GAMIFICATIE IN ENTERPRISE APPLICATIES

Afstudeerscriptie

Student: Martin Bakker
Student #: 1631933

Titel	Afstudeerscriptie
Project/Onderwerp	Gamificatie in Enterprise Applications
Versie	2.0
Status	Definitief
Datum	01-07-2016
Bedrijf	Info Support
Afstudeerder	M. Bakker
Opleiding	HBO ICT
Onderwijsinstelling	Hogeschool Utrecht
Eerste examinator	R. Ruijsenaars
Docent begeleider	B. van Eijkelenburg
Opdrachtgever	J. Snijder

Info Support B.V.
Kenniscentrum

Kruisboog 42, 3905 TG, Veenendaal,
De Smalle Zijde 39, 3903 LM Veenendaal,

Tel. +31 (0) 318 55 20 20 - Fax +31 (0) 318 55 23 55
Tel. +31 (0) 318 50 11 19 - Fax +31 (0) 318 51 83 59

Historie

1.0	Concept	29-04-2016	*Creatie*
1.0	Concept	31-05-2016	*Invulling steekwoorden na overleg met Joop*
1.1	Concept	06-06-2016	*Start deelvragen 5.2.1 t/m 5.2.4*
1.2	Concept	07-06-2016	*Toevoegingen, opgenomen a.d.h.v. review commentaar*
1.3	Concept	09-06-2016	*Eerste verwerking feedback Joop en Thomas*
1.4	Concept	13-06-2016	*Toevoegingen, opgenomen a.d.h.v. review commentaar*
1.5	Concept	16-06-2016	*Verwerking feedback Joop, Thomas en Bart*
1.6	Concept	22-06-2016	*Controle en aanvullingen (focus: PoC)*
1.7	Concept	24-06-2016	*Verwerking feedback Bart, Thomas en Joop*
1.8	Concept	25-06-2016	*Volledige invulling scriptie, core-scriptie 49 pagina's (van de max 50).*
1.9	Concept	25-06-2016	*Paginanummering.*
2.0	Concept	30-06-2016	*Opmaak, taal en opmaak*
2.0	Definitief	01-07-2016	*Compleet*

Distributie

1.0	Concept	31-05-2016	*J. Snijder, T. de Klerk*
1.1	Concept	06-06-2016	*J. Snijder, T. de Klerk*
1.2	Concept	07-06-2016	*J. Snijder, T. de Klerk*
1.3	Concept	10-06-2016	*B. van Eijkelenburg*
1.4	Concept	13-06-2016	*J. Snijder, T. de Klerk, B. van Eijkelenburg*
1.5	Concept	21-06-2016	*J. Snijder, T. de Klerk*
1.6	Concept	22-06-2016	*B. van Eijkelenburg*
1.7	Concept	24-06-2016	*J. Snijder, T. de Klerk, B. van Eijkelenburg*
1.8	Concept	25-06-2016	*J. Snijder, T. de Klerk, B. van Eijkelenburg*
1.9	Concept	01-07-2016	*H. Luijk*
2.0	Definitief	05-07-2016	*Hogeschool Utrecht, Info Support*

© Info Support, Veenendaal 2016

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook, zonder voorafgaande toestemming van Info Support.

No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by Info Support.

Prijsopgaven en leveringen geschieden volgens de Algemene Voorwaarden van Info Support B.V., gedeponeerd bij de K.v.K. te Utrecht onder nr. 30135370. Een exemplaar zenden wij u op uw verzoek per omgaande kosteloos toe.

Voorwoord

Voor u ligt de scriptie 'Gamification in Enterprise Applications'.
Na vier jaar te hebben gestudeerd aan de Hogeschool Utrecht moet er middels deze scriptie worden aangetoond dat ik werkzaam mag zijn in de sector ICT en specifiek als software engineer. Deze scriptie is uitgevoerd in opdracht van Info Support waar ik van maart 2016 tot augustus 2016 bezig ben geweest met het onderzoek, het ontwikkelen van een proof-of-concept en het schrijven van de scriptie.

Naar aanleiding van een geconstateerd probleem door de heer Joop Snijder, tevens mijn opdrachtgever en bedrijfsbegeleider, zijn er onderzoeksvragen opgesteld. Deze onderzoeksvragen zijn gedurende de afstudeerperiode beschreven, verdiept en onderzocht. Na de nodige energie en tijd die het heeft gekost om dit onderzoek te verrichten is er op alle onderzoeksvragen een antwoord gegeven. Gedurende het afstudeerproces hebben de volgende personen mij actief begeleid: Bart van Eijkelenburg (docent begeleider), Joop Snijder (opdrachtgever/ bedrijfsbegeleider), Marieke Keurntjes (proces begeleidster) en Thomas de Klerk (technisch begeleider). Middels hun hulp is het gelukt om in een prettige en professionele sfeer de werkzaamheden te verrichten. Graag wil ik hen bedanken voor de actieve begeleiding. Daarbij hebben mij voornamelijk de feedbackmomenten veel goed gedaan om inzicht te krijgen over het proces en mijn sterkere en minder sterkere kanten, om zo persoonlijk te kunnen groeien.

Ook wil ik collega's bedanken die mij veel inzicht hebben kunnen geven op een tal van gebieden. Er zijn momenten geweest dat ik toch wel even vast zat in het denk en onderzoekproces, daarbij hebben Thomas Bruggink, Wiljag Denekamp en Willem Meints ontzettend veel bijgedragen aan het komen tot mogelijke oplossingen waarbij ik niet langer vast zat.

Robin Altena, een klasgenoot heeft me in het "studie" gedeelte van het proces en persoonlijk veel opgeleverd als gesprekspartner. Door het sparren over verwachtingen en problemen die we tegen zijn gekomen heb ik de ruimte gehad om verder te komen.

Als laatste wil ik mijn ouders en vriendin Hanneke bedanken voor de steun die ik heb ervaren gedurende mijn afstuderen. Afstuderen gaat vrijwel niemand "makkelijk" af aangezien het proces een soort "open einde" is waarbij de lat ligt waar je hem zelf legt. Middels het kunnen delen van de emoties die komen kijken bij een uitdagend proces ben ik in staat geweest er goed uit te komen en uiteindelijk een hele fijne, doch redelijk drukke afstudeerperiode te hebben gehad.

Wat mijn docent Bart van Eijkelenburg in het proces ergens aangaf: "Stand on the shoulders of giants" heb ik veel nieuwe kennis en waarheden kunnen ontdekken door gebruik te mogen maken van kennis en mensen om me heen.

Ik hoop dat u de scriptie met plezier leest en dat u iets mee krijgt van het proces wat ik heb doorlopen.

Martin Bakker

Amersfoort, 24-06-2016

Managementsamenvatting

Gamificatie is een methode welke kan worden ingezet om menselijk gedrag op een positieve wijze te sturen. De methode richt zich op het stimuleren van intrinsieke motivatie waardoor invloed wordt uitgeoefend op gewenst gedrag. Vanuit de gebruiker, niet vanuit een leidinggevende. Dit kan goed worden ingezet binnen softwareapplicaties. Een probleem hiermee is echter dat het relatief gezien te weinig ingezet wordt voor de waarde die eruit gehaald kan worden. Het kost veel tijd en daarmee geld om gamificatie in een applicatie toe te voegen omdat het vaak nieuw terrein is voor een bedrijf.
Om ervoor te zorgen dat de drempel om gamificatie in te zetten binnen enterprise applicaties lager wordt zijn de volgende vragen opgesteld om te onderzoeken.

- Hoe kan een herbruikbare, onafhankelijke gamificatieservice worden opgezet zodat deze toegepast kan worden binnen enterprise applicaties?
    1. Wat zijn kenmerkende eigenschappen van enterprise applicaties?
    2. Hoe kan gamificatie als onafhankelijke service worden aangeboden aan enterprise applicaties?
    3. Welke gamificatie elementen zou een gamificatie service voor enterprise applicaties moeten omvatten?
    4. Hoe kunnen generieke spelregels worden opgesteld voor een gamificatieservice?
    5. Uit welke softwarecomponenten moet een gamificatieservice bestaan?

Om antwoorden te krijgen op deze vragen heeft onderzoek plaatsgevonden. De gekozen onderzoeksmethodiek is van kwalitatieve aard. Daarbij zijn interviews afgenomen met softwarearchitecten die werken in de context van enterprise applicaties. Er is exploratief een relatief nieuw kennisgebied rondom gamificatie onderzocht. Er heeft inventarisatieonderzoek en descriptief onderzoek plaatsgevonden op de gebieden van gamificatie. Dit is gedaan om de mogelijke technologieën voor een service te bepalen. Er is toegepast onderzoek gedaan om de verweven kennis te toetsen in de praktijk.

Uit het onderzoek is gebleken dat er een gamificatie service kan worden aangeboden aan enterprise applicaties op de volgende manier.

- Om te voldoen aan de kwaliteitseisen die een enterprise applicatie verwacht moet worden voldaan aan een aantal eisen beschreven in de ISO-standaard 25010 *"Systems and software Quality Requirement and Evaluation".* Deze zijn in de context van het onderzoek: performance, betrouwbaarheid, security, beheerbaarheid en overdraagbaarheid.
- Er een "software as a service" model wordt gekozen om middels een REST webservice gamificatie te kunnen aanbieden. Het onderzoek toont aan dat het een valide en verantwoorde keuze is, wat de goedkeuring van software architecten ontvangt.
- Er in ieder geval drie gamificatie elementen in de service verwerkt zijn. Deze zijn levels, punten en badges. Er is onderzocht welke elementen er zijn en in hoeverre ze bruikbaar zijn. De conclusie daaruit is dat de genoemde drie een uitstekende basis bieden voor een gamificatieservice.
- De service instelbaar is voor verscheidene enterprise applicaties wanneer het speelveld te modelleren is door de klant. In de vierde deelvraag wordt aangegeven dat door acties, effecten en regels individueel instelbaar te maken in de context van de enterprise applicatie er een vorm van generaliteit kan worden behaald.

Het onderzoek is afgerond met een uitgewerkte proof-of-concept welke aantoont hoe een gamificatieservice eruitziet wanneer het voldoet aan bovengenoemde punten. Dit houdt in dat gamificatie ingezet kan worden in verscheidene enterprise applicaties zonder dat deze binnen het eigen applicatielandschap hoeven na te denken over de logica rondom een gamificatie service. Toekomstig onderzoek moet worden gedaan om mogelijke kostenbesparing door het aanbieden van de service ten opzichte van een eigen gamificatie implementatie te hosten te vergelijken.

Inhoudsopgave

FIGUREN-, VERGELIJKINGEN- EN TABELLENLIJST **7**

LIJST MET AFKORTINGEN **7**

BEGRIPPENLIJST **8**

INLEIDING **9**

1 CONTEXT **10**

1.1 Organisatie - Info Support 10
1.2 Begeleiding 11
1.3 Opdrachtnemer en opdrachtgever 12

2 OPDRACHT **13**

2.1 Opdrachtdefinitie 13
2.2 Aanleiding 13
2.3 Doelstellingen 14
2.4 Hoofd- en deelvragen 15

3 METHODIEK **16**

3.1 Onderzoek 16
3.2 Verificatie en Validatie 17

4 ONDERZOEKSRESULTATEN **19**

4.1 Deelvraag 1: Enterprise applicaties 19
4.2 Deelvraag 2: Onafhankelijke service 22
4.3 Deelvraag 3: Gamificatie elementen 27
4.4 Deelvraag 4: Generieke spelregels 30
4.5 Deelvraag 5: Softwarecomponenten 33

5 PROOF-OF-CONCEPT **37**

5.1 Back-end: Gamificatieservice 38
5.2 Front-end: Speelveldmodelleeringtool 42
5.3 Repository: Database 43

6 CONCLUSIE EINDRESULTAAT **44**

6.1 Aanbevelingen 44

7 EVALUATIE PROCESGANG **45**

BIBLIOGRAFIE **48**

Bijlagen

BIJLAGE 1: PLAN VAN AANPAK 50
BIJLAGE 2: GAMIFICATIE ELEMENTEN 76
BIJLAGE 3: SCHERMONTWERPEN 80
BIJLAGE 4: RETROSPECTIVE RESULTATEN 82
BIJLAGE 5: LOGICA UITDAGING SCENARIO'S 84
BIJLAGE 6: GARTNER HYPE CYCLE 86

Figuren-, vergelijkingen- en tabellenlijst

Figuur 1: Gartner hype cycle 9
Figuur 2: Organogram 10
Figuur 3: Info Support tijdlijn 10
Figuur 4: Info Support Medewerkers 11
Figuur 5: Begeleiding 12
Figuur 6 : Voorbeeld SOAP XML syntax 25
Figuur 7 : Voorbeeld JSON Object 26
Figuur 8 : Gebruik gamificatie elementen 28
Figuur 9 : Gamificatie doelen 30
Figuur 10 : Weg naar eigen gamificatie 31
Figuur 11 : Software componenten 33
Figuur 12 : software componenten GS 34
Figuur 13 : Sequence actie toevoegen 35
Figuur 14 : Voorbeeld gamificatieregels 37
Figuur 15: POC onderdelen 38
Figuur 16: Back-end classes 38
Figuur 17: Scenario 1 - Speelveldmodellering 39
Figuur 18: Scenario 2 – Event providing 40
Figuur 19: Scenario 3 – Consuming results 40
Figuur 20: Overzicht acties, effecten en regels in de tijd 41
Figuur 21: Speelveldmodelleringtool screenshot 43
Figuur 22: Voorbeeld query execution 43
Figuur 23: Feedback methodiek 46

Tabel 1: Begrippenlijst 8
Tabel 2: 47 gamificatie elementen volgens Marczewski 27
Tabel 3 : Voorbeeld experience points voor level 31
Tabel 4 : Communicatie vormen tussen componenten 34
Tabel 5: Voordelen 3-tier architecture 35
Tabel 6: Personen en processen bij het onderzoek 45

Vergelijking 1: Software on premises vs Software as a service 24
Vergelijking 2 : REST vs SOAP 26

Lijst met afkortingen

Afkorting	Betekenis
API	Application programming interface
CRUD	Create, Read, Update, Delete
EA	Enterprise applicatie
EF	Entity Framework
GS	Gamificatieservice
HU	Hogeschool Utrecht
REST	Representional state transfer
SAAS	Software as a service
SMT	Speelveldmodelleringtool
SOAP	Simple object access protocol

Begrippenlijst

Gedurende het document worden enkele begrippen gebruikt.

Begrip	Beschrijving
Actie(s)	Een actie is een handeling binnen de enterprise applicatie. Iets wat de gebruiker kan doen binnen het systeem. In de context van “knowNow” is dit bijvoorbeeld: ‘Een artikel toevoegen’.
Badge(s)	Een badge is een soort “medaille” die een gebruiker kan verdienen door aan iets te voldoen in het systeem.
Consumer	In de context van enterprise applicaties is een enterprise applicatie een consumer wanneer het calls maakt naar de gamificatie service om op te vragen wat de resultaten voor een gebruiker zijn.
Effect(en)	Effecten kunnen worden gezien als “beloningen”. Wanneer een gebruiker bepaalde acties voltooid, kan hier een effect aan worden gekoppeld.
Enterprise applicatie	Een enterprise applicatie is een applicatie dat (grote) bedrijven ondersteunt in processen. Dit kan zowel intern als extern zijn. Een enterprise applicatie is in de context van Info Support bijvoorbeeld “knowNow”. In de context van het onderzoek wordt er gesproken over de enterprise applicatie waar gamificatie op moet kunnen worden toegepast.
Events	Events worden door de enterprise applicatie als provider afgevuurd naar de gamificatie service. Dit zijn ook de acties die bekend worden gemaakt bij de gamificatie service in de SMT.
Gamificatie service	De gamificatie service is verantwoordelijk voor het afhandelen van alle logica rondom gamificatie.
Gebruiker	De gebruiker is de zogenaamde “eindgebruiker”. Dit zijn de gebruikers die gebruik maken van de enterprise applicatie.
knowNow	knowNow is een enterprise applicatie ontwikkeld door Info Support. Het is een kennismanagement systeem waarmee kennis kan worden gedeeld onder collega’s doormiddel van artikelen.
Level	Levels zijn een vorm van “voortgang” van een gebruiker. Wanneer een gebruiker veel activiteit vertoont in een systeem zal hij stijgen in level.
Productowner	De productowner is de “eigenaar” van de enterprise applicatie.
Provider	In de context van enterprise applicaties is een enterprise applicatie een provider wanneer het calls maakt naar de gamificatie service waarbij een actie wordt uitgevoerd.
Punten	Punten zijn een vorm van beloning. De gebruiker kan punten verdienen door iets te doen in het systeem. Wat deze punten “waard” zijn is aan de productowner om te bepalen.
Regels	Regels zijn de definities van welke effecten het resultaat zijn van welke acties.
Speelveldmodelleerder	Dit is een persoon die toegang heeft to de SMT, zie speelveldmodelleringtool.
Speelveldmodelleringtool	De speelveldmodelleringtool (SMT) is er om het speelveld te modelleren. Doordat het mogelijk is gamificatie zelf te modelleren ontstaat er heel veel vrijheid voor de vorm van gamificatie.

Tabel 1: Begrippenlijst

Inleiding

Gamificatie is een methode welke kan worden ingezet om menselijk gedrag op een positieve wijze te sturen. De methode richt zich op het stimuleren van intrinsieke motivatie waardoor invloed wordt uitgeoefend op gewenst gedrag.

"Het geloof in de potentie van gamification is groot en de kansen die Nederlandse managers zien zijn enorm. Gamification leeft in Nederland, maar het bedrijfsleven is nog zoekende naar expertise en bewijs."
(Willemse, 2013)

(Nielson, 2013)

Figuur 1: Gartner hype cycle

In de zogenaamde "hype cylce" van Gartner (bijlage 6) hiernaast is te zien dat gamificatie hoge verwachtingen schept in de komende jaren. Echter zoals in de quote hierboven is te lezen, zijn bedrijven in Nederland nog zoekende naar of het werkt en hoe het ingezet dient te worden. Daarbij lijkt het nog niet de energie en tijd te krijgen die het verdient. Mogelijk slaan bedrijven daarmee uiteindelijk de plank mis wanneer ze te laat mee gaan met gamificatie.

Dit probleem is bij het bedrijf Info Support ook geconstateerd door de heer Joop Snijder. Het bedrijf Info Support wil zijn klanten kunnen helpen en begeleiden om voorop te blijven lopen in de markt. Aangezien Info Support een breed scala aan grootschalige klanten heeft is er een opdracht opgesteld om onderzoek te doen naar het inzetten van gamificatie binnen enterprise applicaties. Het onderzoek dient een resultaat te bieden waarbij gamificatie kwalitatief en vergemakkelijkt ingezet kan worden binnen enterprise applicaties, om zo mee te kunnen bewegen met de verwachte trend.

Om dit te kunnen realiseren is kennis nodig. Daarbij is in grote lijnen kennis nodig over gamificatie, enterprise applicaties en software ontwerp en implementatie. Voor het verzamelen van die kennis is de hoofdvraag opgesteld: Hoe kan een herbruikbare, onafhankelijke gamificatieservice worden opgezet zodat deze toegepast kan worden binnen enterprise applicaties? Na antwoord op deze vraag zal het mogelijk zijn een oplossing te bieden aan klanten van Info Support rondom gamificatie.

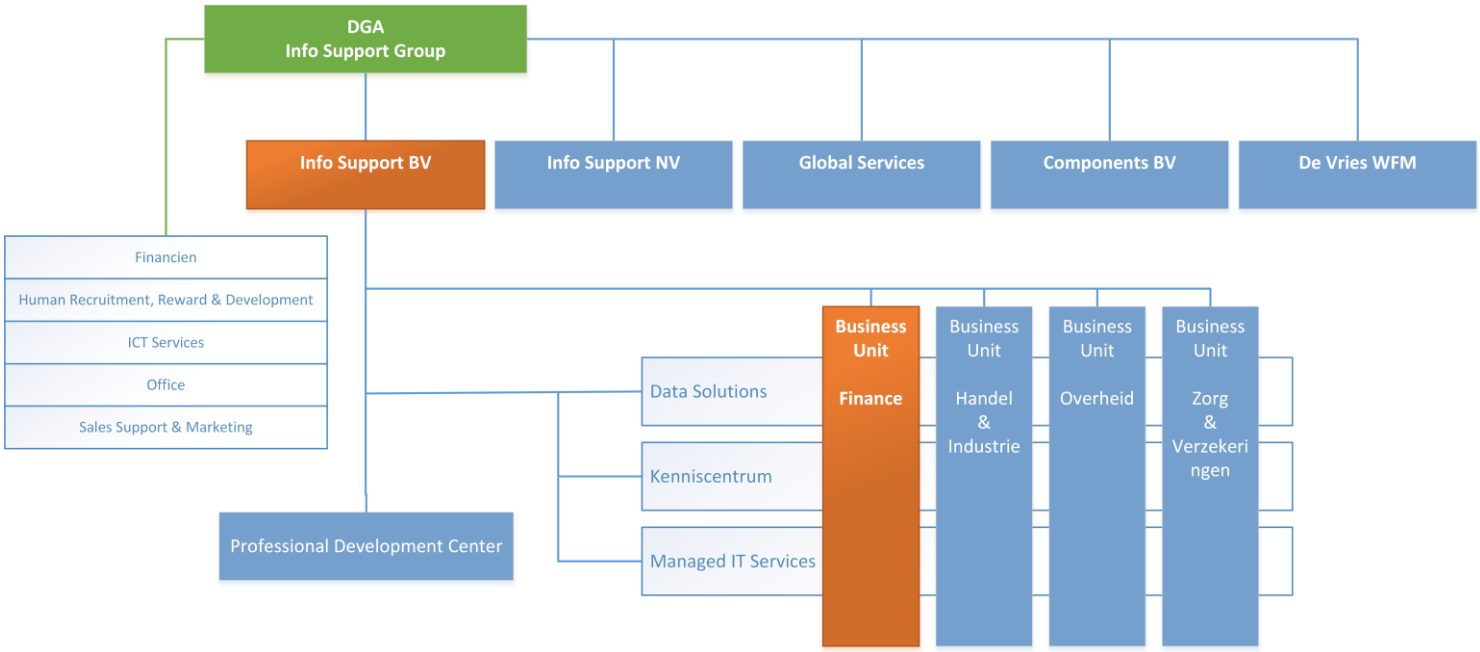
Deze scriptie bevat 7 hoofdstukken ten behoeve van het beantwoorden van de hoofdvraag.

1) Hoofdstuk één zal inzicht geven in de context van het proces en het bedrijf. Er wordt een beeld geschetst van de organisatie Info Support, de begeleiding gedurende de opdracht en de opdrachtgever en opdrachtnemer.
2) In het tweede hoofdstuk zal vervolgens de opdracht worden geschetst met daarin de opdrachtdefinitie, de aanleiding en doelstellingen en vervolgens de opgestelde hoofd- en deelvragen.
3) Vervolgens zal het derde hoofdstuk de methodiek van het onderzoek beschrijven. Daarin zijn onderzoekmethoden beschreven en de manier waarop gevalideerd en geverifieerd is.
4) In het vierde hoofdstuk worden de onderzoeksresultaten beschreven in het volgende format: aanpak, resultaten en conclusie. Deze onderzoeksresultaten zijn antwoorden op de deelvragen.
5) Hoofdstuk vijf beschrijft hoe er tot een proof-of-concept is gekomen. Deze wordt gerealiseerd om aan te tonen dat er gamificatie kan worden aangeboden aan enterprise applicaties.
6) Dan wordt in hoofdstuk zes de conclusie en de eindresultaten besproken over het gehele onderzoek en het proof-of-concept. Hierin zullen alle bevindingen worden meegenomen om tot een antwoord op de hoofdvraag te komen.
7) Als laatst wordt in hoofdstuk zeven de discussie behandeld waarbij wordt gekeken waar in het proces wellicht iets anders had gekund. Dit om de uitvoering van toekomstig onderzoek te kunnen verhogen.

1 Context

1.1 Organisatie - Info Support

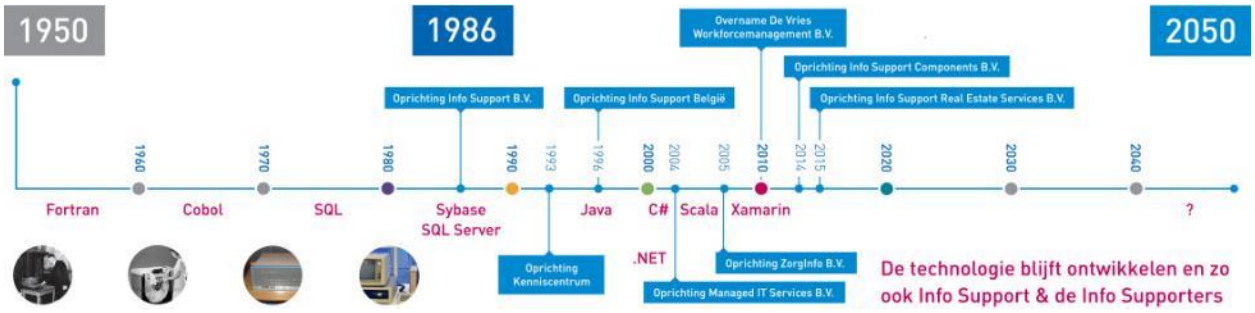
In het Organogram hieronder is te zien dat Info Support een vrij uitgebreide organisatie is. Daarin zijn meerdere takken en vestigingen.
Binnen de organisatie is de afstudeerder werkzaam in de tak Info Support BV en daarin in de businessunit finance. Dit omdat het product knowNow onder deze unit wordt ontwikkeld en beheerd. Dit product is een van de softwareapplicaties die gebruik gaat maken van de uitkomst van het onderzoek: de gamificatieservice.



Figuur 2: Organogram

1.1.1 Geschiedenis

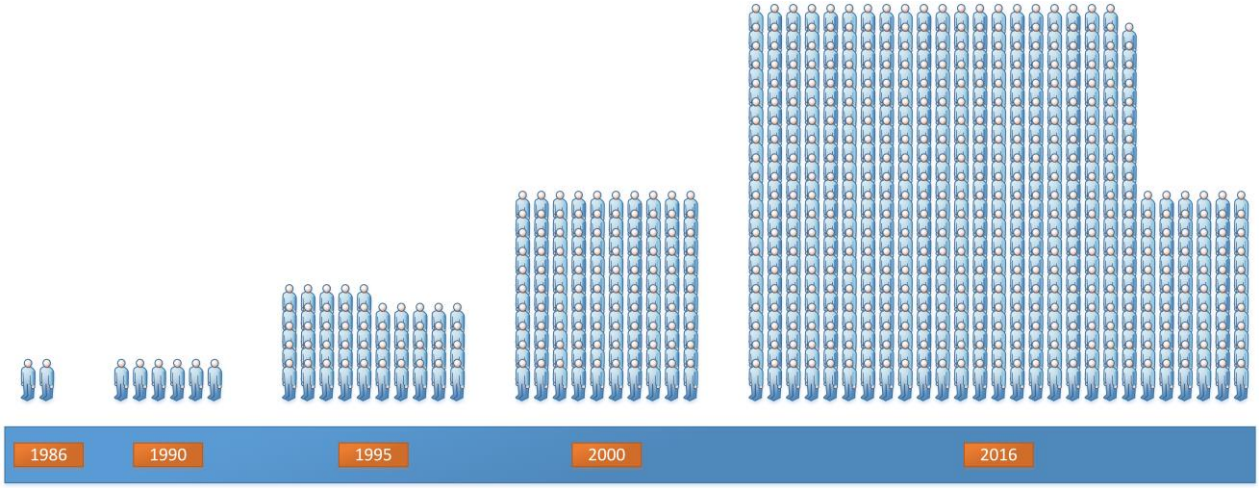
Info Support is in 1986 opgericht door de heer T. Jansen. Hieronder een kort overzicht van hoe Info Support zichzelf heeft ontwikkeld in de afgelopen 30 jaar



Figuur 3: Info Support tijdlijn

1.1.2 Werknemers

Op het moment heeft Info Support ruim 400 medewerkers.



Figuur 4: Info Support Medewerkers

1.2 Begeleiding

1.2.1 Algemeen

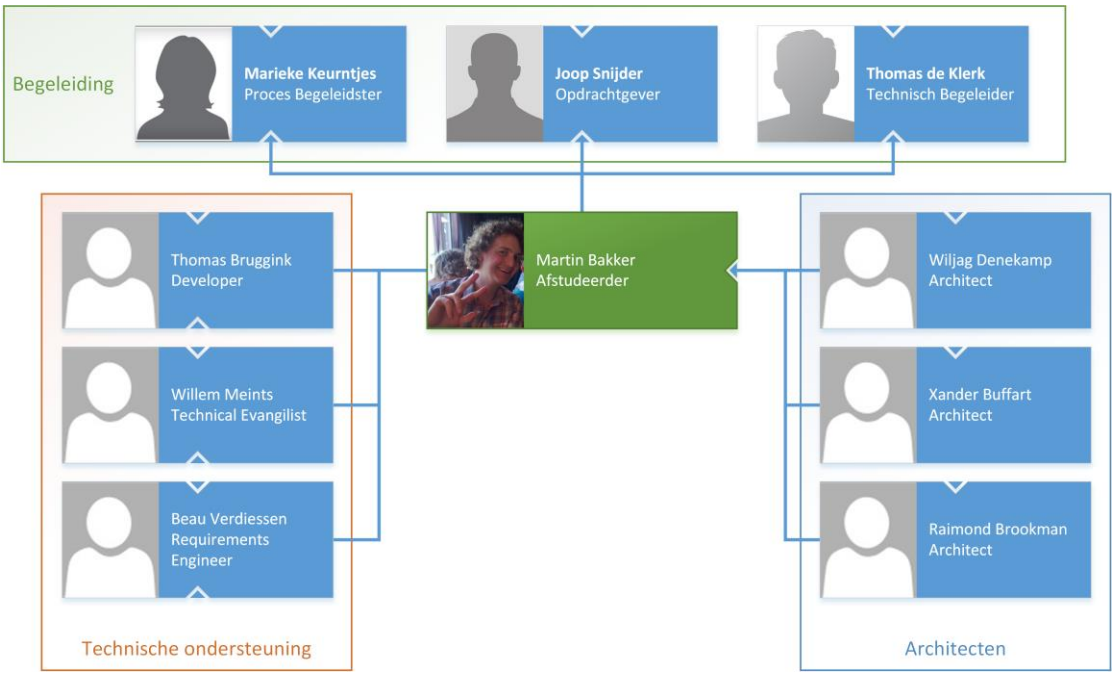
Voor het proces is bij Info Support veel aandacht. Gedurende een afstudeerproces zijn er drie medewerkers die standaard aan een afstudeerder zijn gekoppeld. Dit maakt dat gedurende het proces altijd iemand als aanspreekpunt kan dienen. Deze drie medewerkers hebben verschillende verantwoordelijkheden. Deze zijn:

- **Opdrachtgever**
  De opdrachtgever heeft de opdracht verzonnen en is daarmee verantwoordelijk voor het duidelijk maken wat de opdracht inhoudt en wat het gewenste resultaat is.
- **Procesbegeleiding**
  De procesbegeleiding is een keer in de twee weken, waarbij gekeken wordt hoe het gaat met de afstudeerder.
- **Technische begeleiding**
  De technische begeleiding is ten behoeve van alle vraagstukken van technische aard waar de afstudeerder tegenaan kan lopen. Dit kan op document niveau zijn (functioneel/technisch ontwerp) maar ook ontwikkel niveau (nieuwe technieken als Angular, SQL-server, etc.)

Deze vallen in de sectie “begeleiding” in Figuur 5: Begeleiding.

1.2.2 Specifiek

Gedurende het uitvoeren van de opdracht zijn er nog twee secties van begeleiding die benoemt moeten worden.
De afstudeerder zit in principe altijd samen met Info Support medewerkers op een kamer. Deze vallen in het figuur onder “technische ondersteuning”. Dat wil zeggen dat er op een dag overlegd kon worden, vragen gesteld, code reviews, etc. Dit is een zeer waardevolle toevoeging geweest gedurende het proces.
Vervolgens is de sectie “architecten” specifiek voor de uitgevoerde opdracht. Daarbij is een van de validatiemomenten dat de opdracht wordt gevalideerd met drie software architecten. Deze zijn ook een vorm van begeleiding geweest, aangezien daar vaak waardevolle input uit kwam.



Figuur 5: Begeleiding

1.3 Opdrachtnemer en opdrachtgever

De opdrachtnemer is:
Martin Bakker, 4e jaar software engineer student aan de Hogeschool Utrecht.

Mobiel	0031681864194
Email	martin.bakker@student.hu.nl
LinkedIn	https://nl.linkedin.com/in/martinbakkerict

De opdrachtgever is:
Joop Snijder, werkzaam te Info Support als productowner van knowNow.

Mobiel	0031653643473
Email	joop.snijder@infosupport.com
LinkedIn	https://www.linkedin.com/in/joopsnijder

2 Opdracht

Dit is de opdracht zoals deze was geformuleerd door Info Support.

> *Gamificatie (Engels: gamification) is het gebruik van spelprincipes en speeltechnieken in een niet-spelcontext, om de gebruikers te stimuleren hun taken te vervullen. Gamificatie wordt gebruikt in applicaties en processen voor het bevorderen van de betrokkenheid van gebruikers, datakwaliteit, punctualiteit en leren.*
>
> *De vraag is hoe gamificatie ingezet kan worden voor enterprise applicaties, waarbij de nadruk in deze opdracht ligt in het verhogen van de betrokkenheid van gebruikers en het bevorderen van het leren kennen van een applicatie.*
>
> *Binnen Info Support wordt gebruik gemaakt van een belangrijke enterprise applicatie voor het delen van kennis en het vinden van experts. Dit systeem heet knowNow. knowNow bevat op dit moment twee gamificatie elementen en kan goed dienen als proeftuin voor deze opdracht.*

2.1 Opdrachtdefinitie

Na verdere analyse met de productowner is de opdracht niet volledig vertaalbaar uit de bovenstaande beschrijving. Daarbij is gekeken of de opdracht in één zin was te verwoorden om concreet te krijgen wat hij onderzocht wilde hebben. Daar is het volgende uitgekomen:

- Onderzoek hoe gamificatie ingezet kan worden voor enterprise applicaties.

Dit is hoe de opdracht in eerste instantie was gedefinieerd vanuit de organisatie. Om concreet te krijgen waarom de opdracht wordt uitgevoerd en tot welke resultaten dit moet leiden zal in de onderdelen “Aanleiding” en “Doelstellingen” dit worden toegelicht.

2.2 Aanleiding

Gamificatie is een bewezen methode die in veel contexten ingezet kan worden om gedrag van gebruikers te beïnvloeden. Vaak wordt dit gedaan om de motivatie van gebruikers te verhogen. Er wordt steeds meer bekend over gamificatie van softwareapplicaties, toch wordt het weinig of verkeerd ingezet.

Dit is iets wat de heer J. Snijder constateerde en daarbij vroeg hij zich af of het niet mogelijk was om dit te veranderen. Om hier achter te komen heeft hij het probleem opgeschreven in de vorm van een afstudeeropdracht.

Een van de redenen dat gamificatie weinig wordt ingezet is dat het aanzienlijk veel energie en tijd kost om het goed in te zetten in zowel een nieuwe applicatie als in een bestaand applicatielandschap. Gezien de effecten van gamificatie en de voordelen die hieruit behaald kunnen worden is het een mooi streven om in de toekomst gamificatie met relatief weinig moeite technisch te implementeren in bestaande en nieuwe software.

2.3 Doelstellingen

Het doel van de opdracht is om gamificatie een betere kans te geven in de wereld van softwareontwikkeling. In eerste instantie in de context van Info Support. Het doel van het onderzoekgedeelte is om er achter te komen wat voor problemen je tegenkomt bij het maken van een generieke gamificatieservice voor verschillende applicatielandschappen en daar oplossingen voor te bieden.

Concreet zijn de doelen van deze opdracht:

- Inzetten van gamificatie in software vergemakkelijken
- Kennis opdoen in het kader van gamificatie in verschillende enterprise applicatielandschappen.
- Afstuderen aan de Hogeschool Utrecht.

Er zitten theoretische doelstellingen tussen en praktische doelstellingen. Hierin wordt door de opdracht heen, onderzoek en programmeren gecombineerd.
Er is voldaan aan de doelstellingen als na het proces een acceptabele manier is gevonden om gamificatie aan te bieden aan enterprise applicaties. Daarnaast moet deze scriptie voldoende informatie verschaffen over het proces naar deze uitkomst om toekomstig onderzoek te kunnen ondersteunen.
Met betrekking tot de doelstelling “afstuderen aan de Hogeschool Utrecht” is gekeken naar de competenties die verwacht worden van een student wanneer deze afgestudeerd is. De specificatie hiervan staan in Bijlage 1: plan van aanpak onder hoofdstuk 7 Persoonlijke ontwikkeling.

2.3.1 Voorbeeldscenario

Om te voorkomen dat het concept gamificatie in enterprise applicaties erg abstract blijft hieronder een schets van een voorbeeldscenario:

Info Support heeft een kennismanagement systeem voor zijn medewerkers. Dit product genaamd knowNow wordt gebruikt door de medewerkers van Info Support.
Een medewerker is de **gebruiker** van de **enterprise applicatie** genaamd knowNow. De **productowner** van knowNow heeft aangegeven gamificatie te willen toevoegen aan de applicatie. Hiervoor kan knowNow gebruik gaan maken van de **gamificatie service.** Er is bijvoorbeeld bedacht dat wanneer een gebruiker 5 keer een artikel aanmaakt, hij een badge als **effect** verdient. Dat 5 keer een artikel aanmaken een badge (“Meesterschrijver”) als effect oplevert is een **regel**. De **speelveldmodelleerder** kan in de **Speelveldmodelleringtool (SMT)** deze **actie** (artikel aanmaken) als **event** (AddArticle) configureren. Dit maakt dat dit event bekend is bij de gamificatie service. Wanneer de enterprise applicatie nu dit event (AddArticle) naar de gamificatie service stuurt, is de enterprise applicatie de **provider**. De gamificatie service registreert dat de gebruiker de actie heeft gedaan. Zodra dit 5 keer is voorgekomen voor dezelfde gebruiker, zorgt de gamificatie service ervoor dat aan de gebruiker een badge (“Meesterschrijver”) is gekoppeld. De enterprise applicatie kan de voortgang voor een gebruiker opvragen, dit maakt de enterprise applicatie dan **consumer.** Wanneer de enterprise applicatie opvraagt welke effecten de gebruiker heeft behaald zal de gamificatie service teruggeven dat de gebruiker een badge genaamd “Meesterschrijver” heeft verdiend.

Verderop in het document zullen enkele van deze concepten en termen meer aan de orde komen. Het scenario is alvast beschreven om een algemeen beeld te geven van het modelleren en inzetten van gamificatie bij een enterprise applicatie.

2.4 Hoofd- en deelvragen

In het voorloop van het proces is gekeken naar de aanleiding en de aard van het probleem rondom gamificatie bij enterprise applicaties. Daarbij is de volgende hoofdvraag opgesteld om in de 6 maanden binnen het afstudeerproces onderzoek naar te verrichten.

Hoe kan een herbruikbare, onafhankelijke gamificatieservice worden opgezet zodat deze toegepast kan worden binnen enterprise applicaties?

De hoofdvraag kent enkele begrippen die van belang zijn voor de opdracht. Om inzicht te geven waarom worden de begrippen kort toegelicht.

- Herbruikbaar
  - Een softwareservice die wellicht in de toekomst door meerdere klanten wordt afgenomen dient herbruikbaar te zijn. Herbruikbaar in de zin dat hij door de ene klant wellicht anders wordt ingezet dan de andere klant, maar daarmee het doel niet verliest: gamificatie.
- Onafhankelijk
  - Onafhankelijk is een must voor een softwareservice die gebruikt dient te worden in een ander applicatie landschap. Het kan niet zijn dat een bestaande applicatie afhankelijk wordt van de gamificatieservice, de service dient altijd een aanvulling te zijn.
- Gamificatieservice
  - In de context van de opdracht is het doel gamificatie gemakkelijker in kunnen zetten. Daarbij wordt de service zo opgesteld dat verschillende vormen van gamificatie kunnen worden toegepast, liggend aan de wensen en eisen van de mogelijke klant.
- Enterprise applicaties
  - Het is belangrijk deze afbakening mee te nemen. Tijdens het overleg met de opdrachtgever werd duidelijk dat de gamificatieservice in eerste instantie moet kunnen worden ingezet bij enterprise applicaties. Dit omdat het meer aansluit bij de klantenkring van Info Support.

Om antwoord te geven op de hoofdvraag zijn vijf deelvragen tot stand gekomen:

1. Wat zijn kenmerkende eigenschappen van enterprise applicaties?
2. Hoe kan gamificatie als onafhankelijke service worden aangeboden aan enterprise applicaties?
3. Welke gamificatie elementen zou een gamificatie service voor enterprise applicaties moeten omvatten?
4. Hoe kunnen generieke spelregels worden opgesteld voor een gamificatieservice?
5. Uit welke softwarecomponenten moet een gamificatieservice bestaan?

Zodra er een antwoord is op deze vijf vragen zou een herbruikbare, onafhankelijke gamificatieservice moeten kunnen worden opgezet welke afgenomen kan worden door enterprise applicaties.

3 Methodiek

3.1 Onderzoek

Om de hoofdvraag en deelvragen te kunnen beantwoorden waren twee vormen van onderzoek nodig. Er is eerst gekeken naar wat er bekend is over een bepaald vraagstuk door een deel theoretisch onderzoek. Vervolgens is de uitkomst getoetst op een praktische wijze wanneer de theorie niet afdoende was om een conclusie uit op te stellen.

3.1.1 Theoretisch

Onder de theoretische vorm van onderzoek verstaan we in de context van dit onderzoek voornamelijk:

Exploratief onderzoek	Het verkennen van de huidige stand van zaken. De noodzaak hiervoor was om ervoor te zorgen dat er bekend was welke kant het onderzoek op kon gaan om daar vervolgens besluiten in te maken. Het opdoen van ideeën en verkennen van het onderzoeksgebied. De zogenaamde bronnen voor dit onderzoek zijn geweest: - Literatuur - Experts
Descriptief/beschrijvend onderzoek	Volgend op exploratief onderzoek zijn enkele zaken in kaart gebracht. Daarbij zijn terminologieën beschreven.

3.1.2 Praktisch

Onder de praktische vorm van onderzoek verstaan we in de context van dit onderzoek voornamelijk:

Kwalitatief onderzoek	Semigestructureerd interview met drie architecten. Hierbij was het doel van het interview in het moment van tijd duidelijk, de vorm was niet van tevoren opgesteld. Er kan gesproken worden van een diepte-interview.
Inventarisatieonderzoek	Exploratief onderzoek was nodig om kennis te verkrijgen over verschillende gebieden rondom enterprise applicaties, softwaretechnologie en gamificatie.
Toegepast onderzoek	Het toegepast onderzoek is uitgevoerd doormiddel van het uitwerken van een proof-of-concept. Dit was nodig om conclusies te kunnen trekken en daar aanbevelingen op te doen

3.1.3 Toegepast onderzoek

Het toegepast onderzoek vergt meer toelichting. Het gaat om het proof-of-concept (POC). Om in een gelimiteerde tijd de functionaliteit van het concept “gamificatie service” aan te tonen is gekozen om aan het einde van het proces een ‘Minimum Viable Product’ (MVP) op te leveren. Een MVP is als volgt:

“In product development, the minimum viable product (MVP) is a product which has just enough features to gather validated learning about the product and its continued development. Gathering insights from an MVP is often less expensive than using a product with more features which increase costs and risk in the case where the product fails, for example due to incorrect assumptions.” (Wikipedia, 2016)

Hier is voor gekozen omdat dit in een vroeg stadium aangeeft dat het product levensvatbaar is. Het toont immers aan dat wanneer de kerndoelstellingen worden opgelost door het product, het volledige product uiteindelijk het probleem volledig zou kunnen oplossen.

Bij het maken van een MVP is het belangrijk om het proces van "voor tot achter" uit te werken. Van voor tot achter houdt in dat de volledige kern functionaliteiten van de service zijn gerealiseerd.

3.2 Verificatie en Validatie

"Verificatie: men controleert of het systeem juist is gemaakt (zijn alle eisen verwerkt?)
Validatie: men controleert of het juiste systeem is gemaakt (doet het wat de klant wil?)"

(Wikipedia, 2015)

Het proces van verificatie en validatie is overlegd en overeengestemd met de opdrachtgever. De volgende onderdelen zijn meegenomen in het proces rondom verificatie en validatie:

Het concept van de service	Verificatie
De functionaliteit en bruikbaarheid van de service	Verificatie en validatie
De kwaliteit van de service	Validatie

Het belang van de correctheid van de service ligt bij de volgende twee groepen belanghebbende:

- **Productowner**; wil dat het onderzoek aantoont dat er een hoogwaardige gamificatie service kan worden ontwikkeld, welke in te zetten is voor verscheidene enterprise applicaties.
- **Architecten**; de architecten zijn uiteindelijk bepalend of de service ingezet kan worden in de enterprise applicatie. Men kan ze in principe zien als klant van de gamificatie service. Wanneer de klant het product niet kan of wil gebruiken, heeft het geen meerwaarde.

Hier volgt een korte toelichting van de drie hierboven benoemde punten waarop gevalideerd wordt.

3.2.1 Concept

Het concept van de service omvat wat de service doet, hoe de service gebruikt kan worden en hoe de service beheerd wordt. Dit wordt getoetst aan de hand van de functionaliteit van de service samen met de architecten. De architecten zijn mogelijke toekomstige afnemers van het product, waarmee het van groot belang is dat het product correct en inzetbaar is.

Waar ze naar kijken is:

- Doet de gamificatie service wat ik verwacht van een gamificatie service?
- Ga ik akkoord met de manier van implementatie binnen mijn enterprise applicatie?

3.2.2 Functionaliteit en bruikbaarheid

Functionaliteit:
Functionaliteit omvat in de context beide aspecten: “zijn alle eisen verwerkt?” en “doet het wat de klant wil?”. Bij functionaliteit gaat het om wat er gedaan kan worden met het systeem en waar dat toe leidt.

Bruikbaarheid:
Het succes van een service zit hem deels in de bruikbaarheid van de service. Een service moet laagdrempelig zijn wanneer het komt op beginnersgemak. De service moet begrijpelijk zijn voor nieuwe gebruikers, zodat er weinig weerstand is de service te gebruiken en te blijven gebruiken.

De vorm van validatie voor functionaliteit en bruikbaarheid zijn schermontwerpen.
De schermontwerpen zijn getoetst met de opdrachtgever. Waarbij gekeken is naar design keuzes en of alle gewenste functionaliteiten wel gedekt zijn als de service gemaakt zou worden aan de hand van de schermontwerpen.

3.2.3 Kwaliteit

Om de kwaliteit van de service te kunnen waarborgen wordt er automatisch getest. Er is gekozen om dit te doen op basis van “specification by example”. Daarmee wordt niet enkel de code getest op correctheid, maar ook of de code doet wat er functioneel verwacht wordt van het systeem. Om dit te valideren is gebruik gemaakt van SpecFlow – Gherkin. De manier van werken met SpecFlow is dat je business requirements koppelt aan de source code. Je zegt in een bepaalde syntax wat je verwacht van het systeem, en dit levert een vorm van testen op waarbij gevalideerd wordt of dit klopt.

4 Onderzoeksresultaten

4.1 Deelvraag 1: Enterprise applicaties

Wat zijn kenmerkende eigenschappen van enterprise applicaties?

4.1.1 Aanpak

Om antwoord te kunnen geven op deze vraag is er gekozen voor de volgende onderzoeksmethoden:

- Exploratief onderzoek
- Descriptief/beschrijvend onderzoek

Om tot een uitkomst te komen is gekeken naar literatuur en bronnen die te maken hebben met de algehele vorm van enterprise applicaties. Uit deze bronnen zijn de volgende uitkomsten gekomen:

4.1.2 Uitkomsten

Vaak is een enterprise applicatie een bedrijfsapplicatie van grotere bedrijven. Bij het zoeken naar definities en overeenkomsten bij verschillende enterprise applicaties kan het volgende wellicht gezegd worden:

- De termen enterprise applicatie en enterprise software worden door elkaar heen gebruikt en lijken op hetzelfde neer te komen.
- De enterprise applicatie heeft een toegevoegde waarde voor een organisatie. Niet zozeer voor consumenten of individuele gebruikers.
- De service die de enterprise applicatie verleend zijn vaak typisch bedrijf georiënteerd. Daarbij kan men denken aan iets als een online winkel, facturaties, projectmanagement, etc. Het zijn services die overeenkomen met bedrijfsmatige processen.

Martin Fowler zegt over enterprise applicaties het volgende:
"Enterprise applications are about the display, manipulation, and storage of large amounts of often complex data and the support or automation of business processes with that data." (*Fowler, 2011*)

Dit sluit aan bij de punten hierboven. De algemene omschrijving van een enterprise applicatie lijkt hierin redelijk duidelijk. Karakteristieken beschrijven blijkt echter een ingewikkelder verhaal.

Op de developmentwebsite van Microsoft vind je een stuk beschrijving als volgt:
"In today's corporate environment, enterprise applications are complex, scalable, distributed, component-based, and mission-critical. They may be deployed on a variety of platforms across corporate networks, intranets, or the Internet. They are data-centric, user-friendly, and must meet stringent requirements for security, administration, and maintenance. In short, they are highly complex systems." (*Microsoft, sd*)

Hierbij komen enkele termen naar boven.

- Complex, schaalbaar en component gedreven.

Er is geen algemeen geaccepteerde lijst met “enterprise software karakteristieken”. Wel zijn eisen die veel voorkomen wanneer men spreekt over enterprise applicaties en waar deze aan dienen te voldoen. Dit zegt wat over de verwachtingen die men heeft aan enterprise applicaties. Deze eisen zijn:

- Performance
- Betrouwbaarheid
- Security
- Beheerbaarheid
- Overdraagbaarheid

Deze eisen komen ook voort in de ISO IEC 25010 standaard welke gaat over softwarekwaliteit. Deze standaard wordt opgesteld door de volgende organisaties:
“ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization.”
(International organization for standardization, 2011)

De eisen die men stelt aan enterprise applicaties komen overeen met kwaliteiten die opgesteld zijn in deze standaard. Daarbij staan in het document alle relaties tussen de kwaliteiten en hoe ze verband houden. Voor de diepte dient de standaard geraadpleegd te worden, echter voor een algehele indruk hieronder een beschrijving van de eisen volgens de ISO IEC 25010 standaard. Daarbij zijn de termen in het Engels behouden volgens het document om ze te kunnen vinden, de uitleg is in het Nederlands voor de context van dit document. Niet alle subcategorieën worden meegenomen in de beschrijving, enkel degene die toevoegen en bijdrage aan het huidige onderzoek.

Performance efficiency
Performance relatief tot de hoeveelheid gebruikte resources onder vastgesteld condities. Daarbij gaat het er dus om hoe efficiënt de performance is geregeld. Een systeem kan wel goed performen wanneer 1000 servers constant beschikbaar zijn om 50 klanten aan te kunnen, echter is dit niet efficiënt. Daarbij is er dus een ratio tussen uitkomst en gebruikte middelen.

- <u>Time behaviour</u>
  Hoe het systeem om gaat met input, tijd om het te verwerken en de responsetijd die eruit leidt. Dit dient te voldoen aan de eisen die vooraf gesteld zijn aan het systeem.
- <u>Capacity</u>
  Hierbij wordt gekeken naar wanneer een product of systeem tegen de limieten loopt en of het daarbij nog voldoet aan de eisen.

Reliability
De betrouwbaarheid van een systeem kan worden gemeten aan de hand van hoe het systeem functioneert onder bepaalde condities in een bepaalde tijd. Daar kunnen bijvoorbeeld eisen aan worden opgesteld als “de service moet altijd tussen 08:00 en 18:00 bereikbaar zijn”. En hoe dient er omgegaan te worden als het systeem daar niet aan voldoet.

- <u>Maturity</u>
  In hoeverre voldoet het systeem qua betrouwbaarheid in relatie tot de eisen aan het systeem.
- <u>Fault tolerance</u>
  In hoeverre het systeem functioneert naar behoren, zelfs als er hard- of software fouten voorkomen in het systeem. Denk daarbij aan een deadlock van een database of een server die onbereikbaar is.

Security
Bij veiligheid van een systeem wordt er gekeken of personen of andere producten of systemen de juiste toegang hebben tot het systeem. Dit om informatie en data te beschermen.

- Confidentiality
  Specifiek over dat de data in het systeem enkel toegankelijk is voor de entiteiten die er toegang tot mogen hebben.
- Accountability
  In hoeverre de handelingen gedaan op delen van het systeem terug te vinden zijn tot de entiteit die ze heeft uitgevoerd. Er kan hier gedacht worden aan logging.

Maintainability
Beheerbaarheid van een systeem is belangrijk om ervoor te zorgen dat het effectief en efficiënt kan worden aangepast door de beheerders van het systeem.

- Modularity
  In hoeverre het systeem in verschillende specifieke onderdelen is opgedeeld. Dit zorgt ervoor dat een verandering in een onderdeel geen of minimale impact heeft op andere onderdelen.
- Reusability
  Wanneer een onderdeel herbruikbaar is kan het gebruikt worden in meerdere facetten van het systeem. Hierdoor hoeft het onderdeel niet opnieuw geschreven te worden of op twee plekken te worden onderhouden.

Portability
Omvat hoe effectief en efficient een systeem, product of component kan worden overgezet van een locatie naar een andere locatie. Dit gaat over zowel software als hardware.

- Adaptability
  In hoeverre een product of systeem op vooruitgang kan reageren van hardware, software of andere omstandigheden. Hieronder valt ook schaalbaarheid waarbij het product of systeem wellicht op meerdere systemen moet kunnen draaien om de performance efficiency of reliability te ondersteunen.

4.1.3 Conclusie

Het concept enterprise applicatie heeft geen eenduidige definitie. Daarbij is het belangrijker dat een enterprise altijd zal (moeten) voldoen aan enkele kwaliteitseisen en standaarden. Het is belangrijk om te weten welke deze zijn en hoe daarmee om te gaan.

In relatie tot een softwareservice zal gekeken moeten worden om de eigenschappen performance, betrouwbaarheid, security, beheerbaarheid en overdraagbaarheid. Deze moeten mee worden genomen in de ontwikkeling van een service die kan worden ingezet binnen enterprise applicaties. Door dit te doen wordt er een soort garantie gegeven over de kwaliteit van de software, wat van hoog belang is binnen enterprise applicatie landschappen.

4.2 *Deelvraag 2: Onafhankelijke service*

Hoe kan gamificatie als onafhankelijke service worden aangeboden aan enterprise applicaties?

4.2.1 Aanpak

Om antwoord te kunnen geven op deze vraag is er gekozen voor de volgende onderzoeksmethoden:

- Kwalitatief onderzoek
- Exploratief onderzoek
- Inventarisatieonderzoek
- Vergelijkend onderzoek

De volgende gebieden waren nodig om te onderzoeken om antwoord te kunnen geven op de vraag:

- Welke techniek kan worden gebruikt voor het maken van een onafhankelijke service?
- Waar moet de service aan voldoen om onafhankelijk te worden geacht?

Voor het kwalitatief-, exploratief- en inventarisatieonderzoek hebben interviews met drie architecten plaatsgevonden. Deze interviews hebben zijn op twee momenten in de tijd geweest. Het eerste interview was om een indruk te krijgen van het huidige applicatielandschap en in wat voor vorm daarmee gecommuniceerd zou kunnen worden om op basis van die inventarisatie een implementatiekeuze te maken. Het tweede interview is geweest om de gemaakte keuzes te toetsen en te valideren.

Deze interviews zijn afgenomen met drie softwarearchitecten. Dit zijn architecten die in de toekomst mogelijk de gamificatie service zouden kunnen afnemen. Waarbij de mogelijke afname van de service afhankelijk is van hoe zij de kwaliteit van die service inschatten.

Naam	*Omschrijving*
W. Denekamp	Software Architect bij Info Support bv. *Ontwerpt en ontwikkelt mee aan een online kennisdelingplatform voor alle medewerkers van Info Support.*
X. Buffart	Software Architect bij De Vries bv. *Ontwerpt aan een enterprise applicatie wat een groot aantal medewerkersplanning gerelateerde zaken verwerkt.*
R. Brookman	Software Architect bij Info Support bv. *Ontwerpt administratieve enterprise applicaties.*

Tabel 1: Software architecten

Gezien de verscheidenheid van de drie architecten is het een goede representatieve groep om een service te kunnen toetsen op onafhankelijkheid en kwaliteit. Zodra de service door alle drie de architecten te gebruiken zou zijn, zal het voor de meeste gevallen zo zijn.

4.2.2 Uitkomsten

Onafhankelijk (bijvoeglijk naamwoord, bijwoord)

1. Vrij, zelfstandig
2. Door niets bepaald of geregeld

(Van Dale, sd)

Een belangrijk aspect van de onafhankelijkheid is dat de gamificatie oplossing werkt vanuit de wensen van de klant. De klant wil de oplossing gebruiken op zijn manier. De productowner wil uiteraard dat de service naar behoeve functioneert.
De onafhankelijkheid die we terug willen vinden in de service is zoals beschreven in de tweede definitie: door niets bepaald of geregeld. Daarbij kan de service een eigen invulling krijgen per klant terwijl het nog steeds de bedoelde functionaliteiten bevat.
Daarnaast is er nog een aspect dat de enterprise applicatie die gebruikt maakt van de service, niet afhankelijk wordt van de service. Mocht de service immers tijdelijk onbereikbaar zijn, moet de business gewoon door. Het is een toevoeging, geen vereiste.

In het eerste gesprek met de architecten lag de nadruk op huidige situatie en een mogelijke invulling van een service in het algemeen. Daarbij kwamen enkele punten naar boven:

- De wens om zo min mogelijk beheer te hebben over de service. De service heeft een bepaald doel, een doel wat niet in het huidige business domein valt en daardoor veelal onbekend terrein is.
- De eigen applicatie mag geen last ondervinden als de service eruit ligt.
- De gamificatielogica van de service moet volledig aan de service kant zijn, er mag geen logica in de eigen applicatie terugkomen.
- Er mogen geen klantgevoelige herleidbare gegevens worden opgeslagen aan de servicekant.
- Het systeem zou weleens 1000 actieve gebruikers aan moeten kunnen en mag dan niet omvallen.

In deze eisen/wensen zien we eigenlijk de kwaliteitscriteria terugkomen die zijn besproken in hoofdstuk 4.1 Deelvraag 1: Enterprise applicaties. Dit waren performance, betrouwbaarheid security, beheerbaarheid en overdraagbaarheid.

Allereerst is gekeken naar een techniek om de service te kunnen bewerkstelligen met deze eisen in gedachten. Daarbij is een afweging gemaakt tussen een webservice hosten of een pakket aanleveren binnen de eigen organisatie.

Een tabel overgenomen uit het artikel “Delivering software as a service” op de volgende pagina geeft een indicatie van verschillen in kosten van het aanbieden van software als een service versus de software in eigen context in beheer nemen.

TOTAL COST OF OWNERSHIP, $ THOUSAND

	Software on premises	Software as a service	Sources of savings with SAAS
IMPLEMENTATION, DEPLOYMENT Customization, Integration	108	72	Reduced deployment time, limited customization, self-service through on-boarding scripts
Basic Infrastructure Testing, Deployment	54	0	Does not require infrastructure and application testing
Application Infrastructure Testing, Deployment	30	0	
ONGOING OPERATIONS Training	101	35	Lowers training requirements
Management, Customization Of Business Process Change	94	0	Does not require ongoing business process change management.
Data Center Facilities Rental, Operations; Security, Compliance; Monitoring Of Incident Resolution	750	0	Includes vendor's costs to serve in subscription price
SOFTWARE User Licenses, Subscriptions; Maintenance	480	1500	
OTHER Unscheduled Downtime	308	0	Provides 99,9% general-server availability vs 99% Reduces unused licenses by 20%, users added as needed
Unused Licenses	92	0	
Total costs	2298	1640	

Vergelijking 1: Software on premises vs Software as a service

(Dubey & Wagle, 2007)

Waar in de tabel in eerste instantie de focus ligt op kosten valt er ook iets te zeggen over de werkzaamheden die verricht moeten worden bij de twee vormen van software aanbieden.
Als er naar deze vergelijkingen wordt gekeken scoort SAAS hoger. Er is gekozen om SAAS te adviseren ten opzichte van een pakket in eigen context. De software architecten waren akkoord met dit voorstel.
Om een SAAS-applicatie aan te bieden dient er gecommuniceerd te kunnen worden met de externe applicatie. Communicatie richting de service gebeurt via HTTP, waarbij we spreken over een webservice.
"Web services technology is the most promising choice to implement service oriented architecture and its strategic objectives. A Web service is essentially a semantically well-defined abstraction of a set of computational or physical activities involving a number of resources, intended to fulfill a customer need or a business requirement. A Web service could be described, advertised and discovered using standard-based languages, and interacted through Internet-based protocols. Today, two types of Web services are most popular and widely used: SOAP-based Web services and RESTful Web services"

(Sheng, et al., 2014)

Daarbij is gekozen om een afweging te maken tussen de SOAP gebaseerde webservicetechnologie en een RESTful webservice.

4.2.2.1 SOAP

SOAP is een protocol dat wordt gebruikt voor communicatie tussen verschillende componenten. Soap handelt communicatie af via XML-berichten: request en response. De communicatiemogelijkheden voor deze berichten zijn SMTP, HTTP, TCP, UDP of JMS.
Het SOAP-protocol is op XML gebaseerd en bestaat uit drie onderdelen

- Een envelop dat een framework definieert voor het beschrijven van wat in een bericht staat en hoe het te verwerken.
- Een set van encodeerregels voor de expressie van 'instances' van applicatiegedefinieerde datatypen.
- Een conventie voor de representatie van 'remote procedure calls' en antwoorden. SOAP kan in potentie worden gebruikt in combinatie met een grote verscheidenheid aan andere protocollen. (Microsoft, 2007)

Performance:
XML herhaalt elk element en attribuut voor elke instantie van een element of attribuut. Bij veel data zal er dus een groot XML bestand verstuurd worden. Ook maakt XML gebruik van externe referenties wat kan leiden tot performance verlies, het is namelijk afhankelijk van de server waarvan gedownload moet worden.

Overdraagbaarheid / Schaalbaarheid:
SOAP kan beperkt gebruik maken van het HTTP protocol, enkel POST wordt ondersteund. Daarmee is de implementatie mogelijkheid beperkt.

Beheerbaarheid:
De complexiteit van het definiëren van de regels is redelijk hoog.

Security:
SOAP heeft een uitgebreid veiligheidsprotocol genaamd WS-security.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <SOAP-ENV:Body>
  - <FunctionName>
      <Parameter1 xsi:type="xsd:string">value</Parameter1>
      <Parameter2 xsi:type="xsd:int">value</Parameter2>
      ...
      <Parametern xsi:type="xsd:string">value</Parametern>
    </FunctionName>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figuur 6 : Voorbeeld SOAP XML syntax

"WS-Security allows for a SOAP message to identify the caller, sign the message, and encrypt message contents. Whenever possible, existing specifications are reused to reduce the amount of invention required to securely deliver a SOAP message. Because all of the information is delivered within the message itself, the message becomes transport neutral. The message would be secure if it was delivered by HTTP, e-mail, or on CD-ROM." (Seely, 2002)

4.2.2.2 REST

Representional State Transfer (REST) is een software-webarchitectuur voor gedistribueerde mediasystemen. Een voorbeeld van een gedistribueerd mediasysteem is het internet.
"REST stands for Representational State Transfer. (It is sometimes spelled "ReST".) It relies on a stateless, client-server, cacheable communications protocol -- and in virtually all cases, the HTTP protocol is used.
REST is an architecture style for designing networked applications. The idea is that, rather than using complex mechanisms such as CORBA, RPC or SOAP to connect between machines, simple HTTP is used to make calls between machines."
(Elkstein, 2008)

Het aanspreken van deze service wordt gedaan met behulp van een HTTP-request. Binnen een RESTful-webservice zijn de volgende HTTP-requestmethoden beschikbaar:

- GET
- POST
- PUT
- DELETE

Performance:
Waarbij SOAP performance verlies heeft door de XML notatie, kan een RESTful webservice gebruik maken van JSON (Javascript Object Notation). Het is een zeer compacte manier van datanotatie, wat ook nog eens goed leesbaar is.

Overdraagbaarheid / Schaalbaarheid:
De limitatie van SOAP is niet van toepassing op RESTful. Naast POST worden GET, PUT en DELETE ook ondersteund.

Beheerbaarheid
Naast compact is de data annotatie mogelijkheid van JSON ook goed leesbaar. Waar vroeger wellicht nog niet veel tools waren, ondersteunen de moderne browsers nu zelfs het uitleven van JSON data. Dit maakt het overzichtelijk voor developers aan zowel de front-end als de back-end.

Security:
REST heeft geen eenduidige security. Daarbij is het aan de developer om zich te houden aan de best practices. De OWASP-organisatie beschrijft deze best practices op authenticatie, autorisatie, input validatie, output encodatie, versleuteling en bericht integriteit. (OWASP, 2016)

Example Response

```
Status: 200 OK

{
  "groups": [
    {
      "name":       "DJs",
      "created_at": "2009-05-13T00:07:08Z",
      "updated_at": "2011-07-22T00:11:12Z",
      "id":         211
    },
    {
      "name":       "MCs",
      "created_at": "2009-08-26T00:07:08Z",
      "updated_at": "2010-05-13T00:07:08Z",
      "id":         122
    }
  ]
}
```

Figuur 7 : Voorbeeld JSON Object

4.2.2.3 Vergelijking REST vs SOAP

In de vergelijking wordt geen goed vs slecht gedaan, er is uitgegaan van een “scoort hoger – scoort lager” om vervolgens een keuze te kunnen maken voor welke vorm wordt gekozen. In de vergelijking worden in ieder geval de genoemde criteria opgenomen, verder bevat de tabel ook informatie van een andere bron (benoemd onder de tabel).

	REST	***SOAP***
Performance	▲	▼
Betrouwbaarheid	▼	▲
Security	▼	▲
Beheerbaarheid	▲	▼
Overdraagbaarheid	▲	▼
Gebruik	▲	▼
Fout rapportage	▲	▼
Beschrijvend	▲	▼

(Motamarri, 2013)

Vergelijking 2 : REST vs SOAP

4.2.2.4 Architect gesprekken implementatie keuze

Na het in kaart brengen van de voor en nadelen van REST versus SOAP komt REST beter uit de vergelijking. Daarbij is in het tweede gesprek met de architecten geadviseerd om gebruik te maken van REST. De architecten zijn akkoord gegaan met het advies. Verder is de implementatiekeuze voorgesteld ter verificatie van mogelijke afname, ook daar is twee keer akkoord op gekomen. Eén van de architecten had het helaas te druk voor het tweede gesprek.

4.2.3 Conclusie

REST lijkt beter aan te sluiten bij de wensen en eisen van de software architecten. Daarbij is het advies om een REST webservice te realiseren in de context van een gamificatieservice proof-of-concept. De architecten gaan akkoord met dit advies.

4.3 *Deelvraag 3: Gamificatie elementen*

Welke gamificatie elementen zou een gamificatie service voor enterprise applicaties moeten omvatten?

4.3.1 Aanpak

Om antwoord te kunnen geven op deze vraag is er gekozen voor de volgende onderzoeksmethoden:

- Exploratief onderzoek
- Inventarisatieonderzoek

Door eerst te inventariseren welke gamificatie elementen er bestaan en wat ze omvatten kan er exploratief worden gekeken naar de mogelijke inzet. Uit verschillende iteraties is gebleken welke elementen goed inzetbaar zijn voor de bedachte doelen van gamificatie bij enterprise applicaties. Dit is gedaan door de opdrachtnemer, opdrachtgever, architecten en enkele collega´s werkzaam bij Info Support als software engineer of requirement engineer.

4.3.2 Uitkomsten

Gamificatie is relatief nieuw en wordt nog lang niet overal ingezet. Gamificatie inzetten kan op veel manieren, met verschillende doelen. Om gewenste resultaten uit gamificatie te halen dien je daarom ook de juiste elementen in te zetten.
Om een beeld te krijgen van enkele gamificatie elementen, de volgende tabel:

General	**Schedules**	**Socialiser**	**Disruptor**
On-boarding / Tutorials Signposting Loss Aversion Progress / Feedback Theme Narrative / Story Curiosity / Mystery Box Time Pressure	Random Rewards Fixed Reward Schedule Time Dependent Rewards	Guilds / Teams Social Network Social Status Social Discovery Social Pressure Competition	Innovation Platform Voting / Voice Development Tools Anonymity Light Touch Anarchy
Free Spirit	**Achiever**	**Philanthropists**	**Player**
Exploration Branching Choices Easter Eggs Unlockable / Rare Content Creativity Tools Customisation	Challenges Certificates Learning / New Skills Quests Levels / Progression Boss Battles	Meaning / Purpose Care-taking Access Collect & Trade Gifting / Sharing Sharing Knowledge	Points / Experience Points (XP) Physical Rewards / Prizes Leaderboards / Ladders Badges / Achievements Virtual Economy Lottery / Game of Chance

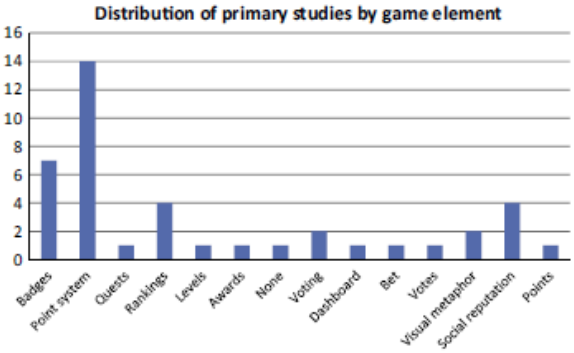
Tabel 2: 47 gamificatie elementen volgens Marczewski

(Marczewski, 47 Gamification elements, mechanics and ideas, 2015)

Allereerst moet de term “gamificatie element” duidelijk zijn binnen de context. Voor het onderzoek is gekeken naar het middel en niet het doel. Onder doel kunnen we verstaan waarom we gamificatie willen inzetten. Een manager kan willen dat zijn klanten vaker terugkomen. Het doel is dan “klantenbinding” of “betrokkenheid van klanten” of iets in die trant. Om dit te bereiken zal hij een middel gebruiken zoals punten, sociale druk, cadeautjes, et cetera.

Na het inventariseren van enkele gamificatieelementen is gekeken naar het gebruik van enkele van deze elementen. In het boek “Gamify” van Brian Burke wordt geschreven:
Game mechanics describe the key elements that are common to many games, such as points, badges, and leaderboards. (Burke, 2014)

Dit zien we terug in het artikel "*Gamification in software engineering – A systematic mapping*" waarin het volgende figuur laat zien hoe vaak verschillende elementen zijn gebruikt.



Figuur 8 : Gebruik gamificatie elementen

(Pedreira, García, Brisaboa, & Piattini, 2015)

Van hoog naar laag komt daar het volgende uit:

14	Point system
7	Badges
4	Rankings; Social reputation
2	Voting; Visual metaphor
1	Quests; Levels; Awards; None; Dashboard; Bet; Votes; Points

Daarin is vervolgens een inclusie-exclusie methodiek toegepast om te bepalen welke gamificatie elementen passen in de huidige context. Deze criteria zijn tot stand gekomen in overleg met de product owner op de gebieden: zijn ze haalbaar in de tijd, zijn ze duidelijk communiceerbaar en leiden ze tot voldoende diversiteit in elementen?

Inclusiecriteria:

- De gekozen elementen moeten een laag instapniveau hebben qua inzet.
- De gekozen elementen moeten snel te begrijpen zijn bij een publiek wat nog weinig ervaring heeft met gamificatie.
- Het element moet generiek zijn, waarbij het inzetbaar is in verscheidenen enterprise applicaties.

Exclusiecriteria:

- Het element mag niet afhankelijk zijn van een ander gamificatieelement.
- Het element mag geen implementatiekeuze zijn.

Voor het POC is samen met de opdrachtgever besloten om drie gamificatie elementen te verwerken. Na de opname van deze inclusie- en exclusiecriteria is gekozen voor de volgende drie elementen:

Type	*Beschrijving*
Punten	Een gebruiker behaalt punten, een score waarmee bijvoorbeeld aangegeven wordt hoe actief een gebruiker is of hoe goed hij zijn taken uitvoert.
Levels	Een level toont aan hoe ver een gebruiker is in het systeem. Hoe hoger het level, hoe meer activiteit de gebruiker heeft getoond binnen het systeem
Badges	Een badge is een soort "medaille" die gebruikers kunnen krijgen voor het uitvoeren van taken. De grootte van zo'n taak kan verschillen, denk aan: - "meest actieve medewerker in maand x" - "gebruiker heeft 3 tickets afgehandeld" - "gebruiker voor het eerst ingelogd"

De overige gamificatieelementen zijn daarbij afgevallen wegens exclusie.

Element	*Reden exclusie*
Rankings	Ter discussie valt of 'Rankings' een op zichzelf staand gamificatie element is, of dat het hierbij puur gaat om het tonen van resultaten aan gebruikers. Om deze reden is rankings dan ook niet meegenomen als element.
Voting	Voting is enkel van toepassing in software waar gestemd kan worden op content. Dit is een software-specifieke vorm van gamificatie en voldoet daarom in eerste instantie niet aan de genericiteit waaraan de service moet voldoen in de beginstaat.
Social reputation	Social reputation valt wederom in een te specifieke context. Dit werkt vaak bij klant-gedreven applicaties, echter is het bij enterprise applicaties niet op zijn plaats. Minder op het sociale aspect kan "reputation" wel, dit kan echter ook behaald worden met punten, levels en badges.

4.3.3 Conclusie

Punten, levels en badges zullen worden gerealiseerd in het proof-of-concept. Deze lijken uitstekend de lading te dekken van inzetbare gamificatie elementen binnen enterprise applicaties. Daarnaast zijn ze goed behapbaar, zelfs voor gebruikers die weinig tot geen ervaring hebben met gamificatie.

4.4 *Deelvraag 4: Generieke spelregels*

Hoe kunnen generieke spelregels worden opgesteld voor een gamificatieservice?

4.4.1 Aanpak

Om antwoord te kunnen geven op deze vraag is er gekozen voor de volgende onderzoeksmethoden:

- Exploratief onderzoek
- Kwalitatief onderzoek

Er is gekozen voor een combinatie tussen kwalitatief en exploratief onderzoek. De reden hiervoor is dat het een redelijk nieuw domein is waarin weinig onderzoek heeft. Het kwalitatief onderzoek houdt in dat er met derden is overlegd. Vervolgens zijn er in iteraties concepten bedacht en wederom getoetst met derden. Dit omvat het exploratief onderzoek waarin van tevoren niet bekend was wat eruit zou komen.

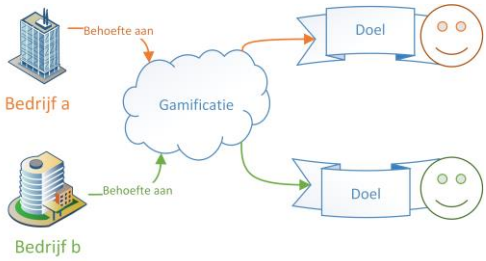
4.4.2 Uitkomsten

Generiek (bijvoeglijk naamwoord) (Van Dale, n.d.)

1. Geslacht-, soort-
2. Voor alle gevallen geldend

Hier is voor het onderzoek vooral uitgegaan van definitie twee: voor alle gevallen geldend.

Binnen de context moet gamificatie namelijk ingezet worden in een theoretische oneindige hoeveelheid softwareapplicatielandschappen. En dat houdt in dat het voor alle gevallen moet passen. Hoe bewerkstelligen we dit? Gamificatie moet voor bedrijf a voorzien in de behoefte aan gamificatie, maar ook bedrijf b. In het volgende figuur zie je hoe dit vorm krijgt.



Figuur 9 : Gamificatie doelen

Beide bedrijven hebben behoefte aan gamificatie. Maar zowel de vorm van “gamificatie” kan verschillen als het doel dat ze voor ogen hebben. Daarom wordt het ook zo belangrijk geacht dat de spelregels generiek kunnen worden opgesteld.

Gamificatiespelregels zijn afhankelijk van een aantal factoren: ‘Acties’ en ‘Effecten’. Een actie is een vorm van interactie dat het bedrijf verwacht in zijn systeem. Een effect is een mogelijk effect dat het bedrijf de gebruiker wil teruggeven. Deze twee zijn nodig om te combineren in ‘Regel’. Een regel kan worden omschreven in een zin zoals:

- Wanneer een gebruiker actie ‘1’ uitvoert, krijgt hij effect ‘2’.

Een concreet hypothetisch voorbeeld:

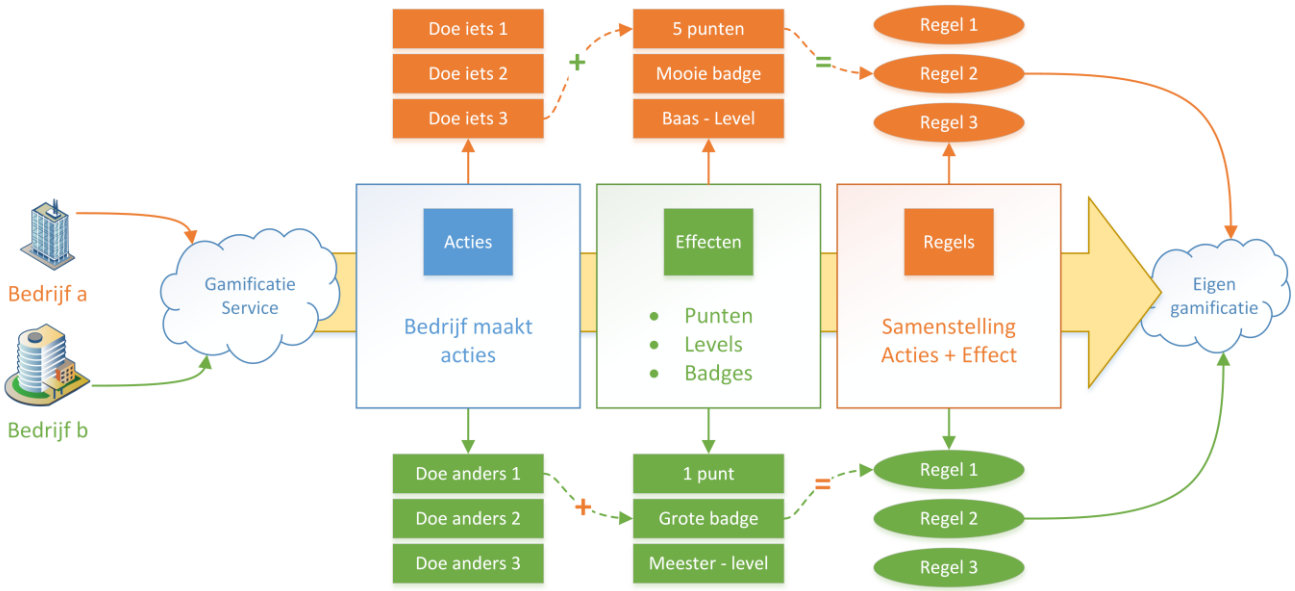
- Wanneer een gebruiker een document upload in het systeem worden er 5 punten bij de gebruiker zijn account toegevoegd.

Om de spelregels generiek te maken is het dus van belang dat het bedrijf volledige vrijheid heeft in:

- Het bepalen welke acties passen bij de bedrijfscontext
- Het bepalen wat voor effecten passen binnen de gebruikersgroep van de software
- Bepalen welke acties gekoppeld kunnen worden aan welke effecten.

Om deze vrijheid te garanderen is er veel focus geweest op “configuratie”. Waarbij een bedrijf dus echt volledige controle heeft over de acties, effecten en regels in gamificatie. Om dit te kunnen bewerkstelligen is uitgegaan van het scenario “er zijn twee bedrijven die dezelfde gamificatieservice inzetten, hoe ziet dat er dan uit en hoe bereiken ze hun eigen variant van gamificatie?”

Hier komt het volgende figuur uit als ideale situatie:



Figuur 10 : Weg naar eigen gamificatie

Hier volgt een voorbeeld om bovenstaande figuur concreet te maken aan de hand van een voorbeeldscenario.

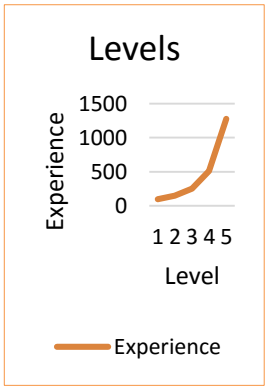
Bedrijf A

Bedrijf A is een school met een “huiswerk-inleversysteem”. De gebruikers van het systeem zijn scholieren die daarop hun huiswerk digitaal inleveren, corrigeren en feedback geven op andermans huiswerk.

Het doel van gamificatie verwerken in het systeem is om studenten met meer plezier het systeem te laten gebruiken waarbij de hoop is dat er meer kwaliteit uit wordt gehaald.

De manager wil levels en badges inzetten. Dat kan er als volgt uitzien:

- Elke keer dat een student zijn huiswerk op tijd inlevert (voor de deadline) krijgt hij 20 experience die bijdragen aan het stijgen in level.
- Wanneer een student 8 keer huiswerk van een ander heeft voorzien van feedback verdient die student een mooie badge.



Hieruit destilleren we het volgende:

Acties	Effecten
Huiswerk inleveren	**Level:** 20 experience
Feedback geven	**Badge**: Mooie badge

Vervolgens worden de regels opgesteld zoals hierboven vermeld.

De resultaten van studenten zijn per klas inzichtelijk. De manager verwacht dat studenten harder hun best zullen doen om het hoogste level en mooie badges te behalen.

Tabel 3 : Voorbeeld experience points voor level

Bedrijf B
Bedrijf B is een bedrijf met een “webshopsysteem”. De gebruikers van het systeem zijn klanten die daarop schoenen kopen en achteraf een beoordeling geven.
Het doel van gamificatie verwerken in het systeem is om klanten meer betrokken te houden met het systeem en wellicht vaker iets te kopen.
De managers wil punten inzetten. Dat kan er als volgt uitzien:

- Elke keer dat een klant iets koopt in de webshop verdient hij punten op basis van het uitgegeven bedrag:

Bedrag eur	Punten
< 10	5
10 – 50	7
> 50	12

- Wanneer een klant een beoordeling geeft ontvangt hij standaard 10 punten.

Hieruit destilleren we het volgende:

Acties	Effecten
Aanschaf laag	**Punten:** 5
Aanschaf midden	**Punten:** 7
Aanschaf hoog	**Punten:** 12
Beoordeling	**Punten:** 10

Vervolgens worden de regels opgesteld zoals hierboven vermeld.
De punten kunnen door klanten worden ingezet om enkele “speciale” producten te kunnen kopen of te gebruiken voor kortingen in de webshop. De manager verwacht dat gebruikers eerder zullen terugkomen om hun punten te besteden aan een nieuwe aankoop en vaker feedback te geven op de gekochte producten.

4.4.3 Conclusie

Doordat het systeem is bedacht vanuit volledige controle over de gamificatie is uitgegaan van de bestaande processen binnen een enterprise applicatie. Om vervolgens de verschillende aspecten van gamificatie (acties, effecten en regels) los in te zetten op deze bedrijfsprocessen maakt dat generieke regels kunnen worden opgesteld die in verscheidenen situaties inzetbaar blijken.

Als het systeem toestaat dat deze apart gemodelleerd kunnen worden is de vraag op:
“Kunnen nu generieke (voor alle gevallen geldend) spelregels worden opgesteld met de gamificatieservice?”

Ja, dit kan met het hierboven beschreven concept.

4.5 *Deelvraag 5: Softwarecomponenten*

Uit welke softwarecomponenten moet een gamificatieservice bestaan?

4.5.1 Aanpak

Om antwoord te kunnen geven op deze vraag is er gekozen voor de volgende onderzoeksmethoden:

- Kwalitatief onderzoek
- Exploratief onderzoek

De uitkomsten van dit onderzoek zijn een soort samenbundeling van gegevens van de hiervoor beschreven deelvragen. Dit komt voort uit het feit dat de gamificatieservice alle bevindingen moet kunnen om- en bevatten.
De vorm van het onderzoek valt onder kwalitatief onderzoek in de zin dat er gedurende iteraties is gekeken naar de mogelijkheden van componenten en waar ze aan voldoen of juist gebreken laten zien. Deze observaties leiden tot conclusies. Deze conclusies zijn aan de hand van eigen expertise en de expertise van collega's die konden meedenken in de context.

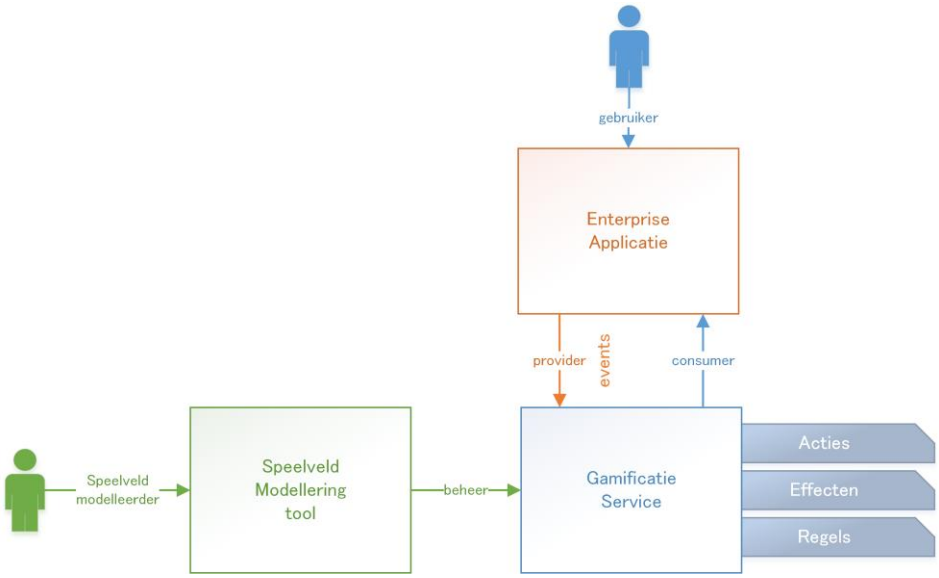
Exploratief is een aanvulling op het hierboven genoemde onderzoek omdat niet bekend was wat de uitkomst kon zijn. Daarbij is door iteraties getracht inzichtelijk te maken waar tegenaan gelopen werd en hoe daarmee verder te gaan.

4.5.2 Uitkomsten

De uitkomsten in dit deel van het onderzoek sluiten voor een deel aan bij de ondervindingen van deelvraag 4.1 Deelvraag 1: Enterprise applicaties
Daarbij is nogmaals gekeken naar de beschreven softwarekwaliteiten die worden gehanteerd binnen de ISO-standaard 25010.

Er is in eerste instantie gekeken naar de eisen. Dit waren op het gebied van performance, betrouwbaarheid, security, beheerbaarheid en overdraagbaarheid.

Op grote lijnen is eruit het onderzoek gekomen dat een gamificatieservice op twee manieren wordt benaderd. Om dit inzichtelijk te krijgen is het volgende model opgesteld.



Figuur 11 : Software componenten

Het model is tot stand gekomen door eerst te kijken welke “fysieke” gebruikers interactie hebben met de gamificatie service. Vervolgens hoe de communicatie tussen de verschillende componenten dan kon worden opgesteld.

Gebruikersinteractie:
Wat beschreven is in het figuur is dat er twee type gebruikers zijn.

- Een gebruiker aan de kant van een enterprise applicatie (EA), deze gebruikt de EA voor het doel waarvoor de EA is gemaakt.
- Daarnaast is er een speelveldmodelleerder, deze is er om het speelveld in te stellen waar de gebruikers van de EA gebruik van maken. De speelveldmodelleerder communiceert middels de speelveldmodelleringtool met de gamificatie service.

Vervolgens is verdieping gekomen over wat de interactie dan wordt voor componenten onderling.

Componenteninteractie:
We zien verschillende lijnen van communicatie lopen in het figuur.

- Enterprise applicatie naar gamificatie service
- Gamificatie service naar enterprise applicatie
- Speelveldmodelleringtool naar gamificatie service

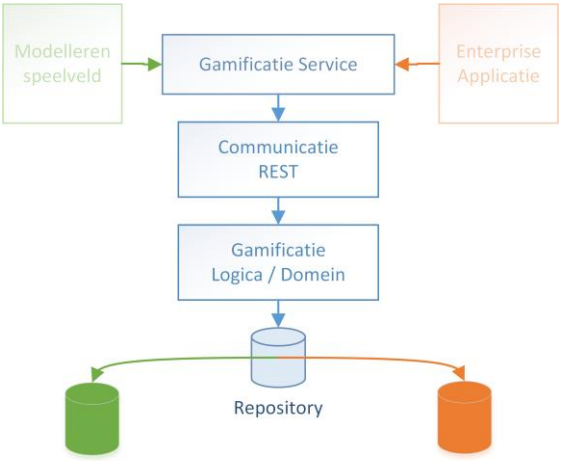
Om deze te kunnen beschrijven de volgende tabel:

	Beschrijving
Enterprise Applicatie ↕ *Gamificatie service*	De enterprise applicatie als **provider** stuurt events naar de gamificatie service, die daar bekend zijn als acties. Voorbeeld: “maak een nieuw artikel aan” als event. De enterprise applicatie als **consumer** vraagt resultaten op van effecten die gekoppeld zijn aan de gebruiker. Voorbeeld: “gebruiker heeft 50 punten verdiend in totaal”
Speelveld Modellering Tool ↓ *Gamificatie service*	De **speelveldmodelleringtool (SMT)** bepaalt hoe de gamificatie eruitziet. Dit kan worden gezien als de configuratie van de gamificatie service. Vanaf de SMT worden acties, effecten en regels gedefinieerd. Dit kun je zien als de configuratie waar de enterprise applicatie uiteindelijk gebruik van maakt.

Tabel 4 : Communicatie vormen tussen componenten

De gamificatie service bestaat op hoog niveau uit enkele onderdelen.

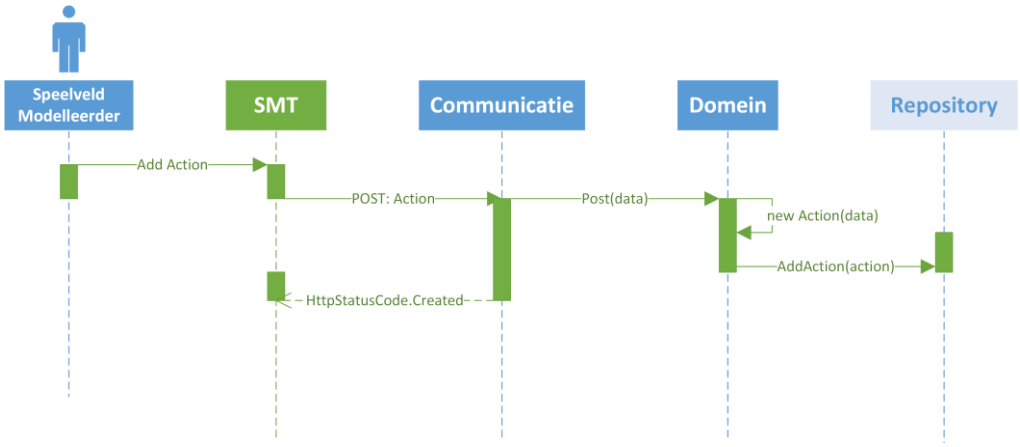
- De communicatielaag (REST)
- De domeinlogicalaag
- De repositorylaag
  - Configuratie gedeelte in een database
  - Resultaten gedeelte in een database



Figuur 12 : software componenten GS

Wanneer er wordt gepraat tegen de gamificatie service komt het dus in eerste instantie binnen bij de communicatielaag. De communicatielaag weet welke functionaliteit moet worden aangeroepen in de domeinlogicalaag.

Om dit in kaart te brengen het volgende sequentiediagram waarbij een speelveldmodelleerder via de speelveldmodelleringtool een nieuwe actie toevoegt.



Figuur 13 : Sequence actie toevoegen

Na deze sequentie is er een nieuwe actie toegevoegd aan de database. Hierboven zien we dat er onderscheid is gemaakt in de 'resultaten' en 'configuratie' in de repositorylogica. Het gaat niet zo zeer om aparte databases in het uitgewerkte proof-of-concept, echter zijn het wel twee redelijk verschillende concepten.

- **Configuratie**: hier worden alle acties, effecten en regels in opgeslagen.
- **Resultaten**: hier worden de gegevens opgeslagen over behaalde effecten bij een gebruiker van de enterprise applicatie.

Onderbouwing componentkeuze

Nadat er is vastgesteld hoe interactie plaats vind is gekeken naar hoe de software kan voldoen aan de opgestelde kwaliteitseisen. Er wordt rekening gehouden worden met de wensen van de architecten, opgesteld in hoofdstuk 4.2.2 Uitkomsten, voor de service. Daarbij is gekozen voor een n-tier architectuur (Wikipedia, 2016) in verband met de aard van zo´n architectuur. Het gebruiken van deze architectuur staat toe dat er goed rekening gehouden kan worden met de opgestelde kwaliteitseisen. De structuur van de software is opgebouwd uit meerdere lagen. De lagen zijn op hoog niveau presentatie (softwaremodelleringtool), verwerking (gamificatieservice) en databeheer (repository). Deze laatste twee lagen vallen zelf binnen een 3-tierarchitectuur waarbij onderscheid is gemaakt tussen communicatie, domein en databeheer.

Beschreven voordelen die behaald kunnen worden door het gebruik van een 3-tier architecture.

Advantages	Disadvantages
Improved Scalability	Complexity of Communication
Enhanced Re-usage	Fewer Tools
Improved Data Integrity	
Enhanced Security	
Reduced Distribution	
Redundant Sever Availability	
Hidden Database Structure	

Tabel 5: Voordelen 3-tier architecture

(Liyanage, 2013)

Deze voordelen worden op de volgende pagina kort toegelicht.

Schaalbaarheid Er kan per component geschaald worden. Hierbij kan worden gedacht aan een hogere belasting van de domeinlaag dan de communicatielaag. Er zou kunnen worden gekozen om de domeinlaag meerdere keren op te zetten met een load-balancer ertussen zodat er altijd een hoge performance wordt behaald.	**Verborgen database structuur** De daadwerkelijke structuur van de database blijft verborgen. Het is aan de communicatielaag om voldoende data te ontvangen en terug te sturen.
Data integriteit Valideren van data kan worden gedaan bij de applicatie en is niet afhankelijk van de verschillende “user interfaces”.	**Security** Een userinterface praat nooit direct met de database. Door de security toe te passen op de communicatielaag kan toegang tot het systeem verder worden voorkomen.
Verminderde distributie Aanpassingen kunnen gedaan worden per component. Er hoeven geen nieuwe versies uitgerold te worden per aanpassing.	**Betrouwbaarheid** Door de verhoogde schaalbaarheid is het mogelijk om een deel van de applicatie meerdere keren te laten draaien. Mocht een van de instanties wegens een fout niet bereikbaar zijn, kunnen andere instanties van het component worden gebruikt.
Herbruikbaarheid Door de communicatielaag kunnen de SMT en de EA onafhankelijk van elkaar van de gamificatieservice gebruik maken.	

Security
Security vergt nodige toelichting bij de implementatiekeuze.
“Protect HTTP methods
Make sure the incoming HTTP method is valid for the session token/API key and associated resource collection, action, and record.” (OWASP, 2016)

Om te voorkomen dat een bedrijf a bij de resultaten van bedrijf b kan, kan er worden gewerkt met een API (application programming interface) key. De enterprise applicatie stuurt altijd dezelfde API key mee, waardoor aan de achterkant autorisatie plaatstvindt. De service geeft enkel resources terug die bij de desbetreffende API key horen.

4.5.3 Conclusie

Het gebruik van een multitierarchitectuur staat toe dat er op een goede manier rekening kan worden gehouden met de opgestelde kwaliteitseisen. Daarbij zijn de verschillende lagen:

- Userinterface (SMT, EA hoeft niet gerealiseerd)
- Communicatie (REST)
- Domein
- Databeheer

Wanneer deze componenten in de context worden gerealiseerd is de basis van een gamificatieservice bewerkstelligd.

5 Proof-of-concept

Alvorens te beginnen is er een scope vastgelegd van het proof-of-concept. Dit is gedaan aan de hand van inclusie en exclusie. De scope is bepaald op basis van tijd, haalbaarheid en minimale functionaliteit van de service.

Inclusie

Voor de volledige werking van een MVP in de context van een gamificatie service is bepaald welke functionaliteiten tenminste in de MVP moesten worden opgenomen. Deze zijn samen met de opdrachtgever en opdrachtnemer bepaald.

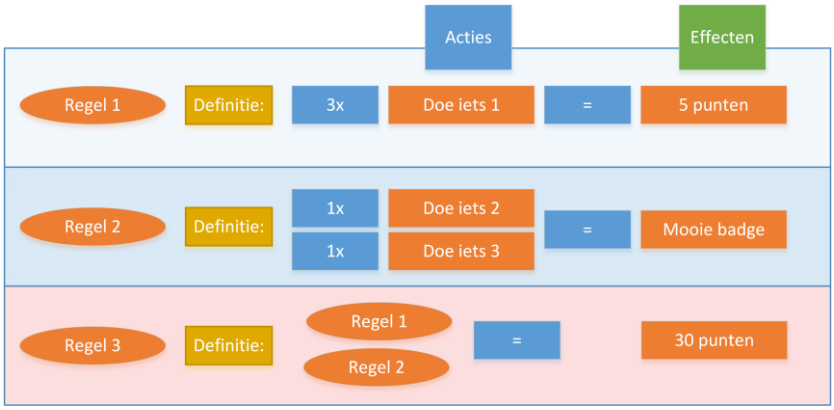
Functie	*CRUD*	*Uitgangspunt*
Er kunnen acties worden beheerd	CRUD	SMT
Er kunnen effecten worden beheerd	CRUD	SMT
Er kunnen regels worden beheerd	CRUD	SMT
Er kunnen acties worden afgevuurd naar de service	-	EA
Er kunnen resultaten worden opgevraagd aan de service	R	EA

CRUD* Create – Read – Update – Delete
SMT Speelveldmodelleringtool
EA Enterprise Applicatie

Exclusie

Enkele functionaliteiten zijn niet uitgewerkt. De belangrijkste zijn:

1. **Samengestelde regels**. De eerste twee 'rijen' die in het figuur hieronder staan worden wel uitgewerkt in de MVP. De derde rij is het voorbeeld van een samengestelde regel. Dit zou zeggen dat wanneer een gebruiker in het systeem heeft voldaan aan regel 1 en regel 2, hij daarmee voldoet aan regel 3 en dus het effect krijgt wat daar voor staat. De complexiteit wordt daar zodanig mee verhoogd dat dit niet is uitgewerkt in de MVP.



Figuur 14 : Voorbeeld gamificatieregels

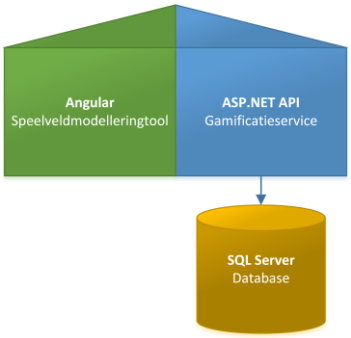
2. **Notificaties**. In het begin is nagedacht over notificaties, dat wanneer een gebruiker een actie uitvoert en daarna voldoet aan een regel met effect als gevolg dat de gebruiker een notificatie krijgt van zijn nieuw ontvangen effect. Ook dit leverde zoveel complexiteit in zowel de architectuur als de implementatie dat hier is gekozen deze niet op te nemen in de MVP.

3. **Replay actions**. Wanneer een regel wordt aangepast, zou de staat van effecten kunnen worden aangepast. In het figuur hierboven heeft 'Regel 1' 3x de actie 'Doe iets 1' nodig om 5 punten op te leveren. Stel een gebruiker voldoet aan die regel en ontvangt 5 punten. Als we nu de regel aanpassen naar 5x 'Doe iets 1' zou de gebruiker er niet langer aan voldoen. Er is nu gekozen om de gebruiker zijn 5 punten te laten behouden voor de oude staat van de regel. Alle acties van de gebruiker worden dus niet opnieuw gevalideerd tegen de nieuwe regel(s).

Verder zal in dit hoofdstuk voornamelijk in grote lijnen worden beschreven waarom gekozen is voor bepaalde technieken en tegen welke problemen is aangelopen en hoe deze zijn opgelost. Voor de werkelijke diepgang van de werking van code wordt er doorverwezen naar de code zelf.

Het proof-of-concept is ondersteunend aan het onderzoek. Daarbij is geprobeerd aan te tonen of wat er uit het onderzoek is gekomen, ook kan worden gerealiseerd. In hoofdlijnen bestaat de gemaakte applicatie uit drie onderdelen:

1. De kern logica van de gamificatieservice is gerealiseerd in ASP.NET
2. De speelveldmodelleringtool is gerealiseerd middels Angular (een Javascript framework)
3. De database is op basis van een Microsoft SQL-server



Figuur 15: POC onderdelen

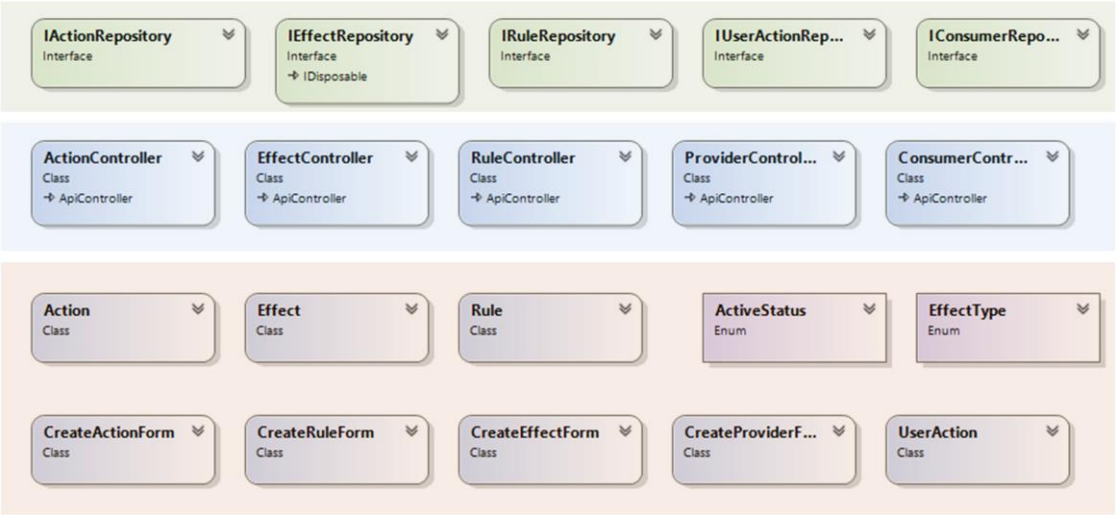
Deze onderdelen zullen in de komende drie subhoofdstukjes aan de orde komen.

5.1 *Back-end: Gamificatieservice*

De gamificatieservice is geschreven in ASP.NET. Het is een API. De functionaliteit van de gamificatieservice in het proof-of-concept is het volgende:

- De HTTP-requests afvangen en verwerken om vervolgens te zorgen dat de logica rondom gamificatie wordt verwerkt en opgeslagen.

Met de wensen en eisen van de architecten aan een service in de context van enterprise applicaties is bij het ontwerpen van de classes gekeken hoe hieraan kan worden voldaan. Daar kwamen drie lagen uit in de verdeling van classes in de back-end.



Figuur 16: Back-end classes

Repository	De repositorylaag wordt in figuur 16 in de eerste lijn weergegeven. De interfaces geven de onderverdeling van verantwoordelijkheden weer in het opslaan van data. Er is gekozen om per controller een repositoryinterface te maken.
Controller	De controllerlaag wordt in figuur 16 in de tweede lijn weergegeven. Alle controllers zijn ApiControllers. *In Web API, a controller is an object that handles HTTP requests.* (Microsoft, 2015)
Model	De modellaag wordt in figuur 16 in de onderste lijn weergegeven. Deze models waren nodig om gamificatie te kunnen realiseren.

De keuze voor deze drie onderdelen is in de basis op de principes 'Low Coupling' en 'High Cohesion'.

- *"Coupling in its very simple form is the degree to which one class is connected to another class. If class is highly coupled, then you are bound to use /edit rest of the connected classes where only editing one class could have served the purpose."*
- *"Another design level example of High cohesion would be of N-Tier architecture. You would want a user interface technology that makes it easy to wire up the business logic and data access layer."*

Deze principes zorgen voor een hoge beheerbaarheid van de applicatie.

5.1.1 Programma-flow

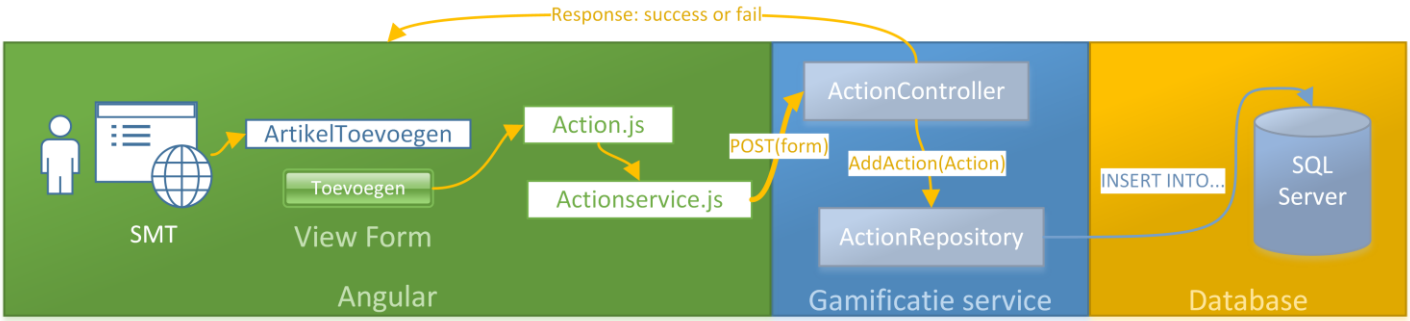
Om de gamificatieservice inzichtelijk te maken qua functionaliteit zijn hieronder drie voor-tot-achter scenario's uitgeschreven. Deze geven globaal een beeld van hoe de verschilende onderdelen werken en waar ze verantwoordelijk voor zijn.
Let erop dat de code in de figuren pseudo-code zijn om inzicht te geven. Ze reflecteren dus niet de gerealiseerde code.

5.1.1.1 Scenario 1: Speelveldmodellering

Een speelveldmodelleerder wil een nieuwe actie toevoegen aan het systeem. Het is een actie in de context van de enterprise applicatie knowNow. De actie heet 'ArtikelToevoegen' wat in knowNow leidt tot een nieuw artikel in het systeem. Om deze bekend te maken gebruikt hij de speelveldmodelleringtool.

Uitgang:

- De speelveldmodelleerder heeft toegang tot het systeem
- De actie 'ArtikelToevoegen' bestaat nog niet in het systeem

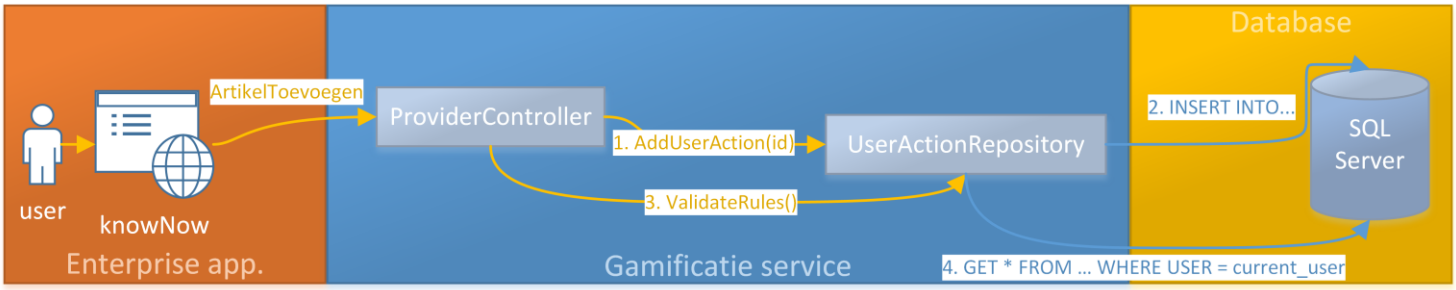


Figuur 17: Scenario 1 - Speelveldmodellering

Dit zijn de handelingen die onderwater gebeuren wanneer een gebruiker naar de Acties pagina gaat in de SMT en een nieuwe actie toevoegt door het formulier in te vullen en op de knop "toevoegen" te drukken.

5.1.1.2 Scenario 2: Event providing

Een gebruiker van knowNow voegt een nieuw artikel toe. Daarbij wordt er een nieuw artikel aan knowNow toegevoegd, en een aanroep gedaan aan de gamificatieservice met de actie 'ArtikelToevoegen'.

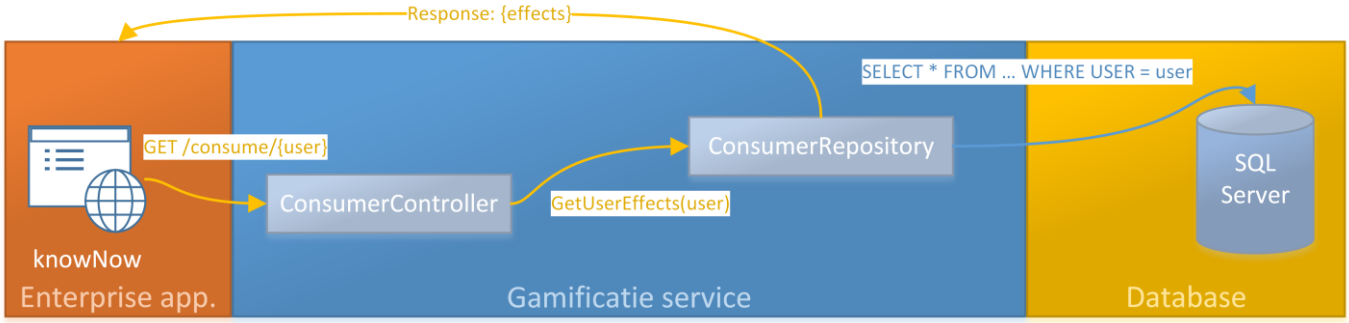


Figuur 18: Scenario 2 – Event providing

Dit zijn de handelingen die onderwater gebeuren wanneer een gebruiker een actie uitvoert binnen de enterprise applicatie, waarbij de enterprise applicatie de gamificatie service heeft geïmplementeerd.

5.1.1.3 Scenario 3: Consuming results

De enterprise applicatie wil zijn gebruiker(s) kunnen tonen welke resultaten zijn behaald rondom gamificatie. Wanneer een gebruiker naar zijn profiel gaat krijgt hij een overzicht van alle effecten die de gebruiker heeft behaald.



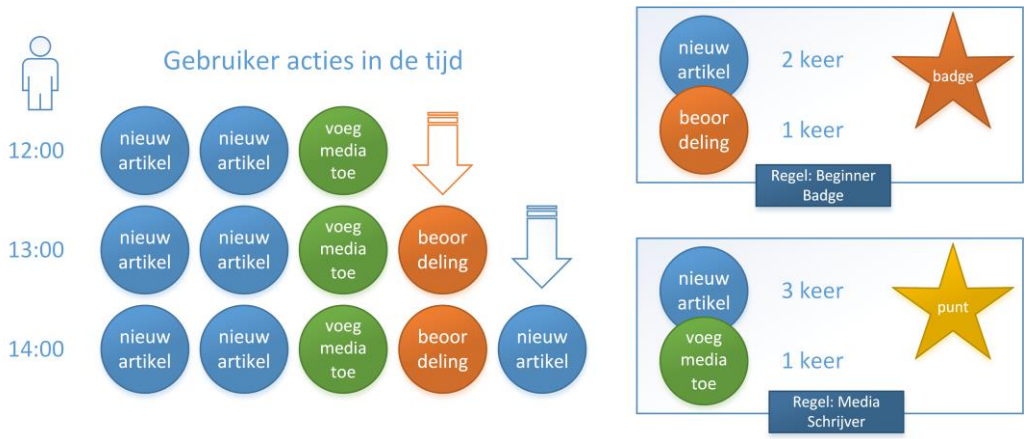
Figuur 19: Scenario 3 – Consuming results

Dit zijn de handelingen die onderwater gebeuren wanneer een gebruiker bij een enterprise applicatie naar zijn profielpagina gaat, waarbij de enterprise applicatie de gamificatieservice heeft geïmplementeerd.

5.1.2 Logica uitdaging

In de basis is het opzetten van een webservice weinig uitdaging. Dit hoofdstuk beschrijft een uitdaging waar tegenaan gelopen is bij het uitwerken van het POC.
Hoe valideer je acties van een gebruiker tegen gedefinieerde regels om effecten uit te kunnen geven?

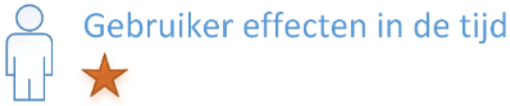
Dit vraagstuk kent enkele moeilijkheden. Acties kunnen immers voorkomen in meerdere regels. Verschillende combinaties acties kunnen tot een effect leiden in een regel.



Figuur 20: Overzicht acties, effecten en regels in de tijd

1. In het figuur zien we om 12:00 dat de gebruikers drie acties heeft gedaan in het systeem. Deze drie acties zorgen niet voor een effect aangezien voor regel één nog de "beoordeling" actie mist, en voor regel twee zijn drie "nieuw artikel" acties nodig. De gebruiker verdient dus geen effect over deze opsomming van acties.

2. Om 13:00 doet de gebruiker een "beoordeling" actie. Daarmee voldoet de gebruiker aan de "Beginner Badge" regel. Het systeem valideert alle acties tegen de bestaande regels en geeft de gebruiker een badge effect:

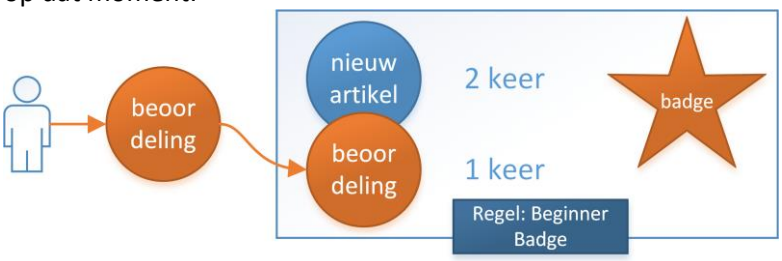


3. Om 14:00 doet de gebruiker een "nieuw artikel" actie. Daarmee voldoet hij met al zijn acties aan de beide regels. Het systeem valideert alle acties tegen de bestaande regels en geeft de gebruiker een badge en een punt effect:

4. Dit gaat fout! De acties die tot een "Beginner Badge" hebben geleid zijn al een keer gevalideerd met een dat effect als resultaat. Op deze manier zou de gebruiker constant het "Beginner Badge" effect krijgen wanneer zijn acties tegenover de regels worden gevalideerd.

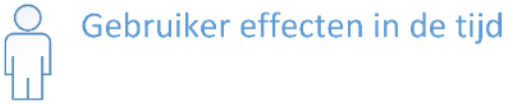
De acties kunnen niet weggehaald worden, er moet immers een combinatie van acties mogelijk zijn ten opzichte van bestaande regels. Om dit op te lossen is er goed nagedacht over vanaf welke kant je valideert. In het bovenstaande voorbeeld vergeleken we de acties op de regels. Na in ieder geval 3 mogelijke vormen is gekozen voor de vorm beschreven op de volgende pagina.

1. Zodra de gebruiker om 13:00 de “beoordeling” actie doet, worden alle regels opgehaald die in ieder geval die actie bevatten; overige regels doen er tenslotte niet toe ter validatie op dat moment.

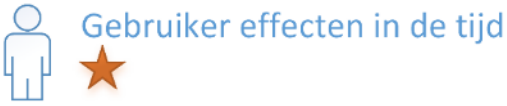


2. Vervolgens worden alle acties van de gebruiker opgehaald

3. Aan de hand van deze actie kan per regel worden gekeken hoe vaak de acties tot een effect hadden moeten leiden. In het voorbeeld is dit 1 keer.
4. Er wordt gekeken naar hoeveel van die effecten de gebruiker al heeft.



5. Er wordt geconstateerd dat de gebruiker nog geen effect heeft ontvangen voor die regel, maar wel voldoet aan de regel. Dus krijgt de gebruiker het effect op die regel.



Nog een voorbeeldscenario is uitgewerkt in bijlage 5: Logica uitdaging scenario’s. Dit geeft bij twijfel meer inzicht in het proces.

5.2 *Front-end: Speelveldmodelleeringtool*

De speelveldmodelleringtool is ten behoeve van de configuratie van de gamificatieservice. De realisatie hiervan is gemaakt door gebruik te maken van Yeoman en AngularJs.

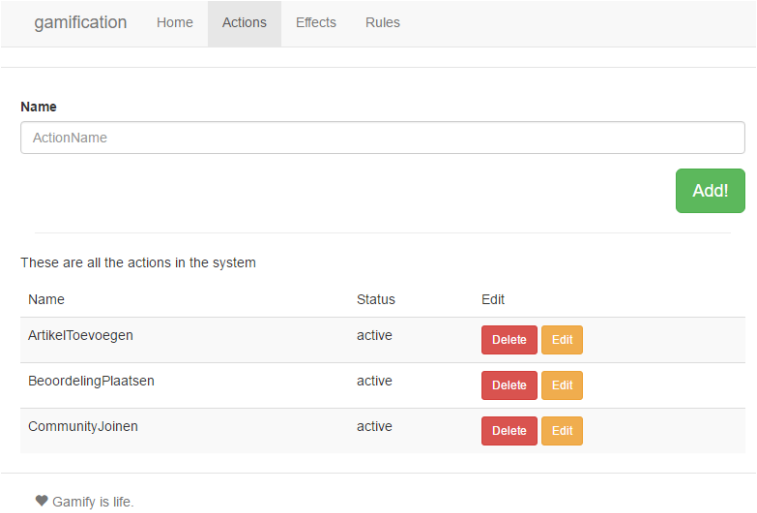
“Yeoman is a generic scaffolding system allowing the creation of any kind of app. It allows for rapidly getting started on new projects and streamlines the maintenance of existing projects.”

(Yeoman, sd)

“AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly.”

(Angular, sd)

Deze technologiën maakte het mogelijk om snel een webapplicatie op te zetten om met de gamificatieservice te communiceren. De SMT bevat de functionaliteiten die zijn opgesteld aan het begin van dit hoofdstuk.



Figuur 21: Speelveldmodelleringtool screenshot

5.3 *Repository: Database*

Voor de database is gekozen voor Microsoft SQL Server.
Daarbij is bewust gekozen om zelf de queries uit te schrijven en geen gebruik te maken van het Entity Framework (EF).

```
        public Action GetActionById(int id)
        {
            string query = "SELECT id, name, status FROM Actions WHERE id = @id";
            using (var connection = new
SqlConnection(ConfigurationManager.ConnectionStrings[SqlConfig.connectionSource].ConnectionString))
            {
                connection.Open();
                using (SqlCommand command = connection.CreateCommand())
                {
                    command.CommandText = query;
                    command.CommandType = CommandType.Text;

                    command.Parameters.Add("@id", SqlDbType.Int).Value = id;
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        Action action = null;
                        while (reader.Read())
                        {
                            var Id = reader.GetInt32(0);
                            var Name = reader.GetString(1);
                            var Status = reader.GetString(2);
                            action = new Action(Id, Name, Status);
                        }
                        if (action == null)
                            throw new NullReferenceException("No action result");

                        return action;
                    }
                }
            }
        }
```

Figuur 22: Voorbeeld query execution

Door geen gebruik te maken van het EF behoudt je veel overzicht over wat er gebeurt. Wat het figuur hierboven inzichtelijk maakt.

6 Conclusie eindresultaat

Om een antwoord te kunnen geven op de onderstaande vraag is onderzoek verricht. Daarbij is gekozen voor een deel theoretisch- en een deel praktisch onderzoek. De methoden zijn grotendeels kwalitatief van aard, waarbij ruimte is om vraagstelling en methodiek enigszins bij te sturen aan de hand van reeds behaalde resultaten. De vraag is opgesteld in de context van een 4e jaar student software engineering waarbij zowel onderzoek als uitvoering van belang waren.

Hoe kan een herbruikbare, onafhankelijke gamificatieservice worden opgezet zodat deze toegepast kan worden binnen enterprise applicaties?

Onderzoek naar enterprise applicaties is dat een enterprise applicatie enkele softwarekwaliteiten volgt. De kwaliteiten die uit literatuur en gesprekken met software architecten kwamen waren performance, betrouwbaarheid, security, beheerbaarheid en overdraagbaarheid. Bij het ontwikkelen van een softwareoplossing voor enterprise applicaties dienen deze softwarekwaliteiten te worden meegenomen in het ontwerpen en implementeren.

Om een stuk software onafhankelijk en kwalitatief aan te kunnen bieden aan verschillende enterprise applicaties is gekeken naar de mogelijkheden daarin. Om dit te realiseren zijn de kwaliteitseisen meegenomen in het onderzoek. Na het vergelijken van verschillende technieken kwam een REST gebaseerde webservice het beste uit deze vergelijking. Dit is als advies gegeven aan de software architecten die hier vervolgens mee akkoord zijn gegaan.

Bij het onderzoeken naar gamificatie was het van belang om een goede selectie te maken van gamificatie elementen die gebruikt zouden worden in het eerste proof-of-concept. De wens was om drie gamificatie elementen te kunnen gebruiken. Uit het onderzoek is gekomen dat punten, levels en badges gamificatie elementen zijn die goed kunnen worden gebruikt in de huidige context.

Verder is er gekeken hoe gamificatie herbruikbaar kan worden aangeboden aan enterprise applicaties, waarbij de klant zijn eigen vorm van gamificatie kan instellen. Hierbij is gekeken naar het instellen van het speelveld middels generieke spelregels. Uit het onderzoek is gekomen dat regels generiek kunnen worden ingesteld door acties en effecten instelbaar te maken door de klant waarbij regels kunnen worden opgesteld met deze acties en effecten. Dit maakt de gamificatie onafhankelijk instelbaar.

Voor het realiseren van een proof-of-concept is onderzoek gedaan naar de softwarecomponenten die een gamificatieservice moet bevatten. Hier is uitgekomen dat middels het gebruikmaken van een multitierarchitectuur de volgende componenten moeten worden ontwikkeld: een userinterface ten behoeve van de configuratie van de gamificatie; een communicatiecomponent; een domein component en een databeheercomponent. Hiervoor is gekozen om constant te kunnen voldoen aan de wensen en eisen die men heeft in de context van enterprise applicaties.

Op basis van deze kennis is het proof-of-concept gerealiseerd. Daarmee is van voor tot achter aangetoond hoe een herbruikbare, onafhankelijke gamificatieservice kan worden opgezet zodat deze kan worden toegepast binnen enterprise applicaties.

6.1 Aanbevelingen

Een aanbevelingen die dient te worden meegenomen is: hoe gaat een enterprise applicatie om met fouten. Daarbij komt de term “fault tolerance” kijken. Fault tolerance is een eigenschap wat het toestaat dat wanneer een systeem een fout tegenkomt, het daar adequaat mee om kan gaan. Waarom dit belangrijk is in de context, is dat de enterprise applicatie er niet vanuit kan gaan dat de gamificatieservice 100% uptime heeft. Daarom moet worden nagedacht over technieken als “graceful degradation” waarbij het systeem een alternatief heeft wanneer de gamificatieservice niet bereikbaar is. (Rouse, 2007)

7 Evaluatie procesgang

Het proces heeft plaatsgevonden tussen 7 maart 2016 en 9 augustus 2016. In deze 6 maanden hebben de volgende personen en processen grotendeels invloed gehad op het onderzoek:

Hogeschool Utrecht	*Docent begeleiding*	Er is mij een docentbegeleider toegewezen. Deze heeft gedurende het afstuderen enkele functies: - Begeleiding en feedback op de op te leveren producten - Ondersteuning algeheel proces - Vraagbaak rondom HU gerelateerde vragen - Beoordeling producten
	Leerteam sessies	De HU deelt alle afstudeerders af in een leerteam. Met het leerteam worden drie momenten afgesproken: 1. Start sessie 2. Plan van aanpak sessie 3. Scriptie sessie
	Afstudeerder	De afstudeerder is verantwoordelijk voor het goed doorlopen van het gehele proces.
Info Support	*Opdrachtgever*	Bij het aannemen van een afstudeeropdracht kreeg ik de desbetreffende opdrachtgever als bedrijfsbegeleider. Hij heeft gedurende het uitvoeren de van de opdracht enkele functies: - Begeleiding en feedback op de op te leveren producten - Scope leveren product - Vraagbaak product gerelateerde vragen - Beoordelen afstudeerproces binnen Info Support
	Technisch begeleiding	Bij Info Support krijgen alle afstudeerders een technisch begeleider. Deze is er om alle technisch gerelateerde vragen aan te stellen. Ook kijkt hij mee naar de kwaliteit van de opgeleverde producten op technisch vlak, niet per se inhoudelijk.
	Proces begeleiding	Bij Info Support krijgen alle afstudeerders een procesbegeleider. Deze is er voor persoonlijke ondersteuning in het afstudeerproces en om alle Info Support gerelateerde vragen aan te stellen. Om de twee weken is daarvoor een meeting.
	SCRUM	Er is volgens een minimale vorm van SCRUM gewerkt waarbij gekozen is om daily standups, sprints van 2 weken en aan het einde van elke sprint een retrospective en sprintplanning te doen.
	ASP.NET MVC cursus	Gedurende het afstudeerproces is één cursus gevolgd op het gebied van ASP.NET MVC 4.5

Tabel 6: Personen en processen bij het onderzoek

In principe zijn al deze processen als prettig ervaren, wel hadden ze hier en daar beter kunnen verlopen. Dit is behandeld in de volgende onderdelen:

- **Tijd en planning**
  - Het plannen van projecten blijft erg lastig wanneer je nog niet volledig weet waar je tegenaan gaat lopen. De hoeveelheid nieuwe informatie gedurende het proces was lastig synchroon te houden met gemaakte planningen in de tijd.
    - <u>Voorkomen</u>: weten dat het onbekende er is in je planning en dat mee nemen in je planning. Agile aanpak werkt hier beter dan de waterval methode die gekozen is in het plannen van het “geheel”.
  - De vorm van werken rondom het plan van aanpak bij de Hogeschool Utrecht sloot weinig aan bij de werkwijze van Info Support. Dit is veel besproken met alle betrokkenen bij Info Support. Uiteindelijk is daar wel acceptatie in gekomen, maar dit gaf toch enige weerstand in het proces. Daarnaast heeft het naar mijn

idee meer tijd gekost dan nodig. Deze tijd had wellicht beter besteed geweest bij het uitvoeren van onderzoek en productontwikkeling.

    - Voorkomen: vroeger communiceren wat de verwachtingen zijn van beide partijen. Er werd te laat gesignaleerd dat die verwachtingen uiteenliepen, waardoor er wellicht al eerder een soort “paniek” optrad.

- **Afspraken**
  - In het proces kwam ik op een kamer te werken waar zodanig veel “all-round-kennis” was dat er weinig behoefte was aan technische begeleiding van de specifieke begeleider. Dit was aan het begin een uitdaging waar we samen uitgekomen zijn. In principe kwam er de afspraak om ongeacht de gemiste behoefte aan begeleiding, hem wel actief te betrekken bij het proces.
    - Voorkomen: eerder concrete afspraken maken met alle mensen betrokken in het proces en inzicht krijgen wat de concrete verwachtingen zijn van deze mensen.
  - Aan het begin was er geen gewenning dat met iedereen afspraken gewoon konden worden ingeschoten. Het vergde gewenning om dat te doen, zeker wanneer ik mensen nog niet had ontmoet. Uiteindelijk is dat wel gewoonte geworden en toegepast wanneer nodig.
    - Voorkomen: dit kan niet voorkomen, het was een ervaring onderdeel wat voor iedereen anders werkt. Dit was al vrij snel opgelost door de onzekerheid te benoemen waarbij die onzekerheid werd weggenomen.
- **Communicatie**
  - De communicatie is over het algemeen erg goed verlopen. Een compliment richting Info Support is dat iedereen goed bereikbaar is en er geen onderscheid wordt gemaakt tussen “werknemers” en “afstudeerders” wanneer het aankomt om communicatie onderling.
  - Wanneer er van beide kanten minder communicatie is geweest wegens uitval van personen is dit over het algemeen goed gecommuniceerd of meegenomen in de retrospective.
- **Terugkoppelingen**
  - Zelf ben ik het meest te spreken over de hoeveelheid feedback die ik heb kunnen ontvangen door de verschillende begeleiders. Dit heeft me ontzettend geholpen gedurende het afstuderen.
  - Ook ben ik er achter gekomen dat feedback ontvangen van veel verschillende personen onhandig kan zijn. Zo zitten de begeleiders met “verschillende petten” op, wat het lastig maakt om iedereen tevreden te stellen.
    - Voorkomen: duidelijk van tevoren afstemmen om wat voor feedback gevraagd wordt. De beoordelingscriteria behandelen met alle betrokken personen.
  - Door middel van een aangedragen methodiek van mijn opdrachtgever is in een kwadrant opgeschreven door de betrokken partijen wat er beter kon, waar men blij mee was, waarvoor dankbaar en welke ideeën voor komende sprint.

Figuur 23: Feedback methodiek

Er is altijd ruimte voor verbetering in een project: zeker achteraf wordt dit vaak inzichtelijk. Het belangrijkste is om voldoende ruimte te hebben voor communicatie waarin eventuele problemen konden worden getackeld. Dit is eigenlijk altijd gebeurd, wat ook terug te vinden is in het proces. Met enkele zinnen wordt hieronder een advies beschreven op basis van het afgelopen proces en hoe dit te verbeteren in de toekomst:

- Met alle partijen een eenduidige communicatie verzinnen. Ervoor zorgen dat de verwachtingen duidelijk zijn en het proces “op één lijn zit”.
- In de planning voldoende ruimte openlaten om onvoorziene zaken te kunnen opvangen in het proces.
- Vaste afspraken maken en die door het gehele proces door te laten lopen.
- Voorkomen dat je als afstudeerder probeert enkel te voldoen en zelf ook duidelijke verwachtingen aangeven die je nodig hebt voor het afstudeerproces.
- Focus houden op de kernzaken. Het is goed om zo nu en dan buiten de lijntjes te komen, echter moet de focus voor de belangrijkste zaken in het proces centraal staan.
- Vroegtijdig aangeven wanneer er onzekerheden zijn.

Bibliografie

Angular. (n.d.). *What is Angular?* Retrieved from AngularJs: https://docs.angularjs.org/guide/introduction

Balasubramanian, R. (2002). *N-Tier Architecture*. Retrieved from Distributed systems: http://www.csc.villanova.edu/~rbalasub/disarti1.doc

Burke, B. (2014). *Gamify, how gamification motivates people to do extraordinary things.* Brookline: Bibliomotion. Inc.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining "Gamification". *Proceeding*.

Dubey, A., & Wagle, D. (2007). Delivering software as a service. *The McKinsey Quarterly*, 5.

Elkstein, M. (2008, februari 12). *Learn REST: A tutorial*. Retrieved from Learn REST: A tutorial: http://rest.elkstein.org/

Fielding, R. T., & Taylor, R. N. (2002). *Principled Design of the Modern Web Architecture.* University of California. ACM Transactions on Internet Technology.

Fowler, M. (2011). *Patterns of Enterprise Application Architecture.* Boston: Pearson Education, Inc.

Hamari, J., & Koivisto, J. (2015). Why do people use gamification services? *Elsevier*, 419-431.

InfoSupport. (2016). *Over Info Support*. Retrieved from Info Support B.V.: http://www.infosupport.com/

International organization for standardization. (2011). *NEN-ISO/IEC 25010.* Geneva: ISO copyright office.

Kim, B. (2015). Designing Gamification in the Right Way. In B. Kim, *Understanding Gamification* (pp. 29-35).

Liyanage, N. (2013, maart 28). *Advantages and Disadvantages of the architectures*. Retrieved from Client/Server Architecture: http://clientserverarch.blogspot.nl/2013/03/advantages-and-disadvantages-of.html

Marczewski, A. (2012). *Gamification: A Simple Introduction.*

Marczewski, A. (2015, februari 4). *47 Gamification elements, mechanics and ideas*. Retrieved from Gamified UK: http://www.gamified.uk/2015/02/04/47-gamification-elements-mechanics-and-ideas/

Megginson, D. (2005, 3 25). *XML Performance and Size*. Retrieved from InformIT: http://www.informit.com/articles/article.aspx?p=367637&seqNum=2

Microsoft. (2007, maart 14). *MSDN*. Retrieved from SOAP: Simple Object Access Protocol: http://web.archive.org/web/20070314063040/http://www.microsoft.com/netherlands/msdn/products/soap.aspx

Microsoft. (2015, mei 28). *Getting started with ASP.NET Web API 2 (C#)*. Retrieved from ASP.NET: http://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api

Microsoft. (n.d.). *Chapter 1: What is an Enterprise Application?* Retrieved from Microsoft Developer Network: https://msdn.microsoft.com/en-us/library/aa267045(v=vs.60).aspx

Motamarri, J. (2013, March 8). *Compare RESTful vs. SOAP Web Services*. Retrieved from DZone / Integration Zone: https://dzone.com/articles/j2ee-compare-restful-vs-soap

Nielson, B. (2013, maart 4). *Gamification and the Hype Cycle*. Retrieved from Your Training Edge: http://www.yourtrainingedge.com/gamification-and-the-hype-cycle/

OWASP. (2016, april 10). *REST Security Cheat Sheet*. Retrieved from OWASP.org: https://www.owasp.org/index.php/REST_Security_Cheat_Sheet

Pedreira, O., García, F., Brisaboa, N., & Piattini, M. (2015). Gamification in software engineering - A systemetic mapping. *Information and Software Technology*, 157-168.

Robson, K., Plangger, K., Kietzmann, J. H., McCarthy, I., & Pitt, L. (2015). Is it all a game? Understanding the principles of gamification. *Business Horizons*, 411-420.

Robson, K., Plangger, K., Kietzmann, J. H., McCarthy, I., & Pitt, L. (2016). Game on: Engaging customers and employees through gamification. *Business Horizons*(59), 29-36. Retrieved maart 10, 2016

Rouse, M. (2007, maart). *Graceful degradation*. Retrieved from TechTarget: http://searchnetworking.techtarget.com/definition/graceful-degradation

Scribbr. (2016, januari 21). *https://www.scribbr.nl*. Opgeroepen op maart 9, 2016, van Scribbr.nl: https://www.scribbr.nl/

Seaborn, K., & Fels, D. I. (2015). Gamification in theory and action: A survey. *Elsevier*, 14-31.

Seely, S. (2002, oktober). *Understanding WS-Security*. Retrieved from MSDN: https://msdn.microsoft.com/en-us/library/ms977327.aspx

Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). *Web services composition: A decade's overview.* Information Sciences.

Van Dale. (n.d.). *Betekenis 'Generiek'*. Retrieved from Gratis woordenboek | Van Dale : http://www.vandale.nl/opzoeken?pattern=generiek&lang=nn#.V2kLjriLRaQ

Van Dale. (n.d.). *Betekenis 'onafhankelijk'*. Retrieved from Gratis woordenboek | Van Dale: http://www.vandale.nl/opzoeken?pattern=onafhankelijk&lang=nn

Wikipedia. (2015, april 26). *Validatie*. Retrieved from Wikipedia: https://nl.wikipedia.org/wiki/Validatie

Wikipedia. (2016, May 17). *Minimum Viable Product*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Minimum_viable_product

Wikipedia. (2016, maart 15). *Multitier architecture*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Multitier_architecture

Wikipedia. (2016, Jun 15). *Software as a service*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Software_as_a_service

Willemse, K. (2013, februari 26). *Nog weinig spelletjes in de directiekamer*. Retrieved from Computable: https://www.computable.nl/artikel/expertverslag/internet/4676995/4573232/nog-weinig-spelletjes-in-de-directiekamer.html

Yeoman. (n.d.). *Getting started with yeoman*. Retrieved from Yeoman: http://yeoman.io/learning/

Figuren

Enkele verwerkte figuren in het document zijn afkomstig van Freepik.com.