MARTI·IT
Automatisering en Advies

AFSTUDEERSCRIPTIE

De ontwikkeling van een communicatietool voor het updaten van software bij Marti-IT

Aanhef:

Module:	Afstuderen	**Studierichting**	IVIT, Applicatieontwikkeling
Docent(en):	Dhr. J.L.A. Schramp Dhr. M. Reijnhoudt	**Bedrijfsmentor:**	Dhr. M.L. Verkade
		Plaats:	Naaldwijk
Organisatie:	Marti-IT	**Datum:**	9 juni 2005

Student:

Naam:	**E-Mail:**
Brüning, Mathieu	20010019

© 2005 Mathieu Brüning (20010019)

(Deze pagina is opzettelijk leeg gelaten)

© 2005 Mathieu Brüning (20010019)

Referaat

Deze scriptie is met name interessant voor mensen die geïnteresseerd zijn in de ontwikkeling van een softwareapplicatie middels de Iterative Application Development methodiek. Daarnaast richt de scriptie zich op het ontwikkelen van een applicatie middels de .NET programmeertaal C# (C-sharp).

Het project heeft plaatsgevonden van 7 februari 2005 tot en met 10 juni 2005, binnen de organisatie Marti-IT en is uitgevoerd door Mathieu Brüning.

Descriptoren:

- ✓ Marti-IT
- ✓ Iterative Application Development (IAD)
- ✓ .NET
- ✓ C#
- ✓ Updaten van software

(Deze pagina is opzettelijk leeg gelaten)

© 2005 Mathieu Brüning (20010019)

Voorwoord

Dit rapport is de afstudeerscriptie van Mathieu Brüning, student van de Haagse Hogeschool, ten gevolg van het afstudeertraject waarin de student zich de afgelopen 18 weken heeft bevonden.

Het rapport is opgebouwd uit twee delen, elk deel gericht op een specifiek aspect van het afstudeertraject.
Lezers die vooral geïnteresseerd zijn in een beschrijving van de organisatie, waar het project is volbracht, worden verwezen naar hoofdstuk 2. Een uitgebreide beschrijving van de uitgevoerde werkzaamheden tijdens het project bevindt zich in hoofdstuk 6. Een evaluatie op zowel proces- als productmatig niveau is te vinden in hoofdstuk 9.

Dit rapport is geoptimaliseerd voor dubbelzijdig afdrukken, waardoor pagina's opzettelijk zijn leeg gelaten om specifieke pagina's op de rechterpagina te plaatsen.

De student is veel dank verschuldigd aan dhr M.L. Verkade voor de projectbegeleiding en dhr. A. Barendse voor het verschaffen van technische informatie tijdens het project. Daarnaast is dank verschuldigd aan dhr. Schramp en dhr. Reijnhoudt voor de begeleiding vanuit de richting van de opleiding.

Naaldwijk, mei 2005, M.D. Brüning

© 2005 Mathieu Brüning (20010019)

Inhoudsopgave

1 INLEIDING 9

DEEL 1: BEDRIJF EN OPDRACHTEN

2 BEDRIJFSSCHETS MARTI-IT 12

2.1 ORGANISATIE ALGEMEEN 12
2.2 ORGANISATIE SPECIFIEK 12
2.3 STRUCTUUR 13
2.4 PLAATS VAN AFSTUDEERDER IN ORGANISATIE 13

3 OPDRACHT 14

3.1 AANLEIDING 14
3.2 PROBLEEMSTELLING 15
3.3 DOELSTELLINGEN 16
3.4 UITGANGSSITUATIE 17
3.5 PRODUCTEN 18
3.6 DOELSTELLINGEN M.B.T. DE OP TE LEVEREN PRODUCTEN 19

4 PLAN VAN AANPAK 21

4.1 METHODEN EN TECHNIEKEN 21
4.2 ACTIVITEITEN 23
4.3 PLANNING 26
4.4 AANVANGSSITUATIE 28

DEEL 2: WERKZAAMHEDEN & EVALUATIE

5 AANPAK PROJECT 30

5.1 PLAN VAN AANPAK 30
5.2 OPLOSSINGSMETHODEN 31
5.3 VERDEDIGING GEKOZEN AANPAK 32

6 WERKWIJZEBESCHRIJVING 34

6.1 ORIËNTATIE / INITIATIEFASE 34
6.2 DEFINITIEFASE 35
6.3 PILOTONTWIKKELING 41
14.4 INVOERING / NAZORGFASE 79

7 ANALYSE VAN DE RESULTATEN 82

7.1 UITBREIDING VAN DE FUNCTIONALITEIT 82
7.2 VERBETERING OP HET GEBIED VAN ONDERHOUDBAARHEID 82
7.3 VERBETERING VAN PROCESSEN BINNEN MARTI-IT 82
7.4 VERBETERING VAN DE GEBRUIKERSVRIENDELIJKHEID 83
7.5 OPGEDANE ERVARING M.B.T. HET .NET FRAMEWORK EN DE C# PROGRAMMEERTAAL 83

8 OPGELEVERDE PRODUCTEN 84

8.1 RAPPORT MET DE SPECIFICATIES VAN HET MARTI-IT UPDATE / BUILD CENTRUM 84
8.2 HANDLEIDINGEN T.B.V. HET GEBRUIK VAN MARTI-IT UPDATE / BUILD CENTRUM 85
8.3 TESTRAPPORTEN ACCEPTATIETEST GEBRUIKERS MARTI-IT SOFTWARE 85
8.4 MARTI-IT UPDATE / BUILD CENTRUM 86

9 EVALUATIES 87

9.1 PROCES 87
9.2 PRODUCT 89

© 2005 Mathieu Brüning (20010019)

LITERATUURLIJST **93**

BOEKEN 93
WEBSITES 93

LIJST VAN FIGUREN **94**

VERKLARENDE WOORDENLIJST **95**

BIJLAGE 1: OPDRACHTOMSCHRIJVING **97**

BIJLAGE 2: PLAN VAN AANPAK **100**

BIJLAGE 3: RESULTATEN DEFINITIESTUDIE **126**

BIJLAGE 4: PILOTONTWIKKELPLANNEN **127**

BIJLAGE 5: REFERENCE CARDS MARTI-IT UPDATE / BUILD CENTRUM **157**

BIJLAGE 6: TESTRAPPORT ACCEPTATIETEST GEBRUIKERS MARTI-IT SOFTWARE **158**

© 2005 Mathieu Brüning (20010019)

(Deze pagina is opzettelijk leeg gelaten)

© 2005 Mathieu Brüning (20010019)

1 Inleiding

Marti-IT, producent en leverancier van met name branche specifieke software, maakt veelvuldig gebruik van het Internet om de communicatie te bewerkstelligen van én naar haar klanten. Middels een speciaal ontwikkelde applicatie genaamd het Marti-IT Update Centrum worden gebruikers in staat gesteld om op eenvoudige wijze contact te leggen met de softwareleverancier, om bijvoorbeeld updates voor de Marti-IT software te downloaden. Inmiddels is het alweer enige tijd geleden dat het Update Centrum ontwikkeld is en zodoende is de applicatie enigszins verouderd en voldoet het niet meer geheel aan de eisen die Marti-IT stelt aan dit Update Centrum. Sommige aspecten van de applicatie kunnen uitgebreid en verbeterd worden en daarom blijkt er de behoefte te zijn naar een nieuwe applicatie. Daarnaast richt het toekomstige beleid met betrekking tot softwareontwikkeling binnen Marti-IT zich op het ontwikkelen middels de .NET programmeer C#. Dit was ook een reden om een nieuw systeem te ontwikkelen, in plaats van het uitbreiden van het bestaande systeem middels de Visual Objects programmeertaal.

Het afstudeertraject, waar student Mathieu Brüning zich de afgelopen 18 weken in bevond, richtte zich zodoende op de ontwikkeling van een nieuwe communicatie applicatie voor het verzenden van gegevens van en naar de gebruikers van de Marti-IT software.

De aanleiding van deze scriptie is het afstudeertraject dat student Mathieu Brüning de afgelopen 18 weken heeft doorlopen. Om een adequaat beeld te schetsen, omtrent het gehele project, dient er een solide scriptie te zijn opgesteld.

Het doel van deze scriptie is een goed inzicht te verschaffen omtrent de werkzaamheden, resultaten en methoden behorend bij het afstudeertraject.

Het rapport is ingedeeld in twee delen, elk deel gericht op een ander aspect van het afstudeertraject. Het eerste deel ("Bedrijf & opdrachten") geeft een beknopte beschrijving van de organisatie. Daarnaast wordt, in datzelfde deel, de opdracht met betrekking tot het project tot in detail beschreven.
Het tweede deel, "Werkzaamheden en evaluatie", bevat de kern van het afstudeerverslag. Hierin wordt onder andere duidelijk uiteengezet met welke werkzaamheden de afstudeerder zich heeft beziggehouden. Naast de beschrijving van de werkzaamheden bevat dit deel van de scriptie tevens een evaluatie van het afstudeertraject. De evaluatie richt zich zowel op het procesmatige deel als het productmatige deel van het afstudeertraject.

© 2005 Mathieu Brüning (20010019)

(Deze pagina is opzettelijk leeggelaten)

© 2005 Mathieu Brüning (20010019)

DEEL 1: BEDRIJF & OPDRACHTEN

De volgende drie hoofdstukken vormen deel 1 van de afstudeerscriptie. Deel 1 richt zich enerzijds op Marti-IT, het bedrijf waar het afstudeertraject bij volbracht is en anderzijds op de opdracht behorend bij het afstudeertraject.

Ten eerste bevat hoofdstuk twee ("Bedrijfsschets Marti-IT") een beschrijving van Marti-IT als organisatie. Het daarop volgende hoofdstuk richt zich op de definitie van het project, waarin onder andere de aanleiding, probleemstelling, de doelstellingen en de uitgangssituatie centraal staan. Het laatste hoofdstuk in deel 1, hoofdstuk 4, richt zich op de vooraf gedefinieerde aanpak van het project, waarin ondermeer een overzicht van de activiteiten en een planning te vinden is.

© 2005 Mathieu Brüning (20010019)

2 Bedrijfsschets Marti-IT

De organisatie, waarin student Mathieu Brüning het afstudeertraject heeft volbracht, wordt in het volgende hoofdstuk beschreven. Dit hoofdstuk dient u van informatie te voorzien, zodat u een adequaat beeld kan schetsen van het bedrijf waarin het afstudeertraject is uitgevoerd.

Het hoofdstuk is opgedeeld in vier paragrafen, waarin de eerste paragraaf een algemene beknopte beschrijving geeft van de ICT-dienstverlener als organisatie. De daarop volgende paragraaf, paragraaf 2.2, bevat een specifieke beschrijving van het bedrijf. Paragraaf 2.3 richt zich op de structuur van de organisatie, waarin aan de hand van een organogram een beeld geschetst wordt van de indeling van het bedrijf. De laatste paragraaf bevat tot slot een beschrijving van de plaats van de afstudeerder in de organisatie.

2.1 Organisatie algemeen

Marti-IT is een jong dynamisch bedrijf, wat met name gespecialiseerd is op het gebied van software voor bakkerijen en orthopedisten. Naast deze branchespecifieke software verzorgt het bedrijf maatwerk software voor diverse bedrijven zoals de Gemeente Den Haag en Anton van Baalen plant, bloem en decor. De organisatie is een bedrijf in de ICT-branche en behoort tot het midden- en kleinbedrijf Nederland.

2.2 Organisatie specifiek

Marti-IT, gevestigd in Naaldwijk, is een jonge groeiende ICT-organisatie, actief op het gebied van software voor bakkerijen en orthopedisten, maar maakt ook maatwerk software voor diverse bedrijven. Marti-IT is een kleine en zeer efficiënte organisatie met direct contact met de markt. Na de overname van bakkerij software van organisaties JMS en Compuvision in 2002 en 2003 is de groei van Marti-IT echt begonnen. Door de overname van zowel de klanten als de software was Marti-IT in staat om haar naam te vestigen in de bakkerij branche. Met name door de geboorte van het 'Bake-it ' softwarepakket in augustus 2003 heeft de groeit doorgezet.

Inmiddels zijn er nog een aantal andere softwarepakketten ontwikkeld in de 'Bake-it' softwarelijn, waaronder ordering via het Internet, automatisering van de etikettering in de bakkerij en touchscreen kassasystemen voor in de winkel.
Na ongeveer twee jaar kan de organisatie dan ook trots zijn op de positie als marktleider op het gebied van automatisering van de bakkerij.
Naast de bakkerij software richt het bedrijf zich ook nog voor een deel op orthopedische software en voor een klein deel op de ontwikkeling van maatwerk software.

Bij Marti-IT staat het verbeteren en optimaliseren van interne bedrijfsprocessen van de klant centraal. De organisatie doet dit middels het leveren van technisch hoogwaardige branche specifieke softwarepakketten. Het gebruik van het internet speelt hierbij een belangrijke rol.

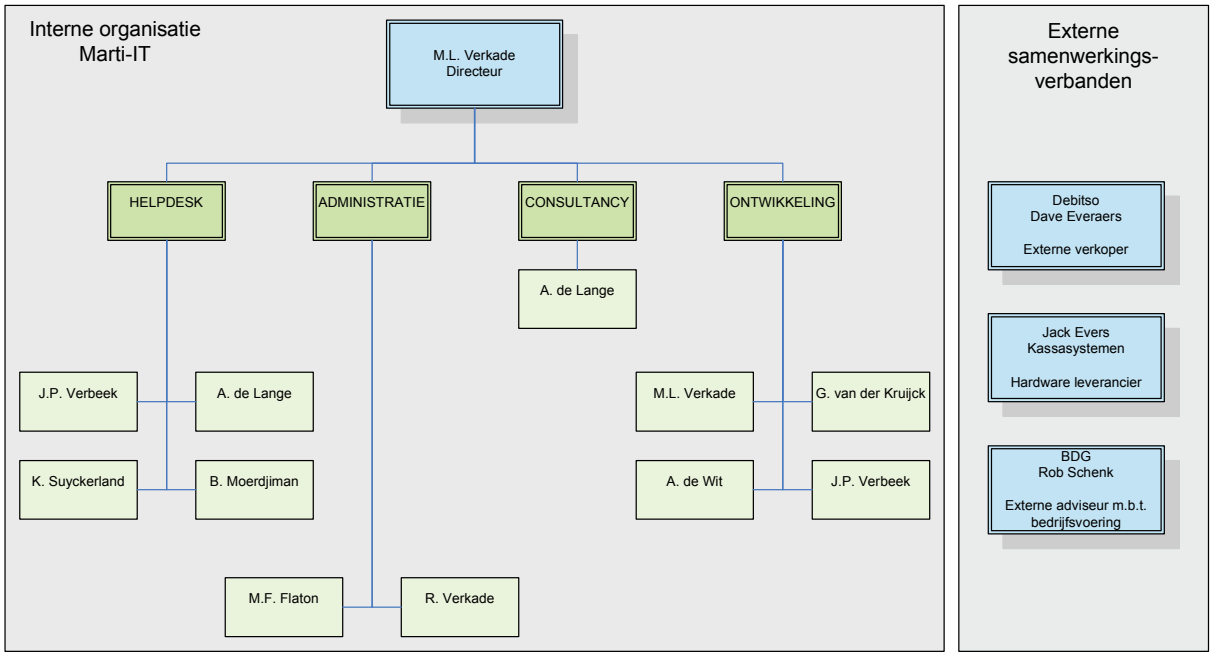
Op het gebied van de ontwikkeling van software is het beleid van Marti-IT hoofdzakelijk dat de wensen en eisen van klanten voorop staan. Continue houdt het bedrijf zich bezig met de ontwikkeling van de softwarepakketten, waardoor er wekelijks nieuwe updates verschijnen voor de Marti-IT software. Sprake van een concreet toegepaste methodiek voor de systeemontwikkeling is er feitelijk niet, aangezien de continue doorlopende ontwikkeling aan de softwarepakketten veelal zorgt voor directe oplossingen zonder voortraject.

Momenteel zijn er ongeveer 500 bedrijven die werken met de Marti-IT software. Via een klein aantal van gespecialiseerde wederverkopers en relaties worden branchespecifieke oplossingen geboden, zoals bijvoorbeeld de volledige automatisering van een bakkerij, van productie tot de afrekening in de winkel. Eigenschappen van de Marti-IT zijn: alle relevante informatie geïntegreerd in één systeem, zeer korte implementatietijd, web enabled, sterke integratie met Microsoft Office en groeit mee met bedrijf. De Marti-IT software is dan ook zowel geschikt voor het midden en klein bedrijf als voor grotere organisaties.

© 2005 Mathieu Brüning (20010019)

2.3 Structuur

Onderstaand wordt het organigram van Marti-IT weergegeven, welke de structuur van de organisatie representeert.



Figuur 1: *Organogram Marti-IT inclusief externe samenwerkingsverbanden*

2.4 Plaats van afstudeerder in organisatie

Marti-IT is een kleine organisatie, met een personeelsbestand van negen personen. In tegenstelling tot grote organisaties is er daarom bij Marti-IT minder sprake van duidelijke scheidingen tussen bepaalde afdelingen.
Zoals u bijvoorbeeld kunt zien in het organogram van Marti-IT (zie vorige paragraaf), worden meerdere functies soms uitgevoerd door dezelfde medewerkers. Zo is te zien in het organogram dat dhr. M.L. Verkade, dhr. J.P. Verbeek en dhr. A. de Lange een dubbele functie bekleden binnen de organisatie.

In het fysieke opzicht is er wel sprake van een duidelijke scheiding, aangezien de medewerkers van de helpdesk afdeling zijn gescheiden van de medewerkers van de afdeling ontwikkeling.

In het kader van de afstudeeropdracht, namelijk het ontwikkelen van een nieuw Marti-IT Update Centrum, kan gesteld worden dat de afdeling ontwikkeling van toepassing is. De plaats van de afstudeerder binnen de organisatie betrof dan ook de afdeling ontwikkeling.

© 2005 Mathieu Brüning (20010019)

3 Opdracht

Hoofdstuk drie richt zich op de definiëring van de afstudeeropdracht. Het hoofdstuk is hierbij onderverdeeld in paragrafen, waarbij elke paragraaf een onderdeel vormt van de definiëring van de opdracht. De eerste paragraaf beschrijft de aanleiding met betrekking tot de opdracht. De probleemstelling van de hoofdopdracht van het afstudeertraject staat centraal in de paragraaf 3.2. De daarop volgende paragraaf ("Doelstellingen") geeft een opsomming van de doelstellingen, die gesteld zijn aan de afstudeeropdracht. Vervolgens bevat paragraaf 3.4 een beschrijving van de uitgangspunten, (vooraf) gesteld aan het afstudeertraject. Een opsomming van de op te leveren producten wordt vermeldt in het paragraaf 3.5. De laatste paragraaf in hoofdstuk 3 belicht de doelstellingen die zijn gesteld aan de op te leveren producten tijdens het project.

3.1 Aanleiding

Marti-IT, een jong en dynamisch bedrijf, gevestigd in Naaldwijk, is specialist op het gebied van software voor bakkerijen en orthopedisten. Momenteel zijn er ruim 425 bakkers en zo'n 75 orthopedisten die gebruik maken van de Marti-IT software. Naast de software voor bakkerijen en orthopedisten besteedt de organisatie ook nog een deel van haar werkzaamheden aan maatwerksoftware en advisering. Het bedrijf telt een negental medewerkers, waaronder twee vaste ontwikkelaars, een ontwikkelaar / helpdeskmedewerker, twee vaste helpdeskmedewerkers, een consultant / helpdeskmedewerker, een directeur / ontwikkelaar en twee administratieve medewerksters.

Zoals vermeldt in § 2.2 is er binnen Marti-IT niet echt sprake van een concreet toegepaste methodiek met betrekking tot de systeemontwikkeling, aangezien de continue doorlopende ontwikkeling aan de softwarepakketten veelal zorgt voor directe oplossingen zonder voortraject. De constante ontwikkeling aan de Marti-IT software draagt dan ook zorg voor het regelmatig verschijnen van nieuwe updates.

Toen in het verleden gebleken is dat het te veel inspanning kostte om updates te verzenden van en naar de gebruikers van de Marti-IT software is er binnen relatief korte tijd een applicatie ontwikkeld dat functioneert als doorgeefluik tussen de klanten en Marti-IT. Het doorgeven van gegevens van en naar de klanten middels het zogenaamde Marti-IT Update Centrum stelt klanten op in staat om de nieuwste updates en andere bestanden te downloaden voor de Marti-IT software. Daarnaast is het tevens mogelijk om met het Marti-IT Update Centrum gegevens zoals back-ups te verzenden naar Marti-IT, zodat eventuele problemen in de software gericht opgelost kunnen worden door de helpdesk. Na enkele jaren veelvuldig en naar tevredenheid gebruikt te zijn, blijkt dat er behoefte is naar een uitbreiding van de functionaliteit van het systeem en dient de interface moderner en sprekender te worden.

© 2005 Mathieu Brüning (20010019)

3.2 Probleemstelling

Het huidige Marti-IT Update Centrum is een ouder product dat al lange tijd geleden is ontwikkeld, wat ervoor gezorgd heeft dat de huidige functionaliteit van het systeem niet meer volledig de behoeftes van de klant dekt. Feitelijk is de basis functionaliteit van de applicatie dus wel in orde, maar echter niet meer volledig ten aanzien van de huidige behoeftes vanuit de gebruikersrichting.

De specifieke kenmerken van een aantal verschillende applicaties van Marti-IT hebben ervoor gezorgd dat er op dit moment vier verschillende Marti-IT Updates Centrums in omloop zijn, elk gericht op een specifieke applicatie. Deze individuele systemen moeten zodoende allemaal apart onderhouden worden, wat onnodig tijdverbruik veroorzaakt.

Op het gebied van de gebruikte functionaliteiten binnen Marti-IT kan de applicatie ook nog sterk verbeterd worden. Tijdsintensieve routinetaken als het downloaden en archiveren van gegevensback-ups van gebruikers worden nu nog handmatig uitgevoerd, waarbij veel tijd verloren gaat.

Daarnaast kan met name de gebruikersvriendelijkheid van de applicatie kan sterk verbeterd worden, aangezien onder andere de menustructuur van de grafische interface nog niet optimaal is. De applicatie is tevens al redelijk gedateerd wat ervoor heeft gezorgd dat het cosmetische gedeelte van de applicatie zich niet meer kan meten met de concurrerende applicaties.

Tot slot is het huidige Marti-IT Update Centrum geschreven in de programmeertaal Visual Objects. Het ontwikkelen in de Visual Objects programmeertaal past echter niet meer bij het toekomstige softwareontwikkelingsbeleid van het bedrijf, aangezien het bedrijf in de toekomst volledig wil overgaan op het ontwikkelen in de .NET programmeertaal C#.

© 2005 Mathieu Brüning (20010019)

3.3 Doelstellingen

De doelstelling van de opdracht richt zich op het ontwikkelen van een nieuw Marti-IT Update Centrum, waar naast de functionaliteit ook de onderhoudbaarheid een belangrijk aspect is.

Een ander belangrijk aspect van het nieuwe Update Centrum is dat het geschreven moet worden in de .NET programmeertaal C#, aangezien het toekomstbeleid van softwareontwikkeling binnen Marti-IT zich richt op ontwikkelen in de .NET programmeertaal C#. De ontwikkeling van het systeem in C# dient er tevens voor te zorgen dat eventuele herbruikbare componenten als input dienen voor het opzetten van een toekomstig C# framework voor Marti-IT.

Tot slot dient de applicatie ervoor te zorgen dat sommige processen, zoals bijvoorbeeld het klaarzetten van een nieuwe update op het Internet of het downloaden van gegevensback-ups, geautomatiseerd dienen te worden. Op deze manier kunnen dergelijke processen binnen de organisatie geoptimaliseerd worden.

Door de duidelijke splitsing tussen functionele systeemeisen voor de gebruikers van de Marti-IT software en de medewerkers van Marti-IT is er sprake van de ontwikkeling van een tweetal applicaties. De doelstellingen met betrekking tot deze producten worden hieronder opgesomd:

- ✓ Een nieuw Marti-IT Update Centrum (voor de gebruikers van de Marti-IT software), bestaande uit:
  - De functionaliteit van de bestaande applicatie, herschreven in de .NET programmeertaal C#
    - Downloaden van updates, persoonlijke bestanden en andere hulpmiddelen;
    - Uploaden van back-ups, persoonlijke bestanden en andere gegevens;
  - Een nieuwe interface, waarin de gebruikersvriendelijkheid meetbaar beter wordt nageleefd dan in de huidige applicatie
    - Nieuwe kleurensamenstelling;
    - Nieuwe menustructuur;
    - Nieuwspagina's in html formaat, die worden getoond tijdens het downloaden van de updates.
- ✓ Een nieuw Marti-IT Build Centrum (voor de medewerkers van Marti-IT), bestaande uit:
  - Extra functionaliteiten
    - Het automatiseren van bepaalde processen, zoals bijvoorbeeld het klaarzetten van een update en het automatisch downloaden en archiveren van verzonden gegevens;
    - Extra mogelijkheden met betrekking tot het in- en uitpakken in het .zip formaat;
    - Extra mogelijkheden op het gebied van commandline opties.

© 2005 Mathieu Brüning (20010019)

3.4 Uitgangssituatie

3.4.1 Software

De volgende softwarematige middelen zijn nodig om de totstandkoming van de (deel)producten te kunnen realiseren:

- Software met betrekking tot de rapportage rondom het project (Microsoft Office)
- Software met betrekking tot het ontwikkelen van technische ontwerpen (Microsoft Visio)
- Software met betrekking tot het ontwikkelen in C# (Microsoft Visual Studio 2003)
- Software met betrekking tot het ontwikkelen in HTML en CSS (Microsoft Visual Studio 2003)
- Software met betrekking tot het (eventueel) grafisch ontwikkelen van afbeeldingen e.d. (Adobe Photoshop)

3.4.2 Hardware

De volgende hardwarematige middelen zijn nodig om de totstandkoming van de (deel)producten te kunnen realiseren:

- Een werkplek
- Een printer
- Een testserver
- Een verbinding met het Internet

3.4.3 Rapporten

Tijdens het afstudeertraject had ik niet de mogelijkheid tot het inzien van reeds bestaande (stage / afstudeer) rapporten. De ontwikkeling van het huidige Marti-IT Update Centrum kenmerkte zich voornamelijk door het direct bouwen van het systeem, zonder enig voortraject. Sprake van een concrete ontwerpfase was er dus feitelijk niet, wat ervoor gezorgd heeft dat er geen documentatie zoals een functioneel ontwerp of een technisch ontwerp aanwezig was.

Samenvattend kan ik stellen dat er geen beschikbare documentatie / rapporten aanwezig was binnen Marti-IT met betrekking tot de ontwikkeling van het bestaande Marti-IT Update Centrum.

© 2005 Mathieu Brüning (20010019)

3.5 Producten

3.5.1 Producten m.b.t. de afstudeeropdracht

- ✓ Plan van aanpak (inclusief gedetailleerde planning);
- ✓ Rapport met de specificaties voor het Marti-IT Update Centrum;
- ✓ Testrapporten acceptatietests gebruikers Marti-IT software / medewerkers Marti-IT;
- ✓ Handleidingen (reference cards) voor gebruikers Marti-IT software / medewerkers Marti-IT;
- ✓ Nieuw Marti-IT Update Centrum.

3.5.2 Producten m.b.t. het opleidingsinstituut

- ✓ Plan van aanpak (inclusief gedetailleerde planning);
- ✓ Een voortgangsrapport;
- ✓ Bouwplan afstudeerscriptie;
- ✓ Afstudeerscriptie.

© 2005 Mathieu Brüning (20010019)

3.6 Doelstellingen m.b.t. de op te leveren producten

3.6.1 Rapport met de specificaties voor het Marti-IT Update / Build Centrum

Het opleveren van een rapport met de specificaties voor het nieuwe systeem heeft als voornaamste doel het definiëren van alle kenmerken omtrent het te ontwikkelen systeem. Door in overeenkomst met de opdrachtgever de specificaties van het te ontwikkelen systeem vast te leggen, is er in de verdere verloop van het project duidelijk waaraan het nieuwe systeem moet voldoen. De ontwikkeling van de pilots tijdens het project zal verlopen op basis van de gedefinieerde specificaties in dit rapport.

Doelstellingen met betrekking tot het rapport met de specificaties van het Marti-IT Update / Build Centrum zijn:

- ✓ Het vastleggen van de kenmerken rondom het ontwikkelscenario;
- ✓ Het analyseren van het bestaande Marti-IT Update Centrum, waarbij de geïmplementeerde functionaliteiten en de positieve en negatieve aspecten van de applicatie in kaart gebracht dienen te worden;
- ✓ Het analyseren van Update Centrums van andere softwarepakketten, zodat zowel positieve als negatieve aspecten van deze systemen in kaart gebracht kunnen worden;
- ✓ Het vastleggen van de systeemeisen waarin het nieuwe systeem dient te voldoen;
- ✓ Het bepalen van een systeemconcept;
- ✓ Het vastleggen van de technische architectuur van het te ontwikkelen systeem;
- ✓ Het vastleggen van de organisatorische inrichting van het te ontwikkelen systeem.

3.6.2 Testrapporten acceptatietests gebruikers Marti-IT software / medewerkers Marti-IT

De doelstelling met betrekking tot het opleveren van een testrapport omtrent de acceptatietest met de medewerkers van Marti-IT richt zich hoofdzakelijk op het vastleggen van de testbevindingen van de medewerkers Marti-IT. Door in kaart te brengen op welke fronten de applicatie geaccepteerd is en op welke fronten de applicatie (eventueel) tekortschiet kan bepaald worden of het systeem gereed is voor implementatie. Het opleveren van het testrapport dient daarom nauwkeurig de resultaten van de acceptatietest met de medewerkers van Marti-IT te bevatten.

De doelstelling omtrent het opleveren van een testrapport met betrekking tot de acceptatietest met de gebruikers van de Marti-IT software richt zich hoofdzakelijk op hetzelfde als hetgeen wat in de vorige paragraaf is besproken. Op opleveren van een testrapport met betrekking tot de acceptatietest met de gebruikers van de Marti-IT software dient met name inzicht te verschaffen omtrent de gereedheid van het systeem.

Concrete doelstellingen met betrekking tot de testrapporten van de acceptatietests zijn:

- ✓ Het vastleggen van de bevindingen van de gebruikers van de Marti-IT omtrent het nieuwe Marti-IT Update Centrum;
- ✓ Het vastleggen van de bevindingen van de medewerkers van Marti-IT omtrent het Marti-IT Build Centrum;
- ✓ Het vastleggen van de gereedheid met betrekking tot de implementatie van het Marti-IT Update Centrum en het Marti-IT Build Centrum.

© 2005 Mathieu Brüning (20010019)

3.6.3 Handleidingen t.b.v. het gebruik van het Marti-IT Update / Build Centrum

Het opstellen van handleidingen in de vorm van een reference cards heeft als voornaamste doel het verschaffen van informatie omtrent het Marti-IT Update Centrum en het Marti-IT Build Centrum. Door deze handleidingen op te stellen kunnen de gebruikers van de Marti-IT software en de medewerkers van Marti-IT in het geval van eventuele onduidelijkheden tijdens het gebruik van de applicaties terugvallen op informatie omtrent de systemen. Hierbij is met name het verschaffen van informatie omtrent de werking en functionaliteit van de applicaties van belang.

De doelstellingen met betrekking tot de reference cards zijn als volgt:

- ✓ Het opleveren van een reference card voor de gebruikers van de Marti-IT software, waarin informatie wordt vermeldt omtrent de werking van de functionaliteiten van het Marti-IT Update Centrum. Deze functionaliteiten zijn:
  - Downloaden van updates
  - Downloaden van historische updates
  - Uploaden van back-ups
  - Uploaden van bestanden of mappen
  - Uploaden van standaard uploads
  - Maken van gegevensback-ups

- ✓ Het opleveren van een reference card voor de medewerkers van Marti-IT, waarin informatie wordt vermeldt omtrent de werking van de functionaliteiten van het Marti-IT Build Centrum. Deze functionaliteiten zijn:
  - Het beheren van gebruikersgegevens
  - Het beheren van softwaregegevens (applicaties en download types)
  - Het beheren van updategegevens

3.6.4 Nieuw Marti-IT Update / Build Centrum

Door de duidelijke splitsing tussen functionele systeemeisen voor de gebruikers van de Marti-IT software en de medewerkers van Marti-IT is er sprake van de ontwikkeling van een tweetal applicaties. De doelstellingen met betrekking tot deze producten worden hieronder opgesomd.

- ✓ Een nieuw Marti-IT Update Centrum (voor de gebruikers van de Marti-IT software), bestaande uit:

  - De functionaliteit van de bestaande applicatie, herschreven in de .NET programmeertaal C#
    - Downloaden van updates, persoonlijke bestanden en andere hulpmiddelen;
    - Uploaden van back-ups, persoonlijke bestanden en andere gegevens;

  - Een nieuwe interface, waarin de gebruikersvriendelijkheid meetbaar beter wordt nageleefd dan in de huidige applicatie
    - Nieuwe kleurensamenstelling;
    - Nieuwe menustructuur;
    - Nieuwspagina's in html formaat, die worden getoond tijdens het downloaden van de updates.

- ✓ Een Marti-IT Build Centrum (voor de medewerkers van Marti-IT), bestaande uit:

  - Extra functionaliteiten
    - Het automatiseren van bepaalde processen, zoals bijvoorbeeld het klaarzetten van een update en het downloaden en archiveren van door gebruikers verzonden gegevens;
    - Extra mogelijkheden met betrekking tot het in- en uitpakken in het .zip formaat;
    - Extra mogelijkheden op het gebied van commandline opties.

© 2005 Mathieu Brüning (20010019)

4 Plan van aanpak

Om het project succesvol laten te verlopen zal er gebruik gemaakt gaan worden van verschillende technieken in combinatie met een ondersteunende methode. De eerste paragraaf in dit hoofdstuk richt zich op de gehanteerde methode en technieken tijdens het project. De verscheidene activiteiten tijdens het uitvoering van het afstudeertraject zullen in de tweede paragraaf worden opgesomd. Aan de hand van de globale planning (zie einde van § 4.2) wordt namelijk voor elke fase van de afstudeerperiode een opsomming gemaakt van de bijbehorende activiteiten. De verschillende fasen van de planning zijn ondergebracht in subparagrafen, waarin de bijbehorende activiteiten voor de betreffende fase zijn weergegeven. Paragraaf 4.3 richt zich uitsluitend op de tijdbeheersing met betrekking tot het project. Een gedetailleerde planning is dan uiteraard onmisbaar en wordt zodoende in deze paragraaf weergegeven. De laatste paragraaf in dit hoofdstuk bevat een beschrijving van de situatie zoals aanvankelijk aan de start van het project.

4.1 Methoden en technieken

4.1.1 Technieken

Technieken die de revue zullen passeren tijdens het project worden hieronder belicht.

Analyseren
Analyseren kan gedefinieerd worden als het signaleren van problemen, het leggen van verbanden tussen gegevens, het herkennen van belangrijke informatie en het ontleden in factoren en opsporen van mogelijke oorzaken. Het toepassen van analyses tijdens dit project is een belangrijke techniek voor het in kaart brengen van bepaalde zaken en het vergaren van informatie.

Rapporteren
Rapporteren is een vorm van (schrifelijke) communicatie, waarbij de schrijver de zender is en de lezers de ontvangers. Om de gehanteerde werkwijze, behaalde resultaten en gebruikte methoden tijdens het project over te brengen op belanghebbenden, is het van belang dat rapporteren wordt toegepast voor het samenstellen van rapporten, verslagen, e.d.

Unified Modelling Language
De Unified Modelling Language (UML) is een modelleringstechniek voor het in kaart brengen van informatiesystemen bij object georiënteerde systeemontwikkeling. UML bestaat hiervoor uit een set van diagrammen en een definitie van een taal, zodat beperkingen, condities en regels die van toepassing zijn op de diagrammen, weergegeven kunnen worden.
Het toepassen van UML in dit project richt zich voornamelijk op de activiteiten met betrekking tot het ontwerpen van de software-bouweenheden.

Prototyping
Prototyping is een techniek die met name gericht is op het maken van conceptuele versies van het te ontwikkelen informatiesysteem. Dit kan zijn in de vorm van een conceptuele versie van het uiteindelijke systeem, een ontwikkeld deel van het systeem na de ontwikkeling van een pilot of ontwerpen van het systeem. Het toepassen van prototyping tijdens dit project is met name gericht op de tussentijdse goedkeuring van ontwikkelde software-bouweenheden.

Interviewen
Interviewen is het techniek voor het vergaren van informatie door middel van het houden van een vraaggesprek. Het toepassen van deze techniek tijdens dit project is met name gericht op het verzamelen van de eisen/wensen met betrekking tot de opdrachtgever. Daarnaast zal de techniek mogelijk gebruikt worden voor het inventariseren van de ervaringen van de gebruikers met het ontwikkelde systeem.

© 2005 Mathieu Brüning (20010019)

MoSCoW-principe
Het MoSCoW-principe is een techniek welke gebruikt kan worden voor het indelen van systeemeisen op prioriteit. Het indelen van systeemeisen op prioriteit speelt namelijk een belangrijke rol bij het structureren van de ontwikkeling van een project. Het MoSCoW principe berust op vier categorieën eisen aan een te ontwikkelen systeem, namelijk:

1. **Must haves**: de eisen aan het systeem die sowieso gerealiseerd moeten worden bij de ontwikkeling
2. **Should haves:** de systeemeisen die feitelijk ook moeten worden gerealiseerd, maar nog wel via work-arounds te realiseren zijn
3. **Could haves:** de eisen die gerealiseerd kunnen worden wanneer er nog voldoende tijd voor is tijdens de ontwikkeling van het systeem
4. **Would like to have:** de systeemeisen die feitelijk niet behoren tot de doelstellingen van de ontwikkeling.

4.1.2 Methode (Iterative Application Development)

IAD is een methodiek voor het beheersbaar maken van een ontwikkeltraject. Deze methodiek is van belang binnen het afstudeertraject, omdat er een concrete ontwikkelfase aanwezig zal zijn. IAD schrijft onder andere voor dat een ontwikkeltraject onderverdeeld moet worden in een definitiestudie, een pilotontwikkelingsfase en een invoer/nazorg fase. De definitiestudie kenmerkt zich met name door het beschrijven van het te ontwikkelen systeem en het vastleggen van functionaliteiten en systeemeisen. De ontwikkelfase volgens de IAD methodiek richt zich op het ontwerpen en ontwikkelen van het systeem, waarbij de participatie van een opdrachtgever en de gebruikers een belangrijke rol speelt. De invoer/nazorg fase richt zich op de afronding van het project en de implementatie van het ontwikkelde systeem.

De motivatie voor het toepassen van de IAD methodiek in mijn afstudeerproject kunt u vinden in § 5.3.1.

4.2 Activiteiten

4.2.1 Oriëntatie / initiatiefase

De initiatiefase kan globaal worden omschreven als de oriëntatiefase. Een belangrijke aspect van de initiatiefase is dan ook dat in deze fase de ideeën omtrent het project worden gevormd. Het vaststellen van de doelstellingen evenals de effecten die het project moet bewerkstelligen, vindt plaats in de initiatiefase.

De onderstaande activiteiten vonden plaats in de initiatiefase:

- ✓ Oriënteren op de organisatie;
- ✓ Oriënteren op het huidige Marti-IT Update Centrum;
- ✓ Vormen van ideeën omtrent het nieuwe systeem;
- ✓ Samenstellen Plan van aanpak.

4.2.2 Definitiefase

Het volledig definiëren en uitkristalliseren van alle aspecten omtrent het project vind plaats in de definitiefase. Centraal staat het projectresultaat, waarop vervolgens bekeken wordt hoe het betreffende resultaat bereikt kan worden.

De bepalingen, resulterend uit de definitiefase, worden samengebundeld in een rapport met de resultaten van de definitiestudie. Dit rapport in combinatie met het plan van aanpak vormt de basis van het gehele project. De activiteiten van de definitiefase worden hieronder opgesomd:

- ✓ Definiëren ontwikkelscenario;
- ✓ Analyseren bestaande Marti-IT Update Centrum;
- ✓ Analyseren Update Centrums andere softwarepakketten;
- ✓ Definiëren systeemeisen;
- ✓ Bepalen systeemconcept;
- ✓ Beschouwen technische structuur;
- ✓ Beschouwen organisatorische inrichting;
- ✓ Opstellen pilotplan.

© 2005 Mathieu Brüning (20010019)

4.2.3 Ontwikkelingsfase

De ontwikkelingsfase draagt zorg voor de verwerkelijking van de bepalingen vastgesteld in de definitiefase. Samenvattend bestaat de ontwikkelingsfase dus simpelweg uit het uitvoeren van het plan van aanpak en het pilotplan. Met behulp van mijlpaalproducten wordt in stapjes het uiteindelijke resultaat behaald.

Onderstaand worden de activiteiten weergegeven die plaatsvonden in het ontwikkelingstraject:

- ✓ Eventueel bijstellen van het pilotplan
- ✓ Specificeren globaal-functionele structuur;
- ✓ Specificeren globaal-technische structuur;
- ✓ Specificeren globaal-organisatorische structuur;
- ✓ Opstellen pilotontwikkelplan;
- ✓ Ontwerpen software-bouweenheden;
- ✓ Bouwen software-bouweenheden;
- ✓ Integreren van de ontwikkelde bouweenheden;
- ✓ Testen en beoordelen pilot.

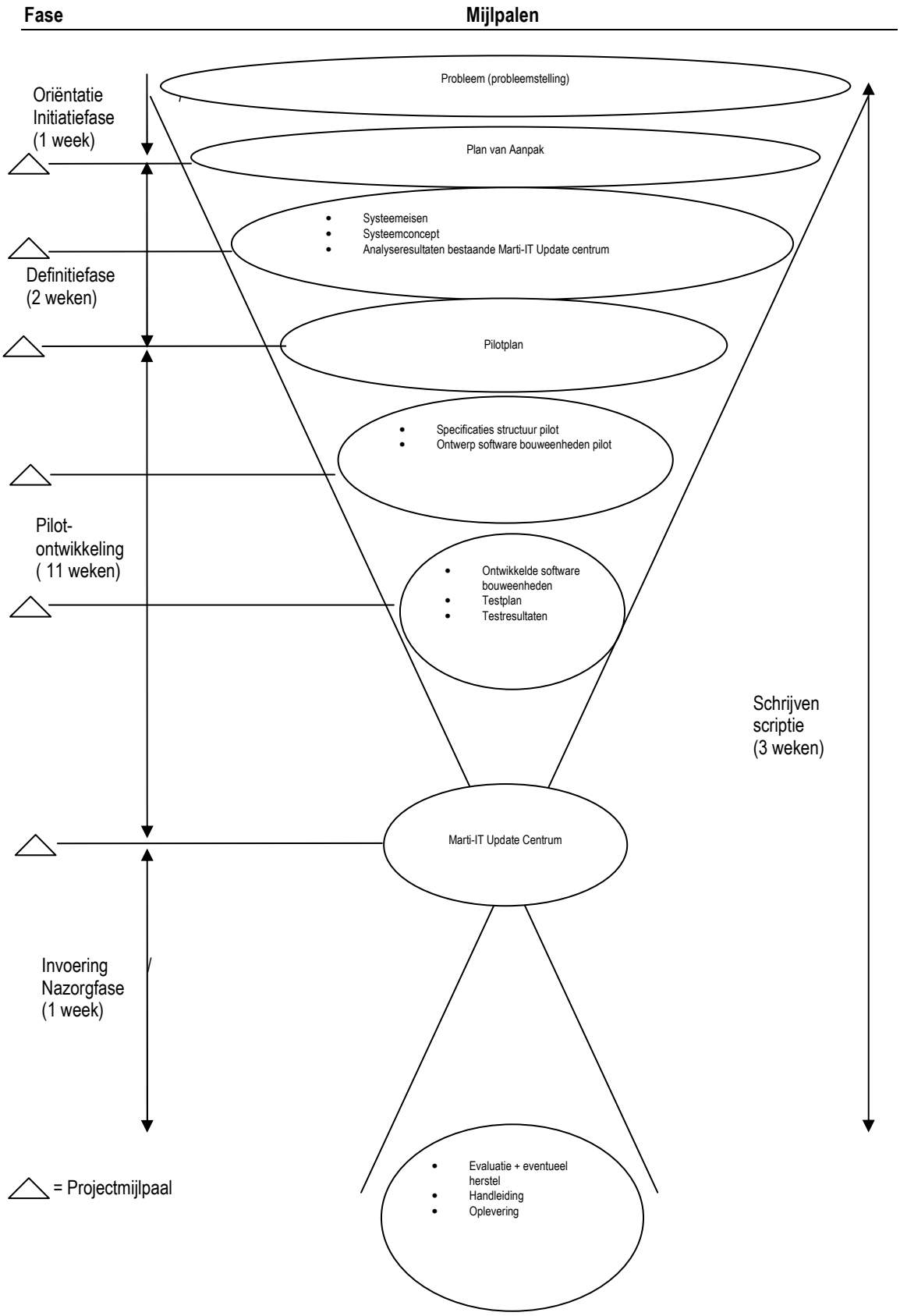
4.2.4 Invoering / nazorgfase

De laatste fase van het project, de invoering / nazorgfase, richt zich voornamelijk op de evaluatie van het project. Op die manier kunnen geleerde lessen uit het project geformuleerd worden, zodat er in de toekomst rekening mee gehouden kan worden.

Onderstaand worden de activiteiten weergegeven die plaatsvonden in de invoering / nazorgfase

- ✓ Opstellen handleiding (reference card) voor gebruikers van de Marti-IT software
- ✓ Opstellen handleiding (reference card) voor medewerkers Marti-IT
- ✓ Acceptatietest met de gebruikers van de Marti-IT software;
- ✓ Acceptatietest met de medewerkers van Marti-IT;
- ✓ Oplevering Marti-IT Update Centrum aan opdrachtgever;
- ✓ Oplevering applicatie documentatie aan opdrachtgever;
- ✓ Evalueren afstudeerperiode op zowel product- als procesmatig niveau;
- ✓ Schrijven scriptie.

© 2005 Mathieu Brüning (20010019)



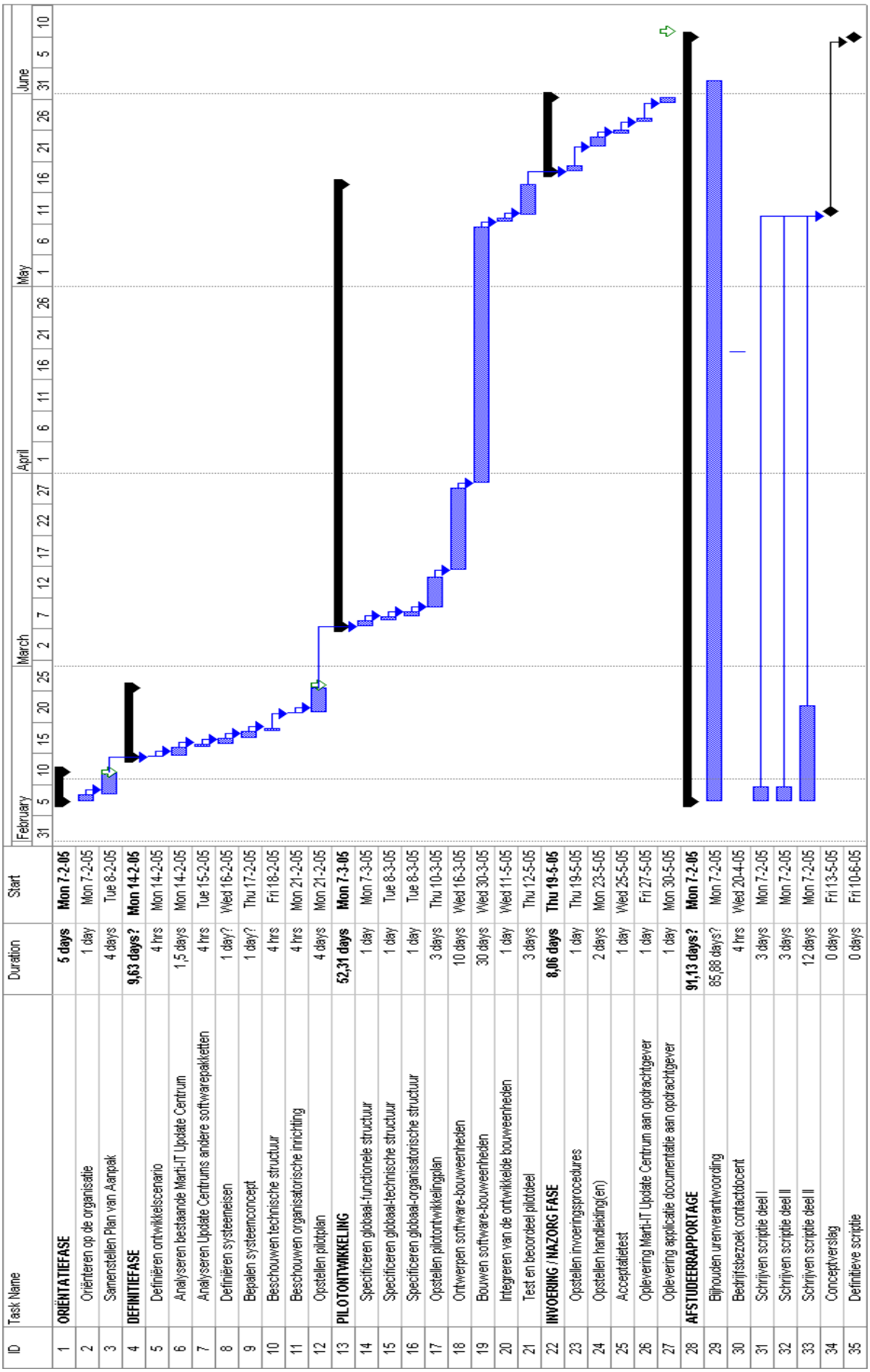
Figuur 2: *Fasering met betrekking tot het afstudeertraject*

4.3 Planning

- ✓ Voor de planning van dit project is er rekening gehouden met een 40-urige werkweek.

Zie volgende pagina voor een planning van de activiteiten en een schematische weergave van het project..

ID	Task Name	Duration	Start
1	**ORIËNTATIEFASE**	**5 days**	**Mon 7-2-05**
2	Oriënteren op de organisatie	1 day	Mon 7-2-05
3	Samenstellen Plan van Aanpak	4 days	Tue 8-2-05
4	**DEFINITIEFASE**	**9,63 days?**	**Mon 14-2-05**
5	Definiëren ontwikkelscenario	4 hrs	Mon 14-2-05
6	Analyseren bestaande Marti-IT Update Centrum	1,5 days	Mon 14-2-05
7	Analyseren Update Centrums andere softwarepakketten	4 hrs	Tue 15-2-05
8	Definiëren systeemeisen	1 day?	Wed 16-2-05
9	Bepalen systeemconcept	1 day?	Thu 17-2-05
10	Beschouwen technische structuur	4 hrs	Fri 18-2-05
11	Beschouwen organisatorische inrichting	4 hrs	Mon 21-2-05
12	Opstellen pilotplan	4 days	Mon 21-2-05
13	**PILOTONTWIKKELING**	**52,31 days**	**Mon 7-3-05**
14	Specificeren globaal-functionele structuur	1 day	Mon 7-3-05
15	Specificeren globaal-technische structuur	1 day	Tue 8-3-05
16	Specificeren globaal-organisatorische structuur	1 day	Tue 8-3-05
17	Opstellen pilotontwikkelingplan	3 days	Thu 10-3-05
18	Ontwerpen software-bouweenheden	10 days	Wed 16-3-05
19	Bouwen software-bouweenheden	30 days	Wed 30-3-05
20	Integreren van de ontwikkelde bouweenheden	1 day	Wed 11-5-05
21	Test en beoordeel pilotdeel	3 days	Thu 12-5-05
22	**INVOERING / NAZORG FASE**	**8,06 days**	**Thu 19-5-05**
23	Opstellen invoeringsprocedures	1 day	Thu 19-5-05
24	Opstellen handleiding(en)	2 days	Mon 23-5-05
25	Acceptatietest	1 day	Wed 25-5-05
26	Oplevering Marti-IT Update Centrum aan opdrachtgever	1 day	Fri 27-5-05
27	Oplevering applicatie documentatie aan opdrachtgever	1 day	Mon 30-5-05
28	**AFSTUDEERRAPPORTAGE**	**91,13 days?**	**Mon 7-2-05**
29	Bijhouden urenverantwoording	85,88 days?	Mon 7-2-05
30	Bedrijfsbezoek contactdocent	4 hrs	Wed 20-4-05
31	Schrijven scriptie deel I	3 days	Mon 7-2-05
32	Schrijven scriptie deel II	3 days	Mon 7-2-05
33	Schrijven scriptie deel II	12 days	Mon 7-2-05
34	Conceptverslag	0 days	Fri 13-5-05
35	Definitieve scriptie	0 days	Fri 10-6-05



Figuur 3: *Gedetailleerde tijdsplanning project*

4.4 Aanvangssituatie

De aanvangssituatie aan de start van het project kenmerkte zich hoofdzakelijk door het bestaan van een systeem, het zogenaamde Marti-IT Update Centrum. Dit systeem is een communicatietool voor het verzenden en ontvangen van gegevens van en naar de Marti-IT klanten. Het systeem was in het verleden ontwikkeld zodat bestanden op eenvoudige wijze geplaatst en opgehaald konden worden bij gebruikers.

Daarnaast kenmerkte de aanvangssituatie zich voornamelijk door het ontbreken van beschikbare documentatie rondom het bestaande systeem. Doordat het systeem in het verleden in vrij korte tijd was ontwikkeld was er niet echt sprake van een gestructureerde ontwikkeling (ontwerpfase, ontwikkelingfase, testfase, implementatiefase), maar kenmerkte de ontwikkeling van het systeem zich door het direct bouwen van de systeem. Het ontbreken van een concrete ontwerpfase voor de ontwikkeling van de applicatie heeft er dus met name voor gezorgd dat er weinig tot geen documentatie beschikbaar was omtrent het systeem.

Tot slot kenmerkte de aanvangssituatie zich door de aanwezige ideeën die er waren voor de ontwikkeling van het nieuwe Marti-IT Update Centrum. De medewerkers van Marti-IT, die al enige tijd gebruik maakte van de applicatie, hadden ook ideeën opgebouwd voor de ontwikkeling van het nieuwe systeem, zodat deze eventueel meegenomen konden worden bij het definiëren van de systeemeisen.

© 2005 Mathieu Brüning (20010019)

DEEL 2: WERKZAAMHEDEN & EVALUATIE

Het volgende deel van het verslag, deel twee: "Werkzaamheden & evaluatie", vormt de kern van het afstudeerverslag. Hierin wordt een uitgebreide beschrijving van de uitgevoerde werkzaamheden en een evaluatie van zowel de procesgang als een evaluatie van de productmatige resultaten weergegeven.

Deel twee is opgesplitst in vijf hoofdstukken, waarin het eerste hoofdstuk, hoofdstuk 5, een beschrijving weergeeft van de gehanteerde aanpak tijdens het project. Hoofdstuk zes bevat een uitgebreide beschrijving van de verrichtte werkzaamheden, welke de revue zijn gepasseerd tijdens het afstudeertraject. De hoofdstukken zeven en acht bevatten respectievelijk een analyse van de projectresultaten en een overzicht van de opgeleverde producten. Het laatste hoofdstuk in deel twee, hoofdstuk negen, bevat de evaluaties van het afstudeertraject. Het laatstgenoemde hoofdstuk is opgesplitst in twee paragrafen; de eerste paragraaf geeft de procesevaluatie weer, de tweede paragraaf bevat de evaluatie van de producten.

© 2005 Mathieu Brüning (20010019)

5 Aanpak project

De gehanteerde aanpak tijdens het project staat centraal in dit hoofdstuk, waarbij het hoofdstuk is ingedeeld in drie paragrafen. De eerste paragraaf bevat een verwijzing naar het plan van aanpak, het document waarin de werkzaamheden, resultaten en procedures met betrekking tot het project zijn vastgelegd aan de start van het afstudeertraject. Paragraaf 5.2 belicht de gehanteerde methoden voor het bieden van een oplossing voor het probleem dat centraal stond tijdens het project. De laatste paragraaf bevat een verdediging van de gekozen aanpak, waarin wordt uiteengezet wat de motivatie is voor de gehanteerde aanpak.

5.1 Plan van aanpak

Het opstellen van een plan van aanpak was een eerste stap in het project. Door het maken van een plan van aanpak konden er werkzaamheden worden gedefinieerd, zodat de structuur van het project in kaart gebracht kon worden. Daarnaast was het opstellen van een gedetailleerde planning een belangrijk aspect van het plan van aanpak om de structuur van het project in kaart te brengen.

Voor het volledige plan van aanpak verwijs ik u naar bijlage 2.

© 2005 Mathieu Brüning (20010019)

5.2 Oplossingsmethoden

5.2.1 Hergebruik

Een belangrijk aspect van de ontwikkeling van het Marti-IT Update Centrum is het toepassen van herbruikbare softwarecomponenten geweest. Het toepassen van het herbruikbare Microsoft Updater Application Block (voor meer informatie verwijs ik u naar § 6.3.1 – Zoeken naar herbruikbare componenten) draagt namelijk zorg voor een significant deel van de functionaliteit van het systeem. Met name de functionaliteiten voor het inventariseren van updates, back-ups en standaard uploads zijn gebaseerd op het herbruikbare component. Het grote voordeel van het Microsoft Application Block is dat het volledig open-source is en zodoende naar eigen inzien aangepast en uitgebreid kon worden.

Het toepassen van hergebruik tijdens de ontwikkeling van informatiesystemen moedig ik zelf aan. Met name doordat het toepassen van hergebruik of het toepassen van standaard oplossingen kan leiden tot een tijdswinst ten opzichte van het van scratch af beginnen.
Daarnaast ben ik van mening dat “het wiel opnieuw uitvinden” niet efficiënt is. Kanttekening hierbij is dat uiteraard de kwaliteit en toepasbaarheid van een herbruikbaar component in eerste instantie moet worden vastgesteld, zodat bepaald kan worden of een herbruikbaar component gebruikt zal worden.

Een zeer belangrijke eis van Marti-IT met betrekking tot het toepassen van herbruikbare componenten is dat het open-source dient te zijn. Open-source componenten kunnen namelijk naar eigen inzicht worden uitgebreid en aangepast, wat niet mogelijk is met componenten die niet open-source zijn. Indien een herbruikbaar component niet open-source is ben je als ontwikkelaar altijd afhankelijk van een derde partij, aangezien uitbreidingen en aanpassingen aan een component dan niet zelf gerealiseerd kunnen worden.

© 2005 Mathieu Brüning (20010019)

5.3 Verdediging gekozen aanpak

In deze paragraaf wordt uiteengezet wat de motivatie is achter het hanteren van de gekozen aanpak tijdens het project. Hierbij wordt de gehanteerde aanpak belicht door de diverse fasen en producten van het project toe te lichten, zodat duidelijk wordt waarom bepaalde activiteiten hebben plaatsgevonden.

Om het afstudeertraject zo goed mogelijk te laten verlaten verlopen heb ik allereerst bewust gekozen voor een gestructureerde aanpak, waarin duidelijke scheidingen zijn te onderkennen in de verschillende fasen van het project. Het structureren van het project heeft met name plaatsgevonden op basis van de Iterative Application Development (IAD) ontwikkelingsmethodiek. De keuze voor het toepassen van IAD wordt in subparagraaf § 5.3.1 belicht.

5.3.1 Iterative Application Development

De keuze voor het toepassen van de Iterative Application Development (IAD) methodiek voor de ontwikkeling van het systeem was met name gebaseerd op de volgende aspecten:

IAD kenmerkt zich door ontwikkeling in delen (pilots), waarbij de cruciale functionaliteiten in een vroeg stadium worden ontwikkeld. Het voordeel van het ontwikkelen van de belangrijkste functionaliteiten in een vroeg stadium is met name dat wanneer het project vroegtijdig wordt afgebroken, om welke reden dan ook, of wanneer er blijkt dat de opgestelde planning niet behaald kan worden, de belangrijkste kenmerken van het systeem reeds ontwikkeld zijn.

Daarnaast was een belangrijke eis van de opdrachtgever dat de tussentijdse resultaten van het project met hem zouden worden besproken. Op deze manier was de opdrachtgever in staat om de ontwikkeling van het systeem te valideren, zodat er meer zekerheid was dat het juiste systeem gebouwd werd. Door middel van besprekingen en tussentijdse presentaties aan het einde van elke pilot zijn de ontwikkelde functionaliteiten van elke pilot doorgenomen en kon er eventueel (bij)sturing plaatsvinden door de opdrachtgever.

Tot slot speelde de bekendheid van de IAD methodiek een belangrijke rol bij de keuze voor het toepassen ervan. Op basis van de vergaarde kennis omtrent systeemontwikkelingsmethodieken vanuit de opleiding bleek dat er hoofdzakelijk aandacht was besteedt aan de IAD methodiek. IAD was dan ook feitelijk de enige methodiek voor het ontwikkelen van systemen, waar grondige aandacht aan besteedt was middels de opleiding. Het kiezen van een andere systeemontwikkelingsmethodiek voor het uitvoeren van het afstudeertraject zou betekenen dat ik tijd zou verliezen aan het inlezen in de theorie van de betreffende methodiek. Het toepassen van de IAD methodiek is hier ook deels op gebaseerd.

5.3.2 Plan van aanpak

Een solide basis voor ieder project is het opstellen van een adequaat plan van aanpak. Hierin staat het gehele project, inclusief structuur, globale & gedetailleerde planning, werkzaamheden, methoden, enz. beschreven. Het plan van aanpak draagt onder meer zorg voor de bewaking van het project en een realisatie van een goede start.
Toen het plan van aanpak was opgesteld, en het project dus feitelijk was gedefinieerd, kon de daadwerkelijke uitvoering van het project volgens de IAD methodiek van start gaan.

5.3.3 Definitiestudie

Het uitvoeren van een definitiestudie speelde een belangrijke rol in de aanpak van het project volgens de IAD methodiek. Om een goede voorbereiding van het ontwikkeltraject te kunnen realiseren heb ik een definitiestudie uitgevoerd, waarin de belangrijkste kenmerken met betrekking tot het ontwikkeltraject werden vastgelegd. Het doel van het uitvoeren van een definitiestudie was dan ook het vaststellen van de doelen van het te ontwikkelen systeem, zodat duidelijk in kaart kon worden gebracht wat er ontwikkeld diende te worden en hoe de ontwikkeling plaats diende te vinden.

Het uitvoeren van de definitiestudie heeft me met name geholpen bij het vormen van een beeld omtrent het te ontwikkelen systeem, zodat voor mij duidelijk was wat de eisen en wensen waren aan het te ontwikkelen systeem.

© 2005 Mathieu Brüning (20010019)

5.3.4 Pilotplan

Een volgende belangrijke stap in mijn aanpak was het opstellen van een pilotplan, zodat de uitvoering van de pilotontwikkeling gestructureerd kon worden. Nadat de globale pilotstructuur in de definitiestudie was vastgelegd, was het uitwerken van de globale pilotstructuur tot concrete pilots en bijbehorende pilotdelen essentieel om de ontwikkeling te structureren. Het uitwerken van een pilotplan heeft een belangrijke rol gespeeld in het formuleren van hapklare componenten voor de ontwikkeling van de verscheidene pilots.

Naar mijn mening heeft het opstellen en hanteren van een pilotplan tijdens de ontwikkeling een cruciale rol gespeeld. Het ontbreken van duidelijke structuur in de pilots door middel van het ontbreken van een pilotplan zou er namelijk voor zorgen dat de ontwikkeling niet gestructureerd zou plaatsvinden. Op deze manier zouden cruciale componenten van het te ontwikkelen systeem mogelijk verkeerd kunnen worden ingedeeld in het ontwikkeltraject, in tegenstelling tot de IAD methodiek, waar de meest cruciale componenten van het systeem als eerst worden ontwikkeld.

5.3.5 Pilotontwikkeling

Op basis van de gedefinieerde structuur in het pilotplan is de ontwikkeling van de pilots van start gegaan. De aanpak met betrekking tot het ontwikkelen van de pilots richtte zich hoofdzakelijk op het uitvoeren van de pilotontwikkeling op basis van de gedefinieerde pilots en pilotdelen uit het pilotplan. Een voordeel hiervan was met name dat er ten allen tijden duidelijk was wat er ontwikkeld diende te worden.

Het ontwikkelen middels voorgedefinieerde pilots had als voornaamste voordeel dat er duidelijk was in hoeveel tijd de betreffende subsystemen ontwikkeld diende te worden

5.3.6 Testen

Een belangrijk kenmerk van de aanpak van systeemontwikkeling middels IAD is dat er na de ontwikkeling van elke pilot testwerkzaamheden uitgevoerd dienen te worden. Op deze manier kan het ontwikkelde subsysteem beoordeeld worden om te verifiëren of het ontwikkelde overeenkomt met hetgeen wat er vooraf gedefinieerd is. Het grote voordeel in deze aanpak is dat eventuele fouten op een vroeg stadium opgemerkt kunnen worden, zodat er nog tijd in gecalculeerd kan worden voor het herstel van deze fouten.

Bewust heb ik in mijn aanpak ook gekozen voor het testen van de ontwikkelde pilotdelen aan het einde van de ontwikkelde van elke pilot. Op deze manier was in namelijk in staat om te verifiëren of elk ontwikkelde subsysteem overeenkwam met hetgeen wat er vooraf gedefinieerd was voor de betreffende pilot.
Indien ik geen testwerkzaamheden zou hebben laten plaatsvinden na de ontwikkeling van elke pilot en het volledige systeem aan het einde van het ontwikkeltraject zou testen, zou ik het risico lopen om grote fouten pas aan het eind van het ontwikkeltraject te ontdekken. Deze fouten zouden dan mogelijk niet meer opgelost kunnen worden door middel van tijdsgebrek.

5.3.7 Conclusie gekozen aanpak

Het hanteren van een duidelijke, gestructureerde aanpak door middel van het onderkennen duidelijke scheidingen van de diverse fasen in het project heeft voornamelijk bijgedragen aan een structurele uitvoering van het project. Deze gestructureerde aanpak zorgde er gedurende het project voor dat de fasering en bijbehorende activiteiten ten allen tijden duidelijk waren. Hierbij droeg in eerst instantie het opstellen van een plan van aanpak zorg voor de bewaking van het project en de realisatie van een goede start. Het uitvoeren van de definitiestudie was vervolgens een essentieel onderdeel van mijn gehanteerde aanpak, om zo de doelen van het te ontwikkelen systeem vast te leggen en de ontwikkeling voor te bereiden.
Door middel van het opstellen van een pilotplan kon de ontwikkeling worden uitgekristalliseerd en werd duidelijk wat er ontwikkeld diende te worden en hoe de ontwikkeling plaats diende te vinden. De gehanteerde pilotstructuur tijdens de ontwikkeling van het systeem speelde een belangrijke rol in de het ontwikkeling van de componenten op basis van prioriteit.
Het testen van de ontwikkelde componenten per pilot, droeg met name zorg voor het tijdig opsporen van fouten, zodat deze in een volgend stadium hersteld konden worden.

© 2005 Mathieu Brüning (20010019)

6 Werkwijzebeschrijving

Een uitvoerige beschrijving van de werkzaamheden, welke hebben plaatsgevonden tijdens het afstudeertraject, worden vermeldt in dit hoofdstuk. Het hoofdstuk is hierbij onderverdeeld in vier paragrafen, waarbij elke paragraaf een fase van het project representeert.

6.1 Oriëntatie / initiatiefase

6.1.1 Oriënteren op de organisatie

De eerste week bij Marti-IT was met name een oriënterende week. In eerste instantie diende mijn werkplek ingericht te worden om de werkzaamheden van het project mee uit te kunnen voeren. Nadat ik mijn draai had gevonden binnen de organisatie heb ik een gesprek gehad met mijn bedrijfsmentor, waarin hij het een en ander uitlegde omtrent Marti-IT als organisatie en omtrent de uit te voeren opdracht. Daarnaast vertelde hij me dat hij verwachtte dat ik volledig zelfstandig aan de slag zou gaan met het uitvoeren van de opdracht. Vervolgens vroeg hij hoe de globale structuur van het project eruit zou komen te zien en waar ik me in eerste instantie mee bezig zou gaan houden. Ik vertelde hem dat ik in eerste instantie een plan van aanpak zou maken om een solide basis van het project te kunnen realiseren. Met dit gegeven ben ik aan de slag gegaan met het maken van een plan van aanpak.

6.1.2 Plan van aanpak

Het opstellen van het plan van aanpak diende in het verloop van het verdere project als basis te dienen voor het uitvoeren van de afstudeeropdracht en het nastreven van de planning. Aan de hand van de opdrachtomschrijving behorend bij het afstudeertraject ben ik begonnen met het formuleren van de projectdefinitie. Zaken als de probleemstelling, aanleiding, doelstelling en het projectresultaat heb ik geformuleerd op basis van de opdrachtomschrijving. Na het definiëren van de projectdefinitie ben ik aan de slag gegaan met het uiteenzetten van de uit te voeren werkzaamheden tijdens het project. Op basis van de IAD theorie ben ik gaan inventariseren welke activiteiten volgens IAD toepasbaar waren en welke niet. Het resultaat was een uiteenzetting van de uit te voeren activiteiten in een zogenaamde Work Breakdown Structure (zie Plan van Aanpak, § 3.1). Deze Work Breakdown Structure bevat een overzicht van de werkzaamheden van het project, weergegeven in schematische vorm.
Nadat een overzicht van de uit te voeren activiteiten was geformuleerd heb ik mezelf gericht op het beschrijven van de mijlpaalproducten & -activiteiten van het afstudeertraject. Toen een duidelijk overzicht gemaakt was van de uit te voeren activiteiten ben ik aan de slag gegaan met het maken van een planning. In eerste instantie heb ik een globale planning geformuleerd, waarin met name de fasering van het project wordt belicht. Op basis van de globale planning was ik in staat om middels Microsoft Project een gedetailleerde planning te formuleren. Hierin staan de concrete activiteiten van het traject opgesomd, gekoppeld aan een tijdsplanning.

Nadat de projectdefinitie, de fasering, de activiteiten en de planning in kaart waren gebracht, waren de essentiële aspecten van het plan van aanpak geformuleerd. Het plan van aanpak heb ik afgerond door de overige zaken als inleiding, voorwoord, en dergelijke te formuleren. Het volledige plan van aanpak is overigens te vinden in bijlage 2.

Het oriënteren op Marti-IT en het maken van een plan van aanpak heeft één week gekost, zoals van tevoren gepland.

© 2005 Mathieu Brüning (20010019)

6.2 Definitiefase

6.2.1 Definiëren ontwikkelscenario

Nadat het plan van aanpak was opgesteld ben ik aan de slag gegaan met het uitvoeren van een definitiestudie volgens IAD. Voor de definitiestudie had ik een tijdsduur van 2 weken ingepland. De definitiefase kenmerkte zich in eerste instantie door het vastleggen van het ontwikkelscenario. Het vastleggen van het ontwikkelscenario richtte zich hoofdzakelijk op het beschrijven van diverse aspecten van het project, zoals bijvoorbeeld de projectkenmerken, de impact van het project, de kenmerken van het ontwikkelproces en de toe te passen strategieën. Ook hier ben ik weer op basis van de informatie uit de opdrachtomschrijving aan de slag gegaan met het formuleren van het ontwikkelscenario.

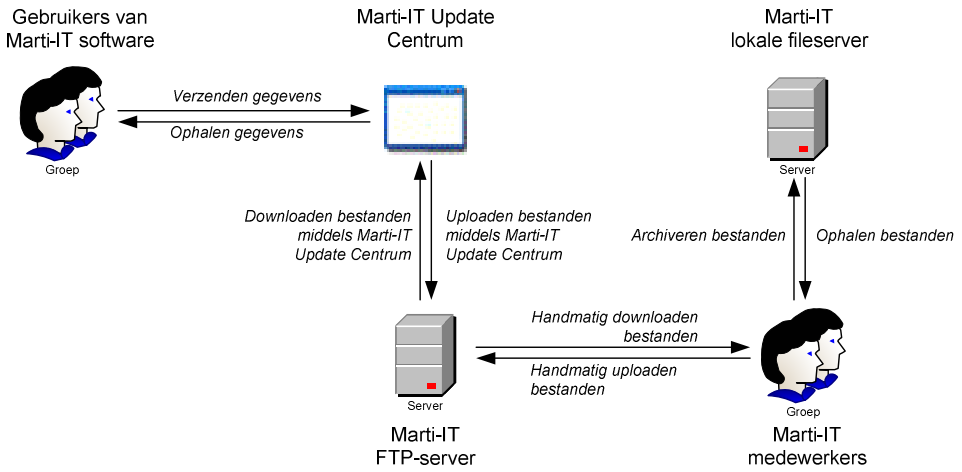
Het resultaat van het definiëren van het ontwikkelscenario betrof de vastlegging van de volgende aspecten:

- ✓ De projectkenmerken als doelstellingen en reikwijdte
- ✓ De impact van het project
- ✓ De cruciale succesfactoren van het project
- ✓ De kenmerken rondom het ontwikkelproces
- ✓ De projectorganisatie
- ✓ De globale pilotstrategie
- ✓ De globale strategie rondom hergebruik en externe-acquisitie
- ✓ De globale teststrategie
- ✓ De beheersstrategie

Het volledige resultaat van de definiëring van het ontwikkelscenario is te vinden in hoofdstuk 2 van het rapport met de specificaties van het Marti-IT Update / Build Centrum, bijlage 3 van dit rapport.

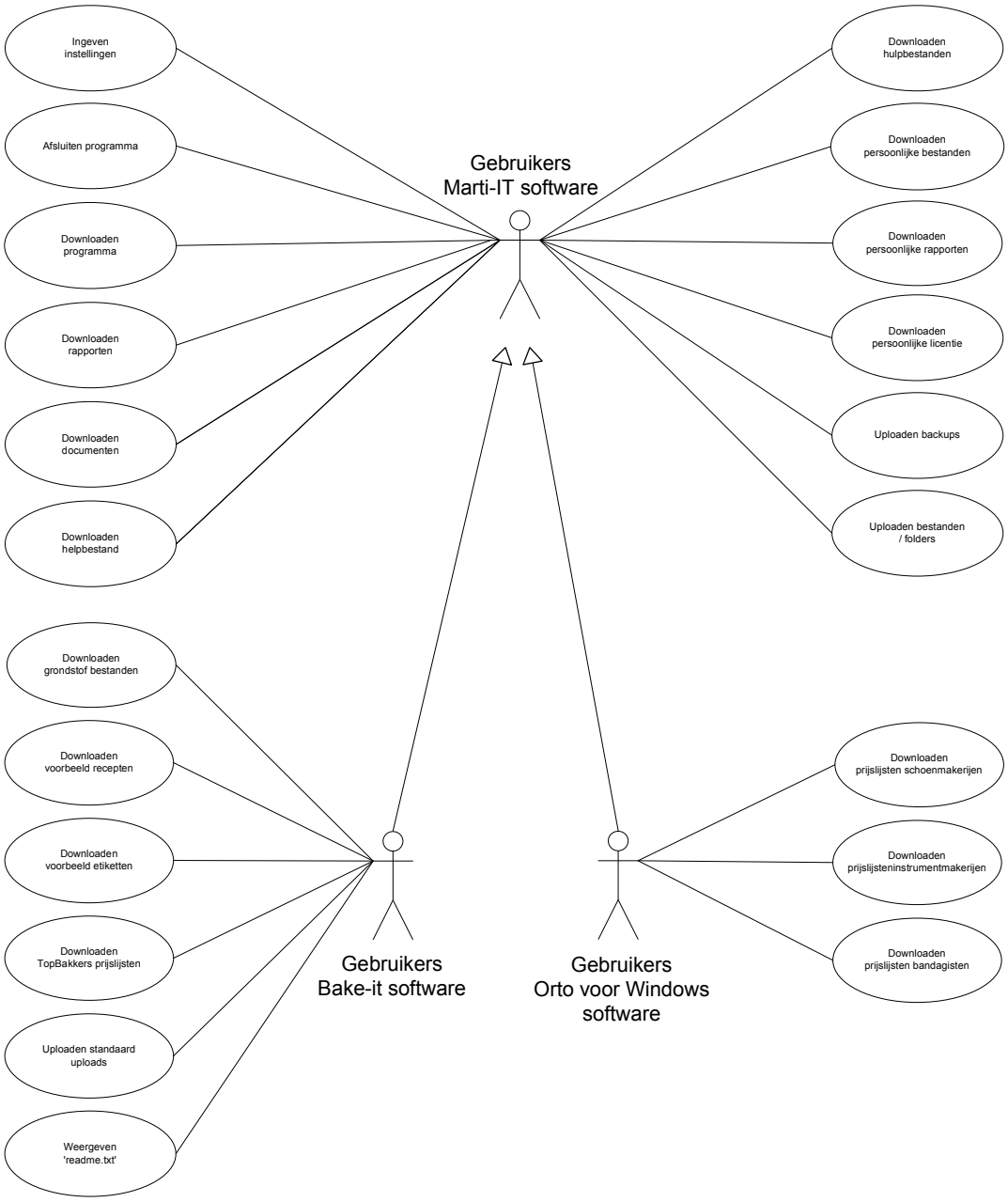
6.2.2 Analyseren van het bestaande Marti-IT Update Centrum

De volgende activiteit in de definitiefase richtte zich op het analyseren van het bestaande Marti-IT Update Centrum. Met name door het ontbreken van documentatie omtrent de bestaande applicatie diende het huidige Update Centrum geanalyseerd te worden. Op deze manier zouden diverse aspecten van het bestaande systeem, zoals bijvoorbeeld functionaliteit, structuur en samenhang, maar ook positieve en negatieve aspecten, in kaart kunnen worden gebracht.
Het analyseren van het bestaande Update Centrum richtte zich in eerste instantie op het verdiepen in de historie van de applicatie. Met name door informatie te vergaren van de ontwikkelaar van het systeem, was ik in staat om een beeld te vormen omtrent het doel van de ontwikkeling van het Update Centrum. Op basis van de vergaarde informatie ben ik aan de slag gegaan met het beschrijven van de historie van de applicatie. Nadat ik de historie van het systeem in kaart had gebracht heb ik een model opgesteld waarin de algemene werking van het systeem wordt weergegeven. Onderstaand wordt dit model getoond.



Figuur 4: *Globale werking bestaande Marti-IT Update Centrum*

Nadat de globale werking van de applicatie is kaart gebracht was ben ik verder gegaan met het inventariseren van de functionaliteiten. Door alle mogelijke gebruikersfunctionaliteiten uit te voeren en op te sommen was ik in staat om een overzicht te maken van de mogelijke functies van de applicatie. Daarnaast kreeg ik toegang tot de broncode van het systeem en ben ik verder gegaan met het inventariseren van functionaliteiten. Door alle gevonden functionaliteiten op te sommen was een overzicht van alle functionaliteiten het resultaat. De functionaliteiten heb ik vervolgens in kaart gebracht door een use-case diagram te ontwikkelen.



Figuur 5: *Use-case diagram met betrekking tot de functionaliteit van het bestaande Marti-IT Update Centrum*

De laatste activiteiten met betrekking tot het analyseren van het bestaande Update Centrum richtte zich op het toepassen van een SWOT-analyse. Door een SWOT-analyse toe te passen konden namelijk de positieve en negatieve aspecten van de bestaande applicatie in kaart worden gebracht, zodat hier rekening mee gehouden kon worden bij de ontwikkeling van het nieuwe systeem. Op basis van informatie van de medewerkers van Marti-IT en eigen ervaringen heb ik een SWOT-analyse kunnen toepassen op het bestaande Marti-IT Update Centrum.
De resultaten van de analyse zijn te vinden in hoofdstuk 3 van bijlage 3.

6.2.3 Analyseren Update Centrums andere softwarepakketten

Naast de analyse van het bestaande Update Centrum heb ik besloten om tevens SWOT-analyses toe te passen op Update Centrums van andere softwarepakketten. Op die manier zouden positieve en negatieve aspecten van de andere applicaties in kaart kunnen worden gebracht, zodat deze ook eventueel meegenomen zouden kunnen worden bij de ontwikkeling van het nieuwe Update Centrum. Bij het analyseren van de andere Update Centrums heb ik een drietal systemen aan een SWOT-analyse onderworpen. Het betroffen de volgende systemen: Microsoft Windows Update, Norton LiveUpdate en de mogelijkheid tot het updaten van Cyberlink PowerDVD. De keuze voor de systemen berustte op de volgende gedachtegang: in eerste instantie wilde ik de Update Centrums analyseren welke wereldwijd in grote mate gebruikt worden. Op basis van eigen ervaringen bleek dat Microsoft Windows Update en Norton LiveUpdate wereldwijd in zeer grote mate gebruikt worden voor het updaten van software. Deze twee systemen wilde ik dus aan een analyse onderwerpen. Bij het derde systeem dat ik wilde analyseren had ik juist de gedachte om een systeem te kiezen welke wereldwijd in veel mindere mate wordt gebruikt dan de twee eerder genoemde. De keuze is toen gevallen op het updatetool van Cyberlink PowerDVD. Op basis van eigen ervaringen door de systemen uitvoerig te gebruiken was ik in staat om de positieve als negatieve aspecten van de systemen in kaart te brengen in de SWOT-analyses.

De resultaten van de analyses zijn de vinden in hoofdstuk vier van het rapport met de specificaties van het Marti-IT Update / Build Centrum.

6.2.4 Definiëren systeemeisen

Toen het ontwikkelscenario was vastgelegd en de positieve en negatieve aspecten van het bestaande Update Centrum en andere Update Centrums in kaart waren gebracht, richtte mijn werkzaamheden zich op het definiëren van systeemeisen. Het definiëren van de systeemeisen heeft hoofdzakelijk plaatsgevonden op basis van twee kenmerken, namelijk enerzijds op basis van de opdrachtomschrijving en anderzijds op basis van een kleine enquête onder te Marti-IT medewerkers. Door namelijk een mini-enquête te houden omtrent het bestaande Update Centrum wilde ik informatie vergaren omtrent de eisen en wensen van de medewerkers van Marti-IT. De enquête was als volgt opgezet: in eerste instantie stonden de reeds geformuleerde systeemeisen (op basis van de opdrachtomschrijving en bestaande functionaliteiten) opgesomd. De medewerkers diende de systeemeisen een prioriteit van 1 tot en met 5 (1 is het hoogst) toe te kennen, zodat er duidelijk naar voren zou komen welke systeemeisen er in ieder geval gewaarborgd diende te blijven. Daarnaast was er de mogelijkheid om zelf systeemeisen te formuleren, zodat de medewerkers hun eventuele ontbrekende functionaliteiten aan het systeem konden meedelen. Nadat de enquêtes waren ingevuld was ik in staat om op eenvoudige wijze te achterhalen welke systeemeisen men het belangrijkst vond en welke ontbrekende functionaliteiten er nog gewenst waren. Op basis van de resultaten van de enquête kon ik vervolgens een overzicht maken van de systeemeisen, geordend op prioriteit.
Vervolgens ben ik aan de slag gegaan met de consolidatie van de systeemeisen, op basis van de toegekende prioriteiten. Door het MoSCoW-principe toe te passen, waarbij de systeemeisen worden onderverdeeld in vier groepen, was ik in staat om een duidelijk overzicht te maken van de relevantie van de verscheidene systeemeisen. Door de geprioriteerde lijst van systeemeisen in te delen in de groepen volgens het MoSCoW-principe, kon er duidelijk vastgelegd worden welke systeemeisen in welke pilot aan bod zouden komen. Het indelen van de systeemeisen heb ik echter niet in vier groepen gedaan, maar in drie groepen. De groep met de laagste prioriteit volgens het MoSCoW-principe heb ik achterwege gelaten, aangezien ik van mening was dat alle geformuleerde systeemeisen, waar mogelijk, ontwikkeld diende te worden. Een overzicht van de indeling van de systeemeisen volgens het MoSCoW-principe is op de volgende twee pagina's weergegeven.

© 2005 Mathieu Brüning (20010019)

Must have

Systeemeis(en)
De bestaande functionaliteit van het huidige Marti-IT Update Centrum, bestaande uit: ▪ Downloaden programma ▪ Uploaden backups ▪ Downloaden programma updates ▪ Uploaden backups ▪ Downloaden rapporten ▪ Downloaden persoonlijke rapporten ▪ Downloaden persoonlijke licentie ▪ Uploaden bestanden / folders ▪ Downloaden persoonlijke bestanden ▪ Downloaden documenten ▪ Downloaden hulp bestanden ▪ Downloaden Bake-it grondstof bestanden ▪ Downloaden Bake-it TopBakkers prijslijsten ▪ Downloaden Orto prijslijsten schoenmakerijen ▪ Downloaden Orto prijslijsten instrumentmakerijen ▪ Downloaden Orto prijslijsten bandagisten ▪ Downloaden Bake-it voorbeeld recepten ▪ Downloaden Bake-it voorbeeld etiketten ▪ Downloaden helpbestand ▪ Uploaden standaard uploads Ontwikkelingomgeving: ▪ Het nieuwe Marti-IT Update Centrum dient geschreven te zijn in de .NET programmeertaal C#

Should have

Systeemeis(en)
▪ Gegevens vastleggen omtrent beschikbare updates ▪ Inventariseren van beschikbare downloads voor gebruiker ▪ Downloaden (alle) beschikbare updates ▪ Autorisatie van een gebruiker via een beveiligde verbinding ▪ Vastleggen van gegevens van de gebruiker ▪ Wijzigen gegevens gebruikers softwaregebruik ▪ Ophalen locaties / versienummers software vanuit register ▪ Maken van een back-up ▪ Nieuwe dynamische menustructuur ▪ Nieuwspagina's in html formaat, die worden getoond tijdens het downloaden van de updates ▪ Vastleggen historie gegevens omtrent het downloaden van de updates van de gebruiker ▪ Klaarzetten van programma updates ▪ Controleren gebruikersinformatie ▪ Extra functionaliteiten op het gebied van .zip

© 2005 Mathieu Brüning (20010019)

Could have

Systeemeis(en)
▪ Downloaden historische updates ▪ Updates ophalen vanaf lokale en/of netwerklocaties in plaats van het Internet ▪ Automatisch downloaden van back-ups naar Marti-IT lokale server ▪ Weergeven informatie Update Centrum ▪ Extra functionaliteiten op het gebied van commandline opties ▪ Nieuwe kleurensamenstelling

Would like to have

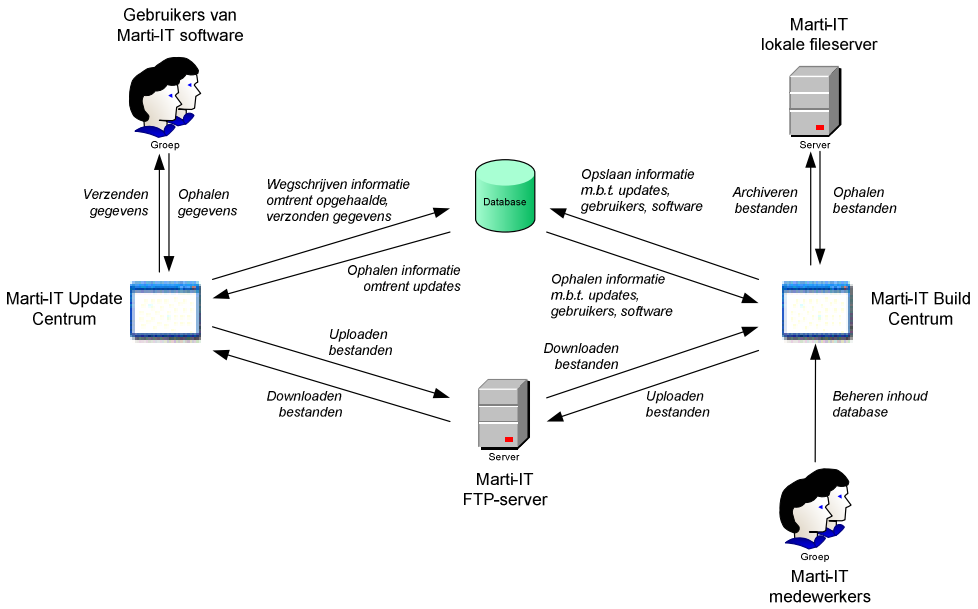
Aangezien ik van mening ben dat alle gedefinieerde systeemeisen ontwikkeld diende te worden heb ik besloten dat er niet echt sprake was van zogenaamde 'Would like to have' systeemeisen.

6.2.5 Bepalen systeemconcept

De volgende activiteit in de definitiestudie was het definiëren van het systeemconcept. In dit systeemconcept is onder andere aandacht besteedt aan het specificeren van actoren, het maken van een procesmodel (onder andere use-cases scenario's) en het maken van een gegevensmodel. In eerste instantie ben ik aan de slag gegaan met het specificeren van actoren, welke de functionarissen representeren van het te ontwikkelen systeem. Het specificeren van de actoren richtte zich hoofdzakelijk op het analyseren van de betrokkenen rondom het Marti-IT Update Centrum en het beschrijven van deze actoren. Het resultaat betrof een drietal actoren, namelijk:

- ✓ gebruikers van de Marti-IT software
- ✓ medewerkers van Marti-IT
- ✓ software ontwikkelaars / beheerders van het Marti-IT Update / Build Centrum

De volgende stap in het definiëren van het systeemconcept was het vastleggen van events. Het vastleggen van events kenmerkte zich met name door een opsomming te maken van de verscheidene functionaliteiten die de verschillende actoren gebruiken. Een belangrijke stap bij het definiëren van het systeemconcept was het formuleren van een globaal procesmodel. De eerste stap van het globale procesmodel richtte zich op het in kaart brengen van de globale werking van het nieuwe systeem. De onderstaande afbeelding geeft deze globale werking weer.



Figuur 6: *Globale werking van het nieuwe systeem*

Op basis van de systeemeisen ben ik vervolgens aan de slag gegaan met het formuleren van use-case scenario's voor de verscheidene systeemeisen. Deze use-case scenario's waren grotendeels gebaseerd op bestaande functionaliteiten, aangezien de bestaande functionaliteit van de applicatie gewaarborgd diende te blijven. De use-case scenario's die niet waren gebaseerd op bestaande functionaliteiten heb ik vervolgens zelf geformuleerd. Het use-case scenario's zijn te vinden in het rapport met de specificaties van het systeem § 6.3.2.
Nadat de scenario's geformuleerd waren ben ik aan de slag gegaan met het in kaart brengen van de use-case scenario's middels use-case diagrammen. Bij het maken van een use-case diagrammen heb ik een duidelijke scheiding gehanteerd tussen de verschillende actoren, zoals gedefinieerd in het specificatierapport § 6.1
Toen de use-case scenario's waren vastgelegd en de use-case diagrammen ontwikkeld waren ben ik aan de slag gegaan met het maken van een globaal gegevensmodel. Het globale gegevensmodel bevat in grote lijnen een overzicht van de te verwerken gegevens tijdens het gebruik van het systeem. Op basis van de analyseresultaten van het bestaande systeem en reeds gevormde eigen ideeën was ik in staat om de primaire entiteittypen te formuleren en een globaal klassendiagram te maken. De primaire entitteittypen betroffen:

- ✓ gebruikers
- ✓ software
- ✓ updates

Het globale gegevensmodel was hiermee gedefinieerd. Tot slot richtte het definiëren van het systeemconcept zich op het specificeren van systeem-interfaces en koppelingen met eventuele andere systeem-interfaces.

6.2.6 Bepalen technische en organisatorische structuur

De laatste activiteiten van de definitiestudie richtten zich op het beschouwen van de technische structuur en de organisatorische inrichting. Hoofdzakelijk richtte deze activiteiten zich op het beschrijven van zowel de technische kenmerken als de organisatorische kenmerken van het te ontwikkelen systeem. Op basis van de reeds vergaarde informatie door middel van de opdrachtgever, de medewerkers van Marti-IT en de gedefinieerde systeemeisen was ik in staat om de technische en organisatorische kenmerken te beschrijven.

6.2.7 Opstellen van een pilotplan

Na het uitvoeren van de activiteiten van de definitiestudie ben ik aan de slag gegaan met het maken van een pilotplan, waarin de structuur en uitvoering van de pilots nauwkeurig in beschreven staat. Op basis van de informatie uit het plan van aanpak en de definitiestudie was ik in staat om een pilotplan te maken, waarin de pilotontwikkelingsfase van het project nauwkeurig uiteen wordt gezet.
Het pilotplan richt zich in eerst instantie op de indeling van de systeemeisen in pilots. Op basis van de globale pilotstrategie, zoals gedefinieerd in de definitiestudie, richtte de structuur van de pilots zich hoofdzakelijk op drie aspecten, namelijk het herschrijven van de bestaande functionaliteit in C#, het ontwikkelen van extra functionaliteiten en het verbeteren van de gebruikersvriendelijkheid. De globale structuur in de drie pilots zorgde ervoor dat de systeemeisen vervolgens toegekend konden worden aan een specifieke pilot.
Op basis van de indeling van de systeemeisen middels het MoSCoW-principe, was er duidelijk wat de prioriteit was van de systeemeisen. Opvallend was dat de indeling van systeemeisen op basis van de prioriteit globaal overkwam met de globale pilot indeling. De systeemeisen met de zogenaamde 'Must have' prioriteit heb ik zodoende toegekend aan pilot 1, zodat deze als eerste worden ontwikkeld en de hoogste prioriteit hebben. De zogenaamde 'Should have' systeemeisen heb ik vervolgens geplaatst in pilot 2 en de zogenaamde 'Could have' systeemeisen in pilot 3. Van 'Would like to have' systeemeisen vond ik dat er niet echt sprake van was en die heb ik zodoende buiten beschouwing gelaten. Het resultaat betrof een indeling van de systeemeisen in drie pilots.

© 2005 Mathieu Brüning (20010019)

6.3 Pilotontwikkeling

6.3.1 Pilot 1: Herschrijven bestaande functionaliteit

Opstellen pilotontwikkelplan

De pilotontwikkelingsfase kon van start gaan toen het pilotplan was afgerond. Aan de start van de eerste pilot in de pilotontwikkelingsfase ben ik begonnen met het opstellen van een pilotontwikkelplan. Het maken van het pilotplan betrof in eerste instantie het definiëren van de delen van de pilot. Op basis van de systeemeisen en de globale pilotstructuur uit het pilotplan ben ik gaan inventariseren uit welke delen de eerste pilot moest bestaan. Hoofdzakelijk kwam het erop neer dat de eerste pilot diende te bestaan uit vier hoofdaspecten, namelijk het ontwikkelen van een basis interface, een login / autorisatie control en functionaliteiten voor het downloaden en uploaden van bestanden. Door de specifieke systeemeisen, behorend bij de pilotdelen, te inventariseren was ik in staat om een overzicht te formuleren van de activiteiten die uitgevoerd diende te worden tijdens de ontwikkeling van de eerste pilot. Op basis van deze gegevens ben ik dan ook aan de slag gegaan met het definiëren van bouweenheden voor de pilot. Bij het definiëren van de bouweenheden heb ik zoveel mogelijk rekening gehouden met het bundelen van gelijksoortige activiteiten, zodat het uitvoeren van dergelijke activiteiten op een zo efficiënt mogelijke wijze zou kunnen plaatsvinden. IAD schrijft voor dat na het definiëren van de pilotdelen en het definiëren van de bouweenheden de bouweenheden gesynchroniseerd dienen te worden. Naar mijn idee was er niet echt sprake van het synchroniseren van bouweenheden, aangezien het synchroniseren van bouweenheden zich voornamelijk richt op het vaststellen van afhankelijkheden en mogelijkheden met betrekking tot parallelle ontwikkeling. Het feit dat het project maar door één persoon wordt uitgevoerd was bij deze keuze essentieel, aangezien er dan veelal geen sprake is van parallelle ontwikkeling.

De volgende stap bij het opstellen van het pilotontwikkelplan was het voorbereiden op het beoordelen en testen van de ontwikkelde pilotdelen. Aangezien de eerste pilot zich hoofdzakelijk richtte op het herschrijven van de bestaande functionaliteit van het Marti-IT Update Centrum, was ik in staat om concrete criteria op te stellen, waaraan het nieuwe systeem diende te voldoen. Op basis van de bestaande applicatie kon ik namelijk een vergelijking maken met de functionaliteit van het nieuwe systeem en de functionaliteit van het oude systeem. Door de functionaliteiten inventarisatie uit de analyseactiviteit van het bestaande systeem te gebruiken was het eenvoudig om een overzicht te formuleren met de testcriteria. Op deze manier heb ik de voorbereiding met betrekking tot het beoordelen en testen van de ontwikkelde pilotdelen uitgevoerd.

Tot slot richt de laatste stap bij het opstellen van een pilotontwikkelplan zich op het opstellen van een invoeringsplan voor de ontwikkelde pilotdelen. Echter, dit was niet aan de orde bij de pilotstrategie die ik heb gehanteerd tijdens het project. De zogenaamde 'big-bang invoeren' strategie kenmerkt zich namelijk door het in één keer invoeren van een ontwikkeld systeem. Er is dus feitelijk geen sprake van het invoeren per pilot en zodoende heb ik deze stap bij het opstellen van het pilotontwikkelplan voor de eerste pilot overgeslagen. Het opstellen van het pilotontwikkelplan was hiermee afgerond.

Oriënteren op de C# programmeertaal

De eerste activiteit na het opstellen van het pilotontwikkelplan betrof het oriënteren op de C# programmeertaal. Het toepassen van de C# programmeertaal was voornamelijk een keuze vanuit Marti-IT, aangezien het toekomstige beleid met betrekking tot softwareontwikkeling zich richt op ontwikkelen middels de C# programmeertaal. Daarnaast was ik zelf zeer geïnteresseerd in het leren ontwikkelen in C#, aangezien ik kennis wilde maken met ontwikkelen middels het Microsoft .NET framework. Het oriënteren op de C# programmeertaal richtte zich hoofdzakelijk op het zoeken naar beschikbare informatie op het Internet en het vergaren van informatie uit boeken. In het verleden had ik al wel enige kleine projecten uitgevoerd in de C# programmeertaal, maar deze projecten waren meer te bestempelen als kleine testjes, waarmee ik de taal heb leren kennen. Naar mijn mening was een groot voordeel van de C# taal dat het vrij veel op de Java programmeertaal lijkt, waar ik wel ruime ervaring mee had. De syntax van C# lijkt sprekend op de syntax van Java, wat naar mijn idee een voordeel was bij het beter leren kennen van C#.

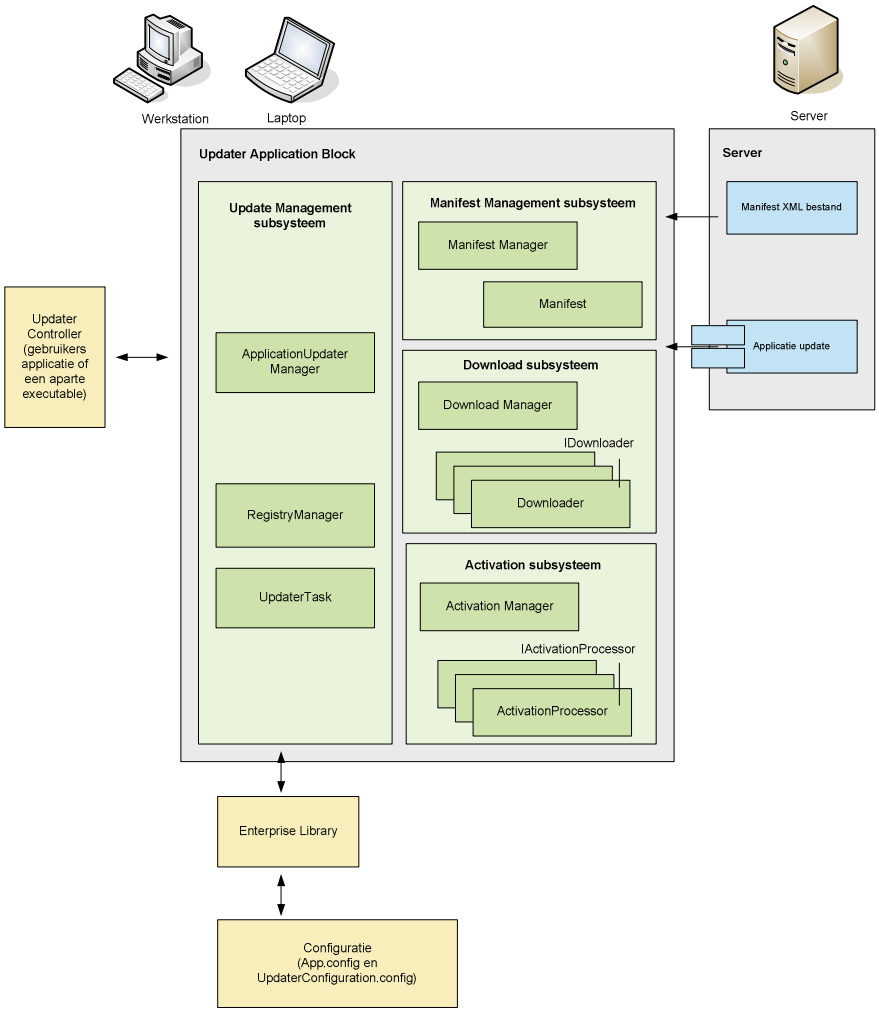
© 2005 Mathieu Brüning (20010019)

Zoeken naar herbruikbare componenten

Toen ik aan het zoeken was naar informatie op het Internet ben ik ook op zoek gegaan naar herbruikbare componenten voor mogelijke kant-en-klare oplossingen. Tijdens het surfen ben ik terecht gekomen op de website van Microsoft met betrekking tot ontwikkeling, de zogenaamde Microsoft Software Developer Network website. Na enige tijd gezocht te hebben op de site ben ik terecht gekomen op een pagina met betrekking tot herbruikbare componenten voor .NET, de zogenaamde *Patterns & Practices*. Deze herbruikbare componenten worden Application Blocks genoemd. Application Blocks zijn kant-en-klare oplossingen voor veel voorkomende problemen, welke toegepast kunnen worden in een .NET project. Deze componenten zijn gratis verkrijgbaar en zijn altijd open-source. Elke .NET ontwikkelaar wordt zo in staat gesteld om eenvoudige wijze een component te gebruiken dat volledig naar eigen inzicht kan worden aangepast. De beschikbaarheid van broncode is een essentieel onderdeel voor de ingebruikname van herbruikbare componenten, aangezien eventuele aanpassingen in een dergelijke component dan door de ontwikkelaar zelf kunnen worden gerealiseerd, zonder dat daarbij de hulp van een derde partij ingeschakeld hoeft te worden.

Mijn belangstelling voor de Application Blocks was tamelijk groot, aangezien ik vermoedde dat er waarschijnlijk ook wel een component voor het updaten van software beschikbaar zou zijn. Dit bleek inderdaad waar te zijn. Aan de hand van een gevonden Application Block, het zogeheten Updater Application Block, ben ik aan de slag gegaan met het analyseren van het betreffende component om te kijken of het wellicht mogelijk was om de Application Block te gebruiken in mijn project. Door de structuur van het component in kaart te brengen bleek al gauw dat het zeker de moeite waard was om het Updater Application Block te gebruiken. Het ontwerp van het component zat qua structuur naar mijn idee goed in elkaar, aangezien bijvoorbeeld het downloaden van files, het afhandelen van taken en het afhandelen van fouten en excepties al goed was verwerkt in de structuur.

Na de analyse heb ik besloten om de verdere ontwikkeling van het Update Centrum te baseren op het gevonden Application Block. Nadat de keuze voor het gebruiken van het Updater Application Block definitief was ben ik aan de slag gegaan met het in kaart brengen van de volledige structuur van het component.

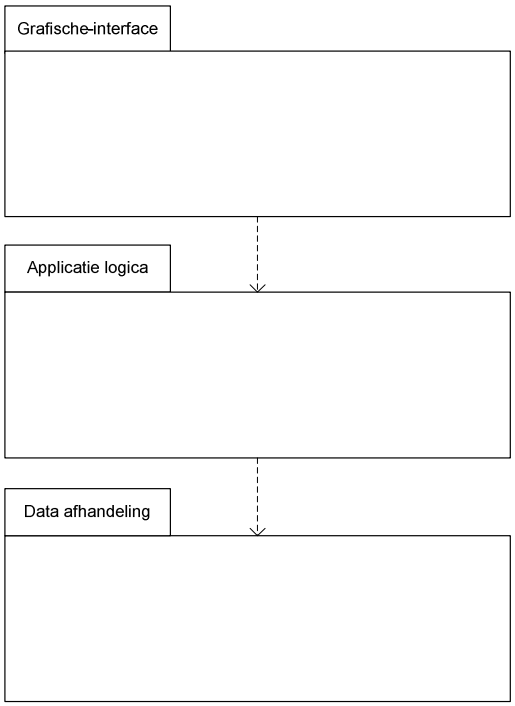


Figuur 7: *Structuur Updater Application Block*

Structureren ontwikkelomgeving

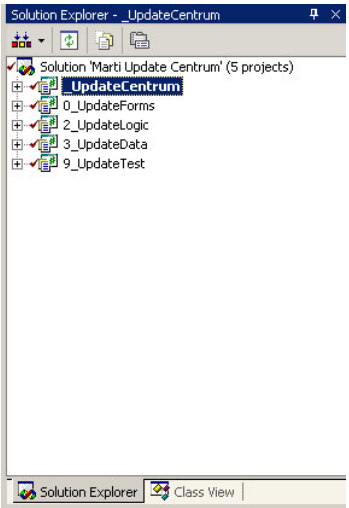
Nadat de structuur van het Updater Application Block in kaart was gebracht ben ik aan de slag gegaan met het maken van een indeling voor mijn .NET project. De gedachtegang hierachter was met name dat de verschillende soorten klassen in het project gerangschikt diende te worden, zodat de klassen ingedeeld zouden worden per categorie. Dit heeft als voordeel dat niet alle klassen in het project door elkaar staan en dat er tijdens de ontwikkeling duidelijk is waar welke klassen zijn gelokaliseerd in het project.

Op basis van een ander .NET project van Marti-IT ben ik aan de slag gegaan met het indelen van mijn project. De standaard naar de werkwijze binnen Marti-IT is een strikte scheiding tussen grafische gebruikers interfaces en controls, de logische klassen en de klassen voor data afhandeling.



Figuur 8: *Globale structuur van het systeem*

Naar aanleiding van deze informatie heb ik mijn project dan ook ingedeeld in een subproject voor grafische interfaces en controls, een subproject voor de logica en een subproject voor het afhandelen van de data (zie afbeelding).

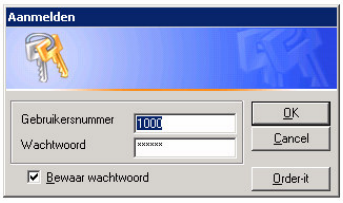


Figuur 9: *Indeling project*

Ontwikkelen superklasse voor toekomstige vensters
Toen de globale indeling van het project eenmaal tot stand gekomen was ben ik begonnen met de ontwikkeling van een superklasse voor alle toekomstige vensters. Deze superklasse diende in de toekomstige ontwikkeling van het project een standaard te worden voor alle vensters, met eigenschappen als een Marti-IT icoon, het eenvoudig toevoegen van een menu en het opvragen van de zogenaamde 'embedded resources' ofwel ingebouwde bronnen als bijvoorbeeld afbeeldingen.
Het voordeel van een dergelijke superklassen was voor mij met name dat ik dan niet telkens dezelfde functionaliteit hoefde te programmeren in de afzonderlijke toekomstige klassen.
Het aanmaken van de superklasse kon worden gerealiseerd door in eerste instantie een normaal *form* aan te maken in mijn project. Door vervolgens de specifieke methoden en kenmerken te programmeren was het zogenaamde formStandard ontwikkeld.

Ontwikkelen basis interface
Nadat de superklasse voor het maken van venster ontwikkeld was ben ik begonnen met het maken van een basis gebruikers interface en het aanpassen van de code van het Updater Application Block. Het maken van een eerste basis interface leek me een handige stap om de toekomstige ontwikkelingen te testen, door ze vanuit de interface aan te roepen. Het maken van de interface kon op eenvoudige wijze bewerkstelligd worden in Microsoft Visual Studio 2003, door een zogenaamd *form* aan te maken. Dit form heb ik vervolgens laten erven van de ontwikkelde superklasse, zodat het form voldeed aan de specifieke kenmerken van de superklasse. Aan deze basis interface heb ik vervolgens een menu toegekend, zodat functionaliteiten in het nieuwe systeem op eenvoudige wijze aangeroepen zouden kunnen worden.

Maken autorisatie / login control
Na de ontwikkeling van de basis interface richtte mijn werkzaamheden zich op een andere pilotdeel, namelijk het maken van een login / autorisatie functionaliteit. In eerste instantie ben ik op zoek gegaan naar een eventueel herbruikbaar component. Na enig zoekwerk bleek dat er geen geschikt standaard component voor het inloggen of autoriseren van een gebruiker geïntegreerd was in het .NET framework. Zodoende ben ik op zoek gegaan naar een herbruikbaar component op het World Wide Web. Ook hier bleek dat er feitelijk geen geschikt herbruikbaar component te vinden was en daarom ben ik zelf aan de slag gegaan met het ontwikkelen van een dergelijk component. Het autoriseren van gebruikers dient plaats te vinden aan de hand van een gebruikersnaam en wachtwoord, welke ingevuld dienen te worden in een venster dat getoond wordt bij het opstarten van het nieuwe systeem. Dat was hoofdzakelijk de gedachtegang achter het component.
In eerste instantie ben ik aan de slag gegaan met het ontwikkelen van een venster voor het inlog proces. Naar aanleiding van het advies van een collega ben ik terecht gekomen op www.windowsforms.com, een website waar het ontwikkelen van Windows applicaties middels het .NET framework centraal staat. Op deze site kunnen onder andere herbruikbare componenten en applicaties gedownload worden. Na het downloaden van een open-source applicatie op de website, was ik in staat om een afbeelding uit deze applicatie te gebruiken voor mijn login venster. Na het plaatsen van een aantal controls en een afbeelding was het grafische gedeelte van het login venster afgerond (zie onderstaande afbeelding). Nu diende echter de functionaliteit nog ontwikkeld te worden.



Figuur 10: *Autorisatie / login venster*

De functionaliteit van het login venster richt zich hoofdzakelijk op het autoriseren van een gebruiker middels een gebruikersnummer en een wachtwoord. Wanneer de gebruiker zijn nummer en wachtwoord invoert wordt er een query uitgevoerd, waarin het wachtwoord wordt opgevraagd van de gebruiker aan de hand van het gebruikersnummer. Indien het ingevoerde wachtwoord overeenkomt met het wachtwoord van het query resultaat, dan is de gebruiker geautoriseerd.

Een belangrijke eis aan de autorisatie functionaliteit was dat het inloggen op een beveiligde manier diende plaats te vinden.
Bij het beveiligen van de autorisatie functionaliteit heb ik een drietal mogelijke oplossingen bekeken, namelijk:

- ✓ het gebruiken van de Windows authenticatie voor het autoriseren van een gebruiker
- ✓ het opslaan van beveiligingsinstellingen in het register
- ✓ het toepassen van encryptie op de gebruikersgegevens

Al gauw bleek dat het gebruiken van de Windows authenticatie van gebruikers vrijwel onmogelijk was. Met name doordat ik geen toegang heb tot de eindsystemen van de gebruikers, is het lastig om een dergelijke manier van beveiligen te implementeren. Alle Windows authenticaties van de gebruikers zijn namelijk verschillend en kunnen mogelijk ook vaak veranderen, in verband met nieuwe medewerkers, nieuwe pc's, etc. De optie met betrekking tot het gebruiken van Windows authenticatie viel zodoende weg.

De tweede optie, het opslaan van beveiligingsinstellingen in het register bleek ook niet erg toepasbaar te zijn. Het register is namelijk op eenvoudige wijze te benaderen en beveiligingsinstellingen zouden op die manier dus op eenvoudige wijze te achterhalen zijn. Deze optie viel ook weg.

Het versleutelen van een wachtwoord van de gebruiker leek zodoende de beste optie. Het idee berust op de volgende gedachte: de wachtwoorden van de gebruikers staan versleuteld in de database. Indien een gebruiker zich probeert te autoriseren wordt het wachtwoord uit de database gehaald en het wordt het wachtwoord ontsleuteld. De wachtwoorden kunnen dan vervolgens vergeleken worden, zodat er bepaald kan worden of de gebruiker geautoriseerd is of niet.

Het toepassen van de encryptie functionaliteit leek me de beste optie en zodoende ben ik aan de slag gegaan met het implementeren van de betreffende functionaliteit. Het gebruikte encryptie algoritme is een standaard component van het Microsoft .NET framework. Middels een zelf gedefinieerde tekst kan er vervolgens een sleutel worden bepaald, waarop de encryptie in de toekomst gebaseerd zou zijn. Het resultaat betrof een functionaliteit voor het versleutelen van tekst, welke gebruik maakt van een sleutel op basis van een zelf gedefinieerde tekst. De wachtwoorden kunnen hiermee versleuteld en ontsleuteld worden, zodat deze vergeleken kunnen worden bij het autorisatieproces. Het beveiligingsaspect is naar mijn idee zo goed in het component verwerkt, aangezien alleen degene die over de tekst beschikken waarmee de sleutel is gemaakt de wachtwoorden kunnen ontsleutelen. Het encryptie algoritme is gebaseerd op de Advanced Encryption Standard (AES) / Rijndael, waarbij een 128 bits sleutel gebruikt wordt. Het toepassen van een dergelijke encryptie is naar mijn mening ruim voldoende voor het versleutelen van gegevens in de database, aangezien de kans naar mijn idee erg klein is dat er pogingen gedaan worden voor het ontsleutelen van de gegevens in de database van het Marti-IT Update Centrum. Mocht dit onverhoopt toch het geval zijn, dan moet er eerst een 128 bits sleutel gekraakt worden.

Selecteren geschikte database

Om de ontwikkelde autorisatie functionaliteit op adequate wijze te testen diende de toekomstige database gekoppeld te worden aan het login venster. De volgende stap was dus het inrichten van een database, zodat er testgegevens ingevoerd konden worden en tests uitgevoerd konden worden op het ontwikkelde autorisatie component. Een belangrijke stap bij het inrichten van de database was in eerste instantie de keuze welke database er gebruikt zou gaan worden in het nieuwe systeem.

Bij het selecteren van een geschikte database ben ik in eerste instantie gaan onderzoeken aan welke eisen de database zou moeten voldoen. Een fundamenteel aspect van het bestaande Update Centrum is dat het goed mogelijk is dat er meerdere gebruikers tegelijkertijd toegang willen hebben tot het systeem. Aangezien het ophalen van informatie omtrent de updates in het nieuwe systeem verloopt via een database, is het belangrijk dat het mogelijk is dat meerdere gebruiker tegelijkertijd toegang kunnen krijgen tot de database. Een belangrijke eis aan de database diende dus te zijn er dat ondersteuning was op het gebied van multi-user toegang.
Een tweede kenmerk van de database richtte zich op de performance. Op basis van de analyse van het oude systeem bleek dat de performance niet heel erg belangrijk was voor de database, aangezien er bij het ophalen van updates niet heel veel transacties plaatsvinden. De voornaamste transacties richten zich namelijk op het ophalen van informatie omtrent updates en het wegschrijven van een record voor iedere download en upload die heeft plaatsgevonden. Een gebruiker maakt bij het gebruik van het Update Centrum maar gebruik van een klein aantal transacties en zodoende was ik van mening dat de performance van de database geen belangrijke systeemeis was.

© 2005 Mathieu Brüning (20010019)

Bij het selecteren van een geschikte database ben ik begonnen met het inventariseren van welke databases er zoals binnen Marti-IT gebruikt worden voor de software. Hieruit bleek dat feitelijk alle software gebruik maakt van Advantage Database Server (ADS) databases. Daarnaast werd er nog voor een klein deel gebruik gemaakt van MySQL databases, welke hoofdzakelijk werden gebruikt voor oplossingen met betrekking tot het World Wide Web. Op basis van deze gegevens ben ik gaan analyseren of deze databases wellicht bruikbaar waren in mijn project. Met behulp van ervaringen van collega's en informatie op het Internet bleek al gauw dat de ADS databases platte tabel bestanden zijn, waarbij elke file één tabel representeert. Daarnaast bleek dat het gebruik van de Structured Query Language op een ADS database alleen mogelijk is middels speciale additionele tools. Dit is voornamelijk te wijten aan het feit dat de ADS databases gebaseerd zijn op Win32, de Microsoft Windows API voor Microsoft Windows 9X, NT en 2000.
De conclusie wat betreft het gebruik van een ADS database kwam er hoofdzakelijk op neer dat het een ietwat verouderde vorm van databasegebruik is, aangezien alle moderne database functionaliteiten als locking, recovery, relaties, e.d. niet mogelijk waren met een ADS database. Ondanks het feit dat er zeer veel ervaring was met de ADS databases binnen Marti-IT, heb ik toch besloten om gebruik te maken van een andere database.

Zodoende ben ik gaan kijken of een MySQL database eventueel bruikbaar was in mijn project. Al snel bleek dat het gebruik van een MySQL database geen kosten met zich meebracht. De MySQL database server is gratis te downloaden en is volledig open-source. Een ander groot voordeel, ten opzichte van een ADS database, is dat met de MySQL database wel alle standaard DBMS functionaliteiten als locking en recovery mogelijk waren. Het gebruik van een MySQL database leek dan ook vele voordelen te hebben. Om niet meteen overhaaste beslissingen te nemen ben ik verder gaan zoeken op het Internet naar eventuele andere toepasbare databases.

Bij het zoeken naar andere databases bleek ik al snel te stuitten op Microsoft SQL Server. Op basis van informatie op het World Wide Web bleek dat deze database server van Microsoft zich kenmerkt door met name top prestaties en gemakkelijke installatie, gebruik en beheer. Op basis van deze informatie ben ik vervolgens gaan kijken om het eventueel mogelijk zou kunnen zijn om deze database te gebruiken. Naar enigszins verdere gezocht te hebben in informatie bleek er echter een groot nadeel te zitten aan de database server van Microsoft, namelijk de kosten. De kosten van Microsoft SQL Server variëren van $ 4.999 tot $ 19.999 per processor. Hiermee viel het gebruik van Microsoft SQL Server feitelijk direct al af. De kosten van een dergelijke database server waren simpelweg te hoog.

Onderstaand wordt een matrix weergegeven waarin de positieve als negatieve kenmerken van de drie databasesoorten worden vergeleken.

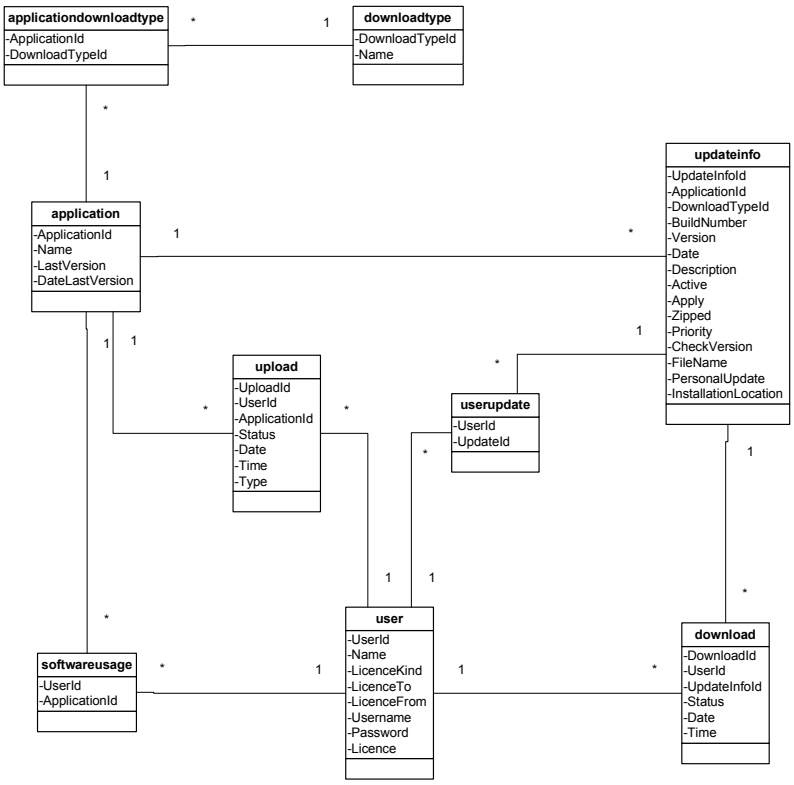
Soort database	Positieve kenmerken	Negatieve kenmerken
ADS	✓ Veel ervaring mee binnen de organisatie ✓ Geen kosten	✓ Zonder additionele tools is SQL niet mogelijk, zodoende lastig te implementeren in C# ✓ Weinig tot geen standaard databasemanagement-system functionaliteiten
MySQL	✓ Geen kosten ✓ Enige ervaring mee binnen de organisatie ✓ Wel standaard databasemanagement-systeem functionaliteiten ✓ Multi-user	✓ De performance van een MySQL database is niet extreem goed bij zeer grote hoeveelheden transacties
SQL Server	✓ Zeer goede performance ✓ Multi-user	✓ Hoge kosten ✓ Desktop versie mag niet gedistribueerd worden onder klanten

Matrix 1: *Vergelijking mogelijk bruikbare databases*

Op basis van de vergaarde informatie omtrent de databases bleek dat het gebruik van een MySQL database de beste keuze was. Een zeer belangrijke eis aan de database richtte zich op de mogelijkheid tot de gelijktijdige toegang van gebruikers. Op basis van informatie op de MySQL website, http://www.mysql.com, bleek dat de MySQL database ondersteuning biedt voor maximaal 100 gelijktijdige verbindingen. Vanuit dat oogpunt was de MySQL database dus zeker wel geschikt, aangezien er hoogstwaarschijnlijk nooit sprake is van 100 gelijktijdige gebruikers op het Update Centrum.
Aangezien het nieuwe Marti-IT Update Centrum niet excessief gebruik maakt van een database, doordat het aantal transacties niet erg hoog ligt, was het gebruik van een MySQL database een goede keuze, aangezien de prestaties van een MySQL database lager liggen dan de prestaties van een SQL Server. Daarnaast was de reeds bestaande ervaring binnen Marti-IT een eveneens meespelende factor voor de keuze van de database.
Op het gebied van kosten bleek dat het toepassen Microsoft SQL Server ten opzichte van het toepassen van MySQL vele malen duurder was. Doordat de MySQL database een goed alternatief is voor de Microsoft SQL Server, bleek het niet nodig te zijn om te investeren in Microsoft SQL Server databases en is er zodoende gekozen voor het toepassen van een MySQL database.

Ontwerpen structuur database

Toen ik eenmaal heb besloten dat er gebruik gemaakt zou gaan worden van een MySQL database heb ik mijn werkzaamheden gericht op het ontwikkelen van een klassendiagram voor de indeling van de database. Op basis van het globale gegevensmodel uit de definitiestudie (zie rapport specificaties Marti-IT Update / Build Centrum § 6.4) ben ik verder gegaan met het in kaart brengen van de indeling van de database. De indeling is met name gebaseerd op een overzicht van de te verwerken gegevens in het nieuwe systeem vanuit de systeemeisen. Een belangrijke gedachtegang achter het nieuwe Update Centrum was dat het opslaan van informatie omtrent updates in ini bestanden tot het verleden zou behoren. De informatie omtrent de updates zou dan worden opgeslagen in een database, zodat de losse ini bestanden niet meer gebruikt zou hoeven te worden. De gedachtegang achter het implementeren van een database voor het opslaan van de informatie omtrent updates is gebaseerd op een centrale opslag van alle informatie. In het kader van onderhoudbaarheid worden namelijk de losse ini bestanden vervangen door één centrale database, zodat het onderhouden van de informatie op één centrale plek kan verlopen.
De informatie omtrent de instellingen, zoals gebruikt in de ini bestanden, heb ik ook meegenomen bij het maken van het model voor de databasestructuur. Het maken van het model heeft plaatsgevonden in Microsoft Visio.



Figuur 11: *Database structuur*

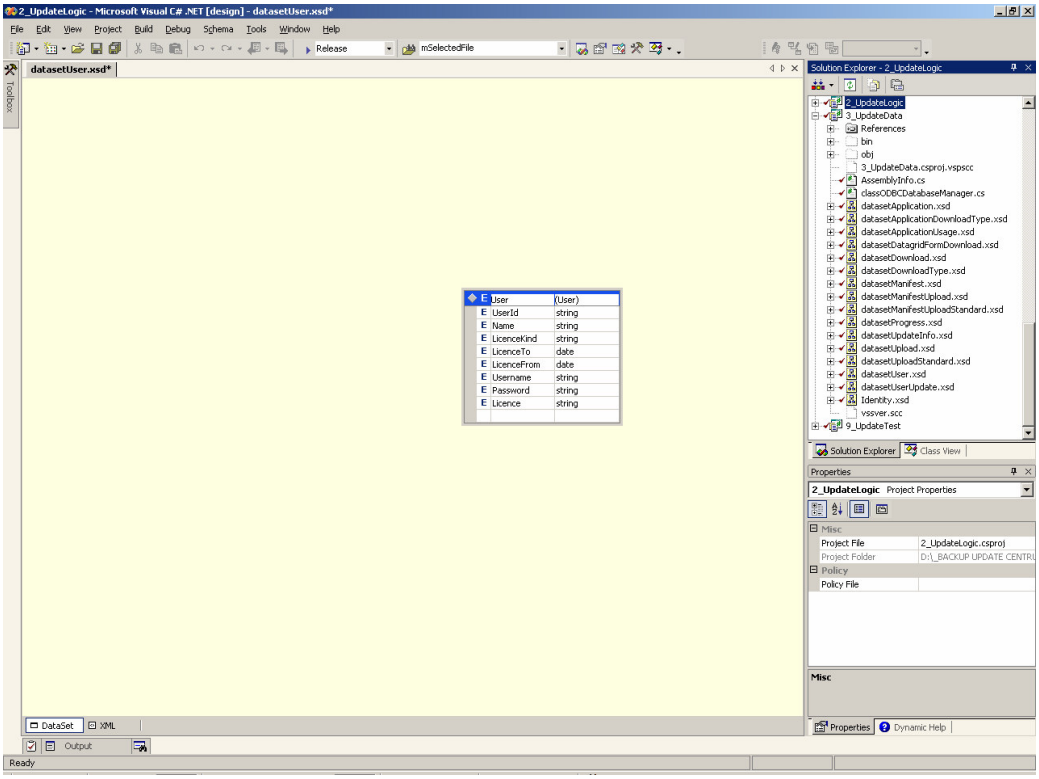
Inrichten database en invoeren testgegevens
Nadat het klassendiagram was opgesteld met de structuur van de database heb ik begonnen met het creëren van de tabellen in de database. Middels een tool, het zogenaamde MySQL Control Center, was ik in staat om middels een grafische interface de tabellen te creëren. Nadat de tabellen gecreëerd waren ben ik wat testgegevens gaan invoeren om zo vervolgens terug te komen op het testen van mijn autorisatie component. Het invoeren van de testgegevens vond ook plaats middels de grafische interface van het MySQL Control Center.

Met de database gevormd en de testgegevens ingevoerd ben ik het autorisatie component gaan testen. Door middel van het invoeren van een gebruikersnummer en wachtwoord worden de gegevens gevalideerd in de database en wordt de toegang tot de applicatie verleent. Dit vond nog plaats door middel van een lokale afhandeling van de SQL queries in de klasse van het login venster. Dit werkte allemaal naar toebehoren, maar dit diende aangepast te worden, zodat de afhandeling van de SQL statements op een centrale plaats diende plaats te vinden.

Ontwikkelen klasse voor database afhandelingen
Voor het afhandelen van een data afhandelingen had ik in een eerder stadium al een plek gereserveerd in mijn project, namelijk de zogenaamde data laag in mijn project. Door namelijk een globale indeling te maken in mijn project konden de verschillende soorten klassen naar categorie worden ingedeeld. Mijn doel was om een data laag te creëren, waar alle afhandelingen met de database door worden gestuurd. De kern van deze laag diende een klasse te zijn die daadwerkelijke SQL statements naar de database zou sturen. Met dit gegeven ben ik aan de slag gegaan met het maken van een klasse voor het afhandelen van de database commando's de zogenaamde classODBCDatabaseManager klasse. De classODBCDatabaseManager klasse is een klasse die een connectie maakt van een ODBC-gegevensbron, in dit geval de MySQL database, om gegevens te kunnen verzenden en ontvangen.

Ontwikkelen datasets
Het maken van de data laag voor het systeem betrof echter niet alleen het maken van een klasse voor het afhandelen van de SQL statements. Het verwerken van data afhandeling in C# vindt plaats via zogenaamde *datasets*. Een dataset is een data cache in het geheugen. Wanneer een database wordt geraadpleegd en data wordt bijvoorbeeld opgehaald vanuit een database wordt er een dataset gevuld met de betreffende gegevens. Deze dataset kan vervolgens gebruikt worden om de data de wijzigen en vervolgens weer terug te sturen naar de database.
Indien bijvoorbeeld een standaard SELECT query wordt uitgevoerd en een dataset opgehaald wordt vanuit de database wordt er een zogenaamde *Untyped dataset* geretourneerd. Een untyped dataset wil zeggen dat er een dataset is opgehaald, waarvan de gegevenstypen van de velden en het aantal velden niet is vastgelegd. Middels een untyped dataset wordt er dus niet gecontroleerd of bepaalde waarden wel overeenkomen met het corresponderende gegevenstype en kan deze dataset met alle mogelijke waarden gevuld worden.
Een *typed dataset* daarentegen is een dataset waarvan juist wel bekend is welke velden er zijn en welke gegevenstypen deze velden hebben. Het voordeel bij het gebruik van typed datasets is dan ook dat er bij het ophalen of versturen van data direct wordt gecontroleerd of de velden en de gegevenstypen wel correct zijn. De volgende stap bij het vormen van de data laag voor het Marti-IT Update Centrum was het maken van typed datasets voor alle tabellen in de database. Het creëren van typed datasets kan op een grafische manier worden gerealiseerd in Microsoft Visual Studio. Hierbij dienen de veldnamen en de gegevenstypen van de velden te worden gespecificeerd. Nadat alle typed datasets voor de tabellen in de database waren gecreëerd was de data laag voor het systeem feitelijk gevormd.

© 2005 Mathieu Brüning (20010019)



Figuur 12: *Middels de Microsoft Visual Studio IDE kunnen de datasets op grafische wijze worden ontwikkeld*

Informatie zoeken omtrent FTP en HTTP

Nadat de data afhandelingslaag voor het systeem was gevormd en het autorisatie component naar toebehoren functioneerde ben ik op zoek gegaan naar informatie omtrent functionaliteiten voor het downloaden van bestanden. Het downloaden van bestanden vond in het huidige Update Centrum plaats door een FTP-verbinding. Echter, wellicht was er een betere manier voor het binnenhalen van de bestanden en zodoende ben ik op het World Wide Web op zoek gegaan naar informatie. Hierbij heb ik met name gezocht op informatie omtrent het File Transfer Protocol (FTP) en informatie omtrent HyperText Transfer Protocol (HTTP). De gevonden informatie betrof het volgende:

Voordelen FTP ten opzichte van HTTP:

- ✓ minder belasting voor het netwerk met kleine bestanden
- ✓ betere ondersteuning met grote bestand
- ✓ het opvragen van map & bestandsinformatie
- ✓ recursief downloaden/uploaden van mapstructuren

Voordeel HTTP ten opzichte van FTP:

- ✓ Geen aparte client nodig voor HTTP, HTTP wordt standaard ondersteund door de C# taal

Op basis van de gevonden informatie heb ik besloten om gebruik te maken van een FTP-connectie voor het downloaden en uploaden van bestanden in plaats van een HTTP-connectie. De reden hiervoor is dat het gebruik van FTP meer voordelen met zich meebrengt ten opzichte van het gebruik van HTTP.
Met name het opvragen van map en bestandsinformatie is een belangrijk aspect dat veelvuldig gebruikt zal worden bij het uploaden en downloaden van bestanden via het Update Centrum. Dit gaf voor mij de doorslag om een FTP verbinding te gebruiken.

Ontwerpen downloadfunctionaliteit

Het ontwerpen van de functionaliteit voor het downloaden van bestanden middels het Update Centrum was een volgende stap in de eerste pilot. Aangezien ik had besloten om gebruik te maken van het Updater Application Block, ben ik in eerste instantie gaan bekijken hoe de download functionaliteit in het component verwerkt zat. Hieruit bleek dat de download functionaliteit voor het overgrote deel herbruikbaar was voor de download functionaliteit voor het Update Centrum. Met die gedachtegang ben ik aan de slag gegaan met het importeren van de bruikbare klassen naar mijn project. Elk klasse in de C# taal bestaat uit een apart bestand met als naam de klassenaam en als extensie .cs. Het importeren van de klassen vanuit het Updater Application Block naar mijn project richtte zich simpelweg op het overbrengen van de bestanden naar de juiste map en vervolgens importeren in mijn C# project. Nadat de klassen geïmporteerd waren ben ik aan de slag gegaan met het compileren van mijn project. Na het aanpassen van een aantal kleine zaken, zoals verwijzingen die niet goed stonden, was in ik staat om het project te compileren.

De hoofdstructuur van updaten middels het Updater Application Block komt hoofdzakelijk neer op het volgende: in eerste instantie wordt er een methode aangeroepen die controleert of er updates beschikbaar zijn. Indien er updates beschikbaar zijn wordt er een lijst opgebouwd met zogenaamde 'manifests'. Vervolgens wordt er gecontroleerd of alle benodigde informatie beschikbaar is en wordt er een taak van gemaakt. De bestanden van deze taak worden gedownload en indien nodig geïnstalleerd. Onderstaand wordt de originele structuur en werking van het Updater Application Block weergegeven.

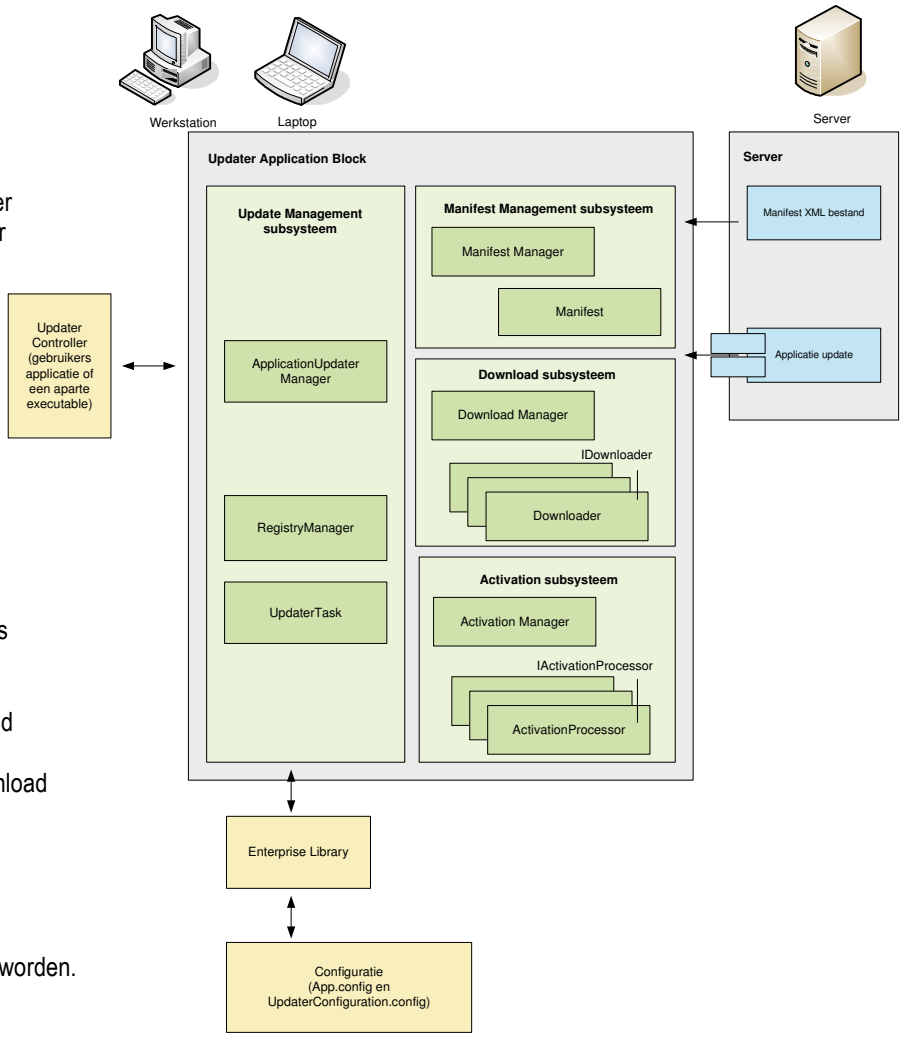
Stap 1: De ApplicationUpdater Manager roept de functionaliteit aan voor het inventariseren van de beschikbare updates

Stap 2: De manifest manager controleert welke updates er beschikbaar zijn aan de hand van xml-bestanden op de server. De informatie omtrent de updates wordt dan gehaald uit de xml-bestanden.

Stap 3: De RegistryManager maakt van de manifests UpdaterTasks

Stap 4: De UpdaterTasks worden doorgestuurd naar de Download Manager, welke ervoor zorgt dat de updates worden gedownload

Stap 5: Indien de updates zijn gedownload worden de UpdaterTasks naar de Activation Manager gestuurd, zodat ze geïnstalleerd kunnen worden.



Figuur 13: *Structuur Updater Application Block*

Naar aanleiding van de bovenstaande globale structuur was ik van mening dat dit overeenkomt met de manier waarop gebruikers updates kunnen overhalen via het nieuwe Update Centrum. In eerste instantie controleert een gebruiker namelijk welke updates er beschikbaar zijn voor hem via de database, vervolgens worden er taken aangemaakt voor de geselecteerde updates die uiteindelijk worden gedownload en worden geïnstalleerd. Het toepassen van het Updater Application Block in mijn project, droeg zorg voor een deel van het ontwerp en de implementatie van een deel van de download functionaliteit. Echter, veel code van het herbruikbare component diende nog aangepast te worden, aangezien er ook veel functionaliteiten in zaten die feitelijk overbodig of nog niet helemaal bruikbaar waren.
Na het toepassen van een groot aantal klassen uit het Updater Application Block ben ik aan de slag gegaan met het zoeken naar informatie of er wellicht nog meer herbruikbare stukken code beschikbaar was in de Microsoft Application Blocks. Daarnaast was ik op zoek naar informatie omtrent de werking en het gebruik van de componenten, aangezien het toepassen van het Updater Application Block niet geheel eenvoudig was. De Application Blocks van Microsoft zijn namelijk zeer uitgebreide componenten, met een complexe architectuur. Met name doordat er in het Updater Application Block weer gebruik gemaakt wordt van een ander Application Block, het zogenaamde Cryptography Application Block, was het op sommige momenten nog wel eens moeilijk om door de bomen het bos te zien. Door echter een bijgeleverde voorbeeld applicatie voor het Updater Application Block te debuggen, was ik in staat om de structuur en werking van het component te doorgronden. Door namelijk stap voor stap te bekijken welke methoden en worden uitgevoerd en welke klassen er worden gebruikt tijdens de uitvoering van de voorbeeldapplicatie werd het me allemaal steeds duidelijker.

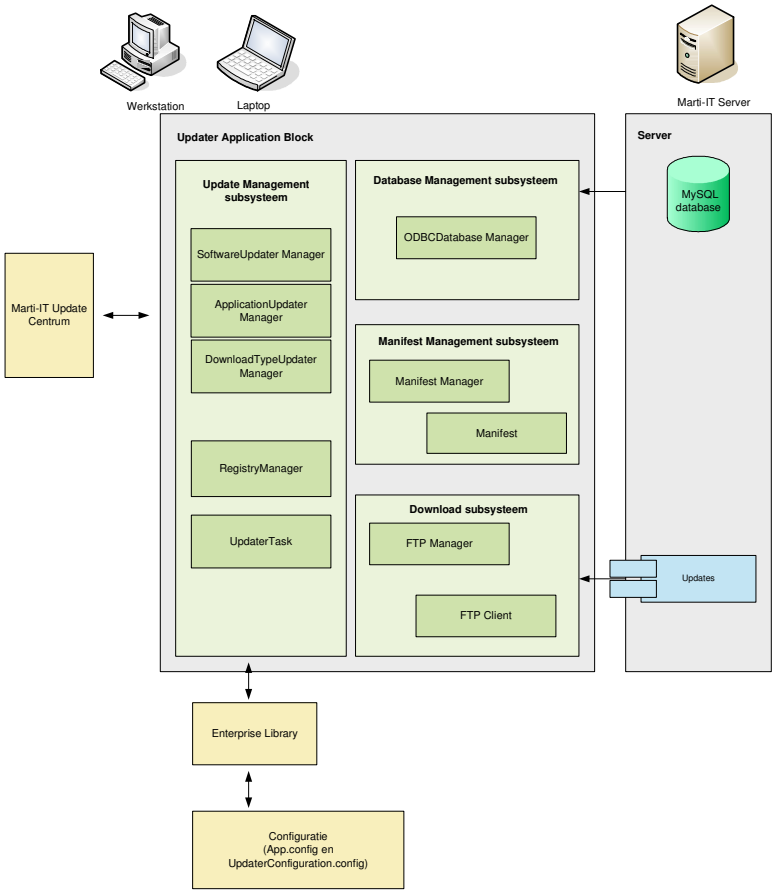
Toen eenmaal de werking en toepassing van het component inzichtelijk was ben ik aan de slag gegaan met het aanpassen van het ontwerp van het Updater Application Block, zodat een ontwerp voor de downloadfunctionaliteit van het nieuwe Marti-IT Update Centrum het resultaat zou zijn. Met behulp van Microsoft Visio ben ik aan de slag gegaan met het maken van een schematisch ontwerp voor de download functionaliteit. Aan de hand van het ontwerp van het herbruikbare component ben ik gaan kijken wel zaken er eventueel zouden vervallen en welke zaken er eventueel aangepast diende te worden. Al gauw bleek dat de bestaande functionaliteit voor het opslaan van informatie omtrent updates in xml-bestanden uit het component verwijderd diende te worden. In het ontwerp van het Updater Application Block wordt namelijk de informatie omtrent beschikbare updates opgeslagen in xml-bestanden. Bij het nieuwe Marti-IT Update Centrum leek me niet zo handig, aangezien er niet alleen informatie opgeslagen diende te worden omtrent updates, maar ook omtrent gebruikers, software, historische gegevens, enz. Wanneer deze informatie wordt opgeslagen in losse xml-bestanden, groeit het aantal bestanden erg snel en is het beveiligen van de bestanden hoofdzakelijk moeilijker dan in een database. Het opslaan van informatie omtrent de updates middels xml-bestanden heb ik vervolgens verwijderd uit het ontwerp. In plaats daarvan heb ik de opslag middels de geselecteerde MySQL database gekozen. Op deze manier kunnen de diverse gegevens wel op een eenvoudige manier worden centraal opgeslagen in grote hoeveelheden.

Naast de verandering met betrekking tot invoering van de MySQL database richtte de tweede verandering zich op het uitbreiden van de Update Management subsysteem. In de oorspronkelijke structuur van het component was het namelijk alleen mogelijk om updates te downloaden voor een enkele applicatie. Een belangrijk aspect van het nieuwe Marti-IT Update Centrum diende echter te zijn dat de individuele Update Centrum samengevoegd diende te worden tot één enkel systeem, zodat de gebruikers van de Marti-IT software alle benodigde updates voor verschillende applicaties in één keer kunnen downloaden. Om deze verandering te realiseren heb ik een klasse toegevoegd die de klasse ApplicationUpdaterManager overkoepeld. Deze klasse, genaamd SoftwareUpdaterManager kan meerdere instanties bevatten van de klasse ApplicationUpdaterManager, zodat het mogelijk zou worden om updates te downloaden voor meerdere applicaties tegelijkertijd.

De derde verandering in het oorspronkelijke ontwerp van het Updater Application Block richtte zich op het aanpassen van het download subsysteem. Op basis van de code bleek dat er gebruik gemaakt werd van een downloadfunctionaliteit die de gegevens binnenhaalde middels het HTTP protocol. In een eerder stadium van pilot 1 had ik echter besloten dat het downloaden en uploaden van gegevens middels het FTP protocol diende te gebeuren en zodoende heb ik het download subsysteem ook verwijderd uit het ontwerp. In plaats daarvan is een zogenaamde FTP Manager gekomen, een download/upload functionaliteit voor het afhandelen van de taken middels het File Transfer Protocol.

Naast het verwijderen/aanpassen van de functionaliteit voor het verwerken van informatie omtrent updates en naast het verwijderen van het download subsysteem heb ik de structuur van het component intact gelaten. De overige functionaliteiten waren wel allemaal goed toepasbaar in de structuur van het nieuwe systeem, aangezien de afhandeling van het downloaden van updates op globaal dezelfde wijze verloopt als in het huidige systeem.

© 2005 Mathieu Brüning (20010019)



Figuur 14: *Structuur download functionaliteit*

De veranderingen in de structuur richten zich met name op de uitbreiding van de ApplicationUpdater Manager, zodat het mogelijk is om meerdere applicaties te beheren.

Daarnaast is de vervanging van de xml-bestanden door middel van een database en een databasemanager essentieel geweest voor het ophalen en opslaan van de informatie omtrent de updates.

Het verwijderen van de Download Manager heeft vervolgens plaatsgevonden om ruimte te kunnen bieden aan de FTP Manager.

Tot slot is de Activation Manager verwijderd, aangezien de installaties in de toekomst zouden verlopen via de SoftwareUpdaterManager.

Programmeren download functionaliteit

Nadat de structuur van de download functionaliteit in kaart was gebracht ben ik aan de slag gegaan met het programmeren van de download functionaliteit. Het programmeren van de download functionaliteit richtte zich in eerste instantie op het aanpassen van het Updater Application Block naar de nieuwe structuur voor de download functionaliteit. Hoofdzakelijk kwam dit neer op het verwijderen en aanpassen van een aantal klassen en methoden. Hierbij ben ik als eerst bezig geweest met het aanpassen van de Manifest manager, zodat de input omtrent de updates niet meer vanuit xml-bestanden werd gehaald, maar vanuit de database. Door de database te raadplegen middels een query kan op eenvoudige wijze bepaald worden welke updates er beschikbaar zijn op de Marti-IT server.

De manifest objecten, die worden gevuld met gegevens wanneer er informatie van de server gevraagd omtrent de updates, bevatte naar mijn idee te weinig informatie. Er diende dus een aantal variabelen toegevoegd te worden in de Manifest klasse, zodat er wel voldoende informatie beschikbaar zou zijn voor het downloaden en installeren van een update. Het aanpassen van de Manifest klasse richtte zich zodoende enkel op het toevoegen van een aantal variabelen.

© 2005 Mathieu Brüning (20010019)

Nadat de Manifest Manager en de Manifest klasse waren aangepast naar het nieuwe ontwerp ben ik aan de slag gegaan met het verwijderen van de download subsysteem. Het verwijderen van het download subsysteem kon op eenvoudige wijze gerealiseerd worden door een aantal klassen te verwijderen. Het verwijderen van de klassen had echter wel als resultaat dat het compileren niet meer mogelijk was. Zodoende ben ik de zogenaamde FTP Manager klasse gaan ontwikkelen, welke de rol diende te vervullen van de Download Manager. Het vervullen van de rol van de Download Manager betekent feitelijk dat de taken (de zogenaamde UpdaterTask objecten) afgehandeld moet worden in de vorm van het downloaden van updates.
Het afhandelen van de taken was over het algemeen geen probleem, aangezien een UpdaterTask simpelweg een Manifest object bevat, waarin de informatie van de update zit. Er ontbrak echter nog een functionaliteit voor het downloaden van de bestand en zodoende heb ik mijn werkzaamheden gericht op het maken van een FTP client. De FTP client klasse diende namelijk in de toekomst alle gegevens te verzenden en ontvangen van en naar de Marti-IT FTP server. Op het Internet ben ik op zoek gegaan naar C# broncode voor een FTP client. Middels deze broncode zou ik vervolgens in staat zijn om een eigen klasse te definiëren die vervolgens diende te functioneren als FTP client. Op een gegeven ogenblik ben ik terecht gekomen op de website van Enterprise Distributed Technologies (http:// http://www.enterprisedt.com). Op deze website was het mogelijk om broncode te downloaden voor een FTP client in C#.

Na het downloaden van de broncode bleek dat het de gedownloade FTP client erg uitgebreid was, daar ik deze vergeleek met andere FTP clients die ik tegen ben gekomen op het Internet. Middels deze broncode was ik vervolgens in staat om een FTP client voor het Marti-IT Update Centrum te maken. Door de broncode simpelweg te importeren in nieuwe klassen in mijn project had ik een FTP client gemaakt.
Het testen van de werking van de nieuwe FTP client vond hoofdzakelijk plaats door contact te leggen met de Marti-IT FTP server en te controleren of het downloaden en uploaden van bestanden goed verliep. Na een aantal simpele download en upload tests ben ik verder gegaan met de resterende ontwikkeling van de download functionaliteit.

Ontwerpen en ontwikkelen identiteit object
Een belangrijke nieuwe feature aan het nieuwe Update Centrum diende te zijn dat gebruikers geen updates meer kunnen download die niet benodigd zijn. Indien een gebruiker bijvoorbeeld de nieuwste update al reeds in bezit heeft, en vervolgens naar het Marti-IT Update Centrum gaat om te controleren op nieuwe updates, zal deze nieuwste update niet meer worden getoond. In het oude Marti-IT Update Centrum was dit echter wel het geval en konden gebruikers dus keer op keer dezelfde nieuwste updates downloaden. Er diende dus een functionaliteit in het systeem te komen die controleert over welke updates te gebruiker al beschikt, zodat de updates in het bezit van de gebruiker niet meer worden getoond. Met die gedachte ben ik aan de slag gegaan met het ontwerpen van een functionaliteit voor het lokaal opslaan van de informatie omtrent reeds gedownloade updates. Het resultaat betrof een zogenaamd Identiteit object, een object waarin informatie omtrent de software en het downloaden van updates van de gebruiker wordt opgeslagen. Het idee is als volgt: wanneer een gebruiker zich aanmeldt bij het Update Centrum worden de gegevens omtrent de software van gebruiker vanuit de database opgehaald. Het Identiteit object draagt dan vervolgens zorg voor de opslag van het softwaregebruik van de gebruiker. Op die manier kan op elk moment in de applicatie opgevraagd worden welke software de gebruiker bezit, zodat updates voor alleen deze specifieke software worden aangeboden. Naast het opslaan van het softwaregebruik van een gebruiker worden eveneens specifieke gegevens vastgelegd omtrent de gedownload updates. Op deze manier kan er in een toekomstig bezoek aan het Update Centrum op eenvoudige manier gecontroleerd worden welke updates er reeds in het bezit zijn van de gebruiker. Het vastleggen van de gegevens vind plaats middels een xml-bestand. Elke keer dat het Update Centrum opstart zal dan in eerst instantie het xml-bestand worden geraadpleegd en zonodig worden bijgewerkt wat betreft het software gebruik. Op deze manier is er ten allen tijdens informatie beschikbaar omtrent het gebruik van de software en reeds gedownloade updates.

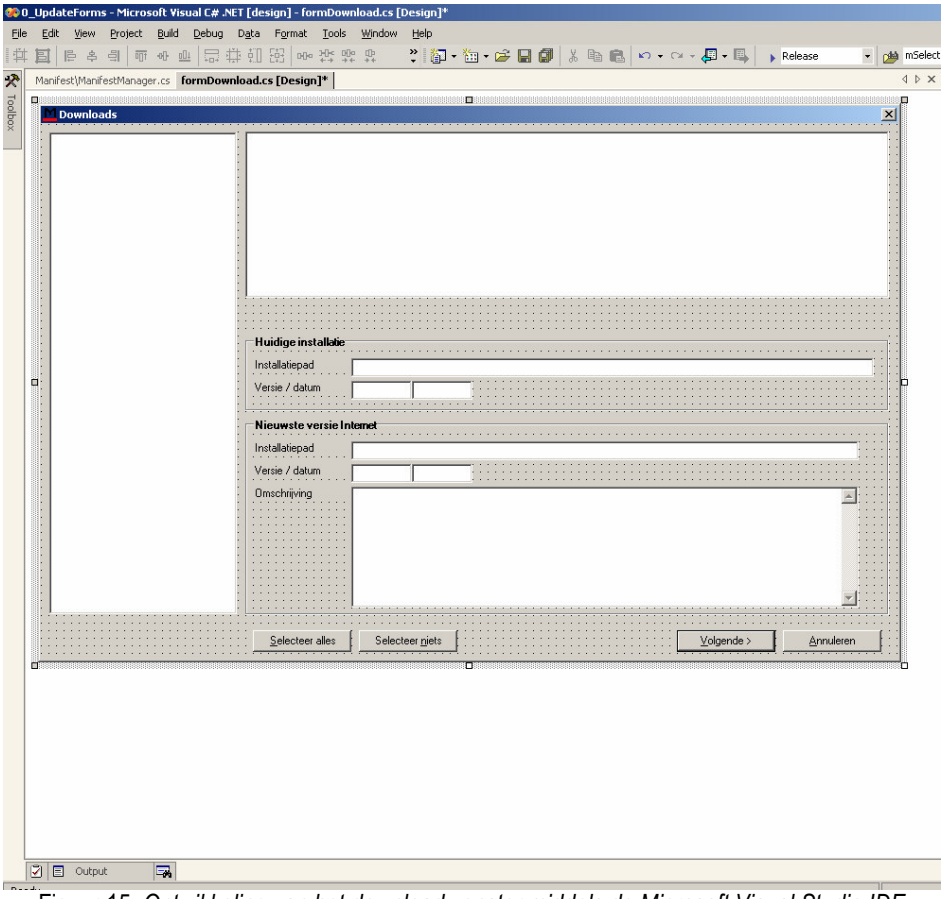
Voor het ophalen van de gegevens uit het xml-bestand en het schrijven naar het xml-bestand heb ik een klasse ontwikkeld. Het ontwerp van de zogenaamde XmlEditor klasse diende methoden te bevatten voor het ophalen en wegschrijven van gegevens omtrent de gebruiker, het softwaregebruik en de reeds binnengehaalde updates.
Nadat het ontwerp voor de XmlEditor klasse was gerealiseerd ben ik aan de slag gegaan met het programmeren van deze klasse. Aangezien de C# programmeertaal standaard mogelijkheden biedt op het gebied van xml, was ik in staat om een klasse te ontwikken waarin methoden zijn geïmplementeerd voor het ophalen en wegschrijven van gegevens in xml-bestanden. Middels deze klasse zou ik dan in de toekomst de afhandelingen met betrekking tot xml-bestanden kunnen verrichten.

© 2005 Mathieu Brüning (20010019)

Ontwikkelen download venster

Na het ontwikkelen van de download functionaliteit en de functionaliteit voor het vastleggen van de gegevens omtrent het softwaregebruik en reeds opgehaalde updates ben ik aan de slag gegaan met het ontwikkelen van een venster voor het downloaden. Middels de grafische designer in Microsoft Visual Studio ben ik aan de slag gegaan met het maken van een venster. Ook het download venster heb ik laten overerven van de superklasse voor vensters, zodat de gewenste standaard eigenschappen van het download venster geïmplementeerd waren.

De voornaamste gedachte achter het download venster was dat alle mogelijke updates voor alle software op eenvoudige wijze gedownload zou kunnen worden. Daarnaast was een belangrijke factor dat de gebruiker in staat moet zijn om meerdere updates tegelijkertijd te selecteren en downloaden, een functionaliteit die in het oude Marti-IT Update Centrum niet mogelijk was. Op basis van deze informatie heb ik een tweetal componenten toegevoegd aan het venster, namelijk een boom structuur met de applicaties van de huidige gebruiker en zogenaamd DataGrid waarin de gebruiker de gewenste updates kan selecteren.



Figuur 15: *Ontwikkeling van het download venster middels de Microsoft Visual Studio IDE*

Op basis van het identiteit object wordt bij het weergeven van het download venster de boom structuur gevuld met het softwareprofiel van de huidige gebruiker. De gebruikers is dan in staat om een filter toe te passen op de weergegeven updates, door simpelweg te klikken op de applicatie of het download type in de boomstructuur aan de linkerzijde van het venster. Daarnaast wordt er een controle uitgevoerd voor welke updates er beschikbaar zijn, zodat deze getoond kunnen worden in het DataGrid.

Ontwikkelen voortgang venster
Middels het ontwikkelde download venster is de gebruiker in staat gesteld om te controleren welke updates er beschikbaar zijn. Door vervolgens updates te selecteren dient de gebruiker dan de mogelijkheid te hebben om de updates te downloaden. Het feitelijke verzendproces van de updates naar de gebruiker vind plaats wanneer de gebruiker een zogenaamd voortgangsvenster getoond krijgt. In dit venster wordt getoond welke updates te gebruiker geselecteerd heeft en welke update er momenteel gedownload wordt.

Het ontwikkelen van het voortgang venster was een volgende stap in het ontwikkelen van de download functionaliteit. Het maken van het voortgang venster richtte zich hoofdzakelijk weer op het toevoegen van een form. Middels de Microsoft Visual Studio designer kon het ontwerp van het venster worden opgemaakt. Ook dit venster heb ik weer laten overerven van de superklasse voor vensters.

De functionaliteit van het voortgang venster speelde echter een belangrijke rol in het Update Centrum, aangezien het daadwerkelijke verzenden en ontvangen van gegevens middels dit venster wordt bewerkstelligd. Op basis van de geselecteerde updates in het download venster wordt er een array aangemaakt met Manifest objecten. De manifest objecten bevatten namelijk alle informatie omtrent de updates en worden daarom gebruikt voor het daadwerkelijke downloaden van de updates. De array wordt vervolgens doorgespeeld aan de SoftwareUpdaterManager klasse, welke de Manifest array omzet in een array met taken (UpdaterTasks). Deze UpdaterTasks worden vervolgens doorgespeeld naar de FTP Manager zodat het daadwerkelijke downloaden kan plaatsvinden.

De ontwikkeling van de functionaliteit van het voortgang venster richtte zich dus hoofdzakelijk op het ontwikkelen van een inventarisatiemethode voor de geselecteerde updates en het doorspelen aan de SoftwareUpdaterManager. Hiernaast richtte de ontwikkeling zich op de visualisering van het downloadproces, zoals het tonen van een voortgangmeter e.d.

Ontwikkelen basis menu
Om de vensters binnen de applicatie op eenvoudige wijze weer te kunnen gegeven heb ik een basis menu ontwikkeld.
Het ontwikkelen van het basis menu richtte zich hoofdzakelijk op het toevoegen van een statisch menu. De ontwikkeling van een dynamisch menu was gepland voor een andere pilot.

Programmeren event handling
Het voortgangsvenster (zoals beschreven in § 6.3.18) is een venster dat het download- of uploadproces visualiseert. Wanneer een gebruiker gegevens binnenhaalt of gegevens verzend wordt dit venster getoond, met de voortgang. Het weergegeven van de voortgang en de huidige status van het upload en download proces is hoofdzakelijk gebaseerd op de huidige status van de FTP Manager, aangezien de FTP Manager zorg draagt voor het daadwerkelijke verzenden en ontvangen. Om de status van de FTP Manager weer te kunnen gegeven in het voortgang venster heb ik gebruik moeten maken van events. Zo wordt er bijvoorbeeld een event afgevuurd wanneer het downloadproces is gestart en wanneer het downloadproces is geëindigd. Door een event van de FTP Manager af te vuren kan in het voortgang venster wordt er gemerkt dat er bepaalde gebeurtenissen zich hebben voortgedaan. Het programmeren van het afhandelen van events was daarom een volgende stap in het ontwikkelen van de download functionaliteit. Het programmeren van de events richtte zich hoofdzakelijk op het definiëren van events in de FTP Manager, zodat deze events ontvangen kunnen worden in het voortgang vensters. Vervolgens heb ik functionaliteiten geschreven in het voortgang venster, zodat de events op de juiste manier worden afgevangen en er een visualisatie wordt weergegeven in het venster.

© 2005 Mathieu Brüning (20010019)

Ontwikkelen extra .zip mogelijkheden
Nadat de download functionaliteit van het Update Centrum ontwikkeld was, ben ik mijn werkzaamheden gaan richtten op het ontwikkelen van extra .zip mogelijkheden. Het ontwikkelen van de extra .zip mogelijkheden was een belangrijke factor bij de ontwikkeling van het nieuwe Update Centrum, aangezien de updates op het Marti-IT Update Centrum hoofdzakelijk worden aangeleverd in het .zip formaat. Echter, de reeds ontwikkelde .zip functionaliteiten binnen de organisatie waren niet toereikend, aangezien er bijvoorbeeld nog niet de mogelijkheid was tot het inpakken van mappen en het behouden van mapstructuren in een .zip bestand.

Voor het aanpassen van de reeds bestaande .zip functionaliteiten diende ik aanpassingen te maken in de reeds bestaande Marti-IT C# library. Een klasse genaamd classZipFile bevatte de .zip functionaliteiten en ik ben aan de slag gegaan met het aanpassen van deze klasse.

Een eerste stap was het ontwikkelen van een functionaliteit om mappenstructuren te kunnen behouden in een .zip bestand. Op basis van informatie uit een module II-005 wist ik dat het inventariseren van een mappenstructuur gebeurt door middel van een recursieve methode. Zodoende ben ik aan de slag gegaan met het schrijven van een recursieve methode waarmee de mappenstructuur geïnventariseerd kon worden. Door de geïnventariseerde bestandspaden inclusief de mappen in stand te houden kan er middels de bestaande functionaliteiten voor het inpakken op eenvoudige wijze een .zip bestand worden gemaakt, waarin de mappenstructuur behouden blijft.

De volgende stap in het ontwikkelen van extra .zip mogelijkheden was het uitpakken van een zip bestand, waarbij de mappenstructuur behouden blijft. Het bewerkstelligen van deze stap richtte zich hoofdzakelijk op het toevoegen van een variabele en het creëren van directories.

De laatste stap betrof de mogelijkheid om meerdere bestanden te selecteren en in te pakken. Op basis van het reeds ontwikkelde was het alleen mogelijk om één bestand of één directory in te pakken. Door simpelweg de methode voor het inpakken van een directory erbij te pakken en aan te geven dat niet de bestanden uit een directory, maar juist een zelf gedefinieerde lijst van bestanden in te pakken, was deze functionaliteit ook geïmplementeerd.

Programmeren installatie procedure
Nadat de extra mogelijkheden met betrekking tot het .zip format waren gerealiseerd, en het mogelijk was om een ontvangen update op de juiste manier uit te pakken, ben ik aan de slag gegaan met het ontwikkelen van de installatie procedure. In de huidige situatie is het zo dat een gebruiker een update download, de update wordt vervolgens geplaatst in een map genaamd “Downloads” in de applicatiefolder en vervolgens wordt de update uitgepakt en geïnstalleerd.

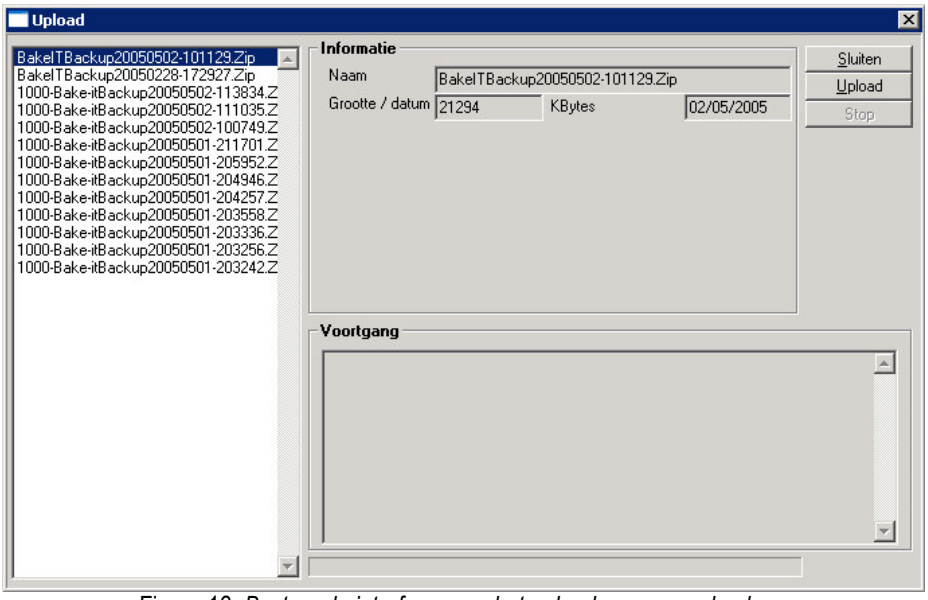
Naar mijn idee was het niet noodzakelijk om deze procedure te veranderen. Met name door het feit dat gebruikers op deze manier over de updates beschikken, kan er eventueel handmatig een update binnen een organisatie worden verspreidt, zodat de verschillende werkstations binnen een organisatie niet allemaal individueel via het Update Centrum hoeven te downloaden. Op basis van dezelfde procedure ben ik aan de slag gegaan met het ontwikkelen van een installatieprocedure voor het installeren van de updates. Zoals in de oude situatie worden de updates nu ook gedownload naar een map in de applicatiemap van de gebruiker. In deze lokale map bevind zich vervolgens een andere map met de naam van het download type, bijvoorbeeld “programma”. In de map van het download type bevinden zich de updates van dit betreffende download type. Deze lokalisering van de updates (<applicatienaam> \ “updateit” \ <download type naam> \ <update bestandsnaam>) zorgt ervoor dat binnengehaalde updates altijd teruggevonden kunnen worden, aangezien ze op een gestructureerde locatie worden weggeschreven.

Na het downloaden van de update(s) naar een lokale map kan de tweede stap van de installatieprocedure beginnen. Indien het update bestand ingepakt is wordt deze uitgepakt naar de uiteindelijke bestemmingslocatie. Indien het update bestand niet is ingepakt wordt het gekopieerd naar de bestemmingslocatie. Het ontwikkelen van de installatieprocedure richtte zich hoofdzakelijk op het programmeren van een algoritme welke zorg draagt voor het uitpakken / kopiëren van de update bestanden.
Het algoritme houdt bij het installeren van de update bestanden rekening met bestanden die mogelijk nog in gebruik zijn, bijvoorbeeld een Marti-IT programma. Indien dit het geval is wordt er een melding gegeven dat de bestanden niet geïnstalleerd konden worden en dat dit mogelijk te wijten is aan bestanden die niet overschreven konden worden doordat ze nog in gebruik waren. De gebruiker wordt er dan bewust van dat er wellicht nog bestanden in gebruik zijn en dat die afgesloten moeten worden voordat de installatie afgerond kan worden.

© 2005 Mathieu Brüning (20010019)

Ontwerpen upload functionaliteit
Nadat de functionaliteit voor het downloaden van updates via het Marti-IT Update Centrum was gecompleteerd ben ik aan de slag gegaan met het ontwerpen van de upload functionaliteit. De systeemeisen omtrent het uploaden van gegevens, zoals gedefinieerd in de definitiestudie, richtte zich hoofdzakelijk op drie functionaliteiten, namelijk het uploaden van gegevensback-ups, het uploaden van bestanden en/of mappen en de mogelijkheid tot het uploaden van standaard mappen. Deze drie functionaliteiten voor het uploaden van gegevens waren in het oorspronkelijke Update Centrum ook geïmplementeerd en diende zeer zeker terug te keren in het nieuwe systeem. Mijn vraag was alleen hoe ik dat het meest efficiënt zou kunnen ontwerpen.

Het uploaden van gegevensback-ups is een functionaliteit die simpelweg een gemaakte gegevensback-up (ingepakt in .zip formaat) verzend naar de Marti-IT FTP server. De medewerkers van Marti-IT kunnen dan vervolgens de back-up handmatig downloaden, zodat de data gebruikt kan worden voor ondersteuning en archivering.

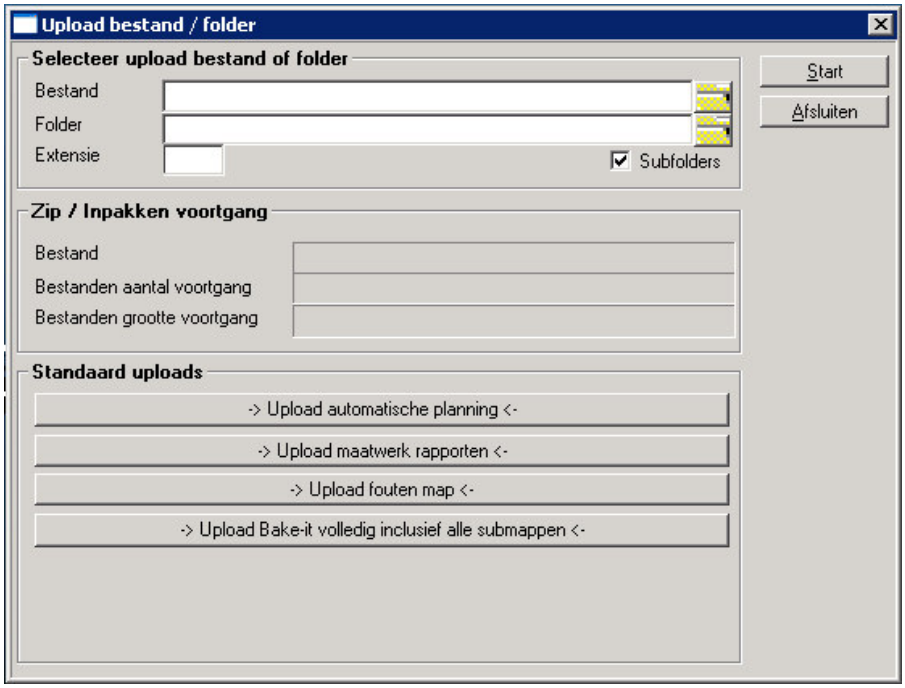


Figuur 16: *Bestaande interface voor het uploaden van een back-up*

De tweede functionaliteit richt zich op het verzenden van elke willekeurige map of bestand. Een gebruiker kan middels een interface een map of bestand selecteren, waarop deze vervolgens wordt ingepakt en wordt verzonden naar de Marti-IT FTP server.

De laatste mogelijkheid tot het uploaden van gegevens richt zich op de zogenaamd ‘standaard uploads’. Een standaard upload houdt in dat er vooraf gedefinieerd is welke map ingepakt zal worden en zal worden verzonden. Middels de ‘standaard uploads’ is het namelijk gemakkelijk om een bepaalde map op eenvoudige wijze te verzenden naar Marti-IT, zodat de gebruiker niet steeds geconfronteerd hoeft te worden met het selecteren van een map. Het nadeel aan de ‘standaard uploads’ is echter dat de keuzemogelijkheden statisch zijn. Door middel van een aantal vaste knoppen op de interface kunnen gebruikers de standaard gegevens verzenden, wat betekent dat de keuze mogelijkheden voor het verzenden van de standaard gegevens beperkt zijn.

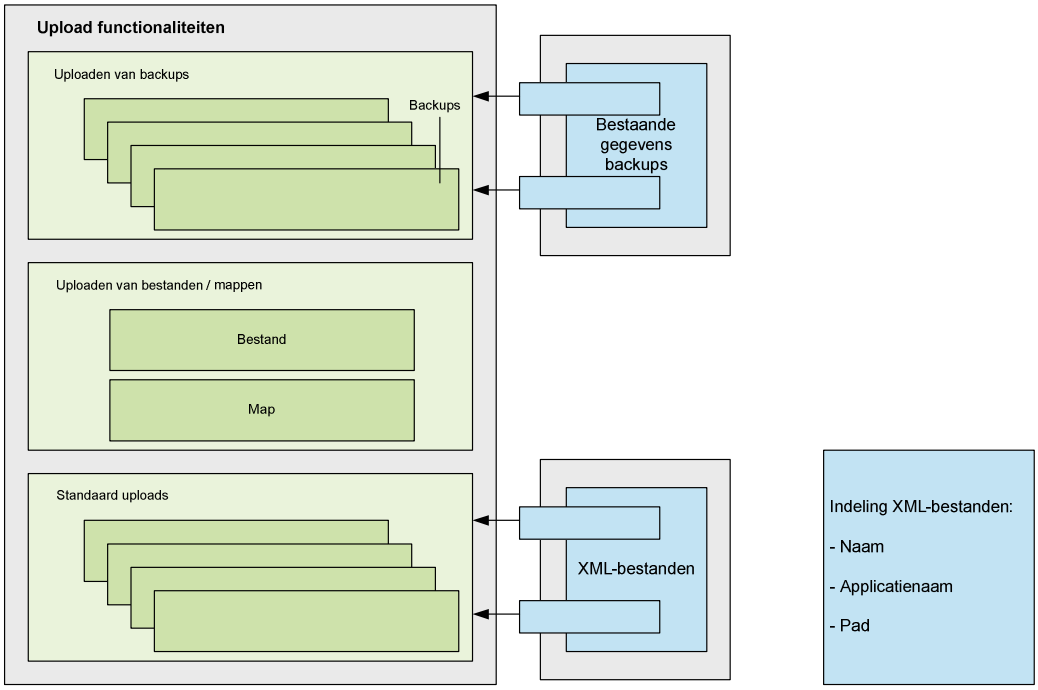
© 2005 Mathieu Brüning (20010019)



Figuur 17: *Bestaande interface voor het uploaden van bestanden, mappen of standaard uploads. De (statische) knoppen onderaan het venster kunnen gebruikt worden voor het uploaden van voorgedefinieerde mappen*

Naar mijn mening was de functionaliteit voor het uploaden van de back-ups en de functionaliteit voor het uploaden van bestanden en/of mappen wel in orde. Met name doordat deze functionaliteiten vrij recht-toe-recht-aan zijn, namelijk alleen het verzenden van ingepakte bestanden naar de Marti-IT server, wilde ik veranderingen in de nieuwe upload functionaliteiten met name richten op de 'standaard uploads'. De statische beperking van het aantal uploads wilde ik sowieso teniet doen, zodat de 'standaard uploads' in de toekomst op eenvoudige uitgebreid zouden kunnen worden.
De eerste gedachte bij het dynamisch maken van de standaard upload functionaliteiten was vrijwel direct het gebruik van XML. Het gebruik XML bestanden is namelijk een universeel geaccepteerde standaard voor het structureren van gegevens. Daarnaast biedt Microsoft .NET en de Microsoft Visual Studio IDE zeer goede ondersteuning wat betreft het ontwikkelen met XML. Met name met de Visual Studio IDE kan op eenvoudige wijze XML geïmplementeerd worden in projecten, zonder dat extra tools gebruikt hoeven te worden.

Het dynamisch maken van de standaard uploads diende dus plaats te vinden middels XML bestanden. De gedachtegang hierachter kwam er hoofdzakelijk op neer dat er voor elke standaard upload een XML bestand aanwezig is in een submap van het Update Centrum. Deze submap <updateit pad>\Uploads\UploadTemplates wordt bij het aanroepen van de standaard upload interface geraadpleegd, zodat er geïnventariseerd kan worden welke XML bestanden er aanwezig zijn. De XML bestanden hebben vervolgens een drietal attributen, namelijk een naam voor de standaard upload, een applicatienaam en een pad naar een map. Op basis van de applicaties in het Identiteit object voor de huidige gebruiker, kan gecontroleerd worden welke standaard uploads van toepassing zijn op de huidige gebruiker. Deze worden vervolgens weergegeven in de interface voor de standaard uploads, zodat de gebruiker de gegevens kan verzenden.



Figuur 18: *Globaal ontwerp upload functionaliteiten*

Nadat de drie globale functionaliteiten voor het upload systeem waren ontworpen ben ik gaan analyseren op welke manier ik de functionaliteiten zou kunnen integreren in de reeds ontwikkelde functionaliteit met betrekking tot het downloaden van updates. Na enige tijd bleek echter dat de uit te voeren stappen voor het downloaden van een update niet eens zo heel veel afwijken van de uit te voeren stappen voor het uploaden van een backup.

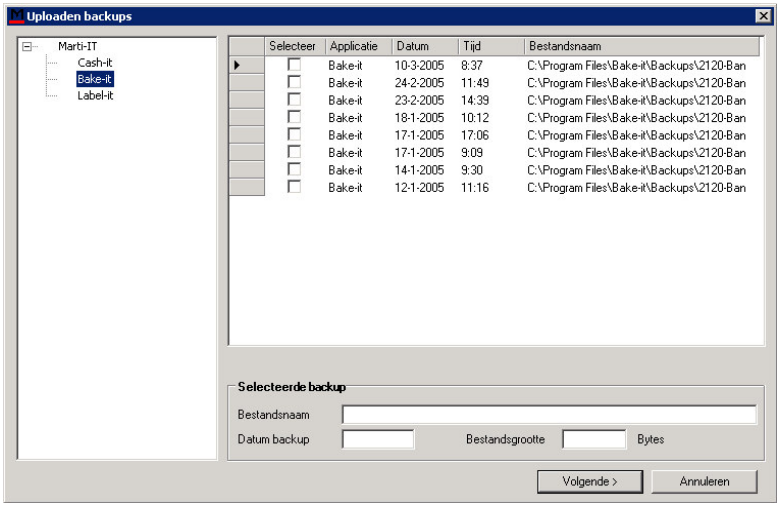
Downloaden updates	Uploaden backups
1. Controleer welke updates er beschikbaar zijn	1. Inventariseer welke back-ups er beschikbaar zijn
2. Vergaar informatie omtrent de beschikbare updates	2. Vergaar informatie omtrent de beschikbare back-ups
3. Maak download taken aan voor de geselecteerde updates	3. Maak upload taken aan voor de geselecteerde back-ups
4. Download de updates	4. Upload de back-ups
5. (Pak de updates uit en installeer deze)	

Over het algemeen lijkt het er vooral op dat de te volgen stappen bij het downloaden van updates en het uploaden van back-ups vrijwel identiek zijn. Het integreren van de upload functionaliteiten richtte zich zodoende hoofdzakelijk op het toevoegen van methodes aan de SoftwareUpdaterManager klasse (voor het inventariseren van back-ups en standaard uploads), de ManifestManager klasse (voor het vergaren van informatie van back-ups en standaard uploads) en de FTPManager klasse (voor het uploaden van gegevens). Op basis van deze informatie ben ik aan de slag gegaan met het programmeren van de upload functionaliteit.

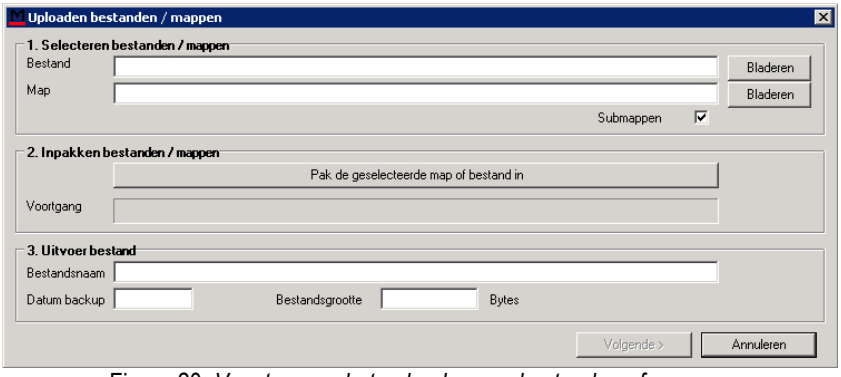
Programmeren upload functionaliteit
Op basis van het ontwerp van de upload functionaliteit ben ik aan de slag gegaan met het programmeren van de upload functionaliteit. Door met name een aantal functionaliteiten toe te voegen aan de SoftwareUpdaterManager klasse, de ManifestManager klasse en de FTPManager klasse kon de upload functionaliteit gerealiseerd worden.
Aangezien de functionaliteiten grotendeels overeenkwamen met de functionaliteiten voor het downloaden van updates, richtte het programmeren zich met name op het hergebruiken van de methoden voor het downloaden van updates, om die vervolgens ietwat aan te passen. Op deze manier kon de functionaliteit voor het uploaden van gegevens gerealiseerd worden.

Maken vensters voor upload functionaliteit
Nadat de functionaliteit voor het uploaden van gegevens was gerealiseerd, ben ik aan de slag gegaan met het maken van interfaces. Middels de Microsoft Visual Studio IDE was ik in staat om op eenvoudige wijze de vensters toe te voegen door een drietal forms toe te voegen aan mijn project. Voor elke functionaliteit voor het uploaden van gegevens (uploaden back-ups, uploaden bestanden/mappen, uploaden standaard gegevens) heb ik een venster gemaakt, wat feitelijk overeenkwam met de venster indeling in het bestaande Update Centrum.

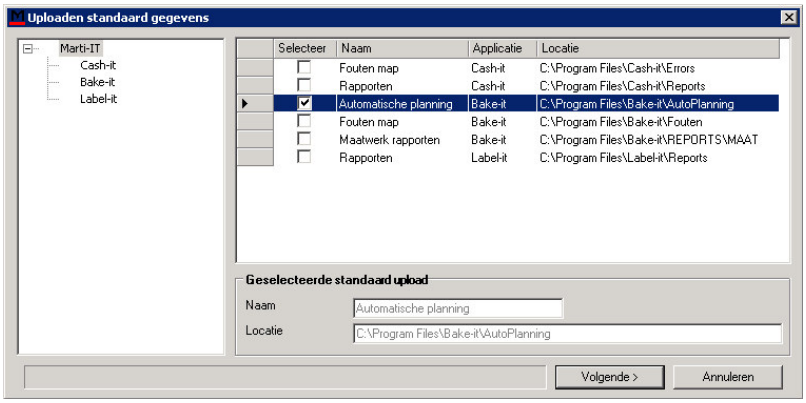
Nadat de vensters grafisch waren ontwikkeld ben ik aan de slag gegaan met het coderen van de logica achter de vensters. De voornaamste logica met betrekking tot de upload functionaliteiten was al geïmplementeerd in de SoftwareUpdaterManager klasse, maar het aanroepen van de SoftwareUpdater Manager diende nog plaats te vinden in de vensters. Nadat de logica op de vensters gekoppeld was aan de overige benodigde logica in het project waren de upload functionaliteiten gerealiseerd.



Figuur 19: *Venster voor het uploaden van een back-up*



Figuur 20: *Venster voor het uploaden van bestanden of mappen*



Figuur 21: *Venster voor het uploaden van standaard uploads*

Online plaatsen van de database

Na het completeren van de functionaliteit voor het verzenden van gegevens, waren de ontwikkelingen voor de eerste pilot zo goed als afgerond. De testactiviteiten voor pilot 1 diende nog plaats te vinden en dan zou de pilot kunnen worden afgerond. Om in een zo realistisch mogelijke omgeving het systeem te testen wilde ik de database online plaatsen, zodat mijn testwerkzaamheden uitgevoerd konden worden in een omgeving die representatief was voor de toekomstige omgeving.
Echter, ik was nog niet op de hoogte hoe de MySQL online geplaatst zou kunnen worden. Het zoeken van informatie op het Internet leverde echter weinig tot geen informatie op. Zodoende ben ik vervolgens eens gaan kijken hoe de andere MySQL databases binnen Marti-IT zijn geplaatst op het Internet. Middels de MySQL Control Center tool, waar ook het inrichten van de database mee bewerkstelligd was, was het mogelijk om de database te plaatsen op een server. Middels het MySQL Control Center heb ik vervolgens de database geplaatst op de Marti-IT server, het mogelijk was om via het Internet informatie kunnen opvragen middels de database.

Testen pilot 1

Het testen van pilot 1 kon van start gaan toen de functionaliteiten voor het downloaden en uploaden van gegevens mogelijk was en de database was gevuld met testgegevens. Op basis van de functionele specificaties van het oude Marti-IT Update Centrum (zoals definieert in de definitiestudie) was het over het algemeen vrij helder om de ontwikkelde functionaliteiten te vergelijken met de bestaande functionaliteit.
Door middel van met name Black Box tests heb ik getest of de functionaliteiten van het nieuwe Marti-IT Update Centrum voldoende overeenkomen met de functionaliteiten van het oude Marti-IT Update Centrum. Tijdens de testwerkzaamheden bleek dat er nog wel enige bugs zaten in het systeem. Dit betroffen met name kleine logische fouten in de functionaliteiten waarbij bijvoorbeeld bepaalde zaken waren gekopieerd, maar niet op de volledige juiste manier waren aangepast. Naast de kleine logische fouten was er ook nog een probleem met de FTPManager. De FTPManager had problemen met het verbinden naar de Marti-IT FTP server wanneer er langere tijd geen commando's waren verzonden naar de server. Er was dan sprake van een timeout en hier crashte de applicatie op. Dit kon echter worden opgelost door enige aanpassingen te realiseren in de FTPclient, zodat in de toekomst in eerst instantie gecontroleerd word of het mogelijk was om een commando naar de server te verzenden. Indien de fout dan optreed, wordt deze afgevangen en wordt er opnieuw verbonden met de FTP server.
Na het uitvoeren van de testwerkzaamheden ben ik tot de conclusie gekomen dat de eerste pilot was afgerond. De functionaliteiten, zoals deze geïmplementeerd waren in het oude Update Centrum, waren ook allemaal geïmplementeerd in het nieuwe systeem.

Het afronden van de testactiviteiten richtte zich op het beschrijven van de resultaten van de testwerkzaamheden. Doordat de eerste pilot qua tijdsduur was uitgelopen (planning 6 weken werd 7 weken) heb ik besloten om geen testactiviteiten meer uit te voeren met betrekking tot White Box testen. De reden dat ik heb besloten om de eerste pilot uit te laten lopen in plaats van door te gaan met de tweede pilot was met name dat de functionaliteit van het bestaande Update Centrum sowieso gerealiseerd diende te worden.

© 2005 Mathieu Brüning (20010019)

6.3.2 Pilot 2: Ontwikkelen extra functionaliteiten

Opstellen pilotontwikkelplan
Het begin van de tweede pilot richtte zich op het opstellen van een pilotontwikkelplan. In het pilotontwikkelplan diende opnieuw op concrete wijze vastgelegd te worden welke bouweenheden er ontwikkeld zouden worden tijdens deze pilot. Op basis van de globale structuur uit het reeds opgestelde pilotplan en de daarbij behorende systeemeisen ben ik aan de slag gegaan met het opstellen van het pilotontwikkelplan.
De eerste stappen in het opstellen van het plan betroffen het definiëren van de globaal-functionele, globaal-technische en globaal-organisatorische structuur van de pilot. Vervolgens ben ik aan de slag gegaan met het structureren van de pilot en het gedetailleerd in kaart brengen van de uitvoering van de pilot. De structuur van de ontwikkeling van de pilot kwam er als volgt uit te zien:

Marti-IT Build Centrum:

- ✓ Ontwikkelen van een functionaliteit met betrekking tot het beheren van gebruikersgegevens
- ✓ Ontwikkelen van een functionaliteit met betrekking tot het beheren van gegevens omtrent de Marti-IT software
- ✓ Ontwikkelen van een functionaliteit met betrekking tot het beheren van gegevens omtrent updates
- ✓ Ontwikkelen van een functionaliteit met betrekking tot het automatiseren en beheren van verzonden gegevens door gebruikers

Marti-IT Update Centrum:

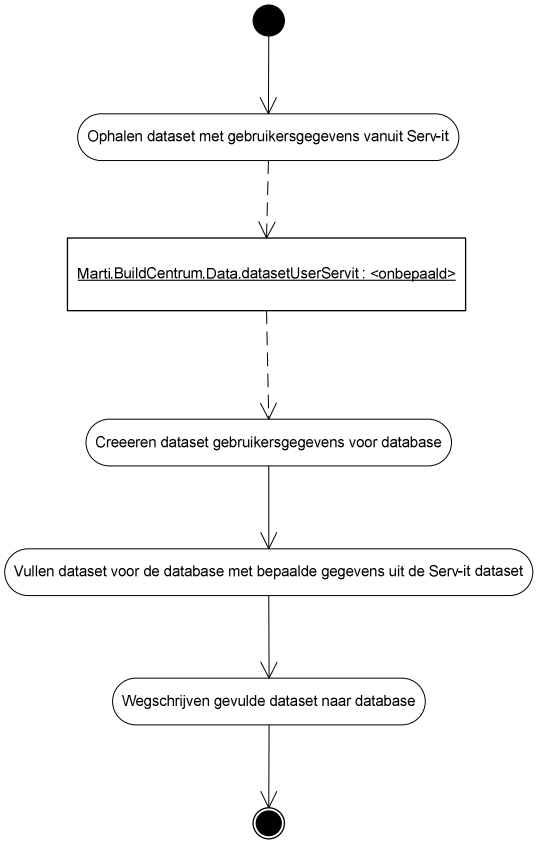
- ✓ Ontwikkelen van extra functionaliteiten met betrekking tot de gebruikers van de Marti-IT software

Nadat de structuur van de pilot volledig was uiteengezet ben ik aan de slag gegaan met het opstellen van een test- en beoordelingsplan voor de pilot. Het opstellen van het test- en beoordelingsplan was hoofdzakelijk gebaseerd op de systeemeisen en de globale testprocedure, zoals gedefinieerd in het rapport met de specificaties van het systeem.

Maken basis interface Marti-IT Build Centrum
Het zogenaamde Marti-IT Build Centrum, het systeem voor de medewerkers van Marti-IT, is een applicatie die ondermeer het beheer van de beschikbare updates op het Marti-IT Update Centrum moet verzorgen. Daarnaast dient het systeem bepaalde processen binnen Marti-IT, zoals bijvoorbeeld het klaarzetten van updates, te automatiseren. De bouw van het Build Centrum is van start gegaan door in eerste instantie een basis interface te maken. Net als bij het Update Centrum is het gemakkelijk om bepaalde functionaliteiten te testen aan de hand van een interface en zodoende ben ik begonnen met het maken van de basisinterface. Het maken van de basisinterface richtte zich feitelijk alleen op het toevoegen van een venster en dit venster laten erven van de standaard venster klasse.

Ontwikkelen functionaliteit met betrekking tot het beheren van gebruikersgegevens
Nadat de basis interface gemaakt was ben ik aan de slag gegaan met het ontwerpen van de functionaliteit voor het beheren van de gebruikersgegevens in de database. Een fundamenteel aspect van het ontwerp richtte zich op de consistentie van gebruikersgegevens. De gegevens omtrent gebruikers worden binnen Marti-IT in een applicatie genaamd Serv-it opgeslagen. Het Marti-IT Update / Build Centrum maakt daarnaast ook gebruik van een database, waarin bepaalde gegevens omtrent gebruikers worden opgeslagen. Aangezien de informatie omtrent gebruikers op twee plaatsen wordt opgeslagen is het van vitaal dat de informatie in de twee databases synchroon blijven. In het ontwerp van de functionaliteit met betrekking het beheren van de gebruikersgegevens diende dus rekening gehouden te worden met het synchroniseren van de gebruikersgegevens. Op basis van dit gegeven ben ik aan de slag gegaan met het ontwerpen van een functionaliteit voor het synchroniseren van de gebruikersgegevens, zodat wijzigingen in de Serv-it administratie kunnen worden doorgevoerd in de database van het Marti-IT Update / Build Centrum. Op de volgende pagina wordt het ontwerp weergegeven:

© 2005 Mathieu Brüning (20010019)



Figuur 22: *Ontwerp synchronisatie functionaliteit*

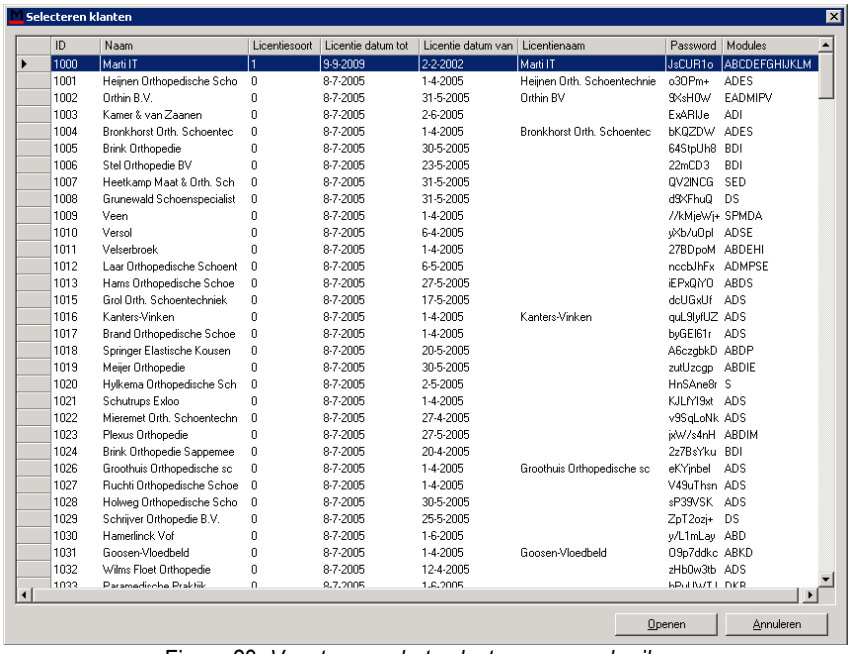
Nadat de functionaliteit voor het synchroniseren van gebruikersgegevens was ontworpen ben ik aan de slag gegaan met het programmeren van de functionaliteit. Op basis van het ontwerp was de eerste logische stap in het synchronisatieproces het ophalen van een dataset met gegevens vanuit de Serv-it applicatie. De Serv-it applicatie maakt gebruik van een ADS database, wat als gevolg had dat ik gebruik diende te maken van een additioneel component, namelijk de Advanced Data Provider (ADP). Door middel van het toepassen van de ADP was het mogelijk om SQL statements uit te voeren op de ADS database, zodat ik in staat was om gegevens op te halen uit Serv-it. Op basis van een bestaande C# applicatie van Marti-IT, welke ook gebruikt maakt van een ADS database en de ADP had ik een voorbeeld hoe het gebruik van de ADP in zijn werk ging. Nadat eenmaal een dataset met gegevens was opgehaald uit Serv-it konden de gegevens op eenvoudige wijze worden gesynchroniseerd met de MySQL database, door middel van het algoritme gebaseerd op het ontwerp.

Nadat de functionaliteit voor het synchroniseren van de gebruikersgegevens was ontwikkeld ben ik aan de slag gegaan met het ontwikkelen van vensters voor het inzien van gebruikersgegevens en het synchroniseren van de gebruikersgegevens. Dit resulteerde in een drietal vensters, namelijk:

- ✓ Een venster voor het selecteren van een gebruiker
- ✓ Een venster voor het inzien van de gegevens van de geselecteerde gebruiker
- ✓ Een venster voor het synchroniseren van de gebruikersgegevens

De ontwikkeling van het venster voor het selecteren van een gebruiker richtte zich hoofdzakelijk op het ontwikkelen van een venster waarin een lijst werd getoond met de gebruikersgegevens, zodat een rij in de lijst kan worden aangeklikt en worden geopend. Het venster bestaat hierbij voornamelijk uit een *datagrid* en een tweetal knoppen voor het openen van een gebruikerskaart en het sluiten van het venster.

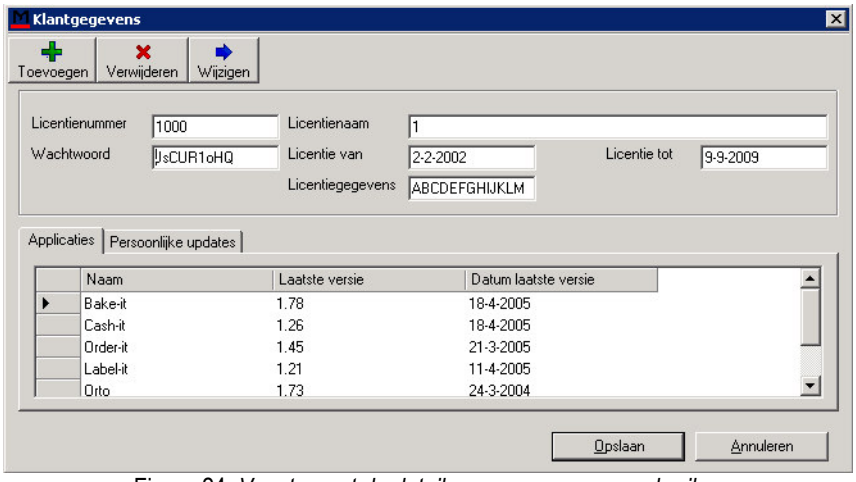
© 2005 Mathieu Brüning (20010019)



ID	Naam	Licentiesoort	Licentie datum tot	Licentie datum van	Licentienaam	Password	Modules
1000	Marti IT	1	9-9-2009	2-2-2002	Marti IT	JsCUR1o	ABCDEFGHIJKLM
1001	Heijnen Orthopedische Scho	0	8-7-2005	1-4-2005	Heijnen Orth. Schoentechnie	o3OPm+	ADES
1002	Orthin B.V.	0	8-7-2005	31-5-2005	Orthin BV	9KsH0w	EADMIPV
1003	Kamer & van Zaanen	0	8-7-2005	2-6-2005		ExARIJe	ADI
1004	Bronkhorst Orth. Schoentec	0	8-7-2005	1-4-2005	Bronkhorst Orth. Schoentec	bKQZDW	ADES
1005	Brink Orthopedie	0	8-7-2005	30-5-2005		64StpUh8	BDI
1006	Stel Orthopedie BV	0	8-7-2005	23-5-2005		22mCD3	BDI
1007	Heetkamp Maat & Orth. Sch	0	8-7-2005	31-5-2005		QV2INCG	SED
1008	Grunewald Schoenspecialist	0	8-7-2005	31-5-2005		d8XFhuQ	DS
1009	Veen	0	8-7-2005	1-4-2005		//kMjeWj+	SPMDA
1010	Versol	0	8-7-2005	6-4-2005		yKb/uOpl	ADSE
1011	Velserbroek	0	8-7-2005	1-4-2005		27BDpoM	ABDEHI
1012	Laar Orthopedische Schoent	0	8-7-2005	6-5-2005		nccbJhFx	ADMPSE
1013	Hams Orthopedische Schoe	0	8-7-2005	27-5-2005		iEPxQiYO	ABDS
1015	Grol Orth. Schoentechniek	0	8-7-2005	17-5-2005		dcUGxUf	ADS
1016	Kanters-Vinken	0	8-7-2005	1-4-2005	Kanters-Vinken	quL9lyfUZ	ADS
1017	Brand Orthopedische Schoe	0	8-7-2005	1-4-2005		byGEl61r	ADS
1018	Springer Elastische Kousen	0	8-7-2005	20-5-2005		A6czgbkD	ABDP
1019	Meijer Orthopedie	0	8-7-2005	30-5-2005		zutUzcgp	ABDIE
1020	Hylkema Orthopedische Sch	0	8-7-2005	2-5-2005		HnSAne8r	S
1021	Schutrups Exloo	0	8-7-2005	1-4-2005		KJLfYI9xt	ADS
1022	Mieremet Orth. Schoentechn	0	8-7-2005	27-4-2005		v9SqLoNk	ADS
1023	Plexus Orthopedie	0	8-7-2005	27-5-2005		jxW/s4nH	ABDIM
1024	Brink Orthopedie Sappemee	0	8-7-2005	20-4-2005		2z7BsYku	BDI
1026	Groothuis Orthopedische sc	0	8-7-2005	1-4-2005	Groothuis Orthopedische sc	eKYjnbel	ADS
1027	Ruchti Orthopedische Schoe	0	8-7-2005	1-4-2005		V49uThsn	ADS
1028	Holweg Orthopedische Scho	0	8-7-2005	30-5-2005		sP39VSK	ADS
1029	Schrijver Orthopedie B.V.	0	8-7-2005	25-5-2005		ZpT2ozj+	DS
1030	Hamerlinck Vof	0	8-7-2005	1-6-2005		y/L1mLay	ABD
1031	Goosen-Vloedbeld	0	8-7-2005	1-4-2005	Goosen-Vloedbeld	O9p7ddkc	ABKD
1032	Wilms Floet Orthopedie	0	8-7-2005	12-4-2005		zHb0w3tb	ADS

Figuur 23: *Venster voor het selecteren van gebruikers*

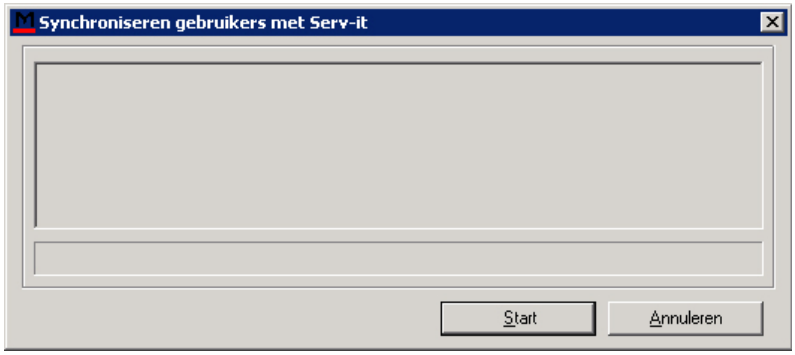
Het venster voor het inzien van gebruikersgegevens bestaat uit een aantal velden, waarin de gegevens worden vermeldt van de geselecteerde gebruiker. Aan de onderkant van het venster bevinden zich twee tabbladen waarop enerzijds het gebruik van de Marti-IT applicaties gezien kan worden en anderzijds de persoonlijke updates van de betreffende gebruiker worden getoond.



Naam	Laatste versie	Datum laatste versie
Bake-it	1.78	18-4-2005
Cash-it	1.26	18-4-2005
Order-it	1.45	21-3-2005
Label-it	1.21	11-4-2005
Orto	1.73	24-3-2004

Figuur 24: *Venster met de detailgegevens van een gebruiker*

Het venster met betrekking tot het synchroniseren van de gebruikersgegevens bestaat hoofdzakelijk uit een tweetal knoppen voor het starten van het synchronisatieproces en het sluiten van het venster. Daarnaast bevindt er zich een voortgangsmeter en een statusveld, zodat de voortgang van het synchronisatieproces gevolgd kan worden.



Figuur 25: *Venster voor het synchroniseren van gebruikersgegevens met Serv-it*

Nadat de vensters waren ontwikkeld heb ik de ontwikkelde functionaliteit gekoppeld aan de vensters, zodat het mogelijk werd om gebruikersgegevens te synchroniseren en in te zien.

Ontwikkelen functionaliteit met betrekking tot het beheren van software

He ontwikkelen van de functionaliteit met betrekking tot het beheren van software in de database richtte zich in eerste instantie op het ontwerpen van de betreffende functionaliteit. Op basis van het gegevensmodel uit het rapport met de specificaties van het systeem bleek dat het beheer van de software zich richt op de volgende aspecten:

- ✓ De Marti-IT software bestaat uit een verzameling van applicaties, waarvoor updates gedownload kunnen worden
- ✓ De updates voor een applicatie zijn onderverdeeld in download types, zoals bijvoorbeeld programma, rapporten, etc.

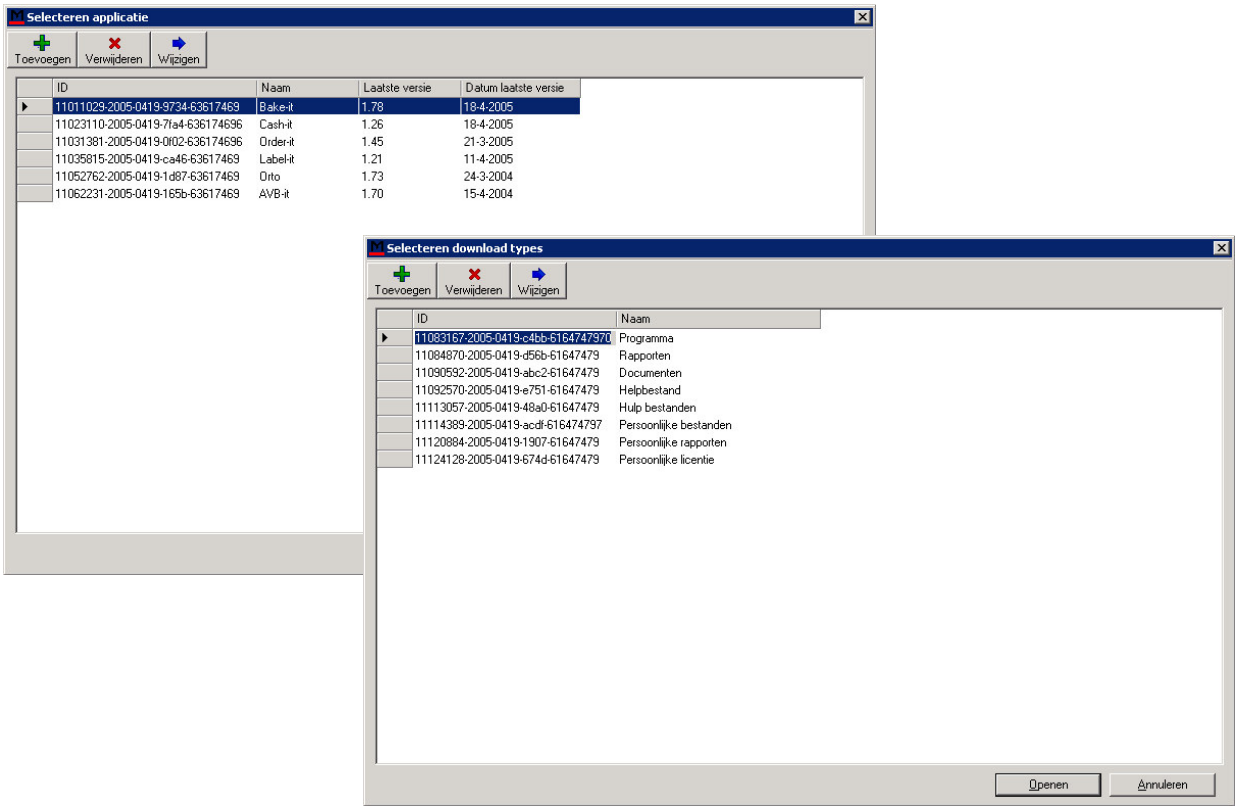
Het ontwerp van de functionaliteit voor het beheren van de software kenmerkte zich zodoende door de volgende functionaliteiten:

- ✓ Het onderhouden (toevoegen, verwijderen, wijzigen) van applicaties in de database
- ✓ Het onderhouden (toevoegen, verwijderen, wijzigen) van download types in de database
- ✓ Het koppelen van download types aan applicaties

Op basis van dit ontwerp ben ik aan de slag gegaan met het programmeren van de functionaliteiten. Het programmeren van de functionaliteiten voor het onderhouden van de applicaties en download types richtte hoofdzakelijk op het aansturen van de classODBCDatabaseManager klasse, zodat deze kon zorgen voor het toevoegen, verwijderen of wijzigen van de gegevens middels SQL statements.
Nadat de functionaliteit voor het onderhouden van de softwaregegevens was ontwikkeld ben ik aan de slag gegaan met het maken van vensters voor het onderhoud. Hierbij was er sprake van een viertal vensters namelijk:

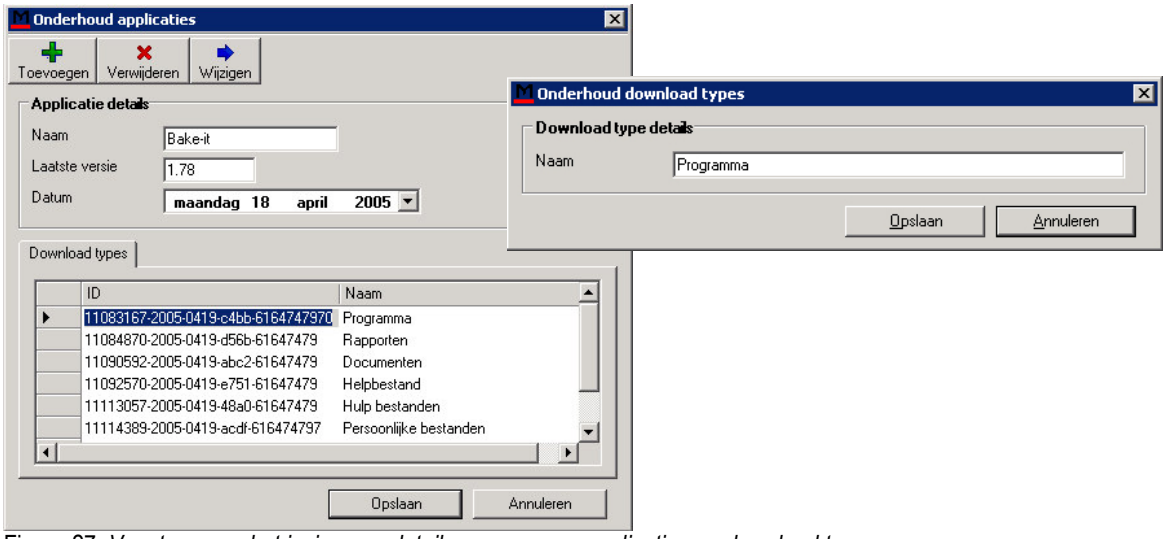
- ✓ Een venster voor het selecteren van een applicatie
- ✓ Een venster voor het inzien van de gegevens van de geselecteerde applicatie
- ✓ Een venster voor het selecteren van een download type
- ✓ Een venster voor het inzien van de gegevens van het geselecteerde download type

De vensters voor het selecteren van een applicatie of een download type konden op eenvoudige wijze worden nagemaakt van het venster voor het selecteren van een gebruiker. Hierbij was er in feite alleen maar sprake van andere data om het venster mee te vullen en andere indeling van de kolommen.



Figuur 26: *Vensters voor selecteren van applicatie en download types*

De vensters voor het inzien van de detailgegevens van een applicatie of download type konden ook gebaseerd worden op het venster voor het inzien van gebruikersgegevens. Hierbij diende echter wel de velden aangepast te worden, zodat de corresponderende velden op het venster werden getoond.



Figuur 27: *Vensters voor het inzien van detailgegevens van applicaties en download types*

Nadat de vensters waren ontwikkeld ben ik aan de slag gegaan met het koppelen van de ontwikkelde functionaliteiten voor het beheren van de software aan de ontwikkelde vensters. Het resultaat betrof de mogelijkheid tot het beheren van de software entiteiten in de database, door middel van het toevoegen, verwijderen en wijzigen van applicaties en download types en het koppelen van download types aan applicaties.

Ontwikkelen functionaliteit met betrekking tot het beheren van updates
Het belangrijkste onderdeel van het Marti-IT Build Centrum betreft de mogelijkheid tot het beheren van de updates van het Marti-IT Update Centrum. Het beheren van de updates dient namelijk zorg te dragen van het consistent houden van de informatie omtrent updates in de database en het consistent houden van de update bestanden op de Marti-IT server.

De eerste stap in de ontwikkeling van deze functionaliteit richtte zich op het ontwerpen van de functionaliteit, zodat in kaart gebracht zou kunnen worden hoe de functionaliteit qua structuur in elkaar diende te zitten. Het ontwerp was gebaseerd op de volgende functionele kenmerken, namelijk:

- ✓ Het onderhouden (toevoegen, verwijderen, wijzigen) van informatie omtrent updates in de database
- ✓ Het onderhouden (toevoegen, verwijderen, wijzigen) van update bestanden op de Marti-IT server

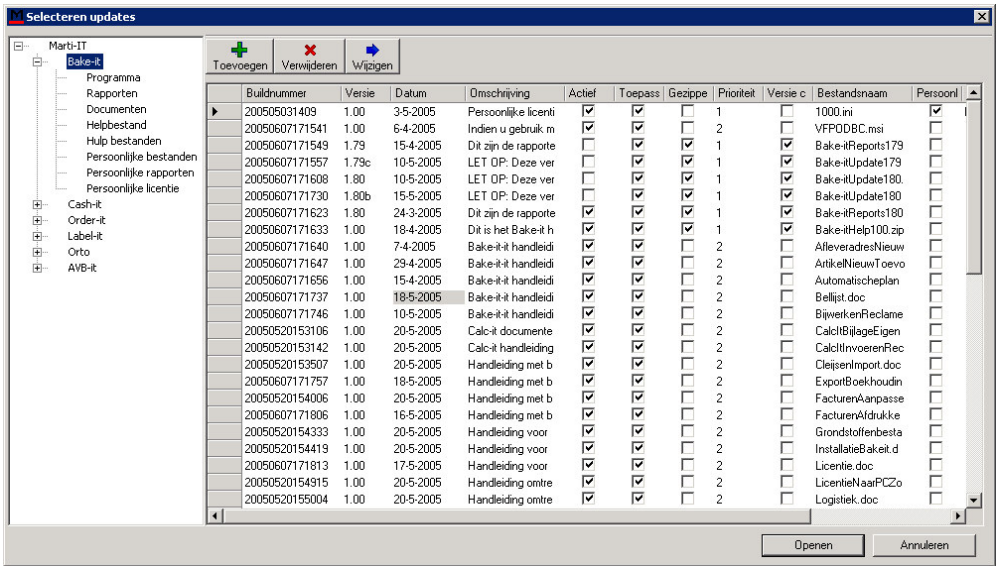
Op basis van dit ontwerp ben ik aan de slag gegaan met het programmeren van de functionaliteit. Hierbij ben ik in eerste instantie aan de slag gegaan met het programmeren van de functionaliteit voor het onderhouden van de informatie omtrent updates in de database. Dit betekende over het algemeen hetzelfde als bij het beheren van de gegevens omtrent de Marti-IT applicaties, namelijk het toevoegen, verwijderen en wijzigen van informatie omtrent updates in de database middels de classODBCDatabaseManager klasse. Middels deze klasse kunnen namelijk alle afhandelingen voor het ophalen en wegschrijven van gegevens naar de database worden gerealiseerd.

Nadat de functionaliteit voor het toevoegen, verwijderen en wijzigen van update informatie in de database was gerealiseerd ben ik aan de slag gegaan met een functionaliteit voor het maken van update bestanden. De update bestanden, zoals deze op de Marti-IT server staan, zijn hoofdzakelijk bestanden in het .zip formaat. Het maken van een functionaliteit voor het maken van updates richtte zich dus hoofdzakelijk op het maken van een functionaliteit waarmee bestanden of mappen ingepakt zouden kunnen worden. Op basis van de reeds ontwikkelde functionaliteit voor het uploaden van bestanden en/of mappen ben ik aan de slag gegaan met het voortborduren op deze functionaliteit, zodat ik een component had waarmee ik een enkel bestand, meerdere bestanden of een mappenstructuur zou kunnen inpakken. Op die manier zou ik een functionaliteit hebben voor het maken van de update bestanden, zodat de gemaakte update bestanden alleen nog op de Marti-IT server geplaatst hoeven te worden.

De mogelijkheid tot het maken van update bestanden was inmiddels bewerkstelligd en zodoende ben ik aan de slag gegaan met het maken van de functionaliteit voor het plaatsen van de update bestanden op de Marti-IT server. Middels de reeds ontwikkelde FTPManager klasse was het mogelijk om de FTPClient aan te sturen en zodoende bestanden te downloaden en uploaden. Voor het uploaden van de updates heb ik vervolgens een extra methode gemaakt voor het uploaden van updates. Deze methode is afgeleid van een methode voor het uploaden van back-ups door gebruikers, met het verschil dat de locatie van de bestanden op de Marti-IT server verschilt.

De functionaliteit voor het uploaden van update bestanden was inmiddels ook gerealiseerd en mijn volgende stap was het maken van de vensters voor het beheren van de updates. Net als bij het ontwikkelen van de vensters voor het inzien van gebruikers- en softwaregegevens ben ik in eerste instantie aan de slag gegaan met het maken van een selectievenster, waarin de reeds bestaande updates worden getoond. Een fundamenteel verschil met de selectievensters van de gebruikers- en softwaregegevens was echter dat er de mogelijkheid zou moeten zijn tot het filteren van de updates. De gedachtegang hierachter was hoofdzakelijk dat de lijst met updates erg groot is en dat het zodoende moeilijk zoeken is naar een specifieke update. Op basis van dit gegeven heb ik besloten om een boomstructuur op het venster te plaatsen, zoals ook op het downloadvenster. De boomstructuur wordt vervolgens gevuld met alle applicaties en download types, zodat het mogelijk is om een filter toe te passen op de weergegeven updates, door middel van een muisklik op een applicatie of download type in de boomstructuur. Nadat de mogelijkheid tot het filteren van de updates was geïmplementeerd in het venster betrof het uiteindelijk resultaat van deze activiteit een selectievenster, waarin de updates geselecteerd en geopend kunnen worden en waarop filters toegepast kunnen worden, zodat een specifieke update zo snel mogelijk gevonden kan worden.

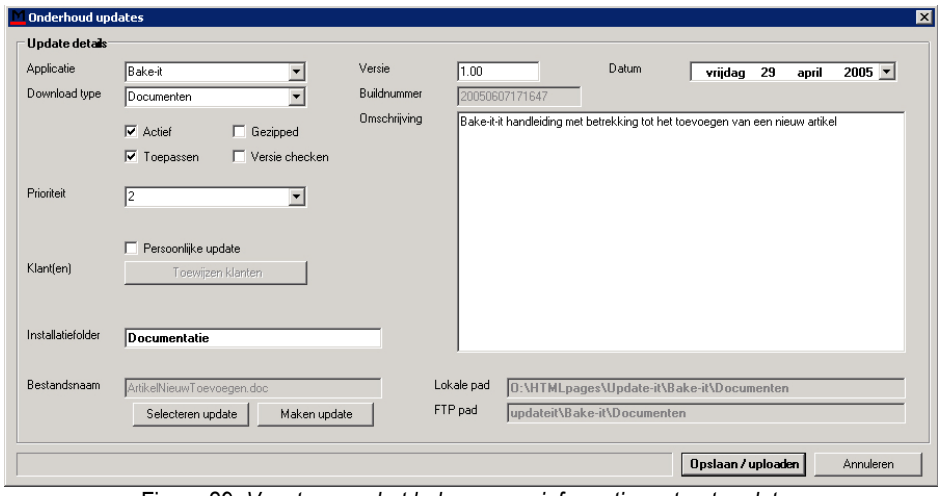
© 2005 Mathieu Brüning (20010019)



Figuur 28: *Venster voor het selecteren van updates*

De volgende stap was het maken van een venster, waarmee de detailgegevens van een update weergegeven, ingevoerd en gewijzigd zouden kunnen worden. Deze activiteit betrof voornamelijk het maken van een venster en het plaatsen van de corresponderende velden op het venster.

Nadat het venster voor het inzien, toevoegen en wijzigen van detailgegevens omtrent updates was ontwikkeld ben ik aan de slag gegaan met het koppelen van de reeds ontwikkelde functionaliteiten aan het venster. Het resultaat betrof de functionaliteit voor het grafisch beheren van de gegevens omtrent updates in de database en het grafisch beheren van de update bestanden op de Marti-IT server.



Figuur 29: *Venster voor het beheren van informatie omtrent updates*

Ontwikkelen functionaliteit voor het beheren van verzonden gebruikersgegevens
Een van de doelstellingen met betrekking tot het te ontwikkelen systeem richtte zich op het automatiseren van bepaalde processen binnen Marti-IT. Tijdsintensieve routinetaken als het downloaden en archiveren van verzonden gegevens door gebruiken zouden in het nieuwe systeem tot het verleden moeten behoren, zodat bepaalde processen binnen Marti-IT verbeterd zouden kunnen worden.
De ontwikkeling in de tweede pilot richtte zich zodoende vervolgens op het ontwikkelen van een functionaliteit voor het automatisch beheren van verzonden gebruikersgegevens. De eerste stap bij het ontwikkelen van deze functionaliteit richtte zich op het ontwerpen van een dergelijke mogelijkheid. Bij het ontwerpproces ben ik in eerste instantie gaan inventariseren wat de procedure is voor het verzenden, downloaden en archiveren in het huidige Marti-IT Update Centrum. De procedure is als volgt:

1. Een gebruiker verzend gegevens (een back-up, bestand of map) middels het Marti-IT Update Centrum, zodat het betreffende bestand geplaatst wordt op de Marti-IT server
2. Een medewerker van Marti-IT dient vervolgens handmatig een FTP-verbinding op te zetten met de Marti-IT server, om te controleren of er bestanden op de Marti-IT server zijn geplaatst door gebruikers
3. Indien er bestanden op de server zijn geplaatst kan de betreffende medewerker de bestanden handmatig downloaden naar een lokale server van Marti-IT, zodat deze gearchiveerd zijn.
4. De gearchiveerde bestanden kunnen dan vervolgens handmatig op de Marti-IT FTP-server worden verwijderd.
5. De gearchiveerde bestanden kunnen dan mogelijk nog gebruikt worden door de medewerkers van de helpdesk voor het bieden van ondersteuning

Op basis van het bovenstaande stappenplan ben ik aan de slag gegaan met het ontwerp van de functionaliteit. De gedachtegang hierachter was dat de mogelijkheid tot het automatisch beheren van verzonden gebruikersgegevens in zoverre geautomatiseerd diende te zijn, dat de medewerkers van Marti-IT niets meer hoeven te doen en dat de gegevens automatisch worden gearchiveerd op de lokale server.

Het ontwerp omvatte zodoende het volgende scenario:

1. Indien een gebruiker gegevens verzend naar Marti-IT middels het Marti-IT Update Centrum wordt er een record weggeschreven naar de database, waarin vermeldt wordt dat er een bestand wordt verzonden. Dit upload record krijgt de status “Uploaden”, wat betekent “bezig met uploaden”.
   Daarnaast wordt een zogenaamde ArchiveStatus bijgehouden, welke een indicatie geeft of een bepaald bestand al gearchiveerd is of niet. Indien een gebruiker een bestand verzend krijgt dit record de ArchiveStatus “ReadyForArchive”.

2. Wanneer het bestand volledig is verzonden naar de server krijgt het record de status “Geupload”, wat betekent “klaar met uploaden”

3. In het Marti-IT Build Centrum word een mechanisme geïmplementeerd welke op een te wijzigen interval (standaard is 10 minuten) controleert of er records aanwezig zijn met de status “Geupload” (bestand is volledig verzonden) en de ArchiveStatus “ReadyForArchive”. De records die aan deze criteria voldoen zijn bestanden op de Marti-IT FTP-server die nog niet zijn gearchiveerd.

4. De bestanden gekoppeld aan deze records worden vervolgens automatisch gedownload door de FTPManager klasse. Voordat de bestanden worden gedownload wordt in eerst instantie de ArchiveStatus gezet op “Busy” wat betekent “bezig met downloaden”. Op deze manier kan het nooit voorkomen dat er geprobeerd wordt om een bestand twee maal te downloaden.

5. Indien het downloaden van een bestand succesvol is afgerond wordt de ArchiveStatus gezet op “Done”. Op deze manier wordt er aangegeven dat een bestand succesvol is gedownload naar de Marti-IT lokale server. Het bestand wordt dan direct in de juiste map weggeschreven.

6. Indien het downloaden wordt onderbroken wordt de ArchiveStatus altijd teruggezet op “ReadyForArchive”, zodat het bestand opnieuw wordt gedownload in de volgende ronde.

Het implementeren van een dergelijke functionaliteit neemt met zich mee dat het Marti-IT Build Centrum continue opgestart moet zijn, om de gegevens automatisch te kunnen beheren. Het idee is dan ook om een instantie van het Marti-IT Build Centrum te installeren op een server, zodat de applicatie continue actief kan zijn voor het beheren van de verzonden gegevens.

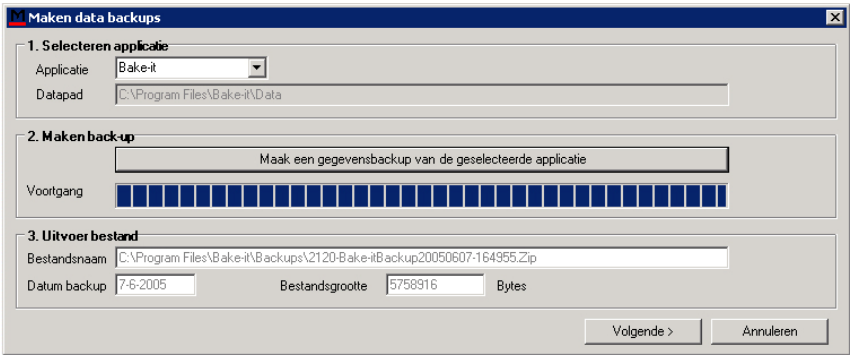
Op basis van dit scenario ben ik aan de slag gegaan met het programmeren. Het programmeren van de functionaliteit was naar mijn idee vrij eenvoudig, aangezien het ontwerp op een vrij gedetailleerd niveau was vastgelegd. Door het toevoegen van een Timer, het aanroepen van de classODBCDataManager klasse en het aanroepen van de FTPManager klasse, was de functionaliteit feitelijk gevormd.

Extra functionaliteiten met betrekking tot de gebruikers van de Marti-IT software

Nadat de functionaliteit voor het automatisch beheren van verzonden gebruikersgegevens was geïmplementeerd, ben ik aan de slag gegaan met het ontwikkelen van extra functionaliteiten voor de gebruikers van de Marti-IT software. Het uitbreiden van de functionaliteit van het systeem was een belangrijk aspect, aangezien er hoofdzakelijk al lange tijd behoefte was naar de mogelijkheid tot het maken van back-ups in het systeem.
Het huidige Marti-IT Update Centrum bevat namelijk niet de mogelijkheid tot het maken van back-ups, wat op sommige momenten onhandig kan zijn. Indien een gebruiker namelijk een actuele gegevensback-up wil versturen via het Marti-IT Update Centrum komt het nog wel eens voor dat een gebruiker in het Update Centrum is, maar dat er geen recente back-up is gemaakt. Op die manier moet de gebruiker de applicatie weer afsluiten om eerst een gegevensback-up te maken en vervolgens weer terug te keren naar het Update Centrum. Een nieuwe mogelijkheid tot het maken van back-ups in het Update Centrum moet hiermee afrekenen.

Het programmeren van de functionaliteit voor het maken van een back-up is gebaseerd op een bestaande functionaliteit voor het maken van back-ups. De eerste C# applicatie van Marti-IT, genaamd Cash-it, bevatte namelijk al een methode voor het maken van back-ups. Deze methode kon ik simpelweg overnemen. Zelf heb ik er echter nog een kleine wijziging in doorgevoerd, zodat het mogelijk werd om een back-up te maken voor een applicatie die een gebruiker zelf selecteert. Op basis van het softwareprofiel van een gebruiker worden de beschikbare applicaties getoond en is de gebruiker in staat om gegevensback-ups te maken voor een geselecteerde applicatie.

Er diende echter nog een venster ontwikkeld te worden, zodat de gebruiker de selectie kon maken en de mogelijkheid kreeg om de back-up te versturen. Het ontwikkelen van het venster is grotendeels gebaseerd op het venster voor het verzenden van bestanden of mappen. Door een aantal kleine aanpassingen op grafisch gebied en functioneel gebied, zoals het toevoegen van een mogelijkheid tot het selecteren van een applicatie en de bijbehorende functionaliteit, was het venster klaar.
Vervolgens heb ik de ontwikkelde functionaliteit aan het venster gekoppeld, zodat de functionaliteit voor het maken van gegevensback-ups vanuit het Marti-IT Update Centrum ook geïmplementeerd was.



Figuur 30: *Venster voor het maken van gegevensback-ups*

De tweede functionaliteit, bovenop de bestaande mogelijkheden van het Marti-IT Update Centrum, richtte zich op het downloaden van historische updates. Het downloaden van historische updates is een functionaliteit die feitelijk volledig gebaseerd is op de normale download functionaliteit. Het verschil is echter dat bij het downloaden van historische updates er alleen updates getoond worden, waarvan de waarde van het veld "Active" op onwaar staat. Het maken van de functionaliteit voor het downloaden van historische gegevens richtte zich dus feitelijk alleen op het kopiëren van het venster voor het downloaden van updates en het kopieren van de methoden voor het inventariseren van de updates, met als kleine verschil dat de waarde in het het veld Active onwaar diende te zijn bij het ophalen van update gegevens uit de database.

De laatste extra functionaliteit wat betreft het Marti-IT Update Centrum richtte zich op het ontwikkelen van een mogelijkheid tot het ophalen van updates vanaf een lokale en/of netwerklocatie. Aangezien de geplande tijdsduur van de tweede pilot, namelijk 3 weken, al bijna was verstreken (nog 1 dag) heb ik besloten om de ontwikkeling van deze functionaliteit niet uit te voeren. In het kader van het eerst ontwikkelen van de belangrijkste functionaliteiten heb ik besloten om de ontwikkeling in de tweede pilot te stoppen, zodat er nog enige ruimte was voor het uitvoeren van testwerkzaamheden. Met name doordat de eerste pilot een week was uitgelopen, aangezien het herontwikkelen van de bestaande functionaliteit de hoogste prioriteit had, heb ik besloten om de ontwikkelingen in de tweede pilot te stoppen en nog enige testwerkzaamheden op de ontwikkelde componenten uit te voeren.

Testen pilot 2

Aangezien er maar één dag was voor het testen van de ontwikkelde functionaliteiten uit de tweede pilot, heb ik de ontwikkelde functionaliteiten van de tweede pilot helaas niet zo grondig kunnen testen als de functionaliteiten van de eerste pilot. De uitgevoerde testwerkzaamheden richtte zich uiteindelijk op het testen van de volgende functionaliteiten:

- ✓ het synchroniseren van gebruikersgegevens met Serv-it
- ✓ het toevoegen, verwijderen en wijzigen van applicaties in de database
- ✓ het toevoegen, verwijderen en wijzigen van download types in de database
- ✓ het toevoegen, verwijderen en wijzigen van updates in de database
- ✓ het toevoegen, verwijderen en wijzigen van update bestanden op de Marti-IT server
- ✓ het automatisch downloaden en archiveren van verzonden gegevens

Vanwege de tijdsdruk heb ik een real-life test toegepast waarbij ik een situatie heb nagebootst zoals mogelijk zou kunnen zijn wanneer het systeem in operationeel gebruik is. Op basis van de functionele specificaties, zoals gedefinieerd in het pilotontwikkelplan van de tweede pilot, ben ik aan de slag gegaan met het uitvoeren van Black Box tests.

Bij het uitvoeren van de testwerkzaamheden heb ik maar een klein aantal fouten ontdekt in de ontwikkelde functionaliteiten. Deze fouten betroffen de volgende:

- ✓ gekoppelde gebruikers aan een persoonlijke update werden niet goed weergegeven bij het opnieuw openen van het detailvenster voor een update
- ✓ bij het automatisch downloaden en archiveren van verzonden gegevens werden er twee upload records geschreven naar de database, waardoor een foutmelding wordt gegeven wanneer er voor de tweede maal geprobeerd wordt om het bestand te downloaden

Naar mijn idee is het aantal fouten met betrekking tot de ontwikkelingen uit de tweede pilot gering. Dit is waarschijnlijk te danken aan het feit dat er tijdens het ontwikkelen van de functionaliteiten ook al aandacht besteedt is aan het opsporen van fouten.

Ondanks het feit dat er weinig tijd was voor het testen van de ontwikkelde functionaliteiten uit de tweede pilot ben ik toch van mening dat de uitgevoerde tests voldoende diepgang hadden. Op basis van de systeemeisen ben ik aan de slag gegaan met het uitvoeren van testwerkzaamheden, waarbij ik naar mijn idee weinig fouten ben tegengekomen.

De reden waarom ik heb besloten om de tweede pilot te stoppen en verder te gaan met de derde pilot in de pilotontwikkelingsfase was als volgt: ik was inmiddels al een week vertraging opgelopen, doordat de eerste pilot was uitgelopen. De doelstelling met betrekking tot de derde pilot richtte zich hoofdzakelijk op het verbeteren van de gebruikersvriendelijkheid van de applicatie. Deze verbetering in gebruikersvriendelijkheid was een fundamenteel aspect van het nieuwe systeem en zodoende heb ik besloten om met de derde pilot verder te gaan in de pilotontwikkelingsfase.

© 2005 Mathieu Brüning (20010019)

6.3.3 Pilot 3: Verbeteren van de gebruikersvriendelijkheid

De derde pilot ging van start toen de testwerkzaamheden met betrekking tot de tweede pilot waren afgerond. Het voornaamste doel van de ontwikkelingen in de derde pilot betrof het verbeteren van de gebruikersvriendelijkheid van de applicatie ten opzichte van het bestaande Marti-IT Update Centrum.

Opstellen pilotontwikkelplan
Om de structuur en de wijze van ontwikkelen vast te leggen richtte de eerste activiteit in de derde pilot zich op het opstellen van een pilotontwikkelplan. Het doel van deze activiteit was het definiëren van het verloop van de derde pilot, zodat er structuur in de ontwikkeling gebracht kon worden en duidelijk in kaart kon worden gebracht welke activiteiten er uitgevoerd diende te worden om de gewenste doelstellingen te behalen.

Op basis van de gegevens uit het pilotplan ben ik aan de slag gegaan met het opstellen van het pilotontwikkelplan. Hierbij heb ik in eerste instantie de globale functionele structuur, de globale technische structuur en de globale organisatorische structuur met betrekking tot de derde pilot beschreven. De beschrijvingen zijn voornamelijk gebaseerd op hetgeen wat gedefinieerd is in de definitiestudie en het pilotplan.

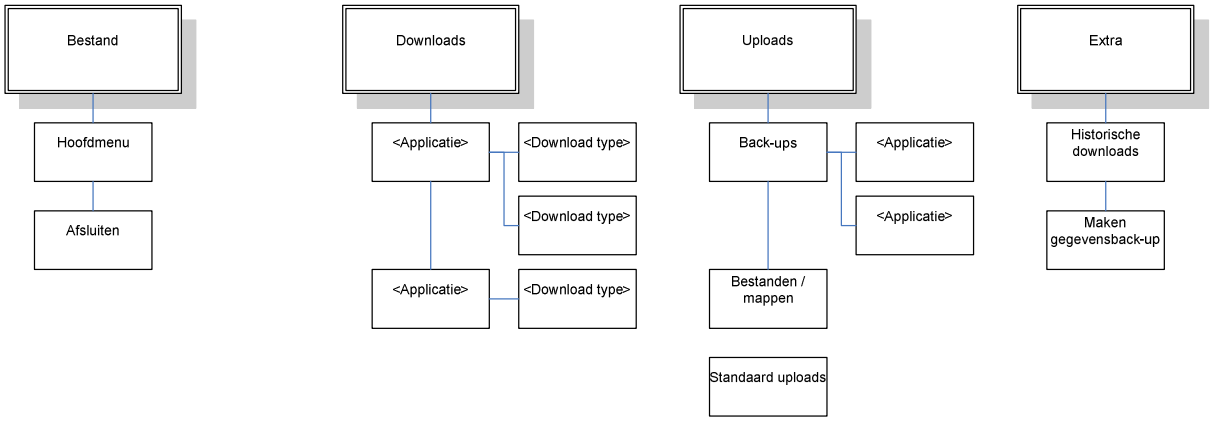
De tweede stap was het structureren van de pilot, op basis van de globale structuur uit het pilotplan. De globale structuur was namelijk al reeds gedefinieerd in het pilotplan, zodat ik op basis van deze globale structuur in staat was om de gedetailleerde structuur in kaart te brengen. Door namelijk in eerste instantie te doelstellingen met betrekking tot de clusters van de derde pilot te definiëren, was ik in staat om een overzicht te vormen omtrent de werkzaamheden om deze doelstellingen te behalen. Op die manier had ik de mogelijkheid om aan elke cluster in de derde pilot doelstellingen en werkzaamheden te koppelen, zodat de uitvoering en beoogde resultaten van de ontwikkeling in kaart kon worden gebracht.
Het resultaat betrof een verfijnd overzicht van de structuur van de derde pilot, zodat ik op basis van dat overzicht de werkzaamheden met betrekking tot de derde pilot zou kunnen uitvoeren.

De laatste stap, wat betreft het opstellen van het pilotontwikkelplan, was het opstellen van een test- en beoordelingsplan voor de derde pilot. Hierbij heb ik in eerste instantie de te volgen testprocedure vastgelegd, welke gebaseerd was op de gedefinieerde testprocedures in de definitiestudie. Na het opstellen van de testprocedure ben ik aan de slag gegaan met het opstellen van de testcriteria. Deze testcriteria zijn voornamelijk gebaseerd op de doelstellingen met betrekking tot de clusters in de ontwikkeling van de derde pilot.

Het pilotontwikkelplan van de tweede pilot is te vinden in bijlage 4.

Ontwikkelen dynamische menustructuur
Nadat het pilotontwikkelplan was opgesteld ben ik aan de slag gegaan met de ontwikkelwerkzaamheden in de laatste pilot. De eerste cluster in deze pilot richtte zich op het ontwikkelen van een dynamische menustructuur. Het doel van het ontwikkelen van een dynamische menustructuur richtte zich voornamelijk op het ontwikkelen van een menustructuur, welke is afgestemd op de gebruiker van het systeem. Het statische menu, zoals ontwikkeld in de eerste pilot, diende vervangen te worden door een dynamisch menu, zodat gebruikers niet de mogelijkheid kunnen krijgen om functionaliteiten aan te roepen via het menu die niet bestemd zijn voor een bepaalde gebruiker.
In de huidige situatie zijn er namelijk een viertal Update Centrums voor vier verschillende Marti-IT applicaties. Het nieuwe systeem diende echter de vier verscheidene applicatie te bundelen, zodat alle gebruikers van de Marti-IT software updates kunnen verkrijgen via een enkel systeem. Dit heeft als grote voordeel dat het onderhoud maar op één Update Centrum hoeft plaats te vinden, in plaats van vier. Echter, doordat gebruikers van verschillende Marti-IT applicaties in de toekomst gebruik zullen maken van een zelfde Update Centrum, dient er een functionaliteit geïmplementeerd te worden dat ervoor zorgt dat de gebruikers alleen de updates te zien krijgen die voor de betreffende gebruiker bestemd zijn. Het ontwikkelen van een dynamisch menu op basis van het softwareprofiel van de gebruiker zorgt ervoor dat de gebruiker alleen de functionaliteiten kan aanroepen die behoren tot het softwareprofiel van de betreffende gebruiker.
Een andere belangrijk reden voor het ontwikkelen van een dynamisch menu is met name dat de verschillende soorten updates, zoals vastgelegd in de database, ook dynamisch zijn. De zogenaamde download types kunnen middels het Marti-IT Build Centrum worden beheert en zodoende is het mogelijk om meerdere soorten download types beschikbaar te stellen voor de gebruikers. Middels een dynamisch menu kunnen de gebruikers dan gericht zoeken op een specifiek download type, ofwel op een specifieke soort update.

Het ontwikkelen van het dynamische menu richtte zich in eerste instantie op het ontwerpen van het dynamische menu. Bij het ontwerpen bleek al gauw dat niet alle aspecten van het menu dynamisch zouden kunnen worden, aangezien sommige functionaliteiten voor elke gebruiker van de Marti-IT software van toepassing zijn. De functionaliteiten die echter wel afhankelijk zijn van het softwareprofiel van een gebruiker, namelijk de beschikbare updates en beschikbare gegevensback-ups, zullen dynamisch worden ingeladen in het menu.



Figuur 31: *Ontwerp dynamisch menu*

Na het ontwerpen van het menu ben ik aan de slag gegaan het maken van een aantal controls, zodat het toevoegen en inrichten van een menu op een venster op eenvoudige wijze bewerkstelligd kon worden. Hiertoe heb ik een drietal klassen aangemaakt, welke overgeërfd zijn van de originele klassen voor het inrichten van een menu.

Nadat de controls waren gemaakt richtte mijn werkzaamheden zich op het inrichten van het menu. De inrichting van het menu vind plaats bij het opbouwen van het hoofdvenster, waarbij door middel van code het statische als dynamische gedeelte van het menu wordt opgebouwd. Op basis van de gegevens omtrent het softwareprofiel in de classIdentity klasse worden de dynamische aspecten van het menu opgebouwd en is de gebruiker alleen in staat om de functionaliteiten aan te roepen die behoren bij het softwareprofiel van deze gebruiker.

Ontwikkelen van een hoofdmenu

Een geheel nieuw aspect van het Marti-IT Update Centrum diende een hoofdmenu te zijn, van waaruit de gebruikers de belangrijkste functionaliteiten van het systeem kunnen oproepen. De doelstelling met betrekking tot het hoofdmenu richt zich voornamelijk op het tonen van een venster bij het opstarten van de applicatie, zodat gebruikers in staat zijn om op eenvoudige wijze direct de belangrijkste functionaliteiten aan te roepen, zonder dat er door een menu gebladerd dient te worden. Hierbij speelt de uitstraling van het hoofdmenu een belangrijke rol, aangezien het venster hoofdzakelijk altijd gebruikt dient te worden voor de besturing van de applicatie en dient bij te dragen aan de verbetering van de gebruikersvriendelijkheid van het systeem.

De eerste stap met betrekking tot het ontwikkelen van het hoofdmenu richtte zich op het ontwerpen van de indeling van het hoofdmenu. Zoals eerder genoemd diende alleen de belangrijkste functionaliteiten van het systeem vanuit het hoofdmenu aangeroepen te kunnen worden en zodoende ben ik aan de slag gegaan met het selecteren van belangrijke functionaliteiten om weer te geven in het hoofdmenu. Op basis van de analyse van het Update Centrum in de definitiestudie en ervaringen van de medewerkers van Marti-IT bleek dat de volgende functionaliteiten het belangrijkst zijn:

- ✓ Het downloaden van:
  - programma updates
  - rapporten updates
  - persoonlijke rapporten (maatwerk) updates
  - persoonlijke licenties
- ✓ Het uploaden van back-ups
- ✓ Het maken van gegevensback-ups

Op basis van deze informatie ben ik aan de slag gegaan met het ontwerpen van het hoofdmenu. Het ontwerp van het hoofdmenu resulteerde in de splitsing tussen de functionaliteiten voor het downloaden van gegevens en functionaliteiten voor het uploaden van gegevens. Aangezien het maken van een gegevensback-up sterk gerelateerd is aan het uploaden van back-ups heb ik besloten om de functionaliteit voor het maken van gegevensback-ups te plaatsen bij de upload sectie.
Het resultaat was dan als volgt:
Downloads:

- ✓ programma
- ✓ rapporten
- ✓ maatwerk
- ✓ licenties

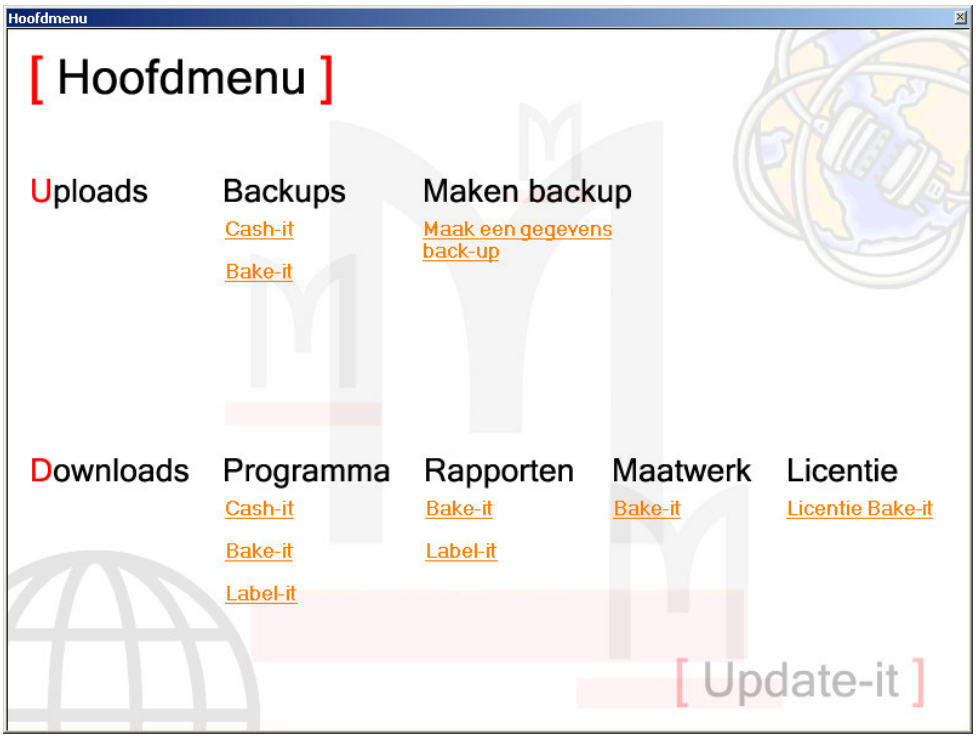
Uploads:

- ✓ maken back-up
- ✓ back-ups

Aan de hand van dit ontwerp ben ik aan de slag gegaan met het programmeren van de functionaliteit van het hoofdmenu. Hierbij heb ik gebruik gemaakt van zogenaamde linklabels, labels waarop een gebruiker kan klikken om een functionaliteit aan te roepen.
De opbouw van het hoofdmenu is dan, net als het menu aan de bovenkant van het venster, dynamisch opgebouwd, zodat een gebruiker alleen de functionaliteiten kan aanroepen die behoren bij het softwareprofiel van de betreffende gebruiker.

Om de uitstraling van het hoofdmenu te verbeteren ben ik aan de slag gegaan met het ontwikkelen van een achtergrond voor het hoofdmenu. Het ontwikkelen van de afbeelding voor de achtergrond heeft plaatsgevonden middels Adobe Photoshop. De gedachtegang achter de afbeelding richtte zich met name op het verwerken van relevante aspecten in de afbeelding, zoals symbolen waarmee de gebruikers de associatie kunnen leggen met het updaten van software en Marti-IT.
Op het World Wide Web heb ik een aantal afbeeldingen gezocht, welke ik mogelijk in het hoofdmenu zou kunnen verwerken. Na enige tijd gezocht te hebben ben ik een aantal afbeeldingen tegengekomen op Microsoft Office Online. Deze afbeeldingen heb ik vervolgens gebruikt in het ontwerp van het hoofdmenu.

De laatste stap met het betrekking tot de ontwikkeling van het hoofdmenu richtte zich op het integreren van het hoofdmenu in het reeds ontwikkelde hoofdvenster van het Marti-IT Update Centrum. Dit kon op eenvoudige wijze bewerkstelligd worden door het venster van het hoofdmenu simpelweg aan te roepen vanuit het hoofdvenster bij het opstarten van de applicatie.

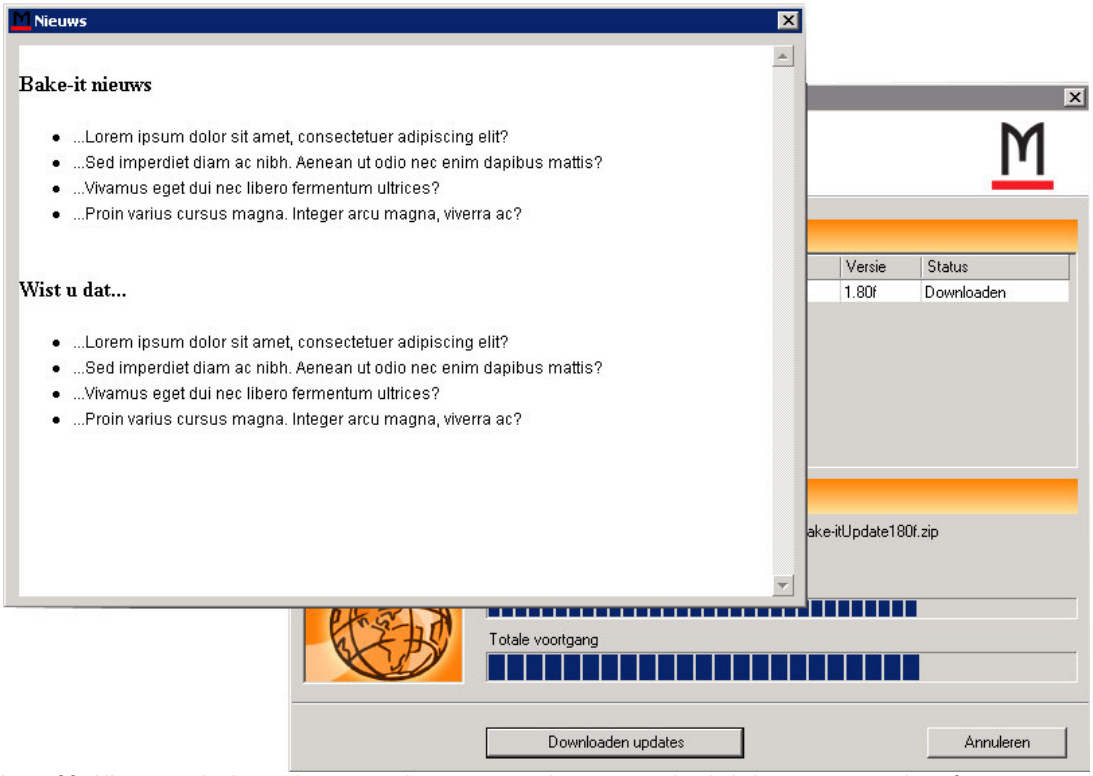


Figuur 32: *Hoofdmenu waarmee gebruikers in staat worden gesteld om de belangrijkste functionaliteiten van de applicatie te kunnen aanroepen*

Ontwikkelen van een functionaliteit voor het weergeven van nieuwspagina's
De derde cluster met betrekking tot de ontwikkeling in de derde pilot richtte zich op het ontwikkelen van een functionaliteit voor het weergeven van nieuwspagina's. Het weergeven van nieuwspagina's is een functionaliteit om informatie weer te kunnen geven, zodat de gebruikers van de Marti-IT software op de hoogte kunnen blijven van de nieuwtjes met betrekking tot Marti-IT en de Marti-IT software. Indien gebruikers de updates voor de Marti-IT software downloaden of back-ups verzenden worden deze nieuwspagina's bovenop het voortgangsvenster weergegeven. Zo is een gebruiker in staat om het laatste nieuws te lezen, terwijl het verzendproces plaatsvindt.
De functionaliteit voor het weergeven van de nieuwspagina's berust op het weergeven van pagina's in de HyperText Markup Language (HTML), aangezien de pagina's in het HTML formaat op eenvoudige wijze aan te passen zijn en goed overweg kunnen gaan met grafische opmaak, zoals afbeeldingen, tabellen, etc.

In eerste instantie ben ik aan de slag gegaan met het ontwerpen van de functionaliteit voor het weergeven van nieuwspagina's. De functionaliteit voor het weergeven van de nieuwspagina's diende te komen bestaan uit hoofdzakelijk twee componenten, namelijk een venster voor het weergeven van de nieuwspagina's en een bron waar vandaan de pagina's worden opgehaald en worden ingeladen in het venster.
Op basis van deze informatie ben ik in eerste instantie aan de slag gegaan met het maken van een venster voor het weergeven van nieuwspagina's. Aangezien de C# programmeertaal geen standaard control heeft voor het weergeven van HTML-pagina's in een C# Windows applicatie, was ik genoodzaakt om zelf een mogelijkheid te vinden voor het weergeven van de HTML-pagina's. Zodoende ben ik op het World Wide Web op zoek gegaan naar een functionaliteit voor het weergeven van HTML-pagina's in een standaard C# venster. Op de website van Tim Anderson's ITWriting heb ik vervolgens een open-source functionaliteit gevonden voor het weergeven van HTML-pagina's in een C# venster.

Nadat een venster voor de nieuwspagina's was gemaakt en de functionaliteit voor het weergeven van de HTML-pagina's was geïntegreerd in het venster ben ik aan de slag gegaan met het opzetten van de bron voor de nieuwspagina's. Op de Marti-IT server bevind zich een map genaamd "News", waarin de nieuwspagina's zijn opgeslagen. Indien een gebruiker het verzendproces start door bijvoorbeeld het downloaden van updates van de Marti-IT server wordt het venster met de nieuwspagina's weergegeven. De nieuwspagina's worden dan in eerste instantie gedownload van de server en worden geplaatst in een lokale map genaamd "News". Vervolgens worden de nieuwspagina's weergegeven en gaat het verzendproces van start.
De keuze van welke nieuwsbestand er wordt gedownload en wordt weergegeven is afhankelijk van het softwareprofiel van de gebruiker. Een gebruiker van de Bake-it krijgt bijvoorbeeld het Bake-it nieuws, etc.



Figuur 33: *Nieuwspagina's worden getoond wanneer een het systeem bezig is het met verzenden of ontvangen van gegevens*

Hanteren nieuwe kleurensamenstelling / verbeteren vensters
De laatste cluster in de ontwikkeling van de derde pilot richtte zich op het hanteren van een nieuwe kleurensamenstelling en het verbeteren van vensters. Deze kenmerken dienden namelijk eveneens bij te dragen aan de verbetering van de gebruikersvriendelijkheid van het nieuwe systeem ten opzichte van het oude systeem.

Door het hanteren van een nieuwe kleurensamenstellingen en het verbeteren van vensters dient het nieuwe Marti-IT Update Centrum namelijk duidelijker te zijn in gebruik, meer uitnodigend te zijn en meer te lijken op hedendaagse applicaties.

De eerste stap in deze cluster was het selecteren van een nieuwe kleurensamenstelling, zodat de huidige kleurensamenstelling (voornamelijk donkerblauw en grijs) veranderd worden in frisse, uitnodigende kleurensamenstelling. Om de herkenbaarheid van de applicatie te vergroten heb ik besloten om de standaard kleuren te hanteren van de Marti-IT applicatie waarover de betreffende gebruiker beschikt. Op deze manier neemt het Marti-IT Update Centrum dus als het ware de standaard kleur aan van de software waarover de gebruiker beschikt. Om dit te kunnen realiseren heb ik een klasse classColorProfile gemaakt. Deze klasse wordt geinitialiseerd wanneer een gebruiker zich autoriseert, zodat op basis van het softwareprofiel van de gebruiker bepaald kan worden welke kleurensamenstelling het systeem moet aannemen.

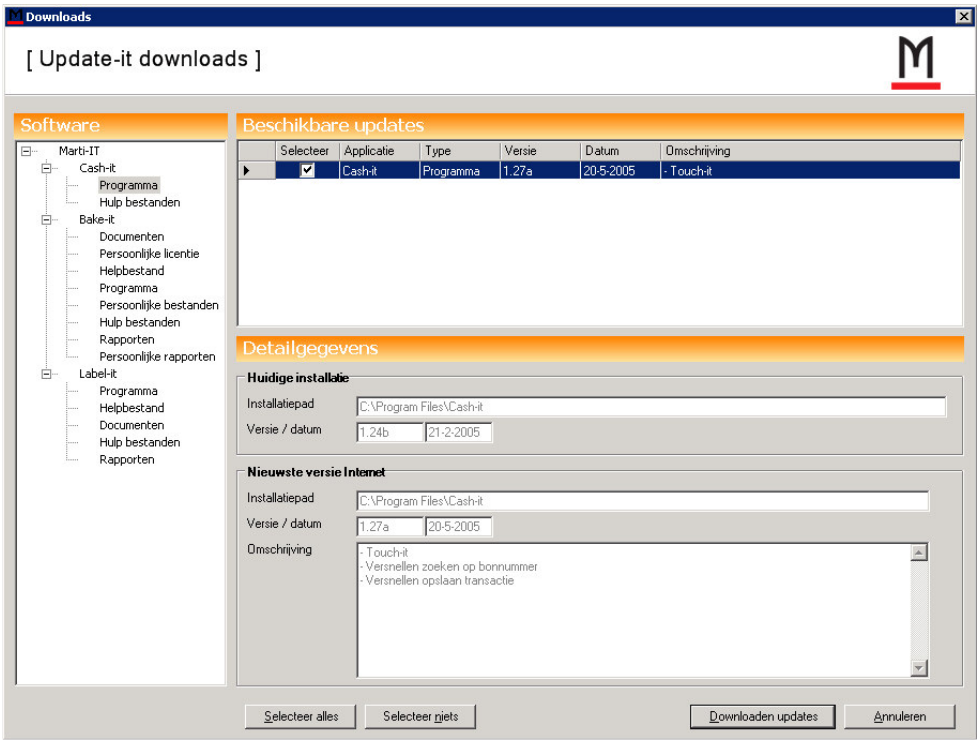
classColorProfile
-mInactiveGradientHighColor -mInactiveGradientLowColor
-CheckColorProfile()

Figuur 34: *classColorProfile*

Het hanteren van een nieuwe kleurensamenstelling heeft als resultaat dat de gebruiker enige vorm van herkenning heeft tijdens het gebruik van de applicatie, aangezien de kleuren worden afgestemd op de standaard kleuren van de betreffende Marti-IT software. Indien een venster wordt ingeladen wordt de classColorProfile benaderd om te kijken aan welke kleurprofiel het venster dient te voldoen. Het venster wordt dan door middel van een standaard methode aangepast, zodat het venster de juiste kleuren weergeeft.

Om het voortgangsvenster grafisch te verbeteren ben ik aan de slag gegaan met het ontwikkelen van een afbeelding, welke wordt getoond op het venster. Het ontwikkelen van deze afbeelding heeft plaatsgevonden middels Adobe Photoshop. Voor een afbeelding van het voortgangsvenster, inclusief de ontwikkelde afbeelding, verwijs ik u naar de volgende bladzijde.

De laatste stap in deze cluster richtte zich op het verbeteren van de reeds ontwikkelde vensters van de applicatie. De vensters, zoals ontwikkeld in de eerste pilot, waren functioneel in orde, maar op het gebied van grafische weergave konden ze nog verbeterd worden. De vensters waren namelijk feitelijk niets meer dan grijze vensters met standaard controls erop. Dit was ongeveer te vergelijken met de uitstraling van de vensters van het huidige Marti-IT Update Centrum, waarvan juist bepaald was het uiterlijk meer uitnodigend diende te worden.
Op het World Wide Web ben ik op zoek gegaan naar voorbeelden en controls om te downloaden ten behoeve van een verbetering qua grafische weergave. Op de website www.windowsforms.com heb ik een tal aan controls en voorbeeldapplicaties kunnen downloaden om mogelijk te kunnen gebruiken bij de verbetering van de vensters van het systeem. De open-source componenten op de website zijn allemaal gratis te downloaden en zodoende heb ik een component gedownload op te gebruiken op mijn vensters. Dit component betrof een zogenaamd *Pane* ofwel een soort van titelbalk met een overlopende kleur. Dit component heb ik aangebracht op bijna alle vensters, om de secties van het venster aan te geven. Op die manier is het voor de gebruiker duidelijker welke informatie waar wordt, aangezien er een titelbalk met een titel boven staat.
Daarnaast heb ik de vensters voorzien van een witte rand aan de bovenkant van het venster met een Marti-IT logo erin verwerkt.



Figuur 35: *Voorbeelden van een aangepaste vensters, waarin een Marti-IT logo en de nieuwe kleurensamenstelling in de titelbalken zijn verwerkt*

Testen pilot 3
De ontwikkelingen in de derde pilot richtte zich hoofdzakelijk op het verbeteren van de gebruikersvriendelijkheid van het reeds ontwikkelde Marti-IT Update Centrum. De ontwikkelingen uit de derde bestaan uit de volgende clusters:

- ✓ het ontwikkelen van een dynamische menustructuur
- ✓ het ontwikkelen van een hoofdmenu
- ✓ het ontwikkelen van een functionaliteit voor het weergeven van nieuwspagina's
- ✓ het hanteren van een nieuwe kleurensamenstelling / verbeteren vensters

Op basis van de functionele eisen uit het pilotontwikkelplan van de pilot ben ik aan de slag gegaan met het opstellen van een testplan. Dit testplan richtte zich met name op de ontwikkelde functionaliteiten van de derde pilot, zoals het functionaliteit achter het dynamische menu, de functionaliteit achter het hoofdmenu en de functionaliteit voor het weergeven van de nieuwspagina's.
Het vaststellen of het hanteren van een nieuwe kleurensamenstelling een verbetering is naar mijn mening een subjectief aspect. Waar de ene gebruiker de vernieuwde kleurensamenstelling vind bijdragen aan een verbetering van de gebruikersvriendelijkheid kan het zijn dat een andere gebruiker juist vind dat het juist een verslechtering van de gebruikersvriendelijkheid teweeg brengt. Zodoende heb ik besloten om dit niet mee te nemen in het testplan. Op basis van de acceptatietests in de invoering / nazorgfase van het project zal ik de testuitvoerders vragen wat zij vinden van de aanpassingen met betrekking tot de interface.

Nadat het testplan was opgesteld ben ik aan de slag gegaan met het uitvoeren van de testwerkzaamheden, zodat ik in staat was om vast te stellen in hoeverre de ontwikkelde functionaliteiten van de derde pilot in overeenstemming waren met hetgeen wat vooraf gedefinieerd was.

De resultaten van de testwerkzaamheden betroffen met name dat de ontwikkelingen voldeden aan hetgeen wat er gedefinieerd was in het pilotontwikkelplan. Hierbij heb ik geen fouten ontdekt. Hoogstwaarschijnlijk was dit te wijten aan het tussentijds testen van ontwikkelde functionaliteiten, tijdens het ontwikkelen ervan. Daarnaast speelde het waarschijnlijk ook mee dat de functionaliteiten van de ontwikkelingen uit de derde pilot naar mijn idee niet zo heel erg complex waren.

© 2005 Mathieu Brüning (20010019)

14.4 Invoering / nazorgfase

Nadat de derde en laatste pilot was afgerond betekende dit het einde van de pilotontwikkelingsfase. De volgende fase in het project betrof de invoering / nazorgfase, waarin het maken van handleidingen, het uitvoeren van de acceptatietests en het overdragen van de documentatie omtrent het systeem centraal stond.

Opstellen handleidingen Marti-IT Update / Build Centrum
De eerste activiteit in de invoering / nazorg fase richtte zich op het maken van handleiding voor zowel het Marti-IT Update Centrum als het Marti-IT Build Centrum. De doelstelling met betrekking tot het maken van de handleidingen richtte zich hoofdzakelijk op het maken van een document, waarin informatie omtrent de werking van de beide applicaties wordt vermeldt. Indien er dan een onduidelijkheid mocht voorkomen tijdens het gebruik van één de applicaties, kan men altijd in eerste instantie de handleiding gebruiken als referentiemateriaal.
Om niet te veel tijd kwijt te raken aan het beschrijven van alle mogelijke aspecten van de applicaties, heb ik besloten om handleidingen te maken in de vorm van reference cards. Een reference card is een beknopte vorm van een handleiding, waarin alleen de belangrijkste kenmerken worden weergegeven. Op die manier hoeft een gebruiker niet een heel document door te worstelen om informatie omtrent de werking van een systeem te achterhalen, maar kan de gebruiker simpelweg de reference card ter hand nemen en snel opzoeken hoe een bepaalde functionaliteit werkt.

De activiteiten met betrekking tot het opstellen van een handleiding voor het Marti-IT Update / Build Centrum resulteerde zodoende in een tweetal reference cards, waarop de werking van de beide applicaties beknopt, maar volledig wordt beschreven. De reference cards zijn te vinden in bijlage 5.

Acceptatietest gebruikers Marti-IT software
Het opstellen van de handleidingen voor het Marti-IT Update / Build Centrum was een belangrijke eerst stap in de invoering / nazorgfase, aangezien de handleidingen gebruikt zouden gaan worden voor het uitvoeren van de acceptatietests.

De eerste acceptatietest richtte zich op de gebruikers van de Marti-IT software, waarbij ik een vijftal klanten heb geselecteerd om de acceptatietest mee uit te voeren. De selectie van de klanten was met name gebaseerd op klanten met een affiniteit voor computers (zodat er niet teveel tijd verloren gaat met het inrichten van de testomgeving bij de klant) en gebaseerd op de verwachte mate van het verlenen van medewerking. Naar aanleiding van informatie van medewerkers van Marti-IT omtrent de mogelijke gebruikers die wel mee wilden doen aan de acceptatietest was ik in staat om een lijst op de te stellen met klantnamen.

Om de acceptatietest met de gebruikers uit te voeren heb ik in eerste instantie een testrapport heb gemaakt. Het maken van het testrapport had als voornaamste doelstelling om de testprocedure bij de gebruikers vast te leggen, door testscenario's en logische testgevallen te specificeren. Op die manier konden de gebruikers volgens een voor gedefinieerde procedure structureel de applicatie testen en hun bevindingen rapporteren.

Nadat het testrapport was gemaakt ben ik contact gaan leggen met de gebruikers op basis van de lijst met klantnamen. Één voor één heb ik de klanten afgebeld en heb ik middels telefonische ondersteuning het nieuwe Marti-IT Update Centrum geïnstalleerd. Vervolgens heb ik de gebruikers het testrapport laten downloaden, zodat zij de bevindingen tijdens het uitvoeren van de testwerkzaamheden terug konden rapporteren. Naast het testrapport heb ik de gebruikers eveneens een reference card laten downloaden, zodat er in het geval van onduidelijkheden terug gevallen kon worden op deze documentatie.

Helaas hebben niet alle geselecteerde klanten gehoor gegeven aan mijn verzoek tot het uitvoeren van de testwerkzaamheden. Één van de gevraagde klanten zag namelijk niet de mogelijkheid en ruimte om mee te kunnen doen aan de acceptatietest. Echter, een viertal klanten hebben dit wel gedaan door hun bevindingen terug te sturen. Op basis van deze bevindingen was ik in staat om een overzicht te maken van de huidige status van de applicatie.

De resultaten van de acceptatietest met de gebruikers van de Marti-IT software bevinden zich in bijlage 6.

© 2005 Mathieu Brüning (20010019)

Acceptatietest medewerkers Marti-IT
Nadat de acceptatietest met betrekking tot de gebruikers van de Marti-IT software was uitgevoerd en de resultaten waren verwerkt ben ik aan de slag gegaan met het opstellen van een testrapport voor de andere acceptatietest. De andere acceptatietest richtte zich op de medewerkers van Marti-IT, waarbij het Marti-IT Build Centrum aan de test onderworpen zou worden. Het doel van deze activiteit was het helder maken van de mate waarin het systeem gereed is voor ingebruikname.

Voor het uitvoeren van de acceptatietest met betrekking tot het Marti-IT Build Centrum heb ik een medewerker geselecteerd voor het uitvoeren van de testwerkzaamheden. Deze selectie was met name gebaseerd op de mate waarmee de medewerker mogelijk in de toekomst met het Marti-IT Build Centrum zou gaan werken.

Net als bij de acceptatietest met betrekking tot de gebruikers van de Marti-IT software, heb ik de tester voorzien van een testrapport en een reference card. De resultaten van de testwerkzaamheden zouden dan in het testrapport vermeldt kunnen worden en indien er onduidelijkheden zouden optreden zou er teruggevallen kunnen worden op de reference card.

Op basis van de testprocedure uit het testrapport kon de medewerker zelfstandig aan de slag gaan met het uitvoeren van de testwerkzaamheden. Door de drukte waar Marti-IT op dit moment mee geconfronteerd wordt, heeft de tester echter de testwerkzaamheden moeten afkappen. Het was gewoon niet mogelijk om naast zijn reguliere werkzaamheden ook nog eens een acceptatietest uit te voeren en de resultaten terug te koppelen. De extreme drukte, welke Marti-IT op dit moment ondervind, zorgde er zodoende voor dat de testwerkzaamheden afgekapt diende te worden. Er was helaas ook geen mogelijkheid tot het uitvoeren van een acceptatietest met een andere medewerker, omdat alle medewerkers het er geen tijd voor vrij konden maken.

Afronden applicatie documentatie
Voordat de applicatie documentatie aan de opdrachtgever overgedragen kon worden diende er nog een aantal zaken beschreven te worden. In het verloop van het project waren er namelijk diverse ontwerpen de revue gepasseerd, maar dit waren allemaal individuele ontwerpen met betrekking tot specifiek aspecten van de applicaties. Het doel met betrekking tot het afronden van de applicatie documentatie richtte zich hoofdzakelijk op het samenvoegen van de ontwerpen en in kaart brengen van de systemen op plaatsen waar het nog niet in kaart gebracht was.

Het resultaat van het afronden van de applicatie documentatie betrof een tweetal documenten waarin de informatie staat omtrent de specificaties van het Marti-IT Update Centrum en het Marti-IT Build Centrum. In deze documenten komen de volgende zaken aan bod:

- ✓ Functioneel ontwerp, waarin de volgende onderdelen aan bod komen:
  - Te registreren gegevens
  - Actoren
  - Procesmodel, bestaande uit
    - Functionele specificaties
    - Use-cases
    - Use-case diagrammen
  - Gegevensmodel

- ✓ Technisch ontwerp, waarin de volgende onderdelen aan bod komen:
  - Globale structuur
  - Gedetailleerde structuur, in de vorm van klassendiagrammen
  - Declaraties van de klassen

Overdragen applicatie documentatie
Nadat de applicatie documentatie besproken is met de opdrachtgever en hij daarvoor zijn goedkeuring had gegeven kon de documentatie overgedragen worden. Het overdragen van de applicatie documentatie richtte zich feitelijk op het plaatsen van de ontwikkelde applicatiedocumentatie op een centrale plaats voor documentatie op het netwerk.

Overdragen applicatie
Het overdragen van de ontwikkelingen tijdens het project richtte zich eveneens op het plaatsen van de bestanden op een centrale plaats binnen het Marti-IT netwerk. Een speciale server binnen Marti-IT genaamd “Develop”, wordt gebruikt om de voor het delen, bewaren en maken van back-ups van bron-code. Het overdragen van de ontwikkelingen uit het project richtte zich zodoende op het plaatsen van de bestanden op de “Develop” server.

© 2005 Mathieu Brüning (20010019)

Schrijven scriptie

Na het afronden en overdragen van de applicatie documentatie en het overdragen van de applicatie ben ik aan de slag gegaan met het schrijven van de afstudeerscriptie. Het schrijven van de scriptie richtte zich in eerste instantie op het beschrijven van de organisatie, waarin het afstudeertraject wordt volbracht. Aan de hand van een globale en een specifieke beschrijving wordt de organisatie in kaart gebracht. De volgende stap bij het schrijven van de scriptie richtte zich op het vastleggen van alle aspecten rondom de uit te voeren opdracht, zoals ondermeer de aanleiding, de probleemstelling en de doelstellingen. Nadat alle kenmerken met betrekking tot de uit te voeren opdracht waren vastgelegd ben ik aan de slag gegaan met het beschrijven van deel 2 van de scriptie. Deel twee richt zich hoofdzakelijk op de uitgevoerde de werkzaamheden tijdens het project, waarbij het eerste hoofdstuk zich richt op de gehanteerde aanpak. Bij het beschrijven van de werkzaamheden heb ik een concrete indeling gemaakt op basis van de betreffende fase in het project, zodat er een duidelijke structuur in de beschrijving aangebracht zou zijn. Nadat de werkzaamheden beschreven waren, ben ik aan de slag gegaan met het formuleren van het hoofdstuk met de analyse van de behaalde resultaten. Toen de resultaten van het project eenmaal beschreven waren heb ik een overzicht geformuleerd van de opgeleverde producten tijdens project. Hierbij wordt voor elk product een beschrijving weergegeven. Het laatste hoofdstuk in het tweede deel van de scriptie richt zich op de evaluaties. Op basis van de doelstellingen van het project en de opgeleverde producten ben ik de productevaluatie gaan beschrijven. Het beschrijven van de procesevaluatie heeft voornamelijk plaatsgevonden op basis van de werkwijzebeschrijving en de gehanteerde aanpak. Nadat de evaluaties waren beschreven was de scriptie feitelijk afgerond. Mijn laatste activiteiten omtrent de scriptie richtten zich op het opstellen van een literatuurlijst, lijst van figuren, een verklarende woordenlijst en het bijvoegen van de bijlagen.

© 2005 Mathieu Brüning (20010019)

7 Analyse van de resultaten

De resultaten als gevolg van de ontwikkeling van het Marti-IT Update Centrum en het Marti-IT Build Centrum worden in dit hoofdstuk uiteengezet. Het hoofdstuk is hierbij ingedeeld in paragrafen, welke beschrijvingen bevatten omtrent de verscheidene behaalde resultaten.

7.1 Uitbreiding van de functionaliteit

Het uitbreiden van de functionaliteit van het Marti-IT Update Centrum was een belangrijke doelstelling, aangezien er is gebleken dat de huidige applicatie niet meer volledig de behoeftes dekt van de klanten. In zowel de eerste, tweede als derde pilot zijn er bovenop de bestaande functionaliteit extra functionaliteiten ontwikkeld, welke het gebruik van de applicatie in de toekomst nog meer moeten vergemakkelijken. Onderstaand wordt een opsomming gegeven van de uitbreidingen van het nieuwe systeem ten opzichte van het oude Marti-IT Update Centrum:

- ✓ Mogelijkheid tot het inventariseren van alle beschikbare updates voor een gebruiker
- ✓ Mogelijkheid tot het downloaden en installeren van meerdere updates tegelijkertijd
- ✓ Mogelijkheid tot het uploaden van meerdere back-ups tegelijkertijd
- ✓ Mogelijkheid tot het maken van gegevensback-ups
- ✓ Mogelijkheid tot het downloaden van historische updates
- ✓ Mogelijkheid tot het eenvoudig uitbreiden van standaard uploads door middel van xml-bestanden
- ✓ Mogelijkheid tot het aansturen van de belangrijkste functionaliteiten door middel van een hoofdmenu
- ✓ Mogelijkheid tot het verkrijgen van informatie door middel van weergegeven nieuwspagina's

7.2 Verbetering op het gebied van onderhoudbaarheid

Naast het feit dat er uitbreidingen hebben plaatsgevonden op het gebied van de functionaliteit is een belangrijk aspect van het nieuwe Marti-IT Update Centrum de onderhoudbaarheid. Waar in het verleden een viertal individuele Update Centrum onderhouden diende te worden, kan nu een enkele applicatie onderhouden worden. Het nieuwe Marti-IT Update Centrum is specifiek ontwikkeld voor alle gebruikers van de Marti-IT software, zodat er geen sprake meer is van het onderhoud aan de individuele systemen.

7.3 Verbetering van processen binnen Marti-IT

De ontwikkelingen van de tweede pilot hebben zich met name gericht op de interne processen binnen Marti-IT. Door namelijk functionaliteiten ontwikkeld te hebben voor het beheren van de database van het Update Centrum is het niet meer nodig om handmatig informatie op de server te wijzigen. De updates kunnen middels één applicatie, genaamd het Marti-IT Build Centrum, onderhouden worden, door middel van functionaliteiten voor het toevoegen, verwijderen en wijzigen van updates op de Marti-IT server. Naast deze functionaliteit voor het beheren van de updates is er tevens een functionaliteit geïmplementeerd voor het beheren van de door gebruikers verzonden gegevens.

De resultaten met betrekking tot het Marti-IT Build Centrum richtten zich voornamelijk op een verbetering van de interne processen, aangezien een aantal van de taken die normaliter handmatig werden uitgevoerd middels het nieuwe systeem geautomatiseerd zijn.

De geautomatiseerde taken zijn:

- ✓ Het inpakken van updatebestanden
- ✓ Het uploaden/klaarzetten van updatebestanden op de Marti-IT server
- ✓ Het downloaden en archiveren van door gebruikers verzonden gegevens

© 2005 Mathieu Brüning (20010019)

7.4 Verbetering van de gebruikersvriendelijkheid

Een fundamentele doelstelling aan het nieuwe Marti-IT Update Centrum betreft een verbetering van de gebruikersvriendelijkheid. De aansturing van de oude applicatie was voor sommige gebruikers niet direct duidelijk en intuïtief en zodoende dient het systeem qua gebruikersvriendelijkheid verbeterd te worden.

De voornaamste verbetering met betrekking tot de gebruikersvriendelijkheid is de implementatie van het hoofdmenu. Het hoofdmenu is een venster dat direct getoond wordt bij het opstarten van het systeem. Middels dit hoofdmenu wordt de gebruiker in staat gesteld om de belangrijkste functionaliteiten van de applicatie aan te roepen, zodat het bijna niet meer nodig is om door een menu heen te bladeren. De voornaamste verbetering in de gebruikersvriendelijkheid zit hem met name in de directe confrontatie van de belangrijkste functionaliteiten van het systeem met de gebruiker, aangezien bij het opstarten direct het venster getoond wordt.

Naast de implementatie van het hoofdmenu is de samenbundeling van de individuele Update Centrums tot één applicatie eveneens een verbetering in de gebruikersvriendelijkheid. Bij de gebruikers van de Bake-it software is er namelijk regelmatig sprake van het gebruik van meerdere Marti-IT applicaties. Het nieuwe systeem zorgt ervoor dat gebruikers die beschikken over meerdere Marti-IT applicaties updates voor deze applicaties kunnen downloaden in één applicatie. Dit is naar mijn mening ook een grote verbetering van de gebruikersvriendelijkheid.

7.5 Opgedane ervaring m.b.t. het .NET framework en de C# programmeertaal

De ontwikkeling van het Marti-IT Update Centrum en het Marti-IT Build Centrum hebben beide plaatsgevonden middels de .NET programmeertaal C#. Voordat ik dit afstudeerproject ben begonnen had ik mezelf al enigszins georiënteerd op de C# taal, door middel van het uitvoeren van kleine testjes. Nu ik echter een afstudeertraject van 18 weken heb doorlopen, met een ontwikkelfase van ongeveer 12 weken kan ik stellen dat ik enige ervaring heb opgedaan wat betreft de C# taal. Met name door de hulp en informatie van één van de ontwikkelaars binnen Marti-IT was ik in staat om de taal beter te leren kennen.

De opgedane ervaring met betrekking tot het .NET framework en de C# programmeertaal heb ik ervaren als zeer prettig. Ik ben onder andere van mening dat ik met deze kennis waardevoller ben geworden als informaticus, aangezien .NET ontwikkelaars op dit moment goed in de markt liggen. Daarnaast heb ik het ontwikkelen/programmeren middels de C# programmeertaal ervaren als zeer prettig. Dit was wellicht te wijten aan het feit dat ik al ervaring had met de Java programmeertaal, welke qua syntax in vele opzichten lijkt op de C# syntax. Doordat de documentatie omtrent C# op de Microsoft Software Developer Network ook erg duidelijk, overzichtelijk en uitgebreid is, was het voor mij gemakkelijk om informatie te vinden in het geval van onduidelijkheden. Een laatste groot voordeel van C# is de zeer krachtige en eindeloze IDE, waarin het ontwikkelen van zowel vensters als code zeer goed en uitgebreid in is geïmplementeerd.

Het resultaat met betrekking tot het ontwikkelen van het Marti-IT Update Centrum en het Marti-IT Build Centrum middels de C# taal betreft dus met name een vergroting van de kennis en opgedane ervaring. Echter, het ontwikkelen middels de C# taal heeft er ook voor gezorgd dat het bestaande Marti-IT Update Centrum is herschreven in de C# taal, zodat de applicatie weer helemaal up-to-date is op het gebied van technologie.

© 2005 Mathieu Brüning (20010019)

8 Opgeleverde producten

Dit hoofdstuk bevat beschrijvingen omtrent de opgeleverde producten tijdens het afstudeertraject. De paragrafen in dit hoofdstuk representeren de producten.

8.1 Rapport met de specificaties van het Marti-IT Update / Build Centrum

Het rapport met de specificaties van het Marti-IT Update / Build Centrum is het resultaat van de definitiestudie van het project en bevat de specificaties welke gebruikt zijn als input voor het te ontwikkelen systeem. Op basis van het rapport met de specificaties is ondermeer het pilotplan opgesteld en de globale structuur van de ontwikkelingsfase vastgesteld.

Het rapport met de specificaties van het Marti-IT Update / Build Centrum omvat de volgende aspecten:

- ✓ een definiëring van het ontwikkelscenario, waarin de volgende aspecten worden vastgelegd:
  - de kenmerken rondom het project
  - de impact van het project
  - de cruciale succesfactoren
  - het ontwikkelproces
  - de projectorganisatie
  - de globale pilotstrategie
  - de globale strategie rondom hergebruik en externe-acquisitie
  - de globale teststrategie
  - de beheersstrategie

- ✓ een analyse van het bestaande Marti-IT Update Centrum, waarin de volgende aspecten worden vastgelegd:
  - de historie van de applicatie
  - een inventarisatie van de functionaliteiten
  - een use-case diagram met betrekking tot de functionaliteiten
  - een SWOT-analyse

- ✓ analyses van Update Centrums van andere softwarepakketten, namelijk:
  - Microsoft Windows Update
  - Norton AntiVirus Professional
  - Cyberlink PowerDVD

- ✓ een definiëring van de systeemeisen aan het te ontwikkelen systeem, onderverdeeld in de volgende aspecten:
  - basis systeemeisen
  - interface systeemeisen
  - integriteitssysteemeisen
  - operationele systeemeisen
  - een consolidatie van de systeemeisen
  - acceptatiecriteria

- ✓ een definiëring van het systeemconcept, onderverdeeld in de volgende aspecten:
  - specificatie van de actoren
  - specificatie van de events
  - een globaal procesmodel
  - een globaal gegevensmodel
  - specificatie van de systeem-interfaces

- ✓ een definiëring van de technische structuur, onderverdeeld in de volgende aspecten:
  - de technische architectuur
  - de technologische basis
  - de aanschaflijst

© 2005 Mathieu Brüning (20010019)

- ✓ een definiëring van de organisatorische inrichting, onderverdeeld in de volgende aspecten:
  - de werkstroom
  - de organisatorische inrichting
  - de doelstellingen met betrekking tot opleidingen
  - de vereiste gebruikersdocumentatie

Het rapport met de specificaties van het Marti-IT Update / Build Centrum is te vinden in bijlage 3.

8.2 Handleidingen t.b.v. het gebruik van Marti-IT Update / Build Centrum

De handleidingen ten behoeve van het gebruik van het Marti-IT Update / Build Centrum bestaat uit een tweetal reference cards, waarop de toekomstige gebruikers van het Marti-IT Update / Build Centrum informatie kunnen vinden omtrent de werking van de applicaties. Zowel de bestaande functionaliteiten van het oude Marti-IT Update Centrum, als de nieuwe functionaliteiten van het systeem worden uiteengezet, zodat een gebruiker in het geval van onduidelijkheden één van de handleiding kan raadplegen voor informatie.

De reference cards zijn de vinden in bijlage 5.

8.3 Testrapporten acceptatietest gebruikers Marti-IT software

Het testrapport met betrekking tot de acceptatietests met de gebruikers van de Marti-IT software bevat de resultaten van de uitgevoerde acceptatietests. De acceptatietest wat betreft de gebruikers van de Marti-IT software betrof een viertal acceptatietests, waarbij een viertal gebruikers van de Marti-IT software de testwerkzaamheden hebben uitgevoerd en de resultaten hebben teruggekoppeld. Aan de hand van vooraf gedefinieerde testscenario's en logische testgevallen konden de gebruikers het systeem testen en hun bevindingen rapporteren. Het resultaat van de acceptatietests betrof een inventarisatie van de testresultaten, gebundeld in een testrapport met betrekking tot de acceptatietest met de gebruikers van de Marti-IT software.

In het testrapport komen de volgende aspecten aan bod:

- ✓ Testomgeving
- ✓ Testscenario's
- ✓ Test sets
- ✓ Testresultaten

Bij volledige testrapport is te vinden in bijlage 6.

8.4 Marti-IT Update / Build Centrum

Het beoogde resultaat van het project was het opleveren van een tweetal applicaties, waarbij één applicatie bestemd was voor de gebruikers van de Marti-IT software en de andere applicatie voor de medewerkers van Marti-IT.

Het nieuwe Marti-IT Update Centrum (bestemd voor de gebruikers van de Marti-IT software) is een applicatie gebaseerd op het oude Marti-IT Update Centrum, welke herschreven in de .NET programmeertaal C#. Alle functionaliteiten van het bestaande systeem zijn in het nieuwe systeem verwerkt, inclusief een aantal nieuwe functionaliteiten als ondermeer het inventariseren van alle beschikbare updates, het downloaden van meerdere updates tegelijkertijd, het maken van gegevensback-ups en het downloaden van historische updates. Daarnaast is de applicatie qua gebruikersvriendelijkheid flink verbeterd door onder andere het implementeren van een hoofdmenu en een dynamische menustructuur.

Het Marti-IT Build Centrum (bestemd voor de medewerkers van Marti-IT) is een applicatie welke als voornaamste doel heeft zorg te dragen voor het beheer van de gegevens op het Marti-IT Update Centrum. Door middel van functionaliteiten voor het onderhouden van de gegevens in de database en mogelijkheden tot het onderhouden van de bestanden op de Marti-IT server, worden de medewerkers van Marti-IT in staat gesteld om de beschikbare gegevens op het Marti-IT Update Centrum te beheren. Daarnaast is er een functionaliteit voor het automatisch downloaden en archiveren van verzonden bestanden door de gebruikers, zodat deze tijdsintensieve routinetaken niet meer uitgevoerd hoeven te worden door de medewerkers van Marti-IT.

© 2005 Mathieu Brüning (20010019)

9 Evaluaties

De evaluaties op zowel proces- als productmatig niveau staan centraal in dit hoofdstuk. De eerste paragraaf richt zich op de evaluatie op procesmatig niveau, waarop de daarop volgende paragraaf zich richt op de evaluatie op productmatig niveau.

9.1 Proces

Om het afstudeertraject zo goed mogelijk te laten laten verlopen heb ik allereerst bewust gekozen voor een gestructureerde aanpak, waarin duidelijke scheidingen zijn te onderkennen in de verschillende fasen van het project. Een korte samenvatting van de werkwijzebeschrijving wordt hieronder vermeldt. Deze is opgesplitst naar de verscheidene fasen van het project. Na de samenvatting van de werkwijzebeschrijving worden de zaken van tevredenheid en ontevredenheid besproken.

Een solide basis voor ieder project is het opstellen van een adequaat plan van aanpak. Hierin staat het gehele project, inclusief structuur, globale & gedetailleerde planning, werkzaamheden, methoden, e.d. beschreven. Het plan van aanpak draagt onder meer zorg voor de bewaking van het project en een realisatie van een goede start.
Toen het plan van aanpak was opgesteld, en het project dus feitelijk was gedefinieerd, kon de definitiefase van het afstudeertraject worden uitgevoerd.

De tweede fase in het project betrof de definitiefase, waarin het te ontwikkelen systeem werd gedefinieerd aan de hand van een aantal aspecten. Door middel van het definiëren van het ontwikkelscenario kon er in eerste instantie een globaal beeld worden gevormd omtrent het te ontwikkelen systeem. Vervolgens ben ik aan de slag gegaan met een analyse van het huidige Marti-IT Update Centrum en SWOT-analyses op Update Centrums van andere softwarepakketten. Op die manier was ik in staat om positieve en negatieve aspecten in kaart te brengen van de Update Centrums en (nog meer) ideeën te vormen omtrent het nieuwe systeem. Na deze analyses heb ik mijn werkzaamheden gericht op het definiëren van de systeemeisen van het systeem, zodat duidelijk in kaart gebracht kon worden aan welke eisen het systeem zou moeten voldoen. Op basis van de systeemeisen was ik in staat om een systeemconcept te maken, waarin op een globaal niveau in kaart gebracht was hoe het ontwerp van het systeem zou moeten zijn. De laatste twee activiteiten in de definitiefase waren het belichten van de technische structuur en de organisatorische inrichting.
Op basis van de resultaten van de definitiestudie ben ik aan de slag gegaan met het opstellen van een pilotplan. In dit pilotplan werd de globale structuur met betrekking tot de pilotontwikkeling gedefinieerd en werden de prioriteiten van de pilots vastgesteld.
Na het opleveren van het rapport met de resultaten van de definitiestudie en het pilotplan was de definitiefase van het afstudeertraject afgesloten. De pilotontwikkelingsfase van de afstudeerperiode kon vervolgens van start gaan.

De daadwerkelijke ontwikkeling van het Marti-IT Update / Build Centrum ging van start met het begin van de eerste pilot. De doelstellingen met betrekking tot de eerste pilot richtte zich hoofdzakelijk op het herschrijven van de bestaande functionaliteit in de .NET programmeertaal C#. De eerste activiteit in de eerste pilot richtte zich op het opstellen van een pilotontwikkelplan, waarin de globale structuur uit het pilotplan zou worden verfijnd. Op deze manier kon het verloop van de pilot in kaart worden gebracht.
Nadat het pilotontwikkelplan was opgesteld kon de daadwerkelijke ontwikkeling van de onderdelen van de pilot worden uitgevoerd. De ontwikkeling van de eerste pilot kenmerkte zich enerzijds door het gebruik van herbruikbare componenten en anderzijds door het ontwikkelen van eigen componenten. Nadat de functionaliteiten met betrekking tot de eerste pilot waren ontwikkeld hebben er testwerkzaamheden plaatsgevonden om te kijken of de ontwikkelde functionaliteiten overeenkwamen met hetgeen wat er vooraf gedefinieerd was. Hierbij bleek dat er nog een aantal kleine foutjes in het systeem zaten. Met de afronding van de testwerkzaamheden was de eerste pilot voltooid. Het uitvoeren van de eerste pilot heeft me echter wel een week vertraging opgeleverd.

Nadat de eerste pilot was voltooid kon de tweede pilot van start gaan. Ook hier ben ik weer begonnen met het opstellen van een pilotontwikkelplan, zodat de gedetailleerde structuur van de pilot in kaart kon worden gebracht. Op basis van het pilotontwikkelplan is de ontwikkeling in de tweede pilot van start gegaan. De ontwikkelingen van de tweede pilot richtte zich op de functionaliteiten voor de medewerkers van Marti-IT, wat resulteerde in het zogenaamde Marti-IT Build Centrum. Tijdens de ontwikkeling van de tweede pilot is er tevens gebruik gemaakt van herbruikbare componenten en gebruik gemaakt van eigen componenten. Nadat de ontwikkeling was afgerond met betrekking tot de tweede pilot vonden er testwerkzaamheden plaats. Na het uitvoeren van de testwerkzaamheden was de tweede pilot afgerond.

© 2005 Mathieu Brüning (20010019)

De derde en tevens laatste pilot in de pilotontwikkelingsfase richtte zich op het verbeteren van de gebruikersvriendelijkheid van het systeem. Het opstellen van een pilotontwikkelplan was ook hier weer een logische eerste stap. Nadat het pilotontwikkelplan was opgesteld kon de daadwerkelijk ontwikkeling in de derde pilot plaatsvinden. De ontwikkeling richtte zich voornamelijk op het maken van een dynamische menu, een hoofdmenu, een functionaliteit voor het weergegeven van nieuwspagina's en het verbeteren van de reeds ontwikkelde vensters. Nadat de ontwikkeling was afgerond hebben er nog wat testwerkzaamheden plaatsgevonden.
Met de afronding van de derde pilot was de pilotontwikkelingsfase ook afgerond. De volgende fase betrof de invoering / nazorg fase.

De invoering / nazorgfase richtte zich hoofdzakelijk op het opstellen van handleidingen, het uitvoeren van acceptatietests en het overdragen van het Marti-IT Update / Build Centrum en de bijbehorende documentatie. Daarnaast bestond de invoering / nazorgfase uit het evalueren van het afstudeertraject en het schrijven van de afstudeerscriptie.

Samenvattend kan ik stellen dat ik over het algemeen zeer tevreden ben over de procesgang met betrekking tot het afstudeertraject. De werkwijze die ik gehanteerd heb tijdens het traject, het structureel aanpakken van het afstudeertraject door onder meer het opstellen van een solide plan van aanpak, heb ik ervaren als prettig. Dit met name omdat de structuur van het project op deze manier zeer duidelijk was.
Het vooraf nadenken over de structurele aanpak van het project resulteerde erin dat er een duidelijk overzicht van de opdrachten en een duidelijke planning was. Daarnaast speelde het uitvoeren van een definitiestudie een belangrijke rol bij het in kaart brengen van de te ontwikkelen systemen.

Het uitvoeren van de werkzaamheden heb ik veelal met veel plezier uitgevoerd, daar ik ze leuk en interessant vond. Met name het beter leren kennen van het Microsoft .NET framework en specifiek de C# programmeertaal heb ik ervaren als zeer prettig. Dit vooral doordat ik van mening ben dat het .NET framework een belangrijk aspect is van toekomstige softwareontwikkeling. Op dit moment is er al een duidelijke trend gaande in de ICT-branche, waarbij veel van de grotere ICT-organisaties (CapGemini, LogicaCMG, etc.) specifiek op zoek zijn naar .NET ontwikkelaars.

Met zekerheid kan ik stellen dat ik in toekomstige projecten een zelfde gestructureerde aanpak zal hanteren, bestaande uit het allereerst opstellen van een solide plan van aanpak met een duidelijk overzicht van de fasen en bijbehorende opdrachten.
Naast de tevredenheid met betrekking tot de procesgang ben ik tevens van mening dat de gestelde doelstellingen met betrekking tot het leerproces behaald zijn. Deze doelstellingen richtten zich over het algemeen op het opdoen van praktijkervaring in de ICT-vakwereld. Het uitvoeren van de werkzaamheden, het functioneren in een professionele bedrijfsomgeving, etc. zorgden er onder andere voor dat de nodige praktijkervaring is opgedaan tijdens het afstudeertraject.

De opgedane ervaringen met het .NET framework en vooral de C# programmeertaal heb ik ervaren als zeer prettig. Het ontwikkelen middels de C# programmeertaal is naar mijn idee een belangrijk aspect van toekomstige softwareontwikkeling, aangezien de mogelijkheden oneindig zijn en ondersteund worden door een zeer krachtige en uitgebreide IDE (Microsoft Visual Studio .NET 2003). Inmiddels heb ik een glimp mogen opvangen van de beta versie van de C# 2.0 programmeertaal en de nieuwe Visual Studio .NET 2005 IDE en ik moet zeggen dat ik er alleen nogm aar goede ervaringen mee heb. Met name doordat er over standaard zaken als ondermeer data afhandeling is nagedacht en er standaard oplossingen voor geboden worden ben ik van mening dat toekomstige softwareontwikkeling een stuk efficiënter uitgevoerd kan worden middels het .NET framework v2.0.

Wat minder tevreden ben ik over het feit dat er op sommige momenten wellicht iets werd afgeweken van de daadwerkelijke afstudeeropdracht. Aangezien het afstudeerproject werd uitgevoerd binnen een professionele organisatie, wordt er van mij als afstudeerder verwacht dat ik op sommige momenten bijspring bij de werkzaamheden van het bedrijf op momenten dat dit van mij verlangt wordt. Op sommige momenten is het zodoende voorgekomen dat ik inkomende telefoontjes heb moeten beantwoorden, omdat het zo enorm druk was dat niemand anders in staat was om de telefoon op te nemen. Dit is echter niet zo heel veel voorgekomen dus hoofdzakelijk was ik wel in staat om goed door te werken aan het project.

9.2 Product

De belangrijkste producten die tijdens het afstudeertraject zijn opgeleverd zijn als volgt:

1. Rapport met de specificaties van het Marti-IT Update / Build Centrum;
2. Testrapport acceptatietests gebruikers Marti-IT software;
3. Handleidingen t.b.v. het gebruik van Marti-IT Update / Build Centrum;
4. Nieuw Marti-IT Update / Build Centrum.

Hieronder zullen de producten individueel worden geëvalueerd.

1. Rapport met de specificaties van het Marti-IT Update / Build Centrum

De doelstellingen met betrekking tot het rapport met de specificaties van het Marti-IT Update / Build Centrum richtte zich voornamelijk op het opleveren van een rapport waarin de specificaties zijn vastgelegd omtrent het te ontwikkelen systeem. Naar mijn mening ben ik zeer tevreden over dit rapport. Dit vooral doordat er op een gedetailleerd niveau is vastgelegd aan welke eisen het systeem moet voldoen, door middel van de definiëring van de systeemeisen. De indeling van de systeemeisen op basis van prioriteit speelde vervolgens een belangrijke rol bij het structureren van de pilotontwikkelingsfase. In het systeemconcept zijn de belangrijkste aspecten van het systeem vastgelegd, waarop ik in het verloop van het project voortgeborduurd heb.

Samenvattend kan ik stellen dat ik over het rapport met de specificaties van het Marti-IT Update / Build Centrum zeer tevreden ben. Door op een gedetailleerd niveau vast te leggen aan welke aspecten het te ontwikkelen systeem diende te voldoen, was ik in staat om structuur van de rest van het project in kaart te brengen. Opstellen van het pilotplan en de pilotontwikkelplannen was hoofdwaarschijnlijk ook niet mogelijk geweest zonder het rapport met de specificaties van het Marti-IT Update / Build Centrum.

2. Testrapport acceptatietest gebruikers Marti-IT software

Het testrapport met betrekking tot de acceptatietest met de gebruikers van Marti-IT is een gestructureerd document, waar in eerste instantie de kenmerken van de testomgeving worden vastgelegd. Vervolgens worden de testscenario's gedefinieerd door middel van een omschrijving van het testscenario, de logische testgevallen met betrekking tot het scenario en de kenmerken van het testscenario. Het hanteren van een duidelijke structuur bij het opstellen van het testplan heeft er met name voor gezorgd dat de benodigde aspecten rondom het uitvoeren van de testwerkzaamheden zijn vastgelegd en dat het uitvoeren van de testwerkzaamheden op een gestructureerde manier plaats kon vinden.
Het opstellen van het testrapport is naar mijn mening een vitaal onderdeel geweest voor de gestructureerde uitvoering van de testwerkzaamheden bij de gebruikers van de Marti-IT software.

3. Handleidingen t.b.v. het gebruik van Marti-IT Update / Build Centrum

De doelstelling met betrekking tot het opleveren van handleidingen ten behoeve van het gebruik van het Marti-IT Update / Build Centrum, richtte zich hoofdzakelijk op het opleveren van beknopte handleidingen in de vorm van reference cards. De reference cards dienen informatie te verschaffen aan de toekomstige gebruikers van de applicaties door op een beknopte, maar volledige manier informatie te verschaffen over de werking van het Marti-IT Update / Build Centrum.

Op basis van de ontwikkelde functionaliteiten van het systeem zal ik aantonen of er informatie wordt vermeldt omtrent de betreffende functionaliteiten op de handleidingen. Op die manier kan ik bepalen of er daadwerkelijk wel voldoende informatie wordt verschaft middels de reference cards aan de toekomstige gebruikers van de applicaties.

© 2005 Mathieu Brüning (20010019)

De ontwikkelde functionaliteiten worden hieronder vermeldt:

Marti-IT Update Centrum:
- Downloaden updates
- Uploaden van back-ups, bestanden en/of mappen en standaard uploads
- Extra functionaliteiten als het downloaden van historische updates en het maken van een back-up

Marti-IT Build Centrum:
- Het beheren van gebruikersgegevens
- Het beheren van software gegevens
- Het beheren van updates in de database
- Het beheren van updates op de Marti-IT server
- Automatisch downloaden en archiveren van verzonden gegevens

Naar mijn mening dekken de reference cards de volledige lading ten behoeve van het gebruik van de systemen, aangezien feitelijk alle ontwikkelde functionaliteiten met voldoende diepgang worden behandeld qua werking in de handleidingen. Zodoende ben ik van mening dat de handleidingen goede producten zijn voor het bieden van ondersteuning tijdens het toekomstige gebruik van het Marti-IT Update / Build Centrum.

4. Nieuw Marti-IT Update / Build Centrum

Het nieuwe Marti-IT Update Centrum en het Marti-IT Build Centrum zijn de twee producten waar de ontwikkeling in het gehele afstudeerproject feitelijk om gedraaid heeft. Om de ontwikkelde systemen op een adequate manier te kunnen evalueren refereer ik hieronder terug naar de doelstellingen aan de producten, zodat vastgesteld kan worden of de doelstellingen behaald zijn.

De doelstellingen aan het ontwikkelen van de eindproducten waren:

- ✓ Een nieuw Marti-IT Update Centrum (voor de gebruikers van de Marti-IT software), bestaande uit:
  - De functionaliteit van de bestaande applicatie, herschreven in de .NET programmeertaal C#
    - Downloaden van updates, persoonlijke bestanden en andere hulpmiddelen;
    - Uploaden van back-ups, persoonlijke bestanden en andere gegevens;
  - Een nieuwe interface, waarin de gebruikersvriendelijkheid meetbaar beter wordt nageleefd dan in de huidige applicatie
    - Nieuwe kleurensamenstelling;
    - Nieuwe menustructuur;
    - Nieuwspagina's in html formaat, die worden getoond tijdens het downloaden van de updates.
- ✓ Een Marti-IT Build Centrum (voor de medewerkers van Marti-IT), bestaande uit:
  - Extra functionaliteiten
    - Het automatiseren van bepaalde processen, zoals bijvoorbeeld het klaarzetten van een update en het downloaden en archiveren van door gebruikers verzonden gegevens;
    - Extra mogelijkheden met betrekking tot het in- en uitpakken in het .zip formaat;
    - Extra mogelijkheden op het gebied van commandline opties.

Op basis van de bovenstaande doelstellingen zal ik nu de vergelijking maken met hetgeen wat er ontwikkeld is, door de doelstellingen één voor één langs te lopen en te evalueren.

Het herschrijven van de functionaliteit van de bestaande applicatie in de .NET programmeertaal is zeer zeker behaald. Alle mogelijke updates die gebruikers konden verkrijgen middels het oude Marti-IT Update Centrum zijn eveneens te verkrijgen middels het nieuwe systeem. Dit nieuwe systeem is volledig herschreven in de .NET programmeertaal C#.
Aangezien de gebruikers in staat zijn om alle mogelijk updates, zoals die beschikbaar waren in het oude systeem, te downloaden is er geen sprake van een achtergang qua functionaliteit. Met de toevoeging van nieuwe functionaliteiten als ondermeer het inventariseren van alle benodigde updates en het downloaden van meerdere updates tegelijkertijd is er juist sprake van een vooruitgang in de geïmplementeerde functionaliteit van het systeem.

© 2005 Mathieu Brüning (20010019)

Het ontwikkelen van een nieuwe interface, waarbij de gebruikersvriendelijkheid beter wordt nageleefd, is naar mijn idee ook zeker geslaagd. Een sterk punt met betrekking tot de verbetering van de gebruikersvriendelijkheid is naar mijn idee de implementatie van het hoofdmenu en het dynamische menu. Deze implementatie heeft er namelijk voor gezorgd dat de aansturing van de functionaliteiten in het systeem een stuk eenduidiger en meer intuïtief geworden. Gebruikers hoeven grotendeels niet eens het door menu te bladeren, aangezien het hoofdmenu zorg draagt voor het aanroepen van de belangrijkste functionaliteiten van het systeem.
De implementatie van het tonen van de nieuwspagina's tijdens het verzendproces is naar mijn idee niet direct gerelateerd aan het verbeteren van de gebruikersvriendelijkheid, maar is met name bedoeld om gebruikers dichter te betrekken bij Marti-IT als organisatie. Doordat informatie omtrent Marti-IT en de Marti-IT software wordt getoond kunnen gebruikers worden voorgelicht, zonder dat daar kostbare tijd van de klant mee verloren gaat.
De nieuwe kleurensamenstelling en de verbeteringen in de vensters van het systeem zijn met name gericht op het duidelijker en meer intuïtief maken van de vensters. Door middel van duidelijke titelbalken in herkenbare kleuren is het naar mijn mening duidelijker geworden waarvoor de verschillende secties op een venster kunnen worden gebruikt. Zodoende een verbetering van de gebruikersvriendelijkheid.

Waar het Marti-IT Update Centrum met name gericht is op de gebruikers van de Marti-IT software, richt het Marti-IT Build Centrum zich voornamelijk op de medewerkers van Marti-IT. De doelstellingen van het Marti-IT Build Centrum richtten zich hierbij met name op functionaliteiten voor het beheren van de informatie in de database voor het gebruik van het Marti-IT Update Centrum. Naar mijn mening zijn de ontwikkelde functionaliteiten voor het beheren van de informatie in de database naar tevredenheid. Het is namelijk mogelijk om de gebruikersgegevens te synchroon te houden met de gegevens van de Marti-IT applicatie Serv-it en de informatie omtrent applicaties, download types en updates te beheren. Daarnaast is het mogelijk om de update bestanden te onderhouden, doordat functionaliteiten voor het maken van updates en het klaarzetten van updates op de Marti-IT server zijn geïmplementeerd in het Marti-IT Build Centrum.
De extra mogelijkheden op het gebied van het .zip formaat zijn eveneens geïmplementeerd in de standaard C# library van Marti-IT, zodat het ondermeer mogelijk is om mappenstructuren recursief in en uit te pakken.
Voor het ontwikkelen van de extra mogelijkheden op het gebied van commandline opties was er helaas geen tijd meer. Door de kleine uitloop van de eerste pilot en de verstreken tijd van de tweede pilot, stond ik voor de beslissing om de tweede pilot af te ronden of de commandline opties nog te ontwikkelen. Aangezien het verbeteren van de gebruikersvriendelijkheid een fundamentele doelstelling was omtrent het Marti-IT Update Centrum, heb ik besloten om de tweede pilot af te ronden en de derde pilot te starten. Het ontwikkelen van de extra mogelijkheden op het gebied van commandline opties was zodoende niet meer aan de orde.

Samenvattend kan ik stellen dat ik over het algemeen zeer tevreden ben over de opgeleverde producten. Met name door het feit dat bijna alle doelstellingen behaald zijn kan ik spreken over een tevreden product. De acceptatietests met de gebruikers van de Marti-IT software en de medewerkers van Marti-IT wezen ook uit dat de uitvoerders van de testwerkzaamheden zeer tevreden waren over de systemen.

De probleemstelling van het project richtte zich voornamelijk op vijf gebieden namelijk:

- ✓ de niet meer volledige functionaliteit ten aanzien van de behoeftes van de klant
- ✓ de moeilijke onderhoudbaarheid doordat er vier individuele Update Centrums in omloop zijn
- ✓ het ontbreken van functionaliteiten ten behoeve van de medewerkers van Marti-IT
- ✓ de gebruikersvriendelijkheid die niet optimaal was
- ✓ de applicatie is ontwikkeld in de Visual Objects programmeertaal

Het uitbreiden van de functionaliteit van de applicatie door ondermeer het toevoegen van een functionaliteit voor het maken van een gegevensback-up (hier was een grote behoefte naar) zorgt ervoor dat het eerste probleemgebied niet meer aan de orde is. Het samenbundelen van de vier individuele Update Centrums tot één enkel Marti-IT Update Centrum omvat het tweede probleemgebied en zorgt ervoor dat de onderhoudbaarheid in de toekomst eenvoudiger kan verlopen. Het implementeren van diverse functionaliteiten voor de medewerkers van Marti-IT, zoals het klaarzetten van updates en het automatisch downloaden van back-ups rekent af met het derde probleemgebied. Het verbeteren van de gebruikersvriendelijkheid door middel van ondermeer het toevoegen van een hoofdmenu en het implementeren van een dynamische menustructuur draagt zorg voor het vierde probleemgebied. Aangezien de ontwikkeling van het nieuwe Marti-IT Update Centrum zo goed als volledig heeft plaatsgevonden in de C# programmeertaal is er ook afgerekend met het laatste probleemgebied.

Op basis van deze informatie in combinatie met het behalen van bijna alle doelstellingen kan ik concluderen dat de producten naar tevredenheid ontwikkeld zijn.

Nawoord

Als gevolg van het afstudeertraject waar ik mezelf de afgelopen 18 weken mee bezig heb gehouden wil ik Marti-IT bedanken voor de leuke en leerzame tijd tijdens het afstudeertraject. Het was erg gezellig met de medewerkers van Marti-IT en de opdracht was daarnaast ook erg interessant om uit te voeren.

Met name wil ik bedrijfsmentor Marc Verkade hartelijk bedanken voor het parttime contract dat ik aangeboden heb gekregen. Op deze manier krijg ik de kans om het ontwikkelde systeem te implementeren en verdere praktijkervaring op te doen in een professionele organisatie.

Naaldwijk, juni 2005, M.D. Brüning

© 2005 Mathieu Brüning (20010019)

Literatuurlijst

Boeken

Tolido, R.J.H., *IAD, Het evolutionair ontwikkelen van informatiesystemen*. Utrecht: Cap Volmac, 1996.

Eriksson, H., M. Penker, *De UML Toolkit*. Plaats: Academic Service, 1999.

Deitel, H., P. Deitel, J. Listfield e.a., *C# for experienced programmers*. Plaats: Prentice Hall, 2003

Jones, A., *C# programmer's cookbook*. Microsoft Press, Redmond: 2004

Scharp, J. en J. Jagger, *Microsoft Visual C# .NET*. Microsoft Press, Redmond: 2004

Teilhet, S. en J. Hilyard, *C# Cookbook*. O'Reilly & Associates, Inc., Sebastopol: 2004

Hejlsberg, A., S. Wiltamuth en P. Golde, *The C# programming language*. Pearson Education, Boston: 2004

Websites

Microsoft Developer Network, http://msdn.microsoft.com. Microsoft corporation.

Tim Anderson's ITWriting, http://www.itwriting.com/. Tim Anderson

MySQL, http://www.mysql.com. MySQL

The Code Project, http://www.codeproject.com. The Code Project

CSharpFriends, http://www.csharpfriends.com. CSharpFriends

C-Sharp Corner. http://www.c-sharpcorner.com. C-Sharp Corner

Active FTP vs. Passive FTP, a Definitive Explanation, http://www.lenglet.com/docs/ftp/ftp.html

Enterprise Distributed Technologies, http://www.enterprisedt.com/. Enterprise Distributed Technologies

Lijst van figuren

FIGUUR 1: *ORGANOGRAM MARTI-IT INCLUSIEF EXTERNE SAMENWERKINGSVERBANDEN* 13
FIGUUR 3: *GEDETAILLEERDE TIJDSPLANNING PROJECT* 27
FIGUUR 4: *GLOBALE WERKING BESTAANDE MARTI-IT UPDATE CENTRUM* 35
FIGUUR 5: *USE-CASE DIAGRAM MET BETREKKING TOT DE FUNCTIONALITEIT VAN HET BESTAANDE MARTI-IT UPDATE CENTRUM* 36
FIGUUR 6: *GLOBALE WERKING VAN HET NIEUWE SYSTEEM* 39
FIGUUR 7: *STRUCTUUR UPDATER APPLICATION BLOCK* 42
FIGUUR 8: *GLOBALE STRUCTUUR VAN HET SYSTEEM* 43
FIGUUR 9: *INDELING PROJECT* 43
FIGUUR 10: *AUTORISATIE / LOGIN VENSTER* 44
FIGUUR 11: *DATABASE STRUCTUUR* 47
FIGUUR 12: *MIDDELS DE MICROSOFT VISUAL STUDIO IDE KUNNEN DE DATASETS OP GRAFISCHE WIJZE WORDEN ONTWIKKELD* 49
FIGUUR 15: *ONTWIKKELING VAN HET DOWNLOAD VENSTER MIDDELS DE MICROSOFT VISUAL STUDIO IDE* 54
FIGUUR 16: *BESTAANDE INTERFACE VOOR HET UPLOADEN VAN EEN BACK-UP* 57
FIGUUR 17: *BESTAANDE INTERFACE VOOR HET UPLOADEN VAN BESTANDEN, MAPPEN OF STANDAARD UPLOADS. DE (STATISCHE) KNOPPEN ONDERAAN HET VENSTER KUNNEN GEBRUIKT WORDEN VOOR HET UPLOADEN VAN VOORGEDEFINIEERDE MAPPEN* 58
FIGUUR 18: *GLOBAAL ONTWERP UPLOAD FUNCTIONALITEITEN* 59
FIGUUR 19: *VENSTER VOOR HET UPLOADEN VAN EEN BACK-UP* 60
FIGUUR 20: *VENSTER VOOR HET UPLOADEN VAN BESTANDEN OF MAPPEN* 60
FIGUUR 21: *VENSTER VOOR HET UPLOADEN VAN STANDAARD UPLOADS* 61
FIGUUR 22: *ONTWERP SYNCHRONISATIE FUNCTIONALITEIT* 63
FIGUUR 23: *VENSTER VOOR HET SELECTEREN VAN GEBRUIKERS* 64
FIGUUR 24: *VENSTER MET DE DETAILGEGEVENS VAN EEN GEBRUIKER* 64
FIGUUR 25: *VENSTER VOOR HET SYNCHRONISEREN VAN GEBRUIKERSGEGEVENS MET SERV-IT* 65
FIGUUR 27: *VENSTERS VOOR HET INZIEN VAN DETAILGEGEVENS VAN APPLICATIES EN DOWNLOAD TYPES* 66
FIGUUR 28: *VENSTER VOOR HET SELECTEREN VAN UPDATES* 68
FIGUUR 29: *VENSTER VOOR HET BEHEREN VAN INFORMATIE OMTRENT UPDATES* 68
FIGUUR 30: *VENSTER VOOR HET MAKEN VAN GEGEVENSBACK-UPS* 70
FIGUUR 31: *ONTWERP DYNAMISCH MENU* 73
FIGUUR 32: *HOOFDMENU WAARMEE GEBRUIKERS IN STAAT WORDEN GESTELD OM DE BELANGRIJKSTE FUNCTIONALITEITEN VAN DE APPLICATIE TE KUNNEN AANROEPEN* 74
FIGUUR 33: *NIEUWSPAGINA'S WORDEN GETOOND WANNEER EEN HET SYSTEEM BEZIG IS HET MET VERZENDEN OF ONTVANGEN VAN GEGEVENS* 75
FIGUUR 34: *CLASSCOLORPROFILE* 76
FIGUUR 35: *VOORBEELDEN VAN EEN AANGEPASTE VENSTERS, WAARIN EEN MARTI-IT LOGO EN DE NIEUWE KLEURENSAMENSTELLING IN DE TITELBALKEN ZIJN VERWERKT* 77

© 2005 Mathieu Brüning (20010019)

Verklarende woordenlijst

Woord	Verklaring
Iterative Application Development, IAD	Methodiek voor het ontwikkelen van informatiesystemen. Zie ook § 7.2
.NET framework	Het .NET framework is een verzameling van Microsoft technologieën als programma's, code libraries en programmeertalen in samenwerking met andere software om systeemontwikkeling te kunnen realiseren.
C#	Programmeertaal van Microsoft gebaseerd op C en C++
Unified Modelling Language	Verzameling van diagrammen en een definitie van taal, waar informatiesystemen mee in kaart kunnen worden gebracht. Zie ook § 7.1.3
Untyped dataset	Verzameling van gegevens in het geheugen van de computer. De structuur van deze gegevens is niet vastgelegd.
Typed dataset	Verzameling van gegevens in het geheugen van de computer. De structuur van deze gegevens is vastgelegd door middel van veldennamen en gegevenstypen.
Pilot	Tussenversies van het uiteindelijke beoogde informatiesysteem.
Serv-it	Applicatie binnen Marti-IT voor het opslaan van gegevens omtrent gebruikers.
Datagrid	Een veld op een venster voor de weergave van gegevens in kolommen en rijen.
SQL	Structured Query Language. Een vraagtaal om gegevens mee op te kunnen vragen in gegevensbanken aan de hand van expressies.
Cascading Style Sheets (CCS)	Opmaak sjablonen voor het gebruik in HTML-pagina's
Open-source	Het beschikbaar zijn van broncode
Visual Objects	Programmeertaal gebaseerd op Clipper, welke veelal gebruikt wordt binnen Marti-IT voor softwareontwikkeling
Microsoft Application Blocks	Herbruikbare componenten voor kant-en-klare oplossingen in de .NET project.

© 2005 Mathieu Brüning (20010019)

BIJLAGEN

Het laatste deel van deze scriptie omvat de bijlagen ten behoeve van het rapport. Deze bijlagen bestaan uit een drietal onderdelen, waar in eerste instantie de opdrachtomschrijving aan bod komt. Bijlage twee richt zich op het volledige plan van aanpak, waarop bijlage drie zich vervolgens richt op de resultaten van de definitiestudie. Bijlage vier bevat de pilotontwikkelplannen, welke gebruikt zijn voor de structurering van elke pilot. De reference cards ten behoeve van het gebruik van de applicaties zijn de te vinden bijlage 5. De laatste bijlage richt zich tot slot op de resultaten van de acceptatietests met de gebruikers van de Marti-IT software.

© 2005 Mathieu Brüning (20010019)

Bijlage 1: Opdrachtomschrijving

Mathieu Bruning

Kenmerk: DOA2005-1.1.023

Ontwikkeling van een communicatietool voor het updaten van software bij Marti-IT

Inleiding (organisatorische omgeving, kader, historie)

Marti-IT, een jong en dynamisch bedrijf, gevestigd in Naaldwijk, is specialist op het gebied van software voor bakkerijen en orthopedisten. Momenteel zijn er ruim 450 bakkers en zo\'n 75 orthopedisten die gebruik maken van de Marti-IT software.
Naast de software voor bakkerijen en orthopedisten besteedt de organisatie ook nog een deel van haar werkzaamheden aan maatwerksoftware en advisering. Het bedrijf telt een negental medewerkers, waaronder drie ontwikkelaars, drie helpdeskmedewerkers, een consultant, een verkoper en een administratieve medewerkster.

Op het gebied van de ontwikkeling van software is het beleid van Marti-IT hoofdzakelijk dat de wensen en eisen van klanten voorop staan. Continue houdt het bedrijf zich bezig met de ontwikkeling van de softwarepakketten, waardoor er wekelijks nieuwe updates verschijnen voor de Marti-IT software. Sprake van een concreet toegepaste systeemontwikkelingsmethodiek is er feitelijk niet, aangezien de continue doorlopende ontwikkeling aan de softwarepakketten veelal zorgt voor directe oplossingen zonder voortraject.

Toen in het verleden gebleken is dat het te veel inspanning kostte om updates te verzenden van en naar de gebruikers van de Marti-IT software is er binnen relatief korte tijd een applicatie ontwikkeld dat functioneert als doorgeefluik tussen de klanten en Marti-IT. Het doorgeven van gegevens van en naar de klanten middels het zogenaamde Marti-IT Update Centrum stelt klanten op zeer eenvoudige wijze in staat om de nieuwste updates en andere bestanden te downloaden voor de Marti-IT software. Daarnaast is het tevens mogelijk om met het Marti-IT Update Centrum gegevens te verzenden naar het bedrijf, zodat eventuele problemen in de software gericht opgelost kunnen worden door de helpdesk. Na enkele jaren veelvuldig gebruikt te zijn, blijkt dat er behoefte is naar een uitbreiding van de functionaliteit van het systeem.

Probleemstelling

Het huidige Marti-IT Update Centrum is een ouder product dat al lange tijd geleden is ontwikkeld, wat ervoor gezorgd heeft dat de huidige functionaliteit van het systeem niet meer volledig de behoeftes van de klant dekt. Feitelijk is de basis functionaliteit van de applicatie dus wel in orde, maar echter niet meer volledig ten aanzien van de huidige behoeftes vanuit de gebruikersrichting. Naast de behoefte naar extra functionaliteiten vanuit de gebruikersrichting schiet het huidige systeem ook tekort op het gebied van beveiliging en de structurering in de beschikbare documentatie.

Op het gebied van de gebruikte functionaliteiten binnen Marti-IT kan de applicatie ook nog sterk verbeterd worden. Tijdsintensieve routinetaken als het downloaden en archiveren van gegevensback-ups van gebruikers worden nu nog handmatig uitgevoerd, waarbij veel tijd verloren gaat.

Daarnaast kan met name de gebruikersvriendelijkheid van de applicatie kan sterk verbeterd worden, aangezien o.a. de menustructuur van de grafisch interface nog niet optimaal is. De applicatie is tevens al redelijk gedateerd wat ervoor heeft gezorgd dat het cosmetische gedeelte van de applicatie zich niet meer kan meten met de hedendaagse applicaties.

Tot slot is het huidige Marti-IT Update Centrum geschreven in de programmeertaal Visual Objects. Het ontwikkelen in de Visual Objects programmeertaal past echter niet meer bij het toekomstige softwareontwikkelingsbeleid van het bedrijf, aangezien het bedrijf in de toekomst volledig wil overgaan op het ontwikkelen in de .NET programmeertaal C#.

© 2005 Mathieu Brüning (20010019)

Doelstelling van de opdracht

De doelstelling van de opdracht richt zich op het ontwikkelen van een nieuw Marti-IT Update Centrum, waar naast de functionaliteit ook de onderhoudbaarheid een belangrijk aspect is. Een ander belangrijk aspect van het nieuwe Update Centrum is dat het grotendeels geschreven moet worden in de .NET programmeertaal C#, aangezien het toekomstbeleid van softwareontwikkeling binnen Marti-IT zich richt op ontwikkelen in de .NET programmeertaal C#. De ontwikkeling van het systeem in C# dient er tevens voor te zorgen dat eventuele herbruikbare componenten als input dienen voor het opzetten van een toekomstig C# framework voor Marti-IT. Tot slot dient de applicatie ervoor te zorgen dat sommige processen, zoals bijvoorbeeld het klaarzetten van een nieuwe update op het Internet of het uitpakken van een gegevensback-up, geautomatiseerd dienen te worden. Op deze manier kunnen dergelijke processen binnen de organisatie geoptimaliseerd worden.

Uitgangssituatie

Benodigde software

- ✓ Software met betrekking tot het ontwikkelen van technische ontwerpen (MS Visio)
- ✓ Software met betrekking tot het ontwikkelen in HTML en CSS (MS Visual Studio 2003)
- ✓ Software met betrekking tot het grafisch ontwikkelen van afbeeldingen e.d. (Adobe Photoshop)

Aanwezige ideeën

- ✓ Ontwikkeling zal geschieden in de .NET programmeertaal C#
- ✓ Nieuwspagina's weergegeven tijdens het downloaden, zodat de gebruikers op de hoogte blijven van de nieuwste ontwikkelingen.

Concrete werkzaamheden

Uit te voeren activiteiten
Het opleveren van een volledig nieuw Marti-IT Update Centrum, bestaande uit:

ORIENTATIEFASE
Oriënteren op de organisatie
Het samenstellen van een Plan van Aanpak

DEFINITIEFASE
Definiëren ontwikkelscenario
Analyseren van het bestaande Marti-IT Update Centrum
Definiëren systeemeisen a.d.h.v. eisen/wensen opdrachtgever en analyse resultaten huidige applicatie
Bepalen systeemconcept
Beschouwen technische structuur
Beschouwen organisatorische inrichting
Opstellen van een pilotplan

PILOTONTWIKKELING
Specificeren globaal-functionele en globaal-technische structuur per pilot
Specificeren globaal-organisatorische structuur per pilot
Opstellen van een pilotontwikkelingplan voor de betreffende pilot
Ontwerpen van de software-bouweenheden van de betreffende pilot
Bouwen van de software-bouweenheden van de betreffende pilot
Het integreren van de ontwikkelde bouweenheden
Opstellen testplan voor de ontwikkelde bouweenheden
Testen ontwikkelde bouweenheden
Beoordelen ontwikkelde bouweenheden a.d.h.v. testresultaten

© 2005 Mathieu Brüning (20010019)

INVOERING
Opstellen van invoeringsprocedures Marti-IT Update Centrum
Opstellen van handleiding
Acceptatietest met betrekking tot de opdrachtgever
Oplevering van Marti-IT Update Centrum aan de opdrachtgever
Oplevering van de applicatie documentatie aan de opdrachtgever

Te hanteren methodieken
De door mij gekozen methodiek voor het ontwikkelen/vernieuwen van de applicatie is Iterative Application Development.

Te gebruiken technieken
- ✓ Unified Modelling Language (UML), voor het ontwerpen van de software-bouweenheden
- ✓ Prototyping, voor de tussentijdse goedkeuring van software-bouweenheden
- ✓ Rapporteren, voor het samenstellen van rapporten/verslagen
- ✓ Interviewen, voor het verzamelen van de eisen/wensen m.b.t. de opdrachtgever

Te vermelden nadrukken
- ✓ Functionaliteit
- ✓ Cosmetisch ontwerp
- ✓ Onderhoudbaarheid
- ✓ Automatisering van processen
- ✓ Gebruikersvriendelijkheid
- ✓ Geschreven in C#

Resultaten voor de opdrachtgever (op te leveren produkten)

- ✓ Rapport met de specificaties voor het Marti-IT Update Centrum
- ✓ Pilotontwikkelrapporten
- ✓ Een nieuw Marti-IT Update Centrum
- ✓ Testrapport acceptatietest medewerkers Marti-IT
- ✓ Testrapport acceptatietest gebruikers Marti-IT software
- ✓ Handleiding in de vorm van een reference card

© 2005 Mathieu Brüning (20010019)

Bijlage 2: Plan van aanpak

PLAN VAN AANPAK

Aanhef:

Module:	Afstuderen	**Opdrachtgever:**	Dhr. M.L. Verkade
Docent(en):	Dhr. J.L.A. Schramp Dhr. M. Reijnhoudt	**Bedrijfsmentor:**	Dhr. M.L. Verkade
		Plaats:	Naaldwijk
Organisatie:	Marti-IT	**Datum:**	11 februari 2004

Student:

Naam:	**E-Mail:**
Brüning, Mathieu	20010019

© 2005 Mathieu Brüning (20010019)

Inhoudsopgave

INLEIDING **103**

PROJECTDEFINITIE **104**

1.1 PROBLEEMSTELLING 104
1.2 AANLEIDING VAN HET PROJECT 104
1.3 DOELSTELLING 104
1.4 PROJECTRESULTAAT 105
1.5 AFBAKENING VAN HET RESULTAAT 105
1.6 EFFECTEN VAN HET PROJECT 105
1.7 RISICOFACTOREN 105
1.8 BENODIGDE MENSEN EN MIDDELEN 106
1.8.1 Benodigde mensen 106
1.8.2 Benodigde software 106
1.8.3 Benodigde hardware 106
1.9 GEBRUIKTE METHODE EN TECHNIEKEN 106
1.9.1 Technieken 106
1.9.2 Methode 106
1.10 RANDVOORWAARDEN 107

FASERING & ACTIVITEITENPLAN **108**

1.11 WORK BREAKDOWN STRUCTURE 108
1.11.1 Plan van aanpak 108
1.11.2 Definiëren ontwikkelscenario 108
1.11.3 Analyseren bestaande Marti-IT Update Centrum 109
1.11.4 Analyseren Update Centrums van overige softwarepakketten 109
1.11.5 Definiëren systeemeisen 110
1.11.6 Bepalen systeemconcept 110
1.11.7 Beschouwen technische structuur 110
1.11.8 Beschouwen organisatorische inrichting 111
1.11.9 Opstellen pilotplan 111
1.11.10 Specificeren globaal-functionele structuur 111
1.11.11 Specificeren globaal-technische structuur 112
1.11.12 Specificeren globaal-organisatorische structuur 112
1.11.13 Opstellen pilotontwikkelplan 112
1.11.14 Ontwerpen software-bouweenheden 113
1.11.15 Bouwen software-bouweenheden 113
1.11.16 Integreren software-bouweenheden 113
1.11.17 Beoordeel en test pilotdeel 114
1.11.18 Opstellen invoeringsprocedures 114
1.11.19 Opstellen handleiding 114
1.11.20 Acceptatietest Marti-IT Update Centrum 115
1.11.21 Oplevering Marti-IT Update Centrum 115
1.11.22 Oplevering documentatie Marti-IT Update Centrum 115
1.11.23 Marti-IT Update Centrum 116
1.11.24 Schrijven afstudeerscriptie 116

© 2005 Mathieu Brüning (20010019)

1.12 BESCHRIJVING MIJLPAALPRODUCTEN, -ACTIVITEITEN 117
1.12.1 Plan van aanpak 117
1.12.2 Definiëren ontwikkelscenario 117
1.12.3 Analyseren bestaande Marti-IT Update Centrum 117
1.12.4 Analyseren Update Centrums andere softwarepakketten 117
1.12.5 Definiëren systeemeisen 117
1.12.6 Bepalen systeemconcept 118
1.12.7 Beschouwen technische structuur 118
1.12.8 Beschouwen organisatorische inrichting 118
1.12.9 Opstellen van een pilotplan 118
1.12.10 Specificeren globaal-functionele structuur 118
1.12.11 Specificeren globaal-technische structuur 118
1.12.12 Specificeren globaal-organisatorische structuur 119
1.12.13 Opstellen van een pilotontwikkelingplan 119
1.12.14 Ontwerpen van de software-bouweenheden 119
1.12.15 Bouwen van de software-bouweenheden 119
1.12.16 Het integreren van de ontwikkelde bouweenheden 119
1.12.17 Beoordeel en testpilotdeel 120
1.12.18 Opstellen van invoeringsprocedures Marti-IT Update Centrum 120
1.12.19 Opstellen van handleidingen 120
1.12.20 Acceptatietest met betrekking tot de opdrachtgever 120
1.12.21 Oplevering van Marti-IT Update Centrum aan de opdrachtgever 120
1.12.22 Oplevering van de applicatie documentatie aan de opdrachtgever 121
1.12.23 Marti-IT Update Centrum 121
1.12.24 Schrijven afstudeerscriptie 121
1.13 GLOBALE PLANNING 122

TIJDBEHEERSING 123

1.14 GEDETAILLEERDE PLANNING 123

COMMUNICATIE 125

1.15 COMMUNICATIE IN EN RONDOM HET PROJECT 125
1.16 COMMUNICATIEMEDIA EN –MIDDELEN 125

1 Inleiding

De aanleiding van dit document is het huidige afstudeertraject waarin student Mathieu Brüning zich verkeert. Om een solide basis en tevens goede start te kunnen realiseren is er de dringende behoefte naar een duidelijk plan van aanpak.

Het doel van dit plan van aanpak is een vakkundig inzicht geven, wat betreft de komende werkzaamheden, resultaten en procedures betreffende komende project.

De opbouw van dit rapport is als volgt: hoofdstuk twee geeft een beschrijving weer van de definiëring van het project. Zaken als probleemstelling, aanleiding, doelstelling, risico's en resultaat met betrekking tot het project worden in dit hoofdstuk weergegeven. Hoofdstuk 3 (Fasering & Activiteitenplan) richt zich op een uiteenzetting van de uit te voeren activiteiten tijdens het afstudeertraject. Het hoofdstuk is opgebouwd uit twee paragrafen, waarbij de eerste paragraaf diagrammen bevat welke de verscheidene activiteiten en deelactiviteiten representeren van het project. De daarop volgende paragraaf beschrijft onder andere in detail welke werkzaamheden er plaats zullen vinden tijdens het project. Tevens wordt er in ditzelfde hoofdstuk een globale planning van het traject weergegeven. Hoofdstuk 4 (Tijdbeheersing) bevat een gedetailleerde planning, waarin duidelijk wordt beschreven wanneer welke (deel)producten worden gemaakt. Een urenverantwoording is ook opgenomen in de gedetailleerde planning. Het hoofdstuk 'Communicatie' beschrijft hoe de communicatie tijdens het project tussen opdrachtgever en student zal plaatsvinden.

© 2005 Mathieu Brüning (20010019)

2 Projectdefinitie

2.1 Probleemstelling

Het huidige Marti-IT Update Centrum is al lange tijd geleden ontwikkeld en laat op sommige fronten een beetje te wensen over.
Met name de gebruikersvriendelijkheid van de applicatie kan sterk verbeterd worden. In feite is de applicatie functioneel wel in orde, maar is geschreven in een enigszins verouderde programmeertaal Visual Objects. De huidige functionaliteit moet echter worden uitgebreid en verbeterd, zodat er onder andere betere ondersteuning is op het gebied van .zip en commandline opties. Daarnaast dient de menustructuur aangepast te worden, zodat de gebruikersvriendelijkheid optimaal wordt.
Tot slot is de applicatie al redelijk gedateerd, wat ervoor heeft gezorgd dat het cosmetische gedeelte van de applicatie zich niet meer kan meten met de hedendaagse applicaties.

2.2 Aanleiding van het project

Marti-IT, een jong en dynamisch bedrijf, gevestigd in Naaldwijk, is specialist op het gebied van orthopedische en bakkerij software. Momenteel zijn er ruim 450 bakkers en zo'n 75 orthopedisten die gebruik maken van de Marti-IT software.
Naast de bakkerij en orthopedische software besteedt de organisatie ook nog een deel van haar werkzaamheden aan maatwerk software en advisering.
Het bedrijf telt een zevental medewerkers, waaronder drie ontwikkelaars, drie helpdeskmedewerkers en één administratieve medewerkster.

Het Internet is een belangrijke factor in de Marti-IT software, aangezien er veelvuldig gebruik gemaakt wordt. Het communiceren van en naar de klanten, middels een speciaal ontwikkelde applicatie genaamd het Marti-IT Update Centrum stelt klanten op zeer eenvoudige wijze in staat om de nieuwste updates te downloaden voor de Marti-IT software. Daarnaast is het tevens mogelijk om met het Marti-IT Update Centrum gegevens te verzenden naar het bedrijf. Op die manier kunnen problemen altijd binnen no-time worden opgelost door de helpdesk.

2.3 Doelstelling

De doelstelling van de opdracht richt zich op het ontwikkelen van een nieuw Marti-IT Update Centrum, waarin met name de aspecten functionaliteit en een nieuw uiterlijk van de applicatie worden benadrukt.
Een ander belangrijk aspect van het nieuwe Update Centrum is dat het grotendeels geschreven moet worden in de .NET programmeertaal C#.
Tot slot dient de applicatie ervoor te zorgen dat sommige processen, zoals bijvoorbeeld het klaarzetten van een nieuwe update op het Internet, geautomatiseerd dienen te worden. Op deze manier kunnen dergelijke processen binnen de organisatie geoptimaliseerd worden.

© 2005 Mathieu Brüning (20010019)

2.4 Projectresultaat

Het project resulteert in een aantal (deel)producten, die zorg dragen voor het uiteindelijke eindresultaat, namelijk het opleveren van een volledig vernieuwd Marti-IT Update Centrum.

Hieronder volgt een lijst van de op te leveren producten gedurende het project:

- Plan van aanpak (inclusief gedetailleerde planning)
- Pilotplan
- Ontwerpen voor het ontwikkelen van de software-bouweenheden per pilot
- Testplannen voor het testen van de ontwikkelde software-bouweenheden per pilot
- Vernieuwd Marti-IT Update Centrum, bestaande uit:
    - Een nieuwe (gebruikersvriendelijke) interface
    - De functionaliteit van het bestaande Update Centrum herschreven in C#
    - Extra functionaliteiten met betrekking tot het automatiseren van processen e.d.
- Applicatiedocumentatie omtrent het nieuwe Marti-IT Update Centrum
- Afstudeerscriptie

2.5 Afbakening van het resultaat

Het resultaat met betrekking tot het project richt zich uiteindelijk alleen op de ontwikkeling van het vernieuwde Marti-IT Update Centrum. Eventuele latere aanpassingen aan de ontwikkelde applicatie behoren dan niet meer tot de verantwoordelijkheid van de student, maar liggen vervolgens volledig in de handen van de opdrachtgever.

2.6 Effecten van het project

Het project zal na uitvoeren van de verscheidene (deel)projecten effecten teweegbrengen.
Deze effecten van het project richten zich voornamelijk tot de ontwikkelen van de nieuwe applicatie. Het voornaamste effect met betrekking tot het project is de vernieuwing van de uiterlijk van de applicatie en het ontwikkelen van extra functionaliteiten, bovenop de reeds bestaande functionaliteiten van het Marti-IT Update Centrum.

2.7 Risicofactoren

Risico's zijn helaas niet uit te sluiten en zullen dus zeker ook in dit project aanwezig zijn. De meest waarschijnlijke risicofactoren ten opzichte van dit project worden hieronder uiteengezet.

- Op sommige fronten is het moeilijk om vooraf een begroting van de uren te maken voor bepaalde (deel)producten, daar de student sommige kennis of vaardigheden nog niet beheerst.
  Zodoende zou een afwijking van de planning wel eens voor kunnen komen, waardoor het tijdsbestek, betreffende het project, in het gedrang zou kunnen komen.

- Langdurige ziekte met betrekking tot de afstudeerder zou kunnen leiden tot een tekortkoming aan tijd om het beoogde resultaat te kunnen behalen.

© 2005 Mathieu Brüning (20010019)

2.8 Benodigde mensen en middelen

Tijdens het project zijn er verscheidene middelen en mensen nodig, enerzijds voor de verstrekking van informatie en anderzijds voor het verschaffen van hulp bij de totstandkoming van (deel)producten.

2.8.1 Benodigde mensen

De verstrekking van informatie zal grotendeels plaatsvinden bij de opdrachtgever / bedrijfsmentor (M.L. Verkade) binnen de organisatie. Voor overige informatie op het gebied van het ontwikkelen in C# kan dhr. A. Barendse worden geraadpleegd.

De verstrekking van feedback en overige informatie met betrekking tot de opleiding zal plaatsvinden bij de beoordelende docenten, dhr. J.L.A. Schramp en dhr. M. Reijnhoudt.

2.8.2 Benodigde software

De volgende softwarematige middelen zijn nodig om de totstandkoming van de (deel)producten te kunnen realiseren:

- Software m.b.t. de rapportage rondom het project (MS Office)
- Software met betrekking tot het ontwikkelen van technische ontwerpen (MS Visio)
- Software met betrekking tot het ontwikkelen in C# (MS Visual Studio 2003)
- Software met betrekking tot het ontwikkelen in HTML en CSS (MS Visual Studio 2003)
- Software met betrekking tot het (eventueel) grafisch ontwikkelen van afbeeldingen e.d. (Adobe Photoshop)

2.8.3 Benodigde hardware

De volgende hardwarematige middelen zijn nodig om de totstandkoming van de (deel)producten te kunnen realiseren:

- Een werkplek
- Een verbinding met het Internet

2.9 Gebruikte methode en technieken

Om het project succesvol laten te verlopen zal er gebruik gemaakt gaan worden van verscheidene technieken in combinatie met een ondersteunende methode.

2.9.1 Technieken

Technieken die de revue zullen passeren tijdens het project zijn hieronder opgesomd:

- Analyseren, voor het vergaren van informatie
- Rapporteren, voor het samenstellen van rapporten/verslagen
- Unified Modelling Language (UML), voor het ontwerpen van de software-bouweenheden
- Prototyping, voor de tussentijdse goedkeuring van software-bouweenheden
- Interviewen, voor het verzamelen van de eisen/wensen m.b.t. de opdrachtgever

2.9.2 Methode

De door mij gekozen methodiek voor het ontwikkelen/vernieuwen van de applicatie is Iterative Application Development.

© 2005 Mathieu Brüning (20010019)

2.10 Randvoorwaarden

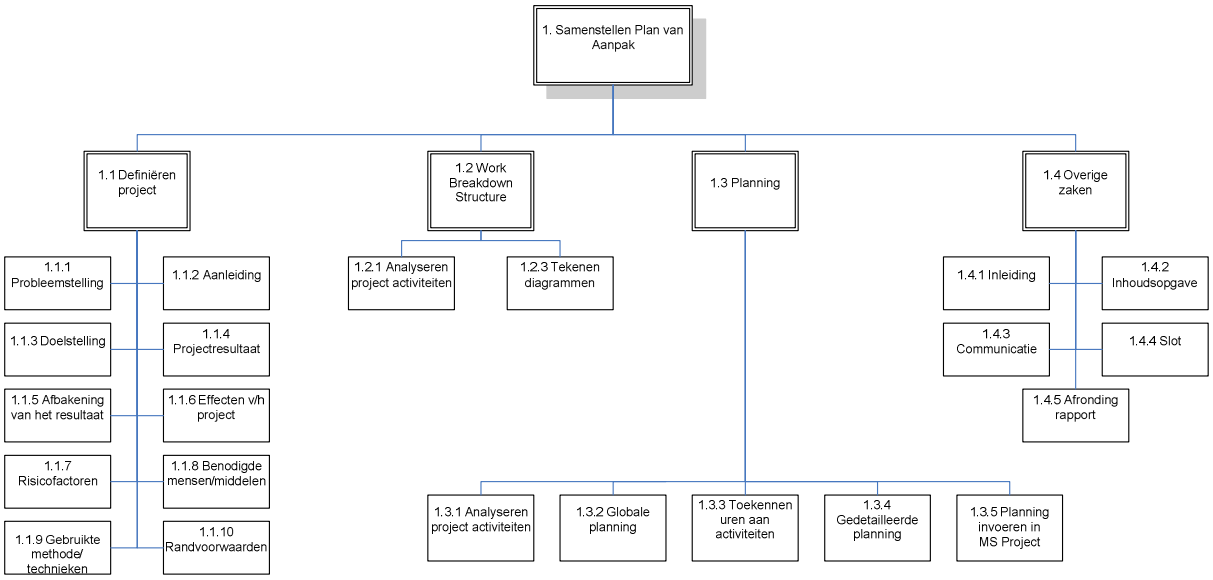
1. Het project betrekt een periode van 18 weken. In deze periode, beginnend vanaf 7 februari 2005 en eindigend op 10 juni 2005, zal het project moeten worden afgerond.

© 2005 Mathieu Brüning (20010019)

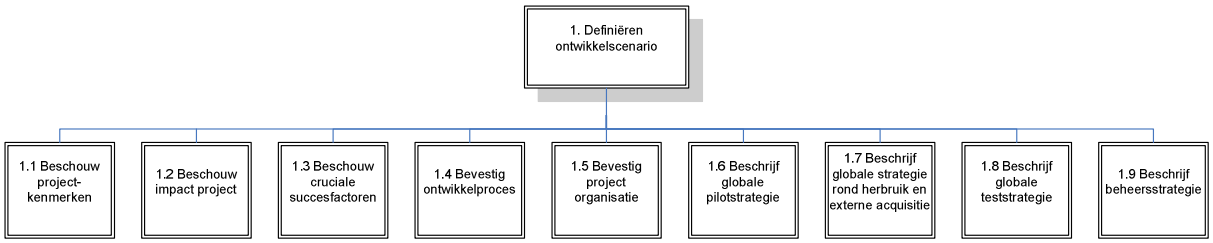
3 Fasering & Activiteitenplan

3.1 Work Breakdown Structure

3.1.1 Plan van aanpak

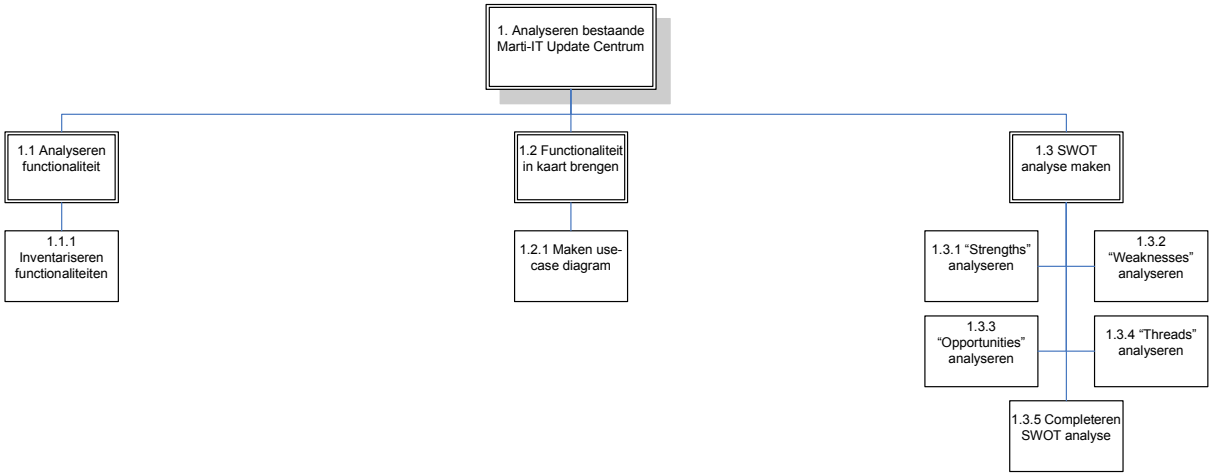


3.1.2 Definiëren ontwikkelscenario

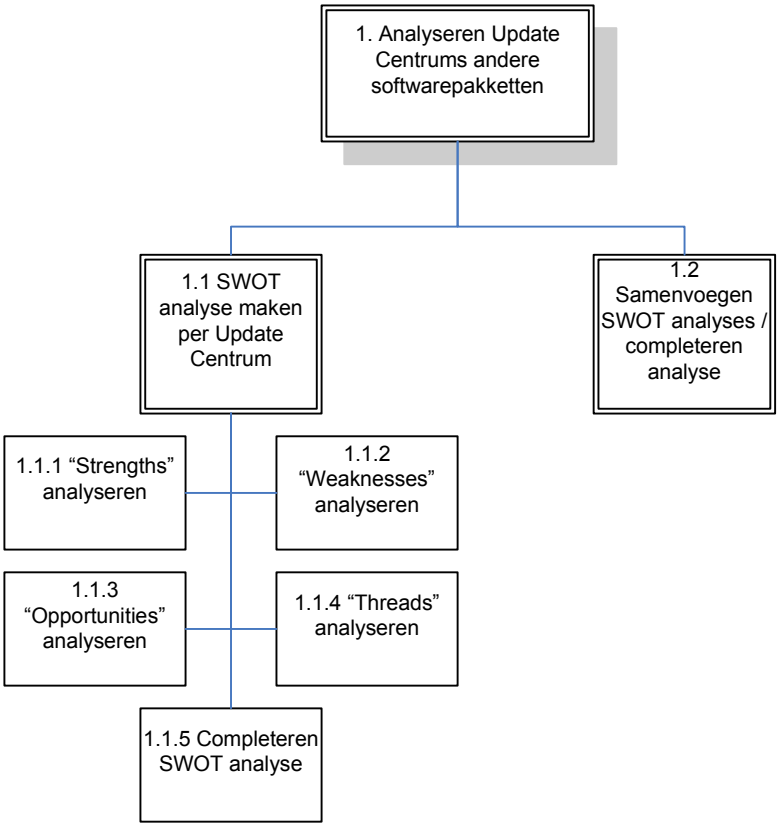


© 2005 Mathieu Brüning (20010019)

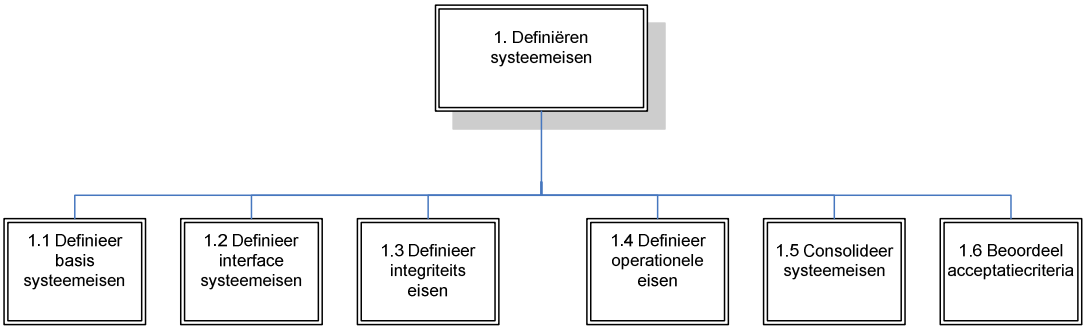
3.1.3 Analyseren bestaande Marti-IT Update Centrum



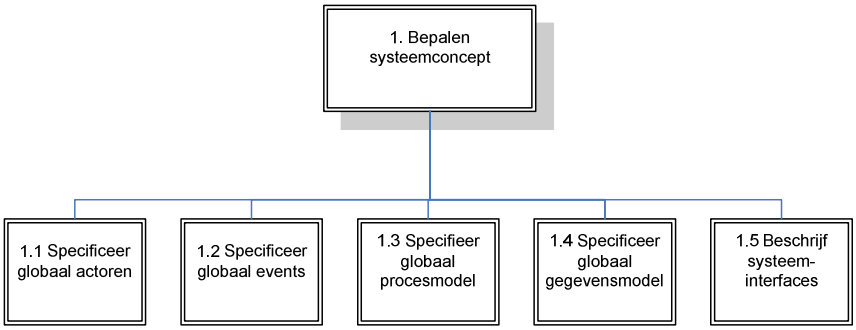
3.1.4 Analyseren Update Centrums van overige softwarepakketten



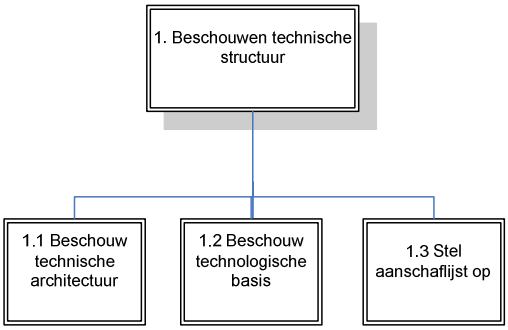
3.1.5 Definiëren systeemeisen



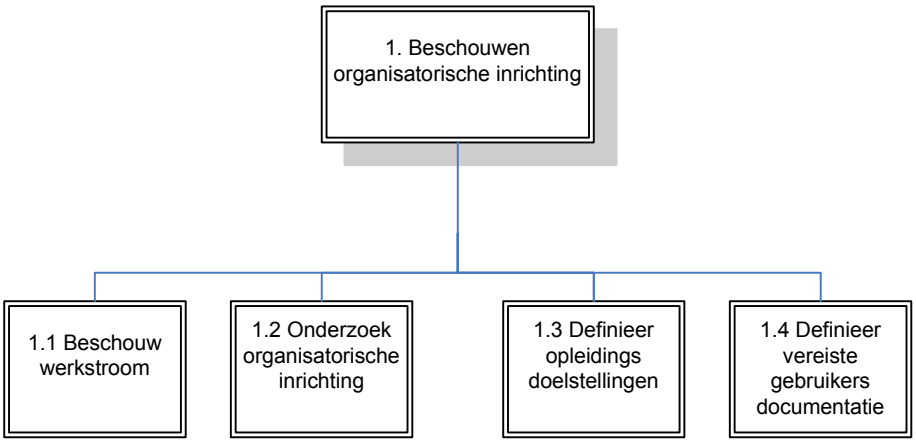
3.1.6 Bepalen systeemconcept



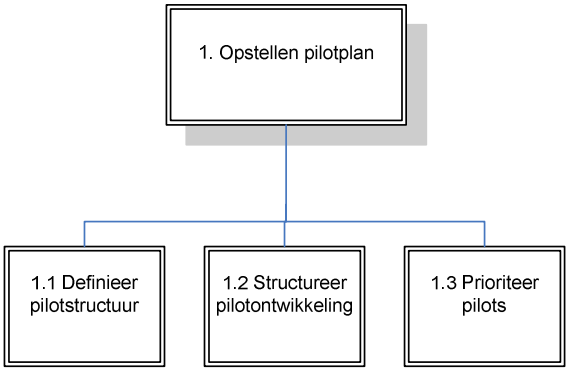
3.1.7 Beschouwen technische structuur



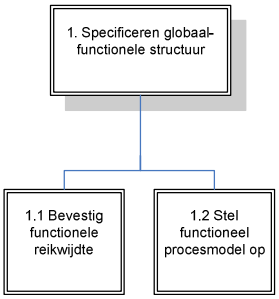
3.1.8 Beschouwen organisatorische inrichting



3.1.9 Opstellen pilotplan

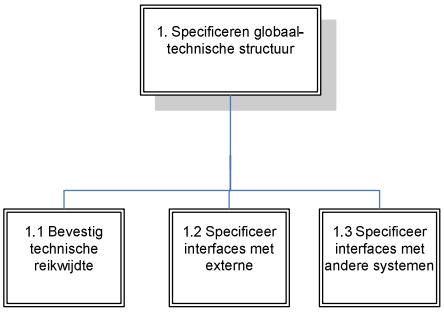


3.1.10 Specificeren globaal-functionele structuur

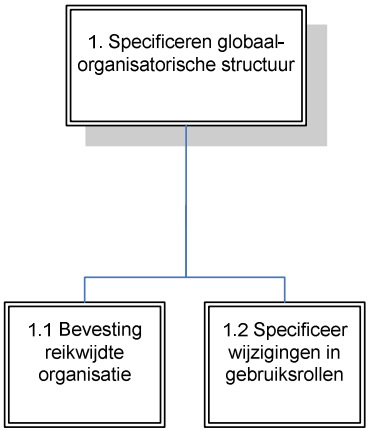


© 2005 Mathieu Brüning (20010019)

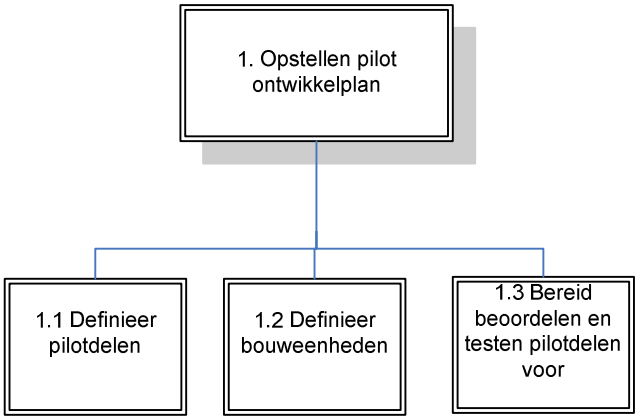
3.1.11 Specificeren globaal-technische structuur



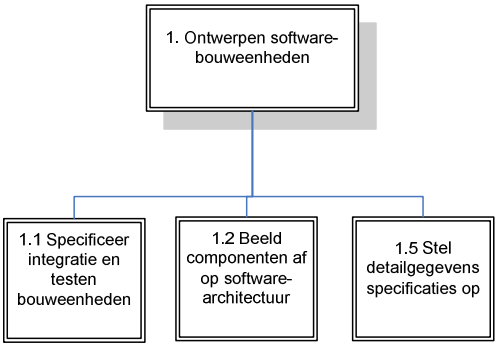
3.1.12 Specificeren globaal-organisatorische structuur



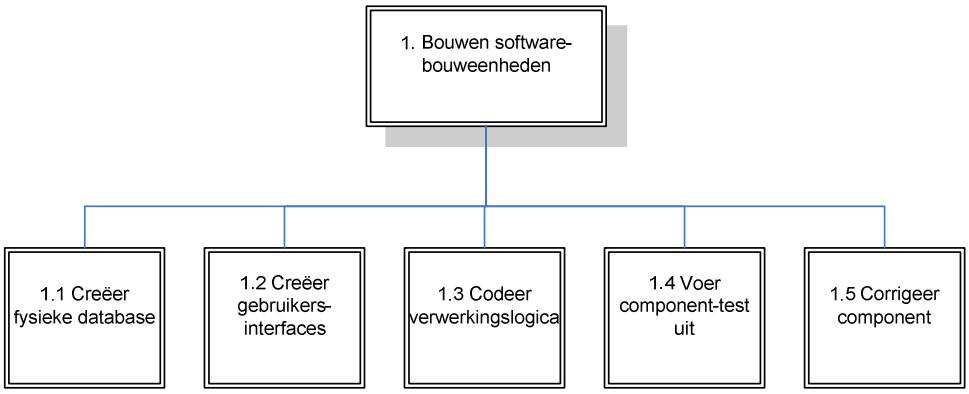
3.1.13 Opstellen pilotontwikkelplan



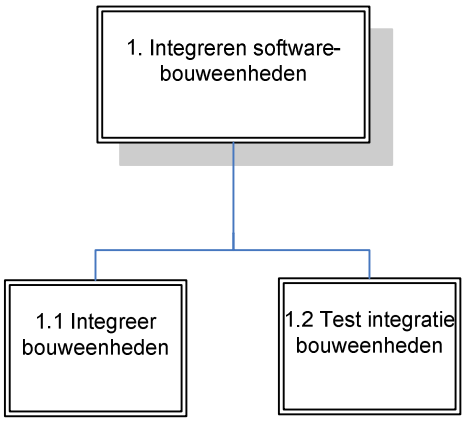
3.1.14 Ontwerpen software-bouweenheden



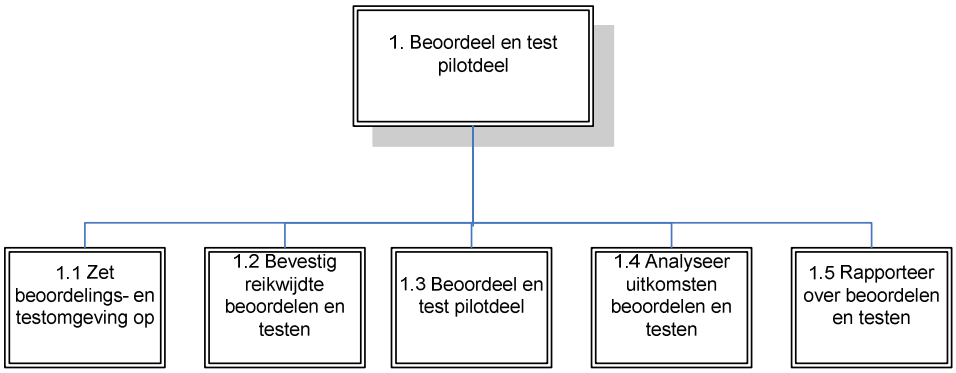
3.1.15 Bouwen software-bouweenheden



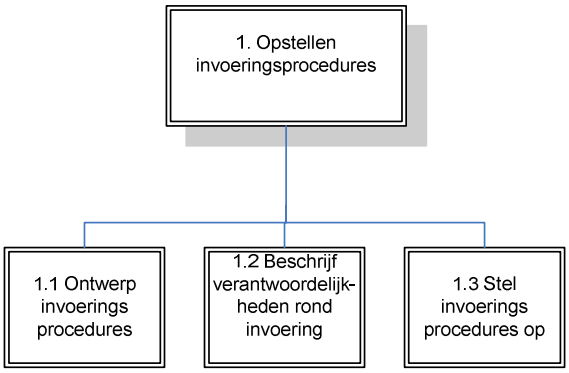
3.1.16 Integreren software-bouweenheden



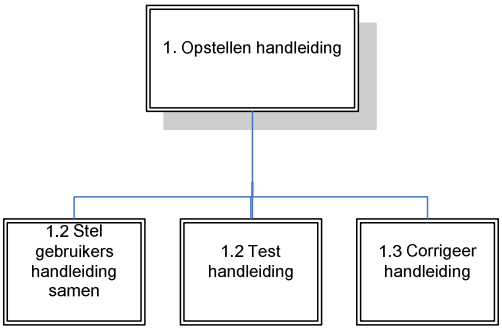
3.1.17 Beoordeel en test pilotdeel



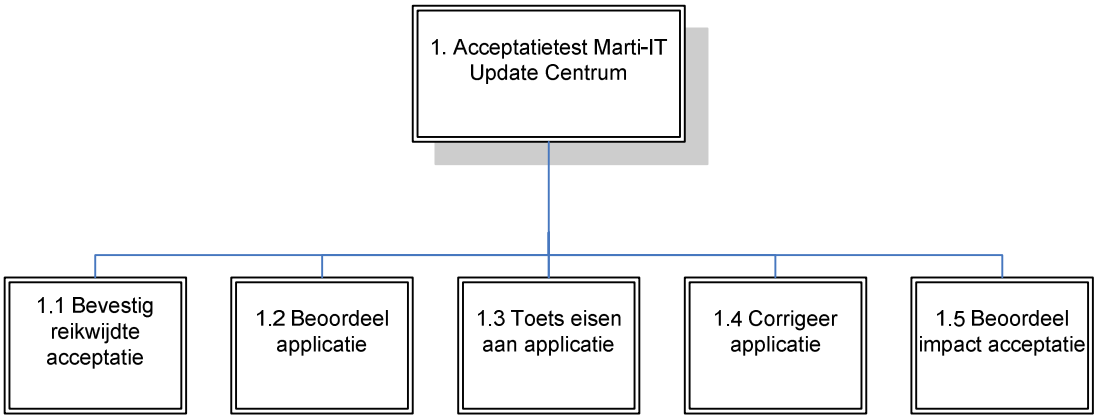
3.1.18 Opstellen invoeringsprocedures



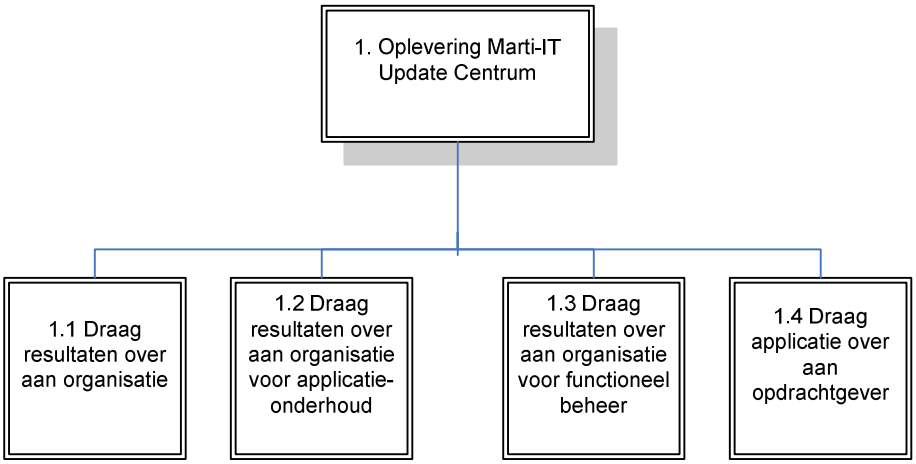
3.1.19 Opstellen handleiding



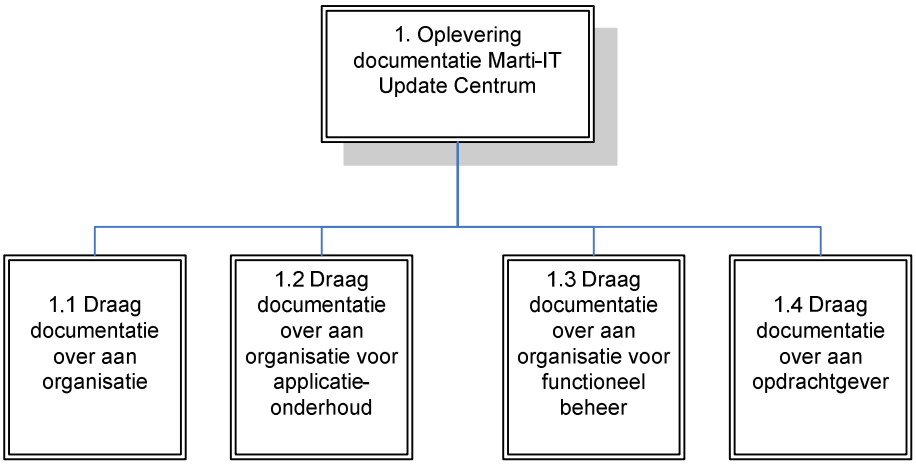
3.1.20 Acceptatietest Marti-IT Update Centrum



3.1.21 Oplevering Marti-IT Update Centrum

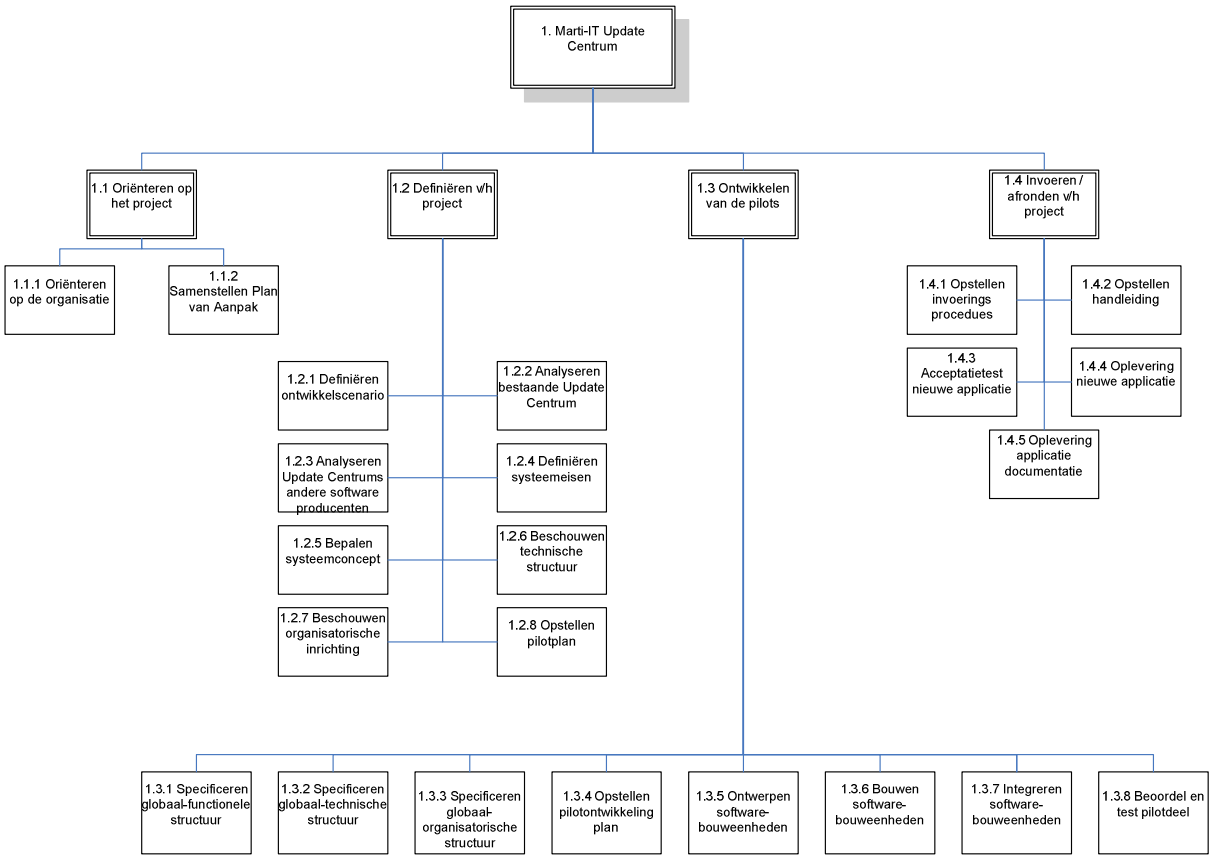


3.1.22 Oplevering documentatie Marti-IT Update Centrum

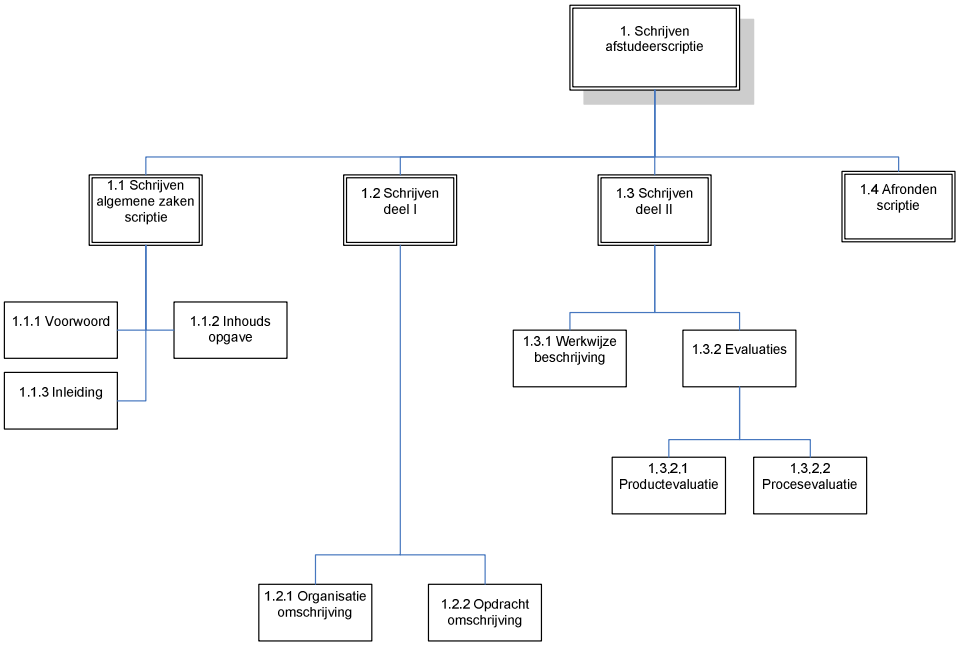


© 2005 Mathieu Brüning (20010019)

3.1.23 Marti-IT Update Centrum



3.1.24 Schrijven afstudeerscriptie



3.2 Beschrijving mijlpaalproducten, -activiteiten

3.2.1 Plan van aanpak

Het Plan van aanpak is een document dat vakkundig inzicht moet verschaffen over de werkzaamheden, resultaten en procedures in en rondom het project.
Dit document draagt zorg voor de basis van het project, daar namelijk alles betreffende het project wordt beschreven. De definitie van het project en een globale en een gedetailleerde planning vormen het hart van het plan van aanpak.

3.2.2 Definiëren ontwikkelscenario

Het definiëren van het ontwikkelscenario heeft als voornaamste doel om de iteratieve ontwikkelstrategie op de juiste manier te laten verlopen en de context van het project in kaart te brengen. Door het definiëren van het ontwikkelscenario en daarmee dus een scherp beeld te ontwikkelen van hetgeen dat er ontwikkeld dient te worden, is het voor alle betrokkenen duidelijk wat er van hen wordt verwacht.

3.2.3 Analyseren bestaande Marti-IT Update Centrum

Om het bestaande Marti-IT Update Centrum zo adequaat mogelijk in kaart te brengen, zodat de reeds bestaande functionaliteit van de applicatie gedocumenteerd kan worden, dient het huidige Update Centrum geanalyseerd te worden.
De verscheidene functionele aspecten van de bestaande applicatie kunnen dan vervolgens volledig worden uitgekristalliseerd, zodat er een helder beeld gevormd kan worden omtrent de minimale functionaliteit van het nieuwe Marti-IT Update Centrum.
De analyse van het bestaande Marti-IT Update Centrum dient er ook voor te zorgen dat de positieve aspecten en negatieve aspecten van de bestaande applicatie worden uiteengezet. Gebruikersparticipatie bij het analyseren van het huidige Marti-IT Update Centrum is niet uitgesloten.

3.2.4 Analyseren Update Centrums andere softwarepakketten

Wanneer het bestaande Marti-IT Update Centrum geanalyseerd is, kan er een duidelijk beeld gevormd worden rondom de sterke en zwakke punten rondom de applicatie. Door vervolgens Update Centrums te analyseren, welke afkomstige zijn van andere softwarepakketten kunnen er ideeën worden opgedaan rondom de ontwikkeling van het nieuwe Marti-IT Update Centrum.
Centraal bij deze activiteit staat dus het analyseren van de positieve en negatieve aspecten van andere Update Centrums om zo ideeën op te doen voor de ontwikkeling van het nieuwe Marti-IT Update Centrum.

3.2.5 Definiëren systeemeisen

Wanneer de analyse van het huidige Update Centrum is uitgevoerd, kunnen de systeemeisen worden gedefinieerd. Het voornaamste doel van deze activiteit is het opstellen en actualiseren van een geprioriteerde lijst van systeemeisen.
Middels het MoSCoW principe (prioriteren van eisen) kunnen de eisen en wensen van de opdrachtgever nauwkeurig worden bepaald. De eisen en wensen van de opdrachtgever, in combinatie met de analyseresultaten van het bestaande Marti-IT Update Centrum kunnen vervolgens als input dienen voor het definiëren van de systeemeisen.

Indien de lijst van geprioriteerde systeemeisen is geformuleerd, vormt deze de basis voor alle ontwerpactiviteiten.
Een belangrijk kenmerk van de systeemeisen is dat de systeemeisen gebruikt worden als referentiemateriaal in de rest van het traject, om de ontwikkelde producten te vergelijken met de vooraf gesteld systeemeisen.

© 2005 Mathieu Brüning (20010019)

3.2.6 Bepalen systeemconcept

Na het formuleren van de systeemeisen (welke zich feitelijk richten op een beschrijving van de probleemstelling) kan het systeemconcept bepaald worden. Het systeemconcept richt zich, in tegenstelling tot de systeemeisen, op een beschrijving van de oplossing. Met name de systeemeisen met een relatief hoge prioriteit worden beschreven in het systeemconcept.

Belangrijk kenmerk van het systeemconcept is dat het wordt gedefinieerd vanuit het oogpunt van de gebruiker. Functionele eisen, zoals de gebruiker deze stelt, dienen middels deze activiteit te worden vastgelegd.

3.2.7 Beschouwen technische structuur

Het beschouwen van de technische structuur is een activiteit die over het algemeen niet tot geheel nieuwe inzichten zal leiden op technisch gebied. De reden hiervoor is dat de voornaamste technische aspecten (hardware, software) al gedefinieerd zijn in het Plan van Aanpak.
Desalniettemin spelen de technische aspecten een niet geheel onbelangrijke rol. De technische aspecten dienen getoetst te worden aan de hand van het systeemconcept, zodat bepaald kan worden of het systeemconcept ingevuld kan worden met behulp van de huidige technische structuur.

3.2.8 Beschouwen organisatorische inrichting

Het beschouwen van de organisatorische inrichting speelt een belangrijke rol tijdens de pilotontwikkelingsfase. Het voornaamste doel van deze activiteit richt zich namelijk op het vaststellen van de impact op de organisatorische inrichting na het implementeren van een pilot. Daarnaast richt de activiteit zich eveneens op het voorspellen van de impact op de organisatorische inrichting voor een nog te ontwikkelen pilot.

3.2.9 Opstellen van een pilotplan

Het opstellen van een pilotplan omvat één van de belangrijkste activiteiten rondom de ontwikkeling van de pilots. Het plan dient namelijk te functioneren als besturingsorgaan voor de ontwikkeling van elke pilot, waarbij het pilotplan per pilot wordt aangepast om te kunnen gebruiken voor het besturen van de ontwikkeling van de volgende pilot.
Het pilotplan is opgebouwd uit een lijst van de te ontwikkelen pilots, geordend op prioriteit, waarbij elke pilot betrekking heeft op een component van het te ontwikkelen systeem. Het pilotplan kan dus gezien worden als de leidraad van het project, waarbij het plan gedurende de ontwikkeling wordt aangepast naar de ervaringen en reeds gemaakte ontwikkelingen van het project.
Belangrijke aspecten van het pilotplan, die niet vergeten mogen worden, zijn per pilot een schatting van benodigde middelen en tijd. Daarnaast dient er ook rekening gehouden te worden met de toekomstige beoordeling van de pilots en de planning voor het invoeren van de pilots.

3.2.10 Specificeren globaal-functionele structuur

Het specificeren van de globaal-functionele structuur per pilot richt zich hoofdzakelijk op het verfijnen van de functionele architectuur zoals vastgesteld in het systeemconcept. Per pilot kan dan nauwkeurig worden vastgesteld aan welke functionele eisen de pilot moet voldoen, door op een hoger niveau van detail in te gaan op de functionele eisen.

3.2.11 Specificeren globaal-technische structuur

De reeds gedefinieerde technische aspecten van het project (zoals vermeldt in het Plan van Aanpak) dienen middels deze activiteit gespecificeerd te worden op een hoger niveau van detail. Daarnaast dienen de technische aspecten gevalideerd te worden om te controleren of de reeds gedefinieerde technische aspecten voldoen aan de opgestelde functionele en organisatorische eisen.
Belangrijk is dat de mate van detail op een dusdanig niveau is geformuleerd, dat de afstudeerder een uitgangspunt heeft voor het ontwikkelen van de betreffende pilot.

© 2005 Mathieu Brüning (20010019)

3.2.12 Specificeren globaal-organisatorische structuur

Het specificeren van de globaal-organisatorische structuur van een pilot richt zich op het specificeren en valideren van de organisatorische aspecten rondom de pilot. Belangrijk is dat er aan de hand van de opgestelde functionele en technische specificaties wordt vastgelegd in hoeverre de pilot succesvol in de organisatie kan worden geïmplementeerd.

3.2.13 Opstellen van een pilotontwikkelingplan

Het opstellen van een pilotontwikkelplan is een activiteit die voor elke pilot uitgevoerd dient te worden. Het doel van de activiteit is het opstellen op van pilotontwikkelplan, welke gebruikt zal worden voor de ontwikkeling van een pilot. Aan de hand van de functionele, technische en organisatorische kenmerken van de pilot kan er op een hoog niveau van detail bepaald worden hoe de pilot ontwikkeld zal gaan worden.
Een belangrijk kenmerk van het opstellen van een pilotontwikkelplan is het indelen van een pilot in verschillende delen. De verschillende delen van een pilot worden vervolgens toegekend aan duidelijk afgebakende software-bouweenheden, zodat de ontwikkelaar de pilot gestructureerd en succesvol kan ontwikkelen.

3.2.14 Ontwerpen van de software-bouweenheden

De software-bouweenheden, waaruit de delen van een pilot zijn opgebouwd, dienen middels deze activiteit ontworpen te worden. De ontwerpen van de software-bouweenheden kunnen vervolgens gebruikt worden voor de feitelijk bouw van de software-bouweenheden. Daarnaast kunnen de producten van de ontwerpfasen in een later stadium gebruikt worden voor het onderhouden en/of aanpassen van de software-bouweenheden.
Belangrijk is dat er bij het ontwerpen van de software-bouweenheden sterk rekening gehouden wordt met de functionele en technische specificaties, zoals vastgesteld op een eerder moment in het project.

3.2.15 Bouwen van de software-bouweenheden

Het bouwen van de software-bouweenheden richt zich op de feitelijke totstandkoming van het uiteindelijke product. De totstandkoming van de pilots vind dan plaats doordat de verschillende software-bouweenheden van een pilot worden ontwikkeld.
Het feitelijke bouwproces kan over het algemeen beperkt blijven tot het coderen van de software-bouweenheden, aangezien de ontwerpen van de software-bouweenheden vooraf gevormd zijn op een hoog niveau van detail.
Een belangrijk aspect van deze activiteit, naast het feitelijke ontwikkelen van de software-bouweenheden is het testen van de ontwikkelde functionaliteiten. Op die manier kan het ontwikkelde deelproduct vergeleken worden met de vooraf gestelde eisen aan het deelproduct. Eventuele aanpassing aan het ontwikkelde kan dan nog plaatsvinden.

3.2.16 Het integreren van de ontwikkelde bouweenheden

Het integreren van de ontwikkelde bouweenheden komt feitelijk neer op het samenvoegen van de reeds ontwikkelde componenten met de nieuwe ontwikkelde pilot of pilotdeel. Wanneer de ontwikkelde bouweenheden samen worden gevoegd worden er twee doelen nagestreefd, namelijk:

- zorgen dat de ontwikkelde software-bouweenheid op de juiste manier samenhangt met de overige eenheden die deel uitmaken van de pilot
- bouweenheden optimaliseren waar dat nodig blijkt te zijn op het gebied van beveiliging- en/of prestatie-eisen ten behoeve van de overige delen van de pilot

Het beoogde resultaat van deze activiteit is dus dat de onderdelen van een pilot, na voltooiing van de integratieprocedure, op de juiste manier met elkaar samenwerken.
Het integreren van de software-bouweenheden zal over het algemeen een continue proces zijn, aangezien het integreren van software-bouweenheden veelal plaats vind tijdens het ontwikkelingsproces.

© 2005 Mathieu Brüning (20010019)

3.2.17 Beoordeel en testpilotdeel

Het opstellen van testplannen voor de ontwikkelde bouweenheden dient plaats te vinden om testwerkzaamheden uit te kunnen voeren op pilotdelen. Door testsets te definiëren die in overeenstemming zijn met de vooraf vastgestelde functionele en technische specificaties, kunnen de ontwikkelde pilotdelen op adequate wijze getest worden.

Nadat een testplan geformuleerd is voor het betreffende pilotdeel kunnen de testwerkzaamheden plaatsvinden.

Op die manier kunnen de verschillende pilotdelen worden beoordeeld of ze voldoen aan de vooraf gesteld eisen, zoals vastgesteld in de functionele en technische eisen.

3.2.18 Opstellen van invoeringsprocedures Marti-IT Update Centrum

Het invoeren van het Marti-IT Update Centrum zal bestaan uit het zogenaamde 'big bang' invoeren van de gehele applicatie in de fase na de pilotontwikkeling. In tegenstelling tot de gebruikelijke manier van invoeren wanneer IAD wordt toegepast, het invoeren van een pilot na de ontwikkeling van een pilot (IAD-ontwikkelstrategie 'Evolutionair ontwikkelen), wordt de gehele applicatie ingevoerd wanneer deze volledig ontwikkeld is (IAD-ontwikkelstrategie 'incrementeel ontwikkelen').

Voor het invoeren van de applicatie is het erg belangrijk dat het invoeringsprocedures zijn opgesteld, zodat het operationeel maken van de applicatie op een succesvolle manier kan verlopen. Het opstellen van dergelijke procedures dient ervoor te zorgen dat alle betrokkene weten wat er van hen verwacht wordt tijdens de invoering, zodat het invoeringsproces optimaal kan verlopen.

Naast het opstellen van strikte procedures rondom het invoeren van de applicatie, is het ook erg belangrijk dat er rekening gehouden worden met eventuele noodsituaties. Het eventuele falen van de invoering zorgt er dan in ieder geval niet voor dat de continuïteit in het gedrang komt.

3.2.19 Opstellen van handleidingen

Het opstellen van handleidingen is een activiteit die wordt uitgevoerd met twee doelen, namelijk: het opstellen van een handleiding voor de gebruikers van de nieuwe applicatie en het opstellen van een handleiding voor degene die de applicatie in de toekomst dienen te beheren en ondersteunen.

De eerste handleiding wordt omschreven als de gebruikershandleiding. Het opstellen van een gebruikershandleiding heeft als doel om de toekomstige gebruikers van de applicatie informatie te verschaffen omtrent de werking en functionaliteit.
De tweede handleiding is met name belangrijk voor toekomstige activiteiten als bijvoorbeeld ondersteuning en beheer rondom de applicatie. In deze handleiding moet nauwkeurig gedocumenteerd staan hoe de structuur van de applicatie is en hoe de applicatie tot stand is gekomen.

3.2.20 Acceptatietest met betrekking tot de opdrachtgever

De acceptatietest met betrekking tot de opdrachtgever is een activiteit die zich hoofdzakelijk richt op een test, waaruit duidelijk moet worden of de ontwikkelde applicatie voldoet aan de vooraf gestelde eisen systeemeisen en functionele specificaties.

3.2.21 Oplevering van Marti-IT Update Centrum aan de opdrachtgever

Het overdragen van de resultaten rondom het project staat centraal tijdens de oplevering van het Marti-IT Update Centrum. Met name de nieuwe applicatie wordt in deze activiteit overgedragen aan de opdrachtgever.
Wanneer de opdrachtgever dan heeft laten merken dat de ontwikkelde applicatie naar tevredenheid is overgedragen, kan de overdracht worden geaccepteerd en worden afgerond.

© 2005 Mathieu Brüning (20010019)

3.2.22 Oplevering van de applicatie documentatie aan de opdrachtgever

Naast de ontwikkelde applicatie dient ook de documentatie rondom het ontwikkelde opgeleverd te worden aan de opdrachtgever. Het overdragen van de documentatie rondom de ontwikkeling van de applicatie is van essentieel belang voor toekomstige ondersteuning en beheer van de nieuwe applicatie.

Belangrijke documentatie met betrekking tot de applicatie omvat onder andere de functionele- en technische ontwerpen van de pilots.

3.2.23 Marti-IT Update Centrum

Het nieuwe Marti-IT Update Centrum is het uiteindelijke beoogde eindresultaat wat betreft het afstudeertraject.

Het huidige Marti-IT Update Centrum is op het gebied van gebruikersvriendelijkheid nog niet helemaal optimaal. Door een nieuw uiterlijk en extra functionaliteiten toe te kennen aan het Update Centrum dient de gebruikersvriendelijkheid verbeterd te worden.
De huidige functionaliteit van het Update Centrum is feitelijk in orde, maar er is echter de behoefte naar extra functionaliteiten, zoals bijvoorbeeld op het gebied van commandline opties en .zip mogelijkheden.

De ontwikkeling van het nieuwe Marti-IT Update Centrum richt zich zodoende op twee gebieden, namelijk: enerzijds dient de gebruikersvriendelijkheid van de applicatie verbeterd te worden, anderzijds dient de applicatie extra functionaliteiten te bevatten.

De ontwikkeling van het programma kan opgesplitst worden in de volgende onderdelen:

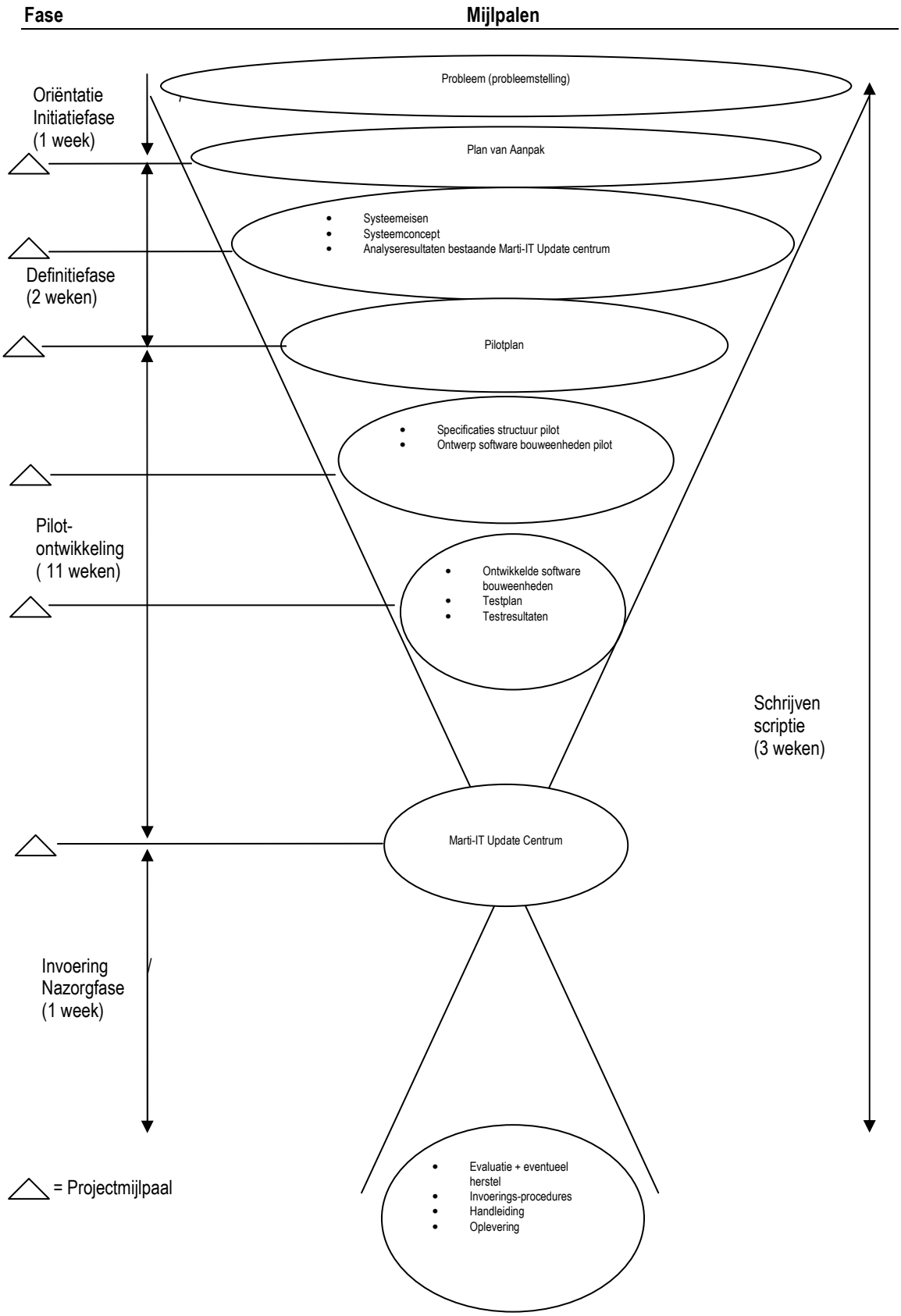
- ✓ Ontwikkeling van een vernieuwde interface
- ✓ Herschrijven van de bestaande functionaliteit in C#
- ✓ Ontwikkelen van extra functionaliteiten

3.2.24 Schrijven afstudeerscriptie

De afstudeerscriptie berust op twee achterliggende gedachten, namelijk: ten eerste zorgt de scriptie ervoor dat de student goed op een rij zet welke leerwinst er door het project behaald is. Vervolgens verschaft het verslag een inzicht in de werkzaamheden tijdens het project. Dit is voornamelijk bedoeld voor de mentor en examinator.
De afstudeerscriptie speelt een grote rol ten opzichte van het project, daar de examinator het verslag gebruikt voor de beoordeling van het project.

© 2005 Mathieu Brüning (20010019)

3.3 Globale planning



4 Tijdbeheersing

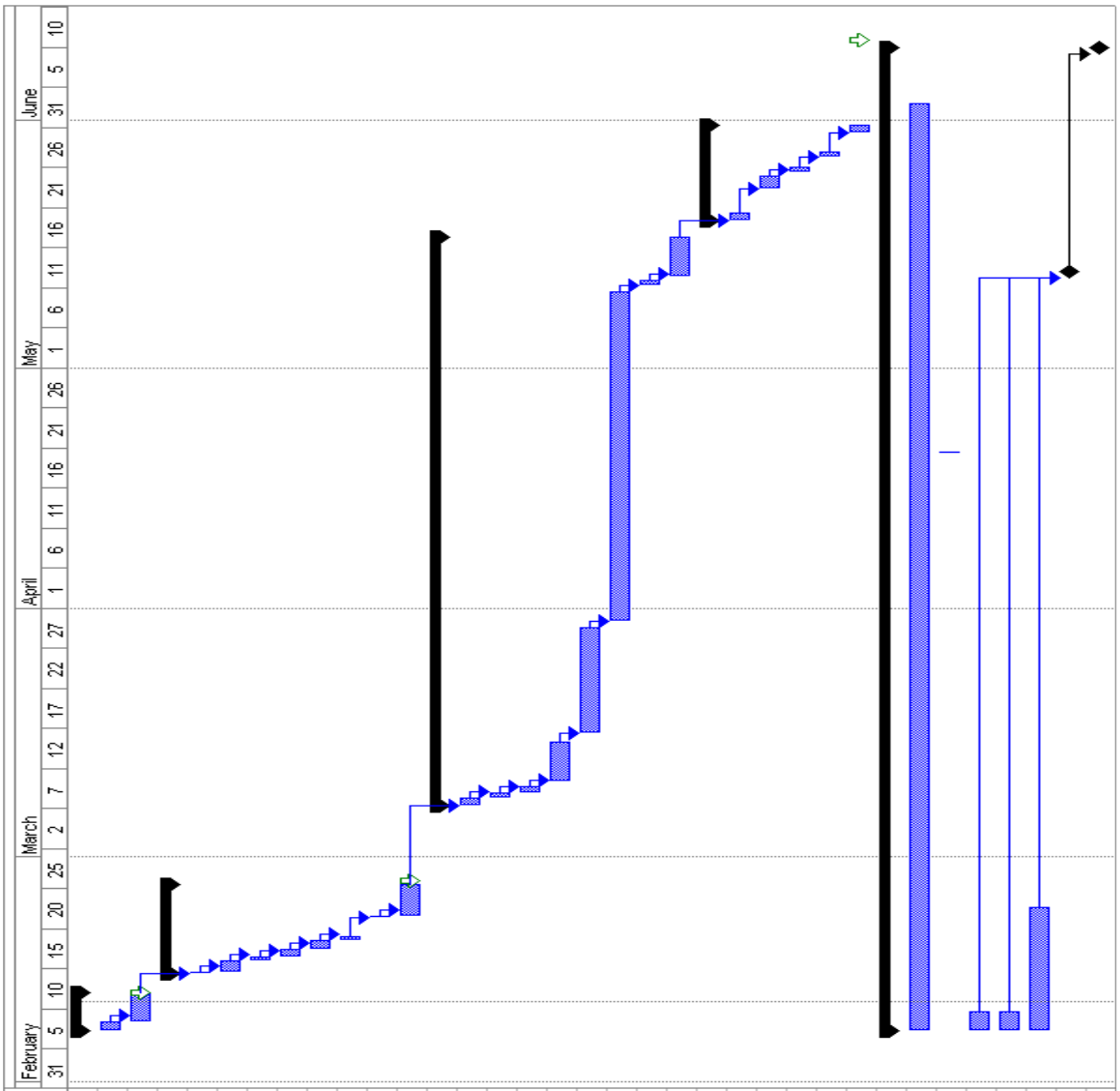
Het volgende hoofdstuk richt zich uitsluitend op de tijdbeheersing met betrekking tot het project. Een gedetailleerde planning is dan uiteraard onmisbaar en wordt zodoende weergegeven in paragraaf 4.1

4.1 Gedetailleerde planning

Zie volgende pagina.

© 2005 Mathieu Brüning (20010019)

ID	Task Name	Duration	Start
1	**ORIËNTATIEFASE**	**5 days**	**Mon 7-2-05**
2	Oriënteren op de organisatie	1 day	Mon 7-2-05
3	Samenstellen Plan van Aanpak	4 days	Tue 8-2-05
4	**DEFINITIEFASE**	**9,63 days?**	**Mon 14-2-05**
5	Definiëren ontwikkelscenario	4 hrs	Mon 14-2-05
6	Analyseren bestaande Marti-IT Update Centrum	1,5 days	Mon 14-2-05
7	Analyseren Update Centrums andere softwarepakketten	4 hrs	Tue 15-2-05
8	Definiëren systeemeisen	1 day?	Wed 16-2-05
9	Bepalen systeemconcept	1 day?	Thu 17-2-05
10	Beschouwen technische structuur	4 hrs	Fri 18-2-05
11	Beschouwen organisatorische inrichting	4 hrs	Mon 21-2-05
12	Opstellen pilotplan	4 days	Mon 21-2-05
13	**PILOTONTWIKKELING**	**52,31 days**	**Mon 7-3-05**
14	Specificeren globaal-functionele structuur	1 day	Mon 7-3-05
15	Specificeren globaal-technische structuur	1 day	Tue 8-3-05
16	Specificeren globaal-organisatorische structuur	1 day	Tue 8-3-05
17	Opstellen pilotontwikkelingplan	3 days	Thu 10-3-05
18	Ontwerpen software-bouweenheden	10 days	Wed 16-3-05
19	Bouwen software-bouweenheden	30 days	Wed 30-3-05
20	Integreren van de ontwikkelde bouweenheden	1 day	Wed 11-5-05
21	Test en beoordeel pilotdeel	3 days	Thu 12-5-05
22	**INVOERING / NAZORG FASE**	**8,06 days**	**Thu 19-5-05**
23	Opstellen invoeringsprocedures	1 day	Thu 19-5-05
24	Opstellen handleiding(en)	2 days	Mon 23-5-05
25	Acceptatietest	1 day	Wed 25-5-05
26	Oplevering Marti-IT Update Centrum aan opdrachtgever	1 day	Fri 27-5-05
27	Oplevering applicatie documentatie aan opdrachtgever	1 day	Mon 30-5-05
28	**AFSTUDEERRAPPORTAGE**	**91,13 days?**	**Mon 7-2-05**
29	Bijhouden urenverantwoording	85,88 days?	Mon 7-2-05
30	Bedrijfsbezoek contactdocent	4 hrs	Wed 20-4-05
31	Schrijven scriptie deel I	3 days	Mon 7-2-05
32	Schrijven scriptie deel II	3 days	Mon 7-2-05
33	Schrijven scriptie deel II	12 days	Mon 7-2-05
34	Conceptverslag	0 days	Fri 13-5-05
35	Definitieve scriptie	0 days	Fri 10-6-05



5 Communicatie

5.1 Communicatie in en rondom het project

Communicatie tussen student en opdrachtgever:
De opdrachtgever/bedrijfsmentor draagt zorg voor het verschaffen van informatie tijdens het project. Op deze manier kan de afstudeerder de opdrachtgever raadplegen in het geval van eventuele vragen.
Deze vorm van communicatie is dus heel belangrijk voor het verloop van het project voor het inhoudelijke deel van het project.

Communicatie tussen student en contactdocent:
Het contact met de contact/beoordelend docent vind hoofdzakelijk plaats via e-mail. Eventuele vragen en/of feedback kan via e-mail worden gepleegd.

5.2 Communicatiemedia en –middelen

Gedurende het project worden er verscheidene communicatiemedia en –middelen ingezet om communicatie te kunnen laten verlopen.

Hieronder bevindt zich een opsomming waaruit blijkt welke media tijdens dit project gebruikt zullen gaan worden:

Persoonlijke media:

- Tweegesprek (communicatie over ondervonden problemen of het stellen van eventuele vragen aan opdrachtgever)
- Groepsgesprek (vergaderingen, intern overleg)

Auditieve media:

- Telefoon (voor het stellen van eventuele vragen aan opdrachtgever)

Schriftelijke media:

- E-mail (voor het stellen van eventuele vragen aan opdrachtgever)
- Verslag / rapport (rapportages naar zowel opdrachtgever als contactdocent)
- Boek (ter behoeve van het verzamelen van informatie)

Audiovisuele media:

- Internetsites (ter behoeve van het verzamelen van informatie)

© 2005 Mathieu Brüning (20010019)

Bijlage 3: Resultaten definitiestudie

Het uitvoeren van de definitiefase in het project resulteerde in een rapport met de specificaties van het te ontwikkelen systeem. Aangezien het rapport naar mijn idee vrij uitgebreid is heb ik de resultaten als externe bijlage toegevoegd. Deze externe bijlage heeft de naam “Specificaties Marti-IT Update / Build Centrum”.

© 2005 Mathieu Brüning (20010019)

Bijlage 4: Pilotontwikkelplannen

PILOTONTWIKKELPLAN PILOT 1

Aanhef:

Module:	Afstuderen	**Opdrachtgever:**	Dhr. M.L. Verkade
Docent(en):	Dhr. J.L.A. Schramp Dhr. M. Reijnhoudt	**Bedrijfsmentor:**	Dhr. M.L. Verkade
		Plaats:	Naaldwijk
Organisatie:	Marti-IT	**Datum:**	7 maart 2005

Student:

Naam:	**E-Mail:**
Brüning, Mathieu	20010019

© 2005 Mathieu Brüning (20010019)

Inhoudsopgave

1 INLEIDING 129

2 GLOBAAL-FUNCTIONELE STRUCTUUR 130

3 GLOBAAL-TECHNISCHE STRUCTUUR 131

4 GLOBAAL-ORGANISATORISCHE STRUCTUUR 132

5 PILOTSTRUCTUUR 133

5.1 BASIS INTERFACE 133

5.2 AUTORISATIE FUNCTIONALITEIT 133

5.3 OPZETTEN EN INRICHTEN DATABASE 133

5.4 DOWNLOADEN VAN GEGEVENS 134

5.5 UPLOADEN VAN GEGEVENS 134

5.6 BAKE-IT SPECIFIEKE FUNCTIONALITEITEN 135

5.7 ORTO VOOR WINDOWS SPECIFIEKE FUNCTIONALITEITEN 135

6 TEST- EN BEOORDELINGSPLAN 136

6.1 TESTPROCEDURE 136

6.2 TESTCRITERIA 136

1 Inleiding

Inmiddels zijn de oriëntatie- en definitiefase van het afstudeertraject afgerond en is de pilotontwikkelingsfase van het project van start gegaan. Het toepassen van de Iterative Application Development methodiek voor de ontwikkeling van het nieuwe systeem zorgt ervoor dat de pilotontwikkelingsfase is opgesplitst in drie verschillende pilots, zoals gedefinieerd in het pilotplan, waarbij de ontwikkeling van elke pilot zich richt op de ontwikkeling van een specifiek deel van het nieuwe systeem. Voor de ontwikkeling van elke pilot wordt er zodoende een pilotontwikkelplan opgesteld, een plan waarin duidelijk uiteengezet staat wat de structuur is van een pilot en hoe de ontwikkeling van de pilot zal verlopen.

Het doel van dit document is het vastleggen van de structuur van de eerste pilot in de pilotontwikkelingsfase en het vastleggen van de wijze waarop de ontwikkeling zal plaatsvinden. Daarnaast dient dit document inzicht te verschaffen omtrent de globale functionele, technische en organisatorische aspecten van de eerste pilot. Tot slot richt dit pilotontwikkelplan zich op het vastleggen van de diverse aspecten rondom het testen van de ontwikkelde pilot.

Het pilotontwikkelplan is als volgt ingedeeld: de hoofdstukken twee, drie en vier bevatten beschrijvingen van de globale functionele, technische en organisatorische structuur van de eerste pilot. In deze hoofdstukken wordt globaal uiteengezet aan welke functionele, technische en organisatorische kenmerken de eerste pilot moet voldoen. Hoofdstuk vijf richt zich op de structuur van de te ontwikkelen pilot. In eerste instantie wordt in dit hoofdstuk een globale indeling van de pilot gemaakt in pilotdelen, waarop vervolgens concreet wordt gespecificeerd welke activiteiten er uitgevoerd dienen te worden. Het laatste hoofdstuk in dit pilotplan richt zich op het vastleggen van procedures en criteria voor het testen van de ontwikkelde pilot.

© 2005 Mathieu Brüning (20010019)

2 Globaal-functionele structuur

De eerste pilot in de pilotontwikkelingsfase richt zich hoofdzakelijk op het herschrijven van de functionaliteit van het bestaande Marti-IT Update Centrum. Hierbij is het van essentieel belang dat de volledige functionaliteit van het huidige systeem geïmplementeerd wordt in het nieuwe systeem, zodat er geen sprake is van het verlies van functionaliteit. Het herschrijven van de bestaande functionaliteit richt zich globaal op het herontwikkelen van de volgende functionaliteiten:

- ✓ Het downloaden van updates, persoonlijke bestanden, e.d.
- ✓ Het uploaden van back-ups, bestanden en/of mappen en standaard gegevens

De bovenstaande opsomming geeft een globaal functioneel overzicht weer van de te ontwikkelen functionaliteiten in deze pilot. In het onderstaande overzicht wordt uiteengezet wat de functionaliteiten zijn van het huidige systeem, welke dus minimaal ontwikkeld dienen te worden in het nieuwe systeem.

- ✓ Downloaden programma
- ✓ Downloaden rapporten
- ✓ Downloaden documenten
- ✓ Downloaden helpbestand
- ✓ Downloaden hulp bestanden
- ✓ Downloaden persoonlijke bestanden
- ✓ Downloaden persoonlijke rapporten
- ✓ Downloaden persoonlijke licentie
- ✓ Inventariseren van (alle) beschikbare downloads
- ✓ Downloaden (alle) beschikbare updates
- ✓ Uploaden backups
- ✓ Uploaden bestand(en) / folder(s)
- ✓ Uploaden standaard uploads
- ✓ Downloaden grondstof bestanden
- ✓ Downloaden voorbeeld recepten
- ✓ Downloaden voorbeeld etiketten
- ✓ Downloaden TopBakkers prijslijsten
- ✓ Downloaden Orto prijslijsten schoenmakerijen
- ✓ Downloaden Orto prijslijsten instrumentmakerijen
- ✓ Downloaden Orto prijslijsten bandagisten

De doelstelling met betrekking tot de ontwikkeling van de eerste pilot richt zich op het ontwikkelen van functionaliteiten, zodat de bovenstaande functionaliteiten uit het oude systeem geïmplementeerd zijn in het nieuwe systeem.

Voor een gedetailleerd overzicht van de use-case scenario's met betrekking tot de functionele structuur verwijs ik u naar het rapport met de specificaties van het nieuwe systeem.

© 2005 Mathieu Brüning (20010019)

3 Globaal-technische structuur

De globaal-technische structuur van de eerste pilot in de pilotontwikkelingsfase richt zich met name op het opzetten en inrichten van een database en het herontwikkelen van de bestaande functionaliteit middels de .NET programmeertaal C#.
De technische architectuur van de eerste pilot richt zich zodoende hoofdzakelijk op het ontwikkelen van een client-server situatie, waarin een gebruiker een verzoek doet aan een database op de server, welke vervolgens informatie vrijgeeft omtrent beschikbare updates e.d. De gebruiker kan op basis van deze informatie vervolgens de update bestanden downloaden van de Marti-IT server.

Voor het herontwikkelen van de bestaande functionaliteiten is er geen koppeling nodig met andere reeds bestaande systemen. Wel dient de database op de Marti-IT Update Centrum server geplaatst te worden en dienen er gegevens als update bestanden e.d. op de server geplaatst te worden.

© 2005 Mathieu Brüning (20010019)

4 Globaal-organisatorische structuur

De eerste pilot in de pilotontwikkelingsfase richt zich qua globaal-organisatorische structuur met name op de gebruikers van de Marti-IT software. De ontwikkeling van de eerste pilot kenmerkt zich namelijk door het ontwikkelen van de bestaande functionaliteit van het Marti-IT Update Centrum, de functionaliteit die gebruikt wordt door de gebruikers van de Marti-IT software voor het verzenden en ontvangen van gegevens.

De actoren met betrekking tot het de eerste pilot zijn hoofdzakelijk in te delen in drie groepen, namelijk:

- ✓ de gebruikers die de algemene functionaliteiten van het systeem gebruiken
- ✓ de gebruikers die met name de specifieke functionaliteiten met betrekking tot de Bake-it applicatie gebruiken
- ✓ de gebruikers die met name de specifieke functionaliteiten met betrekking tot de Orto applicatie gebruiken.

Samenvattend kan dus gesteld worden dat de ontwikkeling van de eerste pilot zich hoofdzakelijk richt op het ontwikkelen van functionaliteiten voor de gebruikers van de Marti-IT software. Hierbij is echter een splitsing gemaakt naar de verschillende gebruikersgroepen van de Marti-IT software.

© 2005 Mathieu Brüning (20010019)

5 Pilotstructuur

In het volgende hoofdstuk wordt de structuur van de eerste pilot uiteengezet. De paragrafen in dit hoofdstuk representeren de pilotdelen, waarin wordt beschreven uit welke concrete activiteiten de ontwikkeling van het betreffende pilotdeel zal bestaan.

5.1 Basis interface

De ontwikkeling van een basis interface dient ervoor te zorgen dat toekomstige functionaliteiten getest kunnen worden door deze simpelweg aan te roepen vanuit deze basis interface.

De doelstelling met betrekking tot het ontwikkeling van een basis interface richt zich op het ontwikkelen en implementeren van een hoofdvenster en een (statisch) menu, zodat functionaliteiten aangeroepen kunnen worden.

5.2 Autorisatie functionaliteit

Een autorisatie functionaliteit dient ervoor te zorgen dat gebruikers zich kunnen identificeren, zodat er op basis van informatie uit een database vastgelegd kan worden om welke gebruiker het gaat en welke software op deze gebruiker van toepassing is.

De doelstellingen met betrekking tot de autorisatie functionaliteit zijn:

- ✓ Een venster met velden voor het ingeven van een gebruikersnummer en een wachtwoord;
- ✓ De mogelijkheid tot het autoriseren van een gebruiker aan de hand van een gebruikersnummer en wachtwoord middels een database;
- ✓ De mogelijkheid dat de Order-it afnemers (klanten van Marti-IT gebruikers zonder gebruikersnummer en wachtwoord) zich ook kunnen autoriseren;
- ✓ De mogelijkheid tot het opslaan van het gebruikersnummer en wachtwoord.

Het ontwikkelen van de autorisatie functionaliteit bestaat uit de volgende activiteiten:

- ✓ Het ontwikkelen van een venster voor de autorisatie functionaliteit
- ✓ Het ontwerpen van de autorisatie functionaliteit
- ✓ Het programmeren van de autorisatie functionaliteit
- ✓ Het koppelen van het reeds ontwikkelde venster aan de functionaliteit

5.3 Opzetten en inrichten database

Het opzetten en inrichten van een database is een belangrijke stap in de ontwikkeling van de eerste pilot. Een database dient namelijk het hart te van het nieuwe systeem te worden, aangezien alle informatie omtrent gebruikers, updates, enz. hierin wordt opgeslagen.

Het opzetten en inrichten van een database richt zich op de volgende activiteiten:

- ✓ Het selecteren van een bruikbare database voor het nieuwe systeem
- ✓ Het verder uitwerken van het globale gegevensmodel tot een volledig model voor de structuur van de database
- ✓ Het creëren van de structuur van de database
- ✓ Het invoeren van testgegevens in de database
- ✓ Het online plaatsen van de database op de Marti-IT server

© 2005 Mathieu Brüning (20010019)

5.4 Downloaden van gegevens

Het ontwikkelen van de functionaliteit voor het downloaden van gegevens middels het systeem is een belangrijk pilotdeel tijdens de ontwikkeling van de eerste pilot. Daar de gebruikers van de Marti-IT veelvuldig gebruik maken van de download functionaliteit is het belangrijk dat deze functionaliteit op de juiste manier functioneert in het nieuwe systeem.

De doelstellingen met betrekking tot de download functionaliteit zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van programma updates
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van rapporten
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van documenten
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van updates voor helpbestanden
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van hulp bestanden
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van persoonlijke bestanden
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van persoonlijke rapporten
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van een persoonlijke licentie

Het ontwikkelen van de functionaliteit voor het downloaden van gegevens richt zich op:

- ✓ Het ontwerpen van de download functionaliteit
- ✓ Het programmeren van de download functionaliteit
- ✓ Het ontwikkelen van vensters voor het downloaden van gegevens
- ✓ Het koppelen van de vensters aan de reeds ontwikkelde download functionaliteit

5.5 Uploaden van gegevens

Naast het downloaden van gegevens voor de Marti-IT software is het uploaden van gegevens ook een belangrijke functionaliteit van het systeem. Met name voor de ondersteuning van de software wordt er veelvuldig gebruik gemaakt van de upload functionaliteiten. Ook hier geldt weer dat de mogelijkheden voor het uploaden van gegevens minimaal gelijk moeten zijn aan de mogelijkheden in het bestaande systeem.

Het ontwikkelen van de upload functionaliteiten richt zich hoofdzakelijk op:

- ✓ Het ontwerpen van de upload functionaliteit
- ✓ Het programmeren van de upload functionaliteit
- ✓ Het ontwikkelen van venster voor het uploaden van gegevens
- ✓ Het koppelen van de vensters aan de reeds ontwikkelde upload functionaliteit

De doelstellingen met betrekking tot de ontwikkeling van de upload functionaliteit zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het uploaden van software back-ups
- ✓ Het ontwikkelen van een functionaliteit voor het uploaden van bestanden en/of mappen
- ✓ Het ontwikkelen van een functionaliteit voor het uploaden van standaard gegevens.

© 2005 Mathieu Brüning (20010019)

5.6 Bake-it specifieke functionaliteiten

Naast de algemene functionaliteiten van het bestaande Marti-IT Update Centrum kent het bestaande systeem een aantal applicatie specifieke functionaliteiten. Zo zijn er een aantal mogelijkheden tot het downloaden van applicatie specifieke gegevens welke eveneens geïmplementeerd dienen te worden in het nieuwe systeem. Het ontwikkelen van de specifieke functionaliteiten voor de Bake-it applicatie staan in deze paragraaf centraal.

Het ontwikkelen van de Bake-it specifieke functionaliteiten kenmerkt zich door:

- ✓ Het ontwerpen van de functionaliteiten voor het downloaden van de Bake-it specifieke gegevens
- ✓ Het programmeren van de functionaliteiten voor het downloaden van de Bake-it specifieke gegevens
- ✓ Het koppelen van de functionaliteiten aan de reeds ontwikkelde vensters voor het downloaden van gegevens

De doelstellingen met betrekking tot de Bake-it specifieke functionaliteiten zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van grondstof bestanden
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van voorbeeld recepten
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van voorbeeld etiketten
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van TopBakkers prijslijsten

5.7 Orto voor Windows specifieke functionaliteiten

Net als de specifieke functionaliteiten voor de gebruikers van de Bake-it software zijn er ook specifieke functionaliteiten voor het downloaden van gegevens voor de Orto voor Windows software. Deze functionaliteiten voor het downloaden van gegevens voor de Orto voor Windows software staan centraal in deze paragraaf.

De ontwikkeling van de functionaliteiten voor het downloaden van specifieke gegevens voor de Orto voor Windows software richt zich op:

- ✓ Het ontwerpen van de functionaliteiten voor het downloaden van de Orto voor Windows specifieke gegevens
- ✓ Het programmeren van de functionaliteiten voor het downloaden van de Orto voor Windows specifieke gegevens
- ✓ Het koppelen van de functionaliteiten aan de reeds ontwikkelde vensters voor het downloaden van gegevens

De doelstellingen met betrekking tot de Orto voor Windows specifieke functionaliteiten zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van prijslijsten schoenmakerijen
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van prijslijsten instrumentmakerijen
- ✓ Het ontwikkelen van een functionaliteit voor het downloaden van prijslijsten bandagisten

© 2005 Mathieu Brüning (20010019)

6 Test- en beoordelingsplan

Nadat de ontwikkeling van de eerste pilot in de pilotontwikkelingsfase is gecompleteerd dienen er testwerkzaamheden plaats te vinden om te beoordelen of de ontwikkelde pilotdelen wel in overeenstemming zijn met hetgeen wat er vooraf vastgesteld is. Het vastleggen van de procedure en criteria rondom het testen van de ontwikkelingen uit de eerste pilot staat daarom centraal in dit hoofdstuk.

6.1 Testprocedure

De testprocedure na het ontwikkelen van de pilotdelen in de eerste pilot richt zich hoofdzakelijk op het vergelijken van de functionaliteit van het bestaande systeem met de functionaliteit van het nieuwe systeem. Aangezien er hier met name sprake is van een functionele vergelijking van de systemen zal er feitelijk alleen sprake zijn van Black Box tests. Black Box tests richten zich namelijk op het uitvoeren van tests op basis van functionele specificaties. Aangezien er in de definitiestudie is geanalyseerd en geïnventariseerd welke functionele specificaties behoren tot het bestaande systeem is het mogelijk om de functionaliteiten van het bestaande systeem te vergelijken met de ontwikkelde functionaliteiten in de eerste pilot. De doelstelling met betrekking tot deze Black Box tests zijn dan duidelijk, namelijk: de functionaliteiten van het nieuwe ontwikkelde systeem dienen overeen te komen met de functionaliteiten van het bestaande systeem.

Het toepassen van Black Box tests richt zich zodoende op het toepassen van hoofdzakelijk een real life test en (indien er nog tijd is) een dataflow test. Het toepassen van een real life test heeft als voornaamste voordeel dat het systeem wordt gebruikt zoals het na implementatie ook zo worden gebruikt. Op deze manier kan een representatie van het operationele gebruik van het systeem worden nagebootst en kan worden gekeken of het systeem daadwerkelijk kan functioneren na implementatie. Het toepassen van een dataflow test dient plaats te vinden om vast te stellen of de verwerking door functies en de relaties tussen functies correct is.

De procedure rondom het testen zal er als volgt uit komen te zien:

1. Opstellen testplan
2. Uitvoeren tests volgens testplan
3. Verslaglegging resultaten testplan
4. Beoordelen pilot a.d.h.v. resultaten testplan

6.2 Testcriteria

De testcriteria met betrekking tot de testwerkzaamheden in de eerste pilot zijn hoofdzakelijk gebaseerd op de functionaliteiten van het bestaande systeem. Deze criteria, zoals opgesomd in hoofdstuk twee van dit pilotontwikkelplan, dienen gebruikt te worden voor de toetsing van de ontwikkelde functionaliteiten in de eerste pilot. Op deze manier kan er op eenvoudige wijze vastgesteld worden of de ontwikkelingen in de eerste pilot overeenkomen met de vooraf gestelde eisen aan de ontwikkeling van de eerste pilot.

Samenvattend kan ik stellen dat alle functionaliteiten van het bestaande Marti-IT Update Centrum, zoals geïnventariseerd in de definitiestudie, ontwikkeld dienen te zijn tijdens de ontwikkeling van de eerste pilot, zodat de functionaliteit van het ontwikkelde systeem uit de eerste pilot minimaal gelijk is aan de functionaliteit van het bestaande systeem.

© 2005 Mathieu Brüning (20010019)

PILOTONTWIKKELPLAN PILOT 2

Aanhef:

Module:	Afstuderen	**Opdrachtgever:**	Dhr. M.L. Verkade
Docent(en):	Dhr. J.L.A. Schramp Dhr. M. Reijnhoudt	**Bedrijfsmentor:**	Dhr. M.L. Verkade
		Plaats:	Naaldwijk
Organisatie:	Marti-IT	**Datum:**	25 april 2004

Student:

Naam:	**E-Mail:**
Brüning, Mathieu	20010019

© 2005 Mathieu Brüning (20010019)

Inhoudsopgave

1 INLEIDING 139

2 GLOBAAL-FUNCTIONELE STRUCTUUR 140

3 GLOBAAL-TECHNISCHE STRUCTUUR 141

4 GLOBAAL-ORGANISATORISCHE STRUCTUUR 142

5 PILOTSTRUCTUUR 143

5.1 FUNCTIONALITEIT M.B.T. HET BEHEREN VAN GEGEVENS OMTRENT GEBRUIKERS 143

5.2 FUNCTIONALITEIT M.B.T. HET BEHEREN VAN GEGEVENS OMTRENT SOFTWARE 143

5.3 FUNCTIONALITEIT M.B.T. HET BEHEREN VAN GEGEVENS OMTRENT UPDATES 144

5.4 FUNCTIONALITEIT M.B.T. HET AUTOMATISEREN EN BEHEREN VAN VERZONDEN GEBRUIKERSGEGEVENS 144

5.5 EXTRA FUNCTIONALITEITEN M.B.T. DE GEBRUIKERS VAN DE MARTI-IT SOFTWARE 145

5.5.1 Maken van back-ups 145

5.5.2 Downloaden van historische updates 145

5.5.3 Ophalen van updates vanaf lokale en/of netwerklocatie 146

5.6 EXTRA FUNCTIONALITEITEN OP HET GEBIED VAN .ZIP 146

6 TEST- EN BEOORDELINGSPLAN 147

6.1 TESTPROCEDURE 147

6.2 TESTCRITERIA 147

© 2005 Mathieu Brüning (20010019)

1 Inleiding

Na de ontwikkeling en afronding van de eerste pilot richten mijn werkzaamheden zich op de volgende pilot in de pilotontwikkelingsfase. Voor de ontwikkeling van de tweede pilot wordt er opnieuw een pilotontwikkelplan opgesteld, zodat duidelijk er is wat de structuur van de pilot is en hoe de ontwikkeling van de pilot zal verlopen.

Het doel van dit document is het vastleggen van de structuur van de tweede pilot in de pilotontwikkelingsfase en het vastleggen van de wijze waarop de ontwikkeling zal plaatsvinden. Daarnaast dient dit document inzicht te verschaffen omtrent de globale functionele, technische en organisatorische aspecten van de tweede pilot. Tot slot richt dit pilotontwikkelplan zich op het vastleggen van de diverse aspecten rondom het testen van de ontwikkelde pilot.

Het pilotontwikkelplan is als volgt ingedeeld: de hoofdstukken twee, drie en vier bevatten beschrijvingen van de globale functionele, technische en organisatorische structuur van de tweede pilot. In deze hoofdstukken wordt globaal uiteengezet aan welke functionele, technische en organisatorische kenmerken de eerste pilot moet voldoen. Hoofdstuk vijf richt zich op de structuur van de te ontwikkelen pilot. In eerste instantie wordt in dit hoofdstuk een globale indeling van de pilot gemaakt in pilotdelen, waarop vervolgens concreet wordt gespecificeerd welke activiteiten er uitgevoerd dienen te worden. Het laatste hoofdstuk in dit pilotplan richt zich op het vastleggen van procedures en criteria voor het testen van de ontwikkelde pilot.

© 2005 Mathieu Brüning (20010019)

2 Globaal-functionele structuur

Waar de eerste pilot zich hoofdzakelijk richtte op de functionaliteiten voor de gebruikers van de Marti-IT software, richt de tweede pilot zich voornamelijk op functionaliteiten voor de medewerkers van Marti-IT. Het ontwikkelen van extra functionaliteiten bovenop de functionaliteit van het bestaande Marti-IT Update Centrum dient ervoor te zorgen dat bepaalde processen binnen de organisatie kunnen worden geoptimaliseerd. Daarnaast is het beheer van de gegevens voor het Marti-IT Update Centrum een essentieel onderdeel voor de ontwikkeling van de tweede pilot. Echter, naast het ontwikkelen van functionaliteiten voor de medewerkers van Marti-IT richt de tweede pilot zich ook op extra functionaliteiten voor het Update Centrum voor de gebruikers van de Marti-IT software. Onderstaand worden de functionaliteiten per gebruikersgroep opgesomd, welke de revue zullen passeren tijdens de ontwikkeling van de tweede pilot.

Functionaliteiten voor de medewerkers van Marti-IT:

- ✓ Gegevens vastleggen omtrent beschikbare updates
- ✓ Vastleggen van gegevens van gebruikers
- ✓ Wijzigen gegevens softwaregebruik gebruikers
- ✓ Klaarzetten van programma updates
- ✓ Extra functionaliteiten op het gebied van .zip
- ✓ Automatisch downloaden van back-ups naar Marti-IT lokale server

Functionaliteiten voor de gebruikers van de Marti-IT software:

- ✓ Maken van een back-up
- ✓ Downloaden historische updates
- ✓ Updates ophalen vanaf lokale en/of netwerklocaties in plaats van het Internet

© 2005 Mathieu Brüning (20010019)

3 Globaal-technische structuur

De ontwikkeling in de tweede pilot richt zich hoofdzakelijk op het ontwikkelen van extra functionaliteiten, welke grotendeels gebruikt zullen gaan worden door de medewerkers van Marti-IT. Aangezien de functionaliteiten met betrekking tot de Marti-IT medewerkers en de functionaliteiten met betrekking tot de gebruikers van de Marti-IT software zeer verschillend zijn richt de tweede pilot zich op de ontwikkeling van een aparte applicatie, het zogenaamde Marti-IT Build Centrum. Afgezien van het ontwikkelen van extra functionaliteiten voor het systeem welke deels ontwikkeld is in de eerste pilot, richt de tweede pilot zich dus op de ontwikkeling van een nieuwe individuele applicatie.

Kenmerkend voor het Marti-IT Build Centrum is dat er een koppeling tussen het bestaande systeem binnen Marti-IT voor het vastleggen van gegevens omtrent klanten en softwaregebruik, genaamd Serv-it, gerealiseerd dient te worden. Net als het Marti-IT Update Centrum maakt het Build Centrum gebruik van een client-server architectuur, waar de server de database bevat en daarnaast de updates bevat, welke benaderd en beheert kunnen worden via het Marti-IT Build Centrum.

© 2005 Mathieu Brüning (20010019)

4 Globaal-organisatorische structuur

Voornamelijk richt de ontwikkeling van de tweede pilot zich op het ontwikkelen van functionaliteiten voor de medewerkers van Marti-IT. De belangrijkste aspecten van de nieuwe functionaliteiten zijn de mogelijkheid tot het beheren van de database gegevens en het optimaliseren van processen binnen de organisatie. Naast de ontwikkeling van deze functionaliteiten richten de ontwikkelingen in de tweede pilot zich ook voor een klein deel op extra functionaliteiten voor de gebruikers van de Marti-IT software.

© 2005 Mathieu Brüning (20010019)

5 Pilotstructuur

5.1 Functionaliteit m.b.t. het beheren van gegevens omtrent gebruikers

De functionaliteit voor het beheren van de gegevens van de gebruikers van de Marti-IT software is een essentieel onderdeel voor het beheer van het nieuwe Marti-IT Update Centrum. Deze functionaliteit draagt in de toekomst namelijk zorg voor het vastleggen van het softwaregebruik van de gebruikers, zodat elke gebruiker aan een softwareprofiel gekoppeld kan worden om alleen de updates weer te geven die van toepassing zijn op deze gebruiker.
Naast de functionaliteit voor het beheren van de gegevens omtrent het softwaregebruik van de gebruikers dient er een koppeling gerealiseerd te worden met de bestaande Marti-IT applicatie "Serv-it", een applicatie voor het vastleggen van gegevens omtrent gebruikers. Middels deze koppeling dient het in de toekomst mogelijk te zijn om de gegevens in de database voor het Update Centrum te synchroniseren met de gegevens uit Serv-it, zodat eventuele veranderingen met betrekking tot gegevens van gebruikers ook worden doorgevoerd voor het gebruik van het nieuwe Marti-IT Update Centrum.

De doelstellingen met betrekking tot de functionaliteit voor het beheren van de gebruikers gegevens zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het beheren van de gebruikersgegevens
- ✓ Het ontwikkelen van een functionaliteit voor het synchroniseren van de gebruikersgegevens met het bestaande systeem Serv-it

De werkzaamheden met betrekking tot het ontwikkelen van deze functionaliteit richten zich op:

- ✓ Het ontwerpen van de functionaliteit voor het beheren van de gebruikersgegevens
- ✓ Het programmeren van de functionaliteit voor het beheren van de gebruikersgegevens
- ✓ Het ontwikkelen van vensters voor het selecteren van gebruikers en het inzien van gebruikersgegevens
- ✓ Het koppelen van de functionaliteit aan de ontwikkelde vensters

5.2 Functionaliteit m.b.t. het beheren van gegevens omtrent software

Naast de functionaliteit voor het beheren van de gebruikersgegevens dient er eveneens een functionaliteit te zijn voor het beheren van de softwaregegevens voor het gebruik van het Update Centrum. Hiermee wordt met name bedoeld dat gegevens omtrent de Marti-IT applicaties opgeslagen kunnen worden, zodat deze gekoppeld kunnen worden aan gebruikers en aan updates.
Naast het beheren van gegevens omtrent software, dient er ook een functionaliteit gerealiseerd te worden voor het beheren van de download types, welke gekoppeld zijn aan de applicaties voor het onderscheiden van soorten updates.

De doelstellingen met betrekking tot de functionaliteit voor het beheren van gegevens omtrent de Marti-IT software zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het beheren van de gegevens omtrent de Marti-IT applicaties
- ✓ Het ontwikkelen van een functionaliteit voor het beheren van de gegevens omtrent download types

De werkzaamheden met betrekking tot het ontwikkelen van deze functionaliteit zijn:

- ✓ Het ontwerpen van de functionaliteit voor het beheren van de gegevens omtrent de Marti-IT applicaties
- ✓ Het ontwerpen van de functionaliteit voor het beheren van de gegevens omtrent de download types
- ✓ Het programmeren van de functionaliteit voor het beheren van de gegevens omtrent de Marti-IT applicaties
- ✓ Het programmeren van de functionaliteit voor het beheren van de gegevens omtrent de download types
- ✓ Het ontwikkelen van een venster voor het selecteren van applicaties
- ✓ Het ontwikkelen van een venster voor het inzien, toevoegen, wijzigen en opslaan van applicatie gegevens
- ✓ Het ontwikkelen van een venster voor het selecteren van download types
- ✓ Het ontwikkelen van een venster voor het inzien, toevoegen, wijzigen en opslaan van download types
- ✓ Het koppelen van de ontwikkelde functionaliteiten aan de ontwikkelde vensters

© 2005 Mathieu Brüning (20010019)

5.3 Functionaliteit m.b.t. het beheren van gegevens omtrent updates

Een belangrijke functionaliteit in het nieuwe systeem richt zich op het beheren van de gegevens omtrent de beschikbare updates op het Marti-IT Update Centrum. Aangezien de informatie omtrent de updates in het nieuwe systeem vastgelegd wordt in een database, is het van essentieel belang dat er een functionaliteit ontwikkeld wordt voor het beheren van deze informatie, zodat de medewerkers van Marti-IT op relatief eenvoudige wijze het nieuwe Update Centrum kunnen beheren.

De doelstellingen met betrekking tot het ontwikkelen van de functionaliteit voor het beheren van de gegevens omtrent de updates zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het beheren van gegevens omtrent updates in de database
- ✓ Het ontwikkelen van een functionaliteit voor het beheren van update bestanden op de Marti-IT server

De werkzaamheden met betrekking tot het ontwikkelen van deze functionaliteit richten zich op:

- ✓ Het ontwerpen van de functionaliteit voor het beheren van de gegevens omtrent de updates
- ✓ Het programmeren van de functionaliteit voor het beheren van de gegevens omtrent de updates
- ✓ Het ontwikkelen van een venster voor het selecteren van de beschikbare updates
- ✓ Het ontwikkelen van een venster voor het inzien, toevoegen, wijzigen en opslaan van gegevens omtrent updates
- ✓ Het programmeren van de functionaliteit voor het beheren van update bestanden op de Marti-IT server
- ✓ Het koppelen van de ontwikkelde functionaliteiten aan de ontwikkelde vensters

5.4 Functionaliteit m.b.t. het automatiseren en beheren van verzonden gebruikersgegevens

Het beheren van de gegevens die gebruikers verzenden naar Marti-IT middels het Update Centrum is veelal een tijdsintensieve routinetaak, aangezien dit beheerproces constant bestaat uit dezelfde handelingen, namelijk het handmatig downloaden en archiveren van de gegevens.
Een belangrijke functionaliteit met betrekking tot de medewerkers van Marti-IT richt zich zodoende op het automatiseren van het beheer van de verzonden gegevens, zodat deze tijdsintensieve routinetaak tot het verleden behoort.

De doelstellingen met betrekking tot het ontwikkelen van deze functionaliteit zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het automatisch downloaden van verzonden gegevens
- ✓ Het ontwikkelen van een functionaliteit voor het archiveren van de verzonden gegevens

De werkzaamheden met betrekking tot het ontwikkelen van deze functionaliteit zijn:

- ✓ Het ontwerpen van een functionaliteit voor het automatisch downloaden van verzonden gegevens
- ✓ Het programmeren van een functionaliteit voor het automatisch downloaden van verzonden gegevens
- ✓ Het ontwerpen van een functionaliteit voor het automatisch archiveren van verzonden gegevens
- ✓ Het programmeren van een functionaliteit voor het automatisch archiveren van verzonden gegevens
- ✓ Het koppelen van de ontwikkelde functionaliteiten aan het reeds ontwikkelde hoofdvenster van het Marti-IT Build Centrum

© 2005 Mathieu Brüning (20010019)

5.5 Extra functionaliteiten m.b.t. de gebruikers van de Marti-IT software

De ontwikkeling van de tweede pilot richt zich hoofdzakelijk op het ontwikkelen van functionaliteiten, welke gebruikt zullen worden door de medewerkers van Marti-IT. Echter, naast deze functionaliteiten richt de tweede pilot zich ook voor een klein deel op het ontwikkelen van extra functionaliteiten voor de gebruikers van de Marti-IT software. Deze functionaliteiten zijn extra bovenop de bestaande functionaliteit van het systeem.

5.5.1 Maken van back-ups

Het bestaande Marti-IT Update Centrum bevat geen functionaliteit voor het maken van back-ups, wat er regelmatig voor zorgt dat een gebruiker moet terugkeren naar de Marti-IT applicatie en het Update Centrum moet afsluiten, om vervolgens een gegevensback-up te kunnen maken. Het maken van gegevensback-ups vanuit het Update Centrum was een belangrijke functionaliteit, aangezien gebruikers dan in de toekomst niet meer hoeven terug te keren naar een applicatie om een gegevensback-up te maken.

De doelstelling met betrekking tot het ontwikkelen van een functionaliteit voor het maken van back-ups is:

- ✓ Het ontwikkelen van een functionaliteit voor het maken van gegevensback-ups voor elke Marti-IT applicatie waarover de betreffende gebruiker beschikt

De werkzaamheden met betrekking tot het ontwikkelen van deze functionaliteit zijn:

- ✓ Het ontwerpen van de functionaliteit voor het maken van gegevensback-ups
- ✓ Het programmeren van de functionaliteit voor het maken van gegevensback-ups
- ✓ Het ontwikkelen van een venster voor het maken van back-ups
- ✓ Het koppelen van de ontwikkelde functionaliteit aan het ontwikkelde venster

5.5.2 Downloaden van historische updates

Het downloaden van historische updates dient gebruikers in staat te stellen om updates uit het verleden te kunnen downloaden. Op deze manier kan er altijd op een oudere versie worden teruggevallen, indien een nieuwe update bijvoorbeeld niet goed functioneert.

De doelstelling met betrekking tot het ontwikkelen van een functionaliteit voor het downloaden van historische updates is:

- ✓ Het ontwikkelen van een functionaliteit waarmee gebruikers alle beschikbare updates uit het verleden kunnen downloaden

De werkzaamheden met betrekking tot het ontwikkelen van deze functionaliteit zijn:

- ✓ Het ontwerpen van de functionaliteit voor het downloaden van historische updates, gebaseerd op het ontwerp van de functionaliteit voor het downloaden van (de nieuwste) updates
- ✓ Het programmeren van de functionaliteit voor het downloaden van historische updates
- ✓ Het ontwikkelen van een venster voor het downloaden van de historische updates, gebaseerd op het bestaande venster voor het downloaden van (de nieuwste) updates
- ✓ Het koppelen van de ontwikkelde functionaliteit aan het ontwikkelde venster voor het downloaden van de historische updates

© 2005 Mathieu Brüning (20010019)

5.5.3 Ophalen van updates vanaf lokale en/of netwerklocatie

Het distribueren van ontvangen updates over bijvoorbeeld een netwerk is in het huidige Marti-IT Update Centrum niet mogelijk, wat feitelijk betekent dat elke computer in een netwerk individueel een update moet downloaden. Dit is een tijdsintensief proces, aangezien er dan voor elke computer een update gedownload dient te worden, terwijl de benodigde update bestanden al aanwezig zijn in het netwerk. Het installeren van updates vanaf een lokale en/of netwerklocatie dient zodoende zorg te dragen voor het distribueren van updates, zodat er in de toekomst niet voor elke computer apart een update gedownload hoeft te worden.

De doelstellingen met betrekking tot het ontwikkelen van een functionaliteit voor het ophalen van updates van een lokale en/of netwerklocatie zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het vergaren van informatie omtrent het bestaande update bestand
- ✓ Het ontwikkelen van een functionaliteit voor het installeren van het bestaande update bestand

De werkzaamheden met betrekking tot het ontwikkelen van deze functionaliteit zijn:

- ✓ Het ontwerpen van een functionaliteit voor het vergaren van informatie omtrent het bestaande update bestand
- ✓ Het programmeren van een functionaliteit voor het vergaren van informatie omtrent het bestaande update bestand
- ✓ Het programmeren van een functionaliteit voor het installeren van het bestaande update bestand, gebaseerd op de reeds ontwikkelde functionaliteit voor het installeren van updates
- ✓ Het ontwikkelen van een vensters voor het selecteren en installeren van bestaande update bestanden
- ✓ Het koppelen van de ontwikkelde functionaliteit aan de ontwikkelde vensters

5.6 Extra functionaliteiten op het gebied van .zip

De ontwikkeling van de nieuwste Marti-IT applicatie Cash-it, wat de eerste applicatie middels de C# programmeertaal is voor Marti-IT, heeft ervoor gezorgd dat er een kleine C# library is opgezet binnen het bedrijf. Deze C# library bevat een klasse voor het gebruik van .zip bestanden, maar deze klasse is echter nog onvolledig. Het in- en uitpakken van bestanden is mogelijk middels deze klasse, maar mist echter functionaliteit op het gebied van het behouden van mapstructuren bij het in- en uitpakken van bestanden. Om update bestanden op de juiste manier in en uit te kunnen pakken, zodat deze qua structuur op de juiste manier geïnstalleerd kunnen worden bij gebruikers, diende de klasse voor het gebruik van .zip uitgebreid te worden.

De doelstellingen met betrekking tot het uitbreiden van de .zip mogelijkheden zijn:

- ✓ Het ontwikkelen van een functionaliteit voor het behouden van een mappenstructuur bij het in- en uitpakken van bestanden
- ✓ Het ontwikkelen van een functionaliteit voor het inpakken van meerdere bestanden

De werkzaamheden met betrekking tot het uitbreiden van de .zip mogelijkheden zijn:

- ✓ Het programmeren van een functionaliteit voor het recursief inventariseren van een mappenstructuur
- ✓ Het programmeren van een functionaliteit voor het inpakken van meerdere bestanden

© 2005 Mathieu Brüning (20010019)

6 Test- en beoordelingsplan

Nadat de ontwikkeling van de tweede pilot in de pilotontwikkelingsfase is gecompleteerd dienen er testwerkzaamheden plaats te vinden om te beoordelen of de ontwikkelde pilotdelen wel in overeenstemming zijn met hetgeen wat er vooraf vastgesteld is. Het vastleggen van de procedure en criteria rondom het testen van de ontwikkelingen uit de tweede pilot staat daarom centraal in dit hoofdstuk.

6.1 Testprocedure

De testprocedure met betrekking tot het testen van de ontwikkelingen in de tweede pilot richt zich hoofdzakelijk op het uitvoeren van Black Box tests om te controleren of de ontwikkelde functionaliteiten overeenkomen met de vooraf gestelde functionele kenmerken. Aangezien de functionaliteiten uit de tweede pilot niet vergeleken kunnen worden met reeds bestaande functionaliteiten, dienen de ontwikkelde functionaliteiten uit de tweede pilot getest te worden op basis van de functionele specificaties.

De procedure voor het testen en beoordelen van de tweede pilot is als volgt:

5. Opstellen testplan
6. Uitvoeren tests volgens testplan
7. Verslaglegging resultaten testplan
8. Beoordelen ontwikkelingen a.d.h.v. resultaten testplan

6.2 Testcriteria

De criteria met betrekking tot de testwerkzaamheden zullen hoofdzakelijk gebaseerd zijn op de functionele specificaties, zoals hieronder vermeldt. Op basis van deze functionele specificaties zal ik analyseren of de ontwikkelde pilotdelen overeenkomen met hetgeen wat er in de definitiestudie is vastgelegd en of de functionaliteiten de juiste werking hebben.

De volgende functionaliteiten voor de medewerkers van Marti-IT dienen geïmplementeerd en functioneel te zijn:

- ✓ Gegevens vastleggen omtrent beschikbare updates
- ✓ Vastleggen van gegevens van gebruikers
- ✓ Wijzigen gegevens softwaregebruik gebruikers
- ✓ Klaarzetten van programma updates
- ✓ Extra functionaliteiten op het gebied van .zip
- ✓ Automatisch downloaden van back-ups naar Marti-IT lokale server

De onderstaande functionaliteiten zijn extra met betrekking tot de bestaande functionaliteit van het Marti-IT Update Centrum. Waar mogelijk dienen ook deze functionaliteiten geïmplementeerd en functioneel te zijn, echter, de prioriteit van de onderstaande functionaliteiten ligt lager dan de prioriteit van de bovenstaande functionaliteiten.

Functionaliteiten voor de gebruikers van de Marti-IT software:

- ✓ Maken van een back-up
- ✓ Downloaden historische updates
- ✓ Updates ophalen vanaf lokale en/of netwerklocaties in plaats van het Internet

© 2005 Mathieu Brüning (20010019)

PILOTONTWIKKELPLAN PILOT 3

Aanhef:

Module:	Afstuderen	**Opdrachtgever:**	Dhr. M.L. Verkade
Docent(en):	Dhr. J.L.A. Schramp Dhr. M. Reijnhoudt	**Bedrijfsmentor:**	Dhr. M.L. Verkade
		Plaats:	Naaldwijk
Organisatie:	Marti-IT	**Datum:**	16 mei 2005

Student:

Naam:	**E-Mail:**
Brüning, Mathieu	20010019

© 2005 Mathieu Brüning (20010019)

Inhoudsopgave

1 INLEIDING 150

2 GLOBAAL-FUNCTIONELE STRUCTUUR 151

3 GLOBAAL-TECHNISCHE STRUCTUUR 152

4 GLOBAAL-ORGANISATORISCHE STRUCTUUR 153

5 PILOTSTRUCTUUR 154

5.1 HET ONTWIKKELEN VAN EEN DYNAMISCHE MENUSTRUCTUUR 154

5.2 HET ONTWIKKELEN VAN EEN HOOFDMENU 154

5.3 HET ONTWIKKELEN VAN EEN FUNCTIONALITEIT VOOR HET WEERGEVEN VAN NIEUWSPAGINA'S 155

5.4 HET HANTEREN VAN EEN NIEUWE KLEURENSAMENSTELLING / VERBETEREN VENSTERS 155

6 TEST- EN BEOORDELINGSPLAN 156

6.1 TESTPROCEDURE 156

6.2 TESTCRITERIA 156

1 Inleiding

De ontwikkeling van de derde en tevens laatste pilot in de pilotontwikkelingsfase kon van start gaan nadat de tweede pilot was afgerond. Het opstellen van een pilotontwikkelplan dient opnieuw inzicht te verschaffen in de structuur van de pilot en een overzicht van de ontwikkeling van de pilot.

Het doel van dit document is het vastleggen van de structuur van de derde pilot in de pilotontwikkelingsfase en het vastleggen van de wijze waarop de ontwikkeling zal plaatsvinden. Daarnaast dient dit document inzicht te verschaffen omtrent de globale functionele, technische en organisatorische aspecten van de derde pilot. Tot slot richt dit pilotontwikkelplan zich op het vastleggen van de diverse aspecten rondom het testen van de ontwikkelde pilot.

Het pilotontwikkelplan is als volgt ingedeeld: de hoofdstukken twee, drie en vier bevatten beschrijvingen van de globale functionele, technische en organisatorische structuur van de laatste pilot. In deze hoofdstukken wordt globaal uiteengezet aan welke functionele, technische en organisatorische kenmerken de derde pilot moet voldoen. Hoofdstuk vijf richt zich op de structuur van de te ontwikkelen pilot. In eerste instantie wordt in dit hoofdstuk een globale indeling van de pilot gemaakt in pilotdelen, waarop vervolgens concreet wordt gespecificeerd welke activiteiten er plaats dienen te vinden. Het laatste hoofdstuk in dit pilotplan richt zich op het vastleggen van procedures en criteria voor het testen van de ontwikkelde pilot.

© 2005 Mathieu Brüning (20010019)

2 Globaal-functionele structuur

Waar de ontwikkeling van de eerste twee pilots zich hoofdzakelijk richtte op het realiseren van de functionaliteit voor het Marti-IT Update Centrum en het Marti-IT Build Centrum, richt de derde pilot zich hoofdzakelijk op het verbeteren van de gebruikersvriendelijkheid. Het aanpassen van de interface speelt hierbij een belangrijke rol, aangezien het verbeteren van de interface zorg dient de dragen voor het vergemakkelijken van het gebruik van het systeem.

De clusters uit pilot drie worden hieronder vermeldt:

- ✓ Het ontwikkelen van een dynamische menustructuur
- ✓ Het ontwikkelen van een hoofdmenu
- ✓ Het ontwikkelen van nieuwspagina's in HTML formaat, die worden getoond tijdens het verzenden of ontvangen van gegevens
- ✓ Het hanteren van een nieuwe kleurensamenstelling / verbeteren van vensters

© 2005 Mathieu Brüning (20010019)

3 Globaal-technische structuur

Wat betreft de globaal-technische structuur van de derde pilot richt de ontwikkeling zich hoofdzakelijk op de uitbreiding van de reeds ontwikkelde Marti-IT Update Centrum applicatie. Deze uitbreidingen vinden hoofdzakelijk plaats middels de C# programmeertaal. Het weergeven van de HTML-pagina's tijdens het downloaden of uploaden van gegevens vind echter plaats via de HyperText Markup Language (HTML). Voor het ophalen van de HTML-pagina's is op opnieuw sprake van een client-server architectuur, waarbij de html-pagina's van de server worden gedownload, om vervolgens te kunnen weergeven in het Marti-IT Update Centrum.

© 2005 Mathieu Brüning (20010019)

4 Globaal-organisatorische structuur

De globaal-organisatorische structuur van de ontwikkeling van de derde pilot richt zich opnieuw op de gebruikers van de Marti-IT software. Het verbeteren van de gebruikersvriendelijkheid van het systeem dient namelijk zorg te dragen voor het vergemakkelijken van de Update Centrum, zodat het gebruik van het systeem vereenvoudigd wordt voor de gebruikers.
Door bijvoorbeeld het plaatsen van een hoofdmenu, waarmee gebruikers middels een druk op een knop direct kunnen de nieuwste updates kunnen inventariseren en downloaden, hoeven de gebruikers niet meer door menustructuren te zoeken.

Hoofdzakelijk richt de globaal-organisatorische structuur van de derde pilot zich dus op de toekomstige gebruikers van het Marti-IT Update Centrum, aangezien de uitbreidingen in de derde pilot met name gericht zijn op het vergemakkelijken van het gebruik van het systeem voor de gebruikers van de Marti-IT software.

© 2005 Mathieu Brüning (20010019)

5 Pilotstructuur

5.1 Het ontwikkelen van een dynamische menustructuur

Het ontwikkelen van een dynamische menustructuur is met name gericht op het verwijderen van redundante informatie voor gebruikers. Op basis van het softwareprofiel van een gebruiker wordt namelijk de menustructuur opgebouwd, zodat een gebruiker nooit de mogelijkheid heeft om menuopties te gebruiken voor applicaties waarover deze gebruiker niet beschikt.

De doelstellingen met betrekking tot het ontwikkelen van een dynamische menustructuur zijn:

- ✓ Het ontwikkelen van een menustructuur waarin een gebruiker alleen de menuopties te zien krijgt welke van toepassing zijn op deze gebruiker
- ✓ Het ontwikkelen van een menustructuur waarin de applicaties en bijbehorende download types dynamisch worden ingeladen, zodat een gebruiker gericht updates kan inventariseren

De werkzaamheden met betrekking tot het ontwikkelen van een dynamische menustructuur zijn:

- ✓ Het ontwerpen van een dynamische menustructuur welke gebaseerd is op het softwareprofiel van een gebruiker
- ✓ Het programmeren van een dynamische menustructuur
- ✓ Het koppelen van de menustructuur aan de reeds ontwikkelde interface van het Marti-IT Update Centrum

5.2 Het ontwikkelen van een hoofdmenu

Het implementeren van een hoofdmenu voor het Marti-IT Update Centrum dient een belangrijke rol te spelen in de verbetering van de gebruikersvriendelijkheid van het nieuwe systeem. Het hoofdmenu dient namelijk zorg te dragen voor het aansturen van de belangrijkste functionaliteiten in het systeem, zodat een gebruiker op eenvoudige wijze gegevens kan ontvangen of verzenden, zonder dat daarbij gebruik gemaakt hoeft te worden van een menu. Het implementeren van een hoofdmenu is een belangrijke uitbreiding ten opzichte van het huidige Marti-IT Update Centrum, aangezien het huidige systeem geen hoofdmenu bevat.

De doelstellingen met het betrekking tot het ontwikkelen van een hoofdmenu zijn:

- ✓ Het ontwikkelen van een venster waarin de belangrijkste functionaliteiten van het systeem op eenvoudige wijze aangeroepen kunnen worden
- ✓ Het grafisch verbeteren van het venster door middels van bijvoorbeeld afbeeldingen

De werkzaamheden met betrekking tot het ontwikkelen van een hoofdmenu zijn:

- ✓ Het selecteren van de belangrijkste functionaliteiten van het Marti-IT Update Centrum, welke aangeroepen dienen te worden vanuit het hoofdmenu
- ✓ Het maken van een venster voor het hoofdmenu
- ✓ Het programmeren van de functionaliteit voor het hoofdmenu
- ✓ Het grafisch verbeteren van het hoofdmenu middels een grafisch softwarepakket als Adobe Photoshop
- ✓ Het implementeren van het hoofdmenu in het reeds ontwikkelde hoofdvenster van het Marti-IT Update Centrum

© 2005 Mathieu Brüning (20010019)

5.3 Het ontwikkelen van een functionaliteit voor het weergeven van nieuwspagina's

Het ontwikkelen van een functionaliteit voor het weergeven van nieuwspagina's dient gebruikers op de hoogte te stellen van de nieuwste ontwikkelingen in de Marti-IT software en Marti-IT als organisatie. Door nieuwspagina's te tonen in HyperText Markup Language (HTML) formaat is het mogelijk om de informatie in de pagina's op eenvoudige wijze te veranderen.

Het tonen van de nieuwspagina's zal plaatsvinden wanneer een gebruiker gegevens aan het verzenden of ontvangen is. Op deze manier is de gebruiker geen extra tijd kwijt aan het verzenden of ontvangen van gegevens, aangezien het downloaden en uploaden van gegevens veelal een moment is waarop gebruikers aan het wachten zijn op het verzendproces.

De doelstellingen met betrekking tot het ontwikkelen van een functionaliteit voor het weergegeven van nieuwspagina's is:

- ✓ Het ontwikkelen van een venster waarin HyperText Markup Language bestanden in weergegeven kunnen worden
- ✓ Het ontwikkelen van een functionaliteit voor het weergegeven van nieuwspagina's tijdens het verzenden of ontvangen van gegevens

De werkzaamheden betrekking tot het ontwikkelen van deze functionaliteit zijn:

- ✓ Het ontwerpen van de functionaliteit voor het weergeven van de nieuwspagina's
- ✓ Het programmeren van de functionaliteit voor het weergeven van de nieuwspagina's
- ✓ Het ontwikkelen van een venster waarin de pagina's kunnen worden weergegeven
- ✓ Het koppelen van de ontwikkelde functionaliteit aan het ontwikkelde venster
- ✓ Het koppelen van het venster aan het venster voor het weergegeven van de verzend status

5.4 Het hanteren van een nieuwe kleurensamenstelling / verbeteren vensters

De laatste cluster in de ontwikkeling van de derde pilot richt zich op het hanteren van een nieuwe kleurensamenstelling en het verbeteren van reeds ontwikkelde vensters. Het hanteren van een nieuwe kleurensamenstelling dient namelijk ook mee te dragen aan de verbetering van de interface, aangezien de vensters door het gebruik van verschillende kleuren duidelijker en meer uitnodigend en intuïtief dient te worden.

De doelstellingen met betrekking tot het hanteren van een nieuwe kleurensamenstelling zijn:

- ✓ Het duidelijker maken van vensters door middel van het gebruik van verschillende kleuren
- ✓ Het meer uitnodigend en maken van vensters
- ✓ Het verbeteren van het uiterlijk van vensters, zodat deze meer lijken op de hedendaagse applicaties

De werkzaamheden met betrekking tot het hanteren van een nieuwe kleurensamenstelling / verbeteren vensters zijn:

- ✓ Het selecteren van kleuren voor het gebruik van het nieuwe systeem
- ✓ Het toepassen van de geselecteerde kleuren in de reeds ontwikkelde vensters
- ✓ Het maken van een afbeelding voor het verzend venster
- ✓ Het grafisch verbeteren van vensters

© 2005 Mathieu Brüning (20010019)

6 Test- en beoordelingsplan

De laatste activiteit in de derde pilot richt zich op het testen en beoordelen van de ontwikkelingen uit de derde pilot. Het vastleggen van de testprocedure en het opstellen van testcriteria staat centraal in dit hoofdstuk.

6.1 Testprocedure

Over het algemeen gezien zijn sommige aspecten van de gebruikersvriendelijkheid naar mijn idee subjectief. Met name de grafische veranderingen kunnen door de ene gebruiker worden opgevat als verbetering, terwijl een andere gebruiker van mening is dat de grafische aanpassingen het gebruik van het systeem verslechten. Wat dat betreft is het naar mijn mening moeilijk om sommige ontwikkelingen met betrekking tot het verbeteren van de gebruikersvriendelijkheid zelf te testen. Het vaststellen of de gebruikersvriendelijkheid van het systeem door de uitbreidingen echt is verbeterd zal ik daarom doen middels een acceptatie test met de gebruikers, in de nazorg / implementatie fase van het project. Op basis van een test bij een aantal gebruikers zal ik vragen wat zij vinden van de veranderingen en of de veranderingen bij hebben gedragen aan een verbetering van de gebruikersvriendelijkheid.

Het testen van de functionele kenmerken van de ontwikkelingen kan ik daarentegen wel testen. De functionaliteit van de dynamische menustructuur, de functionaliteit voor het hoofdmenu en de functionaliteit voor het weergeven van de nieuwspagina's zal ik aan een Black Box real-life test onderwerpen.

De procedure voor het testen en beoordelen van een pilot is als volgt:

9. Opstellen testplan
10. Uitvoeren tests volgens testplan
11. Verslaglegging resultaten testplan
12. Beoordelen ontwikkelingen a.d.h.v. resultaten testplan

6.2 Testcriteria

De volgende criteria zijn aan de orde bij het testen van de ontwikkelingen uit de derde pilot:

- ✓ Er dient een dynamische menustructuur geïmplementeerd te zijn, waarin alleen de software items worden weergegeven die behoren tot het softwareprofiel van de ingelogde gebruiker
- ✓ Er dient een hoofdmenu geïmplementeerd te zijn, waar vanuit de gebruikers de belangrijkste functionaliteiten van het Marti-IT Update Centrum kunnen aanroepen, zoals ondermeer het downloaden van updates, rapporten en licenties en het uploaden van back-ups.
- ✓ Er dient een functionaliteit geïmplementeerd te zijn welke het mogelijk maakt om nieuwspagina's in HTML formaat te downloaden en weer te geven, zodat deze getoond worden tijdens het verzenden en ontvangen van gegevens

© 2005 Mathieu Brüning (20010019)

Bijlage 5: Reference cards Marti-IT Update / Build Centrum

Zie volgende pagina.

© 2005 Mathieu Brüning (20010019)

OPSTARTEN VAN UPDATE-IT

- Start Update-it
- Na het opstarten van Update-it verschijnt het autorisatievenster
- Voer uw gebruikersnummer in
- Voer uw wachtwoord in
- (Optioneel: Vink het veld 'Bewaar wachtwoord' aan om het wachtwoord te onthouden)
- Nadat uw geautoriseerd bent wordt het hoofdmenu van Update-it getoond
- Hierin kunt u de belangrijkste functionaliteiten van Update-it aanroepen, zoals het uploaden van back-ups en het downloaden van updates

- NB: Order-it afnemers kunnen inloggen door op de "Order-it" knop te drukken

DOWNLOADEN

- Middels het hoofdmenu of het menu 'Downloads' wordt het venster weergegeven voor het downloaden van updates
- Het venster voor het downloaden van updates is als volgt opgebouwd:
  - Software
    Aan de linkerkant van het venster ziet u de beschikbare applicaties en bijbehorende download types. Door een applicatie of download type te selecteren aan de linkerkant van het venster worden de daarvoor beschikbare updates weergegeven.

  - Beschikbare updates
    Aan de rechterbovenkant van het venster ziet u de beschikbare updates die voldoen aan uw selectie. Deze selectie is gebaseerd op hetgeen wat u aan de linkerkant van het venster onder het kopje "Software" heeft geselecteerd.

  - Detailgegevens
    Informatie omtrent de geselecteerde applicatie en de geselecteerde update wordt rechtsonder in het venster weergegeven.

- Door één of meerdere updates te selecteren en vervolgens te drukken op [Downloaden updates] wordt het voortgangsvenster voor het verzendproces weergegeven.

EXTRA FUNCTIES

- **Maken backup**
  Middels het hoofdmenu of via het menu 'Extra' wordt het venster voor het maken van een back-up weergegeven. Het venster voor het maken van een back-up is als volgt opgebouwd:
  - 1. Selecteren applicatie
    Voor het maken van een gegevensback-up vanuit Update-it dient u in eerste instantie de applicatie te selecteren waarvoor u de back-up wenst te maken. Middels het veld 'Applicatie' kunt u een applicatie selecteren.

  - 2. Maken van de back-up
    Het maken van de back-up kunt u uitvoeren door te drukken op de [Maak een gegevensback-up voor de geselecteerde applicatie]. Indien het inpakproces succesvol is afgerond wordt de vraag gesteld of het ingepakte bestand direct geupload dient te worden. Druk op de [Ja] knop om direct het bestand te verzenden naar Marti-IT.

  - 3. Uitvoer bestand
    Wanneer de gegevensback-up is gemaakt wordt de informatie omtrent het bestand weergegeven in de sectie ´3. Uitvoer bestand´.

- **Historische downloads**
  Middels het menu 'Extra' wordt het venster voor het downloaden van historische updates weergegeven. Deze functionaliteit is hetzelfde als de functionaliteit voor het downloaden van updates, met het verschil dat er in dit venster alleen updates uit het verleden gedownload en geïnstalleerd kunnen worden.

© 2005 Mathieu Brüning (20010019)

UPLOADEN

Middels het hoofdmenu of het menu 'Uploads' kunt u de functies aanroepen voor het verzenden van gegevens naar Marti-IT. Hiervoor heeft u een drietal mogelijkheden, namelijk het uploaden van:

- back-ups
- bestanden of mappen
- standaard uploads

➢ **Uploaden back-ups**

Middels het hoofdmenu of via het menu Uploads | Uploaden backups | <Applicatienaam> wordt het venster voor het uploaden van back-ups weergegeven. Het venster voor het uploaden van back-ups is als volgt opgebouwd:

- Software
  Aan de linkerkant van het venster ziet u de beschikbare applicaties. Door een applicatie te selecteren aan de linkerkant van het venster worden de daarvoor beschikbare back-ups weergegeven.

- Beschikbare back-ups
  Aan de rechterbovenkant van het venster ziet u de beschikbare back-ups die voldoen aan uw selectie. Deze selectie is gebaseerd op de applicatie die u aan de linkerkant van het venster onder het kopje "Software" heeft geselecteerd.

- Detailgegevens
  Informatie omtrent de geselecteerde back-up wordt rechtsonder in het venster weergegeven.

➢ **Uploaden bestanden / mappen**

Middels het hoofdmenu of via het menu Uploads | Bestanden / mappen wordt het venster voor het uploaden van bestanden of mappen weergegeven. Het venster voor het uploaden van bestanden of mappen is als volgt opgebouwd:

- 1. Selecteren bestanden / mappen
  De eerste sectie in het venster richt zich op het selecteren van bestanden of mappen. Middels de [Bladeren] knoppen aan de rechterkant van het venster kunt u een bestand of een map selecteren om in het vervolg te verzenden.

UPLOADEN (Vervolg)

- 2. Inpakken bestanden / mappen
  Om het geselecteerde bestand of de geselecteerde map te verzenden dient het in eerste instantie ingepakt te worden. Dit kunt u doen door te drukken op de [Pak de geselecteerde map of bestand in]. De voortgang van het inpakproces wordt weergegeven in de voortgang meter.
  Indien het inpakproces succesvol is afgerond wordt de [Volgende] knop geactiveerd, zodat u middels een druk op deze knop het bestand kunt verzenden naar Marti-IT.

- 3. Uploaden bestand
  Wanneer de geselecteerde bestanden of geselecteerde map is ingepakt wordt de informatie omtrent het uitvoerbestand weergegeven in de sectie ´3. Uploaden bestand´.

➢ **Standaard uploads**

Middels het menu Uploads | Standaard uploads wordt het venster voor het verzenden van standaard gegevens weergegeven. Dit venster is als volgt opgebouwd:

- Software
  Aan de linkerkant van het venster ziet u de beschikbare applicaties. Door op een applicatie drukken aan de linkerkant van het venster worden de daarvoor beschikbare standaard uploads weergegeven.

- Beschikbare standaard uploads
  Aan de rechterbovenkant van het venster ziet u de beschikbare standaard uploads die voldoen aan uw selectie. Deze selectie is gebaseerd op de applicatie die u aan de linkerkant van het venster onder het kopje "Software" heeft geselecteerd.

- Detailgegevens
  Informatie omtrent de geselecteerde standaard upload wordt rechtsonder in het venster weergegeven.

© 2005 Mathieu Brüning (20010019)

OPSTARTEN VAN BUILD-IT

- Start Build-it
- Na het opstarten van Build-it verschijnt het hoofdvenster, van hieruit kunt u middels het menu aan de bovenkant van het venster de functionaliteiten van het systeem aanroepen

KLANTGEGEVENS

- Middels het menu 'Namen' kunt u de functionaliteiten aanroepen voor het beheren van de klantgegevens in de database voor Update-it. Hierbij heeft u de keuze tussen twee menuopties, namelijk:

  - Klanten
    Middels de menuoptie 'Klanten' wordt de lijst met de beschikbare klanten in de database getoond. Door te drukken op de [Openen] knop kunnen de gegevens van de betreffende klant bekeken worden.

    N.B. De gegevens van de klant kunnen niet gewijzigd worden, dit kan alleen door de gegevens aan te passen in Serv-it en vervolgens te synchroniseren met de database.

    Naast het inzien van gegevens is het mogelijk om applicaties en persoonlijke updates te koppelen en ontkoppelen aan gebruikers. De tabbladen op de klantkaart genaamd "Applicaties" en "Persoonlijke updates" kunnen gebruikt voor het koppelen en ontkoppelen van applicaties en persoonlijke updates. Dit kan plaatsvinden door middel van de "Toevoegen" en "Verwijderen" knoppen bovenop het venster.

  - Synchroniseren klanten
    Middels de menuoptie 'Synchroniseren klanten' kunnen de gegevens van de database worden gesynchroniseerd met de gegevens uit Serv-it. Het venster voor het synchronisatieproces zal worden getoond waarop u op [Start] dient te drukken. De actuele klantgegevens zullen dan uit Serv-it worden opgehaald en worden weggeschreven in de database.

    Nadat het synchronisatieproces is afgerond wordt er een melding gegevens of er nog fouten geconstateerd zijn tijdens het synchroniseren.

ONDERHOUDEN APPLICATIES & DOWNLOAD TYPES

- Het menu 'Onderhoud' bevat de functionaliteiten voor het beheren van de applicaties en download types in de database voor Update-it. Middels twee menuopties kunnen de vensters voor het beheer worden weergegeven, wat betekent dat u applicaties en download types kunt toevoegen, wijzigen en verwijderen.

  - Applicaties
    Middels de menuoptie 'Applicaties' wordt de lijst met de beschikbare applicaties in de database getoond. Door te drukken op de [Openen] of [Wijzigen] knop kunnen de gegevens van de betreffende applicatie bekeken worden. Het opslaan van aangepaste detailgegevens kan plaatsvinden middels de [Opslaan] knop.

    De [Toevoegen] knop kan gebruikt worden om een nieuwe applicatie toe te voegen aan de database. Het venster voor het invoeren van de detailgegevens van de applicatie zal dan worden weergegeven.
    De [Verwijderen] knop kan gebruikt worden om een bestaande applicatie te verwijderen uit de database.

    Het venster met de detail gegevens van een applicatie bevat een tabblad genaamd "Download types". Middels dit tabblad kunt u download types koppelen aan een applicatie, zodat er updates aan gekoppeld kunnen worden. Het ontkoppelen van een download type is tevens mogelijk.

  - Download types
    Middels de menuoptie 'Download types' kunnen de download types in de database wordt beheert. Een lijst van de download types wordt weergegeven wanneer u de menoptie "Download types" aanklikt. In deze lijst kunt u drukken op [Openen] of [Wijzigen] voor het inzien en aanpassen van de detailgegevens van een download type.
    Het toevoegen of verwijderen van een download type kan plaatsvinden middels de [Toevoegen] of [Verwijderen] knop.

UPDATES

Middels het menu 'Updates' kunt u de functies aanroepen voor het beheren van de updates voor Update-it. Het menu is hierbij ingedeeld op applicatienaam en download type, zodat er gericht gezocht kan worden op specifieke Marti-IT software. Indien u één van de menuopties gebruikt wordt het venster voor het selecteren van updates weergegeven. De updates die dan worden getoond in het venster zijn dan gefilterd op basis van de geselecteerde menuoptie in het hoofdvenster.

De linkerzijde van het venster bevat een boomstructuur met de beschikbare applicaties en download types erin. Door middel van het selecteren van een applicatie of download type kunt u in het selectievenster de filter wijzigen, zodat de updates worden weergegeven op basis van uw selectie.

Middels de [Openen] en [Wijzigen] knop op het venster bent u in staat om de detailgegevens van een geselecteerde update in te zien en te wijzigen. Het venster met de detailgegevens van een update zal dan namelijk worden weergegeven, waarin u vervolgens de gegevens kunt wijzigen en kunt opslaan.

De [Toevoegen] knop op het selectievenster kan gebruikt worden om nieuwe updates toe te voegen aan de database. Indien u kiest voor de [Toevoegen] knop wordt het detailvenster van een update weergegeven, zodat u de gegevens omtrent de update kunt invoeren.

De [Verwijderen] knop stelt u in staat om bestaande updates te verwijderen uit de database.

UPDATES DETAILVENSTER

Het detailvenster voor updates wordt weergegeven wanneer u de gegevens van een update inzien of wanneer u een nieuwe update toevoegt. Onderstaand worden de velden en knoppen van het venster uiteengezet, zodat duidelijk is hoe een update toegevoegd of gewijzigd kan worden.

Applicatie:	Selecteer de applicatie waarvoor de update is
Download type:	Selecteer het download type waarbij de update behoort
Actief:	Vink aan indien de update actueel is
Toepassen:	Vink aan indien de update geïnstalleerd dient te worden
Gezipped:	Vink aan indien het update bestand ingepakt is in .zip formaat
Versie checken:	Vink aan indien de versie gecheckt dient te worden
Prioriteit:	Selecteer een prioriteit voor de update (1 is hoogst)
Persoonlijke update:	Vink aan indien het een persoonlijke update betreft
Toewijzen klanten:	Gebruikers toekennen aan deze update
Installatiefolder:	Voer de installatiemap voor de update in
Bestandsnaam:	De bestandsnaam van het updatebestand (aut.)
Selecteren update:	Selecteer een reeds bestaand updatebestand
Maken update:	Maak een nieuw updatebestand a.d.h.v. bestanden of mappen
Versie:	Voer de versie in voor de update
Datum:	Selecteer de datum voor de update
Buildnummer:	De datum/tijd stempel voor de update (aut.)
Omschrijving:	Voer de omschrijving in van de update
Lokale pad:	Het lokale pad waar de update zich bevindt binnen Marti-IT (aut.)
FTP pad:	Het pad op de FTP-server waar de update wordt geplaatst (aut.)

Indien de benodigde velden zijn ingevuld kan de update opgeslagen worden middels de [Opslaan / uploaden] knop op het venster. De detailgegevens omtrent de update zullen dan worden opgeslagen in de database. Nadat de gegevens zijn opgeslagen wordt de vraag gesteld of u direct het updatebestand wilt uploaden naar de FTP-server. Indien u kiest voor [Ja] wordt het bestand vanaf de lokale server geupload naar de FTP-server. De voortgang van het uploadproces kunt u dan zien in de balk links onderin het venster.

© 2005 Mathieu Brüning (20010019)

Bijlage 6: Testrapport acceptatietest gebruikers Marti-IT software

TESTRAPPORT ACCEPTATIETEST GEBRUIKERS MARTI-IT SOFTWARE

Aanhef:

Module:	Afstuderen	**Opdrachtgever:**	Dhr. M.L. Verkade
Docent(en):	Dhr. J.L.A. Schramp Dhr. M. Reijnhoudt	**Bedrijfsmentor:**	Dhr. M.L. Verkade
		Plaats:	Naaldwijk
Organisatie:	Marti-IT	**Datum:**	31 mei 2005

Student:

Naam:	**E-Mail:**
Brüning, Mathieu	20010019

© 2005 Mathieu Brüning (20010019)

Inhoudsopgave

INHOUDSOPGAVE **158**

1 INLEIDING **158**

2 TESTOMGEVING **158**

2.1 HARDWARE 158
2.2 SOFTWARE 158
2.3 DOCUMENTATIE 158
2.4 LOCATIE 158

3 TESTSCENARIO'S **158**

3.1 TEST SCENARIO 1: AUTORISATIE FUNCTIONALITEIT 158
3.2 TEST SCENARIO 2: DOWNLOADEN PROGRAMMA UPDATE 158
3.3 TEST SCENARIO 3: DOWNLOADEN RAPPORTEN UPDATE 158
3.4 TEST SCENARIO 4: DOWNLOADEN PERSOONLIJKE UPDATE 158
3.5 TEST SCENARIO 5: DOWNLOADEN PERSOONLIJKE LICENTIE 158
3.6 TEST SCENARIO 6: UPLOADEN BACKUP 158
3.7 TEST SCENARIO 7: UPLOADEN BESTAND 158
3.8 TEST SCENARIO 8: UPLOADEN MAP 158
3.9 TEST SCENARIO 9: UPLOADEN STANDAARD UPLOAD 158

4 TEST SETS **158**

4.1 DATABASE GEGEVENS 158
4.2 TEST SET 1 158
4.3 TEST SET 2 158
4.4 TEST SET 3 158
4.5 TEST SET 4 158
4.6 TEST SET 5 158
4.7 TEST SET 6 158
4.8 TEST SET 7 158

5 TESTRESULTATEN **158**

5.1 TEST SCENARIO 1: AUTORISATIE FUNCTIONALITEIT 158
5.2 TEST SCENARIO 2: DOWNLOADEN PROGRAMMA UPDATE 158
5.3 TEST SCENARIO 3: DOWNLOAD RAPPORTEN UPDATE 158
5.4 TEST SCENARIO 4: DOWNLOADEN PERSOONLIJKE UPDATE 158
5.5 TEST SCENARIO 5: DOWNLOADEN PERSOONLIJKE LICENTIE 158
5.6 TEST SCENARIO 6: UPLOADEN VAN EEN BACK-UP 158
5.7 TEST SCENARIO 7: UPLOADEN VAN EEN BESTAND 158
5.8 TEST SCENARIO 8: UPLOADEN VAN EEN MAP 158
5.9 TEST SCENARIO 9: UPLOADEN STANDAARD UPLOAD 158

© 2005 Mathieu Brüning (20010019)

1 Inleiding

Nadat de pilotontwikkelingsfase was afgerond en de invoering / nazorgfase van start is gegaan zullen mijn werkzaamheden zich ondermeer gaan richten op het uitvoeren van acceptatietests. Dit betreffen tests met zowel de gebruikers van de Marti-IT software als met de medewerkers van Marti-IT, om de ontwikkelingen uit de pilotontwikkelingsfase te verifiëren.
In eerste instantie zal ik mijn werkzaamheden richten op het uitvoeren van acceptatietests met de gebruikers van Marti-IT software, zodat vastgesteld kan worden of het nieuwe Marti-IT Update Centrum gereed is voor implementatie bij de gebruikers.

Het doel van dit testrapport is het vastleggen van de resultaten van de uitgevoerde acceptatietests met betrekking tot de gebruikers van de Marti-IT software.

Het rapport is hierbij als volgt ingedeeld: hoofdstuk twee bevat een omschrijving van de testomgeving, waarin de onderdelen hardware, software, documentatie en testomgeving aan bod komen. Het daarop volgende hoofdstuk, hoofdstuk drie, richt zich op beschrijvingen van de test scenario's, waarin vermeldt wordt welke handelingen er uitgevoerd dienen te worden tijdens de testwerkzaamheden. Deze scenario's zijn onderverdeeld in een drietal aspecten, namelijk een omschrijving, de kenmerken van het testscenario en de logische testgevallen. Hoofdstuk vier bevat een opsomming van de gegevens die gebruikt zullen worden voor het uitvoeren van de testwerkzaamheden. Hoofdstuk vijf richt zich op de resultaten van de acceptatietests, welke zijn teruggekoppeld door de uitvoerders van de acceptatietests. Het laatste hoofdstuk in dit rapport bevat tot slot een conclusie omtrent de uitgevoerde acceptatietests met de gebruikers van de Marti-IT software.

© 2005 Mathieu Brüning (20010019)

2 Testomgeving

Hoofdstuk twee van dit testrapport richt zich op de testomgeving, welke van toepassing is tijdens het uitvoeren van de testwerkzaamheden voor de acceptatietest voor de gebruikers van de Marti-IT software.

2.1 Hardware

De hardware welke gebruikt zal worden voor het uitvoeren van de testwerkzaamheden voor de acceptatietest bestaat uit de volgende componenten:

- ✓ Computers van de testgebruikers waarop de testwerkzaamheden op uitgevoerd zullen worden
- ✓ Één verbinding tot het Internet, zodat er contact gelegd kan worden met de Marti-IT server
- ✓ Één Marti-IT server, waarop de database en de benodigde bestanden zijn geplaatst voor het uitvoeren van de testwerkzaamheden

2.2 Software

De onderstaande opsomming representeert de software die gebruikt zal worden voor het uitvoeren van de testwerkzaamheden:

Testgebruikers:

- ✓ Het Marti-IT Update Centrum

Ontwikkelaar:

- ✓ MySQL Control Center, voor het opzetten van de gegevens voor in test sets in de database

2.3 Documentatie

Voor het uitvoeren van de testwerkzaamheden met betrekking tot acceptatietest van het Marti-IT Update Centrum worden een tweetal documenten verschaft aan de testgebruikers. Dit zijn de volgende documenten:

- ✓ Handleiding (reference card) Marti-IT Update Centrum, voor de uitleg omtrent het gebruik van het systeem;
- ✓ Testrapport, een document waarin de structuur van de testwerkzaamheden ligt vastgelegd en de verschillende testscenario's zijn gedefinieerd. Tot slot heeft de gebruiker de mogelijkheid tot het ingeven van bevindingen, ervaringen, opmerkingen, e.d.

2.4 Locatie

Het uitvoeren van de acceptatietest voor het Marti-IT Update Centrum zal niet plaatsvinden op voorgedefinieerde locaties. De keuze voor de locatie voor het uitvoeren van de testwerkzaamheden ligt in handen van de betreffende testgebruiker. De testgegevens zullen bij de betreffende testgebruiker worden uitgezet, zodat deze de acceptatietest op een zelf gekozen locatie kan uitvoeren. Veelal zullen de werkzaamheden worden uitgevoerd binnen de organisatie van de betreffende gebruiker.

© 2005 Mathieu Brüning (20010019)

3 Testscenario's

3.1 Test scenario 1: autorisatie functionaliteit

1.1.1 Omschrijving testsituatie

Test scenario richt zich op het testen van de ontwikkelde autorisatie functionaliteit, waarbij een gebruiker zich dient aan te melden middels een (reeds bestaand) gebruikersnummer en wachtwoord. Op basis van deze gegevens wordt vervolgens een verificatie uitgevoerd aan de hand van gegevens uit de database.

3.1.1 Kenmerken test scenario

Het uitvoeren van test scenario 1 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Gebruikersvriendelijkheid
- ✓ Beveiliging

3.1.2 Logische testgevallen

1. Het invoeren van een correct gebruikersnummer en een correct wachtwoord
2. Het invoeren van een correct gebruikersnummer en een incorrect wachtwoord
3. Het invoeren van een incorrect gebruikersnummer en een correct wachtwoord
4. Het leeg laten van beide velden
5. Bewaren van het gebruikersnummer en wachtwoord
6. Het autoriseren middels de "Order-it" knop

3.2 Test scenario 2: downloaden programma update

3.2.1 Omschrijving test scenario

Het tweede test scenario richt zich op het testen van de functionaliteit voor het downloaden van een programma update voor een Marti-IT applicatie.

3.2.2 Kenmerken test scenario

Het uitvoeren van test scenario 2 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Performance
- ✓ Gebruikersvriendelijkheid

3.2.3 Logische testgevallen

1. Selecteren van een programma update en het downloaden ervan
2. Niet selecteren van een programma update en kiezen voor downloaden
3. Installeren van de programma update
4. Installeren van de programma update wanneer het betreffende programma nog actief is

© 2005 Mathieu Brüning (20010019)

3.3 Test scenario 3: downloaden rapporten update

3.3.1 Omschrijving test scenario

Het downloaden van een rapporten update voor een Marti-IT applicatie staat centraal bij het uitvoeren van de testwerkzaamheden voor test scenario 3.

3.3.2 Kenmerken test scenario

Het uitvoeren van test scenario 3 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Performance
- ✓ Gebruikersvriendelijkheid

3.3.3 Logische testgevallen

1. Selecteren van een rapporten update en het downloaden ervan
2. Installeren van een rapporten update
3. Installeren van een rapporten update wanneer één of meerdere rapporten nog actief zijn

3.4 Test scenario 4: downloaden persoonlijke update

3.4.1 Omschrijving test scenario

Test scenario 4 richt zich volledig op het downloaden van een persoonlijke update, waarvoor geldt dat deze update alleen gedownload kan worden door gebruikers die specifiek gekoppeld zijn aan deze update.

3.4.2 Kenmerken test scenario

Het uitvoeren van test scenario 4 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Performance
- ✓ Gebruikersvriendelijkheid

3.4.3 Logische testgevallen

1. Testen of de juiste persoonlijke updates worden weer gegeven
2. Selecteren van een persoonlijke update en het downloaden ervan
3. Installeren van een persoonlijke update

© 2005 Mathieu Brüning (20010019)

3.5 Test scenario 5: downloaden persoonlijke licentie

3.5.1 Omschrijving test scenario

Het downloaden van een persoonlijke licentie wordt getest tijdens de testwerkzaamheden van test scenario 5.

3.5.2 Kenmerken test scenario

Het uitvoeren van test scenario 5 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Performance
- ✓ Gebruikersvriendelijkheid

3.5.3 Logische testgevallen

1. Inventariseren van persoonlijke licentie voor Bake-it gebruiker en de weergave ervan in het download venster
2. Inventariseren van persoonlijke licentie voor Orto gebruiker en de weergave ervan in het download venster
3. Selecteren van een persoonlijke licentie en het downloaden ervan
4. Het installeren en hernoemen van een persoonlijke licentie

3.6 Test scenario 6: uploaden backup

3.6.1 Omschrijving test scenario

Het testen van de functionaliteiten voor het inventariseren, weergeven en uploaden van beschikbare back-ups voor de Marti-IT software staat central tijdens het uitvoeren van test scenario 6.

3.6.2 Kenmerken test scenario

Het uitvoeren van test scenario 6 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Performance
- ✓ Gebruikersvriendelijkheid

3.6.3 Logische testgevallen

1. Het inventarisatieproces van de beschikbare back-ups
2. Het selecteren van een back-up en het uploaden ervan
3. Het selecteren van meerdere back-ups en het uploaden ervan

© 2005 Mathieu Brüning (20010019)

3.7 Test scenario 7: uploaden bestand

3.7.1 Omschrijving test scenario

Het inpakken en uploaden van een bestand wordt getest tijdens het uitvoeren van test scenario 7.

3.7.2 Kenmerken test scenario

Het uitvoeren van test scenario 7 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Performance
- ✓ Gebruikersvriendelijkheid

3.7.3 Logische testgevallen

1. Selecteren van een bestand en het inpakken ervan
2. Selecteren van geen bestand en het inpakken ervan
3. Uploaden van een ingepakt bestand

3.8 Test scenario 8: uploaden map

3.8.1 Omschrijving test scenario

De testwerkzaamheden van test scenario 8 richten zich op het testen van de functionaliteit voor het selecteren, inpakken en uploaden van een map.

3.8.2 Kenmerken test scenario

Het uitvoeren van test scenario 8 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Performance
- ✓ Gebruikersvriendelijkheid

3.8.3 Logische testgevallen

1. Selecteren van een map (zonder submappen) en het inpakken ervan
2. Selecteren van een map (met submappen) en het inpakken ervan
3. Selecteren van geen map en het inpakken ervan
4. Uploaden van een ingepakte map

© 2005 Mathieu Brüning (20010019)

3.9 Test scenario 9: uploaden standaard upload

3.9.1 Omschrijving test scenario

Test scenario 9 richt zich op het testen van de functionaliteit voor het uploaden van standaard gegevens.

3.9.2 Kenmerken test scenario

Het uitvoeren van test scenario 9 richt zich met name op het vaststellen van de kwaliteit van de volgende kenmerken:

- ✓ Functionaliteit
- ✓ Robuustheid
- ✓ Performance
- ✓ Gebruikersvriendelijkheid

3.9.3 Logische testgevallen

1. Selecteren van een standaard upload en het uploaden ervan
2. Selecteren van geen standaard upload en het uploaden ervan
3. Selecteren van meerdere standaard uploads en het uploaden ervan

© 2005 Mathieu Brüning (20010019)

Test sets

3.10 Database gegevens

3.10.1 Tabel user

UserId	Name	Licence Kind	LicenceTo	LicenceFrom	UserName	Password	Licence
1000	Marti-IT	1	2006-01-08	2005-03-25		(Geheim)	ABCDEFGHIJKLMN OPRSTUVW1234567890
Order-it	Order-it afnemers	1	0000-00-00	0000-00-00	Order-it afnemers	Order-it	

3.10.2 Tabel application

ApplicationId	Name	LastVersion	DateLastVersion
11011029-2005-0419-9734-636174696f6e	Bake-it	1.78	2005-03-29
11052762-2005-0419-1d87-636174696f6e	Orto	1.73	2004-03-24
11031381-2005-0419-0f02-636174696f6e	Order-it	1.45	2005-03-21

3.10.3 Tabel applicationusage

UserId	ApplicationId
1000	11011029-2005-0419-9734-636174696f6e
1000	11052762-2005-0419-1d87-636174696f6e
Order-it	11031381-2005-0419-0f02-636174696f6e

3.10.4 Tabel downloadtype

DownloadTypeId	Name
11083167-2005-0419-c4bb-616474797065	Programma
11084870-2005-0419-d56b-616474797065	Rapporten
11120884-2005-0419-1907-616474797065	Persoonlijke rapporten
11124128-2005-0419-674d-616474797065	Persoonlijke licentie

3.10.5 Tabel applicationdownloadtype

ApplicationId	DownloadTypeId
11011029-2005-0419-9734-636174696f6e	11083167-2005-0419-c4bb-616474797065
11011029-2005-0419-9734-636174696f6e	11084870-2005-0419-d56b-616474797065
11011029-2005-0419-9734-636174696f6e	11120884-2005-0419-1907-616474797065
11011029-2005-0419-9734-636174696f6e	11124128-2005-0419-674d-616474797065

3.10.6 Tabel updateinfo

Update InfoId	Application Id	Download TypeId	Build Number	Version	Date	Description	Active	Apply	Zipped	Priority	Check Version	File Name	Personal Update	Rename FileName	Installation Location
14065357-2005-0520-12bf-7465496e666f	11011029-2005-0419-9734-636174696f6e	11083167-2005-0419-c4bb-616474797065	20050520141042	1.79c	2005-05-20	LET OP: Deze versie van Bake-it functioneert niet met Windows 98!!! Wanneer u met Windows 98 draait	T	T	T	1	T	Bake-itUpdate179c.zip	F		
	11011029-2005-0419-9734-636174696f6e	11084870-2005-0419-d56b-616474797065		1.79	2005-05-20	Dit zijn de rapporten behorend bij Bake-it versie 1.79	T	T	T	1	T	Bake-it Reports179.zip	F		REPORTS
14065357-2005-0520-12bf-7465496e666f	11011029-2005-0419-9734-636174696f6e	11083167-2005-0419-c4bb-616474797065	20050520141042	1.75	2005-05-20	Persoonlijke rapporten Marti-IT	T	T	T	2	F	1000Maatwerk Marti-IT.zip	T		REPORTS\MAATWERK

3.10.7 Tabel userupdate

UserId	UpdateInfoId
1000	14065357-2005-0520-12bf-7465496e666f

3.11 Test set 1

3.11.1 Test scenario's

Test scenario 1: autorisatie functionaliteit zal worden uitgevoerd aan de hand van de gegevens uit test set 1.

3.11.2 Test gegevens

Database:
Zie § 4.1 voor database gegevens.

3.12 Test set 2

3.12.1 Test scenario's

De volgende test scenario's zullen worden uitgevoerd aan de hand van de gegevens uit test set 2:

- ✓ Test scenario 2: downloaden programma update
- ✓ Test scenario 3: downloaden rapporten update

3.12.2 Test gegevens

Database:
Zie § 4.1 voor database gegevens.

Bestanden:

- ✓ Bake-it programma update (bestand Bake-itUpdate179c.zip)
- ✓ Bake-it rapporten update (bestand Bake-itReports179.zip)

3.13 Test set 3

3.13.1 Test scenario's

Test scenario 4: downloaden persoonlijke update zal worden uitgevoerd aan de hand van de gegevens uit test set 3.

3.13.2 Test gegevens

Database:
Zie § 4.1 voor database gegevens.

Bestanden:

- ✓ Persoonlijke update voor maatwerk rapporten Marti-IT (bestand 1000MaatwerkMarti-IT.zip)

© 2005 Mathieu Brüning (20010019)

3.14 Test set 4

3.14.1 Test scenario's

Test scenario 5: downloaden persoonlijke licentie zal worden uitgevoerd aan de hand van de gegevens uit test set 4.

3.14.2 Test gegevens

Database:
Zie § 4.1 voor database gegevens.

Bestanden:

- ✓ Licentie voor Orto gebruiker met gebruikersnummer 1000 (bestand 1000.ini)
- ✓ Licentie voor Bake-it gebruiker met gebruikersnummer 2000 (bestand 2000.ini)

3.15 Test set 5

3.15.1 Test scenario's

Test scenario 6: uploaden back-up zal worden uitgevoerd aan de hand van de gegevens uit test set 5.

3.15.2 Test gegevens

Database:
Zie § 4.1 voor database gegevens.

Bestanden:

- ✓ Gegevensback-up voor Bake-it software (bestand BakeITBackup20050228-172927.Zip)
- ✓ Gegevensback-up voor Orto software (bestand OrtoBackup20050210-171127.Zip)

3.16 Test set 6

3.16.1 Test scenario's

De volgende test scenario's zullen worden uitgevoerd aan de hand van de gegevens uit test set 6:

- ✓ Test scenario 7: uploaden bestand
- ✓ Test scenario 8: uploaden map

3.16.2 Test gegevens

Database:
Zie § 4.1 voor database gegevens.

Bestanden:

- ✓ Willekeurig bestand (in dit geval CURRICULUM VITAE.doc)
- ✓ Willekeurige map (in dit geval D:\Test)

© 2005 Mathieu Brüning (20010019)

3.17 Test set 7

3.17.1 Test scenario's

Test scenario 6: uploaden standaard upload zal worden uitgevoerd aan de hand van de gegevens uit test set 7.

3.17.2 Test gegevens

Database:
Zie § 4.1 voor database gegevens.

Bestanden:

- ✓ Standaard upload sjabloon voor Bake-it automatische planning
  (C:\Program Files\Update-it\Uploads\UploadTemplates\Automatische planning.xml)

- ✓ De map van de automatische planning C:\Program Files\Bake-it\Autoplanning

© 2005 Mathieu Brüning (20010019)

4 Testresultaten

De resultaten, zoals letterlijk teruggekoppeld vanuit de gebruikersrichting, worden vermeldt in dit hoofdstuk. De paragrafen in het hoofdstuk representeren de test scenario's, waarin de resultaten van de gebruikers worden weergegeven. Voor het uitvoeren van de acceptatietests met betrekking tot het Marti-IT Update Centrum heb ik een vijftal gebruikers geselecteerd voor het uitvoeren van de testwerkzaamheden. Helaas bleek dat er één gebruiker niet in staat was om de testactiviteiten uit de voeren en zodoende ben ik met vier gebruikers van de Marti-IT software de acceptatietests gaan uitvoeren.

Deze vier gebruikers zijn:

- ✓ Rein Otten (Bake-it gebruiker)
- ✓ Henk Bos (Bake-it gebruiker)
- ✓ Mike van Meenen (Orto voor Windows gebruiker)
- ✓ Patrick Kamer (Orto voor Windows gebruiker)

4.1 Test scenario 1: Autorisatie functionaliteit

4.1.1 Waargenomen gedrag gebruiker Rein Otten

Alle testgevallen uitgevoerd zoals vastgesteld in het hoofdstuk testscenario's. Hierbij heb ik geen bijzonderheden ondervonden. Het werkte allemaal zoals het hoort.

4.1.2 Waargenomen gedrag gebruiker Henk Bos

Alle functies van test scenario één werken naar toebehoren.

4.1.3 Waargenomen gedrag gebruiker Mike van Meenen

De mogelijkheden met betrekking tot het inloggen gaan goed allen snap ik de "Order-it" knop niet goed.

4.1.4 Waargenomen gedrag gebruiker Patrick Kamer

Bij het autoriseren middels de Order-It knop kreeg ik deze melding:

4.1.5 Verwachte gedrag

1. Invoeren van een correct gebruikersnummer en een correct wachtwoord
   Weergeven van de basis interface met het hoofdmenu van het Marti-IT Update Centrum.

2. Invoeren van een correct gebruikersnummer en een incorrect wachtwoord
   Foutmelding

3. Invoeren van een incorrect gebruikersnummer en een correct wachtwoord
   Foutmelding

4. Het leeg laten van beide velden
   Foutmelding

5. Bewaren van het gebruikersnummer en wachtwoord
   Aanmaken *Identity* element in Identity.xml bestand, waarin gebruikersnummer en wachtwoord worden opgeslagen.

6. Het autoriseren middels “Order-it” knop
   Weergeven basis interface Marti-IT Update Centrum, met softwareprofiel ingesteld op alleen Order-it applicatie.

4.1.6 Verschil waargenomen t.o.v. verwachte gedrag

Geen verschillen waargenomen ten opzichte van verwachte gedrag.

4.1.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

Geen sprake van fouten.

© 2005 Mathieu Brüning (20010019)

4.2 Test scenario 2: Downloaden programma update

4.2.1 Waargenomen gedrag Rein Otten

De programma update voor Bake-it gedownload en geïnstalleerd. Dit werkte prima!

Echter, in het venster “Taken in uitvoering” is het alleen verwarrend dat er wel een checkbox staat, maar je deze checkbox niet kan gebruiken.

Wat betreft punt vier, ik heb Bake-it opgestart, vervolgens Update-it opgestart en de nieuwste versie van Bake-it gedownload. Update-it geeft dan geen melding dat Bake-it nog actief is tijdens het installatieproces.

4.2.2 Waargenomen gedrag Henk Bos

Alle functies van test scenario twee werken naar toebehoren.

4.2.3 Waargenomen gedrag Mike van Meenen

Het downloaden van programma updates werkt naar mijn idee goed. Alleen waneer Orto actief is loopt de boel vast, ik krijg dan een “invalid query error”.

De installatie heeft dan niet plaatsgevonden.

4.2.4 Waargenomen gedrag Patrick Kamer

Het downloaden van de nieuwste Orto programma update verliep vlekkeloos!

4.2.5 Verwachte gedrag

1. Selecteren van een programma update en het download ervan
   Na het selecteren van een update en het drukken op de “Downloaden updates” knop dient het voortgangsvenster te worden getoond. Hierin dienen de geselecteerde updates weer gegeven te worden. Na een druk op de “Download updates” knop wordt het downloadproces gestart.

2. Niet selecteren van een update en het downloaden ervan
   Foutmelding

3. Installeren van een programma update
   Het betreffende update bestand dient vanaf de locatie waarnaar het gedownload is uitgepakt te worden naar de bestemmingsmap.

4. Installeren van een programma update wanneer het programma nog actief is
   Foutmelding

4.2.6 Verschil waargenomen t.o.v. verwachte gedrag

4. Installeren van een programma update wanneer het programma nog actief is
   Er wordt geen foutmelding gegeven vanuit het Marti-IT Update Centrum.

4.2.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

4. Code opgezocht en aangepast, indien er nu een bestand in gebruik is wordt er een foutmelding gegeven vanuit het Marti-IT Update Centrum.

4.3 Test scenario 3: Download rapporten update

4.3.1 Waargenomen gedrag Rein Otten

Het downloaden en installeren van de rapporten voor Bake-it ging goed! Bij het uitvoeren van de testgevallen heb ik geen bijzonderheden aangetroffen.

4.3.2 Waargenomen gedrag Henk Bos

Alle functies van test scenario drie werken naar toebehoren.

4.3.3 Waargenomen gedrag Mike van Meenen

Werkt goed, wat me alleen niet helemaal duidelijk was, was het testgeval waar een report actief moest zijn. Moet het report dan geopend zijn in een reporteditor of moet het report actief zijn in Orto?

4.3.4 Waargenomen gedrag Patrick Kamer

Het downloaden van de nieuwste Orto rapporten update verliep ook vlekkeloos!

4.3.5 Verwachte gedrag

1. Selecteren van een rapporten update en het downloaden ervan
   Na het selecteren van de rapporten update en het drukken op de “Downloaden updates” knop dient het venster met de voortgang weergegeven te worden. Middels een druk op de knop “Downloaden bestanden” wordt het download proces dan in werking gezet.

2. Installeren van een rapporten update
   Na het downloaden van de rapporten update naar de map Bake-it\updateit\Rapporten, dient het bestand uitgepakt te worden naar de daarvoor bestemde map, zoals aangegeven in de database middels het veld InstallationLocation.

3. Installeren van een rapporten update wanneer één of meerdere rapporten nog actief zijn
   Foutmelding

4.3.6 Verschil waargenomen t.o.v. verwachte gedrag

3. In plaats van dat er een foutmelding vanuit het Update Centrum zou worden gegeven werd er een exceptie van het .NET framework gegeven.

4.3.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

3. Code opgezocht en aangepast, indien er nu een bestand in gebruik is wordt er een foutmelding gegeven vanuit het Marti-IT Update Centrum.

© 2005 Mathieu Brüning (20010019)

4.4 Test scenario 4: Downloaden persoonlijke update

4.4.1 Waargenomen gedrag Rein Otten

Ik heb mijn maatwerk rapporten overgehaald en geïnstalleerd. Dit leverde geen problemen op.

4.4.2 Waargenomen gedrag Henk Bos

Alle functies van test scenario vier werken naar toebehoren.

4.4.3 Waargenomen gedrag Mike van Meenen

Het maatwerk bestand wordt op de juiste manier gedownload en weergegeven. Alles werkt goed wat betreft test scenario 4.

4.4.4 Waargenomen gedrag Patrick Kamer

Het downloaden en installeren van persoonlijke Orto rapporten verliep ook goed. Geen fouten tegengekomen.

4.4.5 Verwachte gedrag

1. Testen of de juiste persoonlijke updates worden weergegeven
   De persoonlijke updates dienen weergegeven te worden in het download venster, onder de download types welke beginnen met de tekst ‘persoonlijke ’.

2. Selecteren van een persoonlijke update en het downloaden ervan
   Het aanvinken van persoonlijke updates en het klikken op de “Downloaden updates” knop dient ervoor te zorgen dat het venster met betrekking tot de voortgang van het verzendproces wordt weergegeven. Na het klikken op de knop “Downloaden updates” in het voortgangsvenster wordt het update bestand gedownload.

3. Installeren van een persoonlijke update
   Na het downloaden van het update bestand naar de map <applicatie>\updateit\<download type> wordt het bestand uitgepakt naar de daarvoor bestemde map.

4.4.6 Verschil waargenomen t.o.v. verwachte gedrag

Geen verschillen ten opzichte van verwachte gedrag.

4.4.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

Geen fouten gevonden.

© 2005 Mathieu Brüning (20010019)

4.5 Test scenario 5: Downloaden persoonlijke licentie

4.5.1 Waargenomen gedrag Rein Otten

Testgevallen uitgevoerd en een licentie gedownload en geïnstalleerd. Hierbij geen ik geen bijzonderheden aangetroffen.

4.5.2 Waargenomen gedrag Henk Bos

Alle functies van test scenario vijf werken naar toebehoren.

4.5.3 Waargenomen gedrag Mike van Meenen

Het downloaden van het licentiebestand werkt goed. Het bestand wordt op de juiste plaats geïnstalleerd. Wanneer ik Orto vervolgens opstart wordt de juiste informatie weergegeven.

4.5.4 Waargenomen gedrag Patrick Kamer

Het downloaden en installeren van een persoonlijke licentie leverde voor mij geen problemen op.

4.5.5 Verwachte gedrag

1. Inventariseren van persoonlijke licentie voor de ingelogde gebruiker en de weergave ervan in het download venster
   De licentie voor de ingelogde gebruiker dient weergegeven te worden in het downloadvenster, zodat de gebruiker in staat is op de update te selecteren.

2. Selecteren van een persoonlijke licentie en het downloaden ervan
   Het selecteren van een persoonlijke licentie update dient mogelijk te zijn in het download venster door het vinkveld aan te klikken. Vervolgens dient de gebruiker op “Downloaden updates” te drukken, zodat het voortgangsvenster getoond wordt. Indien de gebruiker in het voortgangsvenster op de “Downloaden updates” knop drukt dient de licentie update gedownload te worden.

3. Het installeren en hernoemen van een persoonlijke licentie
   Wanneer een licentiebestand gedownload naar de applicatiemap is dient het bestand hernoemt te worden naar “licentie.ini”.

4.5.6 Verschil waargenomen t.o.v. verwachte gedrag

Geen verschillen ten opzichte van verwachte gedrag.

4.5.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

Geen fouten gevonden.

4.6 Test scenario 6: Uploaden van een back-up

4.6.1 Waargenomen gedrag Rein Otten

Bij het uploaden van back-ups heb ik geen bijzonderheden ontdekt. Het werkte allemaal naar toebehoren.

4.6.2 Waargenomen gedrag Henk Bos

Bij het selecteren van twee bestanden voor het uploaden ervan krijg ik deze melding:
Fout bij het uploaden van bestand: BakeItBackup20050601-100755.Zip!
System.IndexOutOfRangeException.

4.6.3 Waargenomen gedrag Mike van Meenen

De functies voor het uploaden van gegevensback-ups werken goed.

4.6.4 Waargenomen gedrag Patrick Kamer

Het uploaden van een back-up leverde opnieuw geen problemen op!

4.6.5 Verwachte gedrag

1. Het inventarisatieproces van de beschikbare back-ups
   De gemaakte back-ups van de Marti-IT applicaties van de ingelogde gebruiker dienen weergegeven te worden in het venster voor het uploaden van back-ups.

2. Het selecteren van een back-up en het uploaden ervan
   Na het selecteren van een back-up en het drukken op de “Volgende >” knop dient het voortgangsvenster te worden getoond. Hierin dient de geselecteerde back-up weer gegeven te worden. Na een druk op de “Uploaden back-ups” knop wordt het verzendproces gestart.

3. Het selecteren van meerdere back-ups en het uploaden ervan
   Na het selecteren van meerdere back-ups en het drukken op de “Volgende >” knop dient het voortgangsvenster te worden getoond. Hierin dienen de geselecteerde back-ups weer gegeven te worden. Na een druk op de “Uploaden back-ups” knop wordt het verzendproces gestart.

4.6.6 Verschil waargenomen t.o.v. verwachte gedrag

Geen verschillen ten opzichte van verwachtte gedrag.

4.6.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

Code doorlopen van de functionaliteit van het voortgangvenster om de fout te lokaliseren. De fout echter nog niet gevonden.

© 2005 Mathieu Brüning (20010019)

4.7 Test scenario 7: Uploaden van een bestand

4.7.1 Waargenomen gedrag Rein Otten

Ik kan in het hoofdmenu niet vinden waar ik bestanden of mappen kan uploaden. Dit kan ik wel vinden in het menu Uploads | Bestanden / mappen. Tijdens het inpakken van de map Label-it/Reports krijg ik echter een foutmelding.

4.7.2 Waargenomen gedrag Henk Bos

Ik kan niet vinden waar ik een bestand moet selecteren.

4.7.3 Waargenomen gedrag Mike van Meenen

Testen van inpakken van een bestand is volgens mij niet mogelijk omdat de keuze niet gegeven wordt in het hoofdmenu.

4.7.4 Waargenomen gedrag Patrick Kamer

Het inpakken en uploaden van een bestand leverde voor mij geen problemen op.

4.7.5 Verwachte gedrag

1. Selecteren van een bestand en het inpakken ervan
   Middel de “Bladeren” knop rechts van het veld “Bestand” kan een bestand geselecteerd worden. Vervolgens dient er op de “Pak de geselecteerde map of bestand in” knop gedrukt te worden om het bestand in te pakken. Er dient dan een .zip bestand aangemaakt te worden in de map Update-it \ Uploads \ <bestandsnaam>. Informatie omtrent het ingepakte bestand dient weergegeven te worden in de onderste sectie van het venster.

2. Selecteren van geen bestand en het inpakken ervan
   Foutmelding

3. Uploaden van een ingepakt bestand
   Indien een bestand is ingepakt kan middels de “Volgende >” knop het voortgangsvenster worden getoond. Een druk op de “Uploaden bestand / map” knop zorgt er dan voor dat het bestand verzonden wordt.

4.7.6 Verschil waargenomen t.o.v. verwachte gedrag

1. Selecteren van een bestand en het inpakken ervan
   Bij het inpakken van een bestand kreeg gebruiker Rein Otten een foutmelding, in plaats van dat het bestand werd ingepakt.

4.7.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

1. Selecteren van een bestand en het inpakken ervan
   Code doorlopen om de fout te lokaliseren. Nadat ik de fout niet heb kunnen lokaliseren ben ik aan de slag gegaan met het reproduceren van de fout. Het is me echter niet gelukt om de fout te reproduceren.

© 2005 Mathieu Brüning (20010019)

4.8 Test scenario 8: Uploaden van een map

4.8.1 Waargenomen gedrag Rein Otten

Bij het inpakken van een bestand krijg ik net als bij het inpakken van een map een foutmelding.

4.8.2 Waargenomen gedrag Henk Bos

Ik kan niet vinden waar ik een map moet selecteren.

4.8.3 Waargenomen gedrag Mike van Meenen

Testen voor het inpakken van mappen is volgens mij niet mogelijk omdat de keuze niet gegeven wordt in het hoofdmenu.

4.8.4 Waargenomen gedrag Patrick Kamer

Geen problemen ontdekt bij het uploaden van een map.

4.8.5 Verwachte gedrag

1. Selecteren van een map (zonder submappen) en het inpakken ervan
   Middel de “Bladeren” knop rechts van het veld “Map” kan een map geselecteerd worden. Het vinkveld “Submappen” dient dan afgevinkt te zijn, om zo een map in te pakken zonder submappen. Vervolgens dient er op de “Pak de geselecteerde map of bestand in” knop gedrukt te worden om de geselecteerde map in te pakken. Er dient dan een .zip bestand aangemaakt te worden in de map Update-it \ Uploads \ <bestandsnaam>. Informatie omtrent het ingepakte bestand dient weergegeven te worden in de onderste sectie van het venster. Het ingepakte bestand mag dan geen submappen bevatten.

2. Selecteren van een map (met submappen) en het inpakken ervan
   Middel de “Bladeren” knop rechts van het veld “Map” kan een map geselecteerd worden. Het vinkveld “Submappen” dient dan aangevinkt te zijn, om zo een map in te pakken met submappen. Vervolgens dient er op de “Pak de geselecteerde map of bestand in” knop gedrukt te worden om de geselecteerde map in te pakken. Er dient dan een .zip bestand aangemaakt te worden in de map Update-it \ Uploads \ <bestandsnaam>. Informatie omtrent het ingepakte bestand dient weergegeven te worden in de onderste sectie van het venster. Het ingepakte bestand dient dan de mapstructuur behouden te hebben.

3. Selecteren van geen map en het inpakken ervan
   Foutmelding

4. Uploaden van een ingepakte map
   Indien een map is ingepakt kan middels de “Volgende >” knop het voortgangsvenster worden getoond. Een druk op de “Uploaden bestand / map” knop zorgt er dan voor dat het bestand verzonden wordt.

4.8.6 Verschil waargenomen t.o.v. verwachte gedrag

1. Selecteren van een map (zonder submappen) en het inpakken ervan
   Bij het inpakken van een bestand kreeg gebruiker Rein Otten een foutmelding, in plaats van dat de map werd ingepakt.

4.8.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

1. Selecteren van een bestand en het inpakken ervan
   Code doorlopen om opnieuw de fout te lokaliseren. Nadat ik de fout opnieuw niet heb kunnen lokaliseren ben ik aan de slag gegaan met het reproduceren van de fout. Het is me echter niet gelukt om de fout te reproduceren.

© 2005 Mathieu Brüning (20010019)

4.9 Test scenario 9: Uploaden standaard upload

4.9.1 Waargenomen gedrag Rein Otten

4.9.2 Waargenomen gedrag Henk Bos

Ik kan niet vinden waar ik een bestand moet selecteren.

4.9.3 Waargenomen gedrag Mike van Meenen

Ik kan alleen maar Standard uploads selecteren een keuze voor ander bestand is er niet.

4.9.4 Waargenomen gedrag Patrick Kamer

Geen problemen ontdekt bij het uploaden van een map.

4.9.5 Verwachte gedrag

1. Selecteren van een standaard upload en het uploaden ervan
   Na het selecteren van een standaard upload en het drukken op de “Volgende >” knop dient het voortgangsvenster te worden getoond. Hierin dient de geselecteerde standaard upload weer gegeven te worden. Na een druk op de “Uploaden standaard uploads” knop wordt het verzendproces gestart.

2. Selecteren van geen standaard upload en het uploaden ervan
   Foutmelding

3. Selecteren van meerdere standaard uploads en het uploaden ervan
   Na het selecteren van meerdere standaard uploads en het drukken op de “Volgende >” knop dient het voortgangsvenster te worden getoond. Hierin dienen de geselecteerde standaard uploads weer gegeven te worden. Na een druk op de “Uploaden standaard uploads” knop wordt het verzendproces gestart.

4.9.6 Verschil waargenomen t.o.v. verwachte gedrag

Geen verschillen ten opzichte van verwachtte gedrag.

4.9.7 Uitgevoerde handelingen t.b.v. het achterhalen van de fout

Geen fouten gevonden.

© 2005 Mathieu Brüning (20010019)

5 Conclusie

Samenvattend kan ik stellen dat de gebruikers van de Marti-IT software zeer te spreken waren over het nieuwe Marti-IT Update Centrum. Vrijwel alle testgebruikers vonden het systeem stukken verbeterd op het gebied van uitstraling en gebruikersvriendelijkheid. “Met name door de toevoeging van het hoofdmenu was ik in staat om op eenvoudige wijze te navigeren door de applicatie”, aldus Patrick Kamer.
De geïmplementeerde functionaliteiten van het systeem werken over het algemeen goed, waarbij een aantal uitzonderingen waren, zoals het niet weergeven van een foutmelding indien een bestand nog in gebruik is en het niet kunnen inpakken van bestanden bij Rein Otten.
Tijdens de uitvoeren van de acceptatietests is er tevens gebleken dat het voor niet iedere gebruiker duidelijk is dat alleen de belangrijkste functionaliteiten kunnen worden aangeroepen vanuit het hoofdmenu. Echter, in de handleiding van het Marti-IT Update Centrum wordt echter concreet beschreven hoe alle functionaliteiten aangeroepen kunnen worden, vanuit het hoofdmenu en het reguliere menu.

Bij de vraag of het systeem klaar was voor ingebruikname antwoordde drie van de vier testgebruikers dat het systeem in gebruik zou kunnen gaan. Samenvattend kan ik dus stellen dat het uitkomsten van de acceptatietests positief zijn.

© 2005 Mathieu Brüning (20010019)