrategieën, zoals deze binnen IAD worden aangedragen, zijn bekeken en beoordeeld op bruikbaarheid. Enkele van deze alternatieven met de overwegingen daarbij zijn:

- Brede functionaliteit. In tegenstelling tot het ontwikkeling met diepe functionaliteit worden bij deze pilotstrategie alle onderdelen van het systeem in iedere pilot voor een deel ontwikkeld. Deze strategie is nuttig in situaties waar een sterke afhankelijkheid tussen de subsystemen/onderdelen bestaat. In dit geval is het beoogde systeem goed op te delen in afzonderlijk te ontwikkelen en te testen onderdelen. Pas bij de invoering van alle onderdelen ontstaat de samenhang ertussen.



- Parallellisme. Deze strategie kan gebruikt worden wanneer er met meerdere ontwikkelaars of ontwikkelteams wordt gewerkt en is binnen dit project niet van toepassing.
- Knelpunten. Bij deze strategie worden componenten die dringende knelpunten oplossen of zaken waar de gebruikers de meeste hinder van ondervinden als eerste ontwikkeld. In dit project ligt de nadruk echter niet op het wegnemen van (dringende) knelpunten, maar op het aanbrengen van structuur en het vastleggen van informatie.

In hoofdstuk 6 van de Definitiestudie (bijlage D) is een overzicht te vinden waarin de pilots worden opgedeeld in pilotdelen.

4

Pilotontwikkeling

Nadat in de fase Definitiestudie een globaal beeld van het te ontwikkelen systeem is gevormd en de interactie van de gebruiker met het systeem duidelijk is gemaakt, is in de fase Pilotontwikkeling gedetailleerde invulling aan de pilots gegeven door het opstellen van een Application Style Guide en het vervaardigen van navigatiediagrammen en schermprototypes. In deze fase zijn de pilots ook gebouwd in de Visual Studio .NET ontwikkelomgeving.



Figuur 15: De fase Pilotontwikkeling

De ontwikkeling van de pilots is iteratief doorlopen. De vier pilots zijn, bij iedere iteratie, gespecificeerd in pilotontwikkelplannen en vervolgens gebouwd. Daarbij is de eerste pilot volgens een testmethode beoordeeld. De resultaten van deze fase, in de vorm van pilotontwikkelplannen, zijn opgenomen in bijlagen E t/m H.

4.1 Opstellen Application Style Guide

De fase Pilotontwikkeling is, in het pilotontwikkelplan van de eerste pilot, gestart met het opstellen van een Application Style Guide aan de hand van de GUIDE [2] methodiek. Een Application Style Guide is een document waarin standaarden worden neergezet (die voor een deel of voor de gehele applicatie gelden) met betrekking tot zaken als kleur- en icoongebruik, vensters, taalgebruik, knoppen en andere interface aangelegenheden. Het voordeel hiervan is dat er bij het ontwerpen van de gebruikersinterface kan worden teruggevallen op een set standaarden om zo de consistentie binnen het systeem te waarborgen. Ook kan een dergelijk document nuttig zijn in het geval dat er door anderen aanpassingen of toevoegingen moeten worden gemaakt.

De volgende paragrafen beschrijven de wijze waarop de Application Style Guide tot stand is gekomen en gaat in op enkele van de opgestelde richtlijnen.

4.1.1 Bepalen Style Guide richtlijnen en invloeden

Omdat het doelplatform van het nieuwe systeem een Microsoft Windows besturingssysteem is, is gekeken naar de aanwezigheid van een 'Environment Style Guide': een Style Guide die is opgesteld voor de omgeving waarin een systeem wordt geplaatst. Het bleek dat deze door Microsoft is opgenomen in het boek 'Official Guidelines for User Interface Developers and Designers' [8]. Hierin worden standaarden uiteengezet voor ontwikkelaars die gebruik maken van het Windows besturingssysteem. Veel van deze standaarden zijn aangehouden bij het opstellen van de Application Style.

Daarnaast is gezocht naar de aanwezigheid van een 'Corporate Style Guide': een Style Guide die standaarden vastlegt die gelden binnen de organisatie. Deze bleken aanwezig te zijn, maar alleen te gelden voor externe correspondentie toe. Besloten is om hoofdzakelijk de Microsoft richtlijnen aan te houden en waar nodig nieuwe richtlijnen toe te voegen. Men is binnen de organisatie bekend met Microsoft applicaties en kan dus snel vertrouwd raken met het nieuwe systeem wanneer delen van de interface hierop zijn gebaseerd.

4.1.2 Bepalen iconen en grafische elementen

Na het bepalen van de richtlijnen is gezocht naar geschikte iconen voor het systeem. Na een zoektocht rond verschillende websites is de keuze gevallen op de vrij verkrijgbare iconpakketten uit de 'iCandy Junior' serie [9]. Deze pakketten bevatten een groot aantal bruikbare iconen en zijn ontworpen in de Windows XP stijl, waaronder iconen die zijn gebruikt voor het representeren van incidenten, problemen, Incidentbeheer en Probleembeheer. Figuur 16 geeft deze weer:



Figuur 16: Iconen voor Incidenten, Problemen, Incidentbeheer en Probleembeheer

Naast de iconen die delen van het systeem of objecten representeren zijn ook iconen uitgezocht die gebruikt worden om acties aan te duiden. Deze worden (verkleind) over een ander icoon geplaatst om aan te geven welke acties er op een object mogelijk zijn. Deze iconen zijn in figuur 17 weergegeven:



Figuur 17: Iconen voor toevoegen, wijzigen en verwijderen van objecten

Verder zijn iconen uit het Windows XP besturingssysteem gebruikt. Denk hierbij aan de bekende Windows Explorer mappen

De iconen zijn met het softwarepakket MicroAngelo omgezet naar het Portable Graphics Network (PNG) bestandsformaat, waarbij de transparantielaag bewaard blijft. Deze zijn later geïmporteerd in de Visual Studio .NET ontwikkelomgeving. Waar nodig zijn de PNG bestanden aangepast of aangevuld in Adobe Photoshop.

Voor de eerste, tweede en vierde pilot is het gebruik van grafische elementen beperkt gebleven tot iconen. Deze pilots zijn bestemd voor gebruik door de helpdeskmedewerkers; er is daarom meer de nadruk gelegd op de functionele aspecten. Helpdeskmedewerkers kunnen meer tijd investeren in het zich eigen maken van het systeem en hebben minder houvast nodig dan de gewone medewerkers.

Binnen de derde pilot, uitsluitend bedoeld voor de gewone medewerkers, wordt op een beperkte wijze gebruik gemaakt van grafische elementen. Dit komt naar voren in een grafische representatie van de plek waar de gebruiker zich bevindt, waarvan een voorbeeld is weergegeven in figuur 18:



Figuur 18: Grafische aanduiding van locatie in het systeem

Besloten is dat de gewone gebruikers met een minimale instructie met dat deel van het systeem overweg moeten kunnen, en gemakkelijk moeten kunnen zien waar zij zich bevinden. De kleurstelling en het gebruik van (een deel van) het bedrijfslogo is overgenomen van de Intranet site van de organisatie.

4.1.3 Vaststellen standaarden

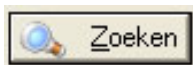
Op basis van de gestelde richtlijnen, invloeden en een inventarisatie van de beschikbare controls in Visual Studio .NET zijn de standaarden vastgelegd. Om een indruk te krijgen van deze standaarden zal in de komende paragrafen een kleine selectie uit de verzameling van standaarden worden behandeld.

Knoppen

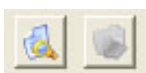
Knoppen kunnen in twee categorieën worden ingedeeld: iconknoppen en tekstknoppen. Iconknoppen moeten worden voorzien van een 16x16 pixel icon, met verduidelijkende tooltips.



Tekstknoppen kunnen eveneens (wanneer de ruimte op het venster dit toelaat, en een icoon beschikbaar is) worden voorzien van een 16x16 pixel icoon. Hierbij is het niet strikt noodzakelijk een tooltip op te nemen. Bij voorkeur is de knop door middel van een hotkey (sneltoets) te activeren.

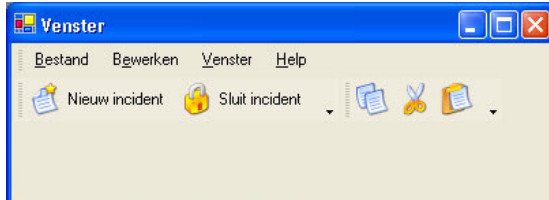


Als een knop niet beschikbaar is, dan wordt deze grijs weergegeven.



Toolbars

Toolbars (knoppenbalken) moeten worden voorzien van iconen om veel gebruikte functies snel uit te kunnen voeren. Deze functies moeten tevens via het menu beschikbaar zijn. Wanneer de naam van de functie niet naast het icoon wordt weergegeven moet het icoon voorzien worden van een tooltip. Wordt deze wel weergegeven, dan wordt deze rechts van het icoon geplaatst. Iconen in toolbars moeten worden weergegeven in een grootte van 20x20 pixels.



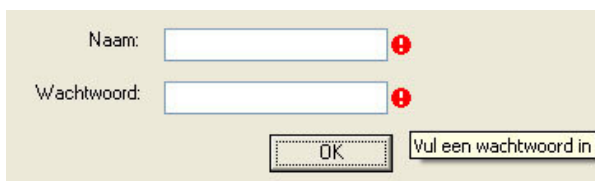
Toolbars kunnen door de gebruiker van de standaardplaats worden gehaald en op een nieuwe plaats worden gezet.



Samenhangende functies, zoals functies uit het 'Bewerken' menu moeten worden gegroepeerd in aparte toolbars.

Invoervalidatie

Invoervalidatie moet worden ondersteund door middel van de Error Provider. Deze plaatst een icoon naast ieder veld dat niet aan de voorwaarden voldoet. Eventueel kan ingesteld worden dat het icoon een aantal malen moet knipperen om de aandacht van de gebruiker te trekken.



In de pilotontwikkelplannen van de vier pilots (te vinden in bijlagen E t/m H) zijn de volledige Style Guide definities terug te vinden.

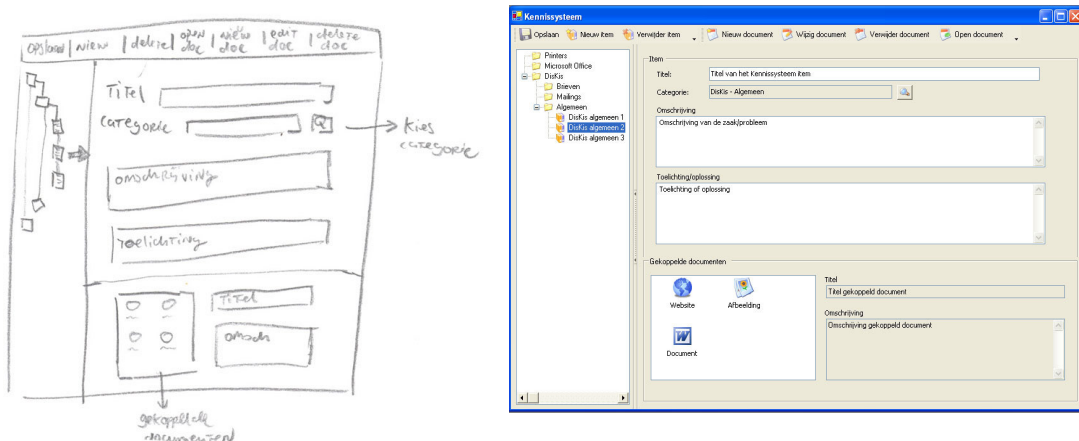
4.2 Vervaardigen Prototypes gebruikersinterface en vaststellen functionele structuur

Bij het opstellen van de pilotontwikkelplannen voor alle pilots is voor ieder scherm of venster een prototype vervaardigd. Een prototype is een weergave van een (deel van) een scherm, waarmee de daadwerkelijke interactie met het systeem plaatsvindt. Deze zijn aangevuld, op basis van aanwijzingen uit GUIDE [2], met een omschrijving van het venster, geldende regels en beperkingen, de classificatie van het type venster, eventuele menu's en toolbars en het maximum aantal instanties dat de gebruiker geopend kan hebben. Door middel van navigatieschema's (zie paragraaf 5.3) is aangegeven hoe de gebruiker navigeert tussen de verschillende schermen waaruit de pilot bestaat.

De schermprototypes zijn tot stand gekomen uit de volgende bronnen:

- De Use-cases, waarmee een beeld is gevormd van de acties die de gebruiker uit moet kunnen voeren.
- De Style Guide definities en Microsoft richtlijnen, met betrekking tot de wijze van implementatie van controls en eventuele grafische elementen.
- Vergelijkbare softwarepakketten zoals TopDesk, Track-IT en SMILE Complaint System.
- Het datamodel en de bijbehorende associaties.
- De beschikbare controls binnen de Visual Studio .NET ontwikkelomgeving en externe, vrij verkrijgbare controls.

Het prototyping proces is gestart met het maken van ruwe schetsen om de globale layout en controls van de schermen vast te stellen. Deze schetsen zijn omgezet naar prototypes. In figuur 19 is de typische evolutie van een schermprototype te zien:



Figuur 19: Evolutie schermprototype (Kennissysteem onderdeel afgebeeld)

Er is gekozen voor het uitwerken van de prototypes in de Visual Studio .NET ontwikkelomgeving. Het voordeel hiervan is dat er een nagenoeg 1 op 1 (visuele) representatie van het uiteindelijke systeem kan worden gemaakt. Als alternatief is er gekeken naar het gebruik van Microsoft Visio. Deze bleek, in elk geval op de machine die beschikbaar was binnen de organisatie, niet te voldoen. Bij het werken met een groot aantal controls werkte het programma te langzaam. Daarnaast zijn niet alle controls uit de Visual Studio .NET omgeving in Visio aanwezig en bleek het importeren van icoonbestanden te omslachtig.

Bij de ontwikkeling van elke pilot zijn de betrokken schermprototypes in de ontwikkelomgeving geplaatst en zijn schermafdrucken gebruikt om in de pilotontwikkelplannen een indruk te geven van de uiteindelijke werking van de pilots. Voor aanvang van de bouw van de pilots zijn de pilotontwikkelplannen goedgekeurd door de bedrijfsmentor. In een aantal gevallen zijn kleine aanpassingen in de pilotontwikkelplannen verwerkt.

Het valt buiten het bereik van dit verslag om iedere beslissing achter ieder venster te bespreken. In de nu volgende subparagrafen zal daarom een selectie worden besproken van de belangrijkste beslissingen met betrekking tot de totstandkoming van de schermprototypes. De volledige verzameling van vervaardigde schermprototypes is te vinden in de pilotontwikkelplannen.

4.2.1 Pilot I - Incidentbeheer

Zoals in het pilotplan (paragraaf 3.1.1) is uiteengezet, richt de eerste pilot zich op het onderdeel Incidentbeheer. Dit onderdeel kan gezien worden als het centrale onderdeel binnen het systeem en zal door de helpdeskmedewerkers tijdens de dagelijkse werkzaamheden het meest worden gebruikt. Ook is, omdat de eerste pilot de start markeerde van de fase Pilotontwikkeling, hier de algemene opzet van het systeem neergezet. Deze zal in de volgende subparagraaf worden besproken.

Algemene opzet

Als algemene opzet voor het systeem is gekozen voor een Multiple Document Interface (MDI) applicatie, waarbinnen de gebruiker de handelingen uitvoert. Een kenmerk van een MDI interface is dat een venster dient als 'container' voor andere vensters. Hierdoor wordt het mogelijk met meerdere vensters tegelijk te werken, vergelijkbaar met bijvoorbeeld fotobewerkingspakketten waarbij de gebruiker meerdere afbeeldingen tegelijk open kan hebben staan. Dit in tegenstelling tot een Single Document Interface (SDI), waarin de gebruiker slechts één venster of document kan gebruiken. Een dergelijke interface is gekozen met de relevante gebruikersgroepen in gedachte.

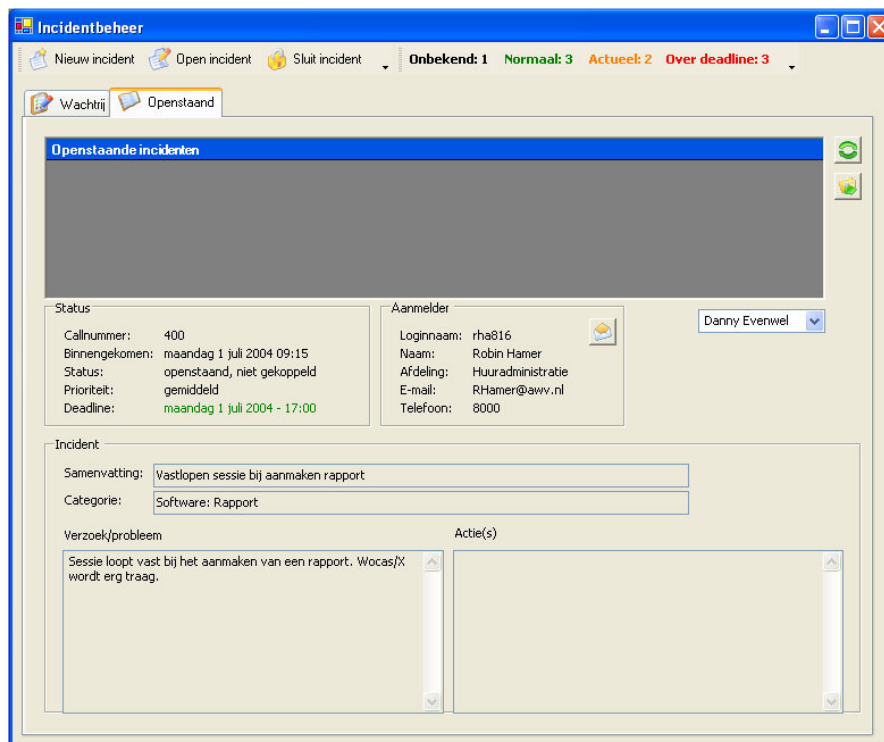
De mogelijkheid wordt de helpdeskmedewerkers geboden om te kunnen multi-tasken. Zo kunnen zij bezig zijn met toevoegen van een Kennissysteem item, een incident uit de wachtrij oppakken en behandelen en vervolgens verder gaan met een Kennissysteem item.

Besloten is om de hoofdonderdelen (Incidentbeheer, Probleembeheer, Historie en Kennissysteem) van het systeem onder te brengen in een verticale menubalk aan de linkerkzijde van de interface, en de beheer- en rapportagefuncties te plaatsen in een horizontale toolbar. Op deze manier heeft de gebruiker altijd toegang tot de systeemfuncties.

4.2.2 Wachtrij en openstaande incidenten

Als uitgangspunt voor het onderdeel Incidentbeheer is het concept overgenomen dat gebruikt wordt bij een aantal vergelijkbare pakketten. De gemaakte vergelijking is terug te vinden in paragraaf 1.4 van het pilotontwikkelplan van de eerste pilot (bijlage E).

De helpdeskmedewerker werkt vanuit een overzicht met incidenten (of 'calls') waar in één oogopslag te zien is welke incidenten openstaan en wat de deadline en prioriteit is. Vanuit het overzicht kan het incident worden geopend en kunnen de gekoppelde gegevens worden aangepast of ingevuld. Hiertoe is het schermprototype ontwikkeld dat in figuur 20 te zien is:



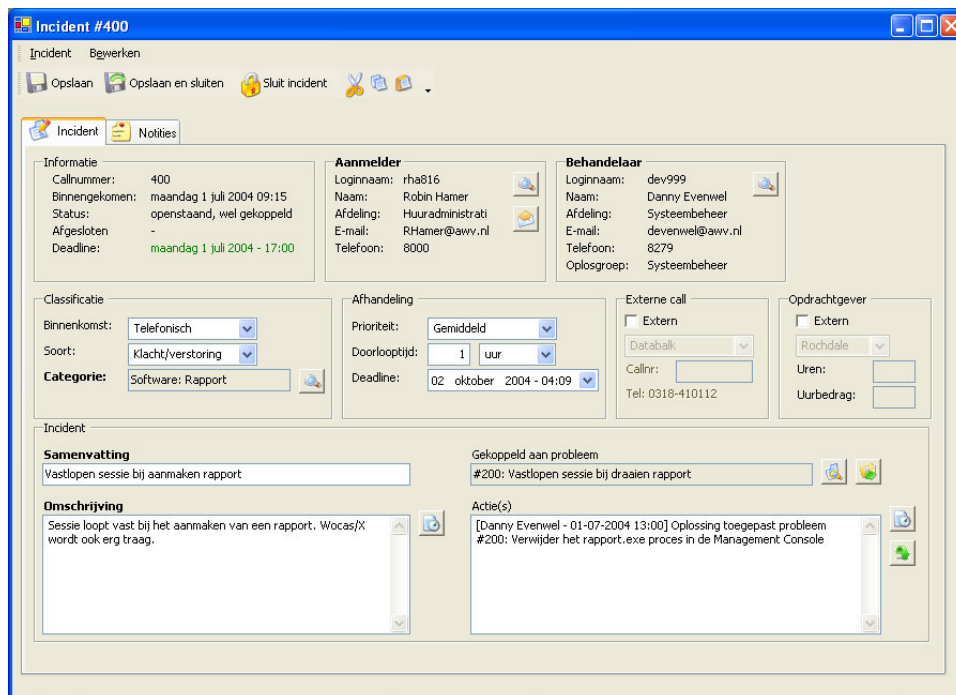
The screenshot shows a software window titled 'Incidentbeheer'. At the top, there are buttons for 'Nieuw incident', 'Open incident', and 'Sluit incident'. Below these are status indicators: 'Onbekend: 1', 'Normaal: 3', 'Actueel: 2', and 'Over deadline: 3'. The main area has two tabs: 'Wachtrij' and 'Openstaand'. The 'Openstaand' tab is active, showing a list of open incidents. Below the list, there are fields for 'Status' (Callnummer: 400, Binnengekomen: maandag 1 juli 2004 09:15, Status: openstaand, niet gekoppeld, Prioriteit: gemiddeld, Deadline: maandag 1 juli 2004 - 17:00) and 'Aanmelder' (Loginnaam: rha816, Naam: Robin Hamer, Afdeling: Huuradministratie, E-mail: RHamer@awv.nl, Telefoon: 8000). A dropdown menu shows 'Danny Evenwel'. Below this, there are fields for 'Incident' (Samenvatting: Vastlopen sessie bij aanmaken rapport, Categorie: Software: Rapport) and 'Verzoek/probleem' (Sessie loopt vast bij het aanmaken van een rapport. Wocas/X wordt erg traag). To the right of the 'Verzoek/probleem' field is an 'Actie(s)' field.

Figuur 20: Prototype venster openstaande incidenten (Incidentbeheer)

Omdat er tevens een soortgelijk overzicht moet worden opgenomen voor de incidenten die in de wachtrij staan, is besloten het venster te splitsen door middel van tabbladen. De helpdeskmedewerker kan op deze manier snel schakelen tussen de incidenten in de wachtrij en de aan hem of haar toegewezen incidenten.

Incidentgegevens

Ook voor het weergeven van de gegevens die horen bij een specifiek incident is gekeken naar de bestaande softwarepakketten. Bij deze pakketten wordt een call of klacht weergegeven in een enkel venster, en waar nodig opgesplitst in verschillende tabbladen. Er is geen reden om hier vanaf te wijken en dus is voor deze aanpak gekozen. De helpdeskmedewerker heeft in deze opzet in veel gevallen slechts één venster - met uitzondering van de popup-schermen voor het selecteren van een categorie, behandelaar en aanmelder - nodig om een incident af te handelen. Figuur 21 toont het ontwikkelde schermprototype voor het Incidentgegevens-scherm:



Figuur 21: Prototype venster incidentgegevens (Incidentbeheer)

Het venster is opgesplitst in twee tabbladen: een tabblad voor de incidentgegevens en een tabblad voor aanverwante aantekeningen. De mogelijkheid bestaat dat er grote stukken tekst bijgehouden moeten worden. Het bleek onmogelijk een tekstveld op te nemen dat groot genoeg was om werkbaar te zijn binnen de schermresolutie van 1024 bij 768 pixels.

Afschermen van gegevens en beperkingen

Omdat er binnen het beoogde systeem gewerkt wordt met oplosgroepen die elk hun eigen (vak)gebied beslaan, is er nagedacht over het afschermen van gegevens. Door het systeem open te stellen en alle medewerkers toegang te geven tot alle in het systeem aanwezige gegevens kan een aantal ongewenste situaties optreden. Zo is het volgende scenario denkbaar:

Een incident wordt door een gebruiker via het systeem aangemeld, maar de verkeerde categorie wordt gekozen. Hierdoor komt het incident terecht in de wachtrij van de verkeerde oplosgroep. Een medewerker van de ontvangende oplosgroep kijkt in de wachtrij en ziet dat het incident niet voor zijn of haar oplosgroep bestemd is, maar wijst deze toch aan zichzelf

toe. Omdat het systeem open staat, kan de medewerker in de historie van problemen kijken en het incident aan een probleem koppelen dat door een andere oplosgroep is aangemaakt. Hierdoor wordt werk ‘afgepakt’ dat eigenlijk voor een andere oplosgroep is bedoeld.

Daarnaast kunnen medewerkers zich niet prettig voelen bij het idee dat medewerkers buiten de directe groep of afdeling toegang hebben tot gevoelige gegevens of aantekeningen. Om situaties als deze te voorkomen, is besloten dat alleen gegevens van incidenten en problemen worden getoond die aan de oplosgroep van de ingelogde helpdeskmedewerker zijn gekoppeld.

Verder is bepaald dat een incident alleen mag worden doorgestuurd naar de wachtrij van een andere oplosgroep, en niet direct naar een andere helpdeskmedewerker in een andere oplosgroep. Hierdoor wordt voorkomen dat helpdeskmedewerkers worden ‘omzeild’.

De uitzondering wordt gevormd door het Kennissysteem; de gegevens van het Kennissysteem zijn voor alle helpdeskmedewerkers te benaderen.

4.2.3 Pilot 2 - Probleembeheer

Voor de tweede pilot, welke zich richt op het onderdeel Probleembeheer, is grotendeels het ontwerp en de werking van de eerste pilot aangehouden. Problemen kunnen in een wachtrij worden gezet en het onderdeel bevat net als bij Incidentbeheer een overzicht van alle openstaande problemen. De belangrijkste verschillen schuilen in de informatie die moet worden bijgehouden van een probleem. Zo kan de helpdeskmedewerker bijvoorbeeld de oorzaak van een probleem bijhouden, waardoor het probleem binnen het systeem wordt gekenmerkt als ‘Known Error’ (Bekend Probleem). Daarnaast is er de mogelijkheid meerdere incidenten aan het probleem te koppelen.

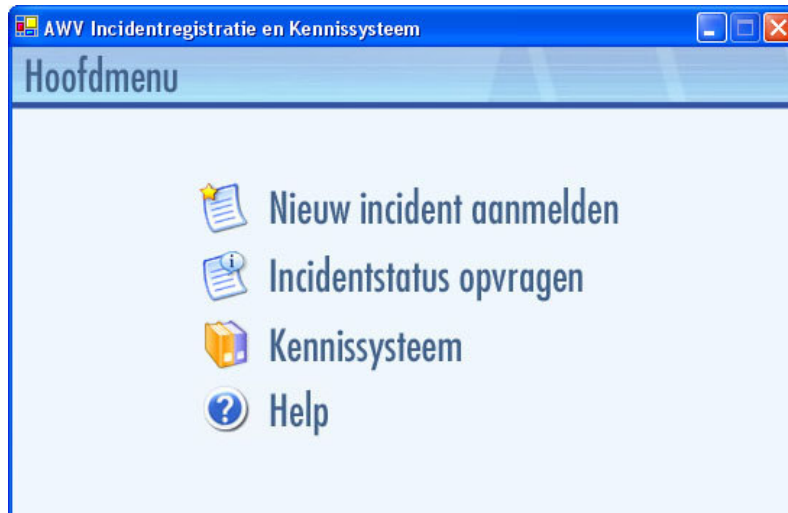
De schermprototypes van deze pilot zijn opgenomen in bijlage F.

4.2.4 Pilot 3 – Incidentinvoer- en status, kennissysteem en historie

De derde pilot omvat onder andere de functionaliteiten die gericht zijn op de gebruikersgroep van ‘gewone gebruikers’: het invoeren van een nieuw incident, de status opvragen van een reeds ingevoerd incident en het opvragen van informatie uit het Kennissysteem.

Met de gebruikersgroep in gedachten is besloten dit deel van het systeem in te delen aan de hand van een SDI (zie paragraaf 5.2.1) indeling. Er is hier dus bewust gekozen voor een meer stapsgewijze of lineaire benadering. De gebruiker voert, na een keuze te hebben gemaakt in het hoofdmenu, één taak uit en kan pas aan een andere taak beginnen wanneer de huidige taak afgerond of afgebroken wordt. Dit is terug te vinden in het navigatieschema van het pilotontwikkelplan van de derde pilot (bijlage G). Een gewone gebruiker zal doorgaans (dit deel van) het systeem gebruiken met één doel voor ogen, deze tot uitvoer brengen, het systeem afsluiten en verder gaan met zijn of haar reguliere werkzaamheden. Daarnaast geeft een SDI indeling minder ‘belasting’ op de gebruiker omdat de gebruiker zich kan concentreren op één enkele taak.

Figuur 21 toont het hoofdmenu van het betreffende deel van de pilot, waarin onder meer te zien is hoe gebruik is gemaakt van tekst en iconen om duidelijk te maken welke keuzen er beschikbaar zijn:



Figuur 21: Prototype hoofdmenu gedeelte systeem voor gewone gebruikers

4.2.5 Pilot 4 – Beheer, rapportage en voortgangscontrole

De vierde en laatste pilot bevat hoofdzakelijk functionaliteiten die de kernfunctionaliteiten (Incident- en Probleembeheer) ondersteunen door middel van het bieden van beheermogelijkheden, voortgangscontrole en rapportage. De meeste van deze functionaliteiten zijn betrekkelijk simpel en omvatten geen in het oog springende schermprototypes. Wel interessant is de wijze waarop de notificatie en voortgangscontrole tot stand is gekomen.

Basissysteemeis A4 stelt het volgende:

Het moet mogelijk zijn een afhandelduur aan een incident toe te kennen. Binnen deze tijd moet het incident normaal gesproken opgelost kunnen worden. Het moet voor de helpdeskmedewerker duidelijk worden wanneer de deadline van een incident in gevaar komt of al is overschreden.

Het laatste deel van de systeemeis stelt dat het voor helpdeskmedewerkers duidelijk moet zijn wanneer de deadline van een incident in gevaar komt of al is overschreden. Op basis van deze eis is besloten notificatiemogelijkheden als pilotdeel op te nemen in het ontwikkelingsstraject.

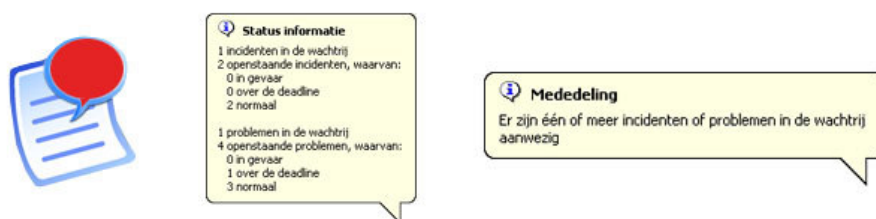
De eerste gedachte bij dit pilotdeel (ten tijde van het opstellen van de Definitiestudie) ging uit naar het implementeren van notificatie via e-mail. Het systeem zou periodiek een controle op de database uitvoeren, bepalen welke incidenten in gevaar of over de deadline zijn en een e-mail sturen naar de desbetreffende helpdeskmedewerker. Eenmaal aangekomen bij deze pilot in de fase Pilotontwikkeling is bepaald, aan de hand van gevonden voorbeelden in het boek 'Beginning Visual Studio .NET' [10], dat notificatie plaats moet vinden via een 'system-tray' icoon. Een dergelijk icoon blijft gedurende het werken met het systeem zichtbaar in de taakbalk en kan informatie verstrekken door het gebruikte icoon en zogenaamde 'balloons': tekstballonnen waar korte berichten in kunnen verschijnen.

De beslissing om voor deze oplossing te kiezen is genomen op basis van de volgende punten:

- E-mail berichten met betrekking tot het systeem komen in de inbox van medewerkers terecht, samen met 'echte' e-mail berichten. Hierdoor moet de medewerker constant een scheiding maken tussen verschillende soorten berichten. Dit is enigszins op te vangen door het instellen van een e-mail 'rule' waarin e-mail berichten van een bepaald karakter automatisch worden verplaatst naar een andere map. Wel zal de mailbox vollopen met berichten afkomstig van het systeem die periodiek verwijderd moeten worden. Dit zal zeker hinderlijk zijn bij een groot aantal openstaande incidenten.
- Het e-mail programma hoeft niet geopend te worden om erachter te komen waar de aandacht naar uit moet gaan.
- De gekozen oplossing is subtieler en minder opdringerig naar de gebruiker toe. Ook kan, door gebruik te maken van verschillende soorten iconen, communicatie naar de gebruiker toe plaatsvinden.
- De statusverandering hoeft niet permanent opgeslagen te worden, de interesse gaat alleen uit naar de actuele (de op dat moment geldende) status.

Naast de controle op incidenten die in gevaar komen of over de deadline zijn, is tevens besloten de genoemde oplossing te gebruiken om de helpdeskmedewerker op de hoogte te brengen van binnenkomende incidenten die in de wachtrij aanwezig zijn.

Er wordt onderscheid gemaakt tussen drie manieren waarop notificatie naar de helpdeskmedewerker plaatsvindt: door middel van kleurcodes in het system-tray icoon waarmee de actuele status is af te lezen, een uitgebreid statusoverzicht bij het klikken op het icoon en een korte melding bij het optreden van een verandering van de status. Figuur 22 geeft aan op welke wijze de notificatiemogelijkheden grafisch zijn geïmplementeerd:



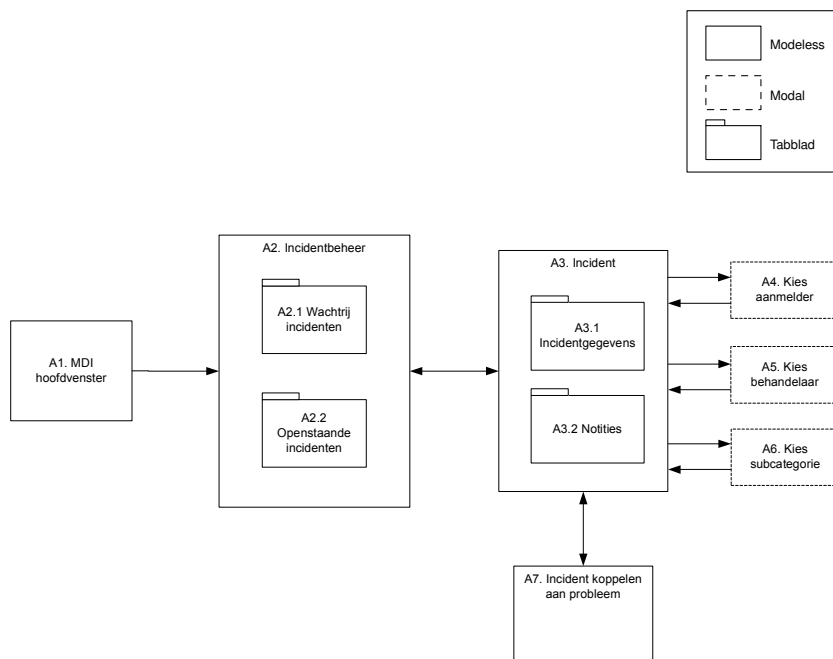
Figuur 22: Grafische implementatie notificaties voor de system-tray

Een complete uitleg van het notificatiesysteem is terug te vinden in het pilotontwikkelplan van de vierde pilot, opgenomen als bijlage H.

4.3 Vervaardigen navigatiediagrammen

Samen met het vervaardigen van de voorgenoemde prototypes, zijn voor elke pilot navigatiediagrammen [2] vervaardigd. Door middel van dergelijke diagrammen kan uiteengezet worden hoe de gebruiker door de verschillende schermen kan navigeren. Hierbij wordt een onderscheid gemaakt tussen ‘modeless’ vensters (aangegeven door middel van een vast kader) en ‘modal’ vensters (aangegeven door middel van een gestreept kader). Bij een modeless venster kan de gebruiker wisselen tussen het betreffende venster en andere vensters, en bij een modal venster moet de gebruiker het betreffende venster afsluiten voordat er met andere vensters gewerkt kan worden.

Figuur 23 geeft een voorbeeld van één van de vervaardigde navigatiediagrammen:



Figuur 23: Navigatiediagram Incidentbeheer (weergegeven: pilot I)

4.4 Pilotontwikkelplan

Voor iedere pilot is gebruik gemaakt van de ‘time-boxing’ [1] projectmanagementtechniek. In het pilotontwikkelplan wordt aangegeven uit welke onderdelen een pilot bestaat en in welke volgorde deze worden ontwikkeld. Hiervoor zijn de pilots op basis van de in het pilotplan (Definitiestudie) gedefinieerde pilotdelen verder onderverdeeld in bouweenheden.

Bij time-boxing wordt de opleverdatum van een pilot vastgesteld en worden de bouweenheden ingedeeld in drie categorieën:

- Basis. Eenheden binnen deze zone moeten binnen de begrenzings van de time-box worden gerealiseerd.
- Comfort. Een aantal eenheden in deze zone zal waarschijnlijk wel en een aantal niet worden ontwikkeld. Het is niet strikt verplicht eenheden in deze zone mee te nemen.
- Luxe. Elke eenheid die in deze zone ontwikkeld wordt is mooi meegenomen. Normaliter wordt niet verwacht dat deze zone binnen de time-box aan de beurt komt.

Wanneer basiseenheden binnen de eerste categorie nog niet zijn gerealiseerd bij het verstrijken van de gestelde datum moeten deze worden meegenomen naar de volgende time-box. Overzicht 1 geeft een voorbeeld van een time-box, in dit geval een gedeelte van de time-box van de eerste pilot:

Time-box pilot I				
Van: 26-07-2004 tot: 16-08-2004				
Code	Omschrijving	Zwaarte	Geschat (uren)	Werkelijk (uren)
A1	Basis MDI interface delen	B	5	5,5
A2	Prepareren en importeren iconen	B	3,5	4
A3	Gebruikersgegevens uit ADS	B	5	4,5
A4	SQL Server 2000 database op tijdelijke server en databasekoppeling	B	2,5	4
A5	Wachtrij voor incidenten	B	5	6
A6	Toewijzen incident aan behandelaar	B	1	1,5
A7	Openstaande incidenten	B	6	8
A8	Selecteren filter (gebruiker) voor openstaande incidenten	B	0,5	2
A9	Nieuw incident algemene velden en notities	B	6	8,5
A10	Gegevensvalidatie d.m.v. Errorprovider	B	2	3
A11	Kies categorie (popup)	B	1,5	1,5
A12	Kies aanmelder (popup)	B	2	3,5
A13	Kies behandelaar (popup)	B	2	1,5
...	...	...	...	...

Overzicht I: Een deel van de time-box van de eerste pilot

Zoals te zien is, is voor de eerste pilot een aantal kernfunctionaliteiten meegenomen (basis MDI interface, gebruikersinformatie (zoals de volledige naam, afdeling en telefoonnummer) uit de Active Directory Service). Bij het bepalen van de volgorde waarin de bouweenheden wordt gerealiseerd, is dus rekening gehouden met de herbruikbaarheid van onderdelen. Hetzelfde geldt voor de popup-vensters, voor het kiezen van een categorie, aanmelder en behandelaar.

4.5 Bouwen pilots

Na het opstellen van elk van de vier pilotontwikkelplannen is iedere pilot in Visual Studio .NET gebouwd, aan de hand van de vastgestelde bouweenheden.

In de nu volgende subparagrafen worden (in grote lijnen) de belangrijkste activiteiten die hiertoe zijn uitgevoerd besproken.

4.5.1 Importeren grafische elementen en iconen

Alle grafische elementen en iconen zijn geïmporteerd in de ontwikkelomgeving en waar nodig omgezet naar het juiste bestandsformaat.

4.5.2 Omzetten prototypes naar definitieve schermen

Op basis van de vervaardigde prototypes zijn de definitieve vensters vervaardigd. De vensters en controls zijn voorzien van een zinnige naamgeving en klaargemaakt voor de programmacode. Hierbij is ook rekening gehouden met de verschillende mogelijke schermresoluties. Bij vensters die vergroot en gemaximaliseerd kunnen worden - binnen de MDI interface - is door middel van 'anchoring' en 'docking' vastgesteld hoe de controls moeten worden geschaald.

4.5.3 Aanmaken classes

Op basis van de gedefinieerde klassen (zie het klassendiagram in paragraaf 3.8.1) zijn de benodigde classes aangemaakt. Het concept van een class wordt binnen 'Professional Visual Basic .NET' [11] samengevat als een abstractie van een 'real-world' concept, in dit geval bijvoorbeeld een gebruiker of een incident. Alle objecten die worden aangemaakt zijn gebaseerd op een class. Een class bestaat doorgaans uit drie hoofdbestanddelen:

- Eigenschappen (properties) die gegevens over een object bevatten, zoals een inlognaam van een gebruiker of een omschrijving van een incident.
- Gebeurtenissen ('events') die kunnen worden 'afgevuurd' wanneer dit nodig is. Zo is vastgelegd dat een specifiek event moet worden afgevuurd wanneer de status van een incident verandert, om zo bepaalde delen van de interface te beïnvloeden. Door in de programmacode hierop in te haken ontstaat een werkend systeem.
- Functies of methodes die instructies bevatten die moeten worden uitgevoerd op het object.

Door de classes binnen de ontwikkelomgeving te vervaardigen is een basis gelegd voor de verdere ontwikkeling van het systeem; de definities met betrekking tot de gegevensstructuur zijn op deze wijze vastgelegd.

4.5.4 Implementeren systeemlogica en werking

Met behulp van de reeds vervaardigde classes is de logica en werking van het systeem ontwikkeld in de Visual Basic programmeertaal en gekoppeld aan de controls op de vensters. Hierbij is gebruik gemaakt van externe Dynamic Link Libraries (DLL's), zoals voor het implementeren van menubalken en toolbars in de Office XP stijl.

Tevens zijn systeemmeldingen, zoals waarschuwingen en mededelingen naar de gebruiker toe geïmplementeerd. Daarnaast zijn beveiligings- en autorisatiemaatregelen doorgevoerd. Er wordt bijvoorbeeld bij het opstarten van het systeem gecontroleerd of de gebruiker die binnen de Windows omgeving ingelogd is, is ingedeeld in een oplosgroep. Is dit niet het geval, dan wordt het systeem afgesloten en krijgt de gebruiker de melding dat hij of zij contact op moet nemen met de systeembeheerder.

4.5.5 Vervaardigen Stored Procedures

De benodigde Stored Procedures voor het ophalen, toevoegen, wijzigen en verwijderen van gegevens in de database zijn met behulp van de SQL Enterprise Manager en Query Analyzer aangemaakt, samen met de programmacode voor het aanroepen hiervan. Het onderstaande voorbeeld toont de syntax van een dergelijke Stored Procedure:

```
CREATE PROCEDURE searchIncidentenHoofdcategorie
@hoofdcategorie INT,
@oplosgroep INT,
@zoekstringINT INT,
@zoekstringVARCHAR VARCHAR(255),
@zoekveld VARCHAR(255)
AS
DECLARE @query as VARCHAR(2000)

/* op INT zoeken */
IF (@zoekstringINT IS NOT NULL)
    SET @query = 'SELECT incident.ID, samenvatting, subcategorie.naam
AS subcategorienaam, oplosgroep.naam AS oplosgroepnaam,
datumtijdingevoerd, statusnaam =
CASE status
WHEN "O" THEN "Openstaand"
WHEN "W" THEN "In wachtrij"
WHEN "G" THEN "Gesloten"
END
FROM incident JOIN oplosgroep on incident.oplosgroep =
oplosgroep.ID
JOIN subcategorie ON incident.subcategorie = subcategorie.ID
JOIN hoofdcategorie ON subcategorie.hoofdcategorie = hoofdcatego-
rie.ID
WHERE hoofdcategorie.id = ' + CAST(@hoofdcategorie AS
VARCHAR(20)) + ' AND incident.oplosgroep = ' + CAST(@oplosgroep AS
VARCHAR(20)) + ' AND ' + @zoekveld + '=' + CAST(@zoekstringINT AS
VARCHAR(20)) + ' AND incident.status <> "W"'
```

Voorbeeld 1: Stored Procedure 'searchIncidentenHoofdcategorie' (gedeeltelijk weergegeven)

Zoals te zien is bevat de Stored Procedure een vijftal parameters (aangegeven met een '@' teken). Vanuit de programmacode worden parameters bij het aanroepen van de Stored Procedure meegegeven, zodat de Stored Procedure hiervan gebruik kan maken bij het uitvoeren van de opdracht. De uitvoer die de Stored Procedure teruggeeft wordt vervolgens in de programmacode in een 'DataSet' gestopt en uitgelezen. Voor het vullen van DataSets is een zogenaamde 'SQL Wrapper Class' gebruikt. Dergelijke classes zijn vrij verkrijgbaar en maken het werken met een SQL database aanzienlijk makkelijker doordat er gewerkt kan worden met relatief simpele functies en properties.

4.5.6 Implementeren applicatie-instellingen

Zoals in paragraaf 4.8.3 is uiteengezet, is besloten gebruik te maken van een XML bestand om de verschillende applicatie-instellingen op te slaan. Dit configuratiebestand is in de loop van de fase Pilotontwikkeling tot stand gekomen en wordt in de programmacode uitgelezen door middel van de .NET class 'appSettingsReader' uitgelezen. Voorbeeld 2 geeft het XML configuratiebestand weer:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="databaseNaam" value="AWVIRS"/>
    <add key="databaseServer" value="AWVDISTEST"/>
    <add key="LDAPAWV"
value="LDAP://OU=AWV,DC=awvnet,DC=local"/>
    <add key="LDAPExtern"
value="LDAP://awvads:389/OU=AWV,DC=awvnet,DC=local"/>
    <add key="statusCheckInterval" value="15000"/>
    <add key="inGevaarTijd" value="60"/>
    <add key="statusPopupBalloon" value="1"/>
    <add key="standaardUurtarief" value="30"/>
  </appSettings>
</configuration>
```

Voorbeeld 2: Applicatie-instellingen middels een XML bestand

Te zien is dat bijvoorbeeld de locatie en naam van de database worden vastgelegd en dat de tijd waarop incidenten en problemen als 'in gevaar' worden gekenmerkt, is vastgezet op 60 minuten voor het verstrijken van de deadline.

4.5.7 Implementeren periodieke statuscontrole

Periodieke statuscontrole (waarvan de interval is in te stellen via het voorgenoemde configuratiebestand) is in de bouwfase van de vierde pilot geïmplementeerd. Hierbij is eveneens gebruik gemaakt van een externe DLL, voor het weergeven van de pop-up ballon in de statusbalk. Gekozen is om de intervaltijd op 15 seconden te zetten. Dit wordt gezien als een goede middenweg tussen twee punten: bij een te lange interval is de status niet actueel genoeg en bij een te korte interval kan er sprake zijn van een onnodige en ongewenste belasting op de databaseserver.

4.5.8 Implementeren rapportagemogelijkheden

Voor de vierde pilot is gekeken naar manieren waarop rapporten kunnen worden gegenereerd op basis van de in de database aanwezige informatie. In eerste instantie was het plan om dit middels Microsoft Word te bewerkstelligen. Word zou worden geopend vanuit Visual Basic, een nieuw Word document (met daarin voorgedefinieerde velden) zou worden aangemaakt en deze zou worden gevuld met gegevens uit de database. Een eerste test wees uit dat dit inderdaad mogelijk, maar wel omslachtig in het gebruik was.

Gelukkig bleek na een korte zoektocht door de documentatie het (commerciële) pakket Crystal Reports van de firma Business Objects geïntegreerd te zijn in de gebruikte versie van Visual Studio .NET. Crystal Reports stelt de ontwikkelaar in staat rapporten aan te maken met behulp van een grafische interface. Deze rapporten kunnen door middel van een ‘viewer’ component door de gebruiker gegenereerd worden waarna ze bekeken, omgezet (naar een ander bestandsformaat) of uitgeprint kunnen worden.

Een drietal rapporten, zoals deze zijn beschreven in het pilotontwikkelplan van de vierde pilot (bijlage H), is volgens deze methode ontwikkeld en geïntegreerd in het systeem. Nieuwe rapporten kunnen in de toekomst door de medewerkers van de afdeling ICT worden toegevoegd waarbij gebruik gemaakt kan worden van de beschrijvingen van het datamodel dat aanwezig is in de Definitiestudie.

4.6 Uitvoeren gebruikerstests

Na het specificeren en bouwen van de eerste pilot is over de gehele pilot een gebruikerstest gehouden. Het doel van deze test was het testen van de functionaliteit (de correcte werking van de pilot), het evalueren van de interface en het blootleggen van specifieke probleem- of verbeterpunten. Dit proces is enigszins te vergelijken met ‘GUI prototyping’ uit GUIDE [2] waarbij feedback op een ontwikkeld prototype of gebouwd onderdeel gebruikt wordt om verbeteringen of aanpassingen aan het prototype aan te brengen.

De volgende subparagrafen behandelen het kiezen van een geschikte testmethode, het opstellen van de testscenario's en het interpreteren van de testresultaten.

4.6.1 Kiezen testmethode

Voor het testen van de eerste pilot is gekeken naar verschillende testmethoden. Hiervoor is het boek ‘Task-Centered User Interface Design’ [4] gebruikt. Hierin worden de testmethoden in twee globale gebieden verdeeld: het testen zonder gebruikers en het testen met gebruikers. Het eerste gebied bevat de volgende testmethoden:

- **Cognitive Walkthrough.** Bij deze methode verplaatst de ontwikkelaar of tester zich in de gedachten van de gebruikers en probeert bij het testen een geloofwaardig verhaal bij iedere stap te bedenken.
- **Action Analysis.** Deze methode is onder te verdelen in twee variaties: ‘formal’ Action Analysis en ‘back of the envelope’ Action Analysis, en richt zich meer op de volgorde waarin acties moeten worden uitgevoerd. De eerste variatie wordt gekenmerkt door het extreme detail waarin de test wordt uitgevoerd en geëvalueerd. De tweede variatie kan gebruikt worden om meer globale zaken aan het licht te brengen.
- **Heuristic Evalution.** Hierbij worden algemene richtlijnen gebruikt om een oordeel te vormen over de testmaterie. Denk hierbij aan punten als ‘zorg voor goede foutmeldingen’ of ‘wees consistent’.

In het gebied waarin met gebruikers getest wordt, wordt het onderscheid gemaakt tussen ‘process data’ en ‘bottom-line data’. Bij Process data wordt gekeken naar de acties van de gebruiker en wat zij denken bij het uitvoeren ervan. Deze geeft antwoord op de vraag **waarom** iets plaatsvindt. Bottom-line data richt zich op de vraag **wat** er plaatsvindt: hoe lang duurde het om de actie uit te voeren?, hoeveel fouten zijn er gemaakt?, etc.

Aan de hand van dit onderscheid zijn de volgende testmethodes beschikbaar:

- **Process data - Thinking Aloud.** Bij deze methode wordt de gebruiker aangespoord hardop te denken bij het uitvoeren van de test. Op deze manier kunnen specifieke problemen of knelpunten aan het licht komen.
- **Bottom-line data – gegevens verzamelen.** Hierbij worden gegevens verzameld over de uitgevoerde acties en beoordeeld aan de hand van vastgestelde formules.
- **Bottom-line data – SUMI vragenlijst.** Na het doorlopen van de tests worden de testpersonen gevraagd een Software Usability Measurement Inventory (SUMI) vragenlijst in te vullen. De uitkomst van deze vragenlijst kan ingevoerd worden in een softwarepakket om zo cijfers te genereren op het gebied van usability.

Omdat er door de geringe beschikbare tijd slechts ruimte was voor het kiezen van één methode is een afweging gemaakt tussen de beschikbare testmethoden. Gekozen is om gebruik te maken van de Thinking Aloud methode. Deze beslissing is genomen mede op basis van de volgende punten:

- De Cognitive Walkthrough wordt doorgaans uitgevoerd door de ontwikkelaar zelf. De kans bestaat daardoor dat knelpunten over het hoofd worden gezien, mede doordat de ontwikkelaar zelf niet geheel objectief is en al vanaf het begin betrokken is geweest bij de totstandkoming van het systeem.
- De Action Analys methode vereist het tot in groot detail opnemen en vastleggen van gegevens en is daardoor door de tijdsbeperkingen afgefallen.
- De SUMI-test levert te algemene cijfers op en laat niet zien waar de problemen liggen. De nadruk is hier gelegd op het blootleggen van specifieke en aanwijsbare knelpunten.
- De Heuristic Evaluation methode moet door verschillende experts of deskundigen worden uitgevoerd.

De voorkeur lag bij het verzamelen van gegevens direct afkomstig van de (eind)gebruikers. Door gebruik te maken van de Thinking Aloud methode kon niet alleen worden afgeleid op welke (specifieke) punten problemen ontstaan, maar kon (de testpersoon) tegelijkertijd gevraagd worden naar oplossingen of suggesties.

4.6.2 Opstellen testscenario's

Als voorbereiding op de test zijn verschillende testscenario's opgesteld. Deze scenario's vormen een representatie van de meest belangrijke functies die de pilot omhelst. Op aanwijzing van het gebruikte boek zijn de testscenario's ontworpen om van simpele naar complexe taken te gaan. Er zijn drie scenario's opgesteld, waarbij de aan testpersoon de instructie wordt gegeven de stappen één voor één te doorlopen en hardop te denken:

- Het oppakken van een nieuw binnengekomen incident uit de wachtrij.
- Het afhandelen van het zojuist toegewezen incident, vanuit de lijst met openstaande incidenten.
- Het aanmaken van een nieuw incident, waarbij alle relevante velden worden ingevuld. Het incident wordt vervolgens door de testpersoon direct doorgestuurd naar een andere medewerker in dezelfde oplosgroep.

Het complete overzicht van de gebruikte testscenario's is te vinden in bijlage I.

4.6.3 Uitvoeren tests en vastleggen resultaten

Met behulp van de testscenario's zijn de gebruikerstests uitgevoerd met een aantal eindgebruikers.

Hiervoor zijn de programmabestanden van de eerste pilot op één van de beschikbare servers gezet en zijn de benodigde snelkoppelingen aangemaakt. De tests zijn uitgevoerd aan de hand van de volgende vier stappen:

1. De gebruiker krijgt een uitleg van de bedoeling van de test en wat er van hem of haar verwacht wordt.
2. De gebruiker voert aan de hand van een serie testscenario's de test uit. Deze bevatten concrete instructies zoals: 'maak een nieuw incident aan' of 'zet de doorlooptijd op 3 uur'. Door middel van een formulier wordt bijgehouden waar de gebruiker moeite mee heeft en welke opmerkingen de gebruiker maakt.
3. Na het uitvoeren van de tests wordt een lijst opgesteld met de punten die zijn aangedragen door de gebruikers.
4. Op basis van de opgestelde lijst wordt bepaald welke punten meegenomen zullen worden in de verdere ontwikkeling van het systeem. Eenvoudige punten of punten met weinig consequenties, zoals het veranderen van een icoon, kunnen direct meegenomen worden naar de volgende time-box. Hetzelfde geldt voor zaken die de correcte werking van het systeem in de weg staan of bepaalde bugs die optreden. Bij andere wensen, zoals het toevoegen van bepaalde informatie aan een venster of het toevoegen van nieuwe functionaliteiten, kan bepaald worden om deze niet mee nemen binnen de time-boxes.

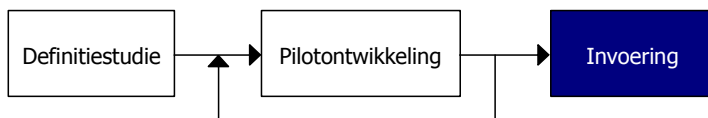
De resultaten van de tests zijn beoordeeld, en er is een lijst met actiepunten opgesteld. De meeste stappen uit de scenario's zijn door de testpersonen met succes uitgevoerd. Verbeterpunten lagen voornamelijk in het vinden van de juiste controls, het oplossen van een aantal kleine bugs en het verbeteren van de usability door bijvoorbeeld het aanbrengen van extra tekstlabels. De punten waarvan is beoordeeld dat deze zonder problemen geïmplementeerd konden worden, zijn samengevoegd en als bouweenheid meegenomen naar de time-box van de tweede pilot. Punten die niet zijn meegenomen in de ontwikkeling hadden voornamelijk betrekking op onduidelijkheden die na een instructie (of door het lezen van documentatie) opgelost werden.

De lijst met punten en de resultaten van de tests is opgenomen in bijlage I.

5

Invoering en afronding

De laatste fase van het ontwikkeltraject, de fase ‘Invoering’ vormt de afsluiting van het ontwikkeltraject. Dit is weergegeven in de onderstaande figuur:



Figuur 24: De fase Invoering

Dit hoofdstuk beschrijft in hoeverre deze fase is uitgevoerd binnen de afstudeerperiode.

5.1 Invoering

Door een aantal verschuivingen in de planning is het ontwikkelde systeem ten tijde van het schrijven van dit verslag nog niet binnen de organisatie ingevoerd. De volledige besluitvorming hieromtrent is terug te vinden in het volgende hoofdstuk.

Wel is in deze fase een helpbestand ontwikkeld, welke in de volgende paragraaf wordt besproken.

5.2 Opstellen helpfiles

Bij het opstellen van de helpfiles voor het gedeelte van het systeem dat is bedoeld voor de gewone gebruikers is, net als bij het opstellen van de Application Style Guide, gebruik gemaakt van de Microsoft richtlijnen in ‘Official Guidelines for User Interface Developers and Designers’ [8]. Hierin worden een aantal vormen van hulp onderscheiden, waarbij de voornaamste de volgende vormen zijn:

- **Procedural help.** Beschrijft de procedure waarin taken moeten worden uitgevoerd. Deze vorm van help richt zich voornamelijk op hoe iets gedaan moet worden.
- **Reference help.** Dient als een online referentie voor bijvoorbeeld snelkoppelingen binnen een applicatie.
- **Conceptual help.** Richt zich op het verstrekken van achtergrondinformatie over een taak of bepaald deel van een applicatie. Deze vorm van help richt zich voornamelijk op wat, of waarom iets gedaan moet worden.

Gekozen is de helpfunctie te implementeren in een vorm van procedural help. Hierbij is gebruik gemaakt van de binnen Windows bekende vorm van help: HTML Help. Door een aantal HTML files aan elkaar te koppelen binnen de applicatie ‘Windows HTML Help Workshop’ is een Windows help bestand gegenereerd.

De beslissing om procedural help te implementeren is voornamelijk gebaseerd op het gegeven dat de gewone gebruikers doorgaans een idee hebben waarom ze het systeem willen gebruiken, bijvoorbeeld om een nieuw incident aan te melden. Het help bestand kan de gebruikers assisteren in het daadwerkelijk uitvoeren van de stappen.

Figuur 25 geeft een indruk van het vervaardigde help bestand:



Figuur 25: Helpfiles voor de gewone gebruikers

Zoals gezegd zijn hier de Microsoft richtlijnen aangehouden. Deze variëren van het gebruik van iconen om onderwerpen en items aan te geven tot het gebruik van verwijzingen (links) binnen het help bestand. Dit laatste is een groot voordeel van het gebruik van een dergelijk help bestand ten opzichte van een papieren handleiding. Het help bestand kan geopend worden vanuit het hoofdmenu.

6

Evaluatie

In dit hoofdstuk wordt een evaluatie gegeven van de afstudeerperiode. Het evalueren wordt grofweg opgesplitst in twee delen: een productevaluatie waarin ingegaan wordt op de opgeleverde producten, en een procesevaluatie waarin het doorlopen proces aan bod komt. Ook wordt geëvalueerd of de doelstellingen zijn gehaald en of de problemen zijn opgelost.

6.1 Productevaluatie

In deze paragraaf zal in worden gegaan op de opgeleverde producten. Er zal aandacht worden geschonken aan de kwaliteit en de bruikbaarheid van de producten en waar ruimte voor verbetering is.

6.1.1 Plan van Aanpak

Het Plan van Aanpak is voor een groot deel gebaseerd op de opdrachtomschrijving, welke voor aanvang van de afstudeerperiode was opgesteld. Het Plan van Aanpak is daarom hoofdzakelijk gebruikt als definitieve vastlegging van de uit te voeren opdracht. De initieel opgestelde planning is in voldoende detail gespecificeerd, per uit te voeren activiteit of op te leveren product. Het doel van het Plan van Aanpak, namelijk overeenstemming bereiken over zaken als de ontwikkelmethode, de uit te voeren activiteiten en de op te leveren producten, is na het voeren van overleg met de bedrijfsmentor bereikt. Als verbeterpunt kan worden aangedragen dat de verplichtingen tussen beide partijen niet zijn vastgelegd. Dit is in dit geval uitsluitend op goed vertrouwen gegaan.

6.1.2 Definitiestudie

De Definitiestudie is de basis geweest voor de daadwerkelijke ontwikkeling van het systeem. Het definiëren van de gebruikersgroepen is voldoende gedetailleerd gedaan, waarbij de nadruk voornamelijk is gelegd op het beoogde systeem. De systeemeisen vormen voldoende een overzicht van de overeengekomen minimum functionaliteiten die het systeem moet bezitten. Wel is een aantal eisen opgenomen die moeilijk, of in ieder geval pas na de invoering, te meten zijn. Het systeemconcept, specifiek de activiteitendiagrammen en Use-cases geven een compleet beeld van het beoogde systeem. De Use-cases zijn met name gedetailleerd, waarbij zelfs is gespecificeerd welke velden de gebruiker in moet gaan voeren. In een GUIDE taakdiagram (voor het uitdiepen van taken) zou een dergelijk niveau van detail moeilijk te verwerken zijn.

Met betrekking tot het databaseontwerp is onvoldoende gekeken naar de gevolgen van het werken in een multi-user omgeving. Zo kan niet worden uitgesloten dat twee gebruikers tegelijkertijd het zelfde incident aan het wijzigen zijn. Wel is de kans op een dergelijke situatie klein, aangezien gegevens alleen gewijzigd kunnen worden wanneer zij de status ‘openstaand en toegewezen’ hebben. Daarnaast kunnen incidenten vanuit de wachtrij niet geopend worden. Achteraf gezien had een ‘locking’ mechanisme ingebouwd moeten worden waarbij gegevens (bijvoorbeeld) vastgezet worden wanneer een andere gebruiker deze aan het bekijken of wijzigen is. Waarschijnlijk was de locking vorm ‘pessimistic locking’ geschikt geweest. Bij deze methode wordt een lock gezet op een set gegevens op het moment dat deze wordt geopend. Andere gebruikers kunnen de gegevens wel lezen, maar niet wijzigen. De lock wordt dan opgeheven wanneer de eerste gebruiker de gegevens afsluit.

Verder is een fout gemaakt bij het opstellen van het klassendiagram. De klassen ‘hoofdcategorie’ en ‘subcategorie’ hadden, zoals bij de klassen ‘helpdeskmedewerker’ en ‘gewone gebruiker’, een specialisatie moeten vormen van een hoofdklasse. In de praktijk heeft deze fout weinig gevolgen, omdat het uitwerken van het juiste model nog steeds kan resulteren in twee aparte tabellen: de tabellen ‘hoofdcategorie’ en ‘subcategorie’. In de huidige opzet bleek het klassendiagram probleemloos om te zetten naar een relationeel databasemodel, welke probleemloos heeft gefunctioneerd.

De Definitiestudie is tot stand gekomen met inbreng en goedkeuring van de bedrijfsmentor, waardoor zeker gesteld kon worden dat het resultaat aansloot bij de wensen en verwachtingen van de organisatie. Het resultaat van deze fase vormde, zo is gebleken, een goede basis voor de verdere specificatie van het systeem middels de schermprototypes.

6.1.3 Testplan en testresultaten

Een voor de hand liggend punt van kritiek is ongetwijfeld de activiteit van het testen van de eerste pilot geweest. Er is zoals reeds beschreven slechts één testmethode gebruikt. Om een volledig en correct oordeel te vormen over de pilot had ook getest moeten worden met een methode waarbij ‘bottom-line’ data wordt verzameld. Echter hier geldt ook de regel: hoe meer er getest moet worden, hoe meer tijd dit kost. De afweging is gemaakt tussen het ontwikkelen van noodzakelijke functionaliteiten binnen de gestelde time-box en het verzamelen van een reeks cijfers over de pilot. Hier is dus gekozen voor het eerste.

De methode die wel is gebruikt, de ‘Thinking Aloud’ methode, is wel met succes toegepast. Er is nuttige informatie verzameld met betrekking tot aanpassingen en toevoegingen aan de pilot en de testpersonen hebben kunnen zien dat een deel van het systeem al voor het grootste gedeelte ontwikkeld was. Daarnaast geven de opgestelde testscenario’s een representatief beeld van de acties die met de pilot zijn uit te voeren.

Naast de gekozen testmethode kan ook de aandacht worden gericht op het feit dat alleen de eerste pilot met eindgebruikers is getest. Idealiter wordt elke pilot meerdere malen getest (meerdere iteratieslagen over een pilot) met verschillende testmethoden, net zo lang tot alle betrokkenen tevreden zijn. Deze aanpak is in een dergelijk project met een strakke planning en deadline simpelweg niet mogelijk.

Wel zijn enkele verzachtende omstandigheden in deze situatie te onderkennen:

- De tweede pilot is qua functionaliteit en werking voor een groot deel gelijk aan de eerste pilot.
- Het deel van het systeem dat voor de gewone gebruikers bestemd is (onderdeel van de derde pilot) is voor de gemiddelde gebruiker zeer makkelijk te gebruiken, zelfs zonder instructie. Dit komt deels door het implementeren van een SDI omgeving en deels door de beperkte functionaliteiten waarover de gebruiker beschikt: een gebruiker kan simpelweg weinig fout doen.
- Een deel van de vierde pilot bevat functionaliteiten die, naar verwachting, weinig gebruikt zullen worden: het beheer van categorieën, gebruikers en oplosgroepen en de rapportagemogelijkheden. Een niet-optimale usability zou vervelend, maar niet cruciaal zijn.

6.1.4 Pilotontwikkelpannen en ontwikkelde pilots

In de pilotontwikkelpannen zijn de pilots, op basis van de Definitiestudie, verder uitgediept. In de originele planning die in het Plan van Aanpak was opgesteld, was rekening gehouden met het opstellen van technische en organisatorische veranderingen specifiek voor elke pilot. Deze activiteiten (voorgeschreven in IAD) bleken overbodig en zijn derhalve geschrapt; het éénmaal specificeren van deze veranderingen voor het gehele systeem in de Definitiestudie bleek voldoende.

In de pilotontwikkelpannen is voornamelijk de nadruk gelegd op het vervaardigen van schermprototypes met ondersteuning van de Application Style Guide. Door gebruik te maken van de uiteindelijke ontwikkelomgeving om de prototypes te vervaardigen kon relatief snel een zeer nauwkeurig beeld worden geschetst van de uiteindelijke schermen zonder programmeerwerk. De Application Style Guide is uitgebreid genoeg en bevat voldoende (eenduidige) standaarden voor een systeem van deze omvang. Omdat het gebruik van grafische elementen beperkt is gebleven, is niet diep ingegaan op kleurgebruik of het gebruik van logo's en lettertypen.

Als laatste zijn de bouweenheden in de time-boxes klein genoeg gespecificeerd. De bouweenheden bleken uiteindelijk prima te realiseren op basis van de schermprototypes, mede door de beschreven logica, regels en beperkingen.

De pilotontwikkelpannen zijn door de bedrijfsmentor gevalideerd alvorens over is gegaan tot de feitelijke bouwactiviteiten.

6.1.5 Ontwikkelde pilots

Op basis van de in de time-boxes gespecificeerde bouweenheden zijn de pilots in de ontwikkelomgeving gebouwd. Door zelf tussentijds routineus te testen op een werkende en correcte functionaliteit kan in ieder geval gesteld worden dat het grootste gedeelte van de pilots voldoet. Natuurlijk kan een volledig oordeel pas worden gegeven na de invoering van het systeem, waarin eventueel gemiste punten aan het licht komen. Het uitvoeren van tests over alle pilots zou ook bijgedragen hebben aan dit oordeel.

6.2 Procesevaluatie

Deze paragraaf zal voornamelijk ingaan op het gevolgde proces binnen de afstudeerperiode. Ingegaan zal worden op de gekozen ontwikkelmethode en iteratiestrategie, evenals de gekozen pilotstrategie en pilots. Daarnaast komen de gebruikte methoden aan bod en zal besproken worden in welke mate de planning is aangehouden en waarom er van is afgeweken.

6.2.1 Gekozen ontwikkelmethode en iteratiestrategie

Bij aanvang van de projectperiode is gekozen om met de IAD ontwikkelmethode te gaan werken. Deze keuze is gemaakt op basis van de reeds aanwezige ervaring met, en kennis van de methode. Door de ontwikkelmethode als leidraad te gebruiken en alleen te gebruiken wat nodig of relevant leek, is een eigen invulling aan de methode gegeven. Zo zijn er geen workshops gehouden maar is de informatie vergaard door (informele) interviews en overleg. Bij eerder uitgevoerde projecten - waaronder het IP-03 project binnen de opleiding - is deze methode goed toepasbaar gebleken bij het ontwikkelen van software. Natuurlijk had er ook onderzoek gedaan kunnen worden naar alternatieve ontwikkelmethoden. Echter, het resultaat was, naar verwachting, uiteindelijk hetzelfde geweest. Gelet op de beperkt beschikbare tijd is derhalve gekozen voor bekende middelen. De problemen en knelpunten waren voornamelijk inhoudelijk van aard en waren (hoogstwaarschijnlijk) ook opgetreden bij het gebruik van een andere methode. Vanuit de organisatie is de keuze open gelaten en op de gekozen methode is positief gereageerd. Met name het vastleggen van keuzes en het opstellen van documentatie (in het bijzonder het databaseontwerp) werd gezien als voordeel. Door het systeem in pilots te ontwikkelen en meteen na het ontwikkelen van de eerste pilot een testronde te houden werd vertrouwen gekweekt en kregen de betrokkenen een idee van de werking en opzet van het systeem.

De gekozen iteratiestrategie, het incrementeel ontwikkelen, heeft eveneens goed uitgepakt. Door niet over de fase Definitiestudie te itereren maar deze vast te zetten na goedkeuring is het risico van het optreden van 'scope creep' vermeden. Dit gaf mij als ontwikkelaar ook de geruststelling dat de functionaliteiten die tot dusver waren overeengekomen gebouwd moesten gaan worden, en niet meer. Door het daadwerkelijke bouwen van de pilots zo lang mogelijk uit te stellen, is zoveel mogelijk tijd (ook buiten de officiële werkuren om) besteed aan het onder de knie krijgen van de ontwikkelomgeving.

6.2.2 Gekozen pilotstrategie en bepaling pilots

De gekozen pilotstrategie, het ontwikkelen van de pilots met een diepe functionaliteit, geïnspireerd door het 'Juicy Bits First' principe heeft eveneens goed uitgepakt. Door het onderdeel Incidentbeheer als het meest 'juicy' te kenmerken, kon bijna zeker de garantie worden gegeven dat bij het verstrijken van de deadline de helpdeskmedewerkers in ieder geval dit onderdeel, met alle beschreven functionaliteiten, in gebruik konden gaan nemen.

Het opdelen van het gehele systeem in vier pilots heeft ook goed gewerkt. De pilots konden los van elkaar worden gebouwd en getest, waarbij enkele componenten (zoals de database en basisinterface elementen) wel eerst als basis dienden te worden ontwikkeld.

6.2.3 Gebruikte technieken

Als voornaamste techniek tijdens het ontwikkeltraject is de modelleringstechniek UML gebruikt. Met deze techniek was voor aanvang van de projectperiode al enige ervaring opgedaan, in het bijzonder met het opstellen van klassendiagrammen en Use-cases. Deze kennis is uitgebreid met kennis over toestandsdiagrammen. Door hoofdzakelijk UML te gebruiken om het systeem in modellen weer te geven is een zekere consistentie binnen de Definitiestudie gevormd. UML is in de praktijk goed bevallen. Zo geeft het klassendiagram een goed abstract beeld van het systeem en bleek deze prima om te zetten naar een relationeel databasemodel.

Naast UML zijn ook zaken uit GUIDE gebruikt, specifiek genomen navigatieschema's en het vervaardigen van schermprototypes. Voor GUIDE is gekozen omdat hier al enige ervaring mee was en UML niet toereikend bleek voor het ontwerpen van de grafische interface. Door gebruik te maken van (delen van) GUIDE werd het mogelijk de gebruikersinterface gedetailleerd te specificeren voor goedkeuring. Hierbij zijn geen noemenswaardige problemen opgetreden.

Het vergaren van informatie is voornamelijk verlopen middels informele interviews en vraaggesprekken. Deze aanpak is bepaald door de heersende bedrijfscultuur (men is immers gewend om informeel met elkaar om te gaan) en is goed bevallen.

6.2.4 Planning en de fase Invoering

Tijdens het ontwikkeltraject is de planning (welke is opgesteld in het Plan van Aanpak) een aantal malen gewijzigd om in te spelen op de actuele situatie. De voornaamste wijziging in de originele planning, welke is opgenomen in bijlage C, had te maken met het verschuiven van de start van de fase Pilotontwikkeling. De laatste versie van de planning, welke het dichtst in de buurt komt van de werkelijke situatie, welke eveneens is opgenomen in bijlage C.

Als voornaamste reden hiervoor kan de onbekendheid met Visual Basic .NET worden genoemd. Tijdens de fase Definitiestudie werd duidelijk dat er veel onderzoek gedaan moest worden naar de wijze waarop zaken die bedacht zijn binnen de fase feitelijk moesten worden geïmplementeerd. Denk hierbij aan de algemene werkwijze binnen Visual Basic .NET door middel van objectgeoriënteerd werken, de wijze waarop de database moet worden benaderd, hoe met gegevens om moet worden gegaan, het gebruik van externe DLL's en controls, het benaderen van de Active Directory Service etc. Hiervoor zijn vele tests uitgevoerd, is veel op internetsites en internetfora rondgezoekt en is een aantal boeken doorgenomen. Zonder zeker te weten hoe de aanwezige ideeën tot uitvoer konden worden gebracht kon immers niet gestart worden met de bouw van de pilots.

Tijdens het doen van onderzoek bleek dat de complexiteit van de werking van Visual Basic .NET enigszins was onderschat. Het implementeren van ogenschijnlijk simpele functionaliteiten, zoals het sorteren van items in een overzicht of het doorgeven van gegevens aan verschillende vensters, bleek moeilijker dan in eerste instantie gedacht. Omdat er geen ervaring was met een volledig objectgeoriënteerde programmeeromgeving als deze is hier veel tijd in gaan zitten. Ondanks dat er door de gekozen iteratiestrategie de daadwerkelijke pilotontwikkeling zo laat mogelijk is neergezet, was het verschuiven van de start van de volgende fase onvermijdelijk.

Toen de benodigde kennis over de te bouwen onderdelen eenmaal was vergaard, zijn de pilots binnen de fase Pilotontwikkeling binnen de in de planning gestelde tijd ontwikkeld. Wel

is de planning aangepast om ruimte te maken voor het opstellen van een bouwplan van dit verslag. Er had vanaf het begin af aan rekening gehouden moeten worden met het schrijven van het bouwplan.

Alle bouweenheden zijn binnen de time-boxes ontwikkeld, behalve het uitvoeren van de gebruikerstests over de eerste pilot. Deze zijn dan ook meegenomen naar de time-box van de tweede pilot. Na de ontwikkeling van de pilots werd duidelijk dat de laatste fase, de fase Invoering, in gevaar kwam. Om alle beoogde activiteiten uit te voeren was door alle verschuivingen binnen de planning geen tijd meer, ondanks het opnemen van een week voor uitloop. De keuze is gemaakt om het invoeringsproces van het systeem uit te stellen en alle aandacht te vestigen op dit eindverslag. Het doorschuiven van onderdelen in de planning heeft dus een 'sneeuwbal effect' gehad op de onderdelen die daarna waren ingepland.

De beschreven situatie had beter beheerst kunnen worden door het incalculeren van tijd voor het uitvoeren van onderzoek. Hierbij moet wel vermeld worden dat een dergelijke uitloop in de planning nooit helemaal voorkomen had kunnen worden. Er had simpelweg niet voorspeld kunnen worden hoeveel tijd deze activiteiten in beslag zouden nemen.

Ook bleek de fase Invoering, en dan met name het opstellen van de benodigde helpfiles voor het gehele systeem, onderschat te zijn. Onder andere in het opstellen van online helpfiles - compleet met kruisverwijzingen - voor een systeem van dergelijke omvang zal nog aardig wat tijd moeten worden geïnvesteerd.

6.3 Doelrealisatie en gerealiseerde eisen

De kern van de doelstelling van het project, zoals deze bij aanvang is gedefinieerd luidt: "De doelstelling van de opdracht is het ontwerpen, ontwikkelen en implementeren van een softwarepakket dat het onder andere mogelijk maakt het complete traject voor het afhandelen van incidenten te automatiseren en te stroomlijnen. Het doel is het aanbrengen van structuur in de betrokken processen en de informatievoorziening naar de gebruiker te vergroten".

Ondanks het gegeven dat het systeem bij het schrijven van dit verslag nog niet binnen de organisatie is ingevoerd kan gesteld worden dat de doelstelling voor het grootste gedeelte is behaald. Een compleet systeem is ontworpen en gebouwd, en alle functionaliteiten die tijdens de fase Definitiestudie naar voren zijn gekomen zijn ook daadwerkelijk geïmplementeerd. Er is een duidelijke structuur aangebracht in de afhandeling van incidenten en problemen en de informatievoorziening kan door het gebruik van het systeem op diverse punten worden verbeterd.

Of alle systeemeisen zijn gerealiseerd is moeilijker te bepalen. Van alle basissysteemeisen kan gesteld worden dat deze zijn ontwikkeld binnen de pilots. Omdat het systeem nog niet is ingevoerd kan performance-eis G1 niet worden beoordeeld. Deze luidt:

Het systeem mag geen verstoring of vertraging (door middel van het aantal database-transacties op de Microsoft SQL 2000 server) veroorzaken van de primaire werkprocessen, zoals het DisKis systeem.

Deze eis is alleen te meten na de invoering van het systeem, over een periode van tijd. Een meting van een drietal servers wees uit dat de huidige belasting van deze servers miniem is. Hierbij is gekeken naar zaken als het netwerkverkeer en de belasting op de processoren van de servers. Het ligt niet in de verwachting dat het nieuwe systeem een significante belasting zal vormen voor de uiteindelijke productieserver. De databaseserver zal wel constant in gebruik zijn door de periodieke voortgangscontrolefunctie, maar dit zal hoogstwaarschijnlijk op een krachtige server niet of nauwelijks impact hebben.

Verder staat de operationele eis H3 open. Deze luidt:

Het systeem moet stabiel kunnen werken binnen de Citrix omgeving.

Ook deze eis kan pas worden beoordeeld na de feitelijke invoering van het systeem binnen de organisatie. Bij het uitvoeren van de gebruikerstests over de eerste pilot en bij het routinieuze testen tijdens de ontwikkeling zijn geen stabiliteitsproblemen naar voren gekomen. Alleen door het langdurig volgen van klachten en meldingen na de invoering kan hierover uitsluit worden gegeven.

6.4 Opgeloste problemen

Doordat het systeem bij het schrijven van dit verslag nog niet is ingevoerd, kan niet definitief worden vastgesteld dat alle problemen daadwerkelijk zijn opgelost en of aan alle wensen is voldaan. Wel kan een voorzichtige schatting worden gemaakt aan de hand van de ontwikkelde onderdelen. Aan de hand van de punten genoemd in de probleemanalyse uit paragraaf 4.2 is in de onderstaande tabel aangegeven op welke wijze deze zijn aangepakt:

Probleem	Oplossing
Er ontbreekt een duidelijke, uniforme wijze van het afhandelen van incidenten. Helpdeskmedewerkers hebben geen overzicht van de openstaande incidenten, wie met welke taken bezig is en hoeveel tijd hiervoor staat.	Het ontwikkelde systeem kan bijdragen aan het aanbrengen van meer structuur in de werkzaamheden. Met name de aanwezigheid van een overzicht met openstaande incidenten en problemen dat te sorteren is op prioriteit of deadline draagt hieraan bij. Daarnaast kunnen incidenten centraal binnenkomen wat de uniformiteit en overzichtelijkheid verhoogt. Tevens zijn de incidenten en problemen van andere helpdeskmedewerkers in de oplosgroep op te vragen.
De informatievoorziening aan de eindgebruikers (de klant) verdient verbetering. Soms is niet duidelijk voor de gebruiker wat de status van zijn of haar incident is of wanneer deze in behandeling wordt genomen.	Het gedeelte van het systeem dat bedoeld is voor de gewone gebruikers voorziet in deze behoefte. Gebruikers kunnen vanuit het hoofdmenu een overzicht opvragen van alle incidenten die door hem of haar zijn aangemeld, samen met alle relevante informatie.
Door het niet vastleggen van binnengekomen incidenten wordt het niet altijd duidelijk dat er achterliggende problemen bestaan. Wanneer problemen wel geïdentificeerd en opgelost zijn wordt soms niet vastgelegd wat de oplossing is geweest.	Het ontwikkelde systeem maakt het mogelijk diverse gegevens met betrekking tot incidenten en problemen vast te leggen. De oorzaak van een probleem of incident kan worden vastgelegd, welke op een later tijdstip kan worden gebruikt voor het identificeren van problemen. Door ook de oplos-



	sing vast te leggen kan deze door alle betrokken medewerkers gebruikt worden bij het oplossen van problemen.
De helpdesks worden belast met veelvoorkomende problemen die de gebruikers gemakkelijk zelf op kunnen lossen. Dit veroorzaakt een onnodige belasting op de helpdesks. Een aanname hierbij is dat de gebruiker het probleem zelf op wil lossen. Uit de praktijk is gebleken dat het sommige gebruikers niet interesseert wat het probleem is en hoe deze opgelost kan worden; zij willen slechts zo snel mogelijk geholpen worden.	Het Kennissysteem moet een groot deel van de informatievoorziening naar de gebruikers toe gaan verzorgen. Het Kennissysteem is door de helpdeskmedewerkers bij te houden. Naar eigen inzicht zal moeten worden bepaald in welke mate de gebruikers moeten worden gewezen op de aanwezige kennis in het systeem.
Het management heeft geen inzicht in de resultaten van de afdelingen met betrekking tot de opgeloste problemen en afgehandelde incidenten.	Dit punt is aangepakt door het implementeren van een rapportagefunctie in het systeem. Nieuwe rapporten kunnen, wanneer nodig, door de medewerkers zelf worden aangemaakt.
Veel kennis moet mondeling worden overgedragen aan nieuwe medewerkers.	Gegevens binnen het systeem worden permanent in de database opgeslagen. Deze gegevens kunnen door nieuwe medewerkers gebruikt worden bij het identificeren van problemen.
Door gebrek aan kennis van het onderwerp kan het voorkomen dat een incident of aanvraag bij de verkeerde afdeling terecht komt.	In de huidige opzet kan dit probleem nog steeds optreden. Een gebruiker kan, om welke reden dan ook, een incident onder een verkeerde categorie aanmelden of de verkeerde afdeling opbellen.
Het komt geregeld voor dat gebruikers een e-mail sturen naar een specifieke medewerker van één van de afdelingen met vragen of verzoeken. Wanneer deze medewerker niet aanwezig is, kan een kwestie onnodig lang blijven liggen.	Binnen een oplosgroep is het mogelijk de incidenten van andere medewerkers over te nemen. Bij afwezigheid kunnen alle openstaande incidenten van een medewerker aan een andere medewerker worden toegewezen.

De overlap van werkzaamheden kan worden opgevangen door een zorgvuldige indeling van de categorieën binnen het systeem te realiseren. Zo is het mogelijk de hoofdcategorie 'Printers' op te splitsen in de subcategorieën 'Toners' en 'Storingen'. Door deze subcategorieën aan verschillende oplosgroepen toe te wijzen kan een duidelijke scheiding worden aangebracht.

De conclusie kan hier getrokken worden dat, wanneer het systeem uiteindelijk in de organisatie is ingevoerd, bijna alle problemen aangepakt zullen worden. Wel zal veel afhangen van de discipline van de helpdeskmedewerkers om het systeem te gaan gebruiken en de gewone gebruikers te wijzen op het bestaan ervan.

6.5 Invoering van het systeem en de toekomst

In overleg met de opdrachtgever en de bedrijfsmentor is afgesproken dat er na de afstudeerperiode nog een aantal weken gewerkt zal worden aan de afronding van het ontwikkeltraject. De onderstaande activiteiten zullen nog worden uitgevoerd:

- Het opstellen van helpfiles voor het deel van het systeem dat door de helpdeskmedewerkers wordt gebruikt.
- Het opstellen van documentatie voor het onderhoud van het systeem (onder andere het aanpassen van instellingen, het onderhoud van de database e.d.).
- Het opstellen van een korte schriftelijke instructie opstellen voor gewone gebruikers en het geven van voorlichting aan de helpdeskmedewerkers en andere betrokkenen over het gebruik van het systeem in de praktijk en de benodigde organisatorische veranderingen.
- Het plaatsen van een definitieve database op één van de aanwezige productieservers en het begeleiden van de betrokken personen met het vullen van de database voor de ingebruikname van het systeem.
- Het creëren van versie 1.0 van het systeem, deze op één van de productieservers plaatsen en het adviseren bij het plaatsen van snelkoppelingen in de gebruikersprofielen.
- Het overzien van een testperiode na de invoering waarin eventuele bugs zullen worden weggewerkt.

De schatting is dat het bovenstaande twee tot drie weken tijd in beslag zal nemen. Omdat er vanuit de organisatie geen (dringende) behoefte is het systeem zo snel mogelijk in te voeren wordt de uitstel van de invoering van het systeem niet als een cruciaal obstakel beschouwd.

6.6 Conclusie

Afsluitend kan ik met genoeg terugkijken op de doorlopen afstudeerperiode. De samenwerking met alle betrokken personen heb ik als prettig ervaren en mogelijkheid voor overleg of het stellen van vragen was altijd aanwezig.

Op het technische vlak is behoorlijk wat vooruitgang geboekt en kennis vergaard. De belangrijkste ontwikkeling ligt in de Visual Basic programmeertaal onder het .NET platform. Hierbij is het werken in een objectgeoriënteerde taal een belangrijk aspect geweest. Daarnaast is ervaring opgedaan met Stored Procedures en Microsoft SQL Server 2000. Deze zaken zullen zonder twijfel in de toekomst goed van pas komen.

Er is een behoorlijk groot, complex en compleet systeem ontwikkeld. Daarnaast was de opdracht breed waardoor er is gewerkt op een tal van gebieden, van systeemontwerp tot grafisch ontwerp. Het resultaat is een maatwerkpakket dat door de betrokkenen enthousiast ontvangen en dat aansluit op de organisatie. Vooral het implementeren van meerdere ingangen naar het systeem gebaseerd op gebruikersgroepen is een mogelijkheid die bij de onderzochte standaardpakketten niet is terug te vinden. Ondanks de aanwezigheid van de genoemde verbeterpunten is de verwachting dat het systeem met succes ingevoerd kan worden binnen de organisatie.

Literatuurlijst

- [1] R. J. H. Tolido – 1997, IAD: het evolutionair ontwikkelen van informatiesystemen, Academic Service, Schoonhoven, ISBN: 9039504016
- [2] David Redmond-Pyle & Alan Moore – 1995, Graphical User Interface Design & Evaluation (GUIDE), Pearson Education Limited, ISBN: 013315193-X
- [3] Joseph Schmuller – 1999, Sams Teach Yourself UML in 24 Hours, SAMS Publishing, ISBN: 0672322382
- [4] C. Lewis & J. Rieman – 1993, Task-Centered User Interface Design: A Practical Introduction, shareware
- [5] ITSMF – werkgroep 'IT Service Management, een introductie' - 2000, IT Service Management, een introductie, ITSMF Nederland, ISBN: 9080492817
- [6] J.A. Vandenbulcke – 1997, Databasesystemen voor de praktijk 6^e druk, Kluwer Bedrijfswetenschappen, ISBN: 9026723172
- [7] Microsoft SQL Server 2000 Online Books,
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/sqlserver2000.asp>
- [8] Official Guidelines for User Interface Developers and Designers,
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwue/html/ch02a.asp>
- [9] Food's Icons, <http://www.foood.net/>
- [10] Willis, et al – 2004, Beginning Visual Basic .NET 2003, Wiley Publishing Inc., ISBN: 0764556584
- [11] Barwell, et al – 2003, Professional Visual Basic .NET 2nd Edition, Wiley Publishing Inc., ISBN: 0764544004



Verklarende woordenlijst

ADS	Active Directory Service. Stelt systeembeheerders onder andere in staat policies aan werkstations te koppelen.
Class	De definitie van een object waar eigenschappen, gebeurtenissen en methoden kunnen worden gedefinieerd.
DataSet	Een object binnen Visual Basic, welke gevuld kan worden met gegevens uit een database.
DLL	Dynamic Link Library. DLL's kunnen binnen een applicatie gebruikt worden om (losse) functionaliteiten aan te roepen.
GUIDE	Graphical User Interface And Evaluation. Een verzameling van methoden en technieken gericht op het ontwerpen van grafische gebruikersinterfaces.
HTML Help	Het door Microsoft geïntroduceerde standaardsysteem voor het weergeven van gebruikershulp middels een gecompileerd bestand. Dit bestand wordt aangemaakt op basis van HTML (HyperText Markup Language) bestanden.
IAD	Iterative Application Development. Een systeemontwikkelingsmethode waarin iteratief werken central staat.
ITIL	IT Infrastructure Library. Een verzameling 'best practices' voor het inrichten van een IT-dienstverlening.
LDAP	Lightweight Directory Access Protocol. Een protocol voor het benaderen van een Active Directory Service.
MDI	Multiple Document Interface. De opstelling van een interface waarin de gebruiker meerdere vensters of documenten tegelijkertijd geopend kan hebben.
MS SQL Server 2000	Microsoft SQL Server 2000. Een Database Management System ontwikkeld door Microsoft.
.NET platform	Een door Microsoft gestart initiatief voor server en desktop ontwikkeling gericht op de toekomst.
Object	Een instantie van een class.
PNG	Portable Network Graphics. Een bestandsformaat voor afbeeldingen.



Prototype	Een indruk of ontwerp van een (deel) van een grafische gebruikersinterface.
SDI	Single Document Interface. De opstelling van een interface waarbij de gebruikers slechts één venster of document tegelijkertijd open kan hebben staan.
Stored Procedure	Een enkele of een verzameling (T-)SQL opdrachten.
Style Guide	Een document waarin standaard worden gedefinieerd op het gebied van grafische elementen, kleurstellingen, controls etc.
SUMI	Software Usability Measurement Inventory. Een testmethode waarbij de kwaliteit van software wordt gemeten vanuit het oogpunt van de gebruiker.
T-SQL	Transact-SQL. De door Microsoft uitgebreide SQL taal welk is uitgebreid met mogelijkheden zoals Stored Procedures en triggers.
UML	Unified Modeling Language. Een derde-generatie modelleringstechniek geschikt voor systeemontwikkeling.
Visual Basic	Een door Microsoft ontwikkelde programmeertaal en (software)ontwikkelomgeving.
XML	eXtensible Markup Language. Standaard hiërarchische datastructuur waarmee platform onafhankelijk gecommuniceerd kan worden.



Bijlagen

Interne bijlage

Bijlage A Opdrachtschrijving

Externe bijlagen

Bijlage B	Plan van Aanpak
Bijlage C	Originele en aangepaste planning
Bijlage D	Definitiestudie
Bijlage E	Pilotontwikkelpilot 1
Bijlage F	Pilotontwikkelpilot 2
Bijlage G	Pilotontwikkelpilot 3
Bijlage H	Pilotontwikkelpilot 4
Bijlage I	Testplan en testresultaten

Bijlage A

Opdrachtomschrijving

Danny Evenwel

Kenmerk: DOA2004-1.2.17

Incident Management, Problem Management en Knowledge Base binnen de AWV

De Algemene Woningbouw Vereniging (AWV) is met bijna 18.000 woningen en bedrijfsruimten een middelgrote woningcorporatie in Amsterdam. De AWV, opgericht in 1910 vanuit de socialistische arbeidersbeweging, kan zich beroepen op een lange traditie op het gebied van wonen in Amsterdam. De AWV is een particuliere non-profit organisatie. De primaire doelstelling van de AWV was oorspronkelijk "het streven naar goede huisvesting voor arbeiders". Door de maatschappelijke ontwikkelingen in de afgelopen jaren is deze veranderd in "uitsluitend werkzaam zijn in het belang van de volkshuisvesting". De belangrijkste taken die hieruit voortvloeien zijn het bouwen, renoveren, onderhouden en beheren van woningen in de sociale sector.

Er wordt bij de AWV momenteel gewerkt aan een nieuw ondernemingsplan voor de periode 2004-2008. Tot voor kort heeft de nadruk vooral op het beheren van woningen gelegen. Vanaf 2004 wil de AWV die nadruk nog verder ontwikkelen. Het nieuwe ondernemingsplan heeft ook gevolgen voor de interne organisatie. Zo zal er ondermeer gekeken gaan worden waar activiteiten van verschillende afdelingen elkaar overlappen. Vervolgens worden er -per afdeling- nieuwe werkplannen ontwikkeld, waarin de raakvlakken met andere afdelingen worden beschreven en er duidelijke afspraken worden gemaakt om zo efficiënt mogelijk met de beschikbare resources om te gaan.

Een voorbeeld van overlappende werkzaamheden, waarvoor betere afspraken gemaakt kunnen worden, betreft de afdelingen ICT en Facilitaire Dienst. Beide afdelingen beschikken over een eigen eerstelijns helpdesk. In het nieuwe ondernemingsplan zal worden vastgelegd dat deze afdelingen nauwer zullen gaan samenwerken en overlappings in de werkzaamheden zoveel mogelijk zullen gaan voorkomen.

Binnen de AWV ontbreekt op dit moment een gestructureerde wijze van afhandelen van klachten voor de helpdeskmedewerkers. Hierdoor worden klachten en problemen soms op een ad hoc wijze afgehandeld wat de efficiëntie en overzichtelijkheid van de werkzaamheden niet ten goede komt. Eindgebruikers nemen via de telefoon, per e-mail of in eigen persoon contact op met één van de medewerkers waarna deze, vaak op eigen wijze, de zaak in behandeling neemt.

Door deze werkwijze wordt er niet altijd vastgelegd wat de oplossingen voor problemen zijn geweest. Nieuwe helpdeskmedewerkers moeten hierdoor langer ingewerkt worden. Ook komen bepaalde problemen vaak terug, zoals problemen met het in- en uitloggen en problemen met printers. Deze zijn in veel gevallen door de gebruikers zelf op te lossen; er is echter geen plek waar gebruikers naar verwezen kunnen worden voor veel voorkomende problemen. Tevens is het vaak niet duidelijk welke knel- of verbeterpunten er binnen de ICT infrastructuur bestaan. Door de ongestructureerde wijze van het oplossen van incidenten komen achterliggende problemen vaak niet naar voren en wordt hier niets mee gedaan.



Het doel van de afstudeeropdracht is het ontwikkelen van een softwarepakket dat het onder andere mogelijk maakt het complete klachtenbehandelingstraject te automatiseren. Dit systeem moet worden ontworpen op basis van het ITIL raamwerk. Wel zal het systeem worden aangepast aan de wensen en eisen van de organisatie.

Het systeem zal bestaan uit de volgende onderdelen:

- Knowledge Base en incidentinvoer. Een eindgebruiker moet hiermee een incident kunnen melden vanaf iedere werkplek. Ook moet de eindgebruiker de status van zijn of haar lopende incidenten kunnen opvragen. De Knowledge Base moet dienen als kennisstelsel waarnaar eindgebruikers verwezen kunnen worden voor oplossingen van veelvoorkomende problemen.
- Incident Management. Na binnenkomst van het incident wordt deze in behandeling genomen door een helpdeskmedewerker. Het gaat hier dus om het complete afhandelingstraject van een incident. Incidenten moeten permanent worden opgeslagen, zodat Problem Management op basis hiervan problemen kan definiëren moet er de mogelijkheid zijn factuuroverzichten op te stellen en uit te draaien met betrekking tot afgehandelde incidenten voor derden.
- Problem Management. Op basis van incidenten moeten problemen (Known Errors) gedefinieerd kunnen worden. Hierna moet het mogelijk zijn een probleembeheerder toe te kennen en oplossing aan het probleem te koppelen. Ook hier gaat het om het complete traject van de afhandeling van problemen.

De volgende software zal gebruikt worden:

- Windows Operating System.
- Visual Basic .NET.
- Microsoft Office.
- Photoshop / Illustrator.
- MS SQL 2000 database front-end.
- CVS software.

De volgende ideeën bestaan binnen de organisatie:

- Het bestaande Windows 2000 domein kan gebruikt worden als basis voor het instellen van gebruikersrechten.
- Het onderdeel dat zorg draagt voor de Knowledge Base en het invoeren van incidenten zal op de centrale applicatieserver geplaatst worden. Deze zal vervolgens vanaf deze op te starten zijn vanaf iedere werkplek door middel van een snelkoppeling, welke geplaatst wordt in het profiel van iedere gebruiker.

In het kader van de afstudeeropdracht zullen de volgende activiteiten verricht worden:

- Opstellen van een plan van aanpak.
- Doen van onderzoek naar de wijze van invoering van ITIL binnen de organisatie.
- Vervaardigen van systeemdokumentatie in de vorm van een Definitiestudie en enkele Pilotontwikkelplannen.
- Vervaardigen van het grafische- en interactieontwerp van het te ontwikkelen systeem.
- De daadwerkelijke bouw van de onderdelen van het systeem zelf door middel van Visual Basic .NET en MS SQL 2000.

- Bouwen van benodigde helpfiles voor gebruik op het Intranet en via de software zelf.
- Uitvoeren van gebruikerstests over de ontwikkelde producten.

Als ontwikkelmethode zal IAD worden gebruikt. De ontwikkeling van de software zal worden opgedeeld in een nog te bepalen aantal pilots, welke volgens een nog te bepalen ontwikkelstrategie zullen worden ontwikkeld. Verder zal ITIL als achterliggende methode worden gebruikt. Waar nodig zal hiervan worden afgeweken om op de wensen en eisen van de organisatie in te spelen.

De volgende technieken zullen gebruikt worden:

- Diverse technieken uit GUIDE.
- UML technieken.
- Interviewtechnieken.
- Testtechnieken.

De nadruk zal gelegd worden op de efficiëntie en gebruiksgemak van het te ontwikkelen systeem. De drempel voor het gebruiken van het systeem voor het melden van klachten moet zo laag mogelijk gemaakt worden, om te voorkomen dat er teruggevallen wordt op de oude manieren van het melden van klachten. Verder is het belangrijk dat medewerkers van de helpdesks genotificeerd worden van bijvoorbeeld binnenkomende incidenten of naderende deadlines. Tevens moet de eindgebruiker op de hoogte worden gehouden van de voortgang van zijn of haar incident, zij het via e-mail of via een voortgangsrapportagemogelijkheid in de te ontwikkelen software.

De volgende producten zullen opgeleverd worden:

- Plan van aanpak.
- Definitiestudie.
- Pilotontwikkelpannen.
- De software zelf in de vorm van één of meer applicaties.
- Online helpfiles voor de eindgebruikers.