



EINDVERSLAG BACHELORPROJECT

RENÉ VERHEIJ

TECHNISCHE UNIVERSITEIT DELFT

FACULTEIT ELECTROTECHNIEK, WISKUNDE EN INFORMATICA

19 JUNI 2008



Stagebegeleider
Ir. P. van Delft

Opdrachtgever
Ir. F. van Vloten

Begeleiding TU Delft
Dr. A.H.J. Oomes
Ir. M. Sepers

VOORWOORD

Het bachelorproject van de bachelor Media en Informatica aan de TU Delft dient als afsluiting van de bachelorfase. Het project heeft een doorlooptijd van circa drie maanden waarbij verschillende fasen van het software-ontwikkeltraject worden doorlopen vanaf het functioneel ontwerp tot en met een werkend systeem. De opdracht die ik heb uitgevoerd voor dit project is het maken van een webshop voor het bedrijf Eco-movement te Amsterdam door middel van een opensource pakket. Deze webshop zal de verkoop van elektrische scooters verder moeten automatiseren. Daarnaast is een deel van de opdracht ook kennisvergaring van het gebruikte opensource pakket voor het werkgevend bedrijf, Mars Media. Dit verslag zal dan ook gaan over het volledige verloop van het project om deze opdrachten succesvol uit te voeren.

INHOUDSOPGAVE

1.	Analyse	
1.1	Opdrachtschrijving	4
1.2	Requirements	6
1.3	Analyse	9
1.4	Planning	10
2	Eerste fase	12
2.1	Analyse Magento	12
2.2	Afsluiting fase	17
3	Tweede fase	18
3.1	3D weergave van producten	18
3.2	Betalingsmodule	19
3.3	Uitbreiding betalingsproces	21
4	Evaluatie	25
5	Conclusie	26
	Referenties	27
	Bijlage A: Gebruikerstest	
	Bijlage B: Opdrachtschrijving	
	Bijlage C: Onderzoeksverslag	

OPDRACHTOMSCHRIJVING

In de context van de opdrachtomschrijving zal ik eerst een uitleg geven van de betrokken bedrijven.

YAIKS B.V.

Het onderliggende bedrijf waar de andere beide bedrijven mee betrokken zijn is Yaiks B.V. Dit bedrijf kan het beste omschreven worden als een incubator, een broedplaats voor ondernemers & ondernemingen, die tot doel heeft om veelal jonge startende bedrijven te ondersteunen om een gezonde, goed draaiende onderneming te worden.

MARS MEDIA

Een van deze ondernemingen is MARS media. MARS media voorziet haar klanten van op maat gemaakte en gebruikersvriendelijke internetmedia. Zij houden zich bezig met het ontwerpen en ontwikkelen van websites, applicaties voor databeheer, content management en e-commerce systemen. Dit is het bedrijf waar ik kom te werken als stagiair, en kan dus gezien worden als werkgever.

ECO-MOVEMENT

De andere onderneming die betrekking heeft op mijn opdracht is Eco-movement. "Eco-movement ontwikkelt en introduceert innovatieve, creatieve en kwalitatief hoogwaardige producten en diensten die daadwerkelijk bijdragen aan een milieubewuste houding bij bedrijven en consumenten en een duurzame leefomgeving". Op dit moment bieden zij daartoe voornamelijk diverse elektrische scooters aan. Dit is het bedrijf met de behoefte aan automatisering en kan dus gezien worden als opdrachtgever.

DE OORSPRONKELIJKE OPDRACHT

Het verkopen van deze elektrische scooters gebeurt uitsluitend via internet, en wel via de site van Eco-movement. Op dit moment kan dat alleen via het invullen van een eenvoudig bestel formulier, waarna Eco-movement contact opneemt voor de verdere afhandeling. Eco-movement wil dit graag automatiseren. Bovendien heeft Eco-movement de intentie om meerdere producten naast de elektrische scooters via het web aan te gaan bieden. MARS media is het 'collega bedrijf' van Eco-movement en zal deze automatisering gaan uitvoeren. Zij hebben een keer eerder een webshop moeten ontwikkelen, en hebben daartoe zelf een eenvoudige webshop geprogrammeerd. Nu is er de verwachting dat er naast Eco-movement in de nabije toekomst meerdere uitgebreide webshops gemaakt zullen worden, en dus zoekt MARS media naar een geschikt opensource e-commerce pakket. De volledige eerste opdrachtomschrijving is terug te vinden als bijlage A en was komt op het volgende neer:

René mag een eco-webwinkel ontwikkelen met state of the art technieken, zoals geïntegreerde betalingssystemen, 3D weergave van producten, flexibel backoffice en mogelijkheden tot CO2 compensatie. Het uitgangspunt bij de eco-webwinkel is dat het toegankelijk is voor iedereen. De eco-webwinkel heeft respect voor milieu en mens. De eco-webwinkel heeft een jonge, frisse en sympathieke uitstraling. De eco-webwinkel heeft als bijzondere eigenschap dat er gecompenseerd kan worden voor de CO2 uitstoot van de duurzame producten. René kan hierbij een model ontwikkelen waarbij producten gekocht kunnen worden en tegelijkertijd gecompenseerd kunnen worden voor hun CO2 uitstoot. Voordat René begint met het bouwen van de eco-webwinkel mag hij een onderzoek doen waarbij de huidige beschikbare technieken in kaart gebracht worden. Op grond van dit markt onderzoek zal de werkwijze en techniek bepaald worden (hierbij kan gedacht worden aan SMS, Creditcard, iDEAL, Flash, Opensource, etc.).

DE UITEINDELIJKE OPDRACHT

Tijdens de eerste week van het project in het 4^e kwartaal, na de onderzoeksfase, bleek tijdens de verschillende

gesprekken met de opdrachtgever, werkgever en begeleider niet alleen dat deze opdrachtoomschrijving wat verder gespecificeerd moest worden, maar dat het idee van de webshop zelf nog in vorming was. Specifieke vragen leidde tot grote veranderingen van opdracht. De grootste verandering die naar boven kwam was dat er maar 1 type scooter en wat andere eco-producten verkocht hoeven te worden via de Eco-movement website en dat het wenselijk was een aparte webshop voor de andere elektrische scooters te ontwikkelen, die niet zichtbaar verbonden was met Eco-movement, maar via hetzelfde backend als de Eco-movement webshop te beheren is. Daarnaast werd duidelijk dat het design intern in MARS media ontwikkeld zou worden en werden de eisen van MARS media verder gespecificeerd. Bovendien werd de nadruk op CO2 compensatie verschoven naar de optie een compenserend cadeautje te kiezen tijdens het betalingsproces, wat verder besproken zal worden bij de requirements. Deze gesprekken vrij hergeformuleerd en samengevat komt de opdracht op het volgende neer:

Ontwikkel twee webshops voor Eco-movement met behulp van een opensource pakket. De eerste webshop wordt de Elektrische Scooter Winkel (ESW) en de tweede webshop wordt de Eco-movement Winkel (EMW). Beide webshops dienen aan hetzelfde backend gekoppeld te zijn. Daarnaast dient opgedane kennis over het werken met het opensource pakket zo uitgebreid mogelijk gedocumenteerd te worden voor toekomstig gebruik van het pakket binnen MARS media.

REQUIREMENTS

In de eerste week van het project heb ik veel met zowel de opdrachtgever als mijn begeleider en beide gesproken om de requirements van het project vast te leggen.

Dit leidde tot de volgende samenvatting:

ELEKTRISCHESCOOTERWINKEL (ESW)

Deze website wordt zoveel mogelijk een opzichzelfstaande winkel. Klanten dienen zo weinig mogelijk en het liefst nergens de naam Eco-movement terug te vinden. Er zullen scooters van verschillende merken verkocht worden en elk merk heeft waarschijnlijk 1 scooter. Elke scooter van een dergelijk merk heeft weer verschillende kleuren waar direct uit gekozen dient te worden bij het bestellen van het product. Tijdens het bekijken en bestellen van een scooter zijn de relevante producten zichtbaar. Dit zijn items als bijvoorbeeld een koffer, een zonnepaneel of een poncho. De site is er op gericht hét verkooppunt van elektrische scooters in Nederland te worden. Scooters kunnen worden vergeleken met elkaar waaruit direct duidelijk wordt wat er hetzelfde is en vooral, wat er verschilt. Bezoekers kunnen reacties / recensies plaatsen bij een scooter en kunnen terecht op het forum voor vragen of discussies over de scooters. Bij het bestellen van producten op deze webshop krijgen de klanten voor het betalen de optie een natuurvriendelijk cadeautje te kiezen op kosten van de website ter compensatie van de productie en het vervoer van de gekochte producten. Bij deze aanbieding staat de melding dat de producten ook los verkrijgbaar zijn op de webshop van Eco-movement. Een instelbaar percentage van het bedrag van de bestelling bepaald de maximale waarde van het te kiezen cadeau. De te kiezen cadeaus zijn gemakkelijk via de backend te beheren. De bezoekers kunnen zich via de webshop aanmelden voor de nieuwsbrief van de ESW. De verkopers die deze site bezoeken moeten onderscheiden kunnen worden van normale klanten en krijgen korting op producten. Deze webshop krijgt zijn eigen URL.

ECO-MOVEMENT WEBSHOP (EMW)

Deze webshop dient simpeler van aard te zijn dan de ESW. Hier wordt één van de scooters aangeboden die ook op de ESW wordt aangeboden, namelijk die scooter waarvan Eco-movement importeur is. Daarnaast worden hier diverse eco-producten aangeboden in verschillende categorieën als T-shirts, boeken, etenswaren, etc. Er is ook mogelijkheid tot aanmelden voor de nieuwsbrief. Er zijn ook relevante producten. Er is geen productvergelijking. Er is wel een compensatie cadeau vergelijkbaar met de ESW. Deze webshop wordt onderdeel van de website www.eco-movement.com.

Beide winkels dienen via het backend onderhouden te kunnen worden. Van het aanmaken en beheren van producten en links tussen producten, tot het instellen van de kortingsregels en het onderhouden van de klanten groep 'verkopers'.

Ten slotte wordt bij de opdracht verwacht dat alle opgedane kennis over het gekozen opensource software pakket wordt gedocumenteerd. Dit zal worden opgedeeld in een verzameling van filmpjes, tutorials en eigen documentatie voor zowel de designer als de gebruiker van de backend. Bovendien worden eventuele uitbreidingen op de software uiteraard goed gedocumenteerd.

In overleg met de begeleider van MARS media is hier het volgende MoSCoW diagram uitgekomen:

REQUIREMENT	MUST	SHOULD	COULD	WOULD
Gemakkelijk producten toe kunnen voegen	X			
Gemakkelijk product 3d weergeven of gemakkelijk filmpje maken		X		
Beide webshops via één backend te beheren	X			
Optie tot direct online betalen	X			
Search Engine Optimisation		X		
Zelf kortingsregels kunnen beheren (bijvoorbeeld alle dealers korting geven)		X		
Onderscheid in gebruikers tussen klanten en dealers zelf kunnen beheren		X		
Product opties zelf opties kunnen beheren (wel of geen zonnepaneel bij de scooter)		X		
Bij producten zelf beschikbare kleuren en bijbehorende afbeeldingen kunnen beheren	X			
Verzekering afsluiten als optie bij scooter			X	
Voorraad beheer van producten		X		
Facturen worden automatisch gemaakt	X			
'related products' weergeven in de winkel en zelf deze links tussen producten kunnen beheren.			X	
Gebruikers kunnen producten vergelijken			X	
Veelgekochte producten overzicht op home van front			X	
Gebruikers kunnen reacties plaatsen bij producten			X	
Klanten van de verschillende frontends moeten gescheiden blijven	X			
De Elektrische Scooter Webshop bevat een forum				X
Vlak voor betaling kan de klant een gratis natuur compensatieproduct kiezen. De keuze is gebaseerd op het bedrag van de bestelling.	X			

Tabel 1. MoSCoW diagram requirements

Een mooi leermoment kwam echter in de 8^e week, ongeveer op tweederde van het project. De opdrachtgever kwam toen nog vrij onverwacht met de volgende set requirements aanzetten:

- Er komt een forum
- Er komt een wiki
- Per product komt er:
 - o Een deeplink naar een topic in het forum
 - o Een overzicht van youtube filmpjes
 - o Een overzicht van overige foto's die door de gebruikers is te beheren
 - o Een overzicht van recensies
 - o Een overzicht van documentatie van het product

De opdrachtgever was zich er wel van bewust dat door de problemen met Magento die verderop besproken worden de eerste set van requirements al krap haalbaar zou worden. Aangezien Magento al aan een hoop van de oude requirements voldeed, werd besloten werd ons eerst te richten op de meest belangrijke functionaliteiten, namelijk:

- Het kunnen betalen via een Nederlandse 'payment provider'.

- Het uitbreiden van het betalingssysteem met een optie tot het kiezen van een natuur compenserend cadeautje
- De weergave van producten in 3d
- De implementatie van de lay-out

Alle overige functionaliteiten, zo werd besloten, zullen opnieuw bekeken worden na het afronden van deze bovenstaande functionaliteiten.

ANALYSE

DE ONDERZOEKSFASE

Tijdens deze fase heb ik onderzocht welk opensource e-commerce software-pakket het beste gebruikt kan worden. Uit dit onderzoek kwam het opensource pakket van Magento. Een verslag van deze onderzoeksfase vind u terug in bijlage C.

3D PRODUCTWEERGAVE

Naast het onderzoek in het 3^e kwartaal heb ik ook verderop tijdens het project nog twee andere kleine onderzoeken verricht. Ten eerste werd mij gevraagd na te denken en te zoeken naar verschillende mogelijkheden om producten 3d weer te geven in de webshop. Verschillende opties zijn hierbij bekeken waaronder de 3d weergave laten maken door hiervoor gespecialiseerde bedrijven. Echter de eenvoudigste en goedkoopste optie is het achter elkaar plaatsen van foto's en de gebruiker hier door middel van een flash animatie met de muis doorheen laten gaan. Al snel was hier een opensource flashscript genaamd Yofla voor gevonden, en al snel na overleg met de opdrachtgever had deze een set opeenvolgende foto's waarop een scooter één keer ronddraaide. Het resultaat was voor een snelle test al erg goed en dus werd nadat ik ten slotte het script, dat de set foto's in XML formaat inleest, succesvol in Magento had geïmplementeerd besloten verder te gaan met deze optie. De opdrachtgever heeft daartoe een kamer omgebouwd tot foto studio zodat er nagenoeg geen achtergrond meer zichtbaar is op de foto's.

BETALINGSWIJZE

In de requirements fase was duidelijk geworden dat er een Nederlandse betalingsmodule ontwikkeld moest worden voor Magento. Rekeninghoudend met toekomstig gebruik waren de eisen van MARS media daarbij dat er betaald moet kunnen worden met creditcard, eenmalige machtiging en iDEAL. Aangezien het succesvol en veilig betalen met al deze opties een omvangrijk iets is waar ook continue veel in veranderd, doen bedrijven dit in de praktijk nooit zelf, maar gaat het betalingsverkeer via een zogenaamde 'payment service' [1]. Aangezien MARS media nog niet met een payment service samenwerkte (tot nu toe werd alleen zelfstandig iDEAL betalingen ondersteund), was het aan mij de vraag een klein onderzoekje te doen naar mogelijke payment services. Uit een prijsvergelijking kwamen 3 mogelijke kandidaten: Buckaroo, Ogone en Multipay. Geen van deze drie bleek bezig te zijn met een plug-in voor Magento, maar aangezien Buckaroo de goedkoopste was en een redelijke service leek met een prima interface en duidelijke implementatiehandleiding werd in overleg besloten met Buckaroo te gaan werken.

PLANNING

Hier volgt een omschrijving van de verschillende fases zoals die doorlopen zullen worden tijdens het project.

1. In deze fase zal bekeken worden wat er precies gedaan moet worden om de opdracht te volbrengen en met welke software daar het beste bij past. Vervolgens zal de gekozen opensource software geïnstalleerd worden en zal er een eerste concept voor de ESW gemaakt worden. Uitbreidingen op de opensource software die nodig zijn voor het eindproduct worden nog buiten beschouwing gelaten. Aangezien er zoveel onbekende factoren meespelen wordt deze fase als verkennende fase beschouwd en zal na deze fase met de opgedane kennis de planning voor de 2^e fase opnieuw bekeken worden.
 - a. Analyse en requirements engineering.
 - b. Ontwerp en implementatie van eerste concept van de webshop
 - c. Gebruikerstesten en feedback verzamelen over het eerste concept
 - d. Documentatie bijhouden

2. In deze fase zal bekeken worden wat er nog moet veranderen aan de huidige functionaliteiten van het concept. Daarnaast zal bekeken worden welke functionaliteiten het softwarepakket nog niet biedt die dus zelf geprogrammeerd moeten worden. Ook wordt de lay-out van de webshops in deze periode geïmplementeerd. Als dit gedaan is wordt er een nieuwe planning gemaakt voor deze fase. Na de implementatie van bovenstaande functionaliteiten en de lay-out zal de webshop getest worden, niet alleen functionele testen door mijzelf, maar ook wederom gebruikerstesten. In deze laatste test en feedback periode zullen eventuele problemen en missende functionaliteiten direct worden verholpen en hierna worden de webshops online gezet. Deze fase eindigt met het afmaken van de documentatie voor Mars Media, waarvan opslaan van opgedane kennis over het opensource pakket de belangrijkste taak is.
 - a. Analyse, requirements bijstellen en faseplanning maken.
 - b. Wijzingen uitvoeren en implementatie nieuwe functionaliteiten.
 - c. Test en feedback periode. Problemen gelijk verhelpen.
 - d. Documentatie afmaken

3. Als er tijd over is zal deze laatste fase volgen na het online brengen van de webshop. Omdat Mars Media bij een succes van de opdracht vaker het gekozen opensource pakket in de toekomst zal willen gebruiken wordt in deze fase gekeken in hoeverre het gebruik van dit pakket geautomatiseerd kan worden en in hoeverre functionaliteiten van de backend geïmplementeerd kunnen worden in het bestaande CMS.
 - a. Analyse en requirements engineering
 - b. Ontwerp
 - c. Implementatie
 - d. Testfase
 - e. Documentatie

Opmerkingen bij de fase keuzes

Ten tijde van het schrijven van deze fases in de fase planning is de keuze voor het software pakket Magento al gemaakt en is gebleken dat Magento nog geen betalingsmodule heeft voor Nederlandse gebruikers (het pakket is net nieuw). Dit wetende leek het mij handiger de implementatie van dit soort uitbreidingen van het softwarepakket te plannen voor 2^e fase, daar waar het nodig is en bovendien is tegen die tijd de functionaliteit misschien wel al beschikbaar. Bovendien wordt de vormgeving van de webshops in het bedrijf intern verzorgd en is afgesproken dat het design eind week 4 af is. Daarom staat de implementatie van de lay-out pas gepland voor de 2^e fase.

De oorspronkelijke planning om deze fases te doorlopen die gemaakt werd in de eerste week zag er als volgt uit:

week	15	16	17	18	19	20	21	22	23	24	25
	07/04	14/04	21/04	28/04	05/05	12/05	19/05	26/05	02/06	09/06	16/06
Fase \ data	-	-	-	-	-	-	-	-	-	-	-
	10/04	18/04	25/04	02/05	09/05	16/05	23/05	30/05	06/06	13/06	20/06
Analyse en requirements	1a										
Ontwerp en impl.		1b	1b								
Test en feedback			1c								
Documentatie			1d	1d							
Analyse en requirements				2a							
Ontwerp en impl.				2b	2b	2b	2b				
Test en feedback						2c	2c	2c			
Documentatie							2d	2d	2d		
Analyse en requirements									3a		
Ontwerp en impl.									3b	3b	
Test en feedback										3c	3c
Documentatie											3d

Tabel 2. Oorspronkelijke planning

EERSTE FASE

In overleg met mijn stagebegeleider is besloten de eerste fase als oriëntatie fase voor Magento te beschouwen. In deze fase heb ik dan ook alle opties van het pakket uitgeplozen, alle tutorials online gevolgd en zelf in andere context geprobeerd toe te passen. Ook heb ik de ontwikkelingen in het forum en het blog van Magento gevolgd en heb ik mij zo goed mogelijk bekend gemaakt met de code. Ik zal nu dan ook eerst mijn ervaringen met Magento bespreken aangezien dit effect had op alle verdere werkzaamheden van het project.

ANALYSE MAGENTO

Magento is een opensource e-commerce webapplicatie gemaakt door het bedrijf Varien. De ontwikkeling van Magento begon in januari 2007 en de eerste publieke versie werd uitgebracht in augustus 2007. 17 maanden na de start de ontwikkeling werd de deadline die vanaf de eerste dag gesteld was voor de 1.0 versie gehaald en zo werd op 31 maart Magento 1.0 uitgebracht.

Over het algemeen heb ik tijdens het project net als vele andere community leden regelmatig problemen gehad met het werken met Magento, waardoor weinig tijd over bleef voor de gestelde 'could' of 'would' requirements uit het MoSCoW diagram. In de volgende secties zal ik dan ook uitgebreid Magento bespreken om deze problemen uit te leggen.

FRAMEWORK

Magento maakt gebruik van het Zend Framework (ZF). Het Zend Framework is een opensource, object georiënteerde webapplicatie framework geschreven in PHP 5. De gedachte achter het Zend Framework is dat gebruikers componenten kunnen hergebruiken daar waar ze maar willen zonder zich zorgen te hoeven maken over de benodigdheid van andere ZF componenten. Er is daardoor geen standaard 'design pattern' dat gebruikers van het framework moeten volgen, hoewel het framework wel diverse componenten aanbiedt voor de in ZF applicaties veelgebruikte Model View Controller (MVC) architectuur. Door dit vrije design, kunnen ZF componenten relatief gemakkelijk gebruikt worden naast componenten van andere PHP webapplicatie frameworks [2].

Bovenop het Zend Framework gebruik Magento zijn eigen framework, het Varien Framework (VF). Een van de belangrijkste kenmerken van dit framework is dat elke klasse een extensie is van het Varien_Object, dat de 'magic methods' van PHP ondersteunen. Dit betekent dat dit object onder andere de functies __get() en __set() ondersteund, die automatisch worden aangeroepen als bepaalde leden of methoden van de klasse worden aangeroepen die niet zichtbaar of gedeclareerd zijn in de klasse zelf. Met andere woorden kan elke klasse allerlei attributen hebben.

ARCHITECTUUR

Magento zelf maakt gebruik van het singleton design pattern. In software engineering wordt het singleton pattern gebruikt om het aantal instanties van elk soort object te limiteren tot 1. Het idee hier achter is dat er nooit onnodig meerdere kopieën gemaakt worden van een instantie en dat er nooit verwarring kan zijn welke instantie je gebruikt. Het singleton pattern wordt door sommige ook wel gezien als een anti-pattern, waarbij geclaimd wordt dat het een eufemisme is voor globale variabelen. [3] In Magento worden alle klassen aangeroepen via de hoofdklasse: Mage. In deze klasse staan verschillende methodes om verschil te maken in verschillende soorten klassen zoals 'models' en 'resource models'. Een voorbeeld is de volgende veelgebruikte methode:

```
//Retrieve model object singleton  
Mage::getSingleton('module/model');
```

De String die bij dit soort methodes meegegeven wordt, wordt gebruikt als een soort afkorting die beschrijft welke klasse je wilt terug krijgen. Het gedeelte voor de schuine streep zegt in welke package de klasse zich bevindt, en het gedeelte erachter zegt welke klasse precies bedoeld wordt. Dit kan echter nog zorgen voor de nodige verwarring zoals zal blijken uit de volgende voorbeelden:

```
Mage::getSingleton('checkout/session')
```

Geeft volgens verwachting een instantie terug van de package checkout en het model session:

```
Mage/checkout/model/session.php
```

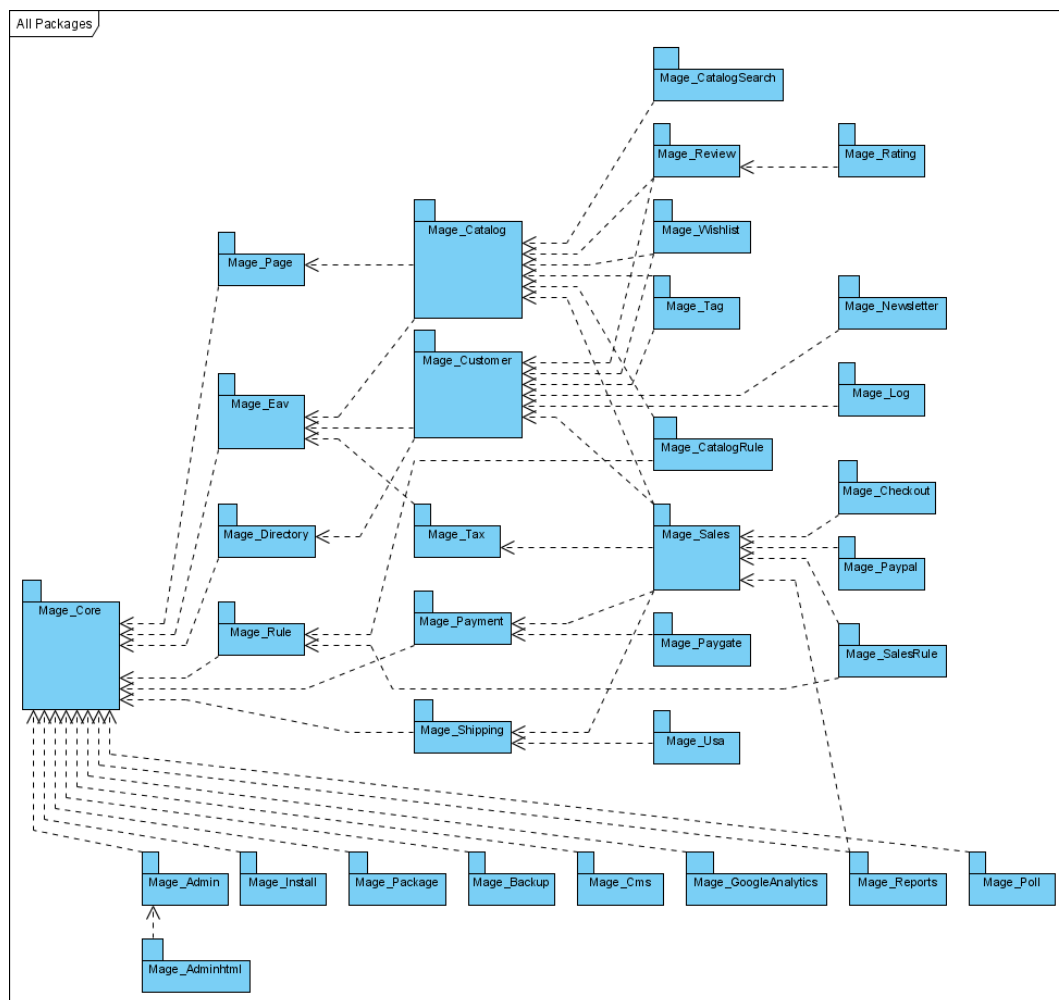
```
Mage::getSingleton('sales/order_address')
```

Het liggende streepje zegt dat er een map 'order' is in de 'model' map en dat daar het bestand 'address.php' te vinden is, wat ook klopt:

```
Mage/sales/model/order/address.php
```

```
Mage::getSingleton('catalog/product_collection')
```

Dit is een voorbeeld waarbij de verwarring optreedt. In de trend van het vorige voorbeeld zou je zeggen dat er een bestand is onder de locatie *Mage/catalog/model/product/collection.php*. Dit is echter niet het geval. En sinds zich meer dan 1000 PHP bestanden in de 'catalog' module bevinden is het niet eenvoudig uit te vinden in welk bestand de klasse zich bevindt die wordt teruggegeven door de methode. De meest gebruikte methode door de community om alsnog de klasse te vinden is door alle bestanden te doorzoeken op de String 'product_collection'.



Figuur 1. Package indeling Magento

SPLITSING VAN CODE EN LAYOUT

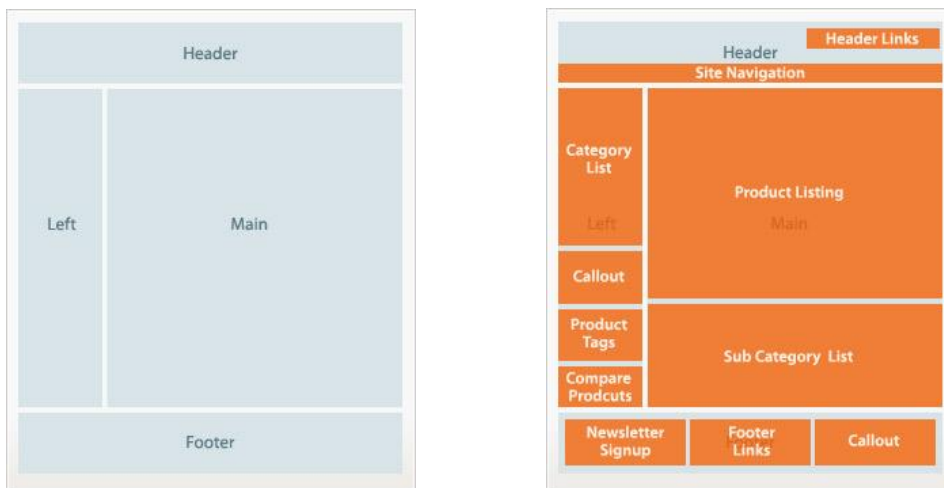
De lay-out van Magento websites is volledig gescheiden van de code. Om dit te bereiken heeft Magento de pagina opgedeeld in de volgende onderdelen

- *Structuur blokken*

Dit zijn de meest basis onderdelen van een lay-out. Het betreft de grote vlakken van een pagina als 'header', 'left' en 'main'. Deze blokken zijn slechts omhulsels van andere blokken en bevatten geen content op zichzelf.

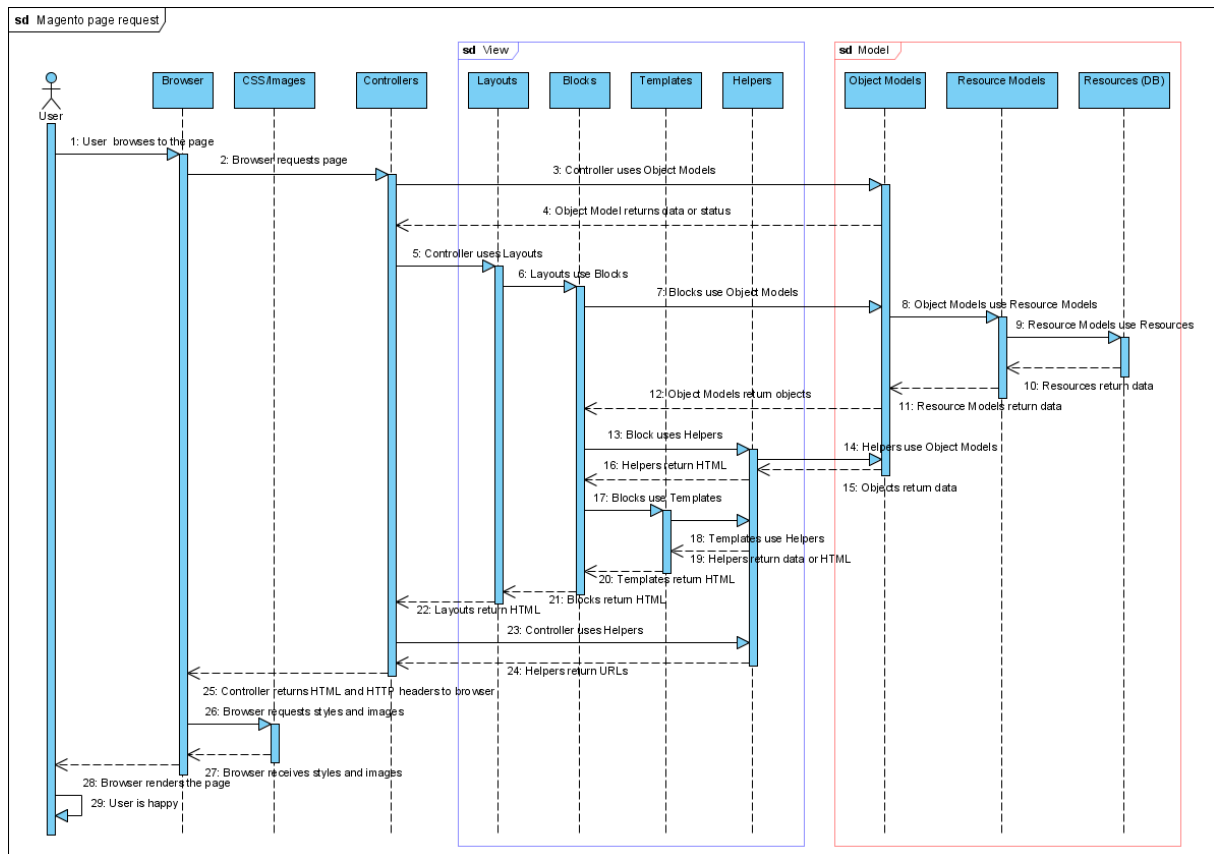
- *Content blokken*

De echte content kan gevonden worden in de content blokken. Deze blokken worden geplaatst in een van de structuur blokken door middel van XML updates. Dit wil zeggen dat er geen bestand is waar alle content blokken van een pagina staan gedefinieerd. Een pagina wordt opgebouwd door meerdere modules die op dat moment actief zijn. Deze modules hebben eigen XML bestanden, waarin voor verschillende 'states' van de module controller beschreven staat welke content blokken met welke 'templates' waar geladen moeten worden.



Figuur 2. Structural Blocks (links) en content Blocks (rechts)

De XML files gebruiken hiervoor een set eenvoudige tags, die voornamelijk bestaat uit module handlers, blok tags, verwijzingen naar 'parent' of 'child' blokken, en acties. Hierbij moet gezegd worden dat voor de aanroep van de templates wederom gebruik wordt gemaakt van Strings als een aanduiding van het pad naar de template waarbij regelmatig dezelfde verwarring optreedt als bij het aanroepen van instanties van klassen zoals hierboven besproken.



Figuur 3. Sequentie diagram Magento

COMPLEXITEIT

Het Magento pakket telt na installatie meer dan 13.000 bestanden en is meer dan 100MB groot, waarvan meer dan de helft code is. Een gemiddelde klasse heeft meer dan 5 bovenliggende klassen en de hoeveelheid code die wordt nagelopen om een pagina te laden is dusdanig veel dat het merkbaar de laadtijd van pagina's beïnvloedt. Hoewel Magento goede progressie maakt in het verbeteren van de laadtijden in de afgelopen updates blijft er nog 1 belangrijk probleem over voor ontwikkelaars. De meest populaire methode om met onbekende softwarepakketten om te gaan blijft namelijk een 'debug mode' van programma's als Zend Development Studio of Eclipse. In zo een 'debug mode' kan je met zogenaamde breakpoints het laden van een pagina onderbreken en stap voor stap door de gebruikte klassen heen bekijken welke code vervolgens aangeroepen wordt tijdens het laden van ene pagina. Hiermee zou het eerder omschreven probleem van erachter komen wat voor instantie er terugkomt via singleton methodes verholpen zijn, ware het niet dat Magento dusdanig groot is dat het bereiken van een breakpoint via de 'debug mode' gemiddeld meer dan een uur kost om te laden, mits het programma niet vastloopt.

DOCUMENTATIE EN ONLINE SUPPORT

Maar zelfs zonder de mogelijkheid de 'debuggen' en met de soms lastig te vinden klassen zou er goed gewerkt kunnen worden met Magento, mits er goede documentatie beschikbaar is. De immense hoeveelheid code is in ieder geval niet gedocumenteerd. De code is zeer zelden van commentaar voorzien en het standaard gegenereerde commentaar boven de functies en de classes is daar waar het al aangepast is, niet veel meerzeggend dan datgene wat je al wist. Dit alles is op zich geen ramp als er buiten de code om wel goede documentatie beschikbaar is. En wie zich gaat verdiepen in Magento als optie voor een webshop krijgt de impressie dat die support in ieder geval goed geregeld is. De website staat vol met video tutorials en een flink aanbod wiki's waarin uitgelegd wordt hoe je jou aanpassingen in Magento voor elkaar krijgt. Het zoeken naar een overzicht van alle klassen en hun methodes leverde dan wel weinig op, maar dit had na dat Magento 1.0 uitkwam, 'hoge prioriteit'. Wie echter aan de slag gaat met Magento en de tutorials gaat volgen zal merken dat

de meeste tutorials meer dan een half jaar oud zijn en eigenlijk niet veel meer dan een globaal idee geven van wat er moet gebeuren omdat de meeste precieze informatie niet meer klopt. Vervolgens zijn er voor al die kwesties wel onderwerpen in het forum waar er wild op los gediscussieerd wordt. Maar veel onderwerpen komen uiteindelijk niet veel verder dan speculaties en scherpe vragen aan het Magento team omdat niemand van de community het antwoord kan vinden. Als het Magento team antwoord geeft is dat vervolgens gelijk zo goed en uitgebreid dat het hele probleem opgelost is, het aantal discussies en problemen is echter zo groot dat het erop lijkt dat het team bij verre na niet alle problemen kan bekijken.

CONCLUSIE

Van de gestelde requirements door MARS media voor een opensource e-commerce pakket waren er twee requirements die je alleen kan inschatten voordat je echt aan het werk gaat met een pakket, namelijk dat pakket makkelijk aan te passen is en dat het goed gedocumenteerd is. Beide deze twee requirements leek Magento, zeker volgens eigen zeggen, zeer goed aan te voldoen. De verouderde tutorials en wiki's zorgen ervoor dat je zelf moet kijken in de Magento code hoe je aanpassingen kan maken. De manier waarop het singleton pattern geïmplementeerd is zorgt er niet alleen voor dat development programma's niet begrijpen wat voor instantie een variabele is, maar ook dat je zelf vaak met relatief veel moeite moet achterhalen wat voor instantie het is. De complexiteit van het pakket schuift alle debug mogelijkheden van tafel en dus rest de programmeur niets anders dan uitgebreide zoekacties uit te voeren en zelf debug informatie uit te printen om er stapje voor stapje achter te komen wat er gebeurd, of alles werkt en hoe het verder moet. Dit alles zorgt voor een zeer trage voortgang van eigen ontwikkeling aan het pakket waarbij ik zelf regelmatig een hele dag de code heb doorgelopen om uiteindelijk tot niet meer dan 10 regels werkende code te komen. Niet omdat de informatie die mij tot beschikking staat zo lastig te begrijpen is, maar simpelweg omdat er niet genoeg informatie beschikbaar is en het verkrijgen van de benodigde informatie zeer tijdsintensief werk is.

Magento heeft zoals gezegd zijn eigen gesteld deadline voor de 1.0 versie gehaald die al stond vanaf de eerste dag van ontwikkeling. Een mooie prestatie op het eerste gezicht. Maar met mijn huidige inzicht zet ik toch de nodige vraagtekens bij het uitbrengen van een dergelijk product met zo weinig relevante documentatie.

AFSLUITING FASE

Omgooi planning 2^e fase. Zoals gepland is na de eerste fase een nieuwe planning gemaakt. Dit bleek een juiste planning geweest te zijn, omdat het inschatten van tijden betreft werken met zo een groot opensource pakket als Magento pas goed in te schatten zijn als je er daadwerkelijk goed mee aan de slag bent gegaan. De problematiek met Magento die ondervonden werd in de eerste fase liet zien dat het programmeren van nieuwe modules voor Magento een zeer tijdrovende klus zou worden. Dit heeft gezorgd voor de volgende planning aan het begin van de 2^e fase en het schrappen van de 3^e fase.

week		18	19	20	21	22	23	24	25
Onderdeel	Stadium	28/04 – 02/05	05/05 – 09/05	12/05 – 16/05	19/05 – 23/05	26/05 – 30/05	02/06 – 06/06	09/06 – 13/06	16/06 – 20/06
Lay-out	Implementatie								
3D weergave producten	Analyse en ontwerp								
	Implementatie								
	Testen en doc.								
Betalingsmodule	Analyse en ontw.								
	Implementatie								
	Testen en doc.								
Cadeaumodule	Analyse en ontw.								
	Implementatie								
	Testen en doc.								
Documentatie									

Tabel 3. Nieuwe planning na 1e fase

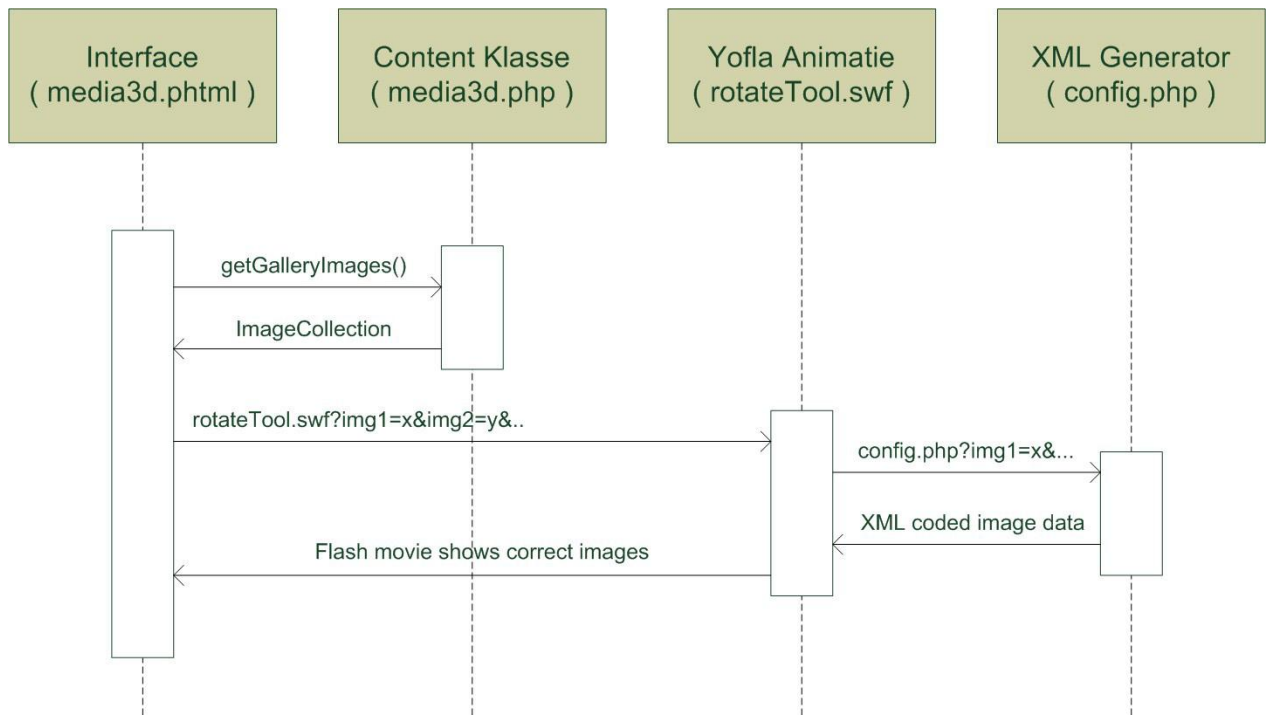
3D WEERGAVE PRODUCTEN

ANALYSE EN ONTWERP

Nadat in de onderzoeksfase het opensource project Yofla gekozen was techniek voor de 3d weergave van producten is bekeken hoe dit verder in Magento geïntegreerd en geautomatiseerd kon worden. Wat betreft de integratie is er een nieuw 'content block' aangemaakt genaamd media3d. Yofla gebruikt een XML bestand om de afbeeldingen in te lezen en is op die manier dus gemakkelijk aanpasbaar. In dit geval is dat echter niet flexibel genoeg, omdat ook de afbeeldingen die gebruikt zullen worden door de beheerder te beheren zijn. Er zijn verschillende mogelijkheden bekeken om dit probleem op te lossen:

- Magento genereert eenmalig een XML bestand bij elke wijziging in de afbeeldingenset door de beheerder. Deze oplossing kost het minste rekenkracht als de bezoeker uiteindelijk een pagina gaat bekijken.
- Yofla wordt aangepast zodat het geen XML bestand meer gebruik maar zijn gegevens door middel van GET variabelen uit de URL haalt die door Magento wordt aangeroepen om Yofla aan te roepen. (GET is een methode waarmee variabelen in een URL verborgen via het http protocol verstuurd kunnen worden)[4].
- Yofla haalt de gegevens van de locaties van de afbeeldingen wel uit de URL die door Magento wordt aangeroepen en geeft deze vervolgens mee in de URL aanroep voor het gebruikte XML bestand. Dit is in dit geval geen XML bestand maar een PHP bestand dat de XML met behulp van de meegestuurde variabelen genereert precies zoals Yofla verwacht.

De laatste optie lijkt hierin de meest omslachtige, maar omdat zowel de eerste als de tweede optie veel tijdrovender zijn om te implementeren en omdat het verschil in rekenkracht zeer minimaal is, is gekozen voor optie 3. De interactie tussen de onderdelen ziet er dan als volgt uit: De interface wordt veranderd zodat bij elk product de interface van media3d wordt aangeroepen. Deze roept de functie `getGalleryImages()` aan van zijn bijbehorende klasse en bouwt daarmee een lange URL op waarin deze gegevens worden verwerkt. Met deze URL wordt het flashbestand van Yofla aangeroepen, met daarin via de GET methode de gegevens van de te gebruiken afbeeldingen. Yofla roept op zijn beurt een PHP bestand aan met deze gegevens en gebruikt de response ervan om de afbeeldingen in te laden en te laten zien. Dit PHP bestand geeft dan ook XML code terug, precies zoals Yofla de informatie over de locatie van de afbeeldingen verwacht. Ondanks dat het hier om integratie en communicatie tussen verschillende systemen en talen gaat en niet tussen verschillende klassen is het volgende sequentie diagram gemaakt om de interactie tussen de verschillende onderdelen weer te geven.



Figuur 4. Sequentie diagram van de interactie tussen de verschillende onderdelen van Magento en Yofla

IMPLEMENTATIE

Aan de hand van het ontwerp is de implementatie van deze functionaliteit vrij eenvoudig zonder verdere problemen verlopen. En dus kan de beheerder nu gebruik maken van de normale functies van Magento om verschillende afbeeldingen te uploaden en hiervan de volgorde te beheren via het beheerderpaneel, waarna de afbeeldingen (mits het product al als zichtbaar is ingesteld) direct als 3d filmpje te bekijken en te besturen zijn op de website. Er heeft zich echter wel een ander zijdelings probleem voorgedaan tijdens de implementatie. Uit de tweede set van requirements bleek dat de opdrachtgever namelijk dat er niet alleen een 3d productweergave per product zou komen, maar dat er ook andere foto's onder die weergave aan te klikken zijn. Aangezien elke afbeelding uit de afbeeldingenset van een product al werden ingelezen voor de 3d weergave moest er een 2^e afbeeldingen set komen om dit te realiseren. In Magento zit de optie zo een 'imageSet' als attribuut toe te voegen aan een soort product. Deze optie bleek echter rare resultaten te geven als het werd gebruikt om een 2^e 'imageSet' aan te maken. De afbeeldingen die je aan deze tweede set toevoegde werden namelijk wel tijdelijk weergegeven maar bleken vervolgens niet opgeslagen te zijn als je de pagina herlaadde. Nadat ik dit aan heb gegeven op het forum van Magento, hebben meerdere mensen de bug erkend en heeft Magento toegezegd een 2^e 'imageSet' mogelijk te maken in de volgende release.

TESTEN

De functionaliteit is qua code vrij simpel, het testen was daarom ook vrij simplistisch. De gebruikte code is getest op het volledig of deels ontbreken van data en het werken met foutieve afbeeldinglocaties.

BETALINGSMODULE

ANALYSE EN ONTWERP

Uit de analyse was Buckaroo als beste 'Payment Provider' uit de bus gekomen. Direct na deze analyse zijn dan ook stappen ondernomen om in contact te komen met Buckaroo en van hen de juiste informatie te krijgen om een module te ontwikkelen voor Magento. De contactpersoon van Buckaroo had wel van Magento gehoord en

dacht dat het bedrijf dat normaal de plug-ins voor Magento ontwikkelde al bezig was met een Magento plug-in. Na nog wat correspondentie waarin ik vroeg om een eventuele datum waarop de plug-in gereed zou zijn bleek echter dat er nog helemaal geen plug-in voor Magento gemaakt werd. De contactpersoon was nu plotseling zeer bereid mee te werken, leverde dus veel informatie aan en maakte een test account voor MARS media aan. Het op tijd beginnen met contact op te nemen met Buckaroo bleek dus een zeer handige tip van mijn begeleider, want voordat de juiste informatie beschikbaar was, was er aardig wat tijd overheen gegaan. Maar het verkrijgen van de juiste informatie bij Buckaroo verbleekt bij het verkrijgen van de juiste informatie van Magento. De tutorial die in de maand na het uitkomen van Magento 1.0 beschikbaar was over het maken van een betalingsmodule ging over een 7 maanden oude release van Magento. Wie de tutorial volgde, kwam niet veel verder dan stap 1, tot grote ergernis van vele gebruikers. Doordat er behalve wat simpele methodes voor de V.S. nog geen uitgebreide betalingsmodules waren die gebruik maakte van een externe applicatie / website voor andere landen was er erg veel vraag naar goede documentatie over het opzetten van een betalingsmodule. Magento leek aanvankelijk te druk met nieuwe features voor de volgende versie en/of dacht dat de community er zelf wel uit zou komen. Dit was echter niet het geval en toen een tijdlang niemand erin slaagde een betalingsmodule te ontwikkelen en de ergernis groeide lijkt Magento besloten te hebben er zelf maar mee aan de slag te gaan. Ze ontwikkelden een betalingsmodule voor iDEAL en zette deze 6 weken na het uitkomen van versie 1.0 (nog steeds zonder enige documentatie of commentaar) online. De timing van het online zetten van deze iDEAL module was echter zeer gunstig voor mij. Ik wist precies hoe een module van de kant van Buckaroo moest gaan werken en volgde al weken op het forum de berichten van andere programmeurs die er niet uit kwamen om een betalingsmodule in Magento te ontwikkelen. Net in de week dat ik mijn afwachende status zou moeten gaan veranderen om mij bij de vele andere programmeurs te voegen die de code indoken om een oplossing te vinden, kwam Magento met de iDEAL plug-in. Het aanpassen van deze plug-in om zo een nieuwe plug-in voor Buckaroo te ontwikkelen bleek voor mij en vele anderen een veel gemakkelijkere taak dan het opzetten van een complete nieuwe module.

Het communiceren met Buckaroo kan op verschillende manieren. Na overleg met Buckaroo is besloten de communicatie op de volgende manier te laten verlopen. Aan het eind van het betalingsproces schakelt Magento door naar de site van Buckaroo en stuurt de benodigde gegevens mee door middel van de POST methode (POST is de tegenhanger van GET waarbij variabelen via het http protocol verstuurd kunnen worden, bijvoorbeeld via HTML formulieren). In deze waarden bevindt zich naar de betalings specifieke waarden ook een digitale handtekening. Deze digitale handtekening wordt opgebouwd door middel van het MD5 algoritme met wat betalingswaarden en een geheime sleutel die via het beheerderpaneel van Buckaroo is ingesteld. Het Message Digest Algorithm 5 (MD5) is een veel gebruikt cryptografische hashfunctie met een 128-bit hashwaarde. Het is opgenomen in de internetstandaard RFC 1321 en is een veel gebruikte techniek in vele veiligheidstoepassingen op het internet [5]. Een andere waarde die meegestuurd is verteld Buckaroo of het gaat om een creditcard betaling, een eenmalige machtiging of een iDEAL transactie. Buckaroo geeft de juiste interface weer aan de gebruiker en er zijn vervolgens verschillende manieren waarop het werk van Buckaroo erop zit die kunnen worden ingedeeld in 3 categorieën: *succes*, *reject* en *error*. Deze hebben de volgende betekenis.

<i>Succes</i>	Een succesvolle betaling
<i>Reject</i>	Een betalingsfout waarbij de bank de betaling weigert. Bijvoorbeeld doordat het saldo van de bankrekening van de klant te laag is.
<i>Error</i>	De betaling werd geannuleerd door een actie van de gebruiker. Bijvoorbeeld door te klikken op annuleren of doordat er te lang gewacht werd.

Het probleem bij dit soort betalingen is dat deze informatie niet meer naar Magento kan gestuurd worden als de gebruiker het venster al direct sluit na betaling (25% van de gebruikers doet dit). Er is daarom besloten om gebruik te maken van het pushmechanisme van Buckaroo. Dit mechanisme loopt elke 15 minuten alle

betalingen af en stuurt bij elk soort resultaat – ook het afgesloten venster – de informatie naar Magento door de gegevens van de betaling naar een vooraf ingestelde pagina van Magento te sturen. Hierbij wordt weer een digitale handtekening gebruikt zodat Magento zeker is dat de informatie van Buckaroo komt. Dezelfde gegevens worden ook al gestuurd als de gebruiker nog gewoon wacht tot de Buckaroo interface terug naar Magento schakelt, maar daar wordt in dit geval niets mee gedaan behalve een passend bericht voor de gebruiker weer te geven over het resultaat van de betaling.

Verder is besloten tijdens de ontwerpfase dat de instellingen zoals de geheime sleutel, het klantnummer en andere vaste waarden van de module in het beheerderpaneel aan te passen moeten zijn.

IMPLEMENTATIE

Een nadeel van de gekozen pushmethode kwam tijdens implementatie omhoog. Aangezien Buckaroo een pagina van Magento aanroept om de waardes door te sturen en niet de klant, beschikt Buckaroo niet over de sessie die de gebruiker had. Aangezien Magento alles via sessies doet was het uiteindelijk wat meer werk om uit te vinden hoe dit het beste geïmplementeerd kon worden.

TESTEN

Bij alle mogelijke interacties en alle mogelijke bijbehorende responsecodes die kunnen worden nagebootst is gecontroleerd of de juiste resultaten bij Magento werden opgeslagen. Daarnaast is er intern in het bedrijf iemand de opdracht gegeven te proberen een normaal illegale actie uit te voeren, zoals zelf waardes naar het pushsysteem te sturen om Magento te laten denken dat er betaald is.

De complete module is ondertussen als Bèta versie aan de Magento community aangeboden en is op het moment van schrijven al meerder malen gedownload.

UITBREIDING OP BETALINGSPROCES

ANALYSE EN ONTWERP

Een van de meest gewenste functionaliteiten was het uitbreiden van het betalingsproces met een extra stap, namelijk het kiezen van een cadeautje. De beheerder wil hierbij verschillende producten in de cadeaucategorie kunnen beheren en in het betalingsproces geldt een instelbaar percentage van het totaalbedrag als het maximale bedrag voor de kosten van het cadeau. Het ontwerp bestaat daarom slechts uit een set handelingen die verricht moeten worden: Op een passend moment in het betalingsproces die producten uit de cadeaucategorie weergeven die lagere kosten hebben dan de maximale cadeauekosten. Nadat de gebruiker een keuze heeft gemaakt (of aangegeven heeft geen cadeau te willen) dient de keuze opgeslagen te worden door het product als gratis toe te voegen aan de winkelwagen. Aangezien al deze stappen betreft implementatie volledig afhankelijk zijn van de mogelijkheden van Magento, valt in dit geval de rest van het uitvinden en analyseren samen met de implementatie.

IMPLEMENTATIE

Deze handelingen bleken echter vrij lastig te implementeren. Magento is zo flexibel mogelijk opgebouwd zodat je alle klassen kan 'overloaden' met eigen code en gemakkelijk eigen nieuwe modules kan schrijven. Toch bleek dit niet te gelden voor het betalingsproces. Het volledige betalingsproces van Magento vindt plaats op 1 pagina, waarin alle opvolgende stappen door middel van AJAX worden ingeladen. Het goed beveiligde betalingsproces is daarmee een van de pronkstukken van Magento, maar ontnemt daarmee de gebruikers de kans het betalingsproces aan te passen. De opzet van de klassen van het betalingsproces volgt een volledig andere indeling dan de rest van Magento, de enorme stukken javascript die zijn gebruikt zijn onoverzichtelijk en wederom niet voorzien van commentaar en als je tenslotte één regeltje verandert waar de module het niet mee eens is een witte pagina zonder inhoud en zonder foutinformatie het gevolg. Het lijkt daarmee of het niet

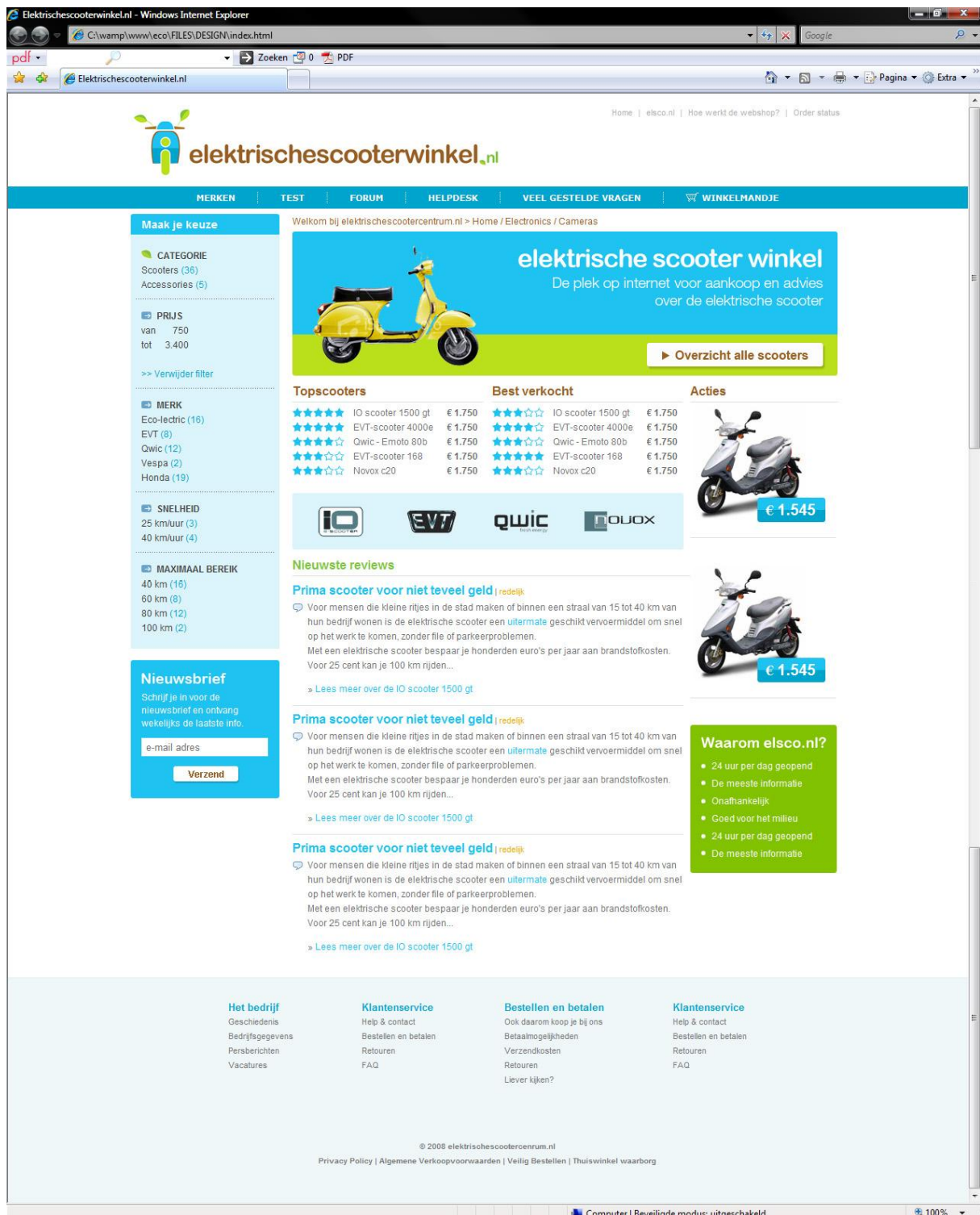
de bedoeling van Magento is geweest om gebruikers de betalingsmodule te laten aanpassen. Gelukkig was ik niet de enige die een dergelijke functionaliteit wilde implementeren. Met mij wilde tientallen andere gebruikers een stap toevoegen aan het betalingsproces, waaronder enkele geïnteresseerden in een 'free gift step', waaronder met name één serieuze programmeur uit China, met de bijnaam *ChineseDream*. Op mijn verzoek heb ik van deze persoon het opzetje overgenomen wat hij al gemaakt had voor deze functionaliteit. Het proces wat volgde was een zeer moeizaam proces waarbij elke keer kleine vooruitgangen werden geboekt door een van ons beide. Regelmatig werd de laatste versie van de code via de mail doorgestuurd. Aangezien ik er full time aan werkte kwamen de vooruitgangen voornamelijk van mijn kant, hoewel 'ChineseDream' ook meerdere malen in onze nacht mijn werk heeft bekeken en zeer bruikbare verbeteringen aangebracht. Een van de duidelijk te omschrijven problemen was dat er zowel een kostenprijs als een betaalprijs bekend moest zijn (de kosten voor de eigenaar van de webshop en het bedrag 0.00 voor de klant). Het bleek namelijk niet mogelijk handmatig een product met de prijs '0.00' aan de winkelwagen toe te voegen als er een andere prijs in de database stond. Toen ook het attribuut 'special price' niet bleek te werken zoals wij verwachtten is er besloten een nieuw attribuut aan te maken waarin de kosten voor de beheerder opgeslagen dienen te worden.

TESTEN

Toen alle functionaliteiten eenmaal geïmplementeerd waren is de code ter beschikking gesteld in het forum, om het te laten beoordelen door andere gebruikers. Er was verder nog niemand in geslaagd een volledige nieuwe betalingsstap te integreren in het betalingsproces van Magento en dus werd de code al snel opgepakt en door velen bekeken. Op wat kleine efficiëntie aanpassingen na is de code vooralsnog ongewijzigd en op dit moment wordt voornamelijk de verschillende te ondernemen handelingen besproken met als doel gezamenlijk een wiki aan te maken over het toevoegen van een stap aan de betalingsmodules in het algemeen.

LAYOUT

De lay-out van alle projecten binnen MARS media word verzorgd door één designer. Zo ook de lay-out van de elektrischescooterwinkel.nl. Is dus weinig te zeggen over de analyse en het ontwerp. Er is echter wel wat te zeggen over de planning en de samenwerking met de ontwerper. In de eerste planning stond de oplevering van de hele lay-out in HTML en CSS in overleg met de ontwerper gepland voor 29 april. Precies op tijd voor de 2^e fase en aangezien het implementeren van de lay-out relatief eenvoudiger is dan het implementeren van nieuwe modules leek het ons ook tijdens het plannen van de 2^e fase het verstandigst dit direct aan het begin te doen om zo misschien al nieuwe kennis op te doen over Magento die benodigd zou zijn bij volgende werkzaamheden. De lay-out bleek echter aan het begin van deze fase nog niet af te zijn en werd daarom twee weken naar voren geplaatst. Ik begon ondertussen al aan de 3d weergave van producten en toen dit af was kon ik verder aan de betalingsmodule omdat de lay-out nog niet af was. Bovendien zorgde de besprekingen van de ontwerper met de opdrachtgever over de lay-out voor wat wijzigingen in de requirements zoals eerder besproken. De lay-out werd keer op keer verder naar achter geplaatst en uiteindelijk is in de 10^e week op 11 juni de lay-out van de 'homepage' aangeleverd. De implementatie van deze pagina is ondertussen volledig voltooid, het is echter op het moment van schrijven (19 juli, 4 dagen voor de presentatie), nog wachten op de lay-out van de andere pagina's van de website. Er zijn dus tijdens de presentatie hoogstwaarschijnlijk verschillende onderdelen op de website nog te zien in de oorspronkelijke Magento lay-out.



Figuur 5. Design homepage elektrischescooterwinkel.nl

Bijlage B bevat een gebruikerstest ontworpen om te bekijken of de interface van de website goed functioneert. Na het uitstellen van het design is overwogen de gebruikerstest uit te voeren met de huidige Magento lay-out. In overleg met de ontwerper bleek echter dat de interface die werd ontworpen dusdanig anders zou worden dat een dergelijke test niet erg relevant zou zijn. Op dit moment staat gepland de test alsnog uit te voeren na het implementeren van het design in de komende twee weken.

OVERIG

Naast de in dit hoofdstuk besproken onderdelen zijn er nog vele andere functionaliteiten geïmplementeerd voor de elektrischescooterwinkel.nl. Hierbij doel ik bijvoorbeeld op het weergeven van de laatste beoordelingen op de homepage, het weergeven van de best verkochte scooters het aanpassen van de menustructuur of het toevoegen van een merk en bijbehorende aan elk product. Deze functionaliteiten zijn echter relatief eenvoudig te implementeren vergeleken met de andere besproken onderdelen in dit hoofdstuk. Bovendien zijn er voor deze handelingen genoeg soortgelijke voorbeelden te vinden op de website van Magento, waardoor de implementatie ervan niet meer was dan het aanpassen van een voorbeeld naar eigen wensen.

EVALUATIE

PERSOONLIJK

Het hele project was voor mij leerzamer dan verwacht. Ik had al de nodige ervaring met het ontwikkelen van webshops en het werken bij een webdesign bedrijf, maar toch bleek er nog een hoop te leren. Vooral het meewerken aan de ontwikkeling van nieuw opensource pakket heb ik erg interessant en leerzaam gevonden.

MAGENTO

Het werken met zo een groot en nieuw opensource pakket heeft zoals beschreven een hoop leerzame momenten met zich meegebracht. Ten eerste zou ik de eerste fase bij zo een groot project verkorten aangezien echt begrip pas komt bij het aanpakken van één specifiek probleem in plaats van het hele pakket in zijn algemeenheid te bekijken. Bovendien zou ik bij dit soort opensource software pakketten voortaan nog eerder in het proces de onderliggende frameworks volledig uitpluizen, om te voorkomen dat er in een later stadium nog onduidelijkheden zijn die later betrekking blijken te hebben op het framework, zoals in dit project het geval was met de 'Magic Functions' van het Zend Framework. Ten slotte heb ik gemerkt dat het handig is om al vroegtijdig mee te discussiëren op het forum zodat je in een later stadium meer response krijgt op je vragen.

MARS MEDIA

De samenwerking met mijn begeleider, de opdrachtgever en de ontwerper zijn zeer goed verlopen. Het enige waar hier volgens mij iets op aan te merken is zijn de afspraken rond de lay-out. Samen met het verkrijgen van informatie van Buckaroo heeft het mij geleerd dat je al in een zeer vroeg stadium het verkrijgen van de benodigde onderdelen van anderen in gang moet zetten en in de gaten moet houden om te zorgen dat je ze ook op tijd ter beschikking hebt. Na de miscommunicatie betreft de eerste oplevering van de lay-out is er echter weinig meer verkeerd gegaan in het contact met de ontwerper. Er is meerdere keren per week besproken hoe ver de lay-out was en in hoeverre ik nog andere werkzaamheden kon verrichten zodat de ontwerper in de drukte goed de prioriteiten kon stellen. Het werverweken van het ontwerp tot een HTML en CSS pagina door een extern persoon duurde echter langer dan verwacht waardoor de implementatie van de lay-out alsnog erg krap uit kwam om het einde van het project te halen.

TU DELFT

De samenwerking met de TU is naar mijn idee ook goed verlopen. Er is mij altijd duidelijk geweest wat er van mij verwacht werd en het contact met mijn begeleider de heer A.H.J. Oomes is altijd soepel verlopen en regelmatig van belang geweest voor het verkrijgen van nuttige feedback.

CONCLUSIE

Magento is een zeer interessante nieuwkomer in de wereld van e-commerce. Het bedrijf heeft wel naar mijn mening het software pakket veel te snel als een eerste officiële versie gelanceerd. Zonder enig commentaar in de code en zonder goede documentatie in het algemeen werd een reusachtig opensource pakket losgelaten in de wereld. De combinatie van het gekozen 'singleton design pattern', het gebruik van het Zend Framework en zijn 'magic methods', de enorme complexiteit van het pakket en de zeer verouderde informatie op de website en het dus ontbreken van goede documentatie maakte het pakket ten tijde van de release van Magento 1.0 simpelweg onhandelbaar. In de drie maanden die daarop volgen heeft het bedrijf achter Magento gelukkig niet stil gezeten. Er is ondertussen steeds meer documentatie beschikbaar en het bedrijf heeft bij cruciale problemen zelf verschillende modules gemaakt als voorbeeld voor de community. De grote problemen beginnen daarmee te slinken en het zal naar mijn mening nog hooguit enkele maanden duren voordat de beschikbare documentatie volwassen genoeg is geworden zodat de programmeur werkelijk het pakket gemakkelijk kan aanpassen naar eigen wensen. De functionaliteiten die het pakket biedt zijn duidelijk veel geavanceerder dan de functionaliteiten die alternatieve pakketten bieden, en dus kan Magento deze opstartprobleempjes makkelijk hebben en zal het naar mijn verwachting alleen nog maar aandeel gaan winnen op de e-commerce software markt.

De keuze voor Magento is daarom een juiste geweest en dit project was een prima mogelijkheid voor MARS media om kennis op te doen van het pakket. De keuze om Magento nu al te gebruiken bleek achteraf echter een tijdsintensieve optie. Dit is daardoor ten koste te gaan van verschillende requirements zoals het integreren van twee webshops in één backend. Magento ondersteund echter wel al alle benodigde functionaliteiten hiervoor, en ook de backend is al opgezet voor de tweede webshop. Echter het aanmaken van een tweede webshop inclusief design viel niet meer te doen in de tijd van dit project.

De requirements die wel behandeld konden worden zijn echter wel succesvol afgerond, en dus staat er nu een webshop die binnen enkele weken live zal gaan. Al met al ben ik zelf daardoor toch zeer tevreden, met als leukste punt voor mijzelf dat ik twee interessante modules heb ontwikkeld die met de huidige groei van Magento nog veel hergebruikt zullen gaan worden in de wereld en daarmee dus succesvol heb bijgedragen aan de ontwikkeling van een opensource pakket.

BIBLIOGRAFIE

1. *A conceptual investigation of the e-commerce industry.* **Veda C. Storey, Detmar W. Straub, Kathy A. Stewart, Richard J. Welke.** New York : ACM, 2000, Communications of the ACM, Volume 43, Issue 7, pp. 119 - 123.
2. **Gutmans, Andi.** Zend Framework 1.0 Released. [Online] [Cited: Mei 8, 2008.] <http://andigutmans.blogspot.com/2007/07/zend-framework-10-released.html>.
3. **E. Gamma, R Helm, R Johnson, J Vlissides.** *Design Patterns, Elements of Reusable Object-Oriented Software.* Harlow, England : Addison-Wesley, 1995. ISBN 5-272-00355-1.
4. **T Berners-Lee, RT Fielding, HF Nielsen.** Hypertext Transfer Protocol. *Graphics and media lab MSU.* [Online] [Cited: mei 26, 2008.] http://graphics.cs.msu.ru/courses/wp_el00/Internet/HTTP/article.html.
5. *RFC1321: The MD5 Message-Digest Algorithm.* **R.Rivest.** s.l. : RFC Editor, 1992.

