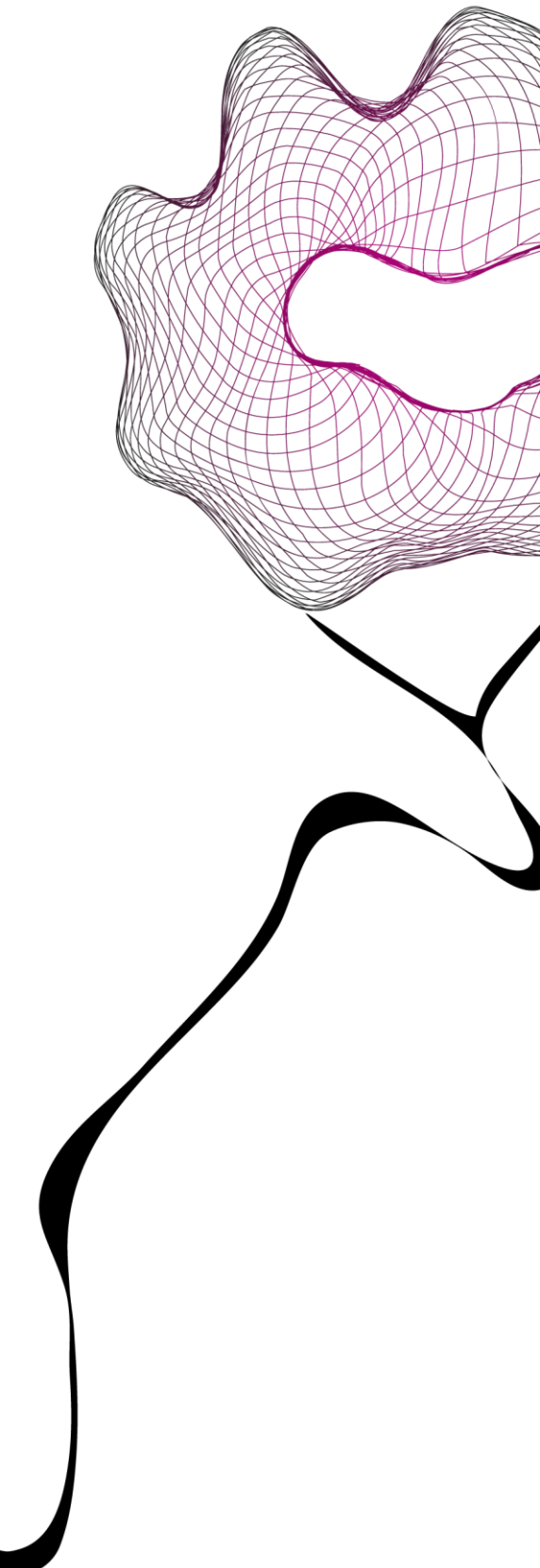




Ontwikkeling van een App store voor een iPaaS

Benno M. Lippok
Studentnummer 1018191

TECHNISCHE BEDRIJFSKUNDE
Behavioural, Management and Social sciences

Eerste begeleider
DR. MARIA-EUGENIA IACOB
Universiteit Twente.

Tweede begeleider
MSC LUCAS MEERTENS
Universiteit Twente.

Begeleider CAPE Groep
ING. SEBASTIAN PIEST

In opdracht van:
Cape Systems Integration B.V.



SAMENVATTING.

In dit verslag wordt onderzocht volgens welke stappen eMagiz B.V. een App store voor haar iPaaS kan ontwikkelen en met welke aspecten rekening gehouden dient te worden. iPaaS staat voor “integration Platform as a Service” en is volgens Gartner¹ een Cloud dienst die het ontwikkelen, uitvoeren en beheren van *integration* flows mogelijk maakt, die een verbinding kan zijn tussen zowel vaste als op Cloud gebaseerde hardware processen, diensten, applicaties en data, zowel binnen een bedrijf als tussen meerdere bedrijven samen. De hoofdvraag van het onderzoek luidt dan ook:

Hoe kan eMagiz een App store (communicatie platform) creëren die bijdraagt aan het hergebruik en beter vinden van de al voorradige applicaties en rekening houdt met het kennisniveau van de verschillende doelgroepen zodat een bloeiende community kan ontstaan?

Aan de hand van verschillende deelvragen zal een antwoord op de hoofdvraag gegeven worden. Om te beginnen worden door gebruik van relevante literatuur en marktonderzoek, de verschillende doelgroepen van eMagiz in kaart gebracht met daarnaast hun bijbehorende kennisniveau. Vervolgens wordt de content in verschillende categorieën verdeeld, zodat deze goed te vinden is en daarmee beter her te gebruiken. Tot slot worden de verschillende functionaliteiten van de App store in kaart gebracht en wordt er gekeken volgens welke designstappen deze geïmplementeerd moeten worden.

eMagiz dient binnen haar App store onderscheid te maken tussen de ontwikkelaars van de content en de daadwerkelijke gebruikers hiervan. Deze onderverdeling zorgt voor een splitsing tussen degene die content aan de App store toevoegen en degene die deze gebruiken om nieuwe applicaties te bouwen. Doordat het technische kennisniveau van de verschillende doelgroepen nauwelijks van elkaar verschilt, is het nog niet nodig om verder onderscheid binnen deze doelgroepen aan te brengen. De belangrijkste voorbeelden uit de markt waaraan eMagiz zich kan spiegelen zijn de App stores van MuleSoft, snapLogic en Mendix.

Er zijn twee type categorieën waarin content in onderverdeeld kan worden, dit zijn de software- en de designcategorie. De softwarecategorie beschrijft het soort toepassing waarvoor de content te gebruiken is en valt te zien als een soort thema. Voorbeelden hiervan zijn onder andere een mobile app integration en social apps. De designcategorie geeft aan wat voor

soort component de content binnen een applicatie inneemt. Zo kan bijvoorbeeld een bijna kant-en-klare template flow worden gedownload of slechts een klein onderdeel waaruit een template flow kan bestaan.

Omdat er binnen eMagiz nog onduidelijkheid bestaat over de definitieve onderverdeling van de designcategorie, is men op dit moment al begonnen aan vervolgonderzoek naar dit punt. Natuurlijk kunnen deze categorieën ook in de toekomst worden aangepast, als hier door bijvoorbeeld marktontwikkeling behoefte aan is.

De belangrijkste functie van de App store is het beschikbaar maken van content. Dit gebeurt door het downloaden van deze content. Om dit mogelijk te maken moet de App store ook de mogelijkheid bieden om steeds nieuwe content naar de App store te uploaden. Om er voor te zorgen dat de gebruiker ook weet welke content voor hem interessant is, is er extra informatie aan de content toegevoegd. Deze bestaat naast de verschillende categorieën en algemene informatie, zoals versiebeheer, ook uit de feedback vanuit de community. Zo kunnen gebruikers een indicatie geven van de bruikbaarheid van de content door middel van een rating of een comment.

VOORWOORD.

Voor u ligt mijn bachelorverslag dat ik in het kader van mij bacheloropleiding Technische Bedrijfskunde aan de Universiteit Twente uitgevoerd heb. In opdracht van CAPE Groep heb ik onderzoek gedaan naar de vraag op welke manier hun dochterbedrijf eMagiz het beste een App store kan gaan ontwikkelen.

Omdat de studie Technische bedrijfskunde een redelijk brede studie is, is er binnen het curriculum niet zoveel aandacht besteed aan IT gerelateerde aspecten. Omdat naar mijn mening de ontwikkeling in de IT de toekomst heeft, leek het mij interessant om mij binnen het bacheloronderzoek hierop te gaan richten. Ik ben CAPE Groep daarom ook zeer dankbaar voor de mogelijkheid om bij dit grensverleggende bedrijf een kijkje in de keuken te hebben mogen nemen.

Daarnaast wil ik ook graag mijn begeleiders bedanken. Vanuit de Universiteit Twente wil ik graag dr. Maria Iacob bedanken voor het vertrouwen dat ze in mij had tijdens de uitvoering van dit onderzoek en het aanreiken van de nuttige literatuur die ik bij mijn onderzoek kon gebruiken alsmede de licentie voor ArchiMate. Ik zou ook graag mijn tweede begeleider msc. Lucas Meertens willen bedanken voor zijn nuttige feedback en aanbevelingen. Binnen CAPE Groep wil ik vooral mijn dagelijkse begeleider Sebastian Piest bedanken die vaak met aanvullende informatie en literatuur kwam, die mij wat meer verdieping in de stof boden en waarmee ik bijzonder prettig kon brainstormen over mogelijke oplossingen en toepassingen.

Binnen eMagiz zou ik graag mijn dank uitspreken naar Alexander, Duc en Bas voor de dagelijkse discussies over de muziek, waarbij vooral de kerstperiode uiterst vermakelijk was, en natuurlijk voor de antwoorden op de vele (soms simpele) vragen die ik kon stellen. Daarnaast wil ik Duc in het bijzonder bedanken voor de hulp die hij mij geboden heeft bij de ontwikkeling binnen Mendix.

Ten slotte zou ik graag nog mijn lieve vriendin Sanne willen bedanken voor het meelezen en het corrigeren, zodat de door mij op papier gezette zinnen ook daadwerkelijk voor andere mensen te begrijpen waren.

Veel plezier tijdens het lezen,

Benno Marc Lippok
6 feb. 2015 te Enschede

INHOUDSOPGAVE.

1	Introductie.....	8
1.1	Inleiding.....	8
1.2	Aanleiding van het onderzoek.....	8
1.3	Onderzoeksdoel.....	9
1.4	Onderzoeksvragen.....	10
1.5	Onderzoeksaanpak.....	11
1.6	Opzet van het verslag.....	13
2	Literatuur overzicht.....	15
2.1	Theoretische Achtergrond.....	15
2.1.1	(Virtuele) Community.....	15
2.1.2	Applicatiesoftware.....	15
2.1.3	App store (Application Store).....	15
2.1.4	Citizen Developer.....	16
2.1.5	Software re-use.....	16
2.1.6	Flow(chart).....	16
2.1.7	Modelleertaal.....	16
2.2	Technische achtergrond.....	17
2.2.1	Model-driven development.....	17
2.2.2	Cloud computing en “the Cloud”.....	17
2.2.3	De Cloud en zijn gebruikers.....	21
2.2.4	Spiral methodology.....	22
2.3	Onderzoeksorganisatie.....	23
2.3.1	ArchiMate.....	23
2.3.2	Mendix.....	25
2.3.3	CAPE Groep.....	25
2.3.4	eMagiz.....	26
2.4	Bestaande App stores.....	26
2.4.1	App stores in het algemeen.....	26
2.4.2	Magic Quadrant for Enterprise iPaaS.....	26
2.4.3	Concurrenten & Partners eMagiz.....	29
3	Ontwerp en Ontwikkeling.....	31
3.1	De Algemene Situatie.....	31
3.1.1	Doelgroepen.....	31
3.1.2	Kennisniveau.....	32
3.1.3	Categorie.....	33
3.1.4	Beheer.....	35

3.2	Functionaliteiten en Eisen	35
3.2.1	User stories	35
3.2.2	Mock-up.	35
3.2.3	Key features.	36
3.3	Reference Architecture.....	37
3.3.1	Business Layer.....	38
3.3.2	Application Layer.....	39
3.3.3	Technology Layer.....	41
3.3.4	Algemene toepassing Reference Architecture.	41
4	Demonstratie.....	43
4.1	General Overview.....	43
4.2	Upload Content.....	43
4.3	Download Content.....	45
5	Validatie.....	48
5.1	Waarborg tijdens onderzoek.....	48
5.2	Functionaliteit	49
5.3	Illustratie	50
6	Conclusie en Aanbevelingen.....	52
6.1	Conclusie.....	52
6.2	Samenvatting & Onderzoeksvragen.....	53
6.3	Discussie van Beperkingen.	54
6.3.1	Wat is niet gelukt.....	54
6.3.2	Onderzoekstechnisch.....	55
6.3.3	Implementatie technisch.....	55
6.3.4	Limitations /Wat moet er nog gebeuren.....	55
6.3.5	Nice to have.	55
6.4	Aanbevelingen.....	55
APPendix.		57
APPendix A - User Stories.....		57
APPendix B – Uitgetekende User Stories.....		59
APPendix C – Uploaden.....		60
APPendix D – ArchiMate notations.....		60
APPendix F - Inspiratie van andere app stores in het algemeen en uit de markt.		63
APPendix G – Mendix impressie		64
Domain model		64
Microflows		65
Gebruikte Literatuur.		67

LIJST MET AFKORTINGEN.

BRS	Bussiness Requirement Specifications
DS	Design Science
DSRM	Design Science Research Methodology
IaaS	Infrastructure as a Service
IS	Information Systems
LOC	Lines of Code
MDD	Model-Driven Development
PaaS	Platform as a Service
aPaaS	application Platform as a Service
dbPaaS	database Platform as a Service
iPaaS	integration Platform as a Service
QoS	Quality of Service
SaaS	Software as a Service
SCM	Source Code Management
SDDC	Software Defined Datacenter
SLA	Service Level Agreement
SRS	System Requirement specifications

1 INTRODUCTIE.

Dit hoofdstuk geeft een overzicht van de bacheloropdracht en beschrijft het onderzoek. Na een korte inleiding wordt er begonnen met de aanleiding van het onderzoek. Vervolgens zal er worden ingegaan op het onderzoeksdoel. Daarop volgen de onderzoeksvragen die tot een oplossing van het onderzoeksdoel moeten dienen. In paragraaf 1.5 zal de onderzoeksaanpak toegelicht worden die bij dit onderzoek gehanteerd is. Ten slotte wordt er een overzicht gegeven van de verdere opzet van dit verslag.

1.1 INLEIDING.

eMagiz, gevestigd te Enschede, is een dochteronderneming van CAPE Systems Integration en biedt een gelijknamig integratiesoftwarepakket aan. Door het gebruik van Model-driven development (MDD), waarbij verschillende '*visuele blokken*' met elkaar verbonden kunnen worden, is men in staat om zonder '*technische code*' een werkende applicatie te maken.

1.2 AANLEIDING VAN HET ONDERZOEK.

In de afgelopen jaren heeft eMagiz verschillende soorten content ontworpen en daar komt elke maand nieuwe content bij. Het huidige systeem mist echter overzicht, waardoor het vooral voor nieuwe gebruikers lastig is om ermee om te gaan. Daar komt bij dat het niet echt duidelijk is wat voor content er allemaal beschikbaar is. Het kan dus goed zijn dat er al een heel goede oplossing bestaat, maar dat men dat niet weet en deze dus niet gebruikt. Door een gebrek aan overzicht is het vinden van de juiste content soms lastig, ook al weet de gebruiker precies wat hij nodig heeft.

Content is vaak specifiek voor een partner ontworpen, maar dat wil nog niet zeggen dat deze niet door anderen (deels) hergebruikt kunnen worden. Een bepaalde toepassing kan voor een andere eindgebruiker ook van nut zijn. eMagiz denkt dat er vooral veel voordeel met de al bestaande content behaald kan worden. Door een nieuwe combinatie van bestaande content kan, volgens het MDD concept, snel een nieuwe applicatie gemaakt worden. Het hergebruik van deze losse componenten gebeurt echter nog nauwelijks.

Buiten CAPE Groep zijn er ook andere bedrijven die behoefte hebben aan specifieke content van eMagiz. Volgens een IBM onderzoek is namelijk 90 procent van alle data pas in de afgelopen twee jaren verzameld. Volgens Hays (2003) zal het computergebruik in de toekomst alleen nog maar groeien. Hierdoor zal het voor veel bedrijven aantrekkelijk zijn om een koppeling te maken tussen hun eigenlijke bedrijfsvoering en de ICT. Dit bedrijf staat dan voor de keuze om een schaarse, hoogopgeleide ICT'er in dienst te nemen of te kiezen voor een ander softwarepakket, waardoor ook mensen zonder enige programmeerkennis koppelingen kunnen maken tussen software en hardware. Voor eMagiz is het op dit moment echter nog niet duidelijk welk kennisniveau deze verschillende doelgroepen hebben.

Doordat steeds meer bedrijven het nut van eMagiz ontdekken, is eMagiz de afgelopen jaren flink gegroeid. Om deze groei te kunnen blijven voortzetten en de verschillende partners trekkend te kunnen ondersteunen, is eMagiz bezig met een vertaalslag van, het puur ondersteunen van CAPE Groep, naar een zelfstandig opererende onderneming met een indirect marktmodel bestaande uit wederverkopers en implementatie-partners die eindklanten bedienen.

Hiervoor is het belangrijk dat deze bedrijven zo veel mogelijk zelfstandig eMagiz kunnen gebruiken. Al de nodige ondersteuning zou online vindbaar moeten zijn, zodat het mogelijk is dat een nieuwe gebruiker zelfstandig leert omgaan met het product. Simpelweg omdat het voor andere bedrijven niet mogelijk is om telkens bij eMagiz langs te gaan als er een kleine vraag is. eMagiz wil daarom onderzoeken op welke manier ze hun support schaalbaar kunnen maken.

De sleutel naar succes ziet eMagiz in een Community, zodat al de vragen online kunnen worden opgelost en er tevens een uitwisseling van ideeën, applicaties en best practices ontstaat.

1.3 ONDERZOEKSDOEL.

Een eerste stap richting de gewenste situatie ligt in het ontwikkelen van een App store. Deze App store moet het voor partners en hun klanten makkelijker maken om toegang te krijgen tot bepaalde content. Hierdoor zou er minder tijd nodig zijn voor het zoeken en maken van applicaties.

Een werkende App store zal de waarde van eMagiz vergroten. Bovendien kunnen diensten hierdoor aan een bredere doelgroep aangeboden worden. Dit moet ertoe leiden dat eMagiz blijft groeien. Om dit te realiseren wordt het belangrijk geacht om een actieve community te hebben. Als mensen dan vragen hebben kunnen ze hier snel kwalitatieve antwoorden op krijgen. De App store moet een stap naar de community zijn, dit uiteindelijke doel moet dus altijd in het achterhoofd gehouden worden.

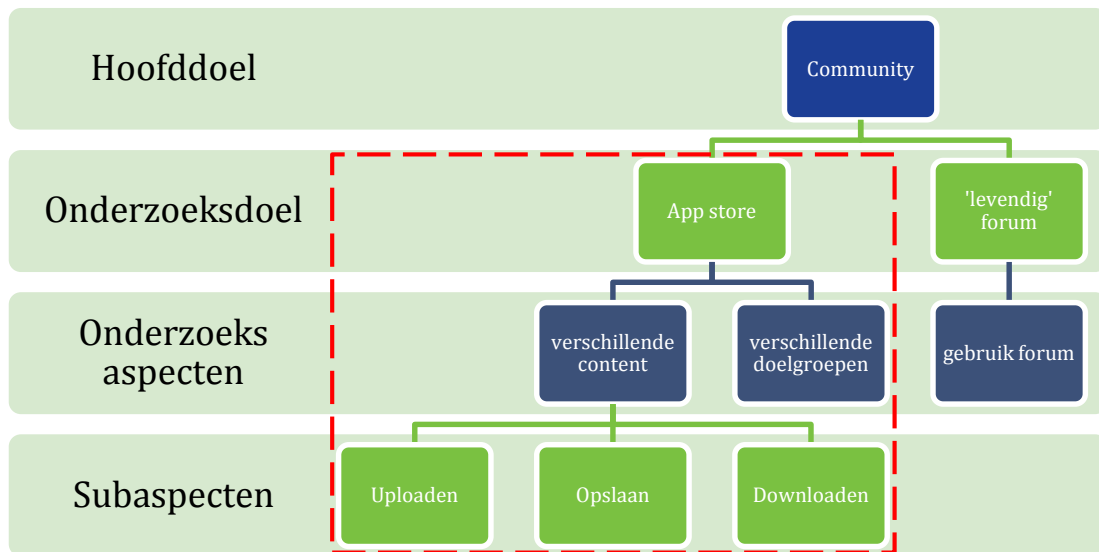
Op de korte termijn wil eMagiz er aan de hand van dit onderzoek achter komen welke stappen ondernomen moeten worden, zodat meer mensen van hun diensten gebruik willen en kunnen maken. Dit geldt in eerste instantie voor de huidige partners, maar moet ook het gebruik van eMagiz voor nieuwe partners makkelijker en aantrekkelijker maken. Hiervoor moet een onderverdeling gemaakt worden in de bestaande applicaties en in de verschillende doelgroepen met bijbehorende kennisniveaus.

Op dit moment levert eMagiz zijn diensten alleen aan partners. In de toekomst moet het echter mogelijk worden om ook de eindgebruikers van eMagiz-partners direct gebruik van deze diensten te laten maken. Dit moet gezien worden als extra service voor de partners, zodat de eindgebruiker niet voor elke kleine toepassing contact moet opnemen met zijn partner, maar zelfstandig content kan downloaden.

Tot slot wil eMagiz graag het gebruik van de community meer leven inblazen. Ook dit moet als een extra service gezien worden. Als iemand een vraag heeft, kan hij deze aan de community stellen en zo snel een antwoord op zijn vraag krijgen. Omdat het noodzakelijk is dat een gebruiker snel antwoord krijgt, moeten mensen betrokken raken bij de community zodat ze vaak naar het 'forum' kijken en waar nodig antwoorden kunnen geven. Als dit goed loopt, is de community een goed middel voor eMagiz om zich te profileren en zo nog meer klanten te kunnen acquireren door een sterk en vooral compleet product op de markt te zetten.

De App store is een stap naar een betere community. Om dit te realiseren zijn er vier aspecten die onderzocht moeten worden. Ten eerste moet er een analyse komen van de al bestaande applicaties en in welke mate deze het beste hergebruikt kunnen worden. Vervolgens moet gekeken worden wat de doelgroepen van eMagiz zijn. Van deze doelgroepen moet bepaald worden wat de behoeftes zijn, gekoppeld aan een bepaald kennisniveau. Tot slot moet er een koppeling komen tussen de bestaande content en de behoeften van de verschillende doelgroepen.

Om een beter beeld van de situatie te krijgen, is een gedetailleerd overzicht van de hele situatie opgesteld:



Figuur 1-1: De scope van het onderzoek binnen het probleemgebied.

Het onderzoek richt zich op het ontwikkelen van de App store en dus op het onderzoeken van de verschillende content en doelgroepen. Voordat de koppeling tussen de content en de doelgroepen gemaakt kan worden, is het van belang om te onderzoeken hoe de verschillende soorten content in de App store geplaatst kunnen worden en op welke manier deze er weer uit gehaald kunnen worden. Dit is in figuur 1 met een rood kader aangegeven. Het hele onderzoek moet in het kader van het ontwikkelen van een community gezien worden.

1.4 ONDERZOEKSVRAGEN.

Om onderzoek naar de problemen van eMagiz te kunnen uitvoeren zullen de onderzoeksgebieden, zoals weergegeven in Figuur 1-1, eerst verder moeten worden geconcretiseerd. Er is gekozen om per onderzoeksgebied deelvragen op te stellen, omdat er zo systematisch naar een antwoord gezocht kan worden voor de vraag hoe een App store voor eMagiz eruit moet gaan zien en waar deze aan moet voldoen. De antwoorden op de onderste blokken zullen steeds moeten bijdragen aan een antwoord op de daarboven gelegen blokken. Om een App store te ontwikkelen, hebben we ten eerste de bestaande applicaties en de doelgroepen waar deze applicaties voor gemaakt zijn nodig. Daarna moet er een koppeling ontstaan tussen deze applicaties en klanten. De zo ontstane App store moet vervolgens geïntegreerd worden in de community. Om deze stappen te kunnen nemen zijn per 'blok' de volgende deelvragen opgesteld:

1. Verschillende Klanten / doelgroepen

Welke onderverdeling kan er gemaakt worden tussen de huidige en potentiële klanten van eMagiz, gelet op het gebruik en de verschillende kennisniveaus binnen deze doelgroepen.

2. Verschillende Applicaties

Welke onderverdeling kan er tussen componenten gemaakt worden zodat deze goed her te gebruiken zijn en hoe kan de content beter vindbaar gemaakt worden?

3. De App store

Welke functionaliteiten moet de app store hebben en welke designstappen moeten hiervoor doorlopen worden zodat deze goed geïntegreerd kan worden in het huidige gebruik van de community?

De antwoorden op de deelvragen leiden samen tot een antwoord op de hoofdvraag:

Hoe kan eMagiz een App store (communicatie platform) creëren die bijdraagt aan het hergebruik en beter vinden van de al voorradige applicaties en die rekening houdt met het kennisniveau van de verschillende doelgroepen zodat een bloeiende community kan ontstaan?

1.5 ONDERZOEKSAANPAK.

Dit onderzoek wordt uitgevoerd aan de hand van het Design Science Research Model (DSRM) (Peffer, Tuunanen, Rothenberger, & Chatterjee, 2007). Ten eerste is voor dit model gekozen omdat het een zeer gangbaar model is binnen de Design science (DS), het onderzoek naar informatie systemen (IS). Daarbij past dit model beter bij dit onderzoek dan een bedrijfskundige onderzoeks aanpak, zoals bijvoorbeeld de algemene bedrijfskundige probleemaanpak (ABP) (Heerkens & van Winden, 2012). Deze richt zich voornamelijk op het opsporen en oplossen van problemen binnen een bedrijfskundige context. In dit onderzoek moet er echter een geheel nieuw proces gestart worden en daar past de aanpak van Peffer beter bij.

Peffer heeft, aan de hand van de 6 punten van Archer (1984), een stappenplan ontwikkeld, dat bij het design van een nieuw model doorlopen moet worden.

1. Probleem identificatie en motivatie

Het definiëren en specificeren van het onderzoeksprobleem, zodat de waarde van de oplossing gewaarborgd kan worden. Omdat de probleemstelling gebruikt zal worden voor het ontwikkelen van een artefact, dat een effectieve oplossing kan bieden, is het aan te raden om het probleem zover op te splitsen zodat de oplossing de complexiteit van het hele probleem omvat.

Het waarborgen van de oplossing dient twee dingen. Ten eerste motiveert het de onderzoeker en de doelgroep van het onderzoek om de oplossing na te streven en de resultaten te accepteren. Daarnaast helpt het om redeneringen en verbanden, die de onderzoeker legt, te begrijpen.

Adequate kennis, van zowel de problemen als de omgeving waarin deze zich bevinden, zijn een voorwaarde voor deze stap.

2. Het definiëren van de doelstellingen van de oplossing

Het afleiden van de doelstelling van de oplossing vanuit de probleemstelling en de kennis over wat mogelijk en haalbaar is, is het doel van dit tweede punt. Deze doelstellingen kunnen kwalitatief zijn, zoals een omschrijving van de gewenste oplossing en hoe verwacht wordt dat een nieuw artefact bijdraagt aan de, tot dan toe nog niet bekende, problemen.

De Doelstellingen moeten rationeel worden afgeleid vanuit de probleemspecificatie. Hiervoor is onder andere kennis nodig van de toestand van de problemen en de huidige oplossingen, en, indien aanwezig, de huidige oplossingen en de effectiviteit hiervan.

3. Ontwerp en ontwikkeling

Het derde punt heeft als doel om een artefact te ontwikkelen. Dit kunnen prototypes, modellen, methodes of andere concretisering zijn (ruim gedefinieerd in (Hevner, March, & Park, 2004)) of "Nieuwe eigenschappen van technische, sociale en/of informatieve hulpbronnen" (Järvinen, 2007, p. 49).

Theoretisch gezien kunnen bij een ontwerponderzoek artefacten elke vorm van ontwerp zijn die iets bijdragen aan het onderzoek. Dit beslaat dus ook het bepalen van de gewenste functies van dit artefact.

Om de stap van ontwerp naar ontwikkeling te zetten, is het belangrijk dat men op de hoogte is van de geldende theorieën en dat men deze in de praktijk kan brengen.

4. Demonstratie

Door het artefact te demonstreren en te testen, kan men achter mogelijke problemen komen en deze trachten op te lossen. Het testen kan gedaan worden door middel van experimenten, simulaties of een andere passende activiteit aan de hand van bijvoorbeeld case studies en resultaten.

Voor deze stap is effectieve kennis over hoe men met dit artefact dient om te gaan vereist.

5. Evaluatie

De Evaluatie is het observeren en meten van de bijdrage van een artefact tot een oplossing van het probleem. In deze stap gaat men de doelstellingen vergelijken met de, uit de demonstratie naar voren gekomen, resultaten van het artefact. Het vereist de nodige kennis om met behulp van deze gegevens een relevante analyse te vormen. Afhankelijk van de aard van het probleem en de vorm van het artefact kan de vorm van de evaluatie verschillen. Zo zou het bijvoorbeeld kunnen bestaan uit vergelijkingen van de doelstellingen uit fase twee, objectieve kwantitatieve prestatie-indicatoren, zoals budgetten of productieaantallen, resultaten van tevredenheidsonderzoek, feedback van klanten of simulaties. Het zou kwantificeerbare systeemprestaties, zoals reactietijd of beschikbaarheid, kunnen omvatten. Theoretisch gezien kan een waardering ook empirisch of logisch bewijs bevatten.

Aan het einde van deze fase moet de onderzoeker bepalen of hij activiteit drie gaat herhalen om zo te proberen de effectiviteit van het artefact te verbeteren, of door te gaan naar de volgende fase en daarmee verdere verbeteringen achterwege laat en deze uitstelt tot latere projecten. Het soort onderzoek kan bepalend zijn voor de keuze om dit wel of niet te doen.

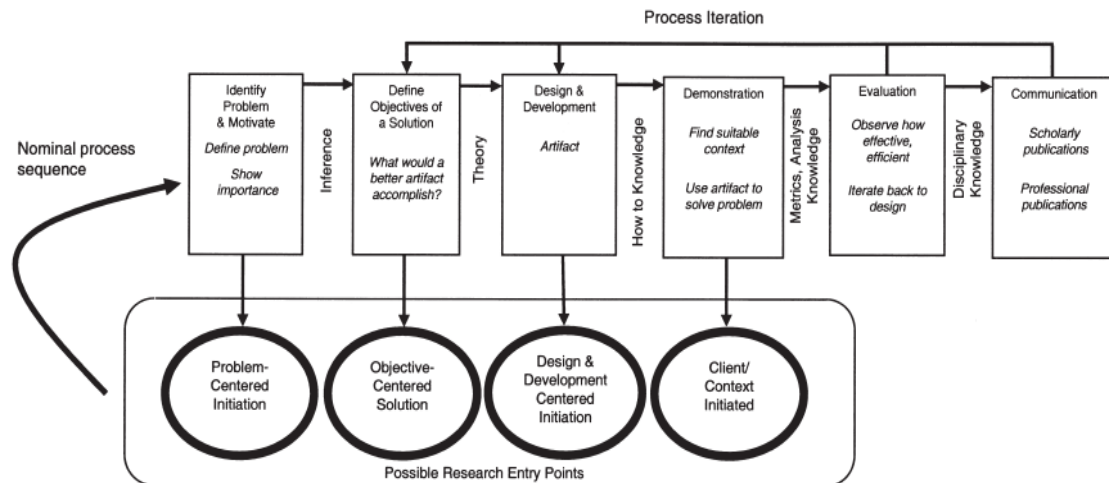
6. Communicatie

Communiqueer het probleem en het belang van het probleem, het nieuwe artefact, het nut hiervan en de vernieuwende factor, de stijfheid van het ontwerp en het nut ervan voor de onderzoekers van andere relevante doelgroepen wanneer nodig.

In wetenschappelijke onderzoekspublicaties kunnen onderzoekers de structuur van dit onderzoek gebruiken om hun eigen onderzoek te structureren zoals bij empirisch onderzoek al het geval is. Denk hierbij aan: probleemstelling, literatuurstudie, hypothese ontwikkeling, het verzamelen van gegevens, analyse, resultaten, discussie en conclusie.

Deze stap vereist kennis over de disciplinaire cultuur.

Een schematische weergave van de zes hierboven beschreven fasen zijn grafisch weergegeven in Figuur 1-2. Hierin zijn ook duidelijk de terugkoppelmomenten te zien die Peffers in zijn methode aangeeft.



Figuur 1-2: Design Science Research Model (DSRM) van Peffers et al., 2007.

1.6 OPZET VAN HET VERSLAG.

Om de onderzoeksaanpak samen te vatten volgt er hier een kort overzicht van de verdere structuur van het verslag in de vorm van een lijst met hoofdstukken. Deze zijn gebaseerd op de tweede case die Peffers in zijn artikel (2007, pp. 60 - 63) beschrijft en zijn aanpak voor empirisch onderzoek:

Hoofdstuk 2: De Literatuur

In dit hoofdstuk wordt er een overzicht gegeven van de behandelde literatuur en andere belangrijke aspecten die in het onderzoek naar voren komen.

Dit hoofdstuk zit een beetje tussen de eerste en tweede stap van Peffers in. Hierin wordt informatie vergaard zodat de doelstelling van de oplossing gedefinieerd kan worden.

Hoofdstuk 3: Ontwerp en ontwikkeling

Hierin wordt begonnen met het ontwerp van een prototype en zal er gekeken worden naar de functionaliteiten die deze zal moeten hebben. Deze zullen moeten resulteren in de architectuur van het ontwerp.

Dit hoofdstuk bestaat uit de tweede en derde stap van het model van Peffers en richt zich vooral op de ontwerpstappen en minder op de daadwerkelijke ontwikkeling.

Hoofdstuk 4: Demonstratie,

Het doel van dit hoofdstuk is het testen van het prototype bij zoveel mogelijk verschillende doelgroepen om zo voldoende feedback te krijgen die vervolgens verwerkt kan worden.

Volgens Peffers zal in dit hoofdstuk het prototype getest en geëvalueerd moeten worden. Omdat dit onderzoek zich echter meer richt op de ontwikkelingsstappen in plaats van de daadwerkelijke ontwikkeling zal in dit hoofdstuk vooral een demonstratie beschreven worden, met daarin de verschillende functionaliteiten.

Het testen en evalueren zal uiteraard wel plaatsvinden maar in de Appendix terug te vinden zijn.

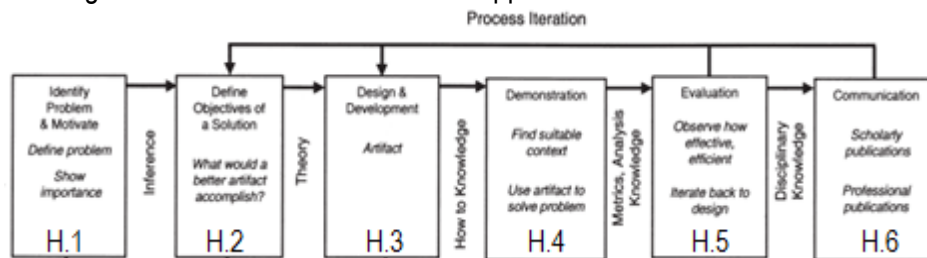
Hoofdstuk 5: Validatie

In dit hoofdstuk zal er terug gekeken worden op de eisen die aan het begin van het onderzoek gesteld zijn. Daarnaast zal het prototype geëvalueerd worden.

Net zoals Peffers stelt, zal er een evaluatie plaats vinden waarin onder andere gekeken zal worden naar de vragen voor wie het prototype uiteindelijk van nut zal zijn, wat de toegevoegde waarde is en op welke manier het prototype in het bedrijf gebruikt zal worden.

Hoofdstuk 6: Conclusie en aanbevelingen

In dit hoofdstuk zal, zoals Peffers suggereert voor wetenschappelijke onderzoekspublicaties, een antwoord op de hoofdvraag gegeven worden. Verder zullen de beperkingen worden besproken en volgen er aanbevelingen over hoe eMagiz verder moet gaan met het ontwikkelen van de App store.



Figuur 1-3: Schematische weergave van de hoofdstukindeling gekoppeld aan Peffers et al., 2007.

2 LITERATUUR OVERZICHT.

Nadat in het eerste hoofdstuk een introductie over dit onderzoek gegeven is, zal er in dit hoofdstuk dieper worden ingegaan op de literatuur die voor dit onderzoek van belang is. Er zal begonnen worden met een theoretische achtergrond, waarin wat algemenere begrippen met behulp van een definitie kort worden toegelicht, vormt de basis.. Daarna wordt onder het kopje technische achtergrond ingegaan op de theorie die in de literatuur gevonden is. Als derde worden de verschillende 'Spelers' besproken die in dit onderzoek aan bod komen. Als laatste wordt gekeken naar al bestaande App stores en concurrenten.

De literatuur moet samen met de onderzoeksopzet en de ervaringen op de werkvloer leiden tot een antwoord op de hoofdvraag.

Volgens Peffers is het hoofdstuk *literature review* alleen nodig bij empirisch onderzoek, in het algemeen dient deze stap als het vergaren van kennis over mogelijke en wenselijke oplossingen.

2.1 THEORETISCHE ACHTERGROND.

2.1.1 (Virtuele) Community.

Een virtuele gemeenschap, vaak simpelweg aangeduid als community, brengt individuen, groepen en bedrijven, die een gezamenlijk doel of gezamenlijke interesse hebben, via het internet tot elkaar². Vaak kunnen op een forum vragen worden gesteld die dan door anderen beantwoord kunnen worden. Daarnaast vindt er een uitwisseling van documenten en overige zaken plaats.

Somani (2012) definieert een virtuele community in het kort als: *"Een verzameling van individuen die via elektronische communicatie een verbond vormen"*.

2.1.2 Applicatiesoftware.

Volgens techopedia³ is er bij applicatiesoftware sprake van een programma of een groep programma's voor eindgebruikers. Deze programma's zijn onder te verdelen in twee klassen, namelijk de systeemsoftware en applicatiesoftware. Systeemsoftware bestaat uit 'low-level' programma's die communiceren met computers op een basisniveau. Applicatiesoftware bevindt zich kwa abstractieniveau in een hoger en daarmee verder van de hardware gelegen systeemsoftware. Deze omvat onder andere databaseprogramma's, tekstverwerkers en spreadsheets. Applicatiesoftware kan worden gegroepeerd samen met systeemsoftware of alleen worden gepubliceerd.

Applicatiesoftware voor mobiele applicaties⁴ wordt ook wel eens aangeduid als *een app*. Dit zijn over het algemeen kleine, individuele software-eenheden met een beperkte functie. In de meeste gevallen voorzien apps diensten, die ook via de pc te bereiken zijn.

Dat apps populair zijn blijkt wel uit het feit dat de American Dialect Society het woord 'app' in 2010 uitriep tot *'Word of the Year'*⁵.

2.1.3 App store (Application Store).

Een Application store, in het algemeen App store genoemd, is een online samengestelde marktplaats die aan ene de kant softwareontwikkelaars in staat stelt om hun product te gelde te maken en aan de andere kant de gebruiker steeds nieuwe functionaliteiten brengt (Jansen & Bloemdal, 2013).

Een succesvolle App store draagt vaak bij aan het succes van de ontwikkeling van een software ecosysteem wat op zichzelf weer gunstig is voor het bedrijf dat de software ontwikkelt (Elmer-DeWitt, 2010).

Volgens Gartner⁶ zal het gebruik van App stores alleen nog maar toenemen en was er tussen de jaren 2009 en 2013 sprake van een vertienvoudiging van het aantal downloads. Dat deze trend alleen maar zal doorzetten, blijkt ook uit een onderzoek van CISCO (2009) waaruit blijkt dat het dataverkeer in de toekomst alleen nog maar zal toenemen.

2.1.4 Citizen Developer.

De term Citizen Developer werd in 2009 door Gartner geïntroduceerd als *“Een eindgebruiker die nieuwe zakelijke toepassingen creëert voor consumptie door de gemiddelde anderen”*. Al snel werd deze term gebruikt voor vrijwel elke *front-line* medewerker die zelf zijn eigen applicaties bouwt en hiermee zijn manier van werken kan verbeteren. Hierdoor kan zijn afdeling, of bedrijf, snel en gemakkelijk informatie bijhouden, werk beheren, resultaten meten en samenwerken met medewerkers. Dit kan zowel gelden voor een mobiel apparaat als voor een desktop versie (Fenton, 2014).

Volgens een onderzoek van Gartner zou in 2014 tenminste 25 procent van alle applicaties ontwikkeld worden door Citizen Developers (Petty, Christy; Meulen, Rob van der, 2011).

2.1.5 Software re-use.

Het hergebruiken van software is het gebruik van bestaande software-componenten of kennis, om zo nieuwe software te maken. Dit hergebruik is een belangrijke manier om bestaande softwarekwaliteit significant te verbeteren en de productiviteit te verhogen (Frakes & Terry, 1996).

De herbruikbaarheid, de zogenoemde *reusability*, is de mate waarin een vorm van content kan worden hergebruikt. Om significante resultaten te behalen moet een re-use program systematisch zijn (Frakes & Isoda, Success factors, 1994).

Organisaties die systematisch gebruik willen maken van hergebruik moeten in staat zijn om hun vooruitgang te meten en zo te identificeren wat de meest effectieve hergebruikstrategie is. Rothenberger en Hershauer (1999) zeggen dat dit het beste gedaan kan worden door onder andere te kijken naar het percentage van hergebruikte coderegels (Lines of Code, LOC). Verder hangt het er vanaf in welke laag (diep, middel of oppervlakte) de code gebruikt wordt, om zo te kijken waarop de nieuwe code van toepassing is. Structuren moeten in zo klein mogelijke stukken worden onderverdeeld, zodat gekeken kan worden welke stukken wel en welke niet hergebruikt (kunnen) worden. Tot slot moet ervoor gezorgd worden dat er geen duplicaten voorkomen.

2.1.6 Flow(chart).

Een flowchart⁷, of in het Nederlands een stroomdiagram, is een type diagram dat met behulp van verschillende componenten een proces in beeld brengt, met daarin informatie over de te doorlopen stappen of elkaar opeenvolgende gebeurtenissen. Door deze componenten met elkaar te koppelen, treden er allemaal losse acties op die samen een heel proces beschrijven.

2.1.7 Modelleertaal.

Een modelleertaal is een grafische en/of tekstuele structuur waarmee van tevoren de belangrijkste functies, eigenschappen voorwaarden en relaties van een systeem kunnen worden gedefinieerd. Hierdoor moet het voor zowel de opdrachtgever als de uitvoerder duidelijk zijn waaraan het product moet voldoen. Nadat het model grafisch is vastgelegd, kan het worden ‘vertaald’ naar een computertaal.

Grafische modellen gebruiken lijnen om diagrammen en symbolen met elkaar te verbinden om zo de onderlinge relaties weer te geven. Bij tekstuele modelleertalen worden vaak alleen gestandaardiseerde woorden verzameld die dan omgezet kunnen worden in parameters (He, Ma, Shao, & Li, 2007).

2.2 TECHNISCHE ACHTERGROND.

2.2.1 Model-driven development.

Zoals in de inleiding (§1.1) al kort werd benoemd, wordt er bij Model Driven Development (MDD) gebruikt gemaakt van verschillende ‘visuele blokken’, die met elkaar verbonden kunnen worden. Hierdoor is men in staat om zonder ‘technische code’ een werkende applicatie te maken.

In tegenstelling tot de meeste programmeertalen richt MDD zich op de vraag, wat een systeem specifiek moet kunnen, in plaats van de vraag hoe dit moet gebeuren. Door deze verschuiving is het in de praktijk mogelijk om de complexiteit van de problemen, beschouwd volgens de huidige normen, drastisch te verminderen. Veel dingen zoals *object allocation*, *method lookup* en *exception handling*, die in het verleden allemaal nog handmatig moesten worden geprogrammeerd, kunnen nu achter de schermen automatisch gegenereerd worden. Hierdoor is MDD vooral goed toe te passen in complexe situaties die nu nog vaak handmatig geregeld moeten worden zoals systeem persistentie, interoperabiliteit en distributie (Atkinson & Kühne, 2003).

Doordat de focus meer op de modellen ligt in plaats van op het computerprogramma, worden concepten gebruikt die veel minder gebonden zijn aan de onderliggende implementatietechnologieën en deze liggen daarom veel dichterbij het probleemgebied dan de meeste andere talen. Hierdoor zijn de modellen vooral makkelijker te specificeren, begrijpen en te onderhouden. Niet de programmeurs, maar domain experts kunnen de producten hierdoor maken, omdat deze automatisch gegenereerd worden uit de corresponderende modellen. Als laatste maakt MDD modellen minder gevoelig voor de gekozen computertechnologie en de evolutionaire veranderingen die met deze technologie gepaard gaan (Selic, 2003).

2.2.2 Cloud computing en “the Cloud”.

Dat er onduidelijkheid bestaat over wat cloud computing is en wat je ermee kan doen blijkt wel uit de woorden van Oracle’s CEO Larry Ellison: *“The interesting thing about cloud computing is that we’ve redefined cloud computing to include everything that we already do.... I don’t understand what we would do differently in the light of cloud computing other than change the wording of some of our ads.”*

Deze onduidelijkheid komt volgens Pallis (2010) vooral voort uit het feit dat Cloud computing een steeds evoluerend paradigma is. Om toch een beeld te geven citeert Pallis de definitie die het *US National Institute of Standards and Technology’s* geeft, welke de belangrijkste punten van Cloud computing omvat:

“Cloud computing is een model dat de gebruiker in de gelegenheid stelt om via een netwerktoegang gebruik te maken van een verzameling van handige, on-demand werkende en configureerbare IT-middelen zoals netwerken, servers, opslag, applicaties en diensten welke snel kunnen worden verkregen en met minimale inspanning beheerd kunnen worden.

Dit Cloudmodel zorgt voor een brede beschikbaarheid en omvat:

- *Vijf essentiële eigenschappen;*
 - *On-demand self-service*
 - *Brede toegang tot het netwerk*
 - *Resource pooling*
 - *Flexibiliteit*
 - *Measured service*
- *Drie servicemodellen;*
 - *Cloud-software as a service (SaaS)*

- *Cloud-platform as a service en (PaaS)*
- *Cloud-infrastructure as a service (IaaS)*
- *Vier deployment modellen;*
 - *Private Cloud*
 - *Community Cloud*
 - *Public Cloud*
 - *Hybrid Cloud.*”

Sommige van de hierboven genoemde punten spreken voor zich, maar andere behoeven nog korte toelichting. Dit zal zoveel mogelijk gedaan worden met definities van vooraanstaande organisaties, zoals Gartner, maar ook door wat diepgaandere uitleg vanuit de literatuur.

2.2.2.1 **Resource pooling.**

Resource pooling treedt op als een Cloud-computing-provider diensten aan meerdere klanten, ook wel ‘*tenants*’ genoemd, levert. De provider kan, afhankelijk van de behoeften van zijn klanten, het service level aanpassen. De provider kan zo inspelen op de behoeftes die zijn klanten op een bepaald moment hebben, zonder zichtbare veranderingen toe te passen⁸.

Het idee hierachter is dat providers van Cloud computing over een bijna oneindige voorraad beschikken en deze middelen beschikbaar stellen, door deze op een meta-niveau te beheren. Hierdoor kunnen klanten hun service level aanpassen naar het, op dat moment, gewenste niveau zonder gebonden te zijn aan beperkingen van fysieke of virtuele middelen. Hierbij moet gedacht worden aan zaken als dataopslag, het verwerken van diensten en bandbreedte van geleverde diensten.

2.2.2.2 **Flexibiliteit.**

Zoals bij resource pooling al werd aangeduid, kan er door gebruik te maken van de Cloud snel toegang verkregen worden tot verschillende toepassingen. Waarbij er bij resource pooling vooral gekeken wordt vanuit de kant van de provider, wordt er bij de flexibiliteit gekeken naar het nut voor de gebruiker⁹.

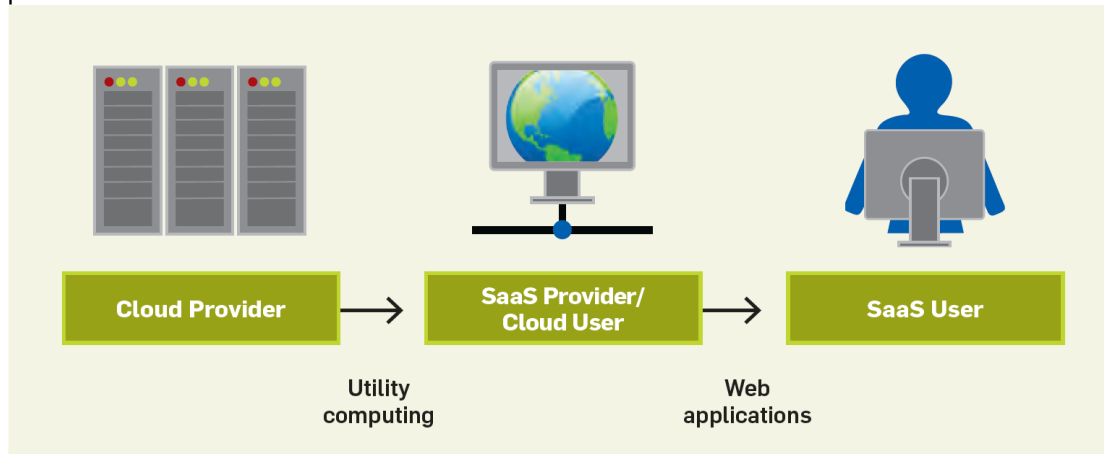
Het voordeel dat hiermee behaald kan worden, is dat het niet uitmaakt of er 1000 servers één uur bezig zijn of dat één server 1000 uur aan het werk is. Deze economische aantrekkingskracht wordt in de praktijk vaak omschreven als “*converting capital expenses to operating expenses (CapEx to OpEx)*”, maar volgens Armbrust (2009) past de term “*pay as you go*” beter bij deze economische situatie. De aangeschafte uren kunnen door de gebruiker op een zelf bepaald moment worden ingezet en hoeven niet gelijkmatig over een periode verdeeld te worden. Zo zal de gebruiker bijvoorbeeld bij het gebruik van 100 serveruren vandaag en geen serveruren morgen alleen betalen voor wat hij gebruikt op die dag, dit principe staat ook wel bekend als “*usage-based pricing*” (Armbrust M. , et al., 2009). Daarnaast hoeft er van tevoren geen kapitaal te worden uitgegeven en kan men dit vrijgekomen bedrag investeren in de eigen *kern business*. In een berekening laten Armbrust et al. zien dat het veel goedkoper is om alleen voor de software te betalen op de momenten dat je deze nodig hebt, in plaats van al de benodigde infrastructuur zelf aan te schaffen (Armbrust M. , et al., 2009, p. 10).

2.2.2.3 **Measured Service.**

Measured service is de term die volgens techopedia¹⁰ gebruikt wordt door een Cloud Provider om zo de omvang van zijn diensten te berekenen. Met deze gegevens kan de provider zijn diensten factureren, onderzoeken hoe hij zijn hulpbronnen efficiënt kan inzetten en algemene voorspellingen doen.

2.2.2.4 Cloud Software as a Service (SaaS)

Software as a Service¹¹ (SaaS) is een software-distributiemodel voor klanten die via het internet toegang tot hun software willen hebben. SaaS staat ook wel bekend als gehoste software of on-demand software. Bij een SaaS verleent de provider namelijk toegang tot software via zijn datacenter en heeft de klant toegang tot deze software via een standaard webbrowser. Deze koppeling is door Armbrust (2010) in Figuur 2-1 duidelijk weergegeven. Aan de linker kant staat het datacenter en aan de rechterkant de eindgebruiker, de koppeling hiertussen is dan de SaaS provider.



Figuur 2-1: Koppeling van de Cloud provider en de eindgebruiker door middel van SaaS, volgens (Armbrust M. et al., 2010).

Er zijn drie belangrijke kenmerken die bijna voor elke SaaS provider van toepassing zijn:

- Updates worden automatisch toegepast, zonder tussenkomst van de klant.
- De service wordt op basis van een abonnement/licentie aangeschaft.
- Het is niet nodig om bij de klant hardware te installeren.

2.2.2.5 Cloud-Platform as a Service (PaaS)

Platform as a Service¹² (PaaS) is een softwareconcept waarbij een geïntegreerde oplossing, solution stack of dienst via een internetverbinding wordt verhuurd of geleverd.

Een solution stack kan bestaan uit een set van componenten of software subsystemen die gebruikt worden om een volledig functioneel product of dienst te ontwikkelen, zoals een webapplicatie die gebruik maakt van een besturingssysteem (Operating System, OS), webserver, database en programmeertaal. In het algemeen kan een solution stack een OS, middleware, database of applicatie leveren

2.2.2.5.1 iPaaS.

eMagiz is zelf een integration Platform as a Service (iPaaS), wat Gartner¹³ omschrijft als een Cloud dienst die het ontwikkelen, uitvoeren en beheren van *integration* flows mogelijk maakt, die een verbinding kunnen zijn tussen zowel vaste als op Cloud gebaseerde hardware processen, diensten, applicaties en data, zowel binnen een bedrijf als tussen meerdere bedrijven samen.

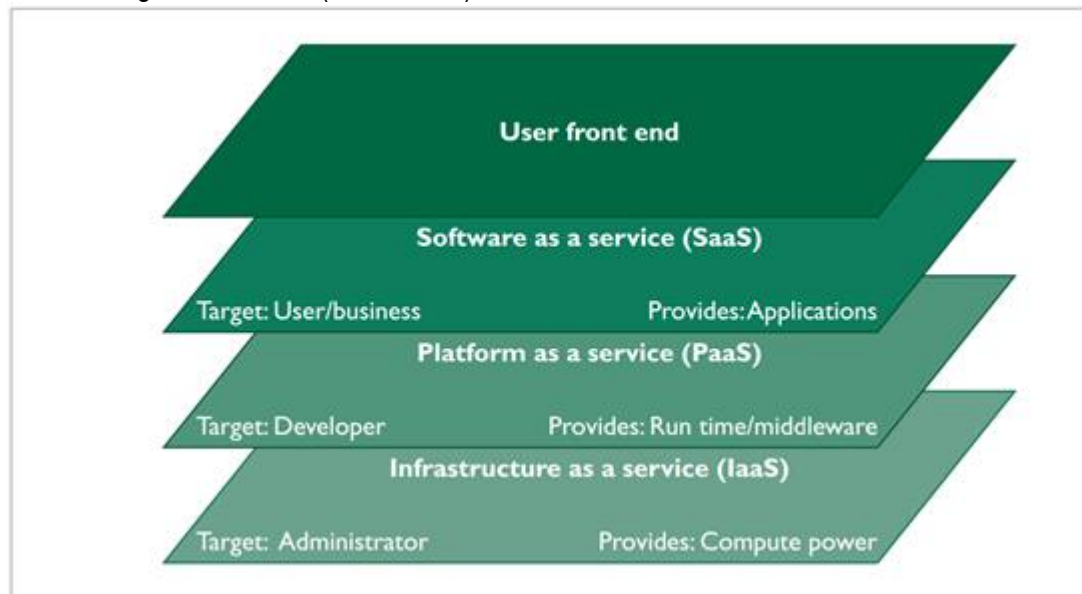
2.2.2.6 Cloud-Infrastructuur als a Service (IaaS).

Infrastructure as a Service (IaaS) is een softwaremodel dat op basis van outsourcing een computerinfrastructuur levert om zo bedrijven te ondersteunen. Dit doen de IaaS providers door hardware, opslag, servers en datacenter ruimte beschikbaar te stellen. Hierdoor staat IaaS ook wel bekend als Hardware as a Service (Haas).

2.2.2.7 Servicemodellen

Hoe deze drie servicemodellen (SaaS, PaaS, IaaS) zich tot elkaar verhouden is weergegeven in Figuur 2-2 (Pallis, 2010). De eindgebruiker heeft vanuit zijn webbrowser toegang tot de applicaties via een SaaS. Deze SaaS applicaties worden ondersteund door een PaaS. Hierop

kunnen ontwikkelaars nieuwe applicaties ontwikkelen en bestaande onderhouden. De onderste laag bestaat uit de IaaS, deze zorgen voor de koppeling tussen de software en de fysieke hardware. Uiteindelijk moet namelijk elke applicatie op fysieke hardware ergens in een datacenter gerund worden (Haan, 2013).



Figuur 2-2: Een algemene weergave van de Cloud infrastructuur (Pallis, 2010)

2.2.2.8 **Private Cloud Computing.**

Volgens Gartner¹⁴ is er bij Private Cloud computing sprake van een Cloud computing vorm waarbij slechts één organisatie gebruik maakt van een Cloud, of een speciaal voor deze organisatie geïsoleerde Cloud.

2.2.2.9 **Community Cloud.**

Volgens de definitie van Gartner¹⁵ verwijst een Community Cloud naar een gedeelde Cloud service omgeving die gericht is op een beperkt aantal organisaties of werknemers, zoals banken of afdelingshoofden die handel met elkaar drijven. De leden hebben vaak een gezamenlijk overkoepelend doel waardoor de doelgroepen doorgaans dezelfde beveiliging, privacy, prestaties en overeenstemmingseisen hebben. Community leden kunnen bovendien vaak zelf regelen wie er toegang heeft tot de Cloud, zonder dat tussenkomst van de Cloud provider noodzakelijk is.

2.2.2.10 **Public Cloud Computing.**

Volgens de definitie van Gartner¹⁶ is Public Cloud Computing een stijl die zich voornamelijk kenmerkt door de brede mogelijkheid aan de flexibele schaalbaarheid van IT-toepassingen die, in de vorm van een service via internettechnologieën, aan externe klanten wordt aangeboden. Een voorbeeld hiervan is het ondersteunen van klanten buiten de organisatie van de provider. Door het gebruik van publieke Cloud diensten worden economische schaalvoordelen behaald in zowel het delen van toegang tot verschillende technologieën, als in het verlagen van deze kosten.

Dit houdt in dat organisaties, in elke denkbare sector, samen gebruik kunnen maken van dezelfde diensten zoals infrastructuur, platformen of software, zonder garanties over de opslagplaats van de gegevens.

2.2.2.11 *Hybrid Cloud Computing.*

Volgens Gartner¹⁷ verwijst Hybrid Cloud computing naar een beleid dat gebaseerde en gecoördineerde dienstverlening gebruikt en beheer over een mix van interne en externe Cloud diensten mogelijk maakt.

Quarati (2015) legt de nadruk op de integratie van interne (private) en externe (public) hulpmiddelen waarbij een combinatie ontstaat van de schaalbaarheid van de public Cloud en de controle van de private Cloud.

Het is hierbij belangrijk dat er een mogelijkheid is om de beschermde en externe gebieden duidelijk te scheiden door middel van grenzen. Dit kan worden gerealiseerd door aan de ene kant veiligheidseisen af te dwingen en aan de andere kant prestatieniveaus te leveren die passen bij de behoefte van de klant. Om dit beheer goed te kunnen uitvoeren, is het vereist dat er van tevoren duidelijke *Quality of Service* (QoS) gedefinieerd zijn en de commerciële overeenkomst tussen de ICT provider en de klant in kaart is gebracht in een *Service Level Agreement* (SLA) (D'Agostino, et al., 2013).

2.2.3 De Cloud en zijn gebruikers

Johan de Haan (2013) heeft getracht om meer duidelijkheid te scheppen in de verschillen van het Cloudlandschap. Hij richt zich hier alleen op de opbouw van de Cloud en de bijbehorende doelgroepen, de hardware laat hij hierbij buiten beschouwing. Dit doet hij aan de hand van een matrix die bestaat uit zes rijen en drie kolommen, te zien in Figuur 2-3.

Layer 6	SaaS	Applications		
Layer 5	App Services	Compute App Services	Communicate App Services	Store App Services
Layer 4	Model-Driven PaaS	bpmPaaS, Model-Driven aPaaS	Model-Driven iPaaS	baPaaS
Layer 3	PaaS	aPaaS	iPaaS	dbPaaS
Layer 2	Foundational PaaS	Application containers	Routing, messaging	Object storage
Layer 1	Software Defined Datacenter	Virtual Machines	Software Defined Networking (SDN)	Software Defined Storage (SDS)
		Compute	Communicate	Store

Figuur 2-3: De Cloudlandschap-matrix van Johan de Haan.

Omdat een beschrijving van de hele matrix te uitgebreid zou worden, is er gekozen om het bij dit onderzoek te beperken tot de zes rijen, ook wel *layers* genoemd:

1. Software Defined Datacenter

VMWare introduceerde de term "Software Defined Datacenter" (SDDC) om aan te geven dat de belangrijkste hardware-elementen van een datacenter tegenwoordig kunnen worden gevirtualiseerd, denk hierbij aan servers, netwerken en opslag. Deze elementen zitten allemaal in een SDDC en worden daar geconfigureerd en beheerd door een centrale softwareoplossing. Door opslagvirtualisatie wordt de opslag niet meer op de fysieke hardware beheerd maar in een softwarelaag en is hierdoor onafhankelijk van een specifieke hardwareleverancier.

2. Foundational PaaS

De foundational PaaS is verantwoordelijk voor het uitvoeren, opslaan en het communiceren met applicaties, gericht op een hoge beschikbaarheid en daarmee horizontale schaalbaarheid. Het is een extra laag bovenop de virtuele infrastructuur uit laag 1, waardoor een hogere mate van abstractie, en daarmee automatisering, mogelijk is.

De Haan (2013) noemt de doelgroep van deze laag de *DevOps*, een samentrekking van "development" en "operations", omdat zij zowel diepgaande kennis van de hardware nodig hebben als zich bezig moeten kunnen houden met de applicaties die hiermee moeten werken.

3. PaaS

Het verschil tussen de tweede en derde laag komt door de focus. De tweede laag richt zich op de implementatie en automatisering op basis van de eerste laag en heeft voornamelijk binaire input. De derde laag richt zich met behulp van code en configuraties tot de hogere lagen zodat deze processen kunnen functioneren, zonder zich te hoeven richten op de koppeling met de hardware. Het verschil met de vierde laag ligt in het type gebruikers, die in de derde laag voornamelijk *professional developers* zijn die over programmeerkennis beschikken.

4. Model-Driven PaaS

De derde en vierde laag zijn gericht op eenzelfde soort artefacten maar de specificaties en configuraties zijn voor de vierde laag van een hoger abstractieniveau. Hierdoor is er minder technische kennis nodig en het is daardoor geschikt voor MDD. De Haan noemt deze doelgroep de *business engineer*, zij kunnen applicaties bouwen om systemen aan te sluiten of data te beheren.

5. App Services

De vijfde laag bestaat uit voorgedefinieerde diensten die door *citizen developers* geconfigureerd kunnen worden om zo samen met reeds bestaande applicaties te kunnen functioneren.

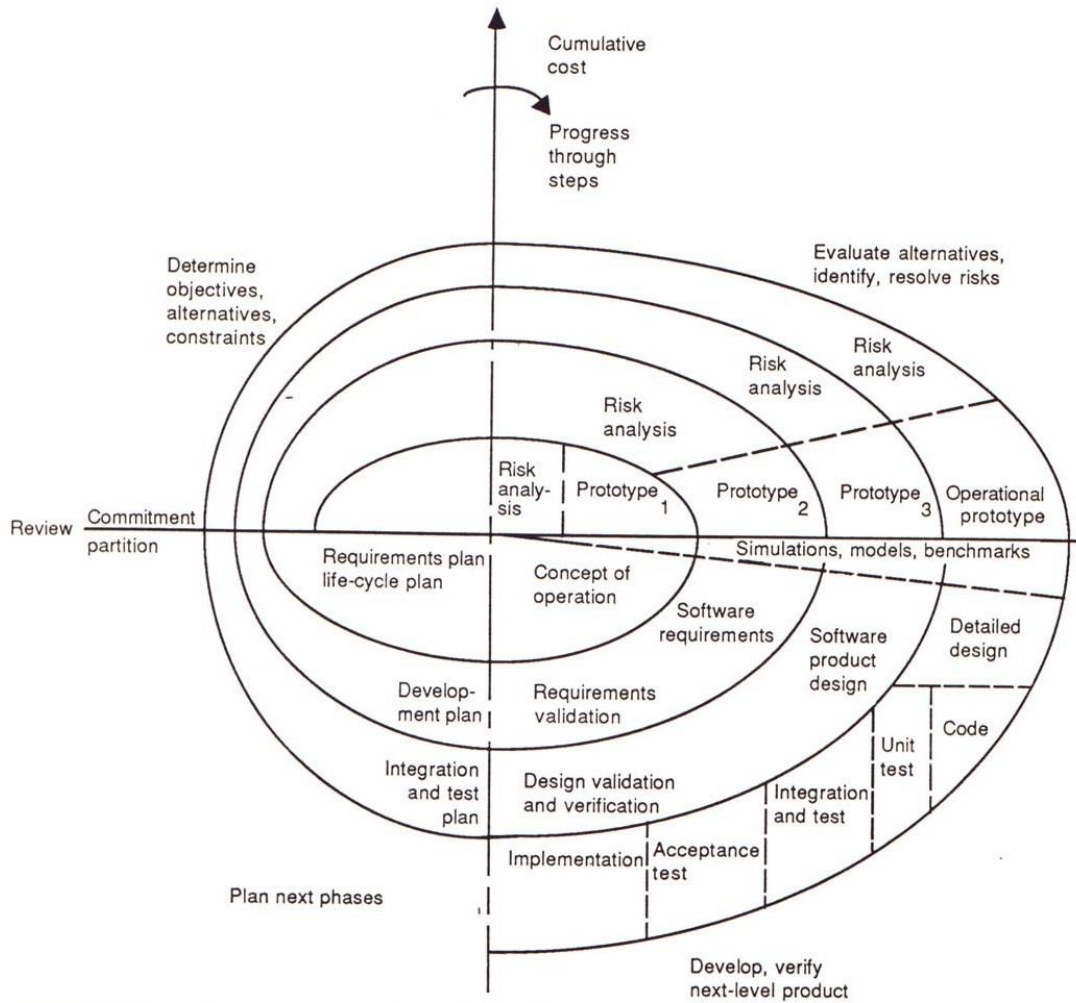
6. SaaS

De bovenste laag houdt zich uitsluitend bezig met het gebruik van toepassingen. Het gaat hier om de applicatie zelf, dit kan alle mogelijke SaaS omvatten. We spreken hier dus ook echt van een *eindgebruiker* voor wie deze hele opbouw gecreëerd is.

2.2.4 Spiral methodology

De Spiral methode van Boehm (1988) is een softwareproces model dat zich vooral richt op het beperken van risico. Het model bestaat uit vier fasen: planning, risico analyse, techniekfase en evaluatie. Binnen het proces worden de verschillende fasen herhaaldelijk in iteraties, de zogenoemde spiralen van het model, doorlopen. De basis van de spiraal begint in de planningsfase, vanaf hier worden de benodigde eisen verzameld en risico's beoordeeld. Elke opvolgende spiraal gaat verder op de basislijn van de spiraal.

In Figuur 2-4 is het model van Boehm te zien, hierin zijn duidelijk de spiraalvorm en de steeds opeenvolgende fasen te zien.



Figuur 2-4: het spiraalmodel van Boehm (Boehm, 1988).

In de planningsfase worden eisen verzameld, vaak zijn dit Business Requirement Specifications (BRS) en System Requirement Specifications (SRS). Tijdens de risicoanalyse worden de risico's van verschillende alternatieven in kaart gebracht, als tijdens de ontwikkeling het risico te groot wordt, wordt er meteen overgegaan op een alternatief. Aan het einde van de risicofase wordt er begonnen met een prototype. In de techniekfase wordt de software ontwikkeld en getest. In de evaluatiefase heeft de opdrachtgever de kans om het project te evalueren en kan deze besluiten om het project te stoppen of door te gaan met de volgende spiraal.

2.3 ONDERZOEKSORGANISATIE.

Aangezien het doel van dit onderzoek het creëren van een App store is zal er veelvuldig gebruik worden gemaakt van verschillende software. Softwareproducten zoals Microsoft Office zijn dermate bekend dat deze niet verder besproken worden. Daarnaast zal er ook van minder bekende software gebruikt worden gemaakt, waarvan de belangrijkste twee hieronder besproken zullen worden. Gevolgd door de twee belangrijkste bedrijven waarmee we binnen dit onderzoek te maken krijgen.

2.3.1 ArchiMate.

ArchiMate beschrijft zichzelf¹⁸ als een onafhankelijke modelertaal waarmee enterprise-architecturen beschreven kunnen worden en wordt sinds 2008 beheerd door The Open Group. ArchiMate biedt een eenvoudige en uniforme structuur voor het beschrijven van de relaties binnen en tussen architectuur domeinen. De taal is gebaseerd op de IEEE-standaard 1471. Door het vastleggen in een standaardtaal wordt het mogelijk om op een eenduidige manier te

communiceren over het ontwerp en de toetsing daarvan. Ook maakt de vastlegging het mogelijk verschillende versies van een architectuur te beheren.

In de relaties tussen domeinen spelen diensten een belangrijke rol. Binnen deze diensten bestaan verschillende mogelijkheden. Zo kunnen diensten verstrekt worden door bedrijven aan hun klanten, door applicaties naar bedrijfsprocessen of door technologische faciliteiten (bijvoorbeeld communicatienetwerken) naar applicaties.

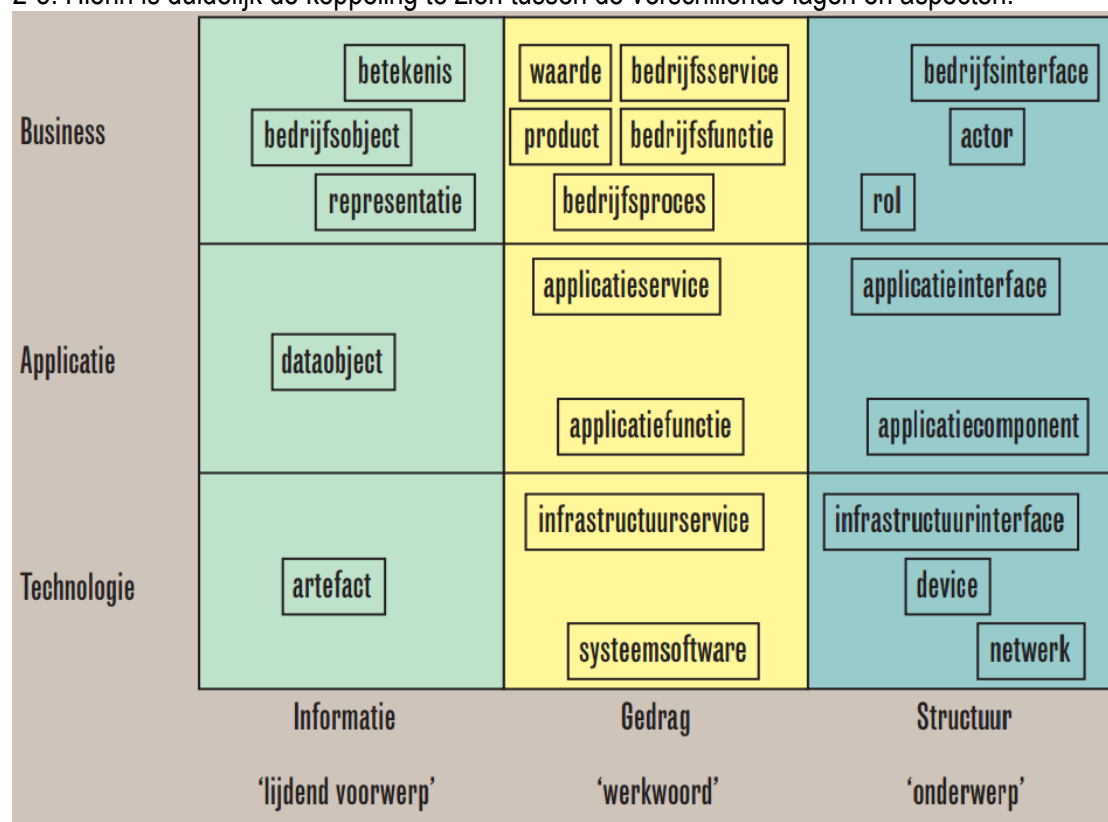
ArchiMate maakt een onderverdeling tussen drie verschillende lagen. De hogere lagen gebruiken diensten die worden geleverd door de onderste lagen. ArchiMate onderscheidt de volgende drie lagen:

- Bedrijfslaag: Al de bedrijfsprocessen, diensten (services), functies, gebeurtenissen en de onderlinge relaties. Deze laag is vooral gericht op externe klanten en gebruikers in het bedrijfsleven.
- Applicatielaag: Het applicatielandschap, bestaande uit onder meer de software en de informatieprocessen. Hiermee ondersteunt deze laag de bedrijfslaag.
- Technische laag: Hardware en communicatie-infrastructuur. Deze zijn nodig om applicaties te laten werken.

Binnen elke laag worden, van de beschreven componenten, drie aspecten vastgelegd, namelijk:

- het gedrag
- de interne structuur
- het gebruik van informatie

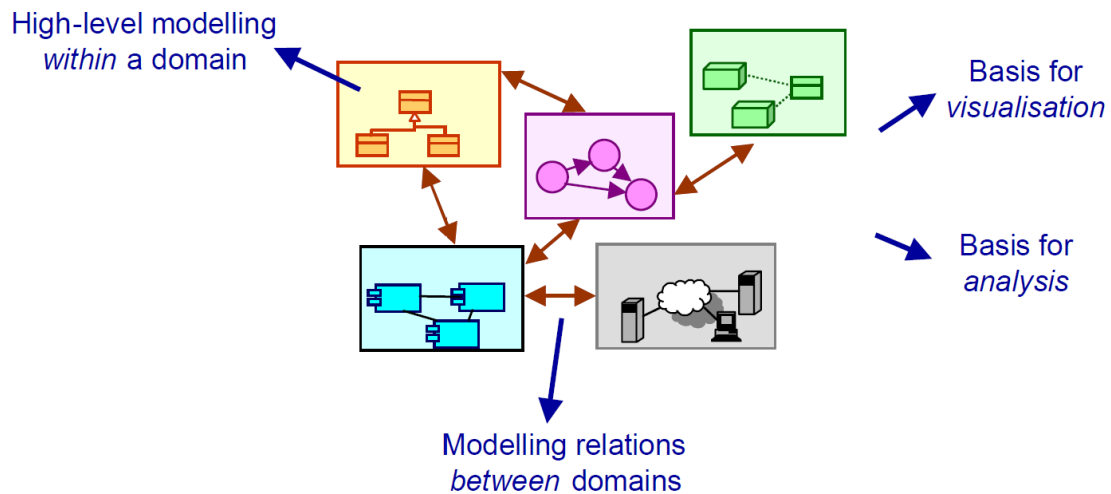
Deze lagen zijn door Jonkers et al. (2009) samengevoegd tot een matrix, die te zien is in Figuur 2-5. Hierin is duidelijk de koppeling te zien tussen de verschillende lagen en aspecten.



Figuur 2-5: Het ArchiMate architectuurraamwerk gebaseerd op Jonkers et al. (2009).

Volgens Lankhorst (2009) ligt de kracht van ArchiMate voornamelijk in onderlinge relaties die tussen verschillende domeinen gemaakt kunnen worden. In figuur 7 zijn door Lankhorst de verschillende rollen van de ArchiMate taal in beeld gebracht. Naast het High-level modeleren

binnen een model is ArchiMate ook goed in staat om de verschillende relaties tussen domeinen in beeld te brengen. Daarnaast biedt ArchiMate een goede basis voor het visualiseren en het analyseren van data zoals te zien is in Figuur 2-6.



Figuur 2-6: De rollen van de ArchiMate taal (Lankhorst, 2009).

In APPendix D is een verzameling van de voor dit gebruikte ArchiMate notaties te vinden, met bijbehorende uitleg.

2.3.2 Mendix.

Mendix is een Nederlands softwarebedrijf dat een gelijknamig PaaS op de markt brengt. Zelf omschrijft Mendix zich als een App Platform for the Enterprise. Door middel van MDD kunnen eenvoudig web- en mobiele toepassingen gebouwd, geïntegreerd en geïmplementeerd kunnen worden. Deze kunnen zowel helemaal vanaf de grond als op een al bestaande laag software gebouwd worden¹⁹.

In de begintijd van Mendix werd er vaak nauw samengewerkt met CAPE Groep, hierdoor bestaat er nog steeds een hechte band tussen deze twee bedrijven. Binnen dit onderzoek wordt Mendix echter alleen gebruikt om de App store te bouwen en blijft het bedrijf zelf buiten beschouwing. Om verwarring te voorkomen zal in de rest van het verslag bij het gebruik van het woord Mendix, steeds het softwareproduct bedoeld worden en niet het bedrijf zelf.

Het grote voordeel van Mendix is de mogelijkheid om heel snel een specifieke applicatie in elkaar te zetten. Zo kan per bedrijf een zo nauwkeurig mogelijke oplossing gemaakt worden. Omdat deze niet geprogrammeerd, maar slechts 'in elkaar geklikt' hoeft te worden, wordt er heel veel ontwikkelingstijd bespaard. Daarnaast is er tijdens de ontwikkeling al meteen te zien hoe de applicatie eruit komt te zien.

2.3.3 CAPE Groep.

CAPE Groep, gevestigd in Enschede, is een IT bedrijf dat zich voornamelijk richt op het ontwikkelen van software voor de logistieke sector. Door het gebruik van Mendix is CAPE Groep in staat om specifieke software aan te bieden die inzicht in de hele supply chain creëert. Zo zijn klanten in staat om in één oogopslag hun hele bedrijfsproces te overzien.

In tegenstelling tot andere IT-bedrijven werken er bij CAPE Groep voornamelijk medewerkers met een bedrijfskundige achtergrond. Hierdoor zijn de consultants in staat om de problemen goed te analyseren en daar doeltreffende software voor te ontwikkelen. Het uitgangspunt ligt dus altijd bij de behoeften van de klant en niet bij de mogelijkheden van de software. CAPE Groep levert 100% fit (maatwerk) oplossingen en integreert deze met bestaande software en hardware van klanten.

Door deze benadering is CAPE Groep de afgelopen jaren flink gegroeid. De groei is zelfs zo groot dat CAPE Systems Integration in 2012 door het Financieel Dagblad werd uitgeroepen tot snelst groeiend kleinbedrijf van Overijssel en daarmee een gouden gazelle in de wacht sleepte. Dat dit geen toeval was, bleek wel toen het bedrijf het jaar erop de prijs voor de tweede keer in de wacht sleepte.

2.3.4 eMagiz.

Omdat de software die CAPE Groep ontwikkelt klantspecifiek is, zijn er steeds andere behoeften. Niet alle integraties met software en hardware zijn met behulp van Mendix te modeleren. Daarom richtte CAPE Groep in 2011 haar iPaaS-dochterbedrijf eMagiz op, met als doel om business consultants in staat te stellen om integraties te kunnen modelleren. eMagiz is hiervoor nodig omdat de business consultants niet over de vereiste programmeerkennis beschikken.

Op dit moment heeft eMagiz verschillende klanten waarvan CAPE Groep nog veruit de grootste wederverkoper is. eMagiz levert op dit moment haar diensten alleen direct aan partners die daarmee in staat zijn een product voor de eindgebruikers te ontwikkelen.

2.4 BESTAANDE APP STORES.

eMagiz is niet het eerste bedrijf dat bezig is met een App store. Er zijn veel bedrijven en veel verschillende soorten App stores. Deze paragraaf dient ertoe om een beter beeld te krijgen van wat een App store nu eigenlijk is en welke toepassingen er allemaal zijn.

2.4.1 App stores in het algemeen

Als eerste is er gekeken naar iTunes van Apple, de Google Play Store van Android en de Windows Phone store. Deze zijn echt vooral gericht op de eindgebruiker en zijn sterk afhankelijk van hun community, waar ze veel mee samenwerken. Omdat eMagiz de App store in het begin vooral voor ontwikkelaars wil gebruiken, is hierna gekeken naar websites waar ontwikkelaars zelf applicaties kunnen ontwikkelen, zoals Het Developer Network²⁰ van Microsoft. Vervolgens is er gekeken naar andere websites waar je online software kunt kopen zoals Steam²¹. Als laatste is er ook naar webshops gekeken zoals bol.com, Wehkamp, Zalando en V&D.

2.4.2 Magic Quadrant for Enterprise iPaaS

In een artikel uit 2014 introduceert Gartner het Magic Quadrant voor de iPaaS markt (Pezzini, Massimo; Natis, Yefim V.; Malinverno, Paolo; Iijima, Kimihiko; Thompson, Jess; Thoo, Eric, 2014). Hierin zet Gartner verschillende iPaaS uit op basis van uitvoerende mogelijkheden en volledigheid van visie. Dit wordt gedaan aan de hand van de volgende criteria:

Voor **uitvoerende mogelijkheden** zijn dit:

- Product/Service. Dit is het kernproduct dat wordt aangeboden. Hierbij wordt er vooral gekeken naar het product zelf, de mogelijkheden waarvoor deze te gebruiken is, de kwaliteit en functionaliteit. Daarnaast wordt er gekeken naar overeenkomsten of partnerschappen die in de markt zijn aangegaan.
- Algemene levensvatbaarheid. Dit is een financiële analyse van het product waarin gekeken wordt in hoeverre het product financieel levensvatbaar is. De financiële situatie van het bedrijf zelf is hierin belangrijk met daarnaast de aantrekkelijkheid van individuele bedrijfsonderdelen, waarin door externe partijen geïnvesteerd kan worden.
- Verkoop en prijs. Hierin wordt gekeken naar het aantal voorverkopen en in hoeverre de verwachtingen, die bij deze verkopen gesteld zijn, te realiseren zijn. Daarnaast worden de prijsonderhandelingen meegenomen en de effectiviteit van het verkoopkanaal.

- Reactievermogen op marktveranderingen. De mogelijkheid tot reageren op marktontwikkelingen, de richting waarin veranderd kan worden, de flexibiliteit waarmee dit kan en de mogelijkheid tot overstappen naar concurrerende successen en trends. Behoeftes van klanten kunnen door de dynamiek van de markt namelijk snel veranderen.
- Marketing. De duidelijkheid, kwaliteit, creativiteit en efficiëntie van marketing van de producten. De boodschap moet duidelijk zijn zodat de potentiële klanten zich bewust zijn van de mogelijkheden die het product hen biedt.
- Klant ervaring. Op welke manier ervaart de klant het omgaan met het product. Dit is voor softwareproducten een zeer belangrijke factor. Het geeft namelijk aan hoe eenvoudig het is om met een product om te gaan. Dit wordt gedaan op basis van onder andere technische ondersteuning, kwaliteit van hulpdesks en mogelijkheden tot gebruikersgroepen.
- Verrichtingen. Het vermogen van de organisatie om te voldoen aan haar doelen en afspraken. Hierbij wordt er gekeken naar de kwaliteit van de organisatiestructuur, ervaring, programma's en andere mogelijkheden die bijdragen aan het effectief en efficiënt leiden van een bedrijf op een permanente basis.

Voor de **volledigheid van visie** gelden de volgende criteria:

- Marktinzicht. Het vermogen van de verkopers om het product aan te prijzen bij klanten die verstand van de markt hebben en dus weten wat hun wensen en behoeften zijn.
- Marketingstrategie. Een duidelijke, onderscheidende en consequente manier van communiceren naar de klant toe.
- Verkoopstrategie. De strategie die voor de verkoop van de producten gehanteerd wordt. Hier wordt gekeken of het juiste netwerk gebruikt wordt, of er sprake is van directe of indirecte verkoop, marketing, service en de communicatiemiddelen die hierbij gebruikt worden.
- Productaanbod naar de klant. De aanpak van de verkoper met de nadruk op de differentiatie, functionaliteit, methodologie en ontwikkeling van de huidige en toekomstige eisen.
- Bedrijfsmodel. De soliditeit en logica van het ondernemingsplan.
- Verticale/industriële strategie. De vertalingsmogelijkheden naar verschillende marktsegmenten.
- Innovatie. Mate van directe, gerelateerde of complementaire vernieuwingen.
- Geografische strategie. Aan welke groepen of regio's wordt het product aangeboden en ondersteund. Waarbij de vraag gesteld kan worden of het gebruik uitsluitend voor op het bedrijf is en of het ook te gebruiken is voor dochterondernemingen.

Al deze criteria leiden samen tot het Magic Quadrant in Figuur 2-7.



Figuur 2-7: Gartner's Magic Quadrant van januari 2014.

In het Magic Quadrant maakt Gartner onderscheid tussen vier segmenten:

- **Leiders**, helemaal rechtsboven staan de bedrijven die op beide criteria hoog scoren. Ze hebben enkele honderden of duizenden gebruikers van hun iPaaS. Ze hebben een solide reputatie en zijn zich bewust van de wensen van de markt. Hun visie is meestal progressief, maar niet revolutionair, in vergelijking met andere spelers in de markt. Door het dynamische karakter van de markt verwacht Gartner dat binnen de komende twaalf maanden nieuwe spelers het leiderskwadrant zullen betreden. Echte leiders zijn niet altijd de beste optie, in veel gevallen kan een andere aanbieder beter aan de eisen van een klant voldoen. Denk hierbij aan geografische ligging, integratiemogelijkheden binnen het bedrijf, affiniteit met het huidige en/of de ontwikkeling van applicatie portfolio, het niveau van ondersteuning en reactievermogen. Daarnaast moet er soms ook rekening gehouden worden met reeds gemaakte commerciële afspraken.
- **Uitdaggers**, deze scoren hoog op het uitvoerend vermogen, maar laag op de volledigheid van een visie. Dit zijn vaak grote, gevestigde, bedrijven die al jaren actief zijn in de markt en vaak ook een voortrekkersrol hebben gespeeld in de vorming van het huidige iPaaS landschap. Gartner beweert echter dat deze bedrijven minimale toekomstplannen voor deze markt hebben en de ontwikkelingen in de markt niet nauwkeurig volgen. Hierdoor

zijn ze niet altijd genoeg op de hoogte van de wensen van de klanten, waardoor hun producten vaak minder compleet zijn dan die van de marktleiders.

Als deze uitdagers hun aandacht naar de iPaaS-markt zouden verleggen, zouden ze binnen twaalf maanden door kunnen schuiven naar het kwadrant van de leiders.

- **Visionairs**, begrijpen de specifieke eisen van de markt en hebben vaak een achtergrond in de traditionele on-premise integratie software en hebben deze producten door overnames en doorontwikkeling toepasbaar gemaakt voor Cloudgebruik. Visionairs zien hun iPaaS product vaak als een onderdeel van hun Cloud-producten, zoals SaaS of PaaS producten.

Ze zijn echter nog niet volwassen genoeg om hun product met de hele markt te delen. Dit komt vaak door een gebrek aan marketing en verkoopstrategieën.

- **Nichespelers**. Meer dan 50 procent van de aanbieders in dit Magic Quadrant is beoordeeld als nichespeler. Dit komt omdat de toetredingsdrempel vrij laag is, zelfs als men kijkt naar de vele mogelijkheden waarop men zich kan onderscheiden in deze dynamische markt. Hierdoor zijn de meeste nichespelers kleinere bedrijven, vaak startups, die pas in de afgelopen jaren de iPaaS-markt betreden hebben.

De andere nichespelers zijn vaak aanbieders met een achtergrond in de traditionele integratiemarkt. Maar omdat deze de dynamiek van de markt nog niet goed begrijpen, zijn ze hierin nog niet gegroeid. Deze bedrijven zouden snel kunnen groeien door een overname van een visionair, omdat deze vaak niet zo groot zijn en een gebrek aan financiële mogelijkheden hebben om echt door te groeien.

2.4.3 Concurrenten & Partners eMagiz

Op basis van het algemene onderzoek en het Magic Quadrant van Gartner is een lijst van belangrijkste concurrenten en partners van eMagiz ontstaan.

Het uitgangspunt hiervan is de mogelijk toegevoegde waarde, die de toepassingen van deze bedrijven voor eMagiz moet hebben. Vanuit het Magic Quadrant zijn vooral MuleSoft en snapLogic naar voren gekomen, deze bedrijven hebben een uitgedachte visie en lopen daarin voorop in de markt. eMagiz ziet zichzelf ook als klein bedrijf met een grote visie, ze hebben verstand van hun marktsegment en breiden hun kennis steeds verder uit.

De belangrijkste bedrijven waar voor de ontwikkeling van de App store naar gekeken kan worden zijn:

- **MuleSoft**
Een integratieplatform voor het 'connecten' van alle mogelijke applicaties (en applicatielagen), databronnen of API's, zowel in de Cloud als 'on-premises' en de koppeling hiertussen²².
- **snapLogic**
Een softwarebedrijf dat gericht is op Cloud-integratieproducenten en klanten de mogelijkheid biedt om Cloud gebaseerde data en applicaties te verbinden met 'on-premises' bedrijfssystemen²³.
- **Windows Dev Center**
Windows 8 is gemaakt voor zowel computers als tablets. Hierdoor is er een keuze gemaakt om veel met apps te werken. Windows geeft in hun Dev Center de mogelijkheid tot het maken en uploaden van eigen ontwikkelde apps²⁴.

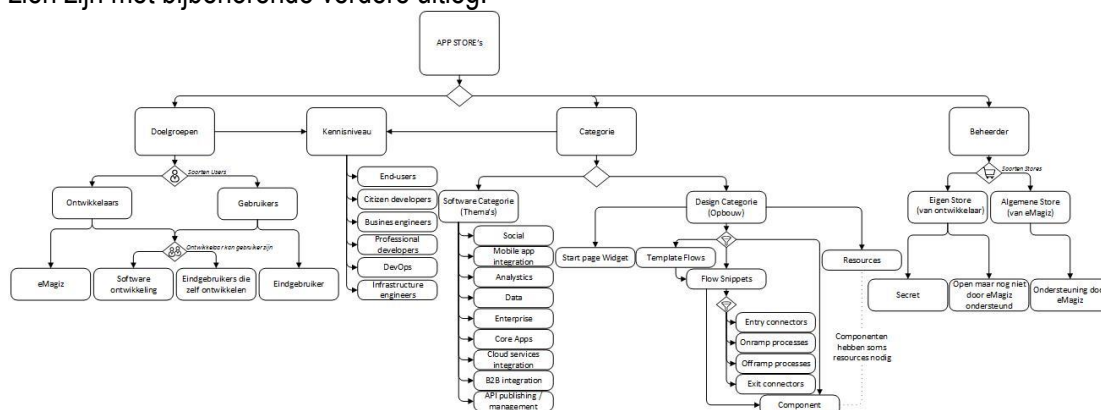
- **Mendix**
Mendix heeft op dit moment ook een App store. Op dit moment wordt de algemene store, door medewerkers van CAPE Groep, echter nog niet veel gebruikt. De eigen store van het bedrijf wordt echter al wel gebruikt.
- **IFTTT**
'If-this-then-that' stelt gebruikers in staat tot het creëren en delen van zogenoemde *recepten*. Hierdoor kunnen de gebruikers zelf een koppeling opstellen tussen verschillende online applicaties. Voorbeelden zijn het automatisch downloaden naar dropbox van een foto waarin je op Facebook getagd bent of het generen van een automatische mail bij een gemiste oproep²⁵. Dit staat ook wel bekend als Integration as a Service.
- **OpenNebula**
Deze open source software maakt het beheren mogelijk van een virtuele infrastructuur van publieke en private IaaS (Moreno-Vozmediano, Montero, & Llorente, 2012).
- **Spring**
Net als OpenNebula is Spring Integration een open source software product. Spring richt zich vooral op het transformeren van informatie tussen het formaat waarin de zender zijn data verstuurt en het formaat waarin de ontvanger het wil hebben²⁶.
- **Tableau**
Een door middel van 'drag and drop' *data analyse self-service business intelligence application*. Tableau wordt door Mendix als voorbeeld gezien voor het ontwikkelen van de Mendix App store.
- **GitHub**
Is een web-gebaseerd source code management (SCM) systeem waarmee ontwikkelaars samen aan software kunnen werken²⁷.

3 Ontwerp en Ontwikkeling.

Volgens Peffers moet in deze stap een design research artefact gecreëerd worden. Voor deze activiteit moet men zich in eerste instantie richten op de gewenste functionaliteiten van het artefact en de architectuur ervan. Vervolgens kan men aan de slag gaan met het daadwerkelijk creëren van het artefact. Voor dit onderzoek ligt de nadruk echter meer op de functionaliteiten die de App store moet hebben die vervolgens in een reference architecture worden weergegeven. Natuurlijk is in het proces ook een prototype van de App store ontwikkeld, waarvan de ontwikkeling in APPendix D terug te vinden is.

3.1 DE ALGEMENE SITUATIE.

Voordat begonnen kan worden met het uitzoeken van de functionaliteiten is eerst, samen met de medewerkers van CAPE Groep en eMagiz de algemene situatie onderzocht, weergegeven in Figuur 3-1. In Figuur 3-2 tot en met Figuur 3-6 zal steeds een uitvergroting van Figuur 3-1 te zien zijn met bijbehorende verdere uitleg.

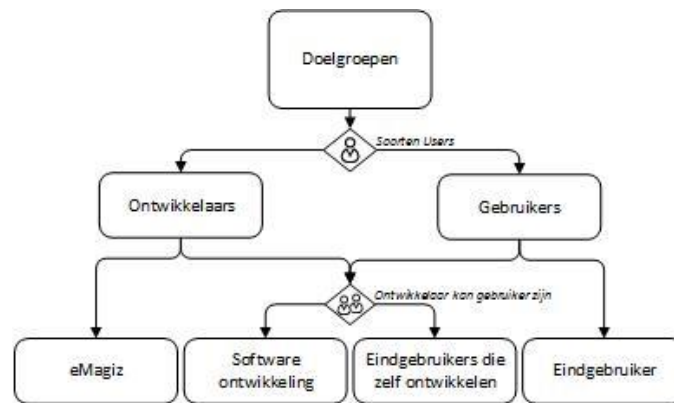


Figuur 3-1: Overzicht van de aspecten van een App store.

Vier aspecten dragen samen bij aan de App store. Een App store heeft namelijk ten eerste een doelgroep, voor wie is de App store gemaakt en wie hebben er allemaal toegang toe. Omdat hier verschillende doelgroepen onderscheiden worden, bestaat er ook een verschil in kennisniveau tussen deze doelgroepen. Dit heeft ten eerste met de achtergrond van deze doelgroepen te maken en ten tweede met de mate waarop gebruik gemaakt kan worden van de App store. Hierdoor zijn verschillende categorieën opgesteld voor de verschillende doelgroepen, met de verschillende kennisniveaus. Deze categorieën zijn weer verder onderverdeeld in een softwarecategorie en een designcategorie. De Softwarecategorie kan gezien worden als een soort thema. De designcategorie is een onderverdeling van het formaattype van de verschillende soorten content. Als laatste moet de App store ook beheerd worden, dit wordt aan de ene kant gedaan door eMagiz zelf, maar er is ook voor partners de mogelijkheid om content in hun eigen store toe te voegen.

3.1.1 Doelgroepen.

Zoals te zien is in Figuur 3-2 is er bij de doelgroepen een onderscheid gemaakt tussen de ontwikkelaars en de gebruikers. Dit onderscheid is gemaakt omdat de ontwikkelaars meer bezig zullen zijn om content aan de App store toe te voegen, die vervolgens weer door de gebruikers uit de App store gedownload kan worden. Dit zijn twee heel verschillende processen die in kaart gebracht moeten worden.

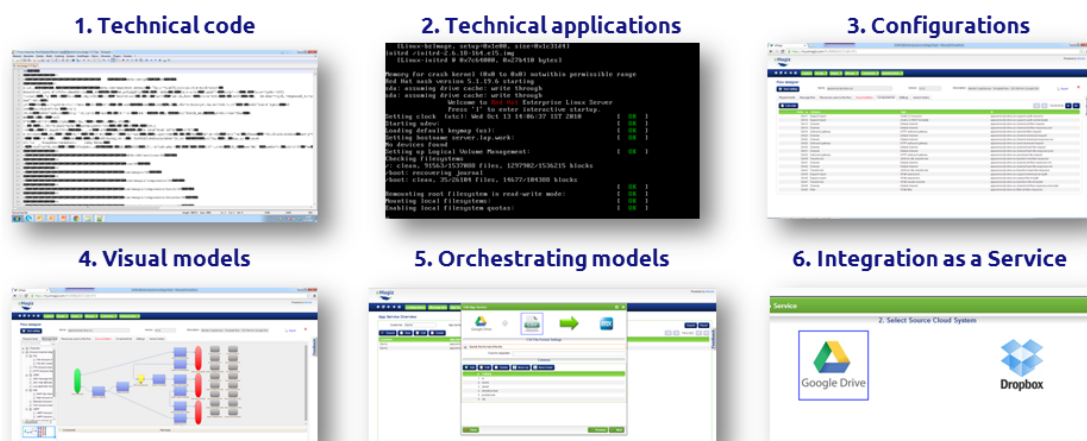


Figuur 3-2: De verschillende doelgroepen uit Figuur 3-1.

Daarnaast zijn de ontwikkelaars en gebruikers zelf ook weer onderverdeeld. Als eerste zijn de medewerkers van eMagiz alleen bezig om aan het product eMagiz te werken en zij voegen zo dus content aan de App store toe. Daarnaast zijn er partners van eMagiz die zelf in opdracht van een klant software ontwikkelen en zo nieuwe content, die ze ontwikkeld hebben, toe kunnen voegen aan de App store. Om deze nieuwe content te ontwikkelen kunnen de partners gebruik maken van reeds bestaande content uit de App store. Deze partners moeten dus zowel de mogelijkheid hebben om content toe te voegen als de mogelijkheid om deze content weer te downloaden. De derde optie zijn klanten die zelf eindgebruiker zijn maar ook voor zichzelf software ontwikkelen. Zij maken dus alleen gebruik van eMagiz en hebben net als de partners zowel de rol van ontwikkelaar als die van gebruiker. Als laatste groep zijn de eindgebruikers gedefinieerd. Op dit moment besteedt eMagiz hier als zodanig niet of nauwelijks aandacht aan. Deze eindgebruikers zouden een bepaalde functie zelf kunnen toevoegen aan hun scherm, hier is dus sprake van een citizen developer. Zo kan het voor een eindgebruiker handig zijn om slechts een bepaalde grafiek op zijn scherm steeds in beeld te hebben. Deze gebruiker zou dan via de App store deze functie, als een soort widget, kunnen downloaden.

3.1.2 Kennisniveau.

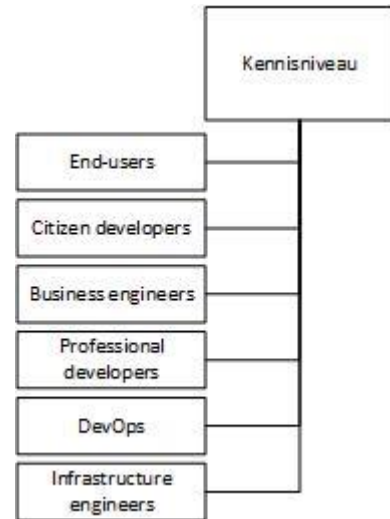
Voor het definiëren van kennisniveaus is gebruik gemaakt van de onderverdeling die Johan de Haan in zijn artikel maakt (Haan, 2013). Zo zijn er zes hiërarchische lagen gedefinieerd waar van onder naar boven steeds minder technische programmeerkennis vereist is. In Figuur 3-3 is een visuele weergave van de interface weergegeven die een beter beeld geeft van de verschillende kennisniveaus.



Figuur 3-3: Visuele weergave van de interface na kennisniveau

Vervolgens is in Figuur 3-4 een koppeling gemaakt van deze kennisniveaus naar de verschillende doelgroepen:

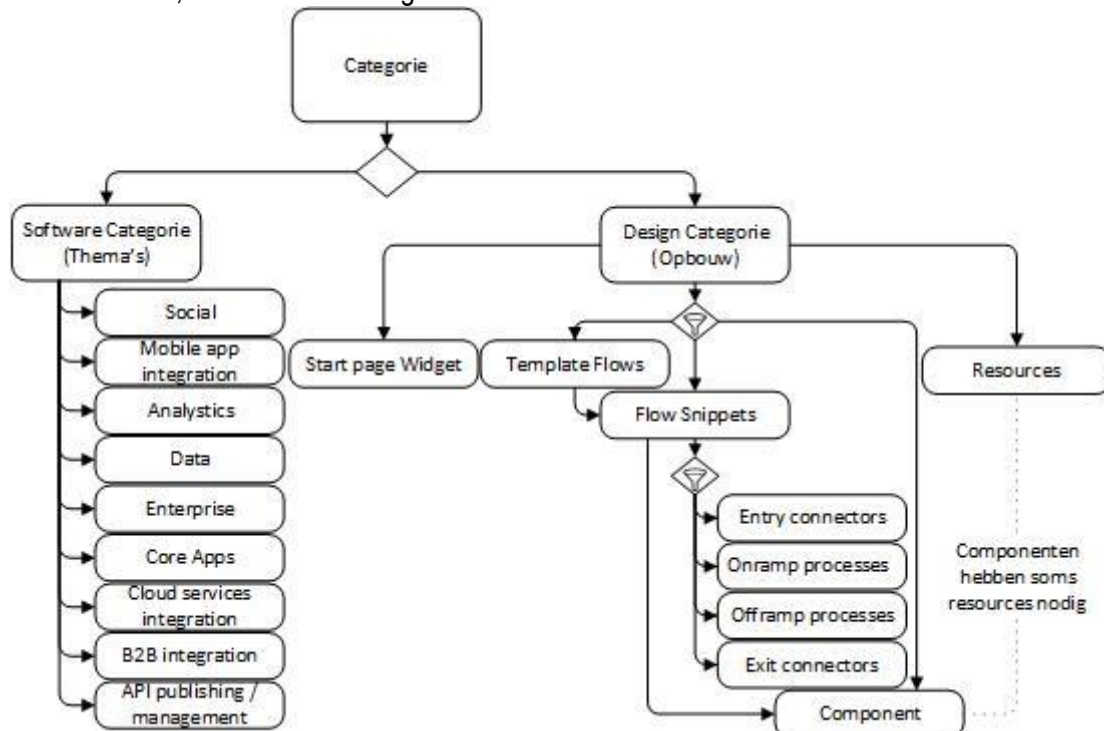
- De *infrastructure engineers*, aangezien deze de onderste laag vormen heeft deze groep het meeste verstand van programmeren. Zij zijn in staat om met technische code om te gaan.
- *DevOps*, houden zich bezig met de integratie tussen de software en de diensten die deze aanbieden en treden vaak op als er gebruik gemaakt wordt van Cloud computing. Ze zijn in staat om technische applicaties te onderhouden.
- *Professional developers*, zijn verantwoordelijk voor de configuraties tussen de technische code en de visuele modellen waarvan de business engineers gebruik maken.
- *Business engineers*, kunnen op basis van de configuraties visuele modellen maken op basis van MDD.
- *Citizen developers*, zijn in staat om door middel van een paar schermen koppelingen op te stellen waardoor ze hun applicatie persoonlijk kunnen specificeren.
- *End-users*, hebben het minst verstand van software en gebruiken Integration as a Service diensten, zoals IFTTT, waarin uitsluitend twee programma's geselecteerd moeten worden waarna het programma de rest doet.



Figuur 3-4: de verschillende kennisniveau's uit Figuur 3-1.

3.1.3 Categorie.

Zoals al kort is aangegeven is bij de categorieën een onderverdeling gemaakt tussen softwarecategorieën en designcategorieën. De designcategorieën zijn daarna weer verder onderverdeeld, zoals te zien in Figuur 3-5.



Figuur 3-5: De verschillende Categorieën uit Figuur 3-1.

Omdat het nog niet mogelijk was deze categorieën te kunnen testen zou het goed mogelijk kunnen zijn dat deze nog niet volledig zijn. Door een verschuiving in de markt zouden er bijvoorbeeld andere softwarecategorieën bij kunnen komen. Daarnaast was het nog niet mogelijk om de designcategorieën in de praktijk te kunnen testen. Het zou dus mogelijk kunnen zijn dat er een handige tussencategorie over het hoofd wordt gezien, maar dit zal in vervolgonderzoek dieper onderzocht moeten worden.

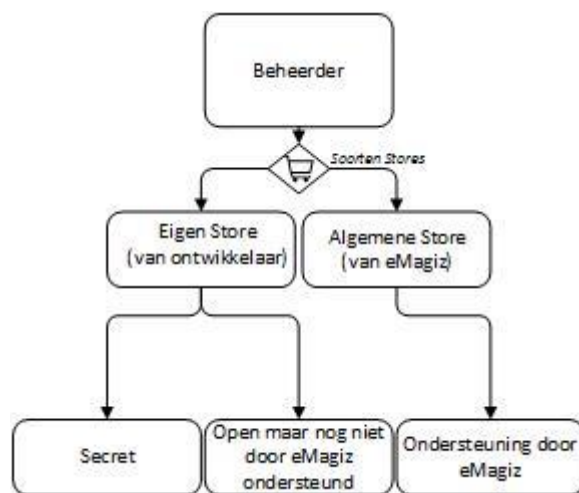
3.1.3.1 **Softwarecategorie.**

De thema's die hieronder gedefinieerd zijn, zijn aan de hand van marktonderzoek tot stand gekomen. Dit zijn de meest gebruikte thema's die zowel door eMagiz als door concurrenten (§2.4) gehanteerd worden.

- Social, hierin zijn voornamelijk koppelingen te vinden tussen de content en social media, zoals facebook en twitter.
- Mobile app integration, maakt een koppeling naar mobiele apparaten zoals mobiele telefoon of tablet.
- Analytics, hiermee kan met data analyseren.
- Data, hieronder kunnen veel toepassingen vallen, zoals het verzamelen en verwerken van data.
- Enterprise,
- Core Apps, bij veel concurrenten was er een korte verzameling van content die belangrijk is of vaak gebruikt wordt.
- Cloud service iteration, koppelingen met cloud diensten.
- B2B integration, business to business integration,
- API publishing/mangement,

3.1.3.2 **Designcategorie.**

De content uit een App store kan heel breed worden toegepast. De te downloaden content zal in de praktijk een van tevoren opgeslagen softwarecomponent zijn, die het maken van nieuwe content moet versimpelen. Maar op welke manier deze content van toegevoegde waarde is, staat nog niet vast. Zo zou een gebruiker ervoor kunnen kiezen om een hele template te downloaden, waarna hij alleen nog wat gegevens moet invullen zodat de template ook binnen de software van de gebruiker werkt. Het kan echter zijn dat de gebruiker alleen een deel van de template nodig heeft en dus slechts een specifiek deel van de template wil downloaden. Deze zou dus ook apart te downloaden moeten zijn in de vorm van een snippet. Men zou de template dus kunnen zien als een verzameling snippets. Maar deze snippets zijn zelf ook weer uit verschillende componenten opgebouwd, die uiteindelijk wederom afzonderlijk te downloaden zouden kunnen zijn. Vanwege de complexiteit hiervan is besloten de verdere opbouw van de content voor dit onderzoek buiten beschouwing te laten. De aandacht van het onderzoek gaat primair uit naar het ontwikkelen van een App store en richt zich als zodanig niet op de software van eMagiz zelf.



Figuur 3-6: de verschillende Beheerders uit Figuur 3-1.

3.1.4 Beheer.

Al de content die in de App store staat, moet ergens beheerd worden. Zo moet het bijvoorbeeld mogelijk zijn vragen te stellen over bepaalde functionaliteiten.

Er zijn verschillende gebruikers die content aan de App store, in het algemeen, kunnen toevoegen. Hierin kan de keuze gemaakt worden tussen verschillende App stores, zoals te zien in Figuur 3-6, welke elk hun eigen voordelen hebben.

De meest voor zichzelf sprekende store is de algemene App store van eMagiz zelf. Al de content die in deze store staat is door eMagiz geverifieerd en voldoet dus aan alle eisen. Hierdoor is het ook mogelijk om aan eMagiz vragen over deze content te stellen. Daarnaast is deze App store voor iedereen toegankelijk waardoor je als uploader een breed publiek bereikt.

De andere mogelijkheid voor een ontwikkelaar is het uploaden naar de company App store. Deze kan door zowel de medewerkers van het bedrijf gebruikt worden als door externe partners van eMagiz. eMagiz zelf is niet verantwoordelijk voor deze content, waardoor deze aan minder strenge eisen hoeft te voldoen. Het voordeel voor de ontwikkelaar hiervan is dat zijn content veel sneller online kan staan. Daarna kan er altijd nog een aanvraag gedaan worden bij eMagiz om de content te controleren en naar de algemene App store te verplaatsen. Partners kunnen op deze manier bij collega's kijken waar ze mee bezig zijn of zien dat sommige dingen al ontwikkeld zijn.

De tweede mogelijkheid voor de ontwikkelaar is het uploaden naar een afgeschermd, zogenoemde *secret*, store. Deze is alleen toegankelijk voor medewerkers binnen het bedrijf en zodoende verborgen voor concurrenten.

3.2 FUNCTIONALITEITEN EN EISEN.

Nadat er duidelijkheid bestaat over waaruit de App store moet bestaan, moeten de functionaliteiten gedefinieerd worden. Dit zal gedaan worden aan de hand van user story's en inspiratie van reeds bestaande App stores (§2.4.4).

3.2.1 User stories.

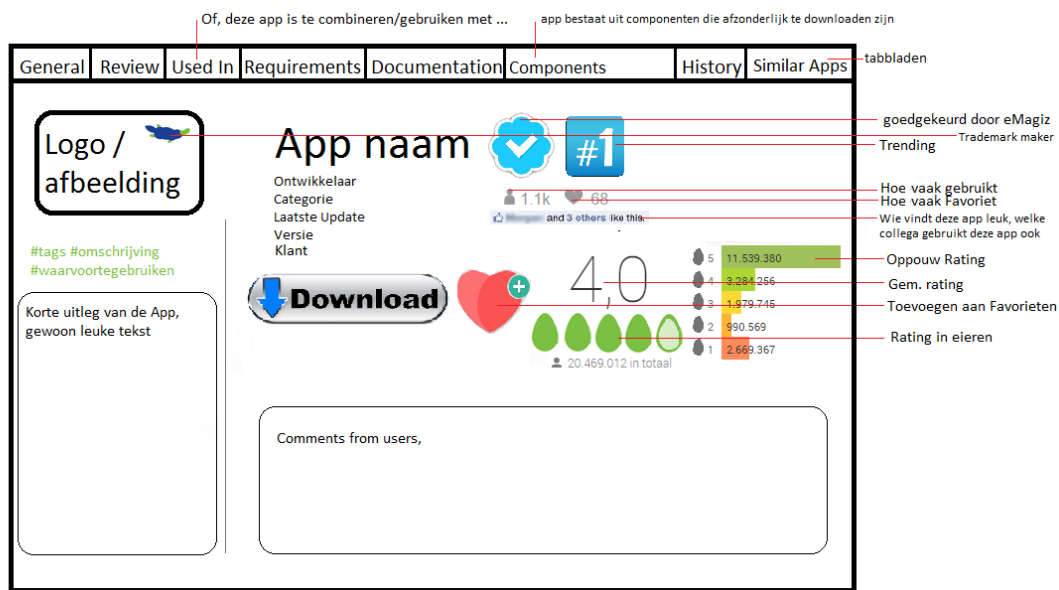
Aan de hand van de doelgroepen en kennisniveaus (§3.1.2 respectievelijk §3.1.3) zijn in overleg met de medewerkers van CAPE Groep verschillende user stories opgesteld, die terug te vinden zijn in APPendix A. Voor de verschillende doelgroepen werd er steeds een onderscheid gemaakt tussen een Admin, die voornamelijk voor het beheer verantwoordelijk was en een daadwerkelijkere gebruiker, die toegang tot de content zelf zou moeten hebben. Al deze user stories zijn aan meerdere medewerkers voorgelegd en waar nodig aangepast.

Om een praktisch beeld van deze user stories te krijgen zijn deze verder uitgewerkt door deze op een groot bord uit te tekenen, zoals te zien en verduidelijkt in APPendix B. Dit gaf een veel helderder beeld dan de afzonderlijke user stories. Zo zat er namelijk vaak overlap tussen de functies per gebruikers. Daarnaast werd het meteen duidelijker hoe de verschillende functionaliteiten met elkaar moeten kunnen communiceren.

3.2.2 Mock-up.

Om een beeld te krijgen van hoe de content er uiteindelijk in de App store uit kan komen te zien is een mock-up gemaakt. Deze is voornamelijk gebaseerd op de weergave van content in al bestaande App stores. Voor de visuele weergave is vooral gekeken naar de App stores van de mobile besturingssystemen, zoals Android en Apple. Voor de inhoudelijke weergave is voornamelijk gekeken naar spelers uit het magic quadrant van Gartner. Belangrijke inspiratie is opgedaan uit de App stores van onder andere Mendix, snapLogic en MuleSoft.

Alle mogelijke functionaliteiten zijn weergegeven in Figuur 3-7. De bedoeling is dat het uiteindelijke ontwerp sterke overeenkomsten vertoont met deze mock-up.



Figuur 3-7: Mock-up van de content weergave.

Hierbij moet wel de kanttekening geplaatst worden dat misschien niet alle functionaliteiten haalbaar zijn binnen deze opdracht. Deze zullen voornamelijk betrekking hebben op de opbouw van verschillende content en de meer 'community' gerichte toepassingen zoals de rating of het weergeven van een trending applicatie.

3.2.3 Key features.

De user stories, met bijbehorende tekening en de Mock-up zorgden ervoor dat al snel bleek dat er een paar key features waren. Dit werd vooral duidelijk uit de reeds gemaakte onderverdeling tussen administrators en gebruikers. Met andere woorden; er is een kant voor het beheer en een kant voor het gebruik. De gebruikerskant kan nog verder worden gespecificeerd naar het "Uploaden, Downloaden en Social" van de content.

3.2.3.1 Beheer.

De administrator moet in staat zijn om verschillende aspecten te beheren. Dit geldt niet alleen voor de administrator van eMagiz zelf, maar ook de andere administrators van partners en klanten. Het beheren kan vooral gedaan worden door middel van licenties en het verlenen van rechten. Deze rechten geven vervolgens toegang tot bepaalde gedeeltes van de App store. Verder is het wenselijk dat bijvoorbeeld de categorieën van de store kunnen worden bijgehouden of bepaalde content kan worden aangepast/afgewezen. Het aanpassen van de content is vooral van belang voor de content die in de algemene store geplaatst kan worden. Medewerkers van eMagiz kunnen zo, waar nodig, de content aanpassen zodat deze door iedereen goed gebruikt kan worden en ook daadwerkelijk door eMagiz kan worden ondersteund.

3.2.3.2 Gebruik.

Uiteindelijk is het daadwerkelijke gebruik van de App store natuurlijk de belangrijkste functie. Op welke manier hier gebruik van wordt gemaakt is echter gebruikers afhankelijk. Zo zal de groep ontwikkelaars zich vooral bezig houden met het uploaden van nieuwe content en zullen de partners van eMagiz de App store vooral gebruiken om zo sneller nieuwe producten te kunnen maken.

3.2.3.2.1 Uploaden.

Voordat enige vorm van content uit een App store gedownload kan worden moet deze eerst aan de App store worden toegevoegd. Het primair uploaden van de alleen de content is echter niet voldoende. Zo is het wenselijk dat een bepaalde content ook een naam krijgt en dat er wordt aangegeven in welke store, bijvoorbeeld secret of openbaar, deze moet komen te staan. Daarnaast is enige informatie en documentatie betreffende de content zelf en hoe deze te gebruiken is, ook gewenst.

Natuurlijk heeft niet alles een even grote prioriteit, zo is in Appendix C te zien dat bijvoorbeeld het toevoegen van screenshots een lagere prioriteit (3) heeft dan het aangeven waarmee deze content te combineren is (prioriteit 2).

3.2.3.2.2 Downloaden.

Uiteindelijk dient de App store natuurlijk voor het downloaden van content. Voordat iemand echter iets kan downloaden, moet de content gevonden worden. Hiervoor zijn in eerste instantie de verschillende categorieën een uitkomst. Naarmate de App store echter blijft groeien, zal het steeds moeilijker worden voordat de juiste content gevonden is. Daarnaast kunnen bij reeds gebruikte content suggesties komen te staan met andere content die ongeveer hetzelfde werkt, een verbeterde versie is of goed met deze content te combineren valt.

Als een bepaalde content vaak gebruikt wordt, is er ook de mogelijkheid om deze toe te voegen aan je 'eigen' store, als een soort bookmark.

In de toekomst willen partners misschien hun content alleen betaald beschikbaar maken. Hierdoor is het voor partners aantrekkelijker om content te ontwikkelen en daardoor de App store voor meer mensen aantrekkelijk te maken.

Omdat de gebruiker voor de download nog niet precies weet of dit de juiste content is kan de ontwikkelaar ervoor kiezen om een trial periode in te stellen. Zo kan de gebruiker de content eerst testen en kijken of hij de content nodig heeft. Daarna zou er een betaling kunnen plaatsvinden of een andere vorm van compensatie, dit valt nu echter buiten de scope van dit onderzoek.

3.2.3.2.3 'Social'.

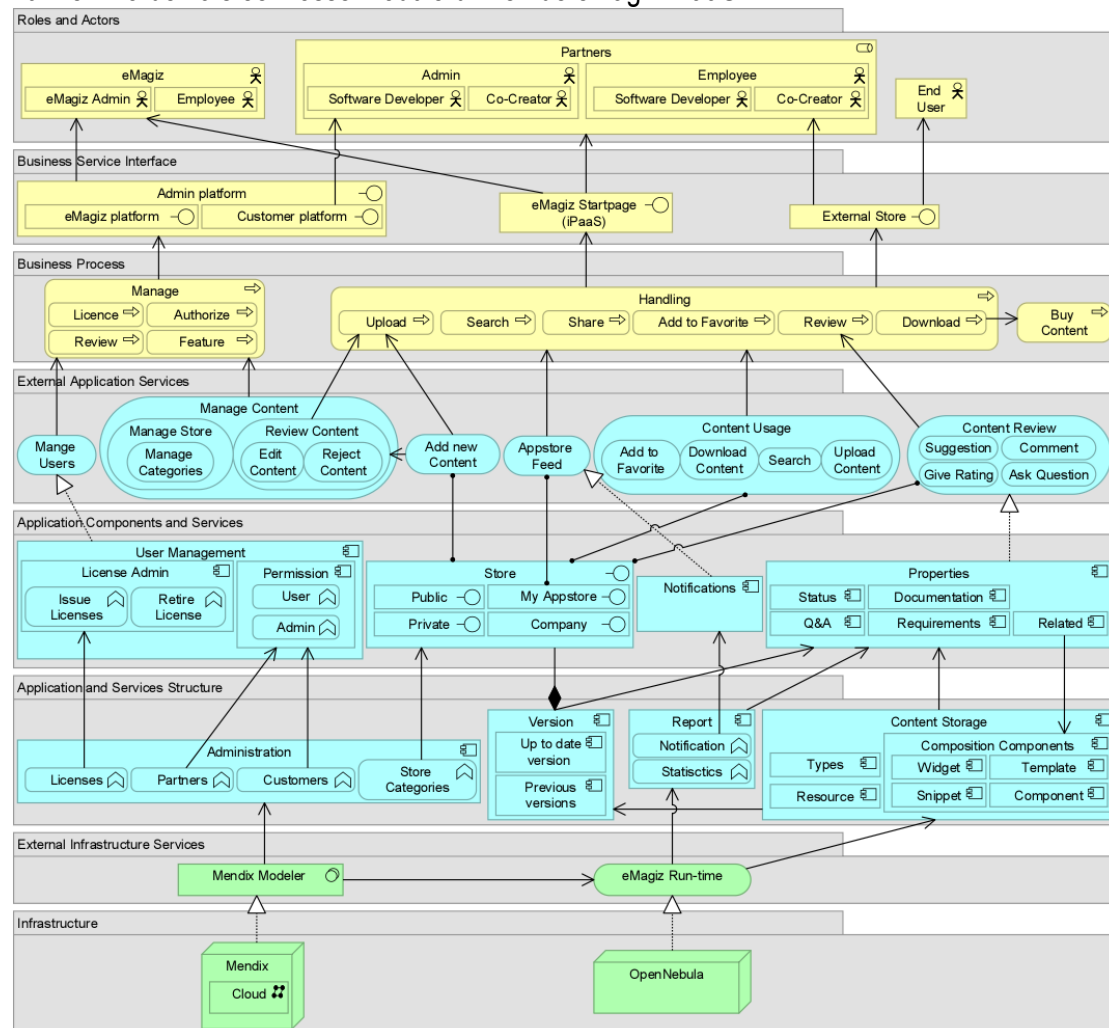
Om het makkelijker te maken en om ervoor te zorgen dat de App store ook binnen de community gebruikt wordt, zijn er ook een paar 'social' aspecten. Zo is het mogelijk om een bepaalde content te 'reviewen'. Dit kan door het maken van een comment of het geven van een rating. Dit geeft andere gebruikers een beter beeld van de content en geeft ontwikkelaars de kans om op de comments in te spelen, door eventuele bugs op te lossen of nieuwe features toe te voegen. Binnen de community kunnen gebruikers ook suggesties geven over content die te combineren is of bijvoorbeeld dezelfde functie heeft maar dan voor een ander systeem. Als een bepaalde content goed bevalt, kunnen gebruikers deze content ook 'sharen' met andere gebruikers, zodat deze hiervan ook op de hoogte zijn.

3.3 REFERENCE ARCHITECTURE.

Nu het beeld omtrent de App store wat duidelijker is geworden kan dit beeld worden omgezet naar een reference architecture, in dit geval ArchiMate (zie §2.3.5). Dit geeft een beter beeld van hoe de App store moet gaan werken en helpt bij het maken ervan.

Dit ArchiMate model is opgebouwd uit de standaard swimlanes die door The open Group (december 2013) zijn gegeven en alom gebruikt worden. Swimlanes zijn de grijze horizontale 'banen' waarbinnen de verschillende processen gemodelleerd zijn. Een korte uitleg van de notaties en relaties die binnen ArchiMate gebruikt worden zijn terug te vinden in APPendix D – ArchiMate notations.

Omdat deze App store binnen de al werkende eMagiz iPaaS gebruikt zal worden, richt dit model zich zoveel mogelijk op de werking van de App store en niet op de iPaaS zelf. Dit is voornamelijk terug te zien in de Technology Layer, die slechts minimaal is uitgewerkt. Dit model zou gezien kunnen worden als een losse module binnen de eMagiz iPaaS.

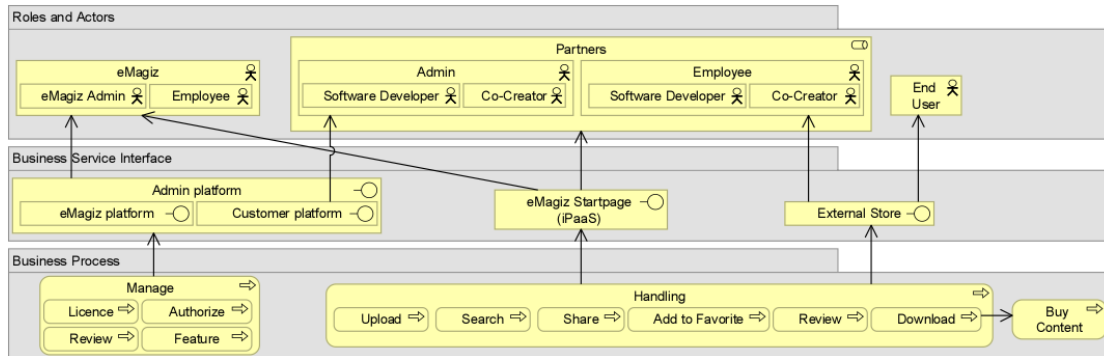


Figuur 3-8: De Reference Architecture van de App store.

3.3.1 Business Layer.

De bovenste laag is die van de rollen en actoren. Deze zijn gebaseerd op de in §3.1.1 gedefinieerde doelgroepen:

- The eMagiz employee's
- The Partners
- The Enduser



Figuur 3-9: De Business Layer van de Reference Architecture uitgelicht.

Er wordt onderscheid gemaakt op basis van hoe mensen omgaan met het systeem, dit zijn de verschillende gebruikersrollen. Om het systeem werkend te krijgen en te houden zullen de medewerkers van eMagiz de meeste toegang tot het systeem hebben.

De verschillende partners zullen zowel content naar de store willen uploaden, als downloaden, zodat ze de content kunnen (her)gebruiken.

In de Business Service Interface layer is weergegeven via welke interfaces de verschillende doelgroepen toegang hebben tot de App store. De meest voor de hand liggende is via de iPaaS van eMagiz. Hierin wordt immers de nieuwe content gemaakt. Daarom is dit ook de plek waar bestaande content hergebruikt wordt en waar de nieuw gecreëerde content kan worden toegevoegd aan de store.

Daarnaast heeft de administrator van eMagiz ook toegang nodig tot het systeem achter de App store (in de iPaaS van eMagiz). Hier kan hij instellingen aanpassen en gebruikers toegang tot de App store verschaffen. De laatste actor, die in dit verslag redelijk buiten beschouwing is gebleven, is de end-user. In de toekomst zou de App store ook voor deze doelgroep uitgebreid kunnen worden zodat ze specifieke content kunnen downloaden, waardoor eMagiz voor een nog grotere doelgroep aantrekkelijk wordt.

In de Business Process Layer zijn de processen weergegeven die door de actoren uitgevoerd moeten kunnen worden. Deze zijn onderverdeeld aan de hand van het gebruik ervan. Het proces manage moet een administrator in staat stellen om de App store te beheren. Hiervoor moet hij mensen toegang kunnen verschaffen tot de App store door de actoren te autoriseren. Hiervoor is altijd een licentie nodig die moet kunnen worden uitgegeven. Natuurlijk zit bij het gebruik van de iPaaS ook al een licentie en wordt de iPaaS ook al beheert, maar ze worden hier nog een keer apart genoemd omdat er binnen de App store gedacht kan worden aan verschillende sublicenties. Partners kunnen bovendien ook hun eigen store beheren.

Daarnaast moet het beheren van de content ook mogelijk gemaakt worden. Zo moet content soms worden aangepast, zodat deze in de algemene store van eMagiz geplaatst kan worden. Voor het daadwerkelijke gebruik van de App store zijn de processen onder het kopje 'usage' opgesteld.

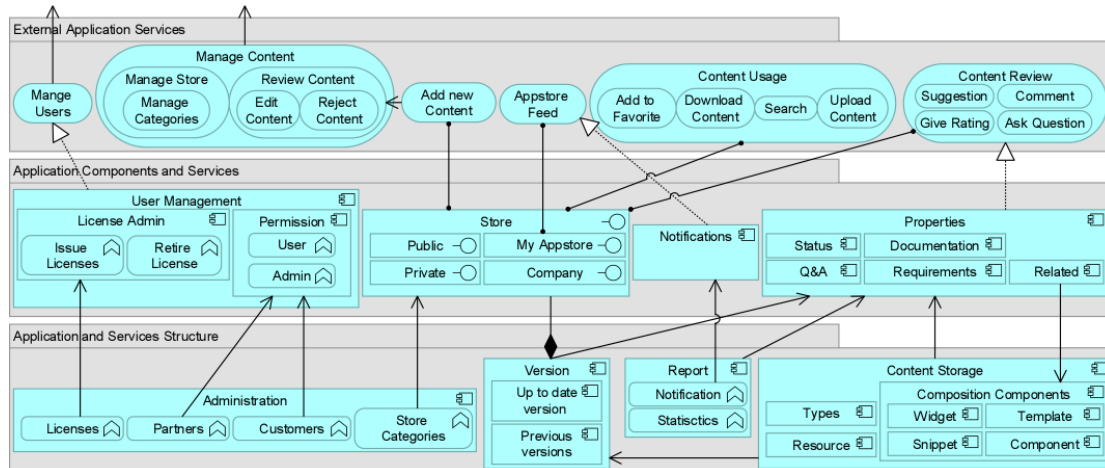
Deze functionaliteiten zijn gebaseerd op acties die uiteindelijk uitgevoerd moeten kunnen worden, waarbij de twee belangrijkste processen het uploaden en het downloaden van content zijn. Uiteraard is het ook belangrijk dat naar content gezocht kan worden.

De module Buy content is verder nog niet gespecificeerd, maar zou een aparte module kunnen worden, die vooral extern geregeld kan worden of aan de licentie gekoppeld kan worden maar dat valt buiten de scope van dit onderzoek.

3.3.2 Application Layer.

De Application Layer vertegenwoordigt de applicaties van de verschillende functies en vormt zo de koppeling tussen de eisen die in de business layer gesteld worden en de technische mogelijkheden die de technology layer biedt.

ArchiMate maakt onderscheid tussen services binnen een eigen layer en services met de bovenliggende layer. Dit laatste is goed te zien in de external application services layer, waar de applicatieservices (→) gebruikt worden door de componenten uit de bovenliggende business layer, waardoor de external business services gelijk zijn aan de business processies. Binnen deze layer bestaat ook een service binnen de eigen layer, zo loopt er een relatie van “Add new Content” naar “Review Content”. Het kan namelijk zo zijn dat als een gebruiker een bepaalde content wil toevoegen, deze content eerst nog moet worden aangepast zodat deze aan alle eisen voldoet.



Figuur 3-10: De Application Layer van de Reference Architecture uitgelicht.

De Application Components and Services layer laat de verschillende modules zien die de algemene functies van de App store vormen. De modules *ondersteunen* elk een deel van de in de vorige paragraaf beschreven functies, zoals manage en handling. Dit wordt gedaan door het realiseren van de External Application Services. De user management en properties componenten (→) zijn op zichzelf opererende modulen, die verder uit gespecificeerd kunnen worden naar meerdere kleine componenten. Binnen deze componenten zijn soms applicatiefuncties (→) nodig, die het intern gedrag van dit component beschrijven en uiteindelijk een of meerdere services kunnen realiseren. Zo kan door de functie “Issue License” binnen de componenten “License Admin” en “User Management” de service “Manage Users” worden ondersteund, die uiteindelijk het Business Process “Manage” realiseert.

Het applicatiecomponent “Properties” wordt gebruikt om de verschillende eigenschappen van een content te kunnen vastleggen. Deze eigenschappen zijn gerelateerd aan de verschillende versies, die in de volgende layer besproken zullen worden.

Daarnaast maakt de Store interface (→) de visuele weergave van de content mogelijk welke uiteindelijk in de Business service interface gebruikt worden. Hierin zijn de verschillende stores te zien en waar deze toegang tot hebben. Zo maakt “My App Store” de specifieke App store feed mogelijk.

In de onderste swimlane van de Applicatie layer wordt de structuur van de verschillende applicatie services weergegeven. Hierin is te zien volgens welke structuur componenten in het systeem worden opgeslagen. Zo beheert het component Administration de verschillende soorten Licenses, Partners, Customers en Store Categories die in de App store gebruikt zullen worden. Het version component maakt het opslaan van verschillende versies van een soort content mogelijk. Door een verschil van softwareversie is het namelijk soms belangrijk dat een oudere versie nog gevonden kan worden. Deze verschillende versies moeten goed in de betreffende store komen te staan en moeten soms afhankelijk van elkaar verschillende eigenschappen, zoals documentatie of eisen, hebben.

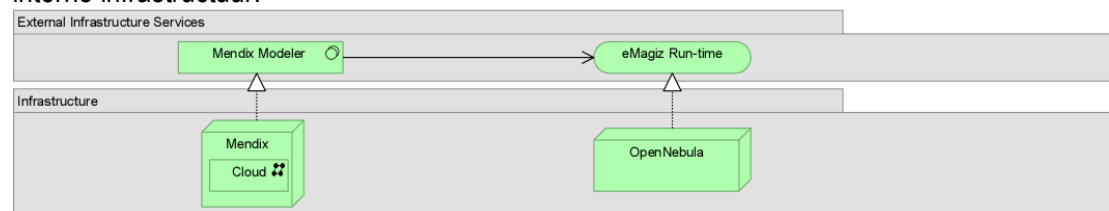
In het laatste component worden de verschillende componenten van de content zelf beheerd. Deze komen voort uit §3.1.4 en zijn, zoals in die paragraaf geconstateerd, nog niet volledig. Als

er in de toekomst een verandering in content-componenten ontstaat, kan hier eenvoudig een nieuw component worden toegevoegd of worden verwijderd.

Binnen de Application Service Structure zelf bestaat ook een interne relatie, deze loopt van Content storage naar de component versie. Een verzameling van deze Content storage is namelijk een soort content, die minimaal één versie heeft. Daarnaast loopt er een relatie van de bovenliggende swimlane terug naar de Content Storage. Vanuit de Application Components and Services swimlane loopt namelijk een relatie vanuit het component Related (binnen het component Properties) terug naar Content Storage. Deze relatie zorgt voor de koppeling van de compositie van de content en de documentatie hiervan.

3.3.3 Technology Layer.

In de onderste layer, wordt de infrastructuur van het systeem beschreven. Het toont de software, hardware en de netwerken die zorgen voor de infrastructuur die de App store ondersteunt. Net als bij de twee bovenliggende layers wordt hier een onderscheid gemaakt tussen externe en interne infrastructuur.



Figuur 3-11: De Technology Layer van de Reference Architecture uitgelicht.

Binnen dit onderzoek wordt hier echter niet diep op ingegaan, dit is ook niet nodig omdat we hier te maken hebben met een iPaaS. Met andere woorden; het interesseert ons niet echt hoe dit gebeurt en daarom kan dit door middel van 'the Cloud' worden uitbesteed. CAPE Groep is van plan om dit zoveel mogelijk via de Cloud van OpenNebula te laten lopen. Daarnaast wordt de App store zelf, net als de hele eMagiz iPaaS, met behulp van Mendix gemodelleerd.

3.3.4 Algemene toepassing Reference Architecture.

Aan de hand van bovenstaande Reference Architecture kan eMagiz de basis van een App store aflezen. Hierin valt te zien op welke manier de technologie en applicaties samen de business layer kunnen ondersteunen.

Voor eMagiz is dit voldoende, maar omdat dit onderzoek vanuit de Universiteit Twente gestart is, dient het ook een academische oplossing te hebben. Het moet dus blijken of deze Reference Architecture ook voor andere App stores van toepassing is. Hiervoor moet men zich afvragen of deze Architecture specifiek voor eMagiz is, of dat deze door een paar naamswijzigingen kan gelden voor een universele App store.

Het is belangrijk om hier een onderscheid aan te brengen tussen 'een universele App store' en een App store voor een iPaaS. Omdat bij het ontwerp van de Reference Architecture van de eMagiz App store gebruik is gemaakt van het feit dat deze geïntegreerd zal worden in de bestaande iPaaS, konden een hoop functies worden weggelaten, omdat deze reeds binnen de Reference Architecture van de iPaaS zelf zijn opgenomen. Als men vervolgens naar de Architecture van de eMagiz App store kijkt ziet men dat uitsluitend in de Technology layer de namen eMagiz en Mendix terug te vinden zijn. Zoals in de vorige paragraaf beschreven, is de technology layer de minst belangrijke laag in dit onderzoek. Omdat we hier te maken hebben met een iPaaS zou deze zelfs weggelaten kunnen worden, maar voor de volledigheid is deze toch kort vermeld. Omdat het dus niet uit maakt hoe deze layer geregeld is, kunnen er ook snel naamswijzigingen in deze layer worden aangebracht. Dit maakt deze Reference Architecture van algemeen nut voor welke iPaaS dan ook.

Als men vanuit dit onderzoek een Reference Architecture wil maken voor een algemene App store, moet men deze toch wat uitgebreider opzetten. Als men echter de Reference Architecture van de iPaaS zelf ook heeft en deze twee architecturen gaat combineren, kan dit een goede Reference Architecture voor een algemene App store opleveren.

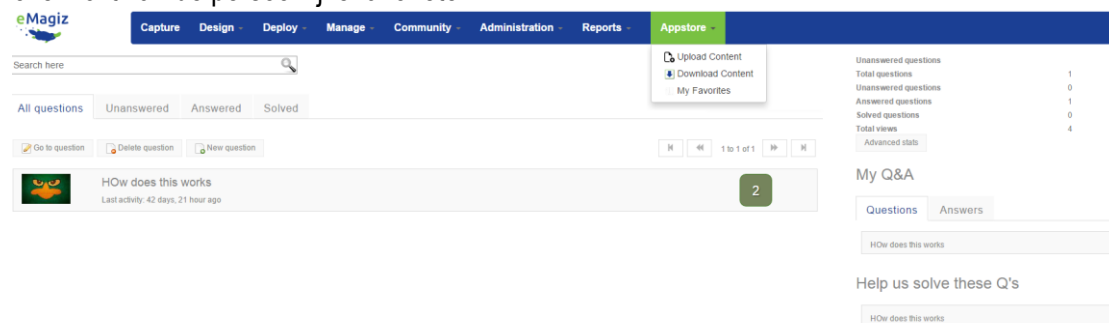
4 DEMONSTRATIE.

Peffers stelt dat demonstratie tot een oplossing van het oorspronkelijke probleem, waarvoor het onderzoek is opgesteld, moet leiden. In dit geval is het probleem dat er nog geen App store is. Door een goede uitleg van het prototype zullen de kansen groter zijn dat het ook daadwerkelijk, gebruikt gaat worden (Boehm, 1988).

In dit hoofdstuk zal de uitleg gegeven worden aan de hand van een aantal screenshots van de App store. De App store is in een aparte branche line ontwikkeld, die de mogelijkheid bood om binnen het bestaande project van het Mendix 5 model te werken, zonder dat hierbij de mogelijkheid bestond dat er problemen met de huidige iPaaS zouden ontstaan. Doordat de App store dus nog niet live gebruikt werd, bestond er ook nog geen live data, waardoor alle gegevens die in de screenshots te zien zijn bestaan uit dummy data.

4.1 GENERAL OVERVIEW

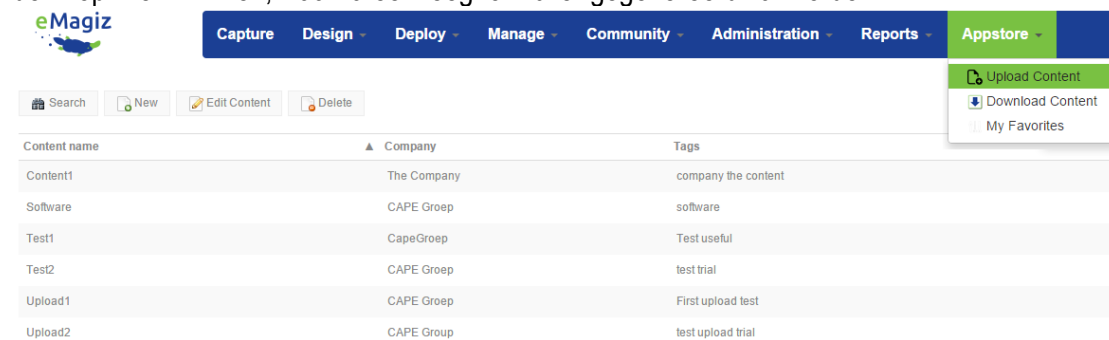
Na het inloggen via de homepage van eMagiz, my.emagiz.com, komt men terecht bij het volgende startscherm (fFiguur 4-1). Hierop ziet men de interactie met de bestaande community en kunnen onder andere Q&A's behandeld worden. Bovenaan de pagina zijn de verschillende tabbladen te zien, met rechtsboven het tabblad voor de App store. Door hierop te klikken verschijnen drie subitems voor het uploaden en downloaden van de content met daarnaast een overzicht van de persoonlijke favorieten.



Figuur 4-1: Screenshot van het startscherm van eMagiz.

4.2 Upload Content.

Het bovenste subitem van het tabblad App store is het uploaden van content. Door erop te klikken komt men tot een overzicht van reeds bestaande content, zoals te zien in Figuur 4-2. Als men een nieuwe versie van aan bepaalde content wil toevoegen kan de desbetreffende content hier worden geselecteerd en door op 'Edit Content' te klikken, bewerken. Natuurlijk kan men ook op de knop 'New' klikken, waarna een leeg formulier gegenereerd zal worden.



Figuur 4-2: Screenshot van Upload content overzicht binnen het tab App store.

Om een duidelijker beeld te kunnen schetsen is ervoor gekozen om in Figuur 4-3 alleen het edit formulier te laten zien, omdat hierin meteen de dummydata kan worden meegenomen.

Figuur 4-3: Screenshot van de Edit content pagina binnen eMagiz.

Helemaal bovenin staat de naam van de content met daaronder de klant voor wie dit product ontwikkeld is. De klant kan gekozen worden uit reeds bestaande klanten die het systeem kent. Hierdoor worden duplicaties voorkomen en wordt de content meteen aan de juiste klant gelinkt. Onder de klant is een knop voor het daadwerkelijk uploaden van de content te zien. Dit opent een apart formulier dat gelijk is aan het tabblad Versions, wat verderop in het verslag besproken zal worden. Hierin kan de content worden geüpload, één eMagiz- en content-versienummer worden meegegeven en één datum worden gekozen.

Daaronder zijn verschillende tabbladen aangebracht. Zoals te zien is in Figuur 4-3 kunnen hier de naam van de ontwikkelaar, het bedrijf waarvoor hij werkt, één korte beschrijving, Tags, één link en één logo worden toegevoegd.

In het tabblad Category kan door middel van checkboxen een keuze gemaakt worden uit de verschillende categorieën die in §3.1.4.1 zijn besproken. Dit kunnen ook meerdere categorieën zijn.

In het tabblad Store kan een keuze gemaakt worden uit de verschillende Stores waaruit de App store bestaat. Ook hier wordt gebruik gemaakt van checkboxen. Als een gebruiker niet voldoende rechten heeft om content direct tot de algemene store van eMagiz toe te voegen, zal in plaats van een checkbox een knop te zien zijn waarmee een verzoek tot goedkeuring verzonden wordt, te zien in Figuur 4-4. In het volgende tabblad, Approval, kan een eMagiz werknemer ervoor zorgen dat de content in de algemene store gevonden kan worden.

Figuur 4-4: Screenshot van tabblad Store in Edit Content

In het tabblad Pricing kan een keuze worden gemaakt of een content gratis is, of dat hiervoor betaald moet worden. Als er gekozen wordt voor een betaalde content verschijnt er de mogelijkheid om een trialperiode toe te voegen zoals te zien in Figuur 4-5 en Figuur 4-6.

Figuur 4-5: Screenshot van het tabblad Pricing is free.

Figuur 4-6: Screenshot van betaalde content.

Het volgende tabblad is Version. Hierin kan een nieuwe versie worden toegevoegd aan de reeds bestaande content, zonder dat de oude content hoeft te worden vervangen. Dit kan bijvoorbeeld het geval zijn bij een nieuwe eMagiz versie of als een bug is verholpen.

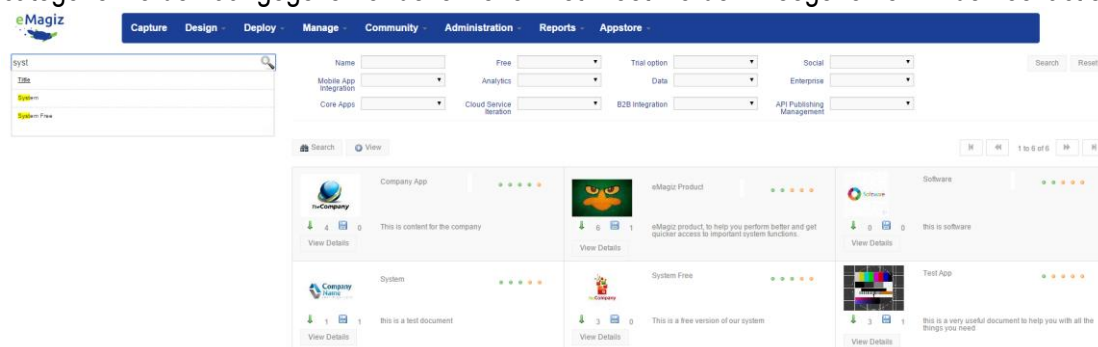
In de tabbladen Requirements en Documentation kan extra informatie worden ingevuld over bijvoorbeeld de implementatie van de content of extra aanwijzingen gaande het gebruik van de content.

Ten slotte kan een Resource worden meegegeven, deze kan nodig zijn voor het daadwerkelijke gebruik van content.

Als alles is ingevuld kan de content worden opgeslagen en is deze, afhankelijk van de rechten, beschikbaar voor de gebruikers.

4.3 DOWNLOAD CONTENT.

Via het tweede subitem van het tabblad Appstore komt men terecht in het overzichtsscherf van de bestaande content, te zien in Figuur 4-7. De verschillende content wordt in een Grid weergegeven met minimale informatie. Zo is naast de naam, de rating en korte beschrijving ook het aantal downloads te zien samen met het aantal als favoriet gemarkeerde content. Als de gebruiker vervolgens meer informatie over de desbetreffende content wil, kan deze via de knop View Details het uitgebreide informatiescherf van Figuur 4-8 openen. Als de gebruiker niet meteen vindt wat hij zoekt kan hij op twee manieren gaan zoeken. Linksboven is een zoekveld te zien, waarin bijvoorbeeld gezocht kan worden op naam of op tags. Daarnaast kan men via de Search knop de, in dit figuur reeds uitgeklapte, search vakken openen. Hier kan bijvoorbeeld per categorie worden aangegeven of deze wel of niet moet worden meegenomen in de zoekactie.



Figuur 4-7: Screenshot van het overzichtsscherf van bestaande content

In Figuur 4-8 is de gedetailleerde weergave van de contentgegevens te zien. Deze weergave is gebaseerd op de mock-up uit §3.2.3. Onder het tabblad General is veruit de meeste informatie duidelijk weergegeven. Aan de linkerkant zijn onder een afbeelding de aangegeven tags te zien met daaronder de bijbehorende beschrijving. In het midden zijn de meer algemene gegevens

van de content te zien. Na de naam van het product, de ontwikkelaar en het bedrijf is te zien onder welke categorieën deze content valt. Daaronder is informatie over de meest recente versie te zien. Helemaal onderaan de pagina is ruimte om een comment te plaatsen.

Aan de rechterkant is een visuele weergave van de rating te zien. De gebruiker kan door met zijn muis over de gekleurde bolletjes te bewegen, geïnspireerd door de eieren van het vogelbekdier, zijn rating aan de content toekennen. Daarachter is te zien uit hoeveel reviews de rating is opgebouwd. Vervolgens kan men zien hoe vaak deze content al is gedownload en kan men deze toevoegen aan zijn favorieten. De knop Add to favorites opent een nieuw scherm waarin een map geselecteerd kan worden waarin deze wordt opgeslagen. Onder de knop is te zien hoe vaak andere gebruikers al gebruik van de desbetreffende knop hebben gemaakt. Na informatie over de prijs en het wel of niet bestaan van een trial periode is de status van de content te zien en in weke store deze beschikbaar is. Helemaal onderaan zijn de verschillende comments te zien die over deze content gemaakt zijn.

App Information

The screenshot shows the 'General' tab for 'eMagiz Product'. It includes a product icon, tags, a description, developer information (Benno Lippok, eMagiz, Social), version details (1.20, previous 1.10, last update 2014-11-17), and a download button for \$0.99. On the right, there's a rating section with 2 reviews, a download count of 6, and a trial period of 3 days. At the bottom, there's a comment box and 'Save'/'Comment' buttons.

Figuur 4-8: Screenshot van de General page van de Content

Als de gebruiker de content wil gaan downloaden, kan hij gebruik maken van de downloadknop. Deze is te vinden in het midden van de pagina, net boven het tekstvak voor de comments. Als, zoals in het geval van Figuur 4-8, het een betaalde content betreft, zal voordat de download gestart wordt eerst een nieuw scherm openen, te zien in

Figuur

4-9. In het Choose Payment scherm is nogmaals de prijs te zien en de eventuele trial periode. Hier kan gekozen worden om gebruik te maken van de trial periode of de content meteen aan te schaffen.

Figuur 4-9: Screenshot van het betalingsscherm bij een betaalde content.

Als een gebruiker behoefte heeft aan een andere versie van de content, kan deze in het tabblad All versions een overzicht vinden met al de voorgaande versies. Hier is de algemene informatie over de versies te vinden, samen met een korte release note, zoals te zien in Figuur 4-10.

The 'Choose Payment' screen displays the price (\$0.99), trial status (Yes), and trial period (3 days). It offers two options: 'Use Trial Period' (selected) and 'Buy' (with a Euro symbol). A 'Cancel' button is at the bottom left.

App Information

General

All versions

Requirements

Documentation

Search

Version number

1.20

bug fix better performance

Previous version

1.10

Last update date

2014-11-17

Initial date

2014-11-17

E magiz version

3.00

...

Upload

Download

Version number

1.10

This is the first version

Previous version

0.00

Last update date

2014-11-17

Initial date

2014-11-17

E magiz version

3.00

...

Upload

Download

Figuur 4-10: Screenshot van het versieoverzicht van een content.

In de twee overgebleven tabbladen zijn respectievelijk de Requirements en een Documentation van de content te vinden.

5 VALIDATIE.

De vijfde stap van het model van Peffers is de evaluatie, welke in een wetenschappelijke context gedaan kan worden door middel van een Validatie (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007, p. 56). Deze validatie dient ervoor om te onderzoeken hoe goed het artefact de oplossing van het probleem ondersteunt. Dit wordt onder andere gedaan door een vergelijking te maken tussen de doelstellingen en de werkelijk waargenomen resultaten.

Binnen het wetenschappelijk onderzoek worden over het algemeen drie verschillende manieren van validatie aangehouden. De eerste is het valideren van de functionaliteiten. Hierbij wordt het artefact met de opdrachtgever geëvalueerd. Een andere mogelijkheid is een kwaliteitscontrole, hierbij wordt het *framework* van het prototype getest door de doelgroep. Als laatste wordt de validatie ook wel eens gedaan door middel van een Illustratie. Hierbij wordt het prototype vergeleken met de van tevoren opgestelde eisen, waardoor gekeken kan worden of alles geïmplementeerd is.

Voor dit onderzoek is ervoor gekozen om het artefact aan de hand van de hierboven eerste en laatste manier te valideren. De waarde van een kwaliteitscontrole werd op dit moment van het onderzoek niet erg hoog geacht, omdat er nog niet voldoende content beschikbaar is, zodat er nog niet goed genoeg getest kan worden. De functionaliteit en illustratie zullen nu verder worden toegelicht, nadat er eerst nog wordt ingegaan op de terugkoppelingen die tijdens het onderzoek hebben plaatsgevonden.

5.1 WAARBORG TIJDENS ONDERZOEK

Om ervoor te zorgen dat het artefact goed aansluit op de wensen van CAPE Groep en haar partners, zijn er tijdens het onderzoek verschillende terugkoppelmomenten geweest. Deze aanpak komt voort uit de in §2.2.5 besproken Spiral methodology waar steeds de ontwikkelde stukken geëvalueerd worden waardoor een goede grip op het onderzoek wordt gehouden.

Deze aanpak had twee voordelen. Ten eerste verschaftte het steeds inwinnen van informatie een duidelijk beeld van de gewenste toepassingen. Vervolgens kon met deze medewerker meteen worden overlegd over de manier waarop dit het beste binnen het onderzoek gebruikt kon worden. Dit sluit ook goed aan op het motto van CAPE Groep, waar men er in een keer voor wil zorgen dat er een goed kwalitatief product wordt afgeleverd zodat men hier achteraf niet meer aan hoeft te sleutelen.

Deze aanpak begon met diverse gesprekken met verschillende medewerkers, over de vraag wat een App store precies zou moeten kunnen. Door deze uitkomsten vervolgens bij de andere medewerkers te toetsen, kon een duidelijk beeld van de verwachtingen worden geschetst.

Deze uitkomsten werden vervolgens nog een keer getest bij het opstellen van de user stories, door deze wederom aan verschillende medewerkers voor te leggen. Hierbij werd vooral gekeken naar de functie en achtergrond van de verschillende medewerkers.

Omdat niet iedereen even nauw bij het onderzoek betrokken kon worden, zijn er ook presentaties voor het hele bedrijf geweest. Tijdens deze zogenoemde Social Friday's kon de voortgang worden besproken en werd er gevraagd om input en opmerkingen. Een van de belangrijkste punten die naar voren kwam bij de Social Friday's was de zoekfunctie. Binnen Mendix 5 kon dit alleen op een omslachtige manier en het was maar zeer de vraag of deze gebruikt zou worden. Daarop kwam een medewerker van CAPE Groep met de vraag of er niet een zoekveld kon worden ingesteld, die ook bij andere CAPE Systemen aanwezig was. Na de nodige microflows is dit uiteindelijk gelukt. Het zoekveld past ook binnen de App store omdat hierin getracht is om zoveel mogelijk de huisstijl van de iPaaS over te nemen. Hierdoor moet het voor gebruikers makkelijker worden om met de App store om te gaan.

Omdat mijn werkplek dicht bij het koffiezetapparaat was, kwamen er dagelijks veel medewerkers langs lopen. Tijdens het wachten totdat de bonen van de koffie gemalen waren, ontstonden er vaak kleine gesprekken waarbij ik steeds mijn voortgang of ideeën kon bespreken, waardoor ook de kleinere vraagstukken uitgebreid aan bod konden komen.

5.2 FUNCTIONALITEIT

De validatie van de functionaliteiten zal onderverdeeld worden in twee gedeeltes, waarbij er met de heer Sebastian Piest zal worden ingegaan op het proces in het algemeen en met de heer Duc Bui Viet gekeken wordt naar de technische functionaliteiten binnen Mendix.

De heer Piest omschrijft de oorspronkelijke opdracht als een ontwikkeling van een prototype op basis van bestaande voorbeelden in de praktijk. Het doel was om erachter te komen wat de toegevoegde waarde van een App store zou zijn en hoe deze eruit zou kunnen komen te zien. Hierdoor was het nog niet bekend wat de App store zou moeten kunnen. CAPE Groep speelde met de gedachte om reeds bestaande softwarestukken universeler op te zetten zodat deze hergebruikt konden worden.

Gezamenlijk is er veel tijd besteed aan het onderzoek en het ontwerp en de uitkomsten zijn goed en volledig vertaald naar een Reference Architecture en een prototype. De Reference Architecture en het prototype omvatten alle aspecten en brengen deze op een logische manier naar voren en weerspiegelen daarmee de denkwijze van eMagiz.

De heer Piest stelt dat dit onderzoek vrij ver is gekomen in het zoeken en vinden van een oplossing en omschrijft het prototype daarmee als “volledig”. Zeker als er gekeken wordt naar de functionaliteiten van het prototype, zijn deze allemaal aanwezig en voldoet het ruim aan de verwachtingen.

Voordat de App store echt als product in gebruik genomen kan worden, denkt de heer Piest dat het nog wel verstandig is om naar de presentatie te kijken. De opmaak is uiteindelijk zeer belangrijk, maar binnen dit onderzoek is er voornamelijk aandacht besteed aan de functionaliteiten in plaats van de styling. Daarnaast zal er nog goed gekeken moeten worden hoe de App store te integreren valt binnen de community. De heer Piest heeft zelf in gedachte om meer gebruik te maken van de startpagina, maar dit wordt momentaan meegenomen in de ontwikkeling van eMagiz binnen Mendix 5.

De heer Piest denkt dat er in de toekomst veel vraag naar de App store zal komen en daarmee zeker tot een toegevoegde waarde van het product eMagiz kan uitgroeien en daarmee het hergebruik van content kan stimuleren. Hij kijkt hier vooral naar de voordelen voor de partners, daarmee heeft de App store ook als showcase een toegevoegde waarde. Hierdoor kan de App store in de toekomst een belangrijke plek innemen in de groei van de Community. Daarom zal de App store goed zichtbaar voor de gebruiker moeten zijn.

De App store is vooral een middel en daarom moet eMagiz ervoor zorgen dat er voldoende content beschikbaar is. Hoe de content technisch geïmplementeerd moet worden, is op dit moment nog een beetje onduidelijk. Om ervoor te zorgen dat de App store toch echt zijn toegevoegde waarde kan verzilveren, is eMagiz al bezig met vervolgonderzoek. In het kader van de EDI-Factoren zullen studenten van de Saxion Hogeschool te Enschede onderzoek doen naar de vraag welke content binnen de App store aangeboden dient te worden. Binnen het onderzoek wordt gekeken naar de verschillende technologiestrategieën van meerdere industrieën. Er zal getracht worden om een algemeen beeld te schetsen en punten te vinden die per industrie verschillen, industrieën die deel uit gaan maken van dit onderzoek zijn onder andere het transport, de zorg en de financiële sector.

Bij de technische ontwikkeling binnen Mendix ben ik ondersteund door de heer Duc Bui Viet, IT architect van CAPE Groep en eMagiz. Bij elke vraag probeerde hij om mij zelf tot het antwoord

te laten komen. Hij vroeg dan bijvoorbeeld wat het gewenste doel was en wat de mogelijkheden waren om dit doel te bereiken. Als ik mijn vraag goed overdacht had en ik wist het nog steeds niet, gaf hij steeds een tip waarnaar gekeken zou kunnen worden. Dit resulteerde uiteindelijk in een werkende App store, die zoals de heer Piest al aangaf, aan alle eisen voldoet.

Het uiteindelijke domain model omschrijft de heer Bui als “voor de eerste keer zeker een goed doordacht model”. Natuurlijk kan er nog wel veel verbeterd worden, maar dat is op zich een algemene opmerking bij de ontwikkeling van software, zo stelt hij, want alles kan altijd beter.

In de basis omvat het model alles wat het zou moeten hebben en er is daarnaast goed gebruik gemaakt van de reeds bestaande entiteiten die binnen de SaaS aanwezig waren, zoals de reeds gedefinieerde User. Door meer ervaring zal men in de toekomst steeds logischere microflows maken. In mijn geval waren de microflows soms wat omslachtig, vaak omdat ik niet alle functionaliteiten binnen Mendix kende en bepaalde informatie op een simpelere en snellere manier kon krijgen. De belangrijkste vraag is altijd hoe je aan je data komt en op welke manier je deze wil gebruiken en laten zien.

De heer Bui stelt dat alles doet wat het doen moet en is daarmee zeker tevreden met het prototype.

5.3 ILLUSTRATIE

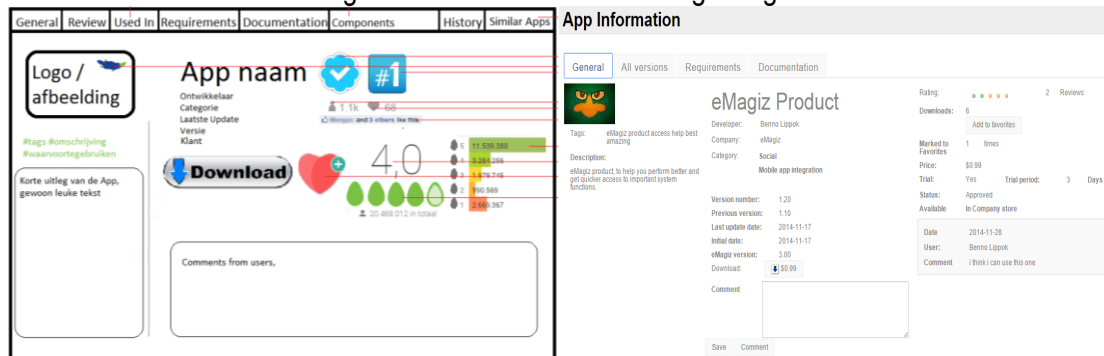
In §3.2.3 zijn aan de hand van een mock-up de eisen voor de content vastgelegd. Door deze mock-up te vergelijken met de contentweergave binnen Mendix kan worden gezien in hoeverre deze overeen komen. Om een beter beeld van deze vergelijking te kunnen geven zijn in Figuur 5-1 de mock-up en de contentweergave in Mendix weergegeven.

Bij een eerste oogopslag vertonen de twee afbeeldingen veel gelijkenissen. Zo is het tabblad general van de Mendixweergave zoveel mogelijk één op één overgenomen van de mock-up. Zo is linksboven in de hoek een afbeelding te zien met daaronder de verschillende tags en een korte uitleg van de content. De middelste strook begint in beide gevallen met de naam van de content, daaronder de naam van de ontwikkelaar, het bedrijf en de categorie waarbinnen deze content valt. In de mock-up staan tussen de hierboven genoemde aspecten ook de laatste update en de versie van de content. In de uiteindelijke mendix weergave is ervoor gekozen om deze van de vorige items te scheiden omdat vervolgens alle informatie betreffend de versie onder elkaar weergegeven kan worden. Uiteraard is in beide gevallen een downloadknop terug te vinden, net als de ruimte om comments te plaatsen.

Voor zover zijn de functionaliteiten allemaal overgenomen. Rechtsboven zien we echter twee punten die niet als zodanig terug te vinden zijn. Op dit moment was het nog niet nodig om binnen de contentweergave een indicatie te tonen die aangeeft of een content ‘trending’ is. Het icoontje dat moet aangeven of de content is goedgekeurd door eMagiz is niet in die hoedanigheid terug te vinden. Er is namelijk voor gekozen om de content een bepaalde status te geven die kan bestaan uit: Draft, Submitted, Approved, Published of Hidden. Dit gaf de mogelijkheid om meer informatie weer te geven en het nut van een visueel icoon werd ook niet als doorslaggevend geacht. Wat wel terug te vinden is, zijn de indicaties van het aantal downloads en het aantal dat ‘als favoriet’ gemaakt is. Dit kunnen namelijk belangrijke indicaties voor de toepasbaarheid van de content zijn.

Naast de al besproken uitgebreide weergave omtrent de versie en de status zijn er in de loop van het onderzoek ook andere extra functionaliteiten toegevoegd. Zo is het nu mogelijk om een trial periode aan het product toe te kennen, met bijbehorende periode. Daarnaast leek het ook verstandig om in de weergave te kunnen zien in welke store de content te vinden is. Helemaal rechtsonderaan zijn de comments te zien die door andere gebruikers geplaatst zijn. In de mock-up was hier nog een apart tabblad (Review) aan geweid, maar omdat deze comments dermate van belang voor de keuze voor een content kunnen zijn, is besloten deze toch op de general

page terug te laten komen. Ten slotte was in de mock-up de prijs van de content nog niet terug te vinden. In de Mendix weergave is deze uiteraard wel toegevoegd.



Figuur 5-1: Vergelijk van de Mock-up en Mendix weergave van de content.

In de Mock-up was er veel ruimte voor verschillende tabbladen, die in de Mendix weergave flink zijn beperkt. Uiteraard is de General pagina in beide gevallen terug te vinden. Zoals hierboven al besproken, is het review tabblad niet terug te vinden in de Mendix weergave omdat deze verplaatst is naar de General pagina. Zoals in §3.1.4.2 is besproken, was er in dit onderzoek onvoldoende tijd om dieper in te gaan op de opbouw van de verschillende designcategorieën. Daarom komen de tabbladen *Used in* en *Components* ook, nog niet, terug in de Mendix weergave. De tabbladen *requirements* en *documentation* zijn wel overgenomen maar behoeven geen verdere toelichting. Het in de mock-up genoemde *History* tabblad is in de uiteindelijke Mendix weergave omgedoopt tot *All versions*. Hierin is namelijk niet alleen informatie over de vorige versies te vinden maar de versies zelf ook.

Het laatste tabblad *Similar Apps* is ook niet terug te vinden in het prototype omdat hiervoor nog niet genoeg tijd was en het een lage prioriteit heeft. Deze zou later zeker toegevoegd kunnen worden.

De uiteindelijke weergave van de content is binnen mendix een stuk zakelijker geworden, wat beter aansluit bij de doelgroep. Zo is de visuele weergave van de rating beperkt tot groene en gele bolletjes. Wat ons bij het laatste punt brengt.

Het valt immers ten diepste te betreuren dat het binnen Mendix, nog niet is gelukt om de rating uit te drukken in vogelbekdiereieren. Uiteindelijk zouden deze redelijk eenvoudig kunnen worden toegevoegd omdat de ratingsmodule, die binnen Mendix gebruikt wordt, de mogelijkheid geeft om twee iconen te selecteren die voor de visuele weergave zorgen. Aangezien deze iconen nog zouden moeten worden ontworpen en vervolgens toegevoegd zouden moeten worden binnen Mendix leek het me niet nodig om de medewerkers van eMagiz hiermee lastig te vallen.

6 CONCLUSIE EN AANBEVELINGEN.

Het laatste hoofdstuk gaat volgens Peffers over de Communicatie. Het is belangrijk dat de omgeving weet heeft van het nieuwe artefact en het nut ervan. Voor het academische onderzoek geeft hij hier de ruimte tot het geven van een conclusie en aanbevelingen. Na de conclusie zal aan de hand van een samenvatting wat uitgebreider worden ingegaan op de deelvragen. Vervolgens zullen aan de hand van een discussie de beperkingen besproken worden, gevolgd door aanbevelingen voor CAPE Groep en eMagiz.

6.1 CONCLUSIE

Aan de hand van dit onderzoek wilde eMagiz zich oriënteren naar de mogelijkheden van een eigen App store. Deze zou het hergebruik van bestaande content moeten bevorderen. Om dit te onderzoeken is de volgende hoofdvraag opgesteld.

Hoe kan eMagiz een App store (communicatie platform) creëren die bijdraagt aan het hergebruik en beter vinden van de al voorradige applicaties en rekening houdt met het kennisniveau van de verschillende doelgroepen zodat een bloeiende community kan ontstaan?

Om het hergebruik van bestaande content te stimuleren is het in eerste instantie belangrijk om voldoende content beschikbaar te hebben. Hierin is een rol weggelegd voor zowel de ontwikkelaars van eMagiz, als voor de partners van eMagiz. eMagiz moet ervoor zorgen dat het voor de partners aantrekkelijk is om content aan de App store toe te voegen. In de eerste plaats kunnen de partners door de App store zelf beter gebruik maken van hun content. Maar daarnaast kan eMagiz de partners ook de mogelijkheid bieden om geld te gaan verdienen met hun content. Hierdoor wordt het voor partners aantrekkelijk om met eMagiz samen te werken en zullen de partnerships groeien. Naarmate de App store steeds meer content beheert zal deze content voor een steeds breder publiek aantrekkelijk worden, waardoor de community ook zal groeien.

Naarmate er steeds meer content beschikbaar komt, wordt het steeds belangrijker om deze goed te kunnen onderscheiden. Dit wordt gedaan aan de hand van twee soorten categorieën. Het blijkt namelijk beter te zijn om een onderscheid te maken op basis van differentiatie in plaats van op kennisniveau. Dit komt voornamelijk omdat de huidige doelgroepen veelal hetzelfde kennisniveau hebben. Er wordt nu vanuit gegaan dat de gebruiker zelf redelijk kan veronderstellen wat hij nodig heeft. Aan de hand van een de software categorie en een design categorie wordt de zoektocht voor de gebruiker versimpeld. Bij de software categorieën kan een keuze gemaakt worden uit een bepaald thema waarvoor de content bedoeld is. Dit kan een keuze zijn uit bijvoorbeeld Mobile app integration of social apps zijn. Afhankelijk van de toepassing kan de gebruiker een geschikt design selecteren. Naast de twee categorieën zijn er ook andere functionaliteiten die de gebruiker tot de juiste keuze moet laten komen. Aan de hand van een rating systeem kan de community een cijfer geven, al dan niet in combinatie met een comment. Daarnaast kan de gebruiker release notes lezen, oudere versies bekijken of zich laten inspireren door vergelijkbare toepassingen.

Al deze zaken kunnen samen als een module worden geïntegreerd in het huidige iPaaS van eMagiz. Hierdoor blijft alles op een platform waardoor het makkelijker te beheren is en de gebruiker optimaal kan profiteren van eMagiz als platform. Als laatste zal het voor de gebruiker ook makkelijker zijn om met de App store om te gaan, als deze dezelfde huisstijl heeft als de rest van de iPaaS.

6.2 SAMENVATTING & ONDERZOEKSVRAGEN.

Doordat er in het begin van het onderzoek nogal wat onduidelijkheid bestond over de vraag wat een App store überhaupt is, was het niet duidelijk welke kant dit onderzoek op zou gaan. Dit maakte het opstellen van de verschillende deelvragen nogal lastig. In eerste instantie leek het vooral van belang om het doel van de App store duidelijk te krijgen. De App store zou moeten bijdragen tot het ontwikkelen van de community. Hier werd dus tijdens het opstellen van de onderzoeksvragen veelvuldig rekening mee gehouden. Pas toen het onderzoek echt gestart was, bleek dat er om een goede App store te kunnen maken, eerst veel dieper op de App store zelf ingegaan diende te worden. Het belang van de community werd wat meer naar de achtergrond verschoven, maar zeker niet uit het oog verloren. Dit bracht het onderzoek tot twee belangrijke vragen. Voor wie is de App store en wat moet de App store kunnen.

De eerste deelvraag diende ervoor om erachter te komen voor wie de App store zou moeten zijn:

Welke onderverdeling kan er gemaakt worden tussen de huidige en potentiële klanten van eMagiz, gelet op het gebruik en de verschillende kennisniveaus binnen deze doelgroepen.

Nadat er een beeld was ontstaan van de bestaande App stores in het algemeen (§2.4.1) lag de nadruk vooral op de kennisniveaus van de verschillende doelgroepen. Hiervoor werden de verschillende klanten en concurrenten van eMagiz onderzocht. Door middel van het Magic Quadrant for Enterprise iPaaS van Gartner (§2.4.2) kon worden nagegaan welke concurrenten eMagiz als voorbeeld zou moeten zien. In §2.4.3 valt te lezen dat eMagiz zich vooral moest gaan richten op de App stores van MuleSoft, snapLogic en Mendix.

In eerste instantie was het de bedoeling om een App store te ontwerpen die heel breed binnen een bedrijf ingezet zou kunnen worden. Dit bleek echter iets te voorbarig, in eerste instantie moet de App store vooral het hergebruik van content mogelijk maken. Voor de kennisniveaus werd gekeken naar de onderverdeling van het model van Haan (§ 2.2.3), deze werden vervolgens in §3.1.1 en §3.1.2 verder toegespitst op de doelgroepen van eMagiz. Hieruit bleek dat de meeste klanten van eMagiz een kennisniveau hebben dat tussen dat van een Professional developer en een Citizen developer valt. De kennisniveaus lagen echter vooral op het niveau van een Business engineer, waardoor het moeilijk bleek om hier echt een onderverdeling tussen te maken. Tijdens het ontwerpen bleek het daarom veel handiger om het onderscheid niet op basis van de kennisniveaus te maken, maar te laten afhangen van de manier waarop de doelgroepen gebruik maken van de App store. In §3.1.1 is daarom het onderscheid gemaakt tussen de ontwikkelaars en de gebruikers, met andere woorden degene die content aan de App store toevoegt en de gebruiker die deze content weer wil downloaden.

Welke onderverdeling kan er tussen componenten gemaakt worden zodat deze goed te hergebruiken zijn en hoe kan de content beter vindbaar gemaakt worden?

Binnen eMagiz bestond er veel onduidelijkheid over wat er in de App store aangeboden zou moeten worden en de manier waarop dat moest gebeuren. Vandaar dat de onderverdeling tussen de componenten als nog niet-volledig kan worden beschouwd. Hier is bewust ruimte gelaten voor aanvullende categorieën, als deze in de praktijk nuttig blijken te zijn.

Er is besloten de categorieën te splitsen in twee verschillende categorieën. De eerste is de software categorie, waarbinnen een onderscheid gemaakt kan worden per doelgebied, waarvoor de content gebruikt dient te worden. Deze categorie kan ook als thema gezien worden en is terug te vinden in §3.1.3 en is geïnspireerd door de reeds bestaande App stores van de in deelvraag één als interessante aangemerkte concurrenten. Voorbeelden van deze Software categorieën zijn onder andere Mobile app integration en social apps.

De tweede categorie wordt de design categorie genoemd. Deze categorieën richten zich op de opbouw van de content. Zo kan bijvoorbeeld een hele template flow worden gedownload of slechts een enkel component. Hierbij kan gedacht worden aan een heel ordeningproces, respectievelijk een zoekfunctie. Vooral binnen de design categorieën kan nog wat onderzoek worden verricht. Hier was binnen dit onderzoek nog geen ruimte voor omdat dit vooral in de praktijk getest moet worden. Maar omdat er nog geen goede content beschikbaar was, was dit tot op heden nog niet mogelijk.

Het tweede voordeel van het aanbrengen van de twee verschillende soorten categorieën, is dat er beter gezocht kan worden. Er kan nu op twee manieren worden gesorteerd waardoor alle niet interessante soorten content buiten beschouwing blijven en er snel een goede keuze gemaakt kan worden. Om ervoor te zorgen dat er een goede content gekozen wordt is een rating systeem toegevoegd. Hierbij is het mogelijk om de content te waarderen op een schaal van een tot vijf. Daarnaast kan men door middel van het plaatsten van comments extra informatie achterlaten.

Welke functionaliteiten moeten App store hebben en welke designstappen moeten hiervoor doorlopen worden zodat deze goed geïntegreerd kan worden in het huidige gebruik van de community?

De App store moet uiteindelijk door de medewerkers van CAPE Groep en andere eMagiz partners gebruikt worden. Daarom is samen met de medewerkers van CAPE Groep begonnen aan het brainstormen over de wenselijke functionaliteiten. Vervolgens zijn deze omgezet naar User Stories. Om een beter beeld te krijgen van deze User Stories en om erachter te komen in hoeverre de verschillende Stories met elkaar overlappen is er een schematisch overzicht van alle User Stories gemaakt, die terug te vinden zijn in APPendix A - User Stories. en APPendix B – Uitgetekende User Stories. Daarnaast is er op basis van voorbeelden van de concurrenten en de eisen die uit de User Stories kwamen een mock-up gemaakt van een mogelijke weergave van de content (§3.2.2). Op basis hiervan zijn een aantal key features geselecteerd die de basis van het prototype moesten gaan vormen (§3.2.3).

Toen de key features gedefinieerd waren, konden deze worden vertaald naar een Reference Architecture (§3.3). Met behulp van ArchiMate is in kaart gebracht op welke manier een werkende App store vormgegeven zou kunnen worden. Hierbij is de meeste aandacht uitgegaan naar de application layer (§3.3.2) omdat deze het meeste te maken heeft met de functionaliteiten van de App store.

Het laatste deel van deze deelvraag ging over het integreren van de App store in het huidige gebruik van de community. Omdat de community al veel met de iPaaS van eMagiz werkt, leek het een logische stap om de App store binnen deze iPaaS onder te brengen. Vandaar dat bij het ontwerp ook rekening is gehouden met de stijl van de iPaaS. Hierdoor moet het voor de gebruiker intuïtiever zijn om de juiste content te vinden.

6.3 DISCUSSIE VAN BEPERKINGEN.

Natuurlijk is niet alles wat van tevoren bedacht is, gelukt. Er zal nu een korte bespreking volgen van de belangrijkste beperkingen en de vraag in hoeverre deze opgelost kunnen worden.

6.3.1 Wat is niet gelukt.

Om de App store dichterbij de community te betrekken was het de bedoeling om het mogelijk te maken om verschillende content te kunnen 'delen' of aan andere medewerkers aan te kunnen raden. Vervolgens zouden deze op de startpagina van eMagiz terug te vinden moeten zijn. De content zou een aparte plaats op de startpagina kunnen krijgen of tussen de Q&A geplaatst kunnen worden. Door tijdgebrek is het echter nog niet gelukt om dit voor elkaar te krijgen.

Een ander punt dat niet gelukt is, is de onderverdeling per kennisniveau. Het doel van het aanbrengen van dit onderscheid was de mogelijkheid om de App store aantrekkelijk te maken

voor een zo breed mogelijk publiek. Hoewel de huidige markt op dit moment nog voldoende perspectief biedt, zou er in de toekomst gekeken kunnen worden of er uiteindelijk toch een onderscheid per kennisniveau opgesteld kan worden.

6.3.2 Onderzoekstechnisch

Zoals bij de tweede deelvraag reeds ter sprake is gekomen, is de onderverdeling van de verschillende categorieën nog niet voldoende uitgewerkt. Dit kwam door een gebrek aan tijd en een lage prioriteit. Dit betekent echter niet dat het niet mogelijk is om hiernaar te kijken. In feite is CAPE Groep al bezig met vervolgonderzoek naar deze categorieën, waardoor het slechts een kwestie van tijd is voordat dat probleem is opgelost.

6.3.3 Implementatie technisch

Door wat technische belemmeringen is het tot op heden niet mogelijk om de verschillende categorieën door middel van checkboxes te selecteren. Op dit moment bestaat er voor elke categorie een dropdown met een boolean (Ja/Nee). Dit werkt natuurlijk wel maar is een beetje omslachtig en het is daardoor maar de vraag of hier gebruik van gemaakt zal worden.

6.3.4 Limitations /Wat moet er nog gebeuren.

Op dit moment is het laten betalen van content nog maar een idee. De App store zou echter een goede vorm van extra inkomsten kunnen worden. Hiervoor moet het wel duidelijk zijn of de gebruikers ook daadwerkelijk bereid zijn om te betalen voor de content. Nu ligt het in de lijn der verwachtingen als de content goed is en er daardoor veel tijd bespaard wordt. Hierdoor zou de content een toegevoegde waarde voor de klant zijn waardoor deze hier waarschijnlijk geld voor over heeft. Daarnaast is in dit verslag de mogelijkheid besproken dat partners ook content aan de App store toe zouden kunnen voegen, en hier al dan niet geld mee te verdienen. Om erachter te komen of de partners hier ook daadwerkelijk in geïnteresseerd zijn, zal dit nog nader onderzocht moeten worden. Als dit niet het geval blijkt te zijn, kan de App store nog altijd als een extra dienst in de markt gezet worden.

6.3.5 Nice to have.

In de toekomst zou men met behulp van de App store een nog bredere doelgroep kunnen bereiken. In zo'n geval zou er bijvoorbeeld gewerkt kunnen worden met een Widget. Een eindgebruiker zou zo bijvoorbeeld de mogelijkheid hebben om een bepaalde grafiek in een widget te laten zien, die specifiek voor die persoon belangrijk is. Dit zou bijvoorbeeld een voorraadniveau kunnen zijn of de gegevens van een server. Deze zouden dan persoonlijk ingesteld moeten kunnen worden.

6.4 AANBEVELINGEN.

Dit onderzoek is een oriëntatie naar de mogelijkheden die een App store voor eMagiz zou kunnen hebben. Hieruit is naar voren gekomen dat een App store zeker van meerwaarde kan zijn voor het gebruik en de groei van het product eMagiz. Als het binnen de financiële mogelijkheden van eMagiz valt, is het ten eerste aan te raden dat eMagiz door gaat met de ontwikkeling van de App store. Daarbij moet de aandacht niet alleen aan de verbeterpunten van dit onderzoek worden besteed, maar ook naar de minder belichte kanten. Zo zou de App store niet uitsluitend content ter beschikking kunnen stellen die voor de ontwikkeling van nieuwe content bedoeld is, maar ook voor aanvullende mogelijkheden.

Hierdoor zou eMagiz voor veel meer doelgroepen interessant kunnen worden. Daarom lijkt het verstandig om met een marketing/sales deskundige te kijken op welke manier eMagiz haar App store in de markt moet zetten.

Voordat de App store echter aan partners aangeboden kan worden dient de App store volledig en goed getest te zijn. De eerste stap die hierin genomen dient te worden is het onderzoeken van de verschillende categorieën, hiervoor is reeds vervolgonderzoek ingesteld. Aan de hand van deze categorieën dient er nuttige content aan de App store toegevoegd te worden. Vervolgens zou men de App store een tijdje binnen CAPE Groep, in een beta versie, kunnen testen.

Als de App store goed genoeg werkt kan er contact op worden genomen met de partners. Ten eerste dient dit om te informeren, maar eMagiz kan er op deze manier ook achterkomen waar de partners behoefte aan hebben en hierop inspelen. Het zou zelfs zo kunnen zijn dat een klant een opdracht aan de community geeft voor het ontwikkelen van een bepaald component. Zo zou de App store als een soort marktplaats gezien kunnen worden waar vraag en aanbod samen kunnen komen. Dit past binnen het financiële perspectief, wat binnen dit onderzoek nauwelijks aan bod is gekomen. De App store kan namelijk een extra service zijn maar daarnaast ook een manier om geld te verdienen. Dit laatste kan voor de partners aantrekkelijk zijn, want waarom zouden ze immers hun content gratis aan de concurrent willen aanbieden. Hier is tijdens het ontwerp al wel rekening mee gehouden bij de download en trial functie. Men zou een aantal downloads bij de licentie kunnen verkopen of met een eigen betaalsysteem gaan werken. Het is natuurlijk ook goed mogelijk om met een soort iDeal betaalmogelijkheid te komen maar hiervoor moeten zeker nog wat veiligheidseisen worden onderzocht.

Volgens Peffers is de communicatie van een product de belangrijkste stap naar het succesvol in gebruik nemen van nieuwe software. De gebruikers moeten dus ten eerste weten dát er een App store er is en vervolgens weten hoe ze deze moeten gebruiken. Uiteraard zou dit ten eerste genoemd moeten worden in de release notes. Maar omdat lang niet iedereen deze leest zullen er ook andere methodes ontwikkeld moeten worden waarop de omgang met de App store wordt uitgelegd. Dit kan gedaan worden aan de hand van een "how-to" waarin stap voor stap de verschillende functionaliteiten worden besproken. Daarnaast kan er ook een demonstratie gegeven worden. Dit zou live kunnen maar men zou een demonstratie ook op kunnen nemen en bijvoorbeeld via youtube kunnen delen met de partners.

APPENDIX.

APPENDIX A - USER STORIES.

Deze stories beginnen met de administrator van eMagiz, deze heeft de meeste stories, waarvan de meeste ook voor anderen gelden, die zijn dan onder dat kopje gezet maar gelden ook voor de user de users die erboven al genoemd zijn.

As a <type of user>, I want <some goal> so that <some reason>.

Automatisch

Generen van notificaties

- Bij nieuwe content, zodat je daarvan op de hoogte bent
- Bij een Comments zoals:
 - Vraag, om deze te beantwoorden.
 - Antwoord, om het antwoord te bekijken.
- Rating, zodat je weet wat anderen van je content vinden
- Als er een Update is van een van je favorieten, zodat je de update kan gaan gebruiken
- Aanbevelingen op basis van gebruik, zodat je deze aanbeveling kunt gaan gebruiken

Admin

eMagiz SuperAdmin

de eMagiz SuperAdmin heeft alle rechten en mogelijkheden die alle andere gebruikers ook hebben afgezien van toegang tot de secret store. Daarnaast heeft hij ook nog de volgende mogelijkheden:

- Toevoegen van
 - Nieuwe partners en bijbehorende admins zodat deze toegang hebben tot eMagiz en ermee kunnen werken.
 - Nieuwe categorieën, zodat er een betere verdeling kan ontstaan als dit nodig is.
- Uitgeven van een licenties zodat anderen met eMagiz kunnen werken.
 - Aangeven voor welke periode de licentie gebruikt kan worden, zodat deze alleen gebruikt wordt zolang er betaald wordt.

Partner admin

- User beheer
 - Nieuwe user toevoegen, zodat deze toegang hebben tot eMagiz.
 - Rechten geven aan gebruikers, zodat ze hun werk kunnen uitvoeren.
 - Rechten intrekken als deze niet meer nodig zijn
- Content beheer
 - Verkoop van content aan andere eMagiz partners zodat er geld verdient kan worden.
- Company Store beheren zodat deze op orde blijft.
- Aanschaffen van
 - Nieuwe licenties als deze noodzakelijk is.
 - Betaalde content als deze noodzakelijk is.
 - Trial licenties, als men iets wil testen.
- Bedrijfsinformatie toevoegen zodat het bedrijf zich kan profileren.

ontwikkelaar

Partner ontwikkelaar

- Applicaties kunnen uploaden naar company en secret store zodat deze beschikbaar zijn voor andere gebruikers.
- Documenten aan de community toe kunnen voegen
- Aan admin vragen om een aanvraag voor de algemene store van eMagiz
- Updates uitvoeren voor
 - Eigen content
 - Company content
- Aanbevelen
 - Nieuwe content
 - Applicaties die vaak gebruikt worden
- Documentatie aan de community toe willen voegen, zodat deze uitgebreider wordt
- Toegang hebben tot de Q&A zodat ik deze kan uitbreiden

eMagiz ontwikkelaar

De ontwikkelaar van eMagiz heeft dezelfde mogelijkheden als een ontwikkelaar van een partner en daarnaast ook nog:

- content voor de algemene store
 - bekijken
 - controleren of de content aan alle eisen voldoet en eMagiz deze content ondersteunen kan.
 - Condoleren of de content met andere applicaties te combineren is.
 - aanpassen zodat de content aan alle eisen voldoet en in de store geplaatst kan worden.
 - goedkeuren en toevoegen aan algemene store
 - afkeuren en feedback naar ontwikkelaar verzenden.
- content van andere user updaten
- koppelen van content
 - die op elkaar lijken
 - die samen gebruikt kunnen worden
 - naar een nieuwere versie / content

Eindgebruiker

- Kunnen zoeken zodat ik iets kan vinden op basis van <...> zodat je iets kunt vinden
 - Omschrijving, tags
 - Een categorie
 - In associaties van reeds gebruikte content
 - In favorieten
 - Content die vaak gebruik wordt
 - Gratis of betaalde content (of trial)
 - Datum van verschijnen
 - Datum van laatste update
 - Klant
 - ontwikkelaar
- Applicaties kunnen downloaden vanuit elke toegankelijke store zodat men deze kan gebruiken.
- Applicaties kunnen toevoegen aan mijn favorieten zodat ik deze vaker kan gebruiken.
- Vragen kunnen stellen over...
- Een applicaties een rating kunnen geven

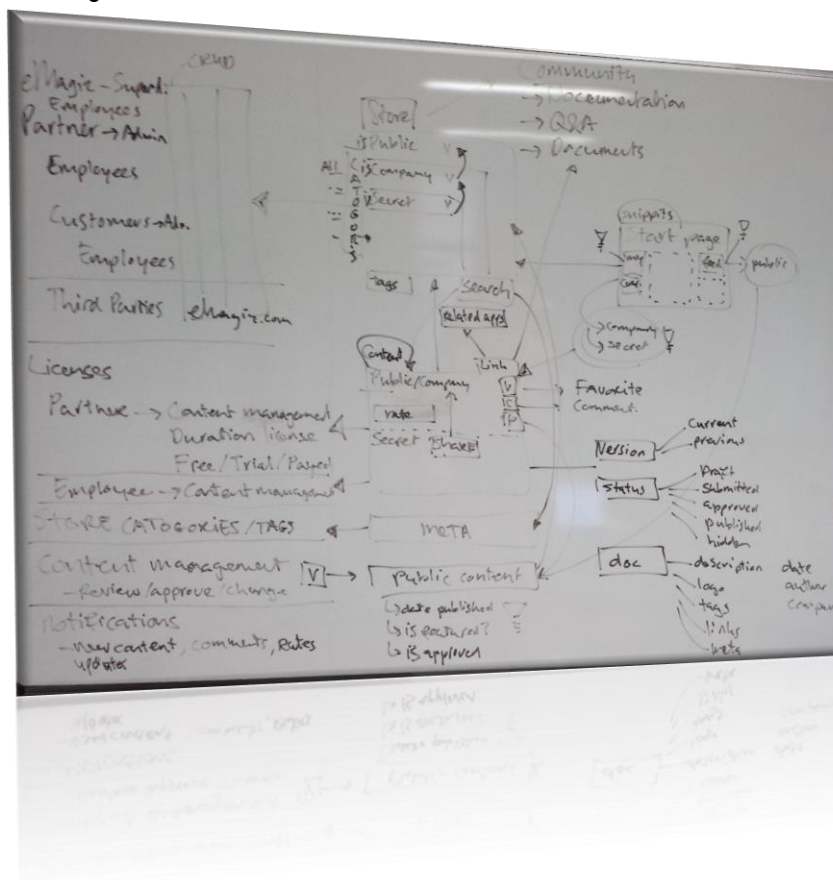
- Een comment geven over een applicaties zodat je feedback kunt geven.
- Een associatie toevoegen zodat andere gebruikers deze kunnen gebruiken.
- Contact opnemen met een ontwikkelaar van de partner of eMagiz
- Suggesties doen aan eMagiz:
 - Voor nieuwe categorieën voor een beter overzicht
 - Voor nieuwe thema's voor een beter overzicht
 - Ontwikkelingsgebied
- De status van een applicatie bekijken, dit kan zijn
 - Draft
 - Submitted
 - Approved
 - Published
 - Hidden
- Voorgaande versies bekijken

Third party

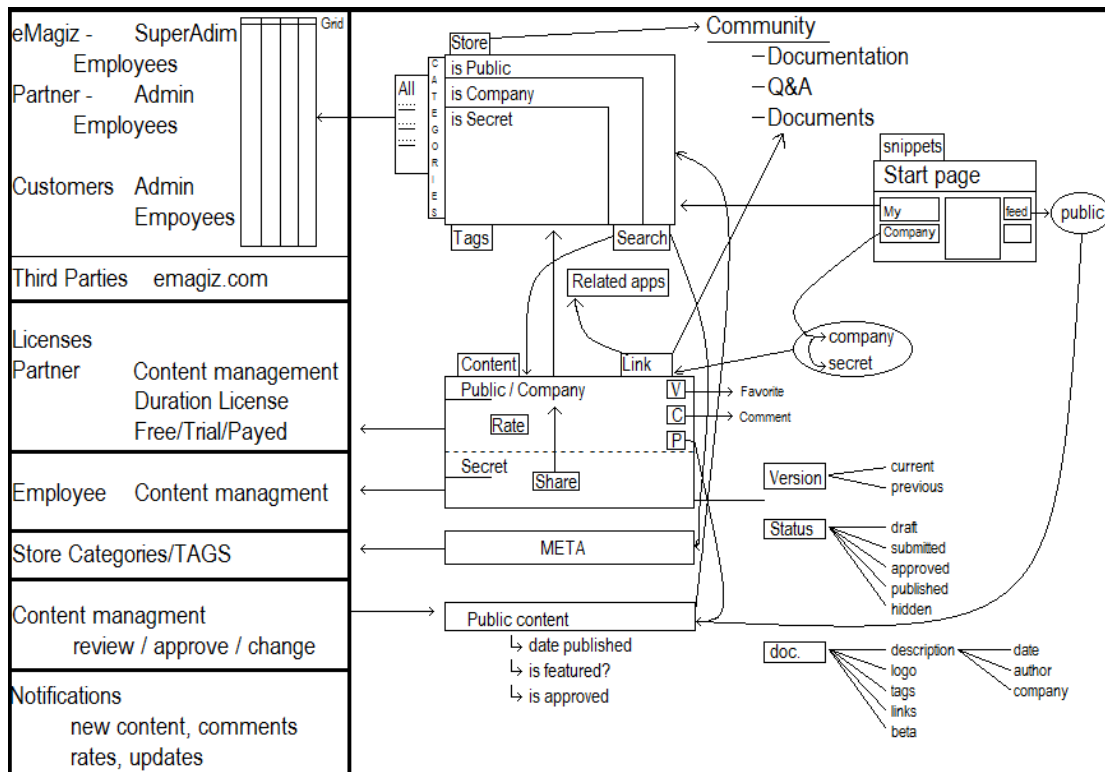
Zelf een account aanmaken zodat je toegang hebt tot eMagiz

APPENDIX B – UITGETEKENDE USER STORIES.

Een uitgetekende weergave van de user stories die een beter beeld van deze stories moet geven. Dit gaf vooral wat meer duidelijkheid in de functionaliteiten en de koppelingen hiervan die binnen een App store gewenst zijn en aan welke doelgroepen deze gekoppeld dienen te worden. In Figuur 0-1 is de daadwerkelijke uitwerking te zien, die voor de leesbaarheid nog een keer is uitgewerkt in Figuur 0-2.



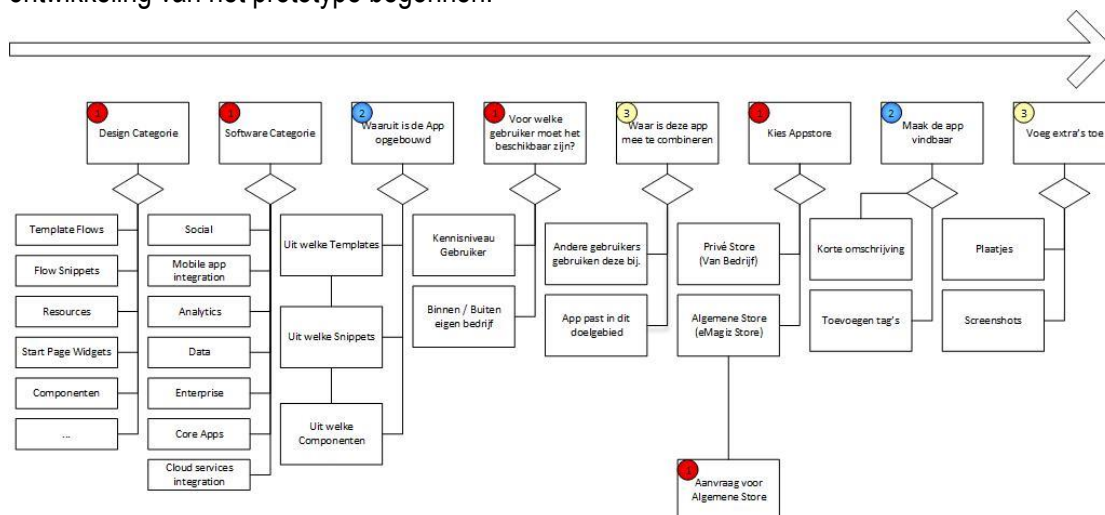
Figuur 0-1: Een foto van de uitgetekende user stories



Figuur 0-2: Een schematische weergave van figuur 0-1

APPENDIX C – UPLOADEN.

Voor het uploadproces is een schematische weergave gemaakt, te zien in Figuur 0-3. Hierin staan alle mogelijke aspecten die bij het uploaden van belang zijn. Daarnaast is een prioriteit aangegeven, van 1 als hoogste prioriteit tot 3 als laagste prioriteit. Op basis hiervan is de ontwikkeling van het prototype begonnen.

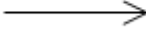


Figuur 0-3: Schematische weergave van het uploadproces met prioriteiten

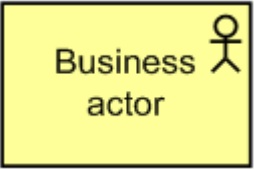
APPENDIX D – ARCHIMATE NOTATIONS

Binnen ArchiMate worden veel verschillende notaties gebruikt. De, voor dit onderzoek meest belangrijke notaties, zijn hier door de ontwikkelaars van ArchiMate (The open Group, december 2013) kort samengevat met de bijbehorende notatie.

Summary of Relationships

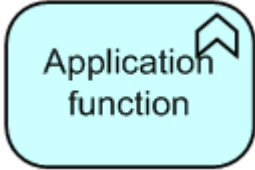
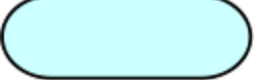
Structural Relationships		Notation
Used by	The used by relationship models the use of services by processes, functions, or interactions and the access to interfaces by roles, components, or collaborations.	
Realization	The realization relationship links a logical entity with a more concrete entity that realizes it.	
Assignment	The assignment relationship links units of behavior with active elements (e.g., roles, components) that perform them, or roles with actors that fulfill them.	
Composition	The composition relationship indicates that an object is composed of one or more other objects.	
Dynamic Relationships		Notation
Flow	The flow relationship describes the exchange or transfer of, for example, information or value between processes, function, interactions, and events.	
Triggering	The triggering relationship describes the temporal or causal relationships between processes, functions, interactions, and events.	

Summary of Business Layer Concepts

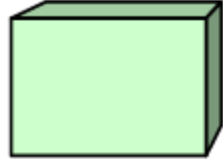
Concept	Description	Notation
Business actor	An organizational entity that is capable of performing behavior.	
Business role	The responsibility for performing specific behavior, to which an actor can be assigned.	
Business interface	A point of access where a business service is made available to the environment.	

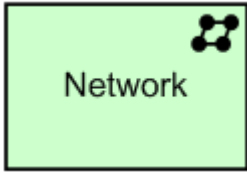
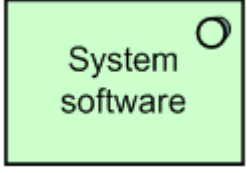
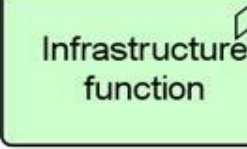
Concept	Description	Notation
Business process	A behavior element that groups behavior based on an ordering of activities. It is intended to produce a defined set of products or business services.	
Business function	A behavior element that groups behavior based on a chosen set of criteria (typically required business resources and/or competences).	
Business service	A service that fulfills a business need for a customer (internal or external to the organization).	

Summary of Application Layer Components

Concept	Definition	Notation
Application component	A modular, deployable, and replaceable part of a software system that encapsulates its behavior and data and exposes these through a set of interfaces.	
Application interface	A point of access where an application service is made available to a user or another application component.	
Application function	A behavior element that groups automated behavior that can be performed by an application component.	
Application service	A service that exposes automated behavior.	

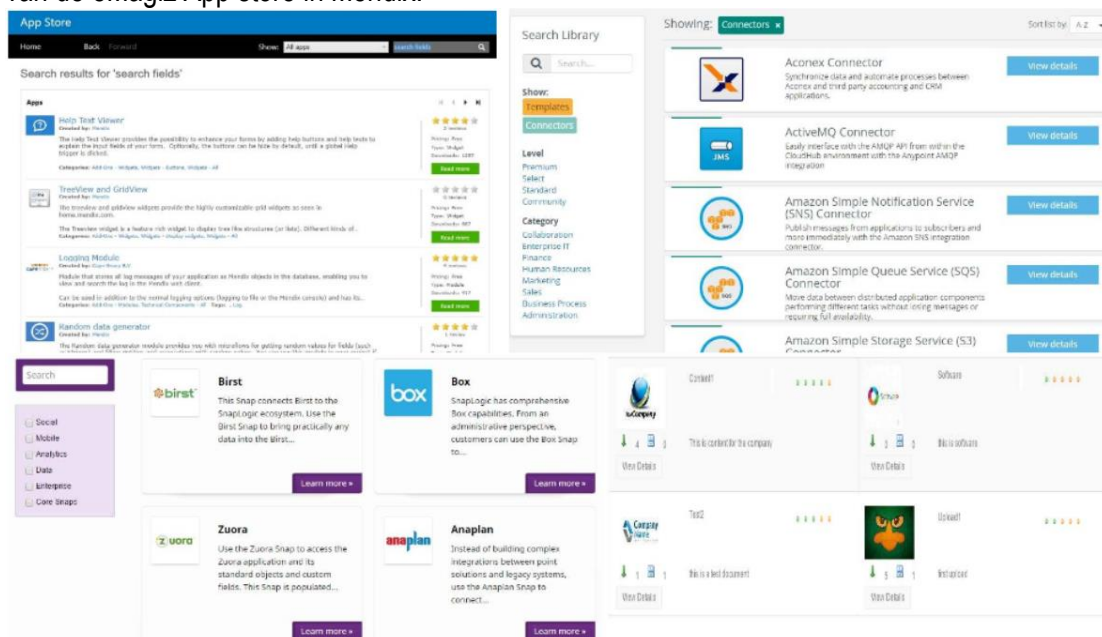
Summary of Technology Layer Concepts

Concept	Definition	Notation
Node	A computational resource upon which artifacts may be stored or deployed for execution.	

Concept	Definition	Notation
Network	A communication medium between two or more devices.	
System software	A software environment for specific types of components and objects that are deployed on it in the form of artifacts.	
Infrastructure function	A behavior element that groups infrastructural behavior that can be performed by a node.	

APPENDIX E - INSPIRATIE VAN ANDERE APP STORES IN HET ALGEMEEN EN UIT DE MARKT.

In de onderstaande Afbeelding vindt u vanaf linksboven naar rechtsonder, een impressie van de App stores vanaf achtereenvolgens Mendix, MuleSoft, Snaplogic en de uiteindelijke weergave van de eMagiz App store in Mendix.

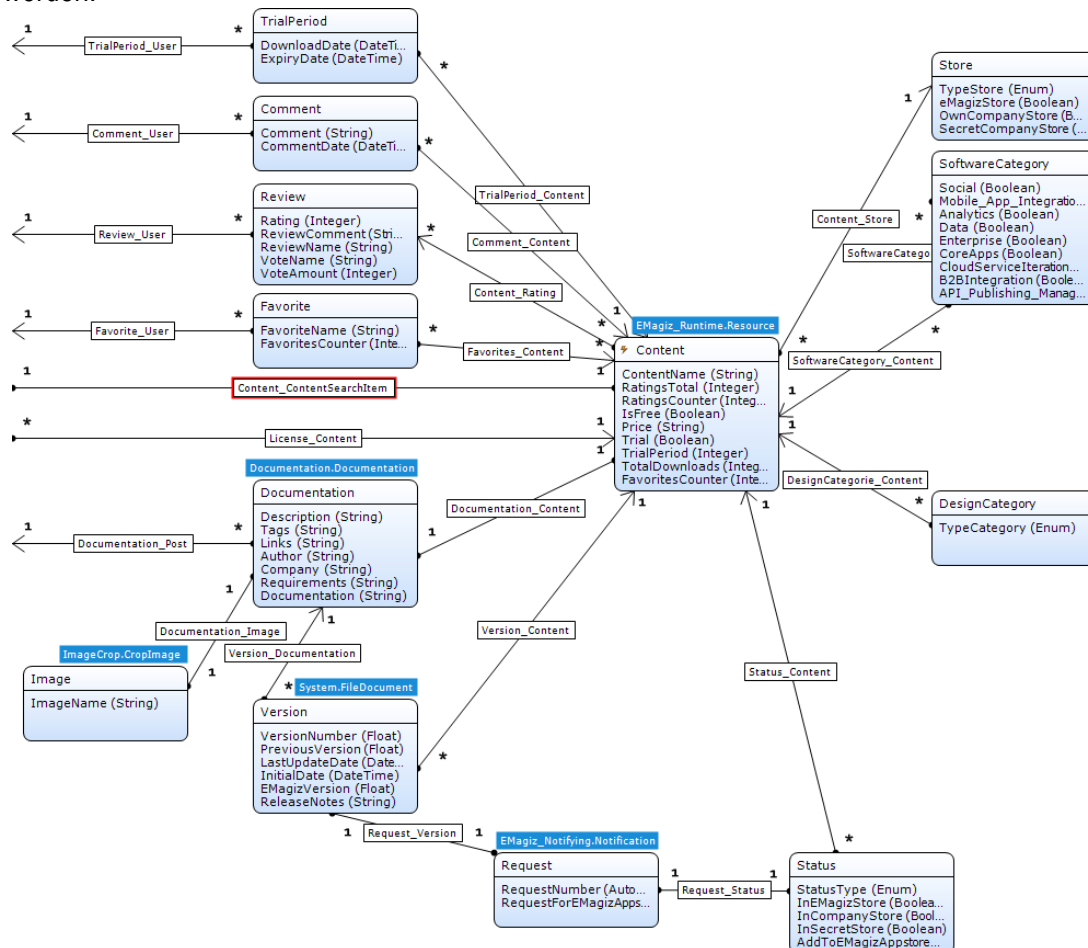


Figuur 0-4: Impressie van de App stores die als inspiratie zijn gebruikt en de uiteindelijke App store

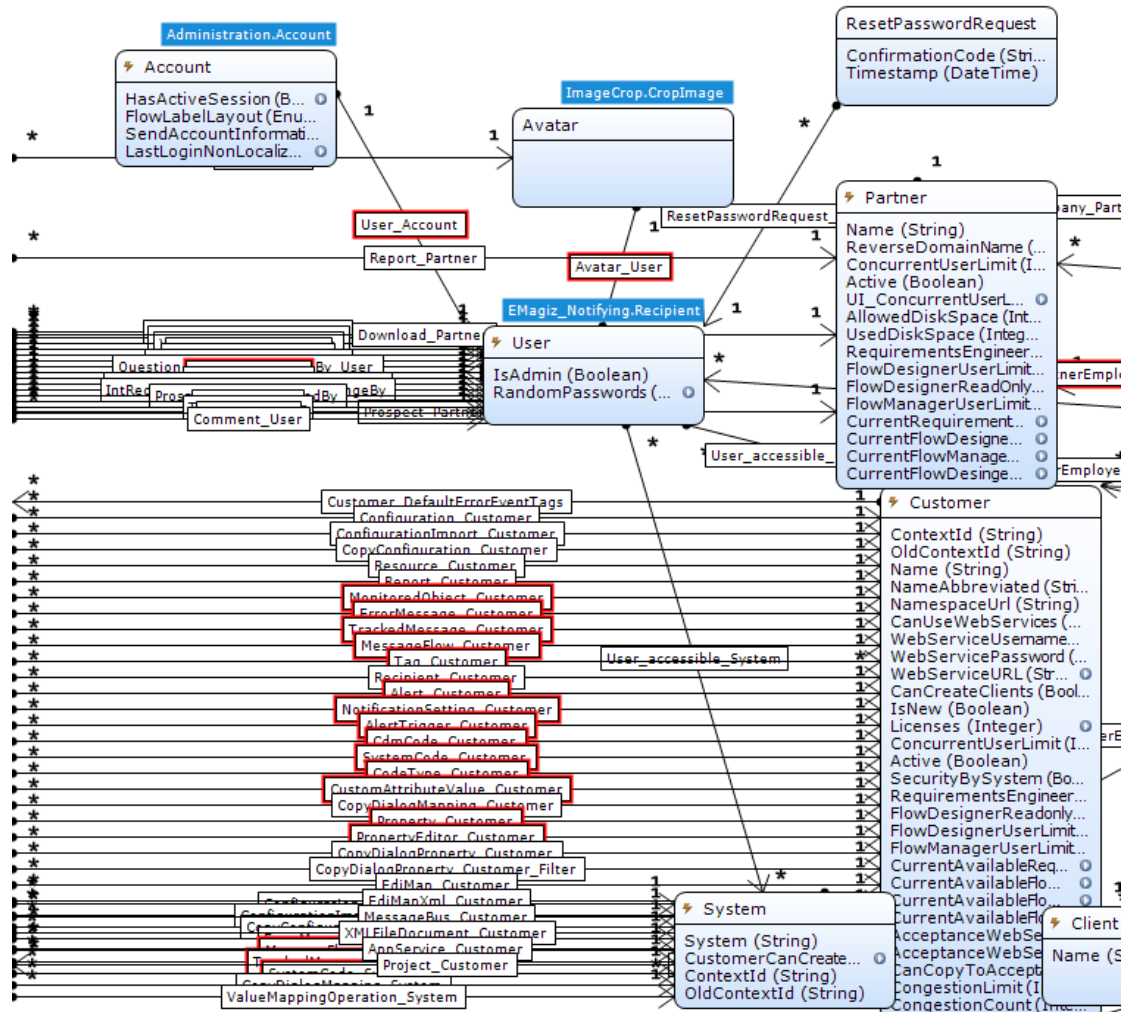
APPENDIX F – MENDIX IMPRESSIE

Domain model

Het domain model vormt de basis van een Mendix module. In het domain model kunnen alle entiteiten, met hun bijbehorende relaties, visueel worden weergegeven. In Figuur 0-5 is het domain model van de App store module te zien. Omdat deze een onderdeel is van een groter project, is soms ook gebruik gemaakt van entiteiten van andere domain modellen, zoals te zien in Figuur 0-6. Hierin is bijvoorbeeld de Customer al gedefinieerd, deze kan ook voor de App store gebruikt worden.



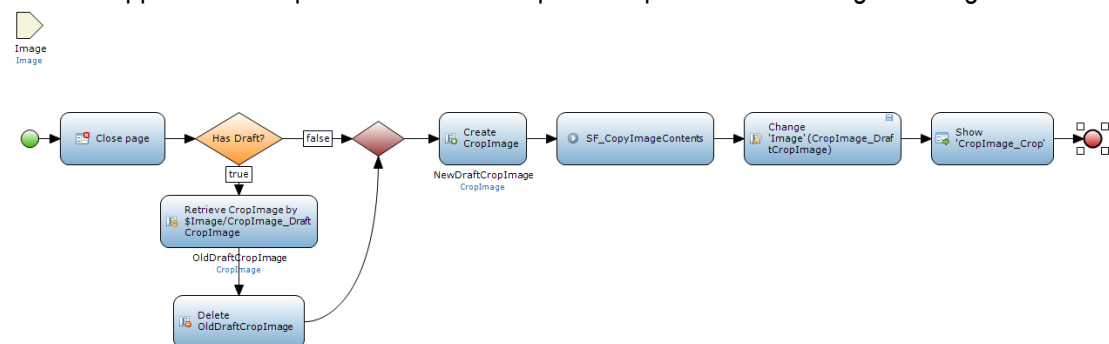
Figuur 0-5: Domain model van de App store



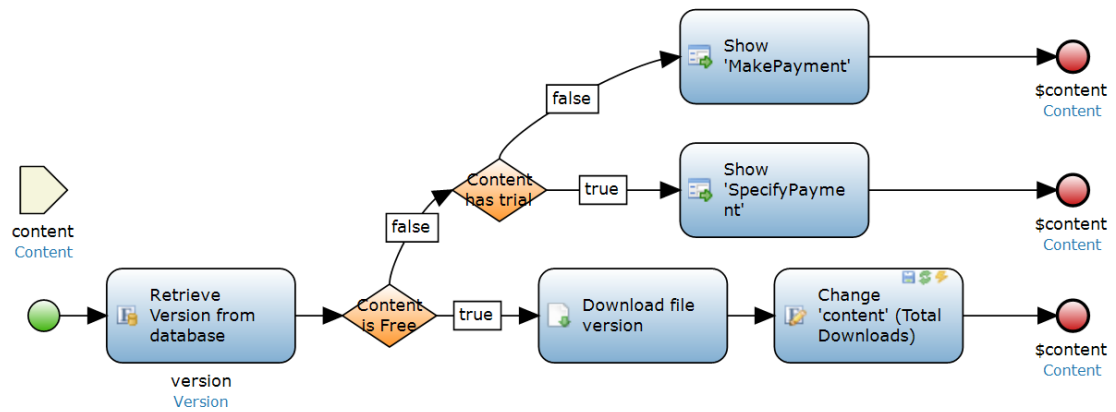
Figuur 0-6: het domain model waar binnen de SaaS onder ander de gebruikers worden gedefinieerd

Microflows

Mendix maakt het mogelijk om allerlei informatie visueel weer te geven, maar soms is weergegeven niet voldoende en moet er ook iets gebeuren. Binnen Mendix maakt men hiervoor gebruik van microflows. Hierin kunnen aan blokken een bepaalde opdracht worden meegegeven. Het is zelfs mogelijk dat dit een nieuwe microflow is, zoals bijvoorbeeld het blok SF_CopyImageContents in Figuur 0-7, waarin het uploadproces van een afbeelding is geregeld. In Figuur 0-8 is te zien welke stappen er doorlopen worden als er op de knop download uit Figuur 4-9 geklikt wordt.



Figuur 0-7: Microflow croplImage waarin het uploadproces van een afbeelding in kaart is gebracht



Figuur 0-8: Microflow DownloadContent

GEBRUIKTE LITERATUUR.

In dit hoofdstuk zijn al de gebruikte bronnen te vinden. Als eerste worden de gebruikte artikelen gegeven, deze worden apart van de andere bronnen genoemd omdat deze belangrijker geacht worden.

Daarna worden de websites genoemd die onder andere gebruikt zijn voor de definities van sommige termen van voornamelijk Gartner of Technopedia.

Archer, L. B. (1984). Systematic Method for Designers. *Developments in design methodology*, 57-82.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., . . . Zaharia, M. (2009). *Above the Clouds: A Berkeley View of Cloud*. Berkeley: University of California .

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., . . . Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 50-58.

Atkinson, C., & Kühne, T. (2003). Model-driven development: A metamodeling foundation. *IEEE Software*, 36-41.

Boehm, B. (1988). Spiral Model of Software Development and Enhancement. *Computer*, 61-72.

CISCO. (2009). *Cisco Global Cloud Index: Forecast and Methodology, 2010–2015*. San Jose, CA: CISCO.

D'Agostino, D., Galizia, A., Clematis, A., Mangini, M., Porro, I., & Quarati, A. (2013). A QoS-aware broker for hybrid clouds. *Computing*, S89-S109.

Elmer-DeWitt, P. (2010). 6 ways iPhone and Android users differ. *Fortune*.

Fenton, W. (2014). *The next generation worker: The Citizen Developer*. Denver, CO: TrackVia.

Frakes, W., & Isoda, S. (1994). Success factors. *IEEE Software*, 14-19.

Frakes, W., & Terry, C. (1996). Software Reuse: Metrics and Models. *ACM Computing Surveys*, 415-435.

Haan, J. d. (2013, October 12). *Building an Agile Enterprise*. Retrieved from The Enterprise Architect: <http://www.theenterpriseearchitect.eu/blog/2013/10/12/the-cloud-landscape-described-categorized-and-compared/>

Hays, J. M. (2003). Forecasting Computer Usage. *Journal of Statistics Education*.

He, X., Ma, Z., Shao, W., & Li, G. (2007). A metamodel for the notation of graphical modeling languages. *Computer Software and Applications*, 219-224.

Heerkens, H., & van Winden, A. (2012). *Geen probleem: Een aanpak voor alle bedrijfskundige vragen en mysteries*. Buren: Business School Nederland.

Hevner, A. R., March, S. T., & Park, J. (2004). Design research in information systems research. *MIS Quarterly*, 75-105.

Jansen, S., & Bloemdal, E. (2013). Defining app stores: The role of curated marketplaces in software. *4th International Conference on Software Business*, (pp. 195-206). Potsdam.

Järvinen, P. (2007). Action research is similar to design science. *Quality & Quantity*, 37-54.

Jonkers, H., Lankhorst, M., Iacob, M.-E., & Proper, E. (2009). Internationale Standaard -- ArchiMate 1.0 is nieuwe modelleertaal voor enterprise architectuur. *Business Process Magazine*, 34-39.

- Lankhorst, M., Proper, H., & Jonkers, H. (2009). The Architecture of the ArchiMate Language. *Enterprise, Business-Process and Information Systems Modeling*, 367-380.
- Moreno-Vozmediano, R., Montero, R., & Llorente, I. (2012). IaaS Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures. *IEEE Computer*, 65-72.
- Pallis, G. (2010). Cloud computing: The new frontier of internet computing. *IEEE Internet Computing*, 70-73.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 45-77.
- Pettey, Christy; Meulen, Rob van der. (2011). *Gartner Says Citizen Developers Will Build at Least 25 Percent of New Business Applications by 2014*. Stamford: Gartner.
- Pezzini, Massimo; Natis, Yefim V.; Malinverno, Paolo; Iijima, Kimihiko; Thompson, Jess; Thoo, Eric. (2014). *Magic Quadrant for Enterprise Integration*. Stamford, CT, United States: Gartner.
- Quarati, A., Danovaro, E., Galizia, A., Clematis, A., D'Agostino, D., & Parodi, A. (2015). Scheduling strategies for enabling meteorological simulation on hybrid clouds. *Journal of Computational and Applied Mathematics*, 438-451.
- Rothenberger, M. A., & Hershauer, J. C. (1999). A software reuse measure: Monitoring an enterprise-level. *Information & Management*, 283-293.
- Selic, B. (2003). The pragmatics of model-driven development. *IEEE Software*, 19-25.
- Somani, C. (2012). VIRTUAL COMMUNITY: The New Hope for E-Commerce. *Indian Journal of Computer Science and Engineering*, 20-23.
- Steenwijk, M. (2011). *A Reference Architecture for Fuel-based Carbon Management Information Systems in the Logistics Industry*. Enschede: University of Twente.
- The open Group. (december 2013). *ArchiMate® 2.1 Specification*. Zaltbommel NL: Haren Publishing, Van.
- Thomson, F. (2014). *Streamlining the distribution channel and partner interactions for eMagiz*. Groningen: Rijksuniversiteit Groningen.

Gebruikte websites

¹ Gartner definition: integration Platform as a Service (iPaaS) <http://www.gartner.com/it-glossary/information-platform-as-a-service-ipaas/>

² Technopedia definition: Virtual Community <http://www.techopedia.com/definition/15504/virtual-community>

³ Technopedia definition: Application Software <http://www.techopedia.com/definition/4224/application-software>

⁴ Technopedia definition: Mobile Application (Mobile App) <http://www.techopedia.com/definition/2953/mobile-application-mobile-app>

⁵ American Dialect Society: "App" voted 2010 word of the year by the American Dialect Society www.american-dialect.org/app-voted-2010-word-of-the-year-by-the-american-dialect-society-updated

⁶ Gartner report: "Consumers Will Spend \$6.2 Billion in Mobile Application Stores in 2010" <http://www.gartner.com/newsroom/id/1282413>

⁷ Technopedia definition: Flowchart <http://www.techopedia.com/definition/5512/flowchart>

⁸ Technopedia definition: Resource Pooling <http://www.techopedia.com/definition/29545/resource-pooling>

⁹ Technopedia definition: Elastic Computing <http://www.techopedia.com/definition/26598/elastic-computing-ec>

¹⁰ Technopedia definition: Measured Service <http://www.techopedia.com/definition/14469/measured-service-cloud-computing>

-
- ¹¹ Techopedia definition: Software as a Service (SaaS) <http://www.techopedia.com/definition/155/software-as-a-service-saas>
- ¹² Techopedia definition: Platform as a Service (PaaS) <http://www.techopedia.com/definition/147/platform-as-a-service-paas>
- ¹³ Gartner definition: integration Platform as a Service (iPaaS) <http://www.gartner.com/it-glossary/information-platform-as-a-service-ipaas/>
- ¹⁴ Gartner definition: Private Cloud Computing <http://www.gartner.com/it-glossary/private-cloud-computing/>
- ¹⁵ Gartner definition: Community Cloud <http://www.gartner.com/it-glossary/community-cloud/>
- ¹⁶ Gartner definition: Public Cloud Computing <http://www.gartner.com/it-glossary/public-cloud-computing/>
- ¹⁷ Gartner definition: Hybrid Cloud Computing <http://www.gartner.com/it-glossary/hybrid-cloud-computing/>
- ¹⁸ Homepage ArchiMate: http://www.archimate.nl/en/about_archimate/what_is_archimate.html
- ¹⁹ <http://www.xconomy.com/boston/2012/03/20/five-facts-about-mendix-another-cloud-enterprise-app-startup/>
- ²⁰ <http://dev.windows.com/en-us>
- ²¹ www.store.steampowered.com/
- ²² <http://techcrunch.com/2013/04/03/readying-for-an-ipo-enterprise-saas-integration-platform-mulesoft-raises-37m-from-nea-salesforce-and-others/>
- ²³ <http://www.snaplogic.com/why-snaplogic/what-we-do>
- ²⁴ <http://blogs.windows.com/buildingapps/2014/09/17/new-dev-center-lifetime-registration-benefits-program/>
- ²⁵ <http://lifehacker.com/5842307/how-to-supercharge-all-your-favorite-webapps-with-ifttt>
- ²⁶ <http://www.christianposta.com/blog/?p=161>
- ²⁷ <http://techcrunch.com/2012/07/09/github-pours-energies-into-enterprise-raises-100-million-from-power-vc-andreesen-horowitz/>