

.

.

.

.

.

Storing Management Systeem

Procesverslag

Afstudeeropdracht

Versie 1.6

Auteur: André de Nijs

Plaats: Zoetermeer

Datum: 06-07-2004

Inhoudsopgave

1 INLEIDING

- 1.1 Aanleiding
- 1.2 Het bedrijf
- 1.3 Vooronderstellingen
- 1.4 Vooronderstellingen bedrijf

2 ANALYSE EN PROBLEEMVERKENNING

- 2.1 Uitgangssituatie
- 2.2 Concurrentieanalyse
- 2.3 Aanwezige kennis
- 2.4 Doelgroep
- 2.5 Doelstellingen

3 UITVOERING

- 3.1 Globale aanpak
- 3.2 Gebruikte hulpmiddelen en technieken
- 3.3 Aanpak definitiestudie
- 3.4 Aanpak pilotontwikkeling
- 3.5 Aanpak invoering

4 RESULTATEN

- 4.1 Definitiestudie
- 4.2 Pilot 1 Aanmelding clientsysteem
- 4.3 Pilot 2 storingmelding
- 4.4 Pilot 3 Oplossingmelding
- 4.5 Pilot 4 Feedback naar gebruiker
- 4.6 Pilot 5 Oplossingherkenning
- 4.7 Invoering
- 4.8 Go-/ no-go moment
- 4.9 Checklist uitvoering
- 4.10 Korte handleiding gebruiker

5 EVALUATIE

- 5.1 Evaluatie product
- 5.2 Evaluatie proces
- 5.3 Aanbevelingen
- 5.4 Vervolgonderzoek/ uitbreiding

BRONVERMELDINGEN

BEGRIPPENLIJST

BIJLAGE 1 DEFINITIESTUDIE

BIJLAGE 2 PILOT 1 – AANMELDING CLIENTSYSTEEM

BIJLAGE 3 PILOT 2 – STORINGMELDING

BIJLAGE 4 PILOT 3 – OPLOSSINGMELDING

BIJLAGE 5 PILOT 4 – FEEDBACK NAAR GEBRUIKER

BIJLAGE 6 PILOT 5 – OPLOSSINGHERKENNING

1 Inleiding

Kuiper & Burger is een Advies- en Ingenieursbureau. Binnen dit bureau is circa 15 milieuadviseurs werkzaam en 2 secretaresses. Door de jaren heen is er veel veranderd in de automatisering van Kuiper & Burger en is er voor elke medewerker een werkplek gerealiseerd. Doordat de automatisering geleidelijk is verlopen, is er veel onderscheid te maken in de verschillende werkplekken.

Het is niet onwaarschijnlijk binnen de ICT-sector dat bij veel gebruik van een onderdeel binnen het netwerk (bijvoorbeeld een werkstation) storingen optreden. Kuiper & Burger bezit niet de kennis om het netwerk te onderhouden, waardoor het netwerkbeheer is uitbesteed.

Het nadeel van extern beheer is dat het meestal prijzig is en als de beheerder niet ter plaatse is, kan het oplossen van een storing veel tijd in beslag nemen. Om deze reden zijn er veel beheertools op de markt beschikbaar; dit zijn software pakketten die de beheerder in staat stellen om het netwerk te beheren vanaf zijn eigen werkplek en waardoor hij niet steeds bij de klant aanwezig hoeft te zijn om een storing op te lossen.

Dergelijke beheertools zijn uiterst complex en bieden geen directe ondersteuning waar deze het hardst nodig is: bij de gebruiker. Hierdoor kan zelfs in de meest simpele situatie een storing niet worden verholpen zonder interactie van de beheerder. Dit maakt het beheren van een netwerk nog steeds duur en tijdrovend.

Kuiper & Burger heeft om deze reden gevraagd om een applicatie te ontwikkelen die meer is toegespitst op de ondersteuning van gebruikers. Hiermee zou bij een storing een hogere reactiesnelheid gerealiseerd kunnen worden omdat de gebruikers zelf de oplossing bieden. Dit is een voordeel voor zowel de gebruiker als de beheerder. Een groot deel van de storingen die de gebruiker meldt, zijn van dusdanig lage impact dat het oplossen ervan een lage prioriteit heeft bij de beheerder. Dat houdt in dat de gebruiker moet wachten op een oplossing van de beheerder. Als de gebruiker zelf al een oplossing of een "work-around" kan bieden, kan de beheerder verder met storingen die wel hoge prioriteit hebben en de gebruiker kan sneller verder met zijn/haar dagelijkse taak.

Het project dat bij Kuiper & Burger is uitgevoerd, had als doel om een Storing Management Systeem (SMS) te ontwikkelen die de bovengenoemde functionaliteit biedt. Dit moet plaatsvinden binnen de afstudeerperiode.

Dit document gaat in op de werkzaamheden en handelingen die de afstudeerder heeft uitgevoerd om het project tot een goed einde te brengen.

1.1 Aanleiding

Vóór de start van dit project heeft de afstudeerder vaker gewerkt in een klantrijke ICT-omgeving. Hierin gebeurt het vaak dat de klant zonder begeleiding zelfs geen simpele storing kan oplossen. Hierdoor kwam de afstudeerder steeds minder toe aan zijn overige taken. Om dit proces te vergemakkelijken, lijkt het een reële oplossing om een softwarematige oplossing te bedenken die de begeleiding bij simpele problemen overneemt.

Toen de afstudeerder daadwerkelijk een plek moest vinden om zijn afstudeerstage uit te voeren, kwam deze in contact met het bedrijf Kuiper & Burger. Dit bedrijf zat met het probleem dat hun netwerkbeheer omslachtig verliep en ze moesten telkens een extern bedrijf inschakelen om storingen te verhelpen. Dit gebeurde ook bij kleine storingen, wat het beheer een duur proces maakte.

Omdat het idee van de afstudeerder een groot deel van de storingen zou kunnen beheren binnen Kuiper & Burger, heeft het bedrijf toestemming verleend om met het afstudeertraject van start te gaan.

1.2 Het bedrijf

In samenwerking met aannemers en gemeenten verricht het bureau bodemonderzoeken om vast te kunnen stellen of bepaalde locaties schoon genoeg zijn om deze te kunnen herbruiken in de bouw. Van deze onderzoeken worden analyseresultaten opgenomen en rapporten opgesteld.

Iedereen binnen Kuiper & Burger hebben een eigen werkplek. Het is ook mogelijk om thuis te werken via het Internet.

De standaardwerkplekken zien er als volgt uit:

- Een computersysteem dat aangesloten is op het netwerk;
- Bureau met documentlades;
- Telefoon.

Naast de standaardwerkplekken is er één werkplek die twee systemen heeft. Eén van deze twee systemen is uitgerust om autocadtekeningen te kunnen maken met behulp van een digitaal tekenblad. Het ander systeem verzorgt de connectie met het hoofdlaboratorium om uitgevoerde onderzoeken te analyseren.

1.3 Vooronderstellingen

In het begin van het project had de afstudeerder de volgende vooronderstellingen:

- De softwarematige oplossing vergemakkelijkt het gebruik van cliëntsystemen;
- Een simplistische interface is direct een gebruiksvriendelijke interface;
- Een weboplossing zorgt ervoor dat het systeem bruikbaar is op alle systemen in het netwerk;
- Eenduidige ontwikkeldocumentatie versnelt het voorbereidende werk op het ontwikkelen.

1.4 Vooronderstellingen bedrijf

De werknemers bij Kuiper & Burger hadden als vooronderstellingen:

- De softwarematige oplossing lost alle problemen van het netwerk op;
- De oplossing heeft bijna geen gebruikersinteractie nodig;
- Het ontwikkeltraject voor een dergelijk product is kort.
- Het ontwikkeltraject houdt alleen programmeren in;
- Gebruiksgemak is eenvoudig te realiseren.

2 Analyse en probleemverkenning

Het aantal verschillende pc's inclusief besturingsysteem, de gebruikte software en de randapparatuur (printers en plotters, scanners, etc) loopt parallel aan de ontwikkeling van Kuiper & Burger. Op de verschillende pc's zijn verschillende besturingsystemen aanwezig variërend van win98 SE, Windows 2000 server, Windows XP pro en Windows XP home.

Bij bedrijfsnetwerken is het niet onwaarschijnlijk dat storingen voorvallen. Door Kuiper & Burger is de wens ontstaan om een betaalbare semi-automatische oplossing voor het beheer van het netwerk (inclusief de systemen en randapparatuur).

2.1 *Uitgangssituatie*

Het netwerk van Kuiper & Burger bestaat uit de volgende onderdelen:

- 14 Werkstations met Microsoft Windows 98;
- 5 Werkstations met Microsoft Windows XP pro;
- 1 Server met Microsoft Windows 2000 server;
- 1 Laptop met Microsoft Windows XP home;
- 1 HP Laserjet 5000;
- 1 HP Laserjet 4P
- 1 HP Deskjet 1220c;
- 1 HP Designjet 500.

Er is rekening gehouden dat er een verschil is in werkstations. Dit omdat Microsoft Windows 98 en Microsoft Windows XP een totaal verschillende architectuur hebben.

Het bedrijf dat systeembeheer uitvoert bij Kuiper & Burger maakt gebruik van de standaard Microsoft beheertools in samenwerking met de beheertool van het bedrijf Netsupport. Er is een bestaand Configuration Item (CI) database in de vorm van een Excel sheet.

2.2 Concurrentieanalyse

Op de softwaremarkt voor netwerkbeheer zijn een groot aantal producenten actief. Van deze producenten is er voor de concurrentieanalyse gekeken naar de volgende:

- HP, <http://www.hp.nl>;
- Gensys, <http://www.gensys.nl>;
- Hammer software, <http://www.hammer-software.com>;
- Netsupport, <http://www.netsupport-inc.com>;
- Microsoft, <http://www.microsoft.com/mom/evaluation/future/mom2005.asp>.

Aan de aangeboden software producten hangen de volgende voor en nadelen:

	HP	Gensys	Hammer	Microsoft	Netsupport
Gebruikersgericht	-	-	-	-	-
Ondersteuning vanaf de fabrikant	+	+	-	+	+
Nederlandse ondersteuning	+ -	+	-	+ -	+
Gebruikt WMI als basis	-	?	+	+	-
Functioneel binnen alle Windows platformen	+	?	+*	+ *	-
Beheerder essentieel bij een storing	+	+	+	+	+
Legenda:					
Volledig ondersteund	+				
Niet ondersteund	-				
Ondersteuning na installatie WMI core applicatie	*				
Onbekend	?				

Figuur 1 Voor- en nadelenschema concurrentie

Wat duidelijk te zien is bij de vier onderzochte concurrenten is dat er geen aandacht is besteed aan een applicatie dat meer gebruikersgericht is. Een beheerder zal altijd bij een storing ondersteuning moeten bieden ook al is de oplossing zeer eenvoudig.

Een ander veel voorkomend nadeel is dat niet alle beheertools gebruik maken van de Windows Management Interface (WMI). Dit is de beheerbasis van Microsoft Windows systemen vanaf Windows 98 SE. Dit maakt beheerapplicaties koppelbaar aan elkaar, waarmee een beheerder uitgebreidere beheeromgevingen kan creëren. Deze flexibiliteit resulteert in een meer directe ondersteuning naar de gebruikers omdat hiermee voor elke soort netwerksituatie een beheersituatie te realiseren is.

2.3 Aanwezige kennis

Bij Kuiper & Burger speelt al een tijd het probleem dat het inzicht dat ze hebben in het beheer van hun eigen netwerk beperkt is. Dit komt door de het gebrek aan kennis op het gebied van ICT en door de onduidelijkheid van veel storingsmeldingen. Om deze reden is een extern bedrijf ingeschakeld om het beheer van het netwerk te verzorgen. Dit is een oplossing die veel geld kost en het verhelpen van storingen is hierdoor tijdsgebonden.

Omdat het bijhouden van bestaande oplossingen door het externe bedrijf gebeurt, wil Kuiper & Burger een informatiesysteem dat ook benaderbaar is door medewerkers van Kuiper & Burger. Het idee van een dergelijk systeem kon niet uitgewerkt worden omdat hiervoor de kennis ontbrak binnen Kuiper & Burger.

2.4 Doelgroep

Het SMS is bedoeld voor bedrijven en beheerders die zich niet druk willen maken om kleine storingen binnen een bedrijfsnetwerk.

Het SMS is toepasbaar binnen ITIL-(Information Technology Infrastructure Library) normen omdat het een deel van de service support set op zich neemt. De tool speelt vooral in op incidentmanagement en problem management. Het houdt bij welke storingen zich hebben voorgedaan en houdt hiervoor de oplossingen en "work-arounds" bij.

2.5 Doelstellingen

Voor het project zijn de volgende eindresultaten geformuleerd:

- Het SMS ondersteunt de gebruiker in zijn/haar dagelijkse taak door een oplossing te bieden in geval van een storing;
- Het SMS biedt voor een storing een éénduidige en geteste oplossing. De oplossing moet geboden zijn binnen de gestelde periode die afhankelijk is van de prioriteit van de storing;
- De melding van de storingen vindt plaats op een centraal meldpunt dat altijd beschikbaar is. Dit centrale meldpunt zal zich bevinden op een Intranet welke toegang biedt aan alle medewerkers;
- Elke storing krijgt een categorie, prioriteit, status, begindatum, einddatum en een uniek id toegewezen om de geschiedenis en herhaling van storingen bij te houden;
- Indien een storing zichzelf herhaalt dan vindt er direct een melding plaats bij de gebruiker en bij de beheerder;
- Zolang de storing niet opgelost is geeft het SMS periodiek een melding met de huidige status van de storing.

3 Uitvoering

Dit hoofdstuk gaat in op het verloop van de uitvoering van het project. Welke aanpak is gehanteerd en welke hulpmiddelen en technieken zijn gebruikt.

3.1 Globale aanpak

3.1.1 Methodieken en strategieën

Hieronder volgt een opsomming van methodieken en strategieën die staan vastgesteld voor dit project:

- Ontwikkelmethodiek: Iterative Application Design (IAD);
- Ontwikkelstrategie: Incrementeel opleveren;
- Modellerings techniek: Unified Modelling Language (UML);
- Teststrategie: Think Aloud.

Hoofdstukken 3.1.2 t/m 3.1.5 gaan dieper in op de keuze van elk methodiek en strategie.

3.1.2 Ontwikkelmethodiek

Bij de start van het project zijn er de drie meest gehanteerde in overweging genomen om te gebruiken als handvat voor de rest van het project. De ontwikkeltechnieken zijn:

- SDM;
- RAD;
- IAD.

Van deze drie bood IAD de meeste vrijheid en de meeste structuur voor een ontwikkeltraject. Dit omdat SDM een vast traject heeft met weinig gebruikersinteractie, hetgeen niet wenselijk is omdat dan aan het einde van het volledige ontwikkeltraject kan worden bepaald of het systeem conform de eisen van de gebruikers zijn. RAD boekt pas resultaten aan het einde van het project hetgeen als nadeel wordt beschouwd omdat dan de opdrachtgever geen inzicht heeft in welke resultaten al zijn geboekt.

IAD bied een ontwikkeltraject dat bestaat uit drie fases:

- Definitiestudie;
- Pilotontwikkeling;
- Invoering.

De fases zijn iteratief uit te voeren. Dit houdt in dat aan het eind van elke fase kan worden bepaald of deze moet worden herzien of dat deze afgerond is. Het voordeel hiervan is dat een ontwikkeltraject open staat voor vooruitgang. Als bijvoorbeeld de techniek die gebruikt is in een pilot verouderd is kan deze pilot afzonderlijk worden herontwikkeld in een nieuwe iteratieslag. Daarnaast is het proces meer inzichtelijk voor de gebruikers binnen het bedrijf, er wordt een stap voor stap methodiek gehanteerd.

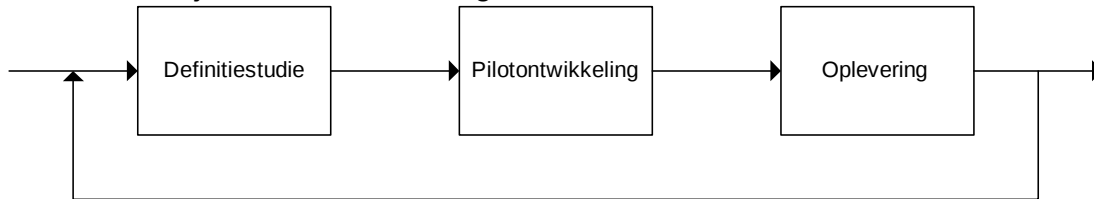
3.1.3 Ontwikkelstrategie

IAD biedt vier (hoofd)iteratiestrategieën waarin ontwikkeld kan worden:

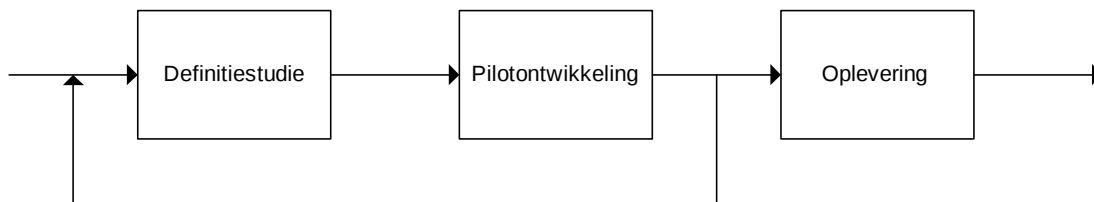
- Evolutionair ontwikkelen;
- Incrementeel opleveren;
- Incrementeel ontwikkelen;
- Big-bang invoeren.

Evolutionair ontwikkelen en Big-bang invoeren hebben beide de overeenkomst dat deze twee de definitiestudie bij elke iteratieslag opnieuw uitvoeren om opnieuw de systeemeisen vast te stellen. Dit was overbodig omdat voor het project de systeemeisen al vastgesteld waren. Hierdoor zijn deze iteratiestrategieën niet gebruikt voor het project.

Schematisch zijn deze twee als volgt af te beelden:



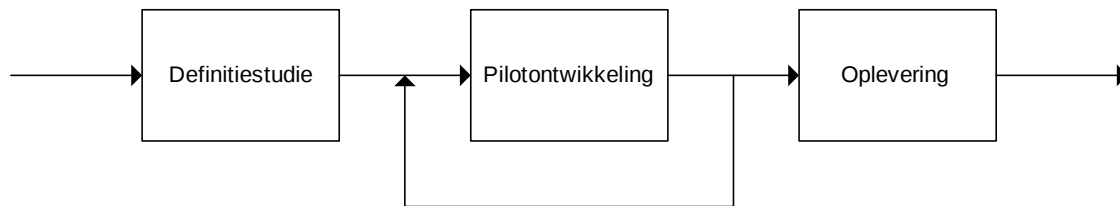
Figuur 2 ontwikkelstrategie "Evolutionair ontwikkelen"



Figuur 3 ontwikkelstrategie "Big-bang invoeren"

Deze situatie biedt ruimte voor de ontwikkelstrategie "Incrementeel ontwikkelen". Dit houdt in dat per pilot de pilotontwikkeling plaatsvindt en dat de uitvoering van de definitiestudie en oplevering maar één keer gebeurt. Het nadeel hierbij is dat de opdrachtgever pas resultaten krijgt aan het einde van het project en de opdrachtnemer dan ook pas feedback krijgt. Aangezien het SMS gebruikersgericht is, is het niet meer als logisch dat de gebruiker zo betrokken mogelijk is bij de ontwikkeling. Hierdoor is ook de ontwikkelstrategie "Incrementeel ontwikkelen" minder geschikt.

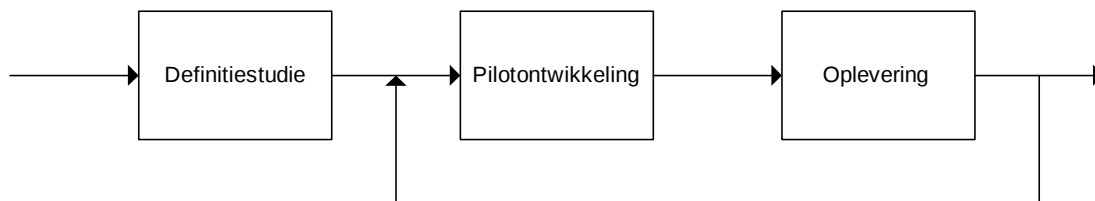
Deze ontwikkelstrategie is schematisch als volgt uit te beelden:



Figuur 4 Ontwikkelstrategie "Incrementeel ontwikkelen"

Uit de verschillende ontwikkelstrategieën die IAD omvatte is gekozen voor "incrementeel opleveren". Dit houdt in dat per pilot de pilotontwikkeling en de invoering plaatsvindt en de uitvoering van de definitiestudie maar één keer gebeurt. Er is gekozen voor deze strategie omdat een groot deel van de systeemeisen al bekend waren en het dus niet nodig is om de definitiestudie vaak te herzien. Tevens gaat het hier om een klein informatiesysteem en is er geen behoefte om de systeemaspecten tot in de details uit te werken. Het nadeel hiervan is wel dat de opstarttijd langer is en de voorbereidende werkzaamheden de meeste tijd vergen.

De gekozen ontwikkelstrategie is schematisch als volgt uit te beelden:



Figuur 5 IAD ontwikkelstrategie "Incrementeel opleveren"

3.1.4 Modellerings techniek

Voor ontwikkeling van het SMS zijn er twee belangrijke modellerings technieken te onderscheiden:

- Universal Modelling Language (UML);
- Yourdon.

Yourdon biedt uitgebreide modellen voor het specificeren van een systeem, maar aangezien alle modellen van elkaar afhankelijk zijn ontstaat er een enorme overhead aan documentatie die mogelijk niet van belang is. Tevens is er extra werk nodig om deze modellen consequent aan elkaar te houden. Voor een project met een “krappe” planning vergt het onderhouden van de documentatie te veel tijd.

UML bood de vrijheid om te kiezen welke modellen van belang waren. Hierdoor kon er gekozen worden voor de meest relevante modellen voor dit project. Bijbehorende modellen die geen extra toegevoegde waarde bieden kunnen achterwegen gelaten worden.

Het enige probleem was dat er geen goed model was voor een weboplossing dus hierdoor lijkt het binnen het klassendiagram dat de gebruikte webpagina's niet object georiënteerd zijn ontwikkeld.

Uit UML zijn drie modellen gebruikt om een globale schets te geven van het SMS:

- Domeinmodel;
- Klassendiagram;
- Use-case diagram.

UML bied ook modellen voor de toestanden waarin het systeem kan verkeren, de volgorde van stappen die het systeem doorloopt in tijd en welke activiteiten er worden uitgevoerd. Voor deze modellen is niet gekozen, omdat het onderhoud van deze drie modellen van elkaar afhankelijk zijn en veel tijd innemen. In plaats daarvan is gekozen voor het “petrinet”-model.

Een petrinet is oorspronkelijk bedoeld om de workflow binnen een bedrijf te definiëren. Een petrinet geeft aan waar knelpunten binnen de communicatie bevinden, in welke toestand een element zich verkeerd, wat de volgorde van stappen is en welke activiteiten binnen een workflow worden uitgevoerd. Hierdoor was dit model ook toepasbaar bij de ontwikkeling van het SMS en nam het werk over van de overige UML modellen en bood één controle meer.

3.1.5 Teststrategie

Het SMS is ontwikkeld in samenwerking met de gebruikers. Om er zeker van te zijn dat het systeem bruikbaar is voor deze gebruikers zijn er usability tests uitgevoerd.

Normaal bestaat een usability test uit twee van de volgende tests:

- Think Aloud;
- SUMI (Software Usability Measurement Inventory);
- WAMMI (Website Analysis and Measurement Inventory);
- CWT (Cognitive Walkthrough);
- GOMS (Goals, Operators, Methods and Selection rules);
- Heuristic Evaluation.

Bij dit project bestond deze uit alleen de Think Aloud test. Dit is de snelste test waar relevante gebruikersinvoer wordt verkregen. De reden dat er geen twee tests uitgevoerd zijn, is dat er te weinig gebruikers zijn binnen Kuiper & Burger om de tests goed uit te voeren.

Goede testresultaten zijn pas te meten bij meer als 50 gebruikers, omdat er dan een goede verspreiding van gebruikerssoorten is. De afstudeerder vond het een tijdverspilling om uitgebreide tests uit te voeren om dan uiteindelijk onbetrouwbare testresultaten te krijgen.

Er was ook de mogelijkheid om de tests in een lab uit te voeren om zo nauwkeurigere resultaten te behalen aan de hand van meerdere gebruikers. Het SMS is bedoeld om alleen gebruikt te worden binnen Kuiper & Burger. De gebruikers van dit netwerk bieden als enige relevante inbreng voor de usability van het systeem.

3.1.6 Tijdplanning

Onderstaand (uren)planning geeft aan hoeveel uren er gepland waren voor het project.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9
	09-feb	16-feb	23-feb	01-mrt	08-mrt	15-mrt	22-mrt	29-mrt	05-apr
Plan van Aanpak	40	10							
Definitiestudie		20	35	10					
Intranet situatie				25	35	35	35	30	
Meldingssysteem								5	30
Oplevering (opdrachtgever)									
Invoering									
Afstudeer presentatie									
Voortgangsverslag		5	5	5	5	1	3	3	
Overige werkzaamheden voor eindverslag									
Procesverslag	5	10	8	8	6	6	6	6	7
Gesprek met begeleider					2				
Totaal	45	45	48	48	48	42	44	44	37
	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
	12-apr	19-apr	26-apr	03-mei	10-mei	17-mei	24-mei	31-mei	07-jun
Plan van Aanpak									
Definitiestudie									
Intranet situatie									
Meldingssysteem	35	40	35						
Oplevering (opdrachtgever)				20	10	10			
Invoering				15	25	25			
Afstudeer presentatie									3
Voortgangsverslag									
Overige werkzaamheden voor eindverslag									
Procesverslag	7	2	4	4	4	4			
Gesprek met begeleider		7	6	6	8	8	8	8	
Totaal	42	49	45	45	47	47	8	8	3

Figuur 6 (Uren)planning

3.1.7 Pilotplanning

Onderstaand (uren)planning geeft weer hoe de pilots zijn opgedeeld voor dit project. Pilot 1 hoort in de globale planning thuis bij "Intranet situatie" en pilot 2 t/m 4 bij "Meldingsysteem".

	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
	1-mrt	8-mrt	15-mrt	22-mrt	29-mrt	5-apr	12-apr	19-apr
Pilot 1 - Aanmelding clientsysteem	35	35	35	35	35	35	35	35
Pilot 2 - Storingmelding								
Pilot 3 - Oplossingmelding								
Pilot 4 - Feedback naar gebruiker								
Pilot 5 - Oplossingherkenning								
Totaal	35	35	35	35	35	35	35	35
	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	
	26-apr	3-mei	10-mei	17-mei	24-mei	31-mei	7-jun	
Pilot 1 - Aanmelding clientsysteem								
Pilot 2 - Storingmelding	35							
Pilot 3 - Oplossingmelding		35						
Pilot 4 - Feedback naar gebruiker			35					
Pilot 5 - Oplossingherkenning				35	35	35	35	
Totaal	35	35	35	35	35	35	35	

Figuur 7 Pilotplanning

3.2 Gebruikte hulpmiddelen en technieken

3.2.1 Werkomgeving

Het ontwikkelsysteem dat is gebruikt voor dit onderzoek bestond uit de volgende hardware:

- Pentium III 999;
- 512 MB geheugen;
- 20 GB harddisk.

Voor de ontwikkeling zijn de volgende softwarepakketten gebruikt:

- Microsoft Windows 2000 met service pack 4;
- Microsoft Visual Studio .Net voor softwareontwikkeling;
- Microsoft Word 2003;
- Microsoft Visio 2003;
- Microsoft Internet Information Services;
- Editplus.

Het besturingsysteem van het workstation waar de uitvoering op plaatsvindt, is Microsoft Windows 2000. Dit is gedaan omdat de ontwikkeltool "Microsoft Visual Studio .Net" een NT omgeving vereist. Er had gekozen kunnen worden voor een andere ontwikkeltool maar dit is het enige .Net omgeving die direct de modellen van UML ondersteund.

Om toch nog te kunnen testen of het SMS werkt op de rest van de workstations, zijn voor testdoeleinden twee workstations gebruikt, één met Microsoft Windows XP en één met Microsoft Windows 98.

3.2.2 Hardware

Het systeem moet de hele ICT-infrastructuur ondersteunen van Kuiper & Burger. Maar omdat de applicatie ontwikkeld is voor een Windows 32 bit omgeving is er één beperking gesteld aan de hardware: Deze moet in staat zijn om minstens Microsoft Windows 98 te kunnen draaien. Dit stelt de programmeur in staat om met de standaard API van deze omgeving te kunnen ontwikkelen.

3.2.3 Programmeeromgeving

Om het SMS te kunnen ontwikkelen is er de keuze tussen een aantal mogelijke programmeer situaties, deze zijn:

- C++ of C# client/server model in een niet-webomgeving;
- C++ client/server model in een CGI-webomgeving;
- C++ of C# in samenwerking in een PHP-webomgeving;
- C# client/server model in een ASP-webomgeving.

In de volgende hoofdstukken staat beschreven wat de afwegingen zijn geweest om voor een bepaalde omgeving te kiezen.

3.2.3.1 C++ of C# client/server model in een niet-webomgeving

Het project heeft een applicatie nodig waarvoor verschillende besturingsystemen als ondergrond dienden. Omdat de architectuur van elk besturingsysteem anders is, dient in de eerste situatie per besturingsysteem een afzonderlijke applicatie ontwikkeld te worden die de gevraagde functionaliteit biedt. Hiermee ontstaat de kans dat er dubbel werk gedaan wordt voor dezelfde functionaliteit en dat zou verloren tijd en moeite zijn.

3.2.3.2 C++ client/server model in een CGI-webomgeving

Het voordeel van het gebruik van één programmeertaal, zoals in situatie twee, is dat je hiervoor eenduidige ontwikkeldocumentatie kunt maken. Het nadeel aan C++ in een CGI-webomgeving is dat hiervoor nooit een echte standaard is ontwikkeld. De verscheidenheid in methodes om de gegevens van de webclient te verkrijgen is te groot om hier een stabiele ondergrond op te bouwen. Om deze reden is er voor gekozen om naar een andere oplossing te zoeken.

3.2.3.3 C++ of C# in samenwerking PHP in een PHP-webomgeving

De derde situatie lijkt initieel de beste omdat PHP al een webgeoriënteerde taal is, maar PHP bied geen mogelijkheid om van het cliëntsysteem de systeemgegevens te verkrijgen. Hierdoor is dus een tweede taal nodig om dit te realiseren (zoals C++ of C#). Het nadeel hiervan is dat voor beide talen afzonderlijke ontwikkeldocumentatie nodig is en omslachtige procedures nodig zijn om de talen met elkaar te kunnen laten communiceren. Hierdoor zal het ontwikkeltraject meer tijd innemen waardoor ook deze methode niet is gekozen.

3.2.3.4 C# client/server model in een ASP-webomgeving

Tot slot blijft de keuze over om een taal te gebruiken welke ontwikkeld is om in meerdere situaties te fungeren, zowel in een webomgeving als in een cliëntomgeving, dit is onder andere de taal C#. Het voordeel hierbij is dat voor de interactie tussen web en applicatie vaste methodes zijn gedefinieerd en slechts voor één taal ontwikkeldocumentatie nodig is. Het enige nadeel voor dit project is dat dit een taal is die de afstudeerder nog moet , hetgeen als minst nadelig is beschouwd ten opzichte van de nadelen van de overige programmeeroplossingen.

3.3 Aanpak definitiestudie

In de fase definitiestudie zijn de doelen van het systeem geanalyseerd, evenals de beperkingen. Hiermee zijn aan de hand van een lijst met systeemeisen de pilots opgezet en geprioriteerd.

3.3.1 Plan van aanpak

Conform IAD is er aan het begin van iedere fase een plan van aanpak opgesteld. Het plan van aanpak beschrijft de inhoud van elke IAD fase.

3.3.2 Ontwikkelscenario

Het tweede onderdeel dat gedefinieerd is binnen IAD is het ontwikkelscenario. Het doel hierachter is om de reikwijdte en de context van het ontwikkelproject vast te stellen. Deze fase is optioneel volgens IAD en hoeft alleen uitgevoerd te worden wanneer de gebruikers onbekend zijn met het ontwikkeltraject of wanneer de technische structuur nog niet door de opdrachtgever is bepaald.

De gebruikers binnen Kuiper & Burger hebben nog nooit een ontwikkeltraject ervaren waarbij IAD is gebruikt. Om de gebruikers meer inzicht te bieden in het ontwikkeltraject is de keuze gemaakt om dit onderdeel van IAD uit te voeren.

3.3.3 Pilotplan-workshop

Binnen de pilotplan-workshop worden de systeemeisen vastgesteld, het systeemconcept opgesteld, voorgaande pilots geëvalueerd, de technische structuur bepaalt, de organisatorische inrichting vastgesteld en het pilotplan gemaakt. IAD omschrijft dit als een essentieel onderdeel van de ontwikkelcyclus. De workshop verzorgt een afbakening van het ontwikkelproces.

Normaal gesproken bestaat een workshop uit een brainstorm sessie met het ontwikkelteam. Het ontwikkelteam voor de workshop bestond in dit geval uit één persoon, de afstudeerder. Hierdoor konden de activiteiten binnen de workshop snel worden uitgevoerd.

De resultaten van deze sessie zijn geverifieerd door de gebruikers. Hiermee is het doel van de pilotplan-workshop bewerkstelligd zonder de langdurige procedures en overlegmomenten van deze workshop te volgen. Hierna in de paragrafen 3.3.3.1 t/m 3.3.3.6 is de uitvoering van de activiteiten van de pilotplan-workshop beschreven.

3.3.3.1 Evaluatie pilot

Binnen dit onderdeel van de workshop wordt de ervaring met voorgaande pilots getoetst in relatie tot de actuele systeemeisen en het oorspronkelijke pilotplan. De uitvoering van dit onderdeel vindt plaats tijdens elke iteratieslag van de definitiestudie omdat hier pas voorgaande pilots geëvalueerd worden.

De ontwikkelstrategie “Incrementeel opleveren” is gebruikt, hierdoor heeft de definitiestudie nooit een iteratieslag meegemaakt en is dit onderdeel nooit uitgevoerd.

3.3.3.2 Definitie systeemeisen

Het doel van deze workshopactiviteit is het opstellen en actualiseren van een geprioriteerde lijst van systeemeisen. Deze activiteit is noodzakelijk omdat hier de reikwijdte van het project wordt doordrongen.

De systeemeisen vormen het belangrijkste referentiemateriaal bij de uitvoering het ontwikkeltraject. Hiermee kan getoetst worden in hoeverre de ontwikkelde pilot aan de afgesproken eisen heeft voldaan.

3.3.3.3 Systeemconcept

In de workshop beschrijft het systeemconcept de weergave van de oplossing N de systeemeisen. De modellering wordt gedaan vanuit het gezichtspunt van de gebruiker.

Het beschrijft de taken die door de gebruiker binnen een bepaald bedrijfsproces worden uitgevoerd. Tevens zijn de operationele kenmerken van het systeem vastgesteld.

3.3.3.4 Technische structuur

Deze activiteit binnen de workshop toetst de technische aspecten rond het project, om een duidelijk beeld te hebben welke hard- en software er nodig is voor de pilotontwikkeling. Aangezien de technische aspecten al vastgesteld zijn in het globale plan van aanpak is deze activiteit niet uitgevoerd.

3.3.3.5 Organisatorische inrichting

Deze op één na laatste activiteit binnen de workshop stelt vast wat de impact was van voorgaande pilots op de organisatorische inrichting van het bedrijf.

De definitiestudie is maar één keer uitgevoerd en hiervoor zijn er nog geen voorgaande pilots geweest, dus kon deze activiteit niet worden uitgevoerd.

3.3.3.6 Pilotplan

Het opstellen van het pilotplan is de laatste en meest belangrijke activiteit binnen de workshop. Binnen IAD is het pilotplan de rode draad door het ontwikkeltraject. Het bepaalt welke pilots er zijn gedefinieerd en in welk tijdbestek deze worden uitgevoerd. Het specificeert globaal wat de pilots inhouden.

Tijdens het opstellen van het pilotplan worden de voorgaande documenten van de workshop in acht genomen.

Door middel van de systeemeisen kan worden bepaald welke pilot de hoogste prioriteit heeft en in welke volgorde de pilots zullen worden uitgevoerd.

3.4 Aanpak pilotontwikkeling

Het doel van deze fase is de ontwikkeling van een pilot. De reikwijdte van die pilot is gedefinieerd in de fase definitiestudie en weergegeven in het pilotplan. Een pilot biedt een beperkte maar operationele subset van de functionaliteit van het uiteindelijk beoogde systeem.

3.4.1 Plan van aanpak

Dit document beschrijft de aanpak van de pilot. IAD definieert dit als een verplichte activiteit, omdat dit een handvat is voor het uitvoeren van een pilot.

3.4.2 Pilotontwerpworkshop

Deze workshop wordt uitgevoerd om de globaal-functionele structuur pilot, globaal-technische structuur pilot, globaal organisatorische inrichting en het pilotontwikkelplan te bepalen.

Ook deze workshop bestaat uit een brainstormsessie met het ontwikkelteam. Waarbij elk lid van het ontwikkelteam betrokken is met de uitvoering van elk onderdeel van de pilotontwikkeling.

De onderdelen van deze workshop worden besproken in de hoofdstukken 3.4.2.1 t/m 3.4.2.4.

3.4.2.1 Globaal-functionele structuur pilot

Het doel van deze workshopactiviteit is het bepalen van de functionele inhoud van de pilot.

De weergave van de oplossing die bepaald is in het systeemconcept is tijdens deze activiteit verder verfijnd. De functionele specificatie documenteert onder meer de vastgestelde events, de manier waarop deze afgehandeld zijn en de betrokken objecten uit het domeinmodel.

Dit onderdeel is niet van belang wanneer de weergave van de oplossing al specifiek genoeg is. Dit was niet het geval na de definitiestudie en is daarom uitgevoerd.

3.4.2.2 Globaal-technische structuur pilot

Deze activiteit binnen de workshop heeft als doel het specificeren en valideren van de technische aspecten rond de pilot in ontwikkeling.

Dit onderdeel werkt subonderdelen van het domeinmodel uit naar de werkelijkheid voor deze pilot. Tevens bepaalt het hoe de database opslagmodel eruit komt te zien en schetst het netwerkcommunicatiemodel.

De globaal-technische structuur pilot is van belang om per pilot uit te voeren. Dit komt omdat elk van de pilots een eigen uitwerking van het domeinmodel heeft en een eigen database opslagmodel.

3.4.2.3 Globaal-organisatorische inrichting

De globaal-organisatorische inrichting van de pilot stelt de taken vast binnen een ontwikkelteam.

Deze activiteit van de workshop kan alleen worden uitgevoerd als het team bestaat uit meerdere personen. Het ontwikkelteam bestaat maar uit één persoon en dus is deze activiteit achterwege gelaten.

3.4.2.4 Pilotontwikkelplan

Het pilotontwikkelplan beschrijft op basis van de gemaakte globaal functionele, technische en organisatorische specificaties hoe de pilot gaat worden gebouwd.

Zoals het pilotplan de rode draad is voor het gehele ontwikkeltraject is het pilotontwikkelplan dit voor elk individuele pilot. Het pilotontwikkelplan verdeelt elke pilot op in softwarebouweenheden die elk afzonderlijk achtereenvolgend dan wel parallel ontwikkeld kunnen worden.

3.4.3 Softwarebouweenheden

De door het pilotplan omschreven bouweenheden worden ontworpen, gebouwd en geïntegreerd in het SMS. Samen vormen ze het eindproduct van een pilotontwikkeling.

3.4.4 Beoordeel en test pilotdeel

Aan het einde van elke pilotontwikkeling vindt er een test en beoordeling plaats. Hier zijn de resultaten van de pilot getoetst met de voor de pilot opgestelde systeemeisen. Hierna is de pilot afgerond en heeft de invoering plaatsgevonden.

3.5 Aanpak invoering

De fase invoering zorgt ervoor dat de pilot operationeel gemaakt wordt in de gebruikersomgeving.

3.5.1 Plan van aanpak

Het plan van aanpak voor de invoering wordt binnen IAD omschreven als verplicht onderdeel. Tijdens de pilotontwikkeling is ook omschreven hoe de invoering plaatsvindt. Het was overbodig om dit twee keer te noemen en dit plan van aanpak is niet gemaakt.

3.5.2 Pilotinvoering

Tijdens deze activiteit is het resultaat van elke pilot ingevoerd en geïnstalleerd. Als hier problemen zijn ondervonden werd dit genoteerd een meegenomen naar de volgende pilot. Indien er geen pilot meer in ging op het softwareonderdeel werd de pilot herzien om de problemen te verhelpen.

Deze activiteit bestaat uit twee handelingen. Deze worden beschreven in paragrafen 3.5.2.1 en 3.5.2.2.

3.5.2.1 Pilotacceptatietest

De acceptatie van de pilot behoort gewoonlijk tot de verantwoordelijkheden van de opdrachtgever. Het doel van de acceptatie is, formeel vast te stellen dat de pilot voldoet aan de gestelde systeemeisen en de daaruit afgeleide functionele specificaties.

3.5.2.2 Verzorg opleidingen

Het doel van deze activiteit is het verzorgen van de opleidingen met betrekking tot de in te voeren pilot voor de vastgestelde doelgroepen.

Omdat het eindproduct gebruiksvriendelijk is geacht zijn er geen volledige opleidingen verstrekt. Bij pilots waar gebruikersinteractie vereist was is kort uitgelegd hoe het SMS functioneerde.

3.5.3 Pilotoverdracht

Nadat de pilot geaccepteerd door de opdrachtgever kan de pilot indien nodig overgedragen worden aan organisaties en instanties die zich bezighouden met onderhoud en verwerking van applicaties.

Dit project is dusdanig kleinschalig, er is weinig onderhoud nodig. De verwerking en onderhoud wordt gedaan door het beherende bedrijf.

3.5.4 Feedback gebruikers

Aan de hand van het verzamelen van feedback van gebruikers kan een basis ontstaan voor de volgende iteratie in de ontwikkelcyclus.

Deze feedback wordt verzameld door middel van de eerder vastgestelde usability tests.

4 Resultaten

Dit hoofdstuk beschrijft aan de hand van de ontwikkeldocumentatie (opgenomen in de bijlagen) de resultaten van het project.

4.1 Definitiestudie

4.1.1 Plan van aanpak

Het eerste document binnen de definitiestudie (Bijlage 1) is het plan van aanpak. Het document bestaat uit de volgende hoofdstukken:

- Reikwijdte definitiestudie, binnen dit hoofdstuk worden de doelstellingen en de afbakening van de definitiestudie vastgesteld;
- Resultaten definitiestudie, dit hoofdstuk omschrijft welke resultaten worden geboekt door deze fase.
- Kwaliteitsborging, in dit hoofdstuk zijn de voorwaarden die de kwaliteit van de definitiestudie toetsen;
- Planning definitiestudie, deze planning geeft het tijdsbestek weer waarin de definitiestudie moet zijn uitgevoerd.

4.1.2 Ontwikkelscenario

Het ontwikkelscenario vormt een document waarin de globale pilotstrategie is vastgesteld. Verder zijn hier de strategieën rond hergebruik, externe acquisitie, globale tests en beheer bepaalt.

4.1.3 Pilotplan workshop

Uit de pilotplan-workshop zijn de volgende drie documenten ontstaan:

- Systeemeisen;
- Systeemconcept;
- Pilotplan.

4.1.3.1 Definitie systeemeisen

De systeemeisen die zijn vastgesteld binnen de definitiestudie zijn:

- Basissysteemeisen, dit zijn de handelingen die het systeem minimaal moet kunnen verrichten;
- Interface-eisen, de gebruiksvriendelijkheid van het SMS wordt aan de hand van deze eisen bepaald;
- Integriteitseisen, deze eisen waarborgen de betrouwbaarheid van het systeem;
- Performance-eisen, om een richtlijn te hebben voor de performance, vindt er een toetsing plaats aan de hand van deze eisen;
- Operationele eisen, aan de hand van deze eisen kan aan het eind van het ontwikkeltraject bepaald worden of het eindproduct ingevoerd kan worden.

4.1.3.2 Systeemconcept

Dit document beschrijft globaalinhoudelijk de structuur van de pilots (door middel van prototypering), de actors, de globale events, de globale domein-objecten en de systeeminterfaces.

4.1.3.3 Pilotplan

Het SMS is binnen het pilotplan opgedeeld in de volgende pilots:

Voor de intranet situatie:

- Pilot 1 – Aanmelding clientsysteem, deze pilot stelt de gebruiker in staat zijn workstation aan te melden bij de database.

Voor het meldingsysteem:

- Pilot 2 – Storingmelding, hier kan de gebruiker melding doen van een voorgevallen storing, het systeem is nog niet in staat hierop een oplossing te bieden;
- Pilot 3 – Oplossingmelding, bij een storingmelding kan de beheerder een oplossing bieden;
- Pilot 4 – Feedback naar gebruiker, deze pilot koppelt pilots 1 en 2 aan elkaar door de gebruiker en de beheerder op de hoogte te stellen van de meldingen;
- Pilot 5 – Oplossingherkenning, in geval van bestaande oplossingen kan het systeem deze bieden zodat de gebruiker deze kan toepassen zonder hulp van de beheerder.

4.2 Pilot 1 Aanmelding clientsysteem

4.2.1 Plan van aanpak

Aangezien het plan van aanpak van elke pilot nagenoeg gelijk zou zijn is deze zo opgesteld dat het concept steeds kon worden herbruikt. Het enige van belang in elk plan van aanpak was de planning per pilot.

4.2.2 Pilotontwerp-workshop

Tijdens de pilotontwerp-workshop zijn de drie documenten voor pilot 1 (Bijlage 2) opgesteld:

- Globaal-functionele structuur;
- Globaal-technische structuur;
- Pilotontwikkelpplan.

4.2.2.1 Globaal-functionele structuur

De globaal-functionele structuur van pilot 1 bestond uit een prototype voor het systeemaanmeldingscherm en het domeinmodel.

Binnen het domeinmodel zijn de volgende entiteiten opgenomen:

- Hardware;
- Software;
- Onderdeel;
- Systeem.

4.2.2.2 Globaal-technische structuur

Binnen de globaal-technische structuur van pilot 1 is de technische reikwijdte, technisch opslagmodel, het communicatiemodel en de interfaces met andere systemen beschreven.

De technische reikwijdte stelt vast welke onderdelen van het clientsysteem ondersteund worden door deze pilot.

Het technisch opslagmodel werkt de entiteiten, welke genoemd zijn in de globaal-functionele structuur, uit zodat deze in de database kunnen worden opgenomen.

Het communicatiemodel beschrijft aan de hand van een petrinet de communicatie zoals deze verloopt binnen de applicatie voor pilot 1.

4.2.2.3 Pilotontwikkelplan

Het pilotontwikkelplan omschrijft de bouweenheden van pilot 1. Deze bestaat uit het invoeren van het domeinmodel naar de database, het ontwikkelen van de clientcontrol en het ontwikkelen van de webclient.

De clientcontrol verzorgt de interactie met het clientsysteem en de webclient. Hiermee kan systeemgegevens worden opgehaald en opgeslagen in de database.

4.3 Pilot 2 storingmelding

4.3.1 Pilotontwerp-workshop

Tijdens de pilotontwerp-workshop zijn de drie documenten voor pilot 2 Bijlage 2(Bijlage 3) opgesteld:

- Globaal-functionele structuur;
- Globaal-technische structuur;
- Pilotontwikkelplan.

4.3.1.1 Globaal-functionele structuur

De globaal-functionele structuur van pilot 2 bestond uit een prototype voor het systeemaanmeldingscherm en het domeinmodel.

Binnen het domeinmodel zijn de volgende entiteiten opgenomen:

- Systeem;
- Incident.

4.3.1.2 Globaal-technische structuur

Binnen de globaal-technische structuur van pilot 2 is de technische reikwijdte, technisch opslagmodel, het communicatiemodel en de interfaces met andere systemen beschreven.

De technische reikwijdte stelt vast welke onderdelen van het clientsysteem ondersteund worden door deze pilot.

Het technisch opslagmodel werkt de entiteiten, welke genoemd zijn in de globaal-functionele structuur, uit zodat deze in de database kunnen worden opgenomen.

Het communicatiemodel beschrijft aan de hand van een petrinet de communicatie zoals deze verloopt binnen de applicatie voor pilot 2.

4.3.1.3 Pilotontwikkelplan

Het pilotontwikkelplan omschrijft de bouweenheden van pilot 2. Deze bestaat uit het invoeren van het domeinmodel naar de database, het ontwikkelen van de clientcontrol en het ontwikkelen van de webclient.

Omdat pilot 2 ook informatie van het clientsysteem nodig had werd de clientcontrol van pilot 1 herbruikt en aangepast. Hetgeen wat nodig was, was het clientsysteemnaam. Hiermee kon aan de hand van de database worden bepaald welk systeem was aangemeld.

4.4 Pilot 3 Oplossingmelding

4.4.1 Pilotontwerp-workshop

Tijdens de pilotontwerp-workshop zijn de drie documenten voor pilot 3 (Bijlage 4) opgesteld:

- Globaal-functionele structuur;
- Globaal-technische structuur;
- Pilotontwikkelplan.

4.4.1.1 Globaal-functionele structuur

De globaal-functionele structuur van pilot 3 bestond uit een prototype voor het systeemaanmeldingscherm en het domeinmodel.

Binnen het domeinmodel zijn de volgende entiteiten opgenomen:

- Incident;
- Oplossing.

4.4.1.2 Globaal-technische structuur

Binnen de globaal-technische structuur van pilot 3 is de technische reikwijdte, technisch opslagmodel, het communicatiemodel en de interfaces met andere systemen beschreven.

De technische reikwijdte stelt vast welke onderdelen van het clientsysteem ondersteund worden door deze pilot.

Het technisch opslagmodel werkt de entiteiten, welke genoemd zijn in de globaal-functionele structuur, uit zodat deze in de database kunnen worden opgenomen.

Het communicatiemodel beschrijft aan de hand van een petrinet de communicatie zoals deze verloopt binnen de applicatie voor pilot 3.

4.4.1.3 Pilotontwikkelplan

Het pilotontwikkelplan omschrijft de bouweenheden van pilot 3. Deze bestaat uit het invoeren van het domeinmodel naar de database en het ontwikkelen van de webclient.

Bij pilot 3 is de clientcontrol niet meer gebruikt. Dit was omdat pilot 3 alle gegevens uit de database kon halen.

4.5 Pilot 4 Feedback naar gebruiker

4.5.1 Pilotontwerp-workshop

Tijdens de pilotontwerp-workshop zijn de drie documenten voor pilot 4 (Bijlage 5) opgesteld:

- Globaal-functionele structuur;
- Globaal-technische structuur;
- Pilotontwikkelplan.

4.5.1.1 Globaal-functionele structuur

De globaal-functionele structuur van pilot 4 bestond uit een prototype voor het systeemaanmeldingscherm en het domeinmodel.

Binnen het domeinmodel zijn de volgende entiteiten opgenomen:

- Incident;
- Oplossing;
- Systeem;
- Gebruiker.

4.5.1.2 Globaal-technische structuur

Binnen de globaal-technische structuur van pilot 4 is de technische reikwijdte, technisch opslagmodel, het communicatiemodel en de interfaces met andere systemen beschreven.

De technische reikwijdte stelt vast welke onderdelen van het clientsysteem ondersteund worden door deze pilot.

Het technisch opslagmodel werkt de entiteiten, welke genoemd zijn in de globaal-functionele structuur, uit zodat deze in de database kunnen worden opgenomen.

Het communicatiemodel beschrijft aan de hand van een petrinet de communicatie zoals deze verloopt binnen de applicatie voor pilot 4.

4.5.1.3 Pilotontwikkelplan

Het pilotontwikkelplan omschreef de bouweenheden van pilot 4. Deze bestond uit het invoeren van het domeinmodel naar de database, het ontwikkelen van de clientcontrol en het ontwikkelen van de webclient.

Pilot 4 maakt gebruik van de clientcontrol om de systeemnaam en ingelogde gebruiker te bepalen. Aan de hand hiervan kan het systeem emails uitsturen naar de gebruiker wanneer er een oplossing is gemeld en naar de beheerder als er een storing is gemeld.

4.6 Pilot 5 Oplossingherkenning

4.6.1 Pilotontwerp-workshop

Tijdens de pilotontwerp-workshop zijn de drie documenten voor pilot 5 (Bijlage 6) opgesteld:

- Globaal-functionele structuur;
- Globaal-technische structuur;
- Pilotontwikkelplan.

4.6.1.1 Globaal-functionele structuur

De globaal-functionele structuur van pilot 5 bestond uit een prototype voor het systeemaanmeldingscherm en het domeinmodel.

Binnen het domeinmodel zijn de volgende entiteiten opgenomen:

- Oplossing;
- Incident;
- Negeerwoorden.

4.6.1.2 Globaal-technische structuur

Binnen de globaal-technische structuur van pilot 5 is de technische reikwijdte, technisch opslagmodel, het communicatiemodel en de interfaces met andere systemen beschreven.

De technische reikwijdte stelt vast welke onderdelen van het clientsysteem ondersteund worden door deze pilot.

Het technisch opslagmodel werkt de entiteiten, welke genoemd zijn in de globaal-functionele structuur, uit zodat deze in de database kunnen worden opgenomen.

Het communicatiemodel beschrijft aan de hand van een petrinet de communicatie zoals deze verloopt binnen de applicatie voor pilot 5.

4.6.1.3 Pilotontwikkelplan

Het pilotontwikkel plan omschreef de bouweenheden van pilot 5. Deze bestond uit het invoeren van het domeinmodel naar de database, het ontwikkelen van de clientcontrol en het ontwikkelen van de webclient.

Binnen pilot 5 wordt er geen interactie uitgevoerd met het systeem maar vind alle interactie plaats tussen webclient en de database. Hieruit kan er bestaande storingen gevonden worden aan de hand van “negeerwoorden”. Door al deze woorden uit een melding van een gebruiker te halen houdt het systeem bruikbare sleutelwoorden over om te vergelijken met de database.

4.7 Invoering

4.7.1 Pilotinvoering

Aan het eind van elke pilotontwikkeling heeft er een invoering van de pilot plaatsgevonden. De resultaten hiervan zijn terug te vinden in Bijlage 7. Paragrafen 4.7.1.1 t/m 4.7.1.5 gaan in op de testaspecten van elke pilot.

4.7.1.1 Pilotacceptatietest pilot 1

Tijdens deze acceptatietest is de bruikbaarheid getest van pilot 1. Hier zijn de volgende aspecten getest:

- Gebruiker weet de clientapplicatie te benaderen;
- Gebruiker weet welke hardware er wordt gemeld;
- Gebruiker kan zonder al te veel uitleg het systeem melden.

De bijlage gaat dieper in op de volledige test.

4.7.1.2 Pilotacceptatietest pilot 2

Pilot 2. teste de volgende aspecten:

- Gebruiker weet de clientapplicatie te benaderen;
- Gebruiker kan bepalen welke software de storing veroorzaakt;
- Gebruiker kan zonder al te veel uitleg de storing melden.

4.7.1.3 Pilotacceptatietest pilot 3

Bij pilot 3 zijn de volgende testaspecten in acht genomen:

- Beheerder weet de clientapplicatie te benaderen;
- Beheerder kan bepalen welk systeem een storing bevat;
- Beheerder kan de systeem specificaties herkennen;
- Beheerder kan zonder al te veel uitleg een oplossing bieden aan de storing;
- Beheerder kan een geschiedenis van oplossingen zien.

4.7.1.4 Pilotacceptatietest pilot 4

Tijdens deze acceptatietest is het volgende getest:

- De beheerder weet wanneer een storing heeft voorgedaan;
- De gebruiker krijgt directe feedback van de handelingen van de beheerder.

4.7.1.5 Pilotacceptatietest pilot 5

Pilot 5 had geen uitgebreide test omdat deze veel aspecten van voorgaande tests gebruikte. Tijdens de acceptatietest van pilot 5 is er gelet op één aspect van de gebruikerinteractie met het systeem. Dit was:

- Met de formulering van een storingen gemeld door een gebruiker kan het systeem herhalingherkennen.

4.7.2 Pilotoverdracht

Na afronding van alle pilots heeft de pilotoverdracht plaatsgevonden. De afstudeerder heeft het beheer van de applicatie overgedragen aan de beheerder van Kuiper & Burger. Hieruit is geen documentatie ontstaan.

4.7.3 Feedback gebruikers

Aan de hand van de resultaten van de bruikbaarheidstests zijn ook de feedback van de gebruikers opgenomen. Dit is gebruikt als basis van de opvolgende pilots.

4.8 Go-/ no-go moment

Aan het einde van elke pilot werd er gekeken naar de testresultaten om te bepalen of de pilot geslaagd was of dat deze in een volgende iteratieslag moest worden herzien.

Dit is tijdens de ontwikkeling maar één keer voorgevallen en dit was bij pilot 1. De reden hiervoor was omdat het initiële SMS onstabiel en traag was. Hierdoor is de volledige pilotontwikkeling opnieuw uitgevoerd en bestaande broncode herschreven.

4.9 Checklist uitvoering

Om het SMS te kunnen implementeren moet men de volgende stappen ondernemen:

- Het ActiveX element moet eenmalig worden geïnstalleerd door de beheerder;
- Op de server moet men IIS installeren met ASP.Net ondersteuning;
- De webapplicatie moet bij IIS geregistreerd worden;
- Voor gebruikersgemak kan de beheerder een link plaatsen naar de weblocatie van de applicatie.

4.10 Korte handleiding gebruiker

Als de gebruiker een storing wil melden gaat deze naar de webpagina welke de beheerder heeft geregistreerd bij IIS voor de applicatie. Als het clientsysteem nog niet bestaat in de database zal het systeem vragen om deze te registreren. Het systeem analyseert alvast de benodigde hard- en software, dit duurt een enige tijd. Na analyse moet de gebruiker op de "Meld Aan" klikken om het systeem te registreren.

Als het systeem eenmaal is geregistreerd kan de gebruiker zijn incident melden op de incidentmelding pagina. Hier vult de gebruiker de symptomen in van de storing. Op deze handeling reageert het systeem met een mail met mogelijke oplossingen indien deze aanwezig zijn. Als er geen oplossing aanwezig is brengt het systeem de beheerder op de hoogte en krijgt de gebruiker pas een oplossing als de beheerder deze in het systeem plaatst.

5 Evaluatie

5.1 Evaluatie product

Zoals de tests aantonen, voldoet het SMS aan de gestelde eisen voor het project. Tijdens de ontwikkeling van het SMS is er niet genoeg rekening gehouden met de performance. Om deze reden is de opstartsnelheid van het SMS onacceptabel. Ook is de uitvoer nog steeds afhankelijk van de invoer van de beheerder. Hierdoor is de gebruiksvriendelijkheid nog niet voldoende af te leiden aan de beknopte tests die zijn uitgevoerd voor het SMS. Als de beheerder een duidelijk gebruiksvriendelijke oplossing biedt voor voorgevallen storingen is het resultaat meer bruikbaar voor de gebruikers.

De keuze om WMI te gebruiken binnen het project was goed uitgevallen omdat dit het onderdeel “systeemaanmelding” volautomatisch kon maken. Alleen was het een verkeerde keuze om dit via een webpagina te doen. De methode om via een webpagina een analyse uit te voeren naar de systeemhardware is omslachtig en vereist dure certificaten om dit toe te staan.

5.2 Evaluatie proces

Dankzij het gebruik van IAD is het proces structureel verlopen. Het enige probleem waar tegenaan is gelopen is dat de pilotplanning niet secuur was.

Doordat het onderzoek in pilot 1 uitliep leek het initieel dat de planning niet meer te halen was. Maar omdat de onderzoeksresultaten ook gebruikt konden worden in de overige pilots werd hier de werklast bij afgenomen.

Na afronding van pilot 1 kon er op de planning in worden gelopen omdat de overige pilots vrij eenvoudig waren. De enige pilot die na pilot 1 grote last droeg was pilot 5. Maar ook deze was eenvoudiger door de onderzoeksresultaten van pilot 1.

5.3 *Aanbevelingen*

Het project had sneller kunnen verlopen als de oplossing niet als webpagina werd ontworpen. Het onderzoek naar webbeveiligingen heeft te veel tijd in beslag genomen.

Ook is niet aan te bevelen om UML te gebruiken voor een webpagina. Een webpagina werkt nauwelijks met objecten. Hierdoor kan één van de hoofdmodellen van UML, het klassendiagram, niet optimaal worden ingezet.

Het was zeer prettig en structureel werken met IAD als ontwikkelmethodologie. De documenten die werden opgeleverd waren zeer eenvoudig te gebruiken en fungeerde goed als naslagwerk. Het enige nadeel is dat IAD ervan uitgaat dat er in teamverband een ontwikkeltraject wordt doorlopen. Het volledige IAD traject was hierdoor niet uitgevoerd.

5.4 *Vervolgonderzoek/ uitbreiding*

Als vervolg zou er onderzoek naar integratie met bestaande beheertools kunnen worden gedaan. Hierbinnen zal WMI een grote rol kunnen spelen omdat dit een vrij uitgebreide API is met directe toegang tot Windows bronnen. Op <http://msdn.microsoft.com/> wordt zelfs uitgelegd hoe je standaard applicaties kan installeren door middel van deze API.

Bronvermeldingen

Literatuur:

- IAD, Het evolutionair ontwikkelen van informatiesystemen, R.J.H. Tolido, ISBN: 90-395-0401-6;
- IT Service management een introductie, J. van Bon, ISBN: 90-806713-2-0;
- Projectmatig creëren, J. Bos en E. Harting, ISBN: 90-5594-216-2;
- De UML Toolkit, H. E. Eriksson en M. Penker, ISBN: 90-395-1015-6;
- Applying UML and patterns, C. Larman, ISBN: 0-13-092569-1;
- Codeguru quick C# tutorial, http://www.codeguru.com/Csharp/Csharp/cs_syntax/article.php/c5837/;
- COM/ActiveX C# tutorial, http://www.devhood.com/messages/message_view-2.aspx?thread_id=16569;
- ASP Tutorial, <http://hotwired.lycos.com/webmonkey/03/30/index3a.html?tw=programmin g>;
- ASP/C# mysql tutorial, <http://www.sitepoint.com/forums/showthread.php?t=158965>;
- WMI made easy Tutorial, <http://www.csharp-help.com/archives2/archive334.html>;
- Microsoft developer reference, <http://msdn.microsoft.com/>;
- Deja reference, <http://www.deja.com>.

Studiemateriaal:

- AP001, Processen en taken in organisaties;
- AP003, Ontwikkelen van informatie systemen;
- BP001, ICT Dienstverlening deel 1;
- CS07, Coöperatieve systemen;
- DB85, Gegevensmodellering;
- II002, Relationele databases;
- SO10, Ongestructureerde gegevens.

Begrippenlijst

Bijlage 1 Definitiestudie

Bijlage 2 Pilot 1 – Aanmelding clientsysteem

Bijlage 3 Pilot 2 – Storingmelding

Bijlage 4 Pilot 3 – Oplossingmelding

Bijlage 5 Pilot 4 – Feedback naar gebruiker

Bijlage 6 Pilot 5 – Oplossingherkenning

Bijlage 7 Testresultaten