SOGETI

Kenniskaartenapplicatie

Project Management System

Hogeschool Utrecht

Eerste examinator: Mariecke Schipper

Roy Straub, 1614878

V1.3

Versie informatie

Versie	Datum	Bijzonderheden	Auteur
1.0	16-03-15	Eerste conceptversie	Roy Straub
1.1	03-04-15	Tweede conceptversie	Roy Straub
1.2	28-04-15	Derde conceptversie	Roy Straub
1.3	29-05-15	Eindversie	Roy Straub

Voorwoord

Gedurende de afstudeerperiode heb ik ontzettend veel geleerd van de mensen die betrokken zijn geweest bij dit onderzoek. Een paar personen wil ik hierbij in het bijzonder bedanken. Allereerst wil ik Esther van der Stappen bedanken voor haar begeleiding gedurende dit project. Dankzij de door haar verstrekte input en feedback is de scriptie in mijn optiek naar een niveau hoger getild. Verder wil ik Sogeti en hun begeleiders Willem van de Griendt en Erwin de Gier bedanken voor hun begeleiding gedurende het afstudeertraject. Tot slot wil ik mijn partner bedanken voor haar feedback en haar steun tijdens het verloop van het afstuderen.

Managementsamenvatting

Binnen het Java Centre of Excellence [CoE] van Sogeti wordt gebruik gemaakt van kenniskaarten. De kenniskaarten worden in Excel bestanden bijgehouden. Ze worden gebruikt om het kennisniveau van projectteams en de expertisedekking van een project in kaart te brengen. Hiermee is voor projectmanagers inzicht te verkrijgen in de bezetting van een project.

De wens van Sogeti is om hier een applicatie voor te hebben. Aspecten van dit systeem vallen onder een categorie systemen waarmee kennis binnen een bedrijf wordt beheerd. De zogenaamde *Knowledge Management Systems* [KMS]. Binnen de KMS valt een subcategorie systemen waarmee vaardigheden van werknemers worden vastgelegd. Deze systemen zijn de zogenaamde *Skill Management Systems* [SMS]. SMS bieden inzicht in de vaardigheden waar werknemers over beschikken en daarnaast welke *gaps* er tussen een werknemer en een taak binnen het bedrijf bestaat. In het geval van de te ontwikkelen applicatie zal het voornamelijk expertisedekking bij een project betreffen. Het inzichtelijk hebben van skillgaps bij een project stelt de projectmanager in staat zijn projectbezetting te monitoren en waar nodig in te grijpen.

Het te ontwikkelen systeem moest qua architectuur zo opgezet worden dat het later ook buiten het CoE kan worden toegepast. Door het proces te automatiseren is het mogelijk om de kenniskaarten gemakkelijker te beheren. Daarnaast centraliseert het de kenniskaarten, waardoor er niet langer meerdere versies circuleren, zoals in de beginsituatie het geval was. Verder maakt een applicatie geavanceerde toepassingen mogelijk.

De realisatie van het systeem bracht enkele problemen met zich mee. Zo lag slechts een deel van de functionaliteiten die het systeem moest bevatten vast. Verder was het van belang dat de efficiëntie van het systeem hoog lag. Uit vooronderzoek bleek usability aan de efficiëntie van het systeem bij te kunnen dragen. De vraag was echter hoe het beste een goede usability kon worden bereikt.
De centrale vraag voor dit onderzoek luidde als volgt:
Hoe kan het kenniskaartensysteem van Sogeti's Java Centre of Excellence het beste geautomatiseerd worden, zodat het project management efficiënter verloopt?
De centrale vraag is opgemaakt uit de volgende deelvragen:

- *Hoe werkt het handmatige kenniskaartenproces van Sogeti's Java Centre of Excellence?*
- *Aan welke functionele- en niet functionele requirements moet het kenniskaartensysteem voor Sogeti's Java Centre of Excellence voldoen?*
- *Is het mogelijk om de huidige data in de kenniskaarten van Sogeti's Java Centre of Excellence in het automatische systeem mee te nemen en zo ja, op welke manier(en)?*
- *Op welke manier(en) kan User Experience en Usability bijdragen aan de efficiëntie van het systeem?*
- *Hoe kan op architecturaal niveau uitbreidbaarheid en toekomstbestendigheid van de kenniskaartenapplicatie worden bereikt?*

De eerste deelvraag werd beantwoord door gebruik te maken van interviews. Hier is een gebruiker van de huidige kenniskaarten voor geïnterviewd. Hieruit bleek het kenniskaartenproces globaal te bestaan uit vier stappen:

1. Het vastleggen van de vereiste expertises voor een project
2. Projectleden leggen hun expertise vast
3. De manager controleert de ingevulde expertiseniveaus

4. Borging van kennis toonbaar maken aan de klant

De tweede deelvraag leverde een requirementanalyse op. Wederom zijn hier interviews voor gebruikt. De interviews zijn gehouden met de product owner en een gebruiker van de huidige kenniskaarten. Functionaliteiten die in de Excel oplossing zaten bleken de hoogste prioriteit te verkrijgen. Een andere functionaliteit met hoge prioriteit bleek het genereren van een (Read-Only) rapport voor de klant te zijn.

Het beantwoorden van de derde deelvraag is met behulp van desk research gedaan. Uit desk research bleek dat er libraries bestonden waarmee data uit Excel bestanden in Java konden worden benaderd. Vervolgens is er een vergelijking gemaakt tussen enkele grote libraries aan de hand van opgestelde criteria. Deze vergelijking toonde aan dat Apache POI de beste library voor de huidige context was.

Usability en User Experience bleken aan de hand van literatuurbronnen bij te kunnen dragen aan de efficiëntie van een applicatie. Het vermindert het aantal fouten die gebruikers maken of vermindert de tijd die ze nodig hebben om een handeling uit te voeren. Daarnaast zijn enkele factoren aan het licht gekomen die effect hebben op de Usability in een webapplicatie. Dit zijn de gebruikers, taken, technologie en context. Het vijf lagenmodel (Garrett, 2011) is gedurende het traject toegepast om een goede User Experience in de applicatie te bieden.

De laatste deelvraag was met behulp van literatuur beantwoord. Voor het mogelijk inzetten van de applicatie in andere Business Lines is het domein ontworpen volgens object-georiënteerde principes. Integratie met externe systemen kan worden bereikt door het Strategy Pattern toe te passen op punten waar deze integratie plaats zou kunnen vinden. Tot slot ondersteunt de applicatie de mogelijkheid om als Confluence-plugin te worden neergezet. Dit is mogelijk gemaakt door de applicatie volgens een Service Oriënted Architecture neer te zetten. De backend wordt via een REST webservice beschikbaar gemaakt. Een webapplicatie is geïmplementeerd als cliënt.

Het Proof of Concept [PoC] is ontworpen, gerealiseerd en getest met de antwoorden op de deelvragen als uitgangspunt. Dit heeft er toe geleidt dat veel van de doelen voor de applicatie zijn behaald. Allereerst zorgt de applicatie voor een gecentraliseerde kenniskaart. Daarnaast is het PoC in de toekomst buiten het CoE te gebruiken, met eventuele integratie van externe systemen. Ook maakt de applicatie uitbreiding mogelijk op architecturaal gebied. Om het PoC in productie te nemen dient er op het gebied van authenticatie, authorisatie en beheer het een en ander te gebeuren. Daarnaast rest nog één belangrijke functionaliteit die geïmplementeerd moet worden: het genereren van een rapport voor de klant.

.

Inhoudsopgave

1	**Inleiding**	**1**
2	**Context**	**2**
2.1	Sogeti	2
2.2	Aanleiding	3
2.3	Theoretisch kader	3
3	**Opdracht**	**7**
3.1	Probleemstelling	7
3.2	Doelstelling	7
3.3	Opdrachtomschrijving	8
4	**Vraagstelling**	**11**
5	**Methoden en Aanpak**	**12**
5.1	Huidig kenniskaartenproces	12
5.2	Requirementanalyse	13
5.3	Legacy data	14
5.4	Efficiëntie	14
5.5	Architectuur	14
5.6	Proof of Concept	15
5.7	Communicatie	15
5.8	Softwareontwikkelmethode	16
5.9	Kwaliteitsbewaking	18
6	**Resultaten**	**20**
6.1	Huidig kenniskaartenproces	20
6.2	Requirements	22
6.3	Legacy data	26
6.4	UX en Usability	28
6.5	Architecturale oplossingen	31
6.6	Proof of concept	35
7	**Conclusie**	**43**
8	**Aanbevelingen**	**45**
9	**Evaluatie**	**46**
10	**Bronvermelding**	**47**
11	**Contactpersonen**	**49**
12	**Bijlagen**	**50**
A.	Plan van aanpak	50
B.	Zelfreflectie	51
C.	Antwoorden interviews kenniskaartenproces	52
D.	Antwoorden interviews requirementanalyse	53
E.	Inception Report	56
F.	Elaboration Report	57
G.	Construction Report	58

1 Inleiding

Sogeti maakt in hun Java Centre of Excellence (CoE) gebruik van zogenaamde kenniskaarten. Deze kenniskaarten brengen de kennis van een projectteam in kaart en zijn onderdeel van een stukje kwaliteitsborging. Dit wordt vervolgens uiteengezet tegen criteria die door de klant zijn opgesteld bij een project. Het wordt zo mogelijk voor de klant om een kennisoverschot- of tekort te zien bij specifieke competenties die aan een project hangen. Momenteel vindt dit proces handmatig met behulp van een Excelbestand plaats.

Dit document beschrijft de afstudeeropdracht met betrekking tot het implementeren van een geautomatiseerd kenniskaartensysteem bij Sogeti. Het systeem zal de bestaande oplossing in Excel vervangen en project management efficiënter trachten te maken. De opdracht zal tot in detail worden toegelicht en er zal een theoretisch kader worden afgebakend. Verder zal de organisatie beknopt worden beschreven.

Tijdens het afstudeerproject wordt onderzoek uitgevoerd dat bepaalde vragen beantwoordt omtrent het systeem dat gerealiseerd zal worden. Deze vraagstelling zal behandeld en toegelicht worden. De onderzoeksvragen zullen worden beantwoord met behulp van methoden die beschreven zullen worden in het hoofdstuk Methoden en Aanpak.

Na de toelichting en keuzeverantwoording van de toegepaste methoden en aanpak zullen de resultaten van het onderzoek worden getoond. Deze resultaten zullen per deelvraag worden gepresenteerd. De verslaglegging van de ontwikkeling van het product zal hierop volgen. Aan de hand van deze resultaten zal er tot slot een conclusie worden aangedragen.

2 Context

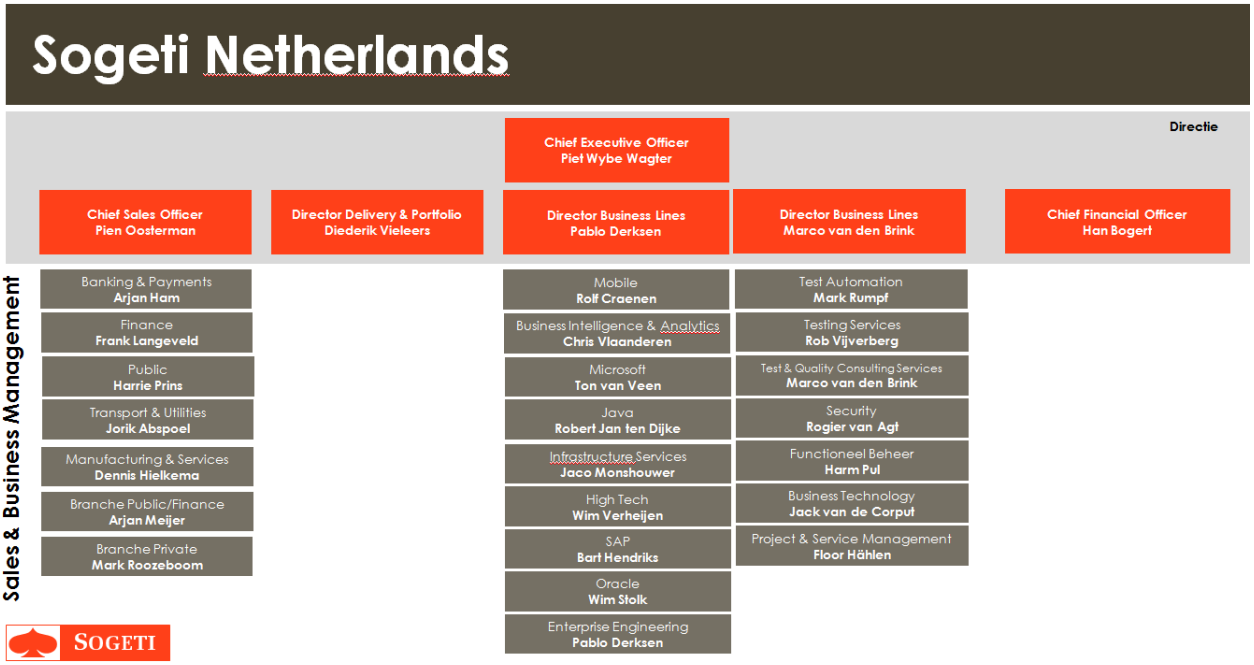
In dit hoofdstuk zal de context van het project ter sprake komen. Onder andere de organisatie waar het afstuderen plaatsvindt zal hier worden toegelicht. Daarnaast zal de aanleiding voor het project worden beschreven, hiermee zal duidelijk de ontstaansreden van het project worden toegelicht. Tot slot volgt een kort theoretisch kader waarin de centrale begrippen van dit traject worden besproken.

2.1 Sogeti

De afstudeeropdracht zal bij het bedrijf Sogeti plaatsvinden. Sogeti (2015) is een IT-dienstverlener die wereldwijd actief is in vijftien landen. Eén van de landen waarin Sogeti actief is, is Nederland. Sogeti Nederland bestaat uit ongeveer vijfentwintighonderd medewerkers en beschikt over kantoren in Amersfoort, Amsterdam, Eindhoven, Groningen en het hoofdkantoor in Vianen.

Naast het feit dat Sogeti geografisch op veel fronten actief is, geldt dit ook voor de verschillende takken van de markt. Zo zijn er verschillende afdelingen voor onder andere Java, Microsoft, Mobile en Oracle. Sogeti staat verder bekend in de markt vanwege hun capabiliteit omtrent softwaretesten en architectuur. Enkele klanten aan wie Sogeti diensten levert zijn ING, KPN, KLM en ProRail.

Zoals eerder genoemd is een groot deel van de business van Sogeti het detacheren van werknemers. Dit proces beslaat verschillende afdelingen, of zoals ze intern worden genoemd: Business Lines (in figuur 1 afgebeeld als grijze blokken). Sales gebruikt het zogenoemde Business Informatie Systeem (BIS) om binnenkomende aanvragen samen met de afdeling planning te vertalen naar mogelijke matches. Een match is het vinden van een passende werknemer bij de openstaande aanvraag. Eén van de belangrijke criteria bij deze matchingprocedure is het afspiegelen van de vereiste kennis voor een project ten opzichte van de kennis van de beschikbare werknemers. De huidige bedrijfsstructuur volgens Sogeti (2015, januari) is in figuur 1 weergegeven.



Figuur 1. Bedrijfsstructuur Sogeti januari 2015

Binnen de Business Line Java bevindt zich het Java Centre of Excellence [CoE]. Dit is een groep Javaspecialisten die volgens E. de Gier (persoonlijke communicatie, 6 maart, 2015) intern applicaties ontwikkelen en beschikbaar zijn voor consultancy. Het bevat werknemers die niet gedetacheerd zijn. Binnen dit CoE zal het project plaatsvinden en het Proof of Concept worden ontwikkeld. De resulterende applicatie zal in eerste instantie voor intern gebruik van het CoE dienen met mogelijk perspectief op toepassing bij andere Business Lines en integratie met andere systemen. De student zal deel uitmaken van het CoE. Hierin zal het ontwerpen, ontwikkelen en testen van de kenniskaartenapplicatie de verantwoordelijkheid zijn van de student.

2.2 Aanleiding

Kennis en kwaliteitsborging is belangrijk binnen Sogeti. Een groot gedeelte van de omzet komt namelijk van het detacheren van werknemers bij klanten. De klanten willen zoveel mogelijk inzicht en zekerheid in de kwaliteit van een project. Het is van belang dat inzichtelijk is of de juiste bezetting bij een project aanwezig is. Door Inzicht te hebben in de vaardigheden waar werknemers over beschikken en dit uiteen te zetten ten opzichte van vereiste vaardigheden bij een project is dit mogelijk. Het maakt het voor projectmanagers in één oogopslag duidelijk waar eventuele kennistekorten- of overschotten zich voordoen. Aan de hand hiervan kan vervolgens actie worden ondernomen, bijvoorbeeld in de vorm van training voor een werknemer die een competentie mist.

De huidige oplossing binnen het CoE is het gebruik van zogenaamde kenniskaarten. Deze kenniskaarten worden gebruikt om kennis van werknemers op bepaalde gebieden te inventariseren en uiteen te zetten tegenover de criteria bij projecten. Dit maakt het voor Sogeti mogelijk om een extra stuk kwaliteitsborging te leveren. Daarnaast is op deze manier te allen tijde inzichtelijk of bij een project de juiste specialisme aanwezig is. Dit maakt de kenniskaarten een krachtig gereedschap voor projectmanagement.

De kenniskaarten worden momenteel bijgehouden in Excel. Het toevoegen van categorieën, kennisgebieden en medewerkers aan de kenniskaarten vindt handmatig plaats. De wens van het CoE is om hier een geautomatiseerd en centraal systeem voor te hebben. Hier moet het bestaande, handmatige proces in verwerkt worden. Echter zou het automatische systeem geavanceerde bewerkingen mogelijk maken zoals bijvoorbeeld het automatisch bijwerken van de lijst met vaardigheden in de kenniskaarten.

2.3 Theoretisch kader

Kennis

Sogeti is een kennisbedrijf, maar wat is kennis eigenlijk? Hoe verhoudt kennis zich tot informatie en data? Alavi & Leidner (2001) verzamelden verschillende definities van kennis, voornamelijk met toespitsing op de IT. Uit de verschillende definities maakten ze op dat de veronderstelling wordt gemaakt dat kennis, informatie en data van elkaar verschillen. Dit aangezien er anders niets nieuws is aan het beheren van kennis ten opzichte van het beheren van data en informatie. Een toepasbare definitie van deze drie begrippen in de context van dit project werd gemaakt door Maglitta (geciteerd in Alavi & Leidner, 2001). Hij definieerde data als ruwe getallen en feiten. Informatie werd omschreven als verwerkte data. Tot slot werd kennis beschreven als informatie die toepasbaar is gemaakt.

In de huidige situatie bij Sogeti worden de expertises van werknemers in kaart gebracht met een skillsmatrix. In essentie zijn de categorieën die hier in vallen en de waarden die de werknemers hier invullen

een kennisinventarisatie. De matrix geeft weer tot in welke mate de werknemers kennis hebben over een specifiek onderwerp. Als hierbij de definitie van kennis door Maglitta wordt aangehouden is dit dus informatie over een onderwerp die werknemers in de praktijk kunnen toepassen.

Vaardigheden en competenties

Centrale begrippen tijdens dit project zijn de begrippen vaardigheden (skills) en competenties (competencies). Hoewel deze begrippen veel van elkaar weg hebben zijn ze niet geheel te substitueren. Volgens TalentAlign (2012, december) is een vaardigheid iets wat is geleerd om een taak uit te kunnen voeren. Een competentie is breder. Competenties kunnen bestaan uit meerdere vaardigheden en zijn dus overkoepelend. Ze bevatten ook mogelijkheden, gedrag en houding.

Een voorbeeld van het verschil tussen deze twee begrippen is gedefinieerd op TalentAlign (2012, december). Gebruik makend van een IT-voorbeeld kan een vaardigheid worden omschreven als bijvoorbeeld het kunnen coderen in Java of C#. Een voorbeeld van een competentie zou in deze analogie analytisch vermogen kunnen zijn. Elke vaardigheid in het spectrum vereist diezelfde competentie. Zo zou het opdoen van de vaardigheid coderen in C++ dezelfde competentie(s) kunnen vereisen.

Knowledge Management

Het beheren van kennis is sinds de jaren negentig een belangrijk aspect van bedrijven geworden. Nonaka (1991) inspireerde velen met zijn opvatting:
"*In an economy where the only certainty is uncertainty, the one sure source of lasting competive advantage is knowledge*".

Het artikel van Nonaka was aanleiding voor het ontstaan van het begrip *Knowledge Management* (Beck, 2003). Kennis werd gezien als een resource van de organisatie (Alavi & Leidner, 2001). Deze resource was minder vatbaar voor veranderingen van de markt in tegenstelling tot vluchtige resources zoals bijvoorbeeld kapitaal. Men zag in dat informatie van belang is voor het innoveren en het voorblijven van de concurrentie. Het beheren van deze resource werd een belangrijk onderwerp. Dit proces van het beheren van informatie en kennis werd ondergebracht onder de term Knowledge Management.

Knowledge Management werd door de groeiende interesse in het onderwerp geautomatiseerd in informatiesystemen. Deze lijn met systemen werden *Knowledge Management Systems* (KMS) genoemd. Het doel van deze systemen is het maken, het verzamelen, het organiseren en verspreiden van kennis binnen een bedrijf (Alavi & Leidner, 1999). Deze systemen zouden bedrijven in staat stellen om sneller te reageren op veranderingen in de markt en het zou een bedrijf flexibeler maken. Dit zou er volgens Harris en Stata (geciteerd in Alavi & Leidner, 1999) voor zorgen dat het bedrijf innovatiever en productiever wordt. Enkele voorbeelden van deze KMS zijn bijvoorbeeld intranet of een tool zoals Confluence (2015).

Skill Management en Competency Management

Onderdeel van het voorgaand benoemde KM is Skill Management (SM). SM houdt volgens Beck (2003) het beheren van kwalificaties, ervaringen en kennis van medewerkers in. Door deze kennis met een bepaald medium vast te leggen is het voor het bedrijf inzichtelijk welke medewerkers bij welke projecten inzetbaar zijn. Dit is gelijk één van de belangrijke toepassingen van SM. Het kan worden gebruikt in KM; werknemerprofielen kunnen tegenover opdrachtprofielen worden gezet. Verder wordt het mogelijk om te zoeken naar een specifiek profiel.

Overigens benadrukt Beck (2003) het belang van de scheiding tussen projecten/taken en personen. Ten eerste zorgt dit er voor dat de vraag gesteld moet worden wat een project vereist en wat een werknemer aan expertise biedt. Verder wordt het mogelijk om een scheiding te maken tussen de

"cans" en "cannots" van een persoon. Deze verschillen kunnen geanalyseerd worden en dit kan informatie bieden die bijvoorbeeld gebruikt kan worden bij evaluatiegesprekken.

Ook voor het automatiseren van SM zijn systemen in het leven geroepen. Deze systemen worden ook wel *Skill Management Systems* (SMS) genoemd. Beck (2003) omschrijft de ontwikkeling van een dergelijk systeem. Hierbij wordt gebruik gemaakt van een nummerwaardering van nul tot vijf bij elke vaardigheid in de lijst. Deze lijst met vaardigheden wordt vervolgens door de werknemer ingevuld. Dit resulteert in een profiel van de werknemer waarin zijn vaardigheden en zijn beheersing van elke vaardigheid zijn vastgelegd.

Gedurende de ontwikkeling van het systeem identificeerde Beck (2003) enkele belangrijke condities waaraan voldaan moet worden om een dergelijk systeem een succes te maken. Zo is het risico aanwezig dat werknemers het confronterend vinden om een dergelijke lijst met vaardigheden in te vullen. Dit resulteert uit het feit dat een vaardighedenlijst eerder de gebreken van een werknemer benadrukt dan de vaardigheden die hij wel beheerst. Daarnaast werd ook het belang van een goede usability voor het systeem benadrukt.

Garro en Palopoli (2003) beschrijven een vergelijkbare aanpak voor de opzet van een dergelijk systeem. Ook in dit systeem wordt gebruik gemaakt van een zogenaamde *skillmap*. In het beschreven systeem is het beheren van deze skillmaps een van de belangrijke bewerkingen die het systeem moet kunnen uitvoeren. Daarnaast wordt ook de functionaliteit van het analyseren van de *"gap"* tussen de vereiste en beschikbare vaardigheden van een werknemer genoemd. Er wordt zelfs een stap bovenop gedaan door een geavanceerde feature voor te stellen. Het betreft het aanbieden van verwijzingen naar *Learning Objects* die het geanalyseerde gap zouden kunnen dichten. Met een learning object wordt een bron van kennis bedoeld waar de benodigde vaardigheid mee kan worden opgedaan. In het artikel wordt een Multi-Agent oplossing voorgesteld voor het automatiseren van deze functionaliteiten.

In essentie is het gebruik van een skillmap dezelfde aanpak als die binnen het CoE wordt toegepast. Hier wordt het gebruikt om de kennis van werknemers vast te leggen. Vervolgens wordt het in de kenniskaarten uiteengezet ten opzichte van de projecteisen. Verder zijn er ook commerciële tools beschikbaar voor SM. Een voorbeeld van een dergelijk systeem is de skills tracking in TribeHR (2015).

Tot slot bestaan er ook systemen voor het managen van competenties (*Competency Management Systems*). In essentie doen deze systemen iets vergelijkbaars als SMS echter betreft het dan competenties in plaats van vaardigheden.

Usability

Eerder werd genoemd dat voor het succes van een SMS usability hoog dient te liggen. Een hoge usability maakt het voor de gebruiker fijner om gebruik te maken van de applicatie. Het is van belang om met het ontwerpen van de applicatie voor Sogeti rekening te houden met usability. Dit kan in het systeem van invloed zijn op de effectiviteit van de SM, zowel positief als negatief.

Usability wordt vaak geassocieerd met gebruiksvriendelijkheid van een interface. Nielsen Norman Group (2012) definieert usability als zijnde een kwaliteitsattribuut voor het gemak waarmee een interface gebruikt kan worden. Verder bestaat usability op zich volgens de Nielsen Norman Group uit vijf kwaliteitscomponenten. Deze componenten luiden als volgt:

1. Learnability
2. Efficiency
3. Memorability

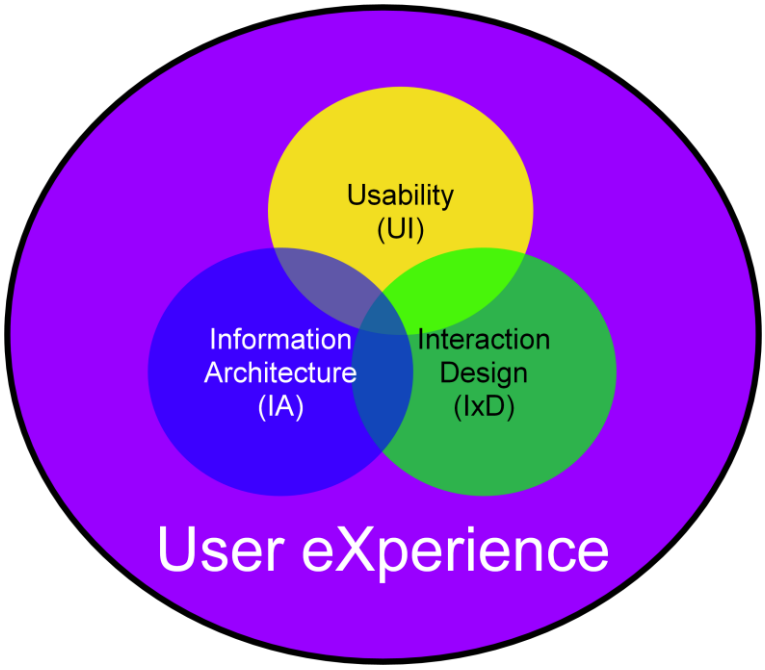
4. Errors
5. Satisfaction

Hierin is *learnability* de maatstaaf voor het gemak waarmee een gebruiker een taak voor de eerste keer uitvoert. *Efficiency* geeft weer hoe snel een gebruiker een taak kan uitvoeren wanneer ze het ontwerp van de applicatie hebben geleerd. Met *memorability* wordt het gemak waarmee een terugkerende gebruiker de applicatie weer onder de knie kan krijgen bedoeld. *Errors* beschrijft het aantal fouten dat gebruikers maken en hoe gemakkelijk ze hier van herstellen. Tot slot geeft *satisfaction* weer hoe fijn het ontwerp voor de gebruikers is om te gebruiken.

Naast een definitie van usability geeft de Nielsen Norman Group (2012) ook redenen waarom usabillity van belang is voor websites. Een slechte usability zorgt er voor dat bezoekers van websites de site verlaten. Voor intranet en dus ook interne applicaties ligt dit net anders. Hier betreft het vooral productiviteit van de gebruikers. Een hogere usability zorgt er voor de er minder tijd nodig is voor de gebruiker om zijn taak te voltooien.

User Experience

User Experience [UX] is een eigenschap van een product waar het bij kan staan of vallen. Het wordt omschreven door Garrett (2011) als zijnde de ervaring die een product creëert voor gebruikers in de werkelijke wereld. UX gaat niet zozeer over de functionaliteiten van een product, maar om de ervaring die een gebruiker met een product heeft. Volgens de meeste UX-experts is UX een breder begrip dan usability, zoals weergegeven in figuur 2 (Jongman, 2014). Dit houdt in dat Usability een onderdeel is van de ervaring van een gebruiker.



Figuur 2. Relatie UX en usability

3 Opdracht

De daadwerkelijke opdracht zal hier worden toegelicht. Allereerst zal de probleemstelling worden beschreven. Hierin worden de problemen van de huidige situatie uitgelicht. Vervolgens zal de doelstelling die Sogeti wilt bereiken met het project worden beschreven. Tot slot zal een concrete opdrachtomschrijving worden aangeleverd waarin de opdracht tot in detail zal worden beschreven.

3.1 Probleemstelling

Het probleem dat de applicatie op dient te lossen is meervoudig. Hierin zijn de volgende aspecten te herkennen:

- Het handmatig toevoegen en beheren van kenniskaarten in Excel
- Dezelfde data bevindt zich dubbel in meerdere systemen
- De requirements van de applicatie zijn niet duidelijk gedefinieerd

Ten eerste is het handmatig toevoegen en beheren van kenniskaarten een overduidelijk probleem. Dit proces is inefficiënt en de data is moeilijk te onderhouden. Dit maakt het lastig om er voor te zorgen dat de data bruikbaar is. Het systeem zal effectiever zijn als de data actueel is. Hierdoor wordt de meest accurate representatie mogelijk van de kennis waar werknemers over beschikken. Het feit dat het een Excel bestand betreft heeft er volgens E. Hendriksen (persoonlijke communicatie, 19 maart, 2015) voor gezorgd dat er verschillende versies van de kenniskaarten circuleren. Dit maakt het lastig om eenduidige data te hebben.

Het tweede probleem is het hebben van de data in meerdere systemen. De data die in de verschillende systemen, zoals in het BIS staan, is vrij overeenkomstig. De medewerkers worden gepresenteerd met daarbij een lijst kennisgebieden en een score die de werknemer hieraan heeft verbonden. Omdat deze data in verschillende systemen staat, maakt dit het wederom moeilijk om de data te onderhouden. Dit kan verder tot gevolg hebben dat data inconsistent is. Op zijn beurt kan dit problemen opleveren voor entiteiten die assumpties maken die gebaseerd zijn op de betreffende gegevens. De afdeling Planning zou aan de hand van niet actuele gegevens een werknemer bijvoorbeeld niet kunnen plaatsen bij een opdracht. Dit kan als consequentie hebben dat het bedrijf geld misloopt.

Tot slot doet zich nog een probleem voor. Dit is een gevolg van het feit dat het systeem uiteindelijk bij meerdere Business Lines ingezet kan gaan worden (dus bijvoorbeeld ook Mobile of Microsoft). De requirements voor het systeem zijn slechts gedeeltelijk concreet. Er dienen duidelijke, concrete requirements te komen om de bruikbaarheid van het systeem te kunnen borgen. Wanneer dit niet gebeurt, zou het systeem mogelijk niet op meerdere Business Lines inzetbaar kunnen zijn.

3.2 Doelstelling

De belangrijkste doelstelling die Sogeti met dit project wil bereiken is het verkrijgen van een geautomatiseerd, centraal systeem wat ook buiten het CoE gebruikt moet kunnen worden. Het systeem moet het makkelijker maken om nieuwe kennisgebieden en medewerkers toe te kunnen voegen en dus het geheel flexibeler maken. Wanneer deze aspecten gerealiseerd zijn is het voor Sogeti mogelijk om het systeem buiten puur en alleen het CoE toe te passen. Dit maakt op zijn beurt verdere integratie met

andere gebruikte systemen van Sogeti mogelijk en relevant. Al met al is het achterliggende strategische doeleinde dat het systeem project management efficiënter moet gaan maken.

Uit de beschrijving van de opdracht (zie sectie 3.3) is af te leiden dat het een productopdracht betreft. Uiteindelijk is het project voor Sogeti een succes als er aan verschillende condities wordt voldaan. Eén van de eerste condities is het hebben van een accurate en grondige requirementanalyse. Vervolgens dienen de met de product owner afgestemde functionaliteiten geïmplementeerd te worden in een applicatie. De correcte werking van het systeem moet aantoonbaar zijn en ook dient de architectuur van het systeem uitbreidbaarheid mogelijk te maken. Per slot van rekening levert de opdracht een stuk software op waar op doorontwikkeld kan gaan worden.

3.3 Opdrachtomschrijving

De opdracht die tijdens het afstudeertraject zal worden uitgevoerd heeft betrekking tot het automatiseren van het bestaande kenniskaartenproces. Dit proces verloopt zoals eerder vernoemd handmatig. Het systeem dat gerealiseerd zal worden, moet de huidige procesgang automatiseren en centraliseren. De wensen vanuit het CoE zijn om dit systeem in de vorm van een webapplicatie of Confluence-plugin te hebben. In eerste instantie zal een webapplicatie worden opgezet waarna uitbreiding naar plugin mogelijk is.

Confluence is het KMS wat wordt gebruikt binnen Sogeti. Het is een applicatie waarmee één centrale plek wordt gecreëerd voor het delen van informatie binnen een bedrijf. Verdere informatie over Confluence is te vinden op de officiële website (https://www.atlassian.com/software/confluence).

Kenniskaarten

In figuur 3 is een voorbeeld te zien van de huidige kenniskaarten. Een kenniskaart is een complexe Excel sheet bestaande uit verscheidene macro's. Een kenniskaart wordt opgesteld voor één project waaraan eisen worden opgesteld door de klant. Deze eisen zijn terug te zien als getallen bij de competenties. De vaardigheden en competenties zijn aan de hand van categorieën geordend, bijvoorbeeld "Algemene kennis en vaardigheden". Vaardigheden zijn geschaald van één tot vijf, waarin één het laagst en vijf het hoogste niveau van beheersing is. De beheersingniveau's zijn in de kenniskaarten terug te zien als kolommen van de tabellen. Daarnaast kan er met de kleuren rood, groen en blauw worden gespecificeerd of een bepaalde expertise een minimumvereiste, afgesproken gemiddelde respectievelijk optioneel is.

Ter voorbeeld: een opdrachtgever kan bijvoorbeeld aangeven dat er twee ervaren mensen nodig zijn met verstand van MongoDB. Dit zou onder het kopje "kennis vereist" te zien zijn als een twee (voor het aantal personen) onder kolom drie (voor kennisniveau ervaren). Aan een project kunnen medewerkers worden toegevoegd. De kennisniveau's worden per vaardigheid ingeschat en ingevoerd voor de bijbehorende medewerker. De cumulatieve kennis die door alle werknemers beschikbaar is, is te zien onder het kopje "kennis beschikbaar". Daarnaast is het in het overzicht van het project zichtbaar wat de skill gaps zijn. Er wordt berekend of de benodigde expertise aanwezig is, het resultaat hiervan is zichtbaar onder het kopje "kennis gap".

Algemene kennis en vaardigheden	Kennis vereist						Kennis beschikbaar						Kennis gap					
niveau :	1	2	3	4	5	Σ	1	2	3	4	5	Σ	1	2	3	4	5	Σ
Cobol		2	3	1		6	1					1	1	-2	-3	-1		-5
.NetCobol		1	2			3			1			1		-1	-1			-2
CICS																		
Mainframe		3	2		1	6			1			1		-3	-1		-1	-5
IDMS		2		3		5				1		1		-2		-2		-4
SQL Server 2000	2					2	1					1	-1					-1
TSO/ISPF		1				1		1				1						
JCL			3			3			1			1			-2			-2
Gestructureerd programmeren				4		4				1		1				-3		-3
SDM					3	3					1	1					-2	-2
testen			2	1		3	1					1	1		-2	-1		-2
Operationeel beheer																		
Endevor																		
Rational ClearCase Explorer		1	2		1	4			1			1		-1	-1		-1	-3
MS Visual Studio																		
FO/TO																		
PC/Windows																		
Office									1			1			1			1
OPC									1			1			1			1
OPALIS																		
Power BSort																		
OLQ																		
Infoman																		
SDFII																		
ADS																		
Assembler																		
REXX																		
CA-11																		

Figuur 3. Voorbeeld kenniskaart

Efficiëntie

Het systeem dient efficiënter te zijn dan de huidige oplossing. Uit vooronderzoek bleek Usability en User Experience [UX] (zie sectie 2.3) hier een rol in te kunnen spelen. Het systeem dient dus een goede Usability en UX te hebben. Echter is nog niet duidelijk hoe dit in het product geborgd kan worden. Er zal gekeken moeten worden naar manieren om dit tijdens het ontwerpen van het systeem mee te nemen. Uiteindelijk dienen deze aspecten van het systeem ook aantoonbaar goed te zijn.

Privacy

Aangezien het hier data van werknemers betreft die niet per se iedereen moet kunnen inzien is privacy een punt van aandacht. Het is belangrijk om van te voren een duidelijke definitie te hebben van welke data wel of niet voor andere werknemers inzichtelijk moet zijn. Dit zal afgestemd moeten worden met de opdrachtgever. De opdrachtgever heeft de duidelijkste visie bij het product en dit maakt hem dan ook de aangewezen persoon om verheldering te verschaffen over de privacy van de gegevens.

Functionaliteiten

Naast het automatiseren van het huidige proces moet het systeem het ook nieuwe functionaliteiten bieden die zich momenteel niet in het handmatige proces bevinden. Een voorstel vanuit de organisatie was bijvoorbeeld het kunnen maken van projectevaluaties. De kern van functionaliteiten van het systeem zien er uit als volgt:

- Kenniskaarten definiëren
- Projecten definiëren
- Werknemers toevoegen
- Gapanalyse maken

Ten eerste de functionaliteit “kenniskaarten definiëren”. Deze functionaliteit is duidelijk te herkennen als kern van het systeem. Dit heeft dan ook hoge prioriteit. Het definiëren van kenniskaarten houdt in dat er

nieuwe kenniskaarten aan het systeem kunnen worden toegevoegd die vervolgens door de gebruikers kunnen worden ingevuld. Bij het definiëren van een nieuwe kenniskaart kan bijvoorbeeld worden gedacht aan het toevoegen van een kenniskaart om een project te managen.

Ten tweede is de functionaliteit "projecten definiëren" een belangrijk onderdeel van het systeem. Een kenniskaart op zich is natuurlijk niet zeer relevant. Door een project aan de kenniskaart toe te kunnen voegen is het mogelijk om hier bepaalde eisen met betrekking tot kennis aan te hangen. Deze eisen zijn te vertalen naar de individuele vaardigheden en competenties. Verder kunnen deze eisen niveaus en het aantal personen met het betreffende niveau vastleggen.

De derde belangrijke functionaliteit is "werknemer toevoegen". Deze functionaliteit maakt het mogelijk om een werknemer toe te voegen aan het systeem. Bij het toevoegen moeten de competenties en vaardigheden van een werknemer vastgelegd kunnen worden. Dit deel van het systeem betreft dus Skill en Competency Management. Door deze data vast te leggen is het globaal inzichtelijk wie over welke kennis beschikt. Als een medewerker in het systeem is toegevoegd moet deze aan een project gekoppeld kunnen worden.

Tot slot volgt de functionaliteit "gapanalyse maken". Als er medewerkers zijn gekoppeld aan een project in een kenniskaart is het mogelijk om te berekenen hoe de bezetting is. Door te kijken naar de kennisniveaus die de werknemers hebben en deze met de projecteisen te vergelijken is dit te bereiken. Hiermee moet een overzicht gemaakt kunnen worden waar kennisoverschotten- en tekorten zijn.

Deze functionaliteiten zijn reeds vastgesteld, echter is de rest van het systeem open voor interpretatie. Een deel van de opdracht zal uit het maken van een requirementanalyse bestaan. Het bedrijf heeft nog geen volledig antwoord op de vraag wat voor functionaliteit het systeem moet gaan bieden om de gewenste rol te vervullen. Hierin zullen de stakeholders van het project geïdentificeerd moeten worden. Wanneer de stakeholders geïdentificeerd zijn is het mogelijk om in overleg tot een concrete lijst requirements te komen voor het systeem. Deze requirements zijn in te delen aan de hand van prioriteit van elke functionaliteit. Door deze analyse op te stellen zijn de eisen en functionaliteiten van het systeem concreet te maken.

4 Vraagstelling

Dit hoofdstuk zal de vragen benoemen en omschrijven die tijdens het onderzoek leidend zijn. Allereerst zal de centrale vraag worden benoemd en toegelicht. Vervolgens zullen de bijbehorende deelvragen worden beschreven.

Uit de probleemstelling is een centrale vraag af te leiden. De vraag dient het gehele probleem te omvatten en alle aspecten te dekken. Deze centrale vraag luidt als volgt:

Op welke manier kan het kenniskaartensysteem van Sogeti's Java Centre of Excellence het beste geautomatiseerd worden, zodat Project Management efficiënter verloopt?

Deze vraag vangt de doelstelling die Sogeti heeft voor het te ontwikkelen systeem. Het huidige proces is tijdrovend en de data staat verspreid over verschillende systemen. Een efficiënter systeem betekent in deze context dat het handmatige proces wordt geautomatiseerd. Daarnaast is de wens van het krijgen van een uniforme oplossing voor de huidige kenniskaarten hierin verwoord.

De centrale vraag is met gebruik van antwoorden op verscheidene deelvragen te beantwoorden. Deze deelvragen dekken elk een deel van het probleem. De deelvragen die samen de centrale vraag opmaken zijn als volgt gedefinieerd:

- *Hoe werkt het handmatige kenniskaartenproces van Sogeti's Java Centre of Excellence?*
- *Aan welke functionele- en niet functionele requirements moet het kenniskaartensysteem voor Sogeti's Java Centre of Excellence voldoen?*
- *Is het mogelijk om de huidige data in de kenniskaarten van Sogeti's Java Centre of Excellence in het automatische systeem mee te nemen en zo ja, op welke manier(en)?*
- *Op welke manier(en) kan User Experience en Usability bijdragen aan de efficiëntie van het systeem?*
- *Hoe kan op architecturaal niveau uitbreidbaarheid en toekomstbestendigheid van de kenniskaartenapplicatie worden bereikt?*

5 Methoden en Aanpak

Dit hoofdstuk beschrijft de methoden en aanpak die tijdens het afstudeerproject toegepast zullen worden. Dit omvat onder andere de onderzoeksmethoden die toegepast zullen worden voor het beantwoorden van de vraagstelling bij het onderzoek. Daarnaast zullen randzaken met betrekking tot communicatie en ontwikkelmethoden in dit hoofdstuk worden besproken. De globale aanpak voor het onderzoek is weergegeven in tabel 1.

Deelvraag	Vraagtype	Aanpak	Paragraaf
Hoe werkt het handmatige kenniskaartenproces van Sogeti's Java Centre of Excellence?	Beschrijvend	Interviews	5.1
Aan welke functionele- en niet functionele requirements moet het kenniskaartensysteem voor Sogeti's Java Centre of Excellence voldoen?	Beschrijvend	Interviews	5.2
Is het mogelijk om de huidige data in de kenniskaarten van Sogeti's Java Centre of Excellence in het automatische systeem mee te nemen en zo ja, op welke manier(en)?	Vergelijkend, Ontwerpend	Desk research Literatuur	5.3
Op welke manier(en) kan usability bijdragen aan de relevantie van de kenniskaarten?	Ontwerpend	Literatuur Desk research Field research	5.4
Hoe kan op architecturaal niveau uitbreidbaarheid en toekomstbestendigheid van de kenniskaartenapplicatie worden bereikt?	Ontwerpend	Literatuur Desk research Field research	5.5

Tabel 1. Vraag en aanpak

5.1 Huidig kenniskaartenproces

Voor het in kaart brengen van het huidige kenniskaartenproces zullen interviews worden toegepast. Er zullen gedurende het onderzoek meerdere interviews gehouden worden. Interviews zijn een onderzoeksmethode die geschikt is voor het verkrijgen van kwalitatieve informatie (Steehouder et al., 2006).

Deze interviews hebben als doel om het proces tot in detail in kaart te brengen. Dit is vereist omdat ditzelfde proces geautomatiseerd zal worden. Het is dus van belang dat dit proces helder is gemaakt. Om heldere en bruikbare data te verkrijgen is het van belang dat tijdens de interviews de juiste vragen aan de juiste personen worden gesteld. Van te voren zal dus een concrete vragenlijst en doelgroep moeten worden opgesteld.

Na het uitvoeren van de interviews zal de verkregen data verwerkt worden. De verwerkte informatie zal worden gevalideerd met persoon die is geïnterviewd. Deze informatie zal een procesgang opleveren van het huidige kenniskaartenproces. Dit betreft niet alleen huidige ondersteunde functionaliteiten, maar ook hoe deze zich onderling verhouden. Verder zal een inzicht in de business zijn verkregen waar in het ontwerpen van het systeem de voordelen zich zullen voordoen. Hier kan bijvoorbeeld worden gedacht aan aspecten als: welke data moet er in het systeem komen? Of welke entiteiten moeten deze data gaan bevatten?

5.2 Requirementanalyse

De requirementanalyse zal uit drie stappen bestaan. Deze drie stappen volgen elkaar logischerwijs op aangezien ze afhankelijk zijn van elkaar. De drie stappen zien er als volgt uit:

- Identificeren stakeholders
- Interviewen/ondervragen stakeholders
- Verkregen data inventariseren en verwerken

De eerste stap, het identificeren van de stakeholders, is cruciaal. Stakeholders zijn in deze context belanghebbenden bij het te ontwikkelen systeem. De lijst met stakeholders zal worden opgesteld in overleg met de product owner (Edo Hendriksen) en de technisch begeleider (Erwin de Gier). De product owner heeft een duidelijke visie van de richting waar het systeem heen zal gaan. Verder weet de product owner veel van de interne bedrijfsstructuur af wat hem de aangewezen persoon maakt om stakeholders te identificeren.

De tweede stap, die opeenvolgend op de eerste stap loopt, is het interviewen van de stakeholders. Zoals eerder genoemd is het voor interviews van belang de juiste mensen de juiste vragen te stellen. Aan de eerste criteria is in dit stadium al voldaan. Dan rest dus slechts het opstellen van de juiste vragen. Wanneer dit is gebeurd kunnen de stakeholders geïnterviewd worden. Deze interviews zullen als belangrijkste resultaat een lijst met requirements voor het systeem opleveren. Wederom zal de verwerkte informatie in samenwerking met de geïnterviewden worden gevalideerd.

De laatste stap in het opstellen van de requirementanalyse is het inventariseren en verwerken van de verkregen data uit de interviews. De functionaliteiten zullen in overleg met de product owner worden ingedeeld naar prioriteit. Dit maakt het plannen van de realisering van het systeem in een later stadium gemakkelijker.

Het indelen naar prioriteit zal gebeuren met behulp van de MoSCoW methodiek. Dit een relatief simpele methodiek die niet tijdsintensief is (Verma, 2010). De methodiek is ontwikkeld voor het gebruik in een van de voorlopers van hedendaagse Agile ontwikkelmethodes: Dynamic Systems Development Method (Ramsin & Paige, 2008). Dit aangezien het prioriteren van functionaliteiten helpt bij het indelen van iteraties. Deze techniek kenmerkt zich door de functionaliteiten in te delen in vier categorieën die beginnen met een medeklinker uit het woord (**M**o**SC**o**W**). De categorieën zijn luiden als volgt:

- Must-haves
- Should-haves
- Could-haves
- Won't-haves

De vier categorieën lopen in aflopende mate van prioriteit van functionaliteiten; Must-have heeft de hoogste prioriteit, won't have de laagste.

5.3 Legacy data

In de tijd waarin het kenniskaartenproces handmatig is uitgevoerd is veel data vergaard. Het zou zonde zijn om deze data niet in het geautomatiseerde systeem mee te nemen. De huidige data staat in de Excel sheets die door de jaren heen zijn opgebouwd. Dit vereist dat er gezocht moet worden naar manieren om deze data te importeren in het systeem.

De technieken waarmee dit bereikt kan worden zullen aan het licht worden gebracht door digitale bronnen te raadplegen en eventueel ook gebruik te maken van literatuur. Hier zal voornamelijk worden gezocht naar het importeren van Excel data en eventueel naar beschikbare libraries. Digitale bronnen zullen bij uitstek geschikt zijn aangezien daar vaak handleidingen en/of opsommingen van verschillende libraries te vinden zijn. Vervolgens kunnen de resultaten worden uiteengezet en worden vergeleken met behulp van vooraf opgestelde criteria.

5.4 Efficiëntie

Het beantwoorden van de deelvraag met betrekking tot efficiëntie door Usability en UX zal voornamelijk met behulp van literatuuronderzoek gebeuren. Dit is bij uitstek geschikt voor een onderwerp als Usability, aangezien hier reeds veel over is geschreven. De uitkomst van deze literatuurstudie is een set met best practices of design patterns. Deze ontwerptechnieken kunnen in het proof of concept worden toegepast om de usability in het product hoog te krijgen.

Overigens is het een mogelijkheid om gebruik te maken van het expertise van de UX-werknemers binnen Sogeti. Deze zouden de usability van de applicatie kunnen testen of feedback verschaffen. Ze zouden echter ook tijdens het ontwerpen al betrokken kunnen worden wanneer de middelen hiervoor aanwezig zijn.

5.5 Architectuur

Voor het correct opzetten van de architectuur zal uitvoerig onderzoek moeten worden uitgevoerd. De doelstelling (zie hoofdstuk 3.2) beschrijft duidelijk dat het systeem uitbreidbaar moet zijn. Dit vereist dat de architectuur zodanig moet zijn opgezet dat dit gemakkelijk te bereiken is.

Het antwoord op de vraag hoe deze uitbreidbaarheid op architecturaal niveau te bereiken zal met behulp van verschillende onderzoeksmethoden worden beantwoord. De onderzoeksmethoden die gebruikt zullen worden zijn een stuk literatuurstudie, desk research en field research.

Literatuur kan worden gebruikt om bijvoorbeeld bewezen oplossingen te zoeken voor het probleem. Dit zal waarschijnlijk een set met bewezen oplossingen voor een probleem opleveren in de vorm van een *design pattern*. Er zijn design patterns beschikbaar voor veel problemen en deze zijn in literatuur uitvoerig beschreven en vastgelegd.

Daarnaast kan er uiteraard naar bestaande applicaties en de structuur daarvan worden gekeken. Deze vorm van deskresearch kan voornamelijk mogelijke herbruikbare structuren of oplossingen op leveren. De oplossingen zouden vervolgens in het systeem kunnen worden toegepast.

Tot slot kan er een deel fieldresearch plaatsvinden. De technisch begeleider van de student is een software architect en bevindt zich op dezelfde locatie. Aangezien hij bedreven is in het ontwerpen van systemen kan hij worden benaderd voor input. Verder kunnen ideeën en uitwerkingen bij de technisch begeleider worden gevalideerd.

5.6 Proof of Concept

De afstudeeropdracht zal als eindresultaat een proof of concept (PoC) opleveren. Het proof of concept zal de kennis die tijdens het beantwoorden van de deelvragen is verzameld toepassen. Het doel van het PoC is uiteindelijk om alle aspecten samen te brengen in een werkend stuk software. Deze software dient om aan te tonen dat het toepasbaar is voor de doeleinden die Sogeti bij de applicatie heeft.

Het PoC zal iteratief ontwikkeld worden (zie hoofdstuk 8.5: softwareontwikkelmethode) en dus zullen er steeds kleine brokken van het systeem worden opgeleverd. Belangrijk voor het systeem is dat er met het ontwerpen rekening gehouden wordt met de uitbreidbaarheid. Dit in verband met het mogelijk breder inzetten dan puur en alleen in het CoE. Daarnaast dient de correcte werking en de usability van het PoC getest en aangetoond te worden (zie hoofdstuk 8.6: kwaliteitsbewaking).

Vanuit Sogeti zijn er enkele technieken waarmee gewerkt dient te worden. Zo is het gebruikelijk dat de applicaties een REST backend hebben. Deze backend wordt ontwikkeld met het Java EE ontwikkelplatform in de Eclipse IDE. Voor de frontend wenst Sogeti dat er wordt ontwikkeld met HTML, CSS en JavaScript (JS). Specifieker met *AngularJS*. Angular is een JS framework wat veel gebruikelijke handelingen abstraheert en vergemakkelijkt. Zo maakt het onder andere bidirectional databinding mogelijk waarmee een view gemakkelijk een model kan updaten. Meer informatie is te vinden op de site van AngularJS (https://angularjs.org/). Tot slot zal het geheel gebruik maken van *Maven* voor buildautomatisering. Maven maakt het onder andere gemakkelijk om projectdependencies te beheren. Voor meer informatie over Maven kan de officiële site worden geraadpleegd (http://maven.apache.org/).

Wat security betreft zal voor de authenticatie zal gebruik worden gemaakt van de LDAP van Sogeti. Dit is een manier om single sign-on over verschillende services te bewerkstellen (Gracion Software, 2015). De usernames en passwords zullen dus niet in het systeem worden opgeslagen, maar slechts tegen deze LDAP gevalideerd. De gebruiker moet geauthenticeerd zijn om het systeem te kunnen gebruiken.

Al met al is het PoC een belangrijk resultaat van de opdracht. Dit schept voor de organisatie een helder beeld van wat de mogelijkheden zijn en hoe de geleerde concepten toegepast kunnen worden in de praktijk. Tot slot zou het PoC naderhand verder doorontwikkeld kunnen worden tot een daadwerkelijke in de praktijk te gebruiken applicatie.

5.7 Communicatie

Communicatie is van groot belang tijdens het ontwikkelen van een systeem. Dit project kent vier personen waarmee door de student regelmatig communicatie plaats dient te vinden. Dit zijn de docentbegeleider, de bedrijfsbegeleider, de technisch begeleider en de opdrachtgever.

Esther van der Stappen vervult de rol van docentbegeleider. Er dient voornamelijk gecommuniceerd te worden over deliverables die vereist zijn vanuit het opleidingsinstituut. De

docentbegeleider is degene die beslist of de afstudeeropdracht haalbaar en van voldoende niveau is. Communicatie zal dan ook voornamelijk plaatsvinden via de mail. Verder zal de docentbegeleider (minimaal) één keer op bedrijfsbezoek komen bij de organisatie.

Willem van de Griendt zal tijdens het afstudeertraject de bedrijfsbegeleider zijn vanuit Sogeti. De bedrijfsbegeleider zal regelmatig op de hoogte moeten worden gesteld omtrent de voortgang van de student. Op deze manier is het mogelijk, indien noodzakelijk, om het project tijdig bij te sturen. De afspraken die zijn gemaakt met de bedrijfsbegeleider omvatten een tweewekelijkse bilateraal en een wekelijkse update via de mail.

Erwin de Gier is de technisch begeleider van het project. Communicatie met Erwin zal voornamelijk tijdens de ontwerp en implementatie van het systeem plaatsvinden. Dit zal voornamelijk ontwerpkeuzes of verheldering met betrekking tot implementatievraagstukken betreffen. Communicatie zal voornamelijk in den lijve plaatsvinden aangezien de student in de aanvangsfase zich op dezelfde locatie zal bevinden.

Tijdens de aanvang van het project bleek dat de eisen en stakeholders nog onduidelijk waren. In eerste instantie zou Erwin de opdrachtgever zijn voor het project. Er bleek echter dat Erwin niet zoveel met de applicatie te maken had als veronderstelt was. Dit leidde er toe dat in overeenstemming met de technisch begeleider en bedrijfsbegeleider een andere opdrachtgever is gekozen. De opdrachtgever van het project zal Edo Hendriksen zijn. Met de opdrachtgever zal vooral in het begin veel afgestemd moeten worden. Dit zal in de vorm van meetings en mail verlopen. Verder zal de voortgang van elke timebox worden gecommuniceerd met de opdrachtgever.

5.8 Softwareontwikkelmethode

Het ontwikkelen van het PoC zal gestructureerd plaatsvinden volgens een zogenaamde softwareontwikkelmethode. Deze methoden structureren verschillende stadia van het softwareontwikkelproces. Afhankelijk van de bekende criteria kan een keuze worden gemaakt tussen verschillende methodieken.

Van de criteria die hierbij bekend zijn, is er één verstrekt door de opdrachtgever. De opdrachtgever heeft aangegeven dat de implementatie iteratief dient te verlopen. Daarnaast zal het project voornamelijk door één persoon worden uitgevoerd. De gekozen methodiek dient flexibel genoeg te zijn om voor een klein ontwikkelteam toepasbaar te zijn.

Aangezien de software iteratief dient te verlopen leidt dit al gauw tot een softwareontwikkelmethode in de categorie *Agile* methodieken. Volgens Abrahamsson, Salo, Ronkainen en Warsta (2002) wordt Agile gekenmerkt door vier kernwaarden. Ook al zijn het eerste en het derde punt meer bedrijfsgericht, punt twee en punt vier zijn daarbuiten ook goed toepasbaar voor de doeleinden van het huidige project. De kernwaarden luiden als volgt:

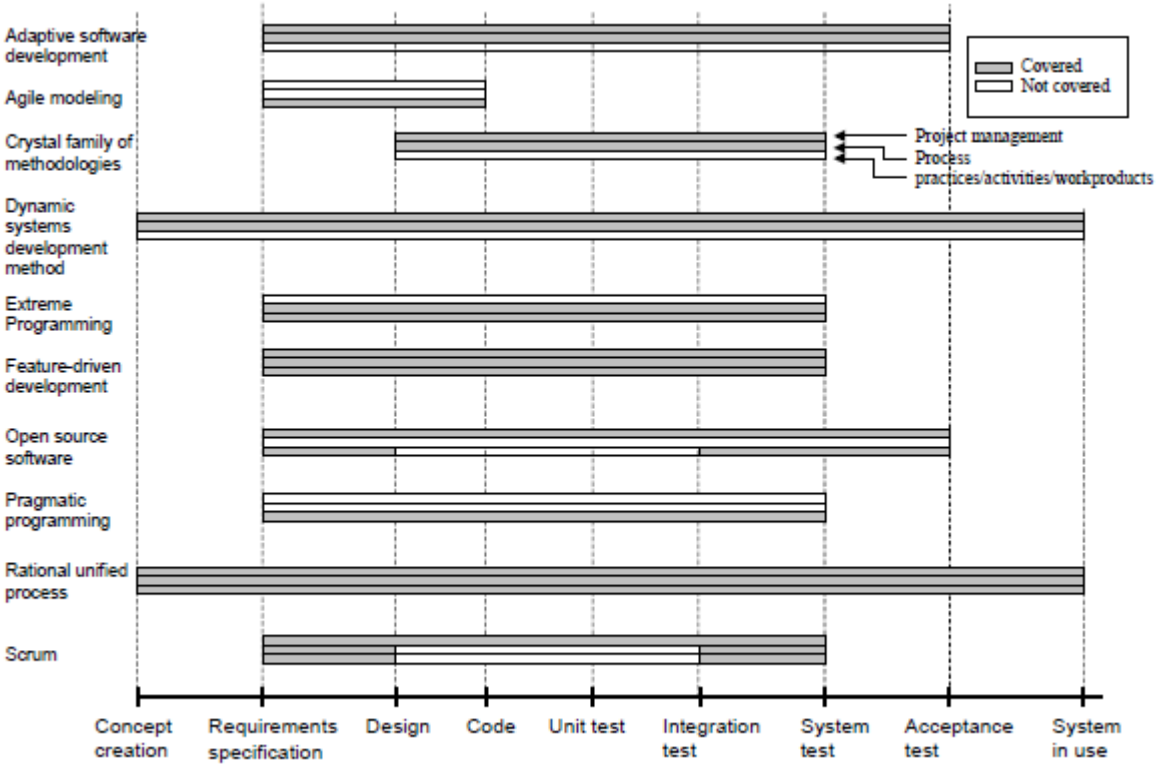
1. Individuen en interacties gaan voor processen en tools
2. Werkende software gaat voor uitgebreide documentatie
3. Samenwerking met de klant gaat boven contractonderhandelingen
4. Reageren op verandering gaat boven het volgen van een plan

Punt twee benadrukt in essentie het belang van het opleveren van werkende stukken software, hiermee krijgt de klant een beeld van de software. Regelmatige opleveringen van software maken tijdige

bijsturing mogelijk. Dit sluit aan bij de wens van de opdrachtgever om ontwikkeling iteratief te laten verlopen.

Het vierde punt beschrijft het feit dat er op verandering ingespeeld moet kunnen worden. Dit is een punt dat ook aansluit bij de context van het project. De verandering in de huidige context betreft dan echter het ontbreken een concrete lijst requirements voor het te ontwikkelen systeem. Daarnaast zou het in theorie mogelijk zijn om op verandering in prioriteit van functionaliteiten in te kunnen spelen. De iteratieve denkwijze van Agile maakt het mogelijk om hier snel op in te spelen.

De Agile denkwijze wordt toegepast in verschillende methodieken. Deze methodieken geven elk een eigen invulling aan toepassing binnen projecten. Tussen methodieken kunnen bijvoorbeeld regels, processen, documentatie en scope verschillen. Sommige methodieken definiëren een zeer uitgebreide set met regels en procesgang zoals RUP (Rational Unified Process) of Scrum en anderen zoals OSS (Open Source Software) dienen meer als guideline (Abrahamsson et al., 2002). Abrahamsson et al. (2002) zette tien Agile methodieken uiteen en gaf de verschillen met betrekking tot scope grafisch weer. Deze uiteenzetting is terug te vinden in figuur 4.



Figuur 4. Uiteenzetting Agile methoden

Als softwareontwikkelingmethode zal OpenUP worden gebruikt. OpenUP is een opensource versie van het door IBM beheerde RUP . Deze methodiek beschrijft een framework voor het gehele proces van softwareontwikkeling. Hieronder valt alles van verdieping tot daadwerkelijke integratie van de software. Het framework is aanpasbaar waar nodig en dit biedt flexibiliteit in de toepassing van de methodiek. Zo kan er voor worden gekozen om bepaalde rollen niet toe te passen of om invulling te geven aan de implementatie van de gedefinieerde fasen binnen OpenUP. RUP wordt niet als echt Agile gezien

(Abrahamsson et al., 2002) echter kan de Construction fase iteratief plaatsvinden waarmee de criteria van de opdrachtgever is vervuld.

Naast het feit dat de methodiek aanpasbaar is biedt het nog twee pluspunten ten opzichte van de andere methodieken. Allereerst wordt OpenUP beheerd door de Eclipse Foundation. Er zijn dan ook integratietools beschikbaar voor de IDE die gebruikt zal worden tijdens het project: Eclipse. Ten tweede is er reeds met OpenUP gewerkt, aangezien het onderdeel is van het curriculum van de opleiding Software Engineering aan het opleidingsinstituut.

OpenUP zal binnen het project toegepast worden door middel van het toepassen van de vier gedefinieerde fasen: Inception, Elaboration, Construction en Transition. De nadruk zal liggen bij de eerste drie fasen, aangezien de Transition fase het klaarmaken voor het uitbrengen van de applicatie naar gebruikers omvat. Het onderzoek levert een Proof of Concept op waardoor deze fase niet relevant is. De Inception fase is een globale oriëntatie in de business. Elaboration is het verdiepen in de eisen van het te ontwikkelen systeem. Construction is het daadwerkelijke implementeren van het systeem. De Construction zal iteratief plaatsvinden met behulp van opgestelde tijdseenheden genaamd "Time-boxes". Aan de hand van de projectrequirements die gedurende de Elaboration-fase op tafel komen krijgen de time-boxes invulling. Tijdens het project zullen time-boxes van twee weken worden aangehouden. Meer informatie over OpenUp is te vinden op de officiële wiki (OpenUp, 2012).

5.9 Kwaliteitsbewaking

Kwaliteitsbewaking is een belangrijk aspect van het projectverloop. Tijdens het afstudeertraject zal kwaliteitsbewaking op verschillende manieren terugkomen. Deze luiden als volgt:

- Functionaliteitstesten
- Usabilitytesten
- Code quality analysis

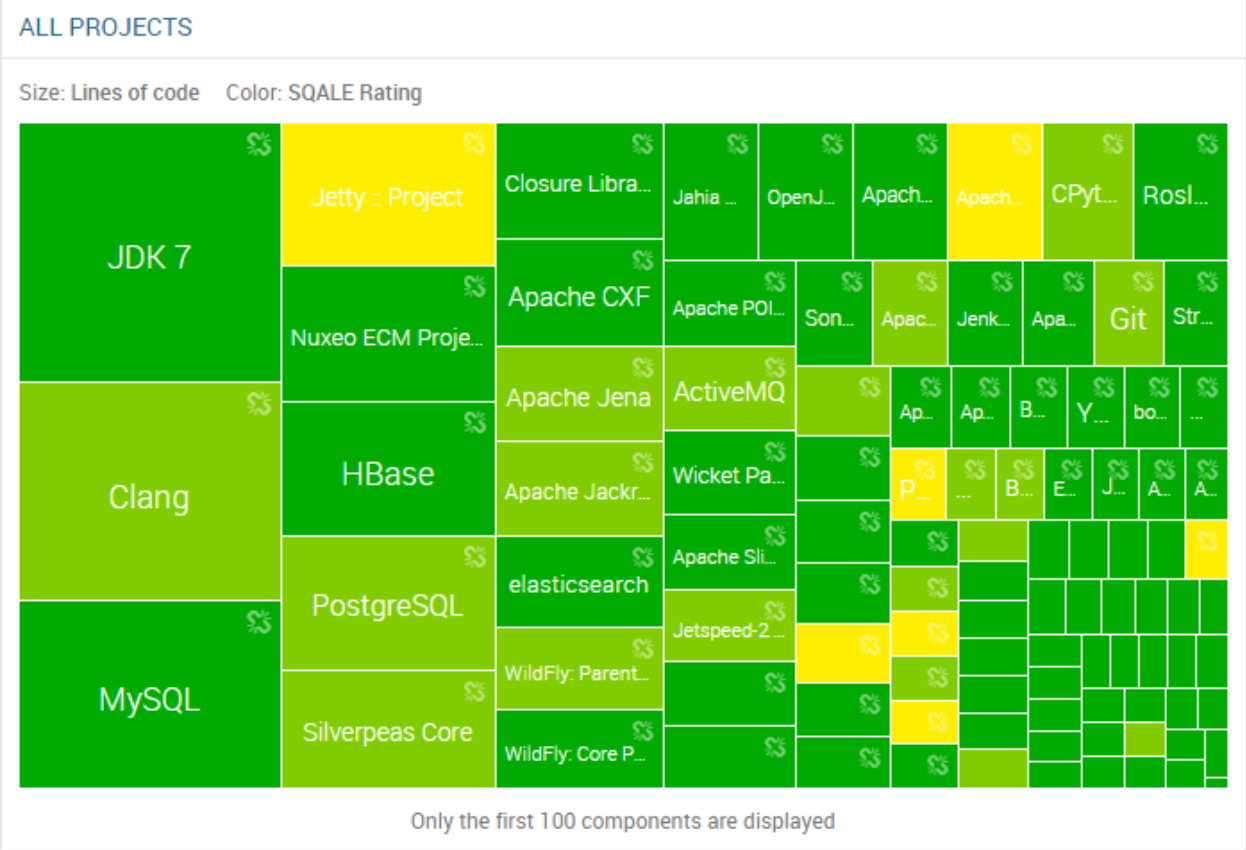
De functionaliteiten van het systeem dienen getest te worden. Er moet kunnen worden aangetoond dat het systeem correct werkt. Hiervoor zullen white box testen worden uitgevoerd. Dit zal worden gerealiseerd met *JUnit*. De keuze voor JUnit is gemaakt vanwege de hoge mate van integratie in de Java SDK, wat het gebruik vergemakkelijkt. Aangezien het implementeren van het systeem iteratief zal verlopen kan ook het testen in een vroeg stadium beginnen. Technisch begeleider E. de Gier (persoonlijke communicatie, 6 maart, 2015) heeft aangegeven dat het streven een code coverage van rond de tachtig procent is. Bullseye (2013) geeft aan dat het gebruikelijk is voor unit tests om een code coverage tussen de zeventig en mogelijk honderd procent te hebben. Echter blijkt uit bijvoorbeeld een internetpost (Google, 2010) dat het fixeren op een specifieke code coverage niet zo belangrijk is. Daarentegen is het van groter belang om zoveel mogelijk logica in de applicatie te testen. De metriek code coverage zal dus dienen als maatstaaf, maar niet als een "make or break" voor de applicatie.

Naast de waarborging van de functionaliteiten van het systeem, zal ook de usability van het systeem getest moeten worden. Nogmaals, het hebben van relevante data zal van groot belang zijn voor de effectiviteit van het systeem. Een applicatie met een goede usability zal de bruikbaarheid van het systeem ten goede komen. Het is dan voor werknemers prettiger om gebruik te maken van het systeem.

Om de usability te testen zal er gebruik worden gemaakt van gebruikerstests. Gebruikerstesten worden door de Nielsen Norman Group (2012) benoemd als manier om de usability op de proef te stellen.

De testen zullen kortweg inhouden dat er een testgroep opgesteld zal worden die met het systeem enkele interacties zullen uitvoeren. Vervolgens kan de gebruikers om hun ervaring en feedback worden gevraagd. De testgroep zal zoveel mogelijk bestaan uit mensen die het huidige systeem gebruiken en de applicatie zullen gaan toepassen. Door de gebruikerstesten gedurende de ontwikkeling te laten plaatsvinden kan op de feedback worden ingespeeld.

Tot slot zal de kwaliteit van de code worden geanalyseerd met behulp van de tool *SonarQube* (http://www.sonarqube.org/). Dit genereert een overzicht van de codebase die wordt geanalyseerd. Hiermee kunnen onder andere issues worden gedetecteerd aan de hand van een verzameling regels. Voorbeelden van deze regels zijn bijvoorbeeld naming conventions. Er zijn standaard regelsets beschikbaar voor verschillende talen. Een voorbeeld van een dashboard in SonarQube is weergegeven in figuur 5.



Figuur 5. Fragment SonarQube dashboard

De dashboards die gegenereerd worden geven een indruk van de algemene kwaliteit van de code en verbinden hier een *SQALE* (Software Quality Assessment based on Lifecycle Expectations) rating aan. Deze rating wordt gebruikt om de toekomstbestendigheid en onderhoudbaarheid van code uit te drukken. De methode definieert een aantal indices waar de code op geïnspecteerd moet worden. Bij deze indices kan bijvoorbeeld worden gedacht aan een index voor security, portability en efficiency. Meer over SQALE is te vinden op de officiële website (http://www.sqale.org/).

6 Resultaten

De resultaten van de in hoofdstuk 5 besproken onderzoeksmethoden zullen hier worden gepresenteerd. Elke paragraaf komt overeen met de resultaten voor een deelvraag (zie hoofdstuk 4) met hierbij de toegepaste onderzoeksmethoden (zie hoofdstuk 5). Daarnaast wordt het ontwikkelde Proof of Concept beschreven. Per paragraaf zal vervolgens een korte samenvatting volgen van de betekenis van de resultaten.

6.1 Huidig kenniskaartenproces

De interviews zijn gehouden na het uitvoeren van voorbereidende stappen. Het uitvoeren van deze stappen diende om het interview toe te spitsen op de data die gekregen moest worden. Voor het verkrijgen van de juiste informatie was het van belang om helder te krijgen welke informatie van welke personen verkregen moest worden. De betreffende stappen zijn weergegeven in figuur 6.

Figuur 6. Voorbereidende stappen interviewproces

De eerste stap was het opstellen van de te verkrijgen gegevens. Hierin werd gekeken welke informatie er nodig was om het huidige kenniskaartenproces in kaart te brengen. Er werd stilgestaan bij welke informatie over het proces reeds bekend was en welke niet. Daarnaast werd gekeken bij welke doelgroep deze informatie verkregen kon worden. Aangezien het huidige proces betreft waren gebruikers van de Excel-sheets de doelgroep voor het interview.

De tweede stap, het opstellen van interviewvragen, volgt logischerwijs op de eerste stap. De vragen zijn bedoeld om de gegevens die in de eerste stap zijn opgesteld boven water te krijgen bij de bepaalde doelgroep. De te verkrijgen gegevens en de bijbehorende opgestelde interviewvragen zijn weergegeven in tabel 2.

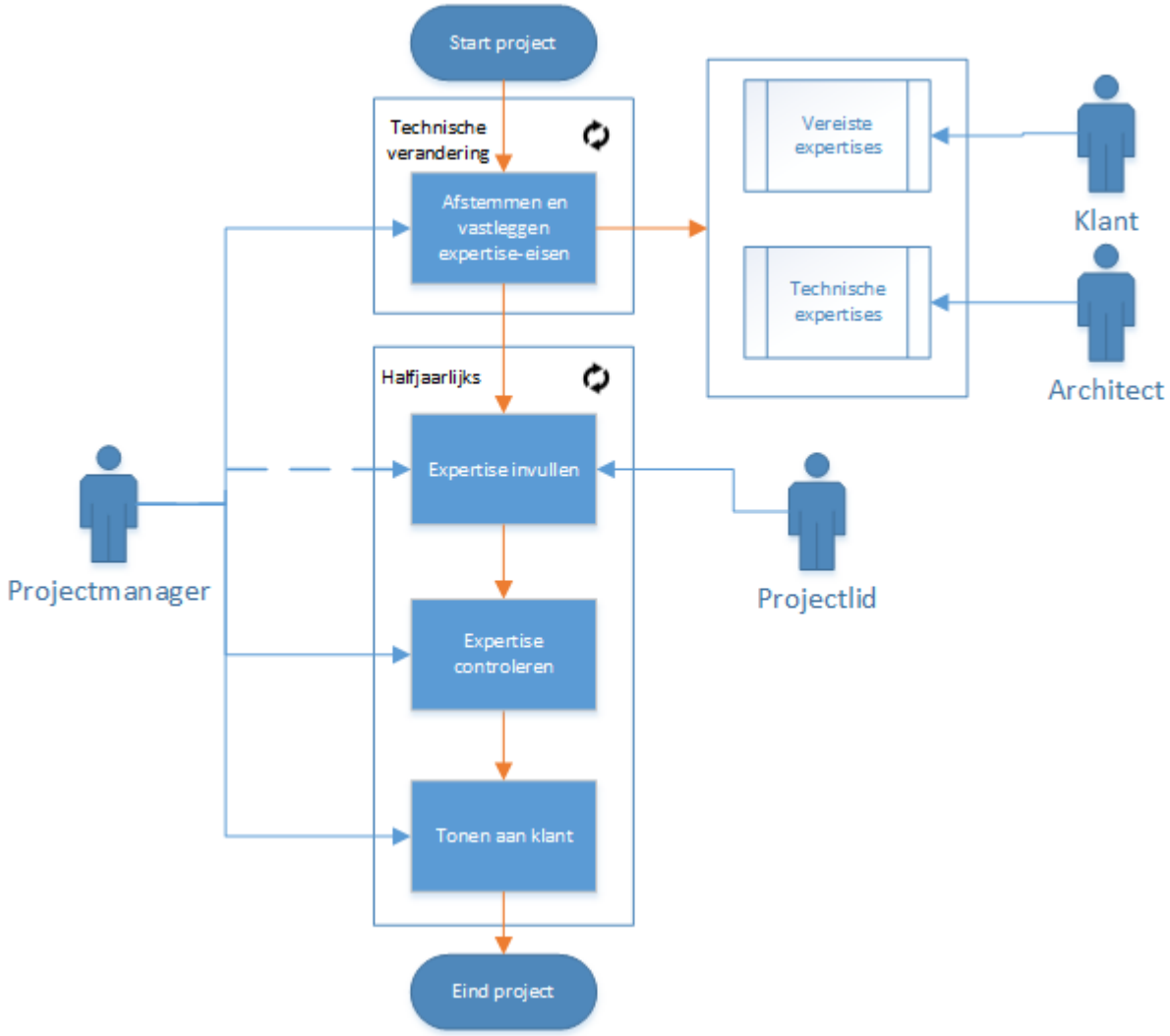
Te verkrijgen gegevens	Interviewvraag
Procesverloop bij het opstellen van een kenniskaart	Wat moet er gebeuren als er een kenniskaart voor een project moet worden opgesteld?
Procesverloop bij het invullen van medewerkerexpertises	Wat moet er gebeuren als de kennis van een werknemer in de kenniskaart vastgelegd moet worden?
De groep gebruikers die daadwerkelijke data invullen	• Welke gebruikersgroep/personen vult de vaardigheden van werknemers in? • Welke gebruikersgroep/personen vult de projecteisen in?
Bepaling en actualiteit expertises/categorieën	• Hoe worden de categorieën en expertises bepaald waar het kennisniveau van vastgelegd moet worden?

	• Wanneer en aan de hand waarvan worden deze categorieën en expertises die in de kenniskaarten staan bijgewerkt?
Raadplegers van de informatie, de frequentie en eventuele aard van raadpleging	• Welke gebruikersgroep/personen raadplegen de kenniskaart • Met welke doeleinden en frequentie doen zij dit?
Frequentie bijwerken kenniskaarten en bijbehorende aanleiding(en)	• Hoe vaak en met welke aanleiding worden de projecteisen in de kenniskaarten aangepast? • Hoe vaak en met welke aanleiding worden de kennisniveaus van werknemers in de kenniskaarten aangepast?

Tabel 2. Gegevens en interviewvragen kenniskaartenproces

De laatste stap was het afstemmen van de te interviewen personen. Deze personen dienden ervaring te hebben met het huidige kenniskaartenproces en dit ook daadwerkelijk gebruikt te hebben. Uit een persoonlijke communicatie met E. Hendriksen (19 maart, 2015) bleek dat kenniskaarten door servicecoördinatoren ingevuld worden. Eén van de personen die door de product owner werd voorgesteld als gebruiker van de huidige kenniskaarten was Christiaan Bennik. Christiaan is servicemanager bij Sogeti en dit maakt dat hij aansluit bij de doelgroep van de interviews. Het interview werd dan ook bij hem afgenomen. In verband met het tijdbestek voor het project is in overleg met de product owner besloten om het interview in eerste instantie slechts met Christiaan te houden. Dit was de enige gebruiker die bij de product owner op dat moment bekend was. Het houden van meerdere interviews zou dus vereisen dat andere gebruikers opgespoord moesten worden. De uitwerking van dit interview is terug te vinden in bijlage C.

Het kenniskaartenproces begint bij aanvang van een project met het afstemmen en vastleggen van de vereiste expertises. Dit gebeurt in eerste instantie met de klant. De technische expertises worden afgestemd met de architect. De vereiste expertises kunnen naar aanleiding van technische veranderingen worden gewijzigd. Wanneer de vereiste expertises zijn vastgelegd worden door de afdeling planning projectleden toegekend aan het project. Als dit is gebeurd kunnen de projectleden hun bijbehorende expertiseniveaus in gaan vullen. Vervolgens zal de projectmanager controleren of de projectleden dit realistisch hebben ingevuld. Tot slot wordt de kenniskaart aan de klant gepresenteerd. Het invullen en controleren van de kenniskaart gebeurt net als het presenteren van de kenniskaart ongeveer halfjaarlijks. Als een werknemer echter zelf aangeeft ergens meer bedreven in te zijn geworden dan kan hier een uitzondering op volgen. Dit proces is te zien in figuur 7.



Figuur 7. Procesverloop kenniskaart

Conclusie

Het kenniskaartenproces bestaat uit vier stappen:

1. Het vastleggen van de vereiste expertises voor een project
2. Projectleden leggen hun expertise vast
3. De manager controleert de ingevulde expertiseniveaus
4. Borging van kennis toonbaar maken aan de klant

Dit proces zal tijdens het ontwikkelen van het PoC als uitgangspunt dienen. Wanneer dit proces in grote mate wordt ondersteund in de applicatie dan zal het de huidige oplossing daadwerkelijk kunnen vervangen. Handelingen in het proces en hun bijbehorende actoren zijn door het verduidelijkt en dienen ook in de ontwikkeling van het PoC mee te worden genomen.

6.2 Requirements

De interviews waarmee de requirements konden worden opgesteld volgden dezelfde voorbereidende stappen als die beschreven in hoofdstuk 6.1. De vragen die zijn opgesteld waren dit keer toegespitst op het verduidelijken en vastleggen van de wensen die de stakeholders hebben. Het doel van

de vragen was om de functionele en niet-functionele eisen voor het systeem concreet te maken. De gegevens die nodig zijn om dit doel te bereiken waren eerst opgesteld, vervolgens werd de doelgroep bepaald. In dit geval bestond de doelgroep uit stakeholders. Dit waren zowel opdrachtgevers als gebruikers. Tot slot zijn de interviewvragen bedacht. De gegevens en bijbehorende vragen zijn weergegeven in tabel 3.

Te verkrijgen gegevens	Interviewvraag
De beoogde doelgroep en de aanleiding van hun interactie met het systeem	• Wie moeten er van het kenniskaartensysteem gebruik gaan maken ? • Met welke doeleinden doen zij dit?
Het beoogde doel voor het systeem	Wat zou je idealiter met het kenniskaartensysteem bereiken?
De functionaliteiten die het systeem moet gaan bieden	Wat moet het kenniskaartensysteem kunnen?
De prioriteit van de functionaliteiten	Geordend van hoge naar lage prioriteit: wat is de prioriteit van deze functionaliteiten?
Hoe het deze functionaliteiten moet gaan bieden	Op welke manier moet het kenniskaartensysteem deze functionaliteiten bieden? (Hierbij kan aan snelheid of security worden gedacht)

Tabel 3. Gegevens en interviewvragen requirements

De vragen die zijn opgesteld waren duidelijk gericht op het verkrijgen van een lijst functionele- en niet-functionele requirements. Echter was het zoals vernoemd van belang dat de juiste mensen werden geïnterviewd. Bij het maken van een requirementanalyse is dit van nog groter belang aangezien de personen invloed hebben op het eindresultaat. Het was dus belangrijk om een concrete lijst met stakeholders te hebben. In eerste instantie kwam hiervoor de belangrijkste stakeholder in aanmerking, Edo Hendriksen. Naast het interview met Edo, werd ook een interview gehouden met een eindgebruiker: Christiaan Bennik. Het interview met Christiaan diende vooral om een ander perspectief te bieden dan dat van de product owner. De uitgevoerde interviews zijn vervolgens vastgelegd en de antwoorden op de vragen zijn terug te vinden in bijlage G.

Interview Edo

Uit het interview met Edo Hendriksen bleek dat het systeem door drie typen gebruikers gebruikt zal worden. Dit zijn de managers, medewerkers en de beheerders. De manager wil met het systeem inzicht hebben in de expertises die op dat moment door de actieve projectleden worden aangeboden. Op deze manier kan hij reageren op een eventuele skill gap. De medewerkers worden opgedragen om de kenniskaarten in te vullen. Ze willen hiermee hun expertise kenbaar maken. Tot slot beheert de beheerder logischerwijs de applicatie.

Het interview bracht daarnaast een lijst met user stories aan het licht. Deze zijn ingedeeld naar prioriteit in een MoSCoW-lijst. Hierin waren de must-haves functionaliteiten die voornamelijk reeds in de Excel-versie voorkwamen. Dit is een gevolg van het feit dat de applicatie de Excel versie dient te vervangen. Vervolgens waren er veel geavanceerde functionaliteiten geïntroduceerd die een lagere prioriteit hebben. Deze functionaliteiten zijn ingeschaald naar aanleiding van hun prioriteit in de originele lijst met user stories. De functionele requirements zien er uit als volgt:

Must-have

- Inloggen met Sogeti gegevens
- Gebruikers toevoegen en rechten geven
- Kenniskaarten definiëren
- Benodigde expertises en expertiseniveaus vastleggen
- Medewerkers toevoegen
- Medewerker aan kenniskaart koppelen
- Medewerkerexpertise vastleggen
- Skillgapanalyse uitvoeren

Should-have

- Kenniskaarten delen
- Teams van medewerkers beheren
- Teams aan kenniskaarten koppelen
- Expertiseblokken beheren
- Expertiseblokken aan kenniskaarten koppelen
- Kenniskaarttemplate kopiëren

Could-have

- Kenniskaarten hiërarchisch definiëren
- Expertises hiërarchisch definiëren
- Confluence plugin
- Pregedefinieerde expertises selecteren
- Wijzigingen in expertiseniveau werknemer laten doorwerken in andere kenniskaarten
- Manager verwijderen

Would-have

- Expertises en medewerkers samenvoegen
- Expertiseniveaus medewerkers aanpassen en niveaucorrectie toepassen
- Expertises uit BIS importeren
- Expertiseniveaus uit BIS importeren

Tot slot zijn tijdens het interview twee niet-functionele eisen ter sprake gekomen: security en concurrency. Het systeem moet zich houden aan dezelfde veiligheidsstandaarden als dat van vergelijkbare systemen in het CoE. Dit houdt voornamelijk in dat gebruikers zich met hun Sogeti gegevens moeten authenticeren en dat de data in het systeem niet zomaar voor iedereen te raadplegen is. Wat concurrency betreft moet het mogelijk zijn voor verschillende managers om tegelijkertijd op dezelfde kenniskaarten te kunnen werken.

Interview Christiaan

Christiaan gaf in het interview aan voornamelijk zelf het systeem te gaan gebruiken. Hij zou het echter handig vinden als de kenniskaart voor de klant te in een read-only manier te raadplegen is. Daarnaast vond hij dat het mogelijk moet zijn dat de medewerkers zelf hun expertiseniveaus aanpassen. Tot slot zag hij mogelijkheid voor een externe review manager om gebruik te maken van het systeem.

De projectmanager zou volgens Christiaan het systeem gebruiken om te zien hoeveel projectleden en welke kennis er bij een project aanwezig zijn. Daarnaast noemde hij de mogelijkheid om het te gebruiken om mensen met bepaalde kennis te vinden. Christiaan wilt dus de expertise aan de klant

kunnen tonen en hij wil een overview van zijn project hebben. Daar wil hij expertises mogelijk kunnen aanpassen wat betreft niveau. Christiaan gaf aan hier een dashboard voor nodig te hebben, dit was volgens hem tevens de hoogste prioriteit. Hij wil verder op expertises en niveaus kunnen zoeken naar werknemers, dit vond hij volgend op het dashboard de belangrijkste functie. Tot slot werd aangegeven dat hij de expertiseniveaus van de werknemers niet per kenniskaart, maar per werknemer wilt hebben. Dit maakt het mogelijk reeds ingevulde expertiseniveaus mee te nemen naar andere kenniskaarten.

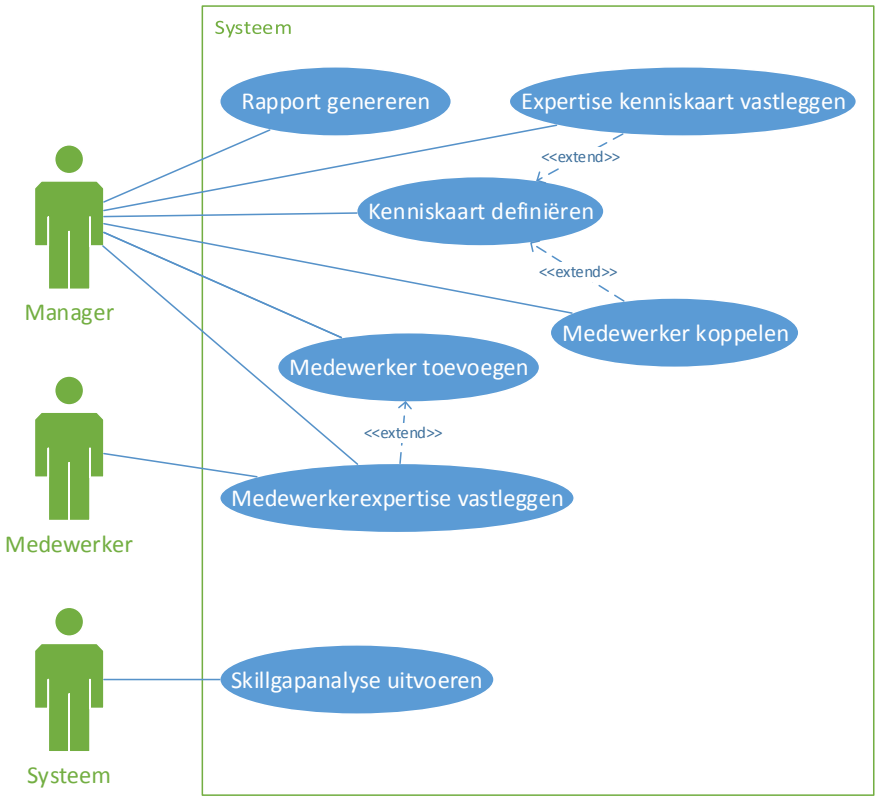
Aangezien de kenniskaarten bij zijn project slechts halfjaarlijks gebruikt worden, gaf Christiaan aan dat snelheid niet heel belangrijk was. Daarentegen noemde hij het belang van het feit dat de kenniskaarten volgens hem buiten het Sogeti-netwerk te raadplegen moeten zijn. Tot slot benoemde hij het belang van een goede security in de applicatie.

Conclusie

De interviews hebben verduidelijking geschapen wie de gebruikers van het systeem zijn en welke doeleinden ze bij het gebruik hebben. De projectmanager is de hoofdgebruiker en hij zal het systeem gebruiken voor het verkrijgen van inzicht in de beschikbare expertise bij een project. De medewerker wil zijn expertise kenbaar maken door deze in te vullen. De beheerder wil de applicatie kunnen beheren. Naast de User Needs (zie sectie 6.4) is het doel van de applicatie verhelderd. De belangrijkste doelen die met de applicatie bereikt dienen te worden zijn volgens de geïnterviewden:

- Het kenniskaartenproces centraliseren
- Functionele uitbreiding mogelijk maken
- Expertisedekking naar de klant aantoonbaar kunnen maken

Verder zijn de functionele eisen verhelderd. Hierin hebben de in de Excelsheet bestaande functies de hoogste prioriteit. Hierbij is naar aanleiding van het interview met Christiaan het genereren van een read-only view voor de klant een must-have geworden. De use cases met must have prioriteit zijn weergegeven in figuur 8. Tot slot zijn ook de niet-functionele eisen van het systeem duidelijk gemaakt. Voornamelijk security en het mogelijk maken van gelijktijdig werken in kenniskaarten bleek van belang te zijn.



Figuur 8. Must haves

Deze functionele en niet-functionele eisen zullen meegenomen worden tijdens het uitvoeren van de timeboxes in de Construction fase. Hierin zullen de user stories worden uitgewerkt en aan de hand van de prioriteit en afstemming met de product owner worden geïmplementeerd. Daarnaast zal er met het ontwerpen van het systeem rekening worden gehouden met de user stories die lagere prioriteit hebben. Het systeem moet de mogelijkheid hebben om deze functionaliteiten in een later stadium te kunnen ondersteunen.

6.3 Legacy data

Het zou een pre zijn als de huidige kenniskaarten in Excelvorm overgezet konden worden naar de applicatie. Er moest dus worden gezocht naar manieren om data uit een Excel sheet in een Java applicatie te kunnen benaderen. Hiervoor werd gezocht naar Java libraries die de vereiste mogelijkheden bieden, het importeren van data uit Excel sheets in .xls formaat. Het was hiervoor van belang dat criteria werden opgesteld om libraries op te kunnen vergelijken. Alternatieven konden aan de hand van de criteria worden uiteengezet en onderling worden vergeleken. Vervolgens was het aan de hand van deze uiteenzetting mogelijk een keuze te maken tussen verschillende alternatieven.

Criteria

Er waren bij dit onderwerp verschillende criteria van belang. Zo was het belangrijk dat de libraries goed gedocumenteerd waren. Dit maakt het mogelijk om implementatie te vergemakkelijken door een referentiepunt te hebben. Ook kan er worden gekeken of een library actief wordt ondersteund. Actieve support en dus actuele releases zijn hierbij positief. Daarnaast was het van belang dat het gebruik van de library intuïtief en niet overmatig complex is. Hiervoor werd steeds naar een voorbeeld gezocht waarin data uit een sheet werd uitgelezen. Dit voorbeeld werd gebruikt, omdat dit het dichts bij de

daadwerkelijke toepassing komt. Implementatiegemak was dus het tweede criterium. Verder kunnen de libraries verschil hebben in de ondersteunde bestandsformaten. Aangezien er verschillende versies van de kenniskaarten circuleren zou het bestandsformaat onderling kunnen verschillen. Hierbij kan worden gedacht aan het bewerken en exporteren van een bestand in een Excel alternatief wat in een andere extensie kan resulteren. Meer ondersteuning was dus een pluspunt, waar .xls en .xlsx vereist waren. Tot slot spelen de eventuele kosten van een oplossing een rol in de uiteindelijke afweging. Al met al worden dus de volgende criteria gehanteerd:

- Documentatie
- Support/Development
- Complexiteit
- Bestandsformaten
- Kosten

Alternatieven

Met behulp van desk research werden al gauw vele libraries gevonden op het gebied van Excel manipulatie in Java. De volgende libraries werden gevonden:

- **Apache POI** (Apache, 2015)
- **JExcelApi** (JExcelApi, 2015)
- **JExcel** (JExcel, 2015)
- **SmartXLS** (SmartXLS, 2015)
- **Aspose.Cells** (Aspose, 2015)

Elk van de alternatieven zijn in tabel 4 uiteengezet met behulp van de opgestelde criteria. Hierin zijn de attributen documentatie en implementatie ingedeeld van zeer slecht (--) tot uitstekend (++). De overige attributen zijn meetbaar en zijn in zijn geheel weergegeven. Voor het verzamelen van de gegevens omtrent deze criteria zijn de officiële websites van de libraries geraadpleegd. Deze zijn terug te vinden in de bronnenlijst.

	Apache POI	JExcelApi	JExcel	SmartXLS	Aspose
Documentatie	+	+-	++	--	+
Support/ Development (Laatste update)	28-02-2015	08-10-2012	30-08-2013	18-08-2014	13-03-2015
Implementatiegemak	+	+-	+-	+-	+
Bestandsformaten	.xls / .xlsx	.xls	.xls / .xlsx	.xls / .xlsx / .xlsb	.xls / .xlsx / .xlsm / .csv / .ods / SpreadSheetML / Tab Delimited
Kosten	Gratis	Gratis	$799 p. developer*	$299 p. developer	$999 p. developer

Tabel 4. Resultaten Java Excel Libraries

*Gratis bij open-source projecten

Conclusie

Het is dus mogelijk om met behulp van bestaande Java libraries data uit Excel sheets in te lezen. Dit maakt het mogelijk om de data uit bestaande kenniskaarten te importeren. Er zijn verschillende beschikbare libraries gevonden voor het beoogde doel. Voor de vergelijking is gekeken naar vijf criteria: documentatie, support/development, implementatiegemak, bestandsformaten en de kosten. Uit tabel 4 is op te maken dat vooral Apache POI en Aspose goede kandidaten zijn. Het verschil tussen de twee zit hem in de licentiekosten. Apache POI is open-source, daarentegen kost Aspose $999 per developer. JExcelApi ondersteunt geen .xlsx bestanden waardoor deze snel afvalt. SmartXLS en JExcel zijn complexer wat betreft implementatie. Al met al is Apache POI de beste optie, het biedt gedegen documentatie, wordt bijgehouden, het is redelijk intuïtief en ondersteund de voornaamste bestandsformaten. Tot slot is het feit dat het open-source is een welkome eigenschap. Wanneer het importeren in het PoC ondersteund zal worden zal Apache POI hiervoor worden gebruikt als gevolg van het feit dat het opensource is en voldoet aan de opgestelde criteria.

6.4 UX en Usability

Usability is een belangrijk aspect in skill- en knowledge managementsystemen. Dat heeft Beck (2003) geconcludeerd aan de hand van zijn implementatie van een Skill Management System. Het was dus van belang rekening te houden met dit belangrijke aspect van de applicatie. Volgens Garrett (2011) zorgt een goede UX voor een hogere efficiëntie. Dit gebeurt volgens hem op twee manieren in dergelijke systemen, namelijk:

- Het laat mensen sneller werken
- Mensen maken minder fouten

Er is literatuuronderzoek gedaan naar manieren om een hoge Usability en goede UX te krijgen in een webapplicatie. Deze sectie bouwt verder op het theoretisch kader (zie sectie 2.3). Er werd eerst gekeken wat invloed heeft op de usability van een webapplicatie. Vervolgens werd er gekeken naar manieren om een goede User Experience te bieden.

Invloedrijke factoren

Bruno, Tam en Thom (2005) beschrijven dat een webapplicatie bepaalde karakteristieken heeft die belangrijk zijn voor de usability van een webapplicatie. De definitie van usability die ze aanhouden is opgemaakt uit vier verschillende factoren: *gebruikers*, *taken*, *technologie* en *context*. Onder elke factor noemen ze vervolgens bijbehorende karakteristieken die invloed hebben op usabilityaspecten van een systeem.

De gebruikersfactor omvat meerdere aspecten. Deze factor beschrijft voornamelijk karakteristieken met betrekking tot het niveau van de gebruiker. Een gebruiker kan vallen in expertiseniveaus rangerend van beginner tot expert. Een hoger niveau van de gebruiker kan tot gevolg hebben dat *learnability* in de webapplicatie minder van belang is dan bijvoorbeeld efficiency. Daarnaast kan een karakteristiek als vormgeving een sterke invloed hebben op de *satisfaction* van de gebruiker.

De taak als factor beschrijft de impact die de uit te voeren werkeenheid met de applicatie kan hebben op de usability. Verschillende taken kunnen verschillende niveaus van complexiteit bevatten. Daarnaast kan tussen taken het type interactie met het systeem (denk aan formulieren invullen of het

gebruiken van menu's) verschillen. Dit alles kan invloed hebben op de attributen learnability, efficiency en satisfaction van de webapplicatie.

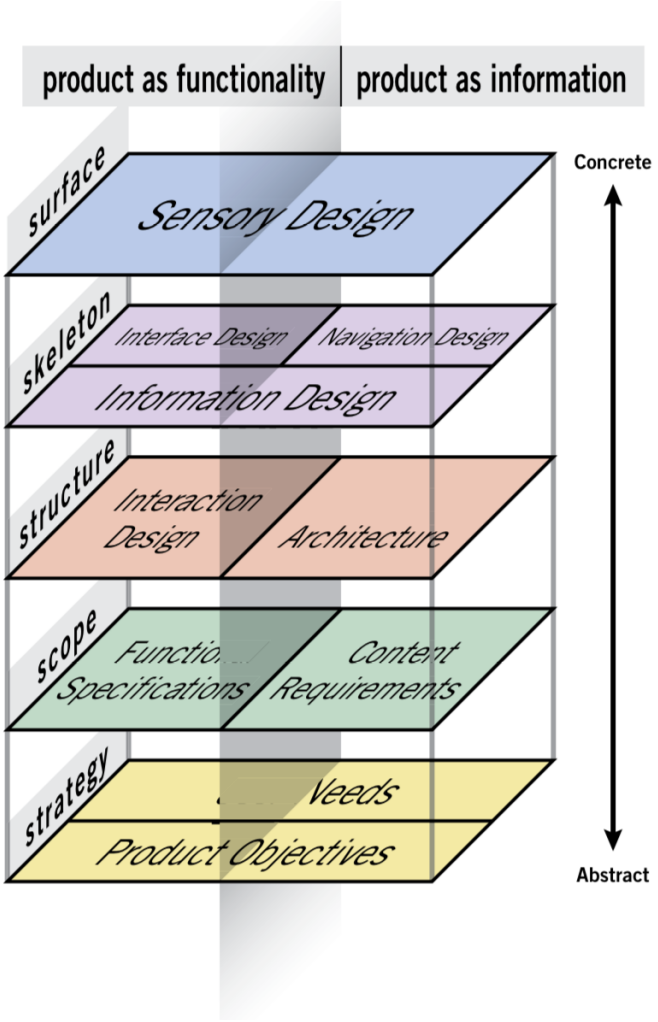
De technologie en context kunnen verder ook een rol spelen, al zijn ze wat minder relevant tijdens dit project. Een voorbeeld hoe technologie usability kan beïnvloeden is bijvoorbeeld in de vorm van een klein beeldscherm. Een beeldscherm met een lagere resolutie kan er bijvoorbeeld voor zorgen dat onderdelen van een website minder goed vindbaar zijn voor de gebruiker, wat logischerwijs kan leiden tot een lagere learnability. De context heeft volgens Bruno et al. (2005) betrekking tot bijvoorbeeld de aanpasbaarheid van een website aan de doelgroep. Het zijn aspecten waarmee een applicatie aan de stakeholders toegespitst kan worden.

UX en Usability design

Garrett (2011) beschrijft een uitgebreide strategie voor het borgen van User Experience in een website/webapplicatie. Hij benoemt het gebruik van vijf verschillende *planes* of oppervlakken (zie figuur 9). Elk van deze oppervlakken beschrijft zaken waarover nagedacht moet worden en keuzes die men moet maken. De lagen bouwen door op de keuzes die in de onderliggende niveau's zijn gemaakt. Wanneer dit model wordt toegepast wordt er gestructureerd gewerkt van abstract naar concreet. In dit geval betreft het dus vanuit het model gezien een bottom-up aanpak. Het model dwingt af dat er constant rekening wordt gehouden met wat van belang is voor de eindgebruiker. Dit resulteert in een applicatie waarin de bruikbaarheid hoog is. De vijf lagen zijn de volgende (van abstract naar concreet):

1. Strategy
2. Scope
3. Structure
4. Skeleton
5. Surface

Naast deze vijf lagen wordt er ook een distinctie aangebracht tussen twee vormen websites: de zogenaamde Functional en Informational sites. Een Functional website is in deze context een site waarmee een gebruiker een taak kan voltooien. Daarentegen is een Informational website een site waar gebruikers informatie kunnen vinden en hun eigen maken. Deze twee vormen van websites verschillen zodanig dat sommige stappen in het lagenmodel niet hetzelfde zijn voor beide vormen. In het PoC was het functionele deel duidelijk te herkennen, echter zal het ook een informationeel deel bevatten. Dit informationele deel zal bestaan uit de expertises en expertisevereisten.



Figuur 9. Elements of UX

De Strategy-laag is de meest abstracte laag in het model. Deze laag dient voornamelijk om de doelstellingen van het project te expliciteren, zowel van de opdrachtgever als de gebruikers. Het vastleggen van de strategie maakt het duidelijk waar de zwaartepunten van het project liggen. Tijdens het werk aan deze laag worden ook Persona's opgesteld. Dit zijn fictieve figuren die de gebruikersgroep representeren. Door deze karakters op te stellen is het mogelijk keuzes aan de gebruikersgroep te relativeren.

Ten tweede volgt de Scope-laag. In deze laag worden de doelstellingen uit de Strategy-laag omgezet in een concrete specificatie van het product. Het verkrijgen van de functionele requirements kan het beste worden bereikt door met gebruikers te praten. Met behulp van de eerder opgestelde Persona's kunnen zogenaamde Scenario's worden opgesteld. Dit zijn in essentie korte verhaaltjes waarmee kan worden aangetoond of gebruikersdoeleinden worden gedekt.

Vervolgens wordt in de Structure-laag de conceptuele structuur bedacht die uiteindelijk als tastbaar resultaat op de site terug zal komen. Hier worden de interacties die een gebruiker met de site kan hebben en hoe het systeem hierop reageert vastgelegd. Ook wordt de structuur van de informatie op de site vastgelegd.

In de Skeleton-laag wordt van de conceptuele structuur uit de structuurlaag gewerkt naar een tastbare interface bestaande uit verschillende componenten. Hierbij kan een Wireframe worden opgesteld. Dit is een ontwerpconcept waarmee kan worden gekeken of de onderliggende doeleinden

worden gedekt, maar het kan ook dienen als houvast tijdens de laatste laag. De laatste laag, de Surface-laag betreft alles wat zichtbaar is voor de gebruiker. Het is de samenkomst van functionaliteit, content en uiterlijk.

Conclusie

UX en Usability kunnen bijdragen aan de efficiëntie van het projectmanagementsysteem door de tijd en/of aantal fouten bij een uit te voeren taak te verminderen. Het maken van een systeem met een goede Usability vereist dat er met verschillende factoren rekening gehouden dient te worden. Deze factoren zijn voornamelijk de gebruikers, de uit te voeren taken, technologie en context (Bruno et al, 2005). Verder beschreef Garrett (2011) een model bestaande uit verschillende elementen. Het betreffende model zal tijdens het ontwikkelen van het Proof of Concept worden toegepast. Elementen uit het model zullen worden uitgewerkt waarna ze in de OpenUP reports vastgelegd worden, rekening houdend met de invloedrijke factoren die zijn vastgesteld.

Zo zullen er tijdens de Inception fase Persona's worden opgesteld die gebaseerd zijn op de gebruikersdoelgroep. Deze karakters zullen de gebruikersfactor concreet maken. Daarnaast zullen er Scenario's worden opgesteld aan de hand van de Persona's. Deze zullen de taakfactor verhelderen. Verder kunnen hier concrete user stories uit worden gehaald die stuk voor stuk geïmplementeerd kunnen worden. De interacties van de gebruikers met het systeem zullen ook in deze fase worden vastgelegd. Tot slot zal er tijdens de Elaboration fase Wireframes worden opgesteld voor de belangrijke schermen in de applicatie bij bepaalde entiteiten zoals kenniskaarten of projectleden. Hierbij kan worden gedacht aan het overzicht van entiteiten, het toevoegen van een nieuwe entiteit en het gedetailleerde overzicht bij een entiteit.

6.5 Architecturale oplossingen

Op architecturaal gebied zal het systeem met meerdere problemen rekening dienen te houden. Het systeem zal zoals gezegd uit te breiden moeten zijn naar andere Business Lines. Het onderliggende ontwerp van het systeem moet deze mogelijkheid bieden. Daarnaast is er een goede mogelijkheid voor de kenniskaartenapplicatie om gebruik te maken van bestaande systemen bij Sogeti, zoals bijvoorbeeld het Business Informatie Systeem [BIS]. Tot slot is de wens vanuit het CoE om de applicatie in Confluence te integreren. De frontend van de applicatie dient dus los van de backend te staan.

Inzetbaarheid diverse Business Lines

De inzetbaarheid over meerdere Business Lines is vooral afhankelijk van het domein van de applicatie, de daadwerkelijke klassen die in het systeem worden vastgelegd. Door dit domein niet direct vast te leggen aan concrete afhankelijkheden binnen de Business Line waar het systeem in eerste instantie voor wordt ontwikkeld is het mogelijk om dit in de toekomst uit te breiden. Er dient gebruik gemaakt te worden van bekende object-georiënteerde methodologieën zoals inheritence en composition (Gamma, Helm, Johnson & Vlissides, 1995). Door deze concepten abstract vast te leggen in het ontwerp van het domein wordt de implementatie niet afhankelijke van slechts het huidige domein.

Integratie externe systemen

Op het gebied van object georiënteerde software zijn vraagstukken op het gebied van uitbreidbaarheid een bekend fenomeen. Experts hebben hier in de loop der jaren oplossingen voor bedacht. De essentie van de oplossingen is vaak terugkerend, een "Pattern". Deze terugkerende

oplossingen worden in de softwareontwikkeling Design Patterns genoemd. Een Design Pattern is als het ware een template voor het oplossen van een ontwerpprobleem en is toepasbaar in verschillende situaties (Gamma et al, 1995). Het dient als een bewezen oplossing voor veel voorkomende ontwerpproblemen.

Enkele veelgebruikte Design Patterns zijn door Gamma et al. (1995) gedocumenteerd en gecategoriseerd. Hierin is een onderscheiding aangebracht tussen het doel en scope van de patterns. Het doel beschrijft wat een pattern doet. Hierin worden de categorieën "Creational", "Structural" en "Behavioral" aangedragen. Creational heeft betrekking tot ontwerpproblemen omtrent de creatie van klassen en objecten. Structural betreft de compositie tussen klassen en objecten. Tot slot gaat het bij behavioral patterns om de manier waarop klassen of objecten met elkaar interacteren en verantwoordelijkheid verdelen. De scope beschrijft of een pattern op klasse- of objectniveau slaat.

Een van de beschreven patterns is het "Strategy Pattern". Het betreffende pattern valt onder de behavioral patterns en het werkt op de object scope. Het Strategy Pattern is uitermate geschikt voor het oplossen van dit architecturale vraagstuk. Het pattern maakt het namelijk mogelijk om een familie van algoritmes te encapsuleren en op deze manier verwisselbaar te maken (Gamma et al, 1995). In dit geval kan het toegepast worden om de manier waarop competenties en vaardigheden in het systeem komen verwisselbaar te maken. Daarnaast kan ditzelfde pattern worden toegepast voor het abstraheren van de manier waarop kennisniveaus van werknemers in het systeem komen. Dit maakt het gemakkelijk om in de toekomst een implementatie te maken die gebruik maakt van bijvoorbeeld het BIS.

Onafhankelijke backend

Een wens van Sogeti is om de kenniskaartenapplicatie in een later stadium als plugin voor Confluence te hebben. Echter is de implementatie van een dergelijke plugin niet het directe zwaartepunt van het project. In eerste instantie zal het resultaat een webapplicatie betreffen. Echter moet het mogelijk zijn om in een later stadium de applicatie alsnog om te zetten naar een plugin voor het KMS. Dit vereist dat de architectuur dit ondersteunt. Belangrijke criteria om deze uitbreiding mogelijk te maken zijn:

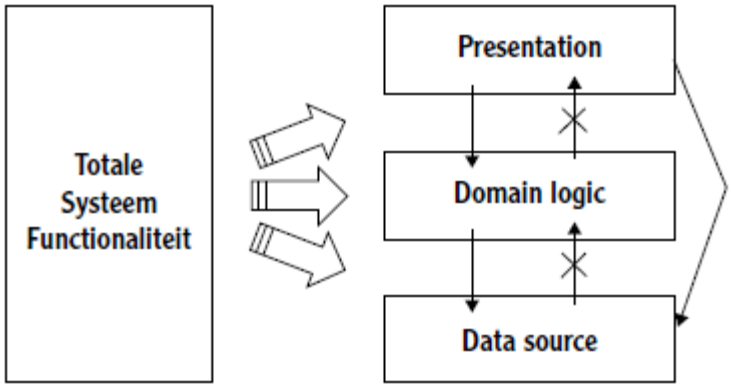
- Hoge mate van abstractie
- Sterke encapsulatie
- Herbruikbaarheid mogelijk maken
- Loose Coupling stimuleren

Microsoft (2009) benoemt verschillende architecturale stijlen, ook wel architecturale patterns genoemd. Deze patterns zijn vergelijkbaar met het idee achter design patterns, echter werken ze op een hoger niveau. Elk van de architecturale patterns hebben voor- en nadelen en vaak werken ze in combinatie met elkaar. De patterns omschrijven elk doelstellingen die ze kunnen bereiken. Twee van de patterns, namelijk de Layered Architecture en Service Oriented Architecture zijn mogelijke kandidaten voor het kenniskaartensysteem, omdat beide voldoen aan de opgestelde criteria.

Layered architecture

Layered Architecture [LA] (zie figuur 10) bestaat volgens Pruijt (2010) en Microsoft (2009) uit verscheidene verticale stacks waarin overeenkomstige rollen of verantwoordelijkheden worden geplaatst. Deze lagen hebben een loosely coupled communicatie. Resultaat hiervan is dat onderliggende lagen herbruikbaar zijn gemaakt. Onderliggende lagen worden hiermee geëncapsuleerd en dit biedt abstractie. Volgens Pruijt (2010) heeft een LA door zijn logische indeling verder de voordelen dat de lagen hierdoor analyseerbaar, testbaar en wijzigbaar worden. Door de encapsulatie hebben wijzigingen in één laag geen impact op andere lagen, wat onderhoudbaarheid, portabiliteit en aanpasbaarheid ten goede komt.

Daarentegen komen deze voordelen niet zonder bijbehorende nadelen. Pruijt (2010) benoemt de moeilijkheid van het correct indelen van de lagen als gevolg van het ontbreken van een eenduidige definitie van verschillende lagen in het model. Dit bemoeilijkt het proces waarin functionaliteiten in lagen worden geplaatst, waardoor niet-functionele eisen als nog niet behaald kunnen worden. Daarnaast is de efficiëntie en performance van een gelaagde applicatie lager dan een ongelaagde applicatie (Pruijt, 2014). De voor- en nadelen van dit architecturale pattern zijn opgesomd in tabel 5.



Figuur 10. Layered Architecture (Pruijt, 2010)

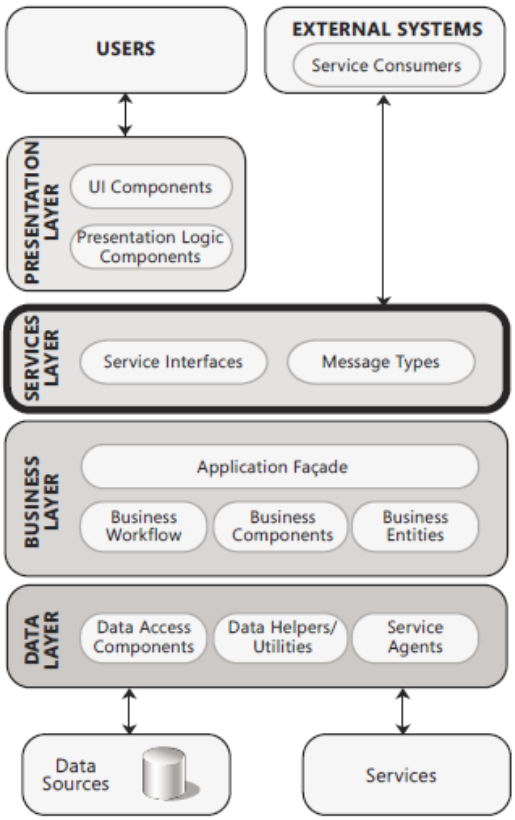
Voordelen	Nadelen
• Loose coupling • Herbruikbaarheid • Abstractie	• Complexiteit plaatsing logica • Performance/efficiëntie

Tabel 5. Voor- en nadelen Layered Architecture

Service Oriented Architecture

Een Service Oriented Architecture [SOA] (zie figuur 11) lijkt op de LA, het berust op het lagenprincipe. Daarentegen is bij SOA een "Service Layer" toegevoegd. Deze laag stelt de onderliggende logica beschikbaar voor externen via een communicatie protocol, vaak via internet of intranet. Hiermee kunnen zogenaamde consumers de beschikbare service afnemen. Dit betekent dat meerdere clients op dezelfde data zouden kunnen gaan werken. Service Oriented Architecture biedt volgens Microsoft (2009) de voordelen dat dergelijke systemen loose coupling mogelijk maken. Microsoft (2009) benoemt verder de voordelen dat dergelijke systemen gemakkelijk te hergebruiken zijn, dat ze abstractie bieden, duplicatie van functionaliteiten kunnen voorkomen en tot slot hoge mate van interoperabiliteit bieden.

Echter heeft SOA ook nadelen, enkele nadelen die in de huidige context van toepassing zijn zullen worden benoemd. Een groot nadeel is beschikbaarheid (Pearson, 2015). Dit resulteert uit het feit dat het via het internet beschikbaar wordt gemaakt. Als er geen verbinding is kan het systeem niet worden geraadpleegd. Pearson (2015) benoemd verder de nadelen dat er vanwege het http-protocol geen garantie is dat een request wordt uitgevoerd en dat SOA de performance niet ten goede komt. De voor- en nadelen van SOA zijn weergegeven in tabel 6.



Figuur 11. Service Oriented Architecture (Microsoft, 2009)

Voordelen	Nadelen
• Meerdere consumers/clients • Loose coupling • Herbruikbaarheid • Abstractie • Interoperabiliteit	• Beschikbaarheid • Betrouwbaarheid • Performance

Tabel 6. Voor- en nadelen Service Oriented Architecture

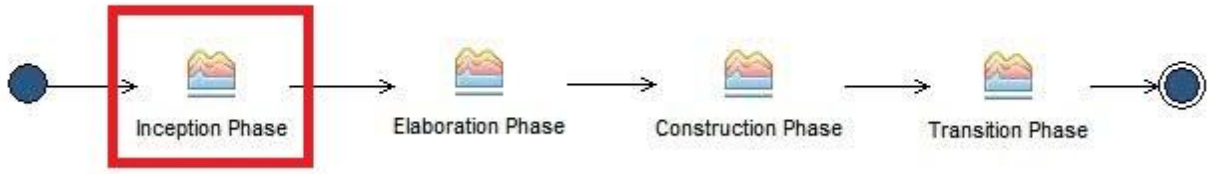
Conclusie

De uitbreidbaarheid van het domein kan met het opzetten van een object-georiënteerd ontwerp worden bereikt. Door niet aan implementatie van slechts het huidige domein gebonden te zijn kunnen andere Business Lines worden ondersteund. Integratie met externe systemen voor functionaliteiten zoals het verkrijgen van het kennisniveau van werknemers of de vaardigheden hiervan kan door het toepassen van het Strategy Design Pattern gemakkelijk worden gemaakt. Dit maakt het mogelijk om dit algoritme dynamisch te selecteren.

Tot slot is het met een Layered Architecture mogelijk om in de toekomst de applicatie alsnog om te zetten naar een Confluence plugin. Een Service Oriented Architecture is hierbij ook een optie, echter is dit minder performant en betrouwbaar. Met oog op de toekomst is de SOA daarentegen de beste optie. Het maakt het mogelijk om met een centraal aanspreekpunt verschillende clients te ondersteunen. Aangezien de applicatie verder ontwikkeld zal worden is het wenselijk om deze flexibiliteit te behouden. Het PoC zal dus werken op een Service Oriented Architecture.

6.6 Proof of concept

Het Proof of Concept [PoC] bracht alle aspecten van het onderzoek samen. De opzet hiervan was bepaald door de antwoorden op de deelvragen van het onderzoek. Het verloop van het ontwerpen, implementeren en testen van het PoC was ingedeeld in drie OpenUP fasen (zie sectie 5.8): Inception, Elaboration en Construction. Deze drie fasen volgen elkaar logischerwijs op. Het proces zal dan ook worden beschreven aan de hand van deze fasen.



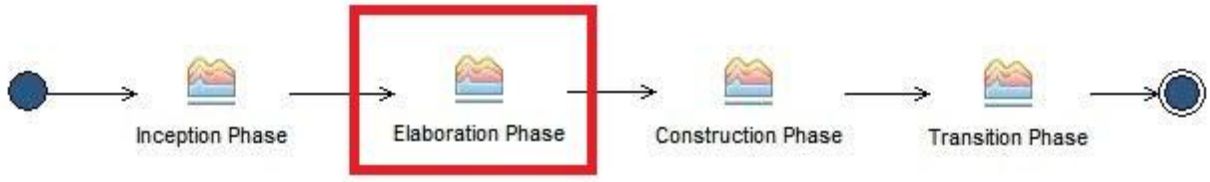
Figuur 12. Inception fase

Inception

Tijdens deze fase zijn voornamelijk de oriënterende taken uitgevoerd. Dit waren onder andere de interviews. Daarnaast is er in het Inception Report (zie bijlage E) rekening gehouden met UX en Usability, gebruik makend van het vijf oppervlakken model (zie sectie 6.4). Tot slot werd er op functioneel niveau een lagenmodel, componentenmodel en domein van het systeem opgesteld.

Aan de hand van interviews en gesprekken met de product owner zijn er in deze fase ter invulling van de Strategy-laag gebruikersdoeleinden en doeleinden van de stakeholders vastgesteld. Daarnaast zijn er als gevolg hiervan gebruikerstaken en Persona's opgesteld. Vervolgens werden voor de Scope-laag de functionele- en niet-functionele requirements opgesteld. Ook werden content requirements vastgesteld en werden fictieve scenario's bedacht die gebruik maakte van de Persona's. Voor Structure-laag werden de taken uit de Strategy-laag uitgewerkt in het Interaction Design waarin werd beschreven hoe het systeem met de gebruiker moet interacteren. Verder werd de Information Architecture opgesteld. Voor de Skeleton-laag werden Wireframes opgesteld.

Op technisch gebied werd aan de hand van de uitgevoerde interviews en vastgelegde requirements gewerkt. Er werd onder andere een lagenmodel opgesteld waarmee de structuur van het systeem vast werd gelegd. Verder werden de hoofdcomponenten van het systeem in het componentenmodel vastgelegd. Daarnaast werd een functioneel domein opgesteld. Dit domein moest aantonen dat de requirements vervuld konden worden. Ook moest het domein laten zien dat de functionaliteiten met lagere prioriteit door het systeem mogelijk toe te voegen zijn.

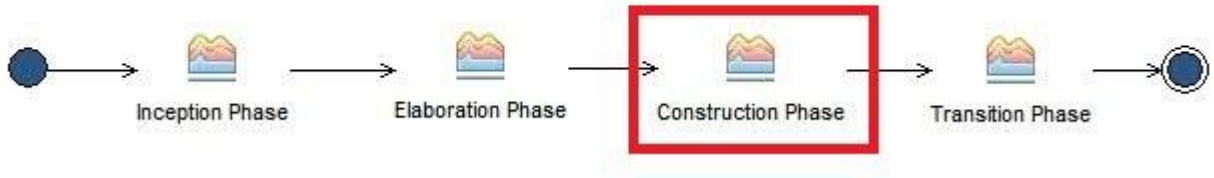


Figuur 13. Elaboration fase

Elaboration

Tijdens deze fase werd vooral op een technisch niveau ontworpen. Dit resulteerde onder andere in een opzet voor het technische domein model waarin het eerdere functionele domein model werd uitgebreid. Hierin zijn voornamelijk attributen, methoden en relatiebenamingen toegevoegd. Naast het technische domein model werd er ook een initieel implementatiemodel opgesteld (zie figuur 14). Hierin werden entiteiten vastgelegd die logica bevatten en uitvoeren, maar niet gepersisteerd moeten worden. Daarnaast is de gelaagde architectuur duidelijk te zien in de vorm van de service, manager en repository klassen. Verdere toevoegingen aan deze modellen gebeurden per timebox tijdens de Construction fase. De uitwerking van deze fase is terug te vinden in het Elaboration Report (zie bijlage F).

Figuur 14. Implementation model



Figuur 15. Construction fase

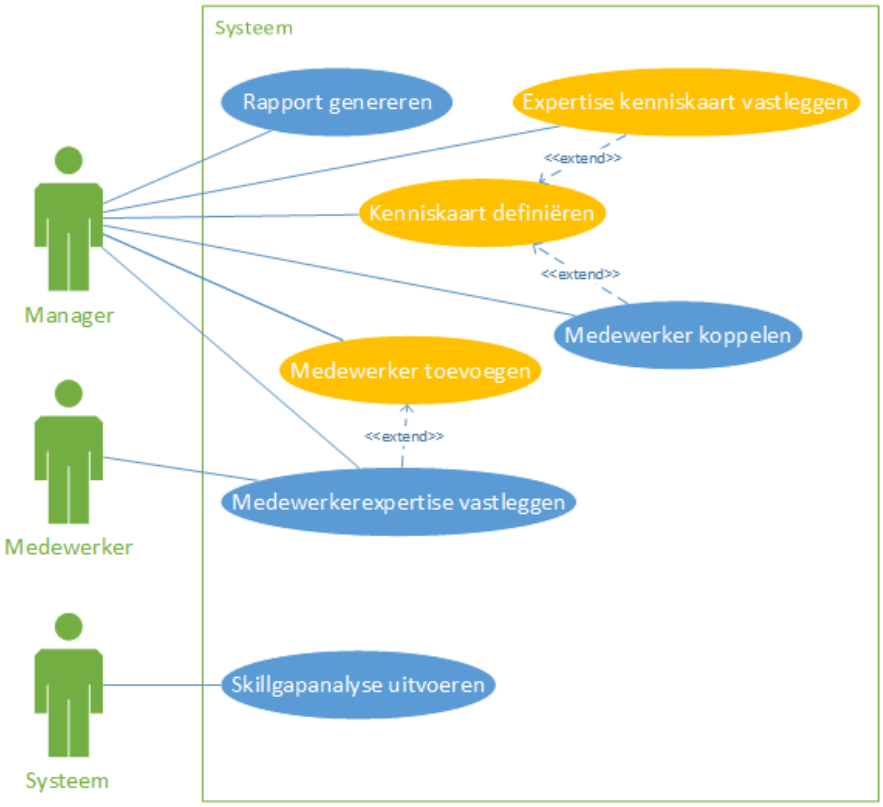
Construction

Na het afsluiten van de Inception en Elaboration fase was er een duidelijk uitgangspunt en kon de Construction fase beginnen. Deze fase was opgedeeld in twee timeboxes van twee weken waarin verschillende user stories (zie sectie 6.2) werden geïmplementeerd. Elk van de user stories werd opgedeeld in kleine, behapbare werkeenheden: work items. Na elke timebox werd er met de opdrachtgever gereflecteerd op de uitgevoerde werkzaamheden en werd afgestemd wat de werkzaamheden voor de opvolgende timebox zouden worden. De hier beschreven staat van de applicatie omschrijft de huidige stand van zaken, echter zal hier na het inleveren van de scriptie nog verder aan worden ontwikkeld.

Timebox 1

De eerste timebox bestond initieel uit het opzetten van de projectconfiguratie. Na het voltooien hiervan kon aan de eerste set user stories worden begonnen. De tijdens deze timebox gerealiseerde use cases waren als volgt (zie ook figuur 16):

- Kenniskaart definiëren
- Expertise bij een kenniskaart vastleggen
- Medewerker toevoegen



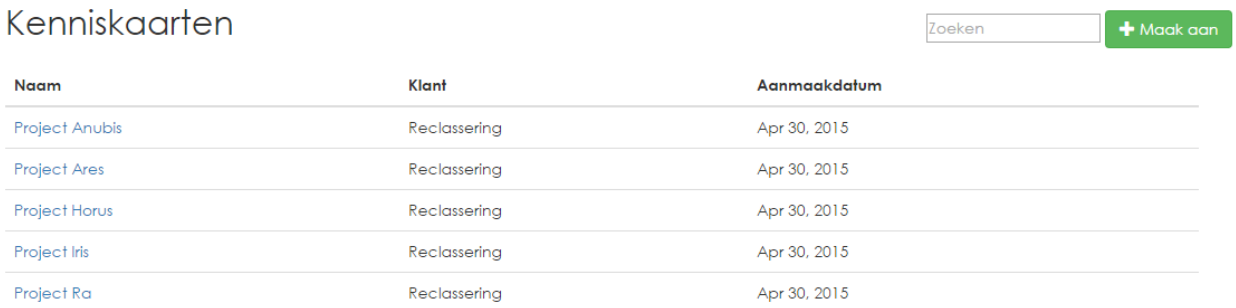
Figuur 16. Use cases timebox 1

Voor de te implementeren user stories werd steeds eerst het domein geïmplementeerd. Die bestond in deze timebox uit de kenniskaart, expertise en medewerker. Na het opstellen van het domein werd per entiteit de REST-service opgezet waarmee CRUD-functionaliteit werd ondersteund. Hiervoor werd de SOA structuur opgezet, bestaande uit de servicelaag, domeinlaag en datalaag. Deze structuur is visueel weergegeven in het implementatiemodel, te vinden in het Elaboration Report (bijlage F).

Na het implementeren van de REST-service kon de cliënt worden ontwikkeld. In de webapplicatie werden allereerst views aangemaakt voor de kenniskaarten. Deze waren gebaseerd op de opgestelde wireframes uit de Inception Fase (zie bijlage E). De views die waren opgesteld bestonden uit een overzicht van kenniskaarten (zie figuur 17), een scherm voor het definiëren van een kenniskaart en een detailscherm van een kenniskaart. Elk van de views maakt gebruik van de service om de vereiste data te verkrijgen en versturen. Hetzelfde werd vervolgens gedaan voor expertises en medewerkers.

De nu ondersteunde CRUD-functionaliteit voor de kenniskaarten en expertises maakte de eerste koppeling mogelijk. Dit betrof de koppeling tussen kenniskaarten en expertises, waarmee de projectmanager expertise-eisen bij zijn project kan vastleggen. Deze handeling is deel van de eerste stap in het huidige kenniskaartenproces (zie sectie 6.1). Bij het aanmaken of aanpassen van een kenniskaart is het mogelijk gemaakt om expertises die in het systeem stonden te selecteren en toe te voegen. De gebruiker kan verder per expertiseniveau het aantal vereiste projectleden specificeren

Tot slot werden unit tests opgesteld voor de logica in de domeinlaag. Deze laag bestond voornamelijk uit validatie van data die door de servicelaag wordt verschaft. Daarnaast werd de usability van de applicatie getest door het uitvoeren van user tests (zie bijlage G). De feedback die hiermee is verkregen, is verwerkt en heeft geleid tot enkele aanpassingen in de cliëntapplicatie.

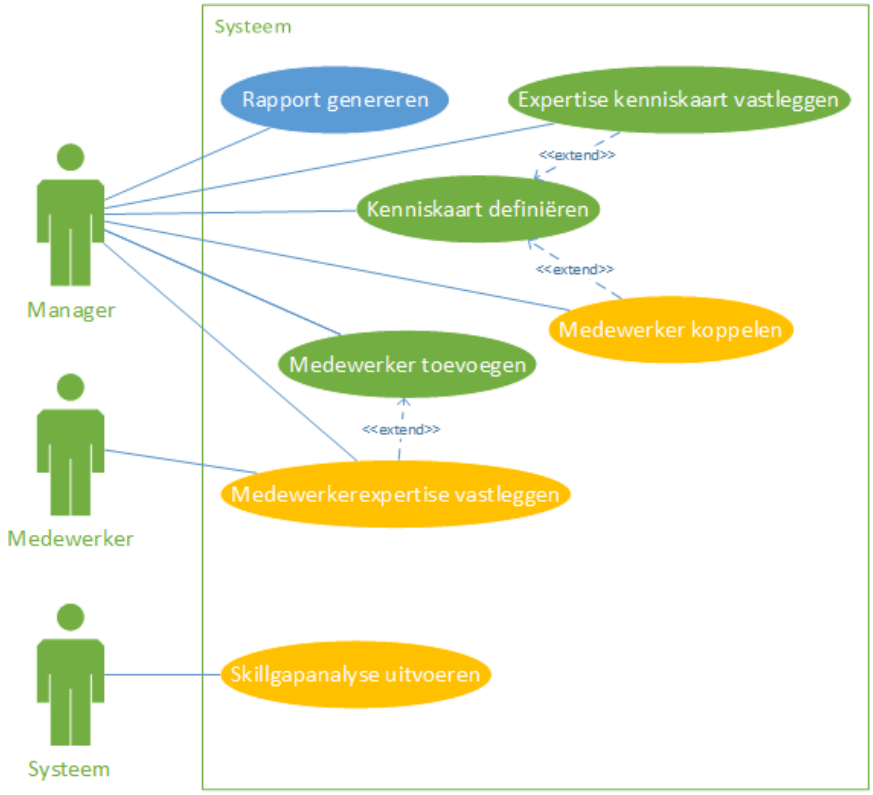


Figuur 17. Kenniskaartenoverview

Timebox 2

In de tweede timebox werd een nieuwe set user stories geselecteerd die geïmplementeerd zouden worden. Deze timebox bestond uit use cases waarin voornamelijk koppelingen werden gemaakt tussen de entiteiten uit de eerste timebox. Wanneer deze koppelingen geïmplementeerd waren kon de functionaliteit voor het uitvoeren van skillgapanalyses worden geïmplementeerd. De use cases voor de tweede timebox waren als volgt (zie ook figuur 18):

- Medewerker aan kenniskaart koppelen
- Medewerkerexpertise vastleggen
- Skillgapanalyse uitvoeren



Figuur 18. Use cases timebox 2

Het koppelen van medewerkers aan een kenniskaart is in de cliëntapplicatie mogelijk gemaakt. Dit is gerealiseerd door een toevoeging aan de bestaande pagina's voor het definiëren en aanpassen van een kenniskaart. Hier is het mogelijk gemaakt om bestaande werknemers in het systeem te kunnen zoeken, selecteren en toe te voegen aan de kenniskaart. Dit stelt de gebruiker in staat om op een gebruiksvriendelijke manier projectleden aan een project toe te kennen. Naast het koppelen van projectleden is ook het ontkoppelen mogelijk gemaakt. Met deze functionaliteiten kan de projectmanager zijn actuele projectleden bij een opdracht beheren.

Elke medewerker brengt bepaalde expertises met zich mee, in een project maakt dit deel uit van de totale aangeleverde kennis. In de webapplicatie is deze koppeling gerealiseerd door bij het aanmaken of aanpassen van een medewerker expertises te kunnen selecteren en toe te voegen. Wanneer een expertise is geselecteerd kan hierbij een niveau van beheersing worden geselecteerd. Wederom is het mogelijk gemaakt expertises ook te ontkoppelen of het bijbehorende niveau aan te passen. Reflecterend op het huidige kenniskaartenproces zijn deze interacties te plaatsen onder de processtap "expertise invullen".

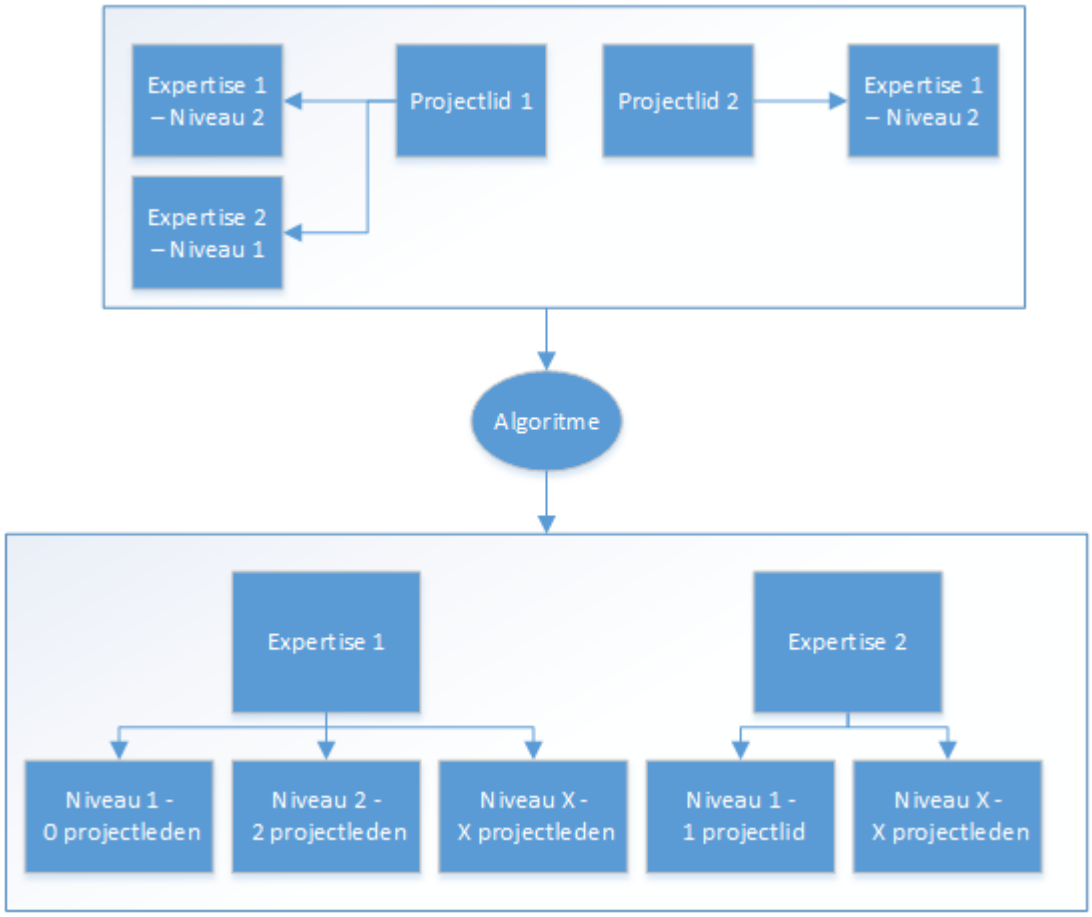
Alle tot op dit punt geïmplementeerde functionaliteiten maakte het mogelijk om de skillgapanalyse te implementeren. Deze analyse biedt een projectmanager zoals gezegd inzicht in zijn expertisedekking bij een project. De analyse is gerealiseerd door een simpel algoritme te maken dat uit twee stappen bestaat:

- Het berekenen van de cumulatieve expertises van projectleden.
- Het berekenen van de expertisedekking bij een kenniskaart.

Deze stappen zijn het gevolg van de manier hoe de expertisedekking wordt berekend. De dekking is namelijk een product van de vereiste expertises en aanwezige expertises bij een kenniskaart. Vereiste expertises worden ingevoerd bij een project en vereisen verder geen berekeningen voorafgaand aan de

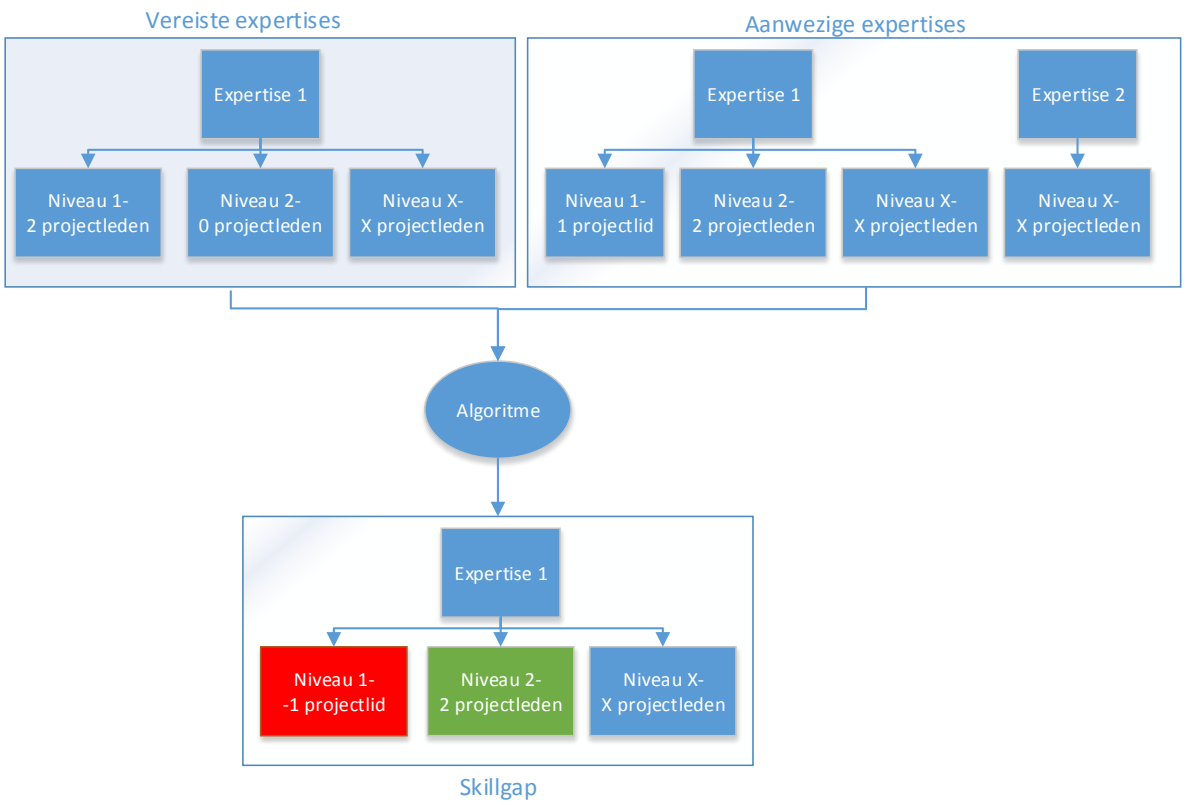
skillgapanalyse. Daarentegen is de aanwezige kennis bij een project afhankelijk van gekoppelde medewerkers en de aan hun gekoppelde expertises met bijbehorende niveaus.

De eerste stap in het algoritme is het berekenen van de aanwezige expertise, afhankelijk van aangeleverde projectleden. Deze stap levert een verzameling met alle unieke expertises die een groep medewerkers bezitten. Bij elke expertise is vervolgens per niveau het aantal personen te zien die het betreffende kennisniveau bezit. Deze stap is weergegeven in figuur 19.



Figuur 19. Cumulatieve expertises projectleden

De tweede stap, bestaande uit het berekenen van de expertisedekking kan na het voltooien van de eerste stap plaatsvinden. De expertisedekking wordt opgesteld door alle vereiste expertises bij een kenniskaart langs te lopen en deze naast de totale aanwezige kennis te houden. Expertises die in zowel de vereiste expertises en aanwezige kennis aanwezig zijn worden vervolgens vergeleken. Per expertise wordt elk kennisniveau langsgelopen en wordt van het aantal aanwezige personen het aantal vereiste personen afgetrokken. Dit resulteert bij een tekort aan personen in een negatieve waarde en bij een overschot in een positieve waarde. Het resultaat van dit algoritme is te zien in figuur 20. Een positieve skillgap is groen weergegeven, een negatieve rood.



Figuur 20. Skillgapanalyse

De berekende expertisedekking is gekoppeld aan de kenniskaart en kan in de cliëntapplicatie worden geraadpleegd. Hiervoor was de detailpagina bij kenniskaarten uitgebreid met een overzicht waarin bij elke vereiste expertise de dekking per expertiseniveau wordt getoond. Deze dekking is de kern van de kenniskaartenapplicatie. Het stelt projectmanagers in staat om gebreken en overschotten in kennis te detecteren bij hun projecten. De resulterende pagina van de cliënt is te zien in figuur 21. Hierin is duidelijk te zien dat de controls zich, zoals in de gehele applicatie, rechtsboven bevinden. Deze uniformiteit is volgens Garrett (2011) een manier om het design effectief naar gebruiker over te brengen. Daarnaast is contrast gebruikt om aandacht te trekken naar belangrijke elementen op de pagina. Garrett (2011) noemt dit de primaire manier om aandacht van de gebruiker te trekken.

Expertise	Onervaren	Gemiddeld	Ervaren	Bovengemiddeld	Adviseur
HTML5	0	0	0	-1	0
CSS3	-1	0	0	0	0
C#	-1	0	1	0	1

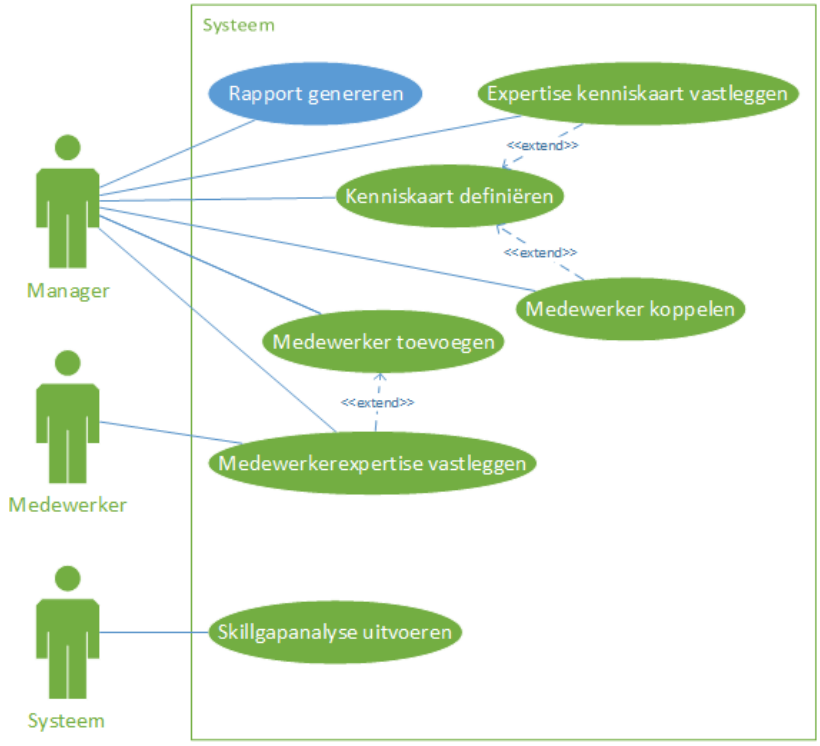
Figuur 21. Skillgap in de cliëntapplicatie

De timebox werd na implementatie van de geselecteerde user stories afgerond met het opstellen van unit tests en het uitvoeren van user tests. De unit tests betroffen dit keer voornamelijk het algoritme waarmee de skillgap wordt berekend. De user tests waren dit keer toegespitst op het koppelen van medewerkers aan kenniskaarten, het toevoegen van expertises aan medewerkers en de skillgapanalyse. Het toonde dit keer aan dat de terugknop boven in het scherm naar de vorige pagina dient te verwijzen en niet naar het overzicht van de entiteit (zie figuur 21). Tot slot is er een code quality analysis uitgevoerd met SonarQube. De uiteindelijke SQALE-rating van de applicatie was het hoogst haalbare, rating A. Zowel de User Tests als de code quality analysis zijn te vinden in het Construction Report (zie bijlage G).

Conclusie

Al met al zijn in het PoC bijna alle must-haves geïmplementeerd. De applicatie biedt de functionaliteiten die reeds in de Excel versie bestonden. Dit was waar de initiële focus lag. Daarnaast zal het fungeren als basis voor de implementatie van de rest van de functionele requirements. De user stories met betrekking tot het inloggen en beheren van de applicatie dienen nog te worden geïmplementeerd. Dit geldt ook voor het genereren van een Read-Only rapport voor de klant en het importeren van data uit de bestaande Excel sheets. Het ontwerp van de applicatie maakt het echter mogelijk om user stories met lagere prioriteit op een later punt te implementeren. Het eindresultaat was een REST service en een webapplicatie als cliënt waarin de volgende use cases waren geïmplementeerd (zie ook figuur 22):

- Kenniskaarten definiëren
- Benodigde expertises en expertiseniveaus vastleggen
- Medewerkers toevoegen
- Medewerker aan kenniskaart koppelen
- Medewerkerexpertise vastleggen
- Skillgapanalyse uitvoeren



Figuur 22. Geïmplementeerde use cases

7 Conclusie

De conclusie van het onderzoek zal in dit hoofdstuk worden gepresenteerd. Hierbij worden de deelvragen één voor één gepresenteerd. Hierbij zal ook het antwoord op de betreffende deelvraag worden gegeven. Tot slot zal de hoofdvraag nog eenmaal worden herhaald. Het antwoord op de hoofdvraag zal vervolgens worden aangedragen.

Hoe werkt het handmatige kenniskaartenproces van Sogeti's Java Centre of Excellence?

Aan de hand van een interview met een gebruiker van de kenniskaarten was het huidige kenniskaartenproces in kaart gebracht. Hierbij waren de handelingen uit het proces en de mensen die ze uitvoeren verhelderd. Daarnaast was het doel en belang van deze handelingen vastgelegd. Grofweg bleek het proces te bestaan uit de volgende stappen:

1. Het vastleggen van de vereiste expertises voor een project
2. Projectleden leggen hun expertise vast
3. De manager controleert de ingevulde expertiseniveaus
4. Borging van kennis toonbaar maken aan de klant

Aan welke functionele- en niet functionele requirements moet het kenniskaartensysteem voor Sogeti's Java Centre of Excellence voldoen?

De requirements van het systeem zijn opgesteld aan de hand van interviews. Eén interview is gehouden met de product owner. Dit interview resulteerde in een concrete lijst met functionele eisen. Het interview is ook gehouden met een gebruiker van de huidige kenniskaarten. Zijn eisen waren minder concreet, maar zijn inzicht bood een ander perspectief. De twee interviews hebben samen tot een heldere lijst met requirements voor de applicatie geleid, waarin de must-haves bestonden uit functionaliteiten die de Excel oplossing reeds bood. Wat niet-functionele eisen betreft waren veiligheid en concurrency belangrijk. De snelheid van de applicatie diende niet storend te zijn.

Is het mogelijk om de huidige data in de kenniskaarten van Sogeti's Java Centre of Excellence in het automatische systeem mee te nemen en zo ja, op welke manier(en)?

Het meenemen van de data uit de Excel kenniskaarten is mogelijk, er bestaan libraries in Java die data uit .xls sheets kunnen lezen. Uit een vergelijking tussen enkele grote libraries die deze functionaliteit bieden is een keuze gemaakt aan de hand van enkele opgestelde criteria. Voor het Proof of Concept [PoC] bleek Apache POI de beste keuze te zijn vanwege het feit dat het aan alle criteria voldoet met als bijkomstigheid dat het Open-Source is.

Op welke manier(en) kan User Experience en Usability bijdragen aan de efficiëntie van het systeem?

Usability en User Experience bleek van groot belang te kunnen zijn in systemen zoals het ontwikkelde PoC. Daarnaast bleek dat het de effectiviteit bij het uitvoeren van een handeling kan verhogen. Door het vijf lagenmodel toe te passen tijdens het ontwikkelen van het PoC is er constant rekening gehouden met de eindgebruikers van het systeem. Bij het ontwerpen van de cliëntapplicatie is voornamelijk gelet op aspecten die in dit model als belangrijk werden benoemd. Dit hele proces komt de vijf kwaliteitscomponenten (zie sectie 2.3) van usability ten goede.

Hoe kan op architecturaal niveau uitbreidbaarheid en toekomstbestendigheid van de kenniskaartenapplicatie worden bereikt?

Architecturaal gezien moest het systeem het mogelijk maken in meerdere Business Lines [BLS] toegepast te worden, integratie met externe systemen in de toekomst mogelijk maken en moest het mogelijk als Confluence-plugin neergezet kunnen worden. Het inzetten bij andere BLS is mogelijk doordat in het domein van het systeem gewerkt is met object georiënteerde principes zoals inheritence en composition. Integratie met externe systemen is mogelijk gemaakt door punten in het systeem te detecteren waar dit plaats zou kunnen vinden en hier het Strategy Pattern toe te passen. Tot slot is het systeem opgezet volgens een Service Oriented Architecture en de logica met een REST-webservice aan te bieden. Dit maakt het mogelijk voor meerdere cliëntapplicaties (zoals een webapplicatie of Confluence-plugin) om op dezelfde logica en data te werken.

Op welke manier kan het kenniskaartensysteem van Sogeti's Java Centre of Excellence het beste geautomatiseerd worden, zodat Project Management efficiënter verloopt?

Al met al heeft het PoC zijn doel bereikt. Het onderzoek heeft het bedrijf een concrete lijst met requirements opgeleverd. Dit maakt duidelijk wat de gebruikersgroep in een kenniskaartenapplicatie wil hebben. De requirements zullen dan ook een grote rol spelen bij de doorontwikkeling van de applicatie, ze tonen aan welke kant de applicatie op moet gaan. Het systeem maakt toekomstige uitbreiding mogelijk door de manier hoe het is opgezet. Door het hebben van een hoge Usability in de cliënt is de applicatie efficiënter dan de huidige oplossing in Excel. De belangrijkste functionaliteiten, bestaande uit de functies in de Excel-oplossing, zijn geïmplementeerd. Op de cliëntapplicatie dient echter nog te worden doorontwikkeld voordat het in productie kan worden genomen.

8 Aanbevelingen

De aanbevelingen naar aanleiding van de conclusies uit hoofdstuk 7 zullen hier worden aangedragen en beargumenteerd. Deze aanbevelingen volgen uit de antwoorden op de deelvragen en hoofdvraag van het onderzoek. De aanbevelingen zijn met name bestemd voor de opdrachtgever.

De applicatie dient voornamelijk op het gebied van beheer, authenticatie en authorisatie verder te worden ontwikkeld. Vanwege het feit dat er tijdens de ontwikkelfase niet zo snel een testserver beschikbaar was, zijn deze aspecten nog niet ver genoeg uitgewerkt. Ten behoeve van de bruikbaarheid van de applicatie zou dit een eerste vervolgstap zijn. Dit maakt het mogelijk voor de gebruikers om sneller de data te verkrijgen die ze nodig hebben. Daarnaast maakt dit de applicatie veiliger. Het beheeraspect is vereist om de applicatie ook daadwerkelijk in productie te nemen. Dit maakt het mogelijk om gebruikers toe te voegen aan de applicatie en ze rollen en rechten toe te kennen.

Tot slot vereist de user story met betrekking tot het genereren van een Read-Only rapport hoge prioriteit. Dit bleek volgens de gebruiker, als resultaat van het interview voor de requirementanalyse, een zeer belangrijke functionaliteit in de applicatie te zijn. Uiteindelijk heeft het tijdsbestek voor de ontwikkelfase de implementatie van deze functionaliteit echter nog niet toegelaten.

9 Evaluatie

In dit hoofdstuk zal de procesgang van het onderzoek onder de loep worden genomen. Er zal kritisch worden gekeken naar keuzes die gedurende het onderzoek zijn gemaakt. Tot slot zullen enige punten van verbetering worden genoemd.

Misconceptie kenniskaarten

Een enorme complicatie deed zich voor toen bleek dat Erwin de Gier toch niet de product owner zou zijn gedurende het project. In eerste instantie was dit wel het geval. Een groot deel van de opdrachtomschrijving en interpretatie van de opdracht was dan ook resultaat van overleg met Erwin. Na de toekenning van de daadwerkelijke product owner zijn gesprekken gevoerd over de inhoud van de opdracht. Aanleiding hiervoor was een review van het ingediende Plan van Aanpak. Uit dit gesprek bleek dat er zich een misconceptie heeft voorgedaan omtrent het begrip "kenniskaarten". De veronderstelling van de originele product owner was dat een kenniskaart analoog stond aan de skillsmatrix. Dit bleek echter niet zo te zijn.

Uit de gesprekken met Edo bleek dat een kenniskaart een compleet ander doel heeft dan dat van de skillsmatrix. Waar de skillsmatrix kennis inzichtelijk probeert te maken is een kenniskaart breder. Een kenniskaart bleek een inzicht te bieden op de bezetting van een project, als het ware de kennis waar een team over beschikt. Dit zorgde er voor dat het zwaartepunt en de context in een laat stadium grote mate van verandering vereiste. Dit resulteerde in aanpassingen van de onderzoeksvragen en de aanpak. In grote lijnen klopte dit echter nog wel. Er is dus geprobeerd om zoveel mogelijk van de originele vragen te behouden en dit slechts aan te passen aan de nieuwe context.

Betrouwbaarheid interviews

Een verbeterpunt voor mogelijk vervolgonderzoek ligt bij de interviews die zijn gehouden. Bij beide interviews zijn slechts één of twee personen ondervraagd. Wanneer er meerdere personen ondervraagd worden is het mogelijk dat er meer perspectieven aan het licht worden gebracht. Dit is voornamelijk een punt van verbetering bij het in kaart brengen van het huidige kenniskaartenproces waar slechts één gebruiker is geïnterviewd. Het zou wenselijk zijn om hier meer gebruikers voor te ondervragen, omdat het niet zeker is dat deze ene ondervraagde de volledige doelgroep representeert.

Bruikbaarheid gebruikerstests

Vanwege het feit dat de gebruikers van de kenniskaarten op locatie bij de klant zaten is er vanwege praktische redenen voor gekozen om ze niet elke iteratie te benaderen. In plaats daarvan is vanwege beschikbaarheid gekozen om personen in de naaste omgeving te vragen om user tests uit te voeren. De gebruikerstests zijn daarom vooral op oppervlakkige aspecten en minder op de functionaliteit gefixeerd. Wanneer de directe gebruikers van de applicatie dichter bij het ontwikkelproces betrokken konden zijn, had het eindresultaat beter aan de wensen van de gebruikers kunnen voldoen.

Beschikbaarheid testserver

Aangezien er pas in het eindstadium van het onderzoek een testserver beschikbaar was om de applicatie op te draaien zijn bepaalde aspecten niet volgens plan verlopen. Dit resulteerde allereerst in een andere implementatievolgorde dan wat vooraf gepland was. De authenticatie diende met een LDAP te verlopen, wat niet buiten bepaalde servers beschikbaar bleek te zijn. Dit kon niet vanaf de ontwikkelmachine worden benaderd. Dit heeft er toe geleidt dat de eerstvolgende must-haves, die onafhankelijk waren van authenticatie, als eerste werden geïmplementeerd.

10 Bronvermelding

Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J. (2002). *Agile software development methods: Review and analysis.* Oulu: VTT Electronics.

Alavi, M., Leidner, D. E. (1999). Knowledge management systems: issues, challenges, and benefits. *Communications of the AIS*, *1*(2es), 1.

Alavi, M., Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly*, 107-136.

Apache (2015). Apache POI. *POI-HSSF and POI-XSSF.* Geraadpleegd op 7 april 2015 via http://poi.apache.org/spreadsheet/index.html.

Aspose (2015). Aspose.Cells. *Aspose.Cells for Java*. Geraadpleegd op 7 april 2015 via http://www.aspose.com/java/excel-component.aspx.

Beck, S. (2003). Skill and competence management as a base of an Integrated Personnel Development (IPD). *Journal of Universal Computer Science*, 9(12), 1381-1387.

Bruno, V., Tam, A., Thom, J. (2005). Characteristics of web applications that affect usability: a review. In *Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future* (pp. 1-4). Computer-Human Interaction Special Interest Group (CHISIG) of Australia.

Bullseye (2013). Bullseye Testing Technologies. *Minimum acceptable code coverage.* Geraadpleegd op http://www.bullseye.com/minimum.html

Confluence (2015). Confluence. *Overview*. Geraadpleegd op 12 maart 2015 via https://www.atlassian.com/software/confluence.

Gamma, H., Helm, R., Johnson, R., Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software.* Pearson Education.

Garrett, J. J. (2011). *The elements of user experience: User centered design for the web and beyond.* Berkeley: New riders.

Garro, A., Palopoli, L. (2003). An xml multi-agent system for e-learning and skill management. In *Agent Technologies, Infrastructures, Tools, and Applications for E-Services* (pp. 283-294). Springer Berlin Heidelberg.

Google (2010). Google Testing Blog. *Code coverage goal: 80% and no less!* Geraadpleegd op http://googletesting.blogspot.nl/2010/07/code-coverage-goal-80-and-no-less.html

Gracion Software (2015). Gracion Software. *What is LDAP?* Geraadpleegd op 28 april via http://www.gracion.com/server/whatldap.html.

JExcel (2015). JExcel. Geraadpleegd op 7 april 2015 via http://www.teamdev.com/jexcel.

JExcelApi (2015). JExcelApi. *Java Excel Api.* Geraadpleegd op 7 april 2015 via http://jexcelapi.sourceforge.net/.

Jongman, A. (2014). User experience & usability. *Kickoff*. Geraadpleegd op 19 maart 2015 via https://cursussen.sharepoint.hu.nl/fnt/37/TCIF-V2UXUI1-14/Studiemateriaal/UX%20-%200%20-%20Kickoff.pptx.

Microsoft (2009). Microsoft Patterns and Practices. *Microsoft application architecture guide 2nd edtion.* Geraadpleegd op https://msdn.microsoft.com/en-us/library/ff650706.aspx.

Nielsen Norman Group (2012). Usability. *Usability 101: Introduction to usability*. Geraadpleegd op http://www.nngroup.com/articles/usability-101-introduction-to-usability/.

Nonaka, I. (1991). The knowledge-creating company. *Harvard business review*, 69(6), 96-104.

OpenUp (2012). OpenUp. *OpenUp Wiki.* Geraadpleegd op http://epf.eclipse.org/wikis/openup/.

Pearson (2015). Pearson. *Disadvantages and pitfalls of web services.* Geraadpleegd op 3 april via http://www.pearsonhighered.com/samplechapter/0672325152.pdf.

Pruijt, L. (2010). Meer inzicht in een gelaagde architectuur: deel 1. *Release, 2010(12)*, 22-26.

Pruijt, L. (2014). Software architecture. *Logical layers.* Geraadpleegd op https://cursussen.sharepoint.hu.nl/fnt/6/TCIF-V2SARCH1-10/Studiemateriaal/Presentations/SA%2004%20-%20Logical%20Layers.pptx.

Raman, R., Paige, R. F. (2008). Process-centered review of object oriented software development methodologies. *ACM Computing Surveys, 40(1),* 3:1-3:89. doi: 10.1145/1322432.1322435

SmartXLS (2015). SmartXLS. *SmartXLS for Java.* Geraadpleegd op 7 april 2015 via http://www.smartxls.com/indexj.htm.

Sogeti (2015). Sogeti Nederland. *Facts en Figures.* Geraadpleegd op 24 februari 2015 via https://www.sogeti.nl/over-sogeti/facts-en-figures.

Sogeti (2015, januari). Bedrijfsstructuur Sogeti Nederland. *Introductieprogramma 2015.* Geraadpleegd op https://sharenet.sogeti.nl/.

Steehouder, M., Jansen, C., Maat, K., Staak, J. van der, Vet, D. de, Witteveen, M., & Woudstra, E. (2006). *Leren communiceren: Handboek voor mondelinge en schriftelijke communicatie* (5e dr.). Groningen/Houten: Wolters-Noordhoff.

TalentAlign (2012, december). Skills and competencies. *Skills vs. Competencies. What's the difference?* Geraadpleegd op http://www.talentalign.com/skills-vs-competencies-whats-the-difference/.

TribeHR (2015). Talent management. *Skills tracking*. Geraadpleegd op 12 maart 2015 via http://tribehr.com/social-hr-software/talent-management/skills-tracking.

Verma, A. (2010). *Requirements analysis phase reengineering: Decision making & requirements prioritisation among stakeholders for a new software project.* Geraadpleegd op http://undergraduate.csse.uwa.edu.au/year4/Current/Students/Files/2010/AbhijeetVerma/CorrectedDissertation.pdf

11 Contactpersonen

Willem van de Griendt

Telefoon:	+31(0)88 660 66 00 (nakiesnummer 2781)
Mobiel:	+31(0)6 50 60 13 76
Email:	willem.vande.griendt@sogeti.nl

Erwin de Gier

Telefoon:	+31(0)88 660 66 00 (nakiesnummer 2625)
Mobiel:	-
Email:	erwin.de.gier@sogeti.nl

Edo Hendriksen

Telefoon:	-
Mobiel:	+31 (0)6 51 82 67 06
Email:	edo.hendriksen@sogeti.nl

Esther van der Stappen

Telefoon:	088 481 82 83
Mobiel:	-
Email:	esther.vanderstappen@hu.nl

Roy Straub

Telefoon:	-
Mobiel:	+31(0)6 83 67 81 90
Email:	roy.straub@sogeti.nl

12 Bijlagen

A. Plan van aanpak

SOGETI

Plan van Aanpak

Kenniskaartenapplicatie:

Skill Management System

Hogeschool Utrecht

Eerste examinator: Mariecke Schipper

Roy Straub, 1614878

2.1

Versie informatie

Versie	Datum	Bijzonderheden	Auteur
1.0	13-02-2015	Eerste conceptversie	Roy Straub
1.1	25-02-2015	Hoofdstukindeling methoden en aanpak aangepast n.a.v. feedback.	Roy Straub
2.0	02-03-2015	Structuur document aanpassen n.a.v. feedback	Roy Straub
2.1	17-03-2015	Laatste revisie n.a.v. feedback en transitie naar definitieve versie	Roy Straub

Managementsamenvatting

Binnen het Java Centre of Excellence (CoE) van Sogeti wordt gebruikt gemaakt van kenniskaarten. Deze kenniskaarten worden gebruikt om de expertise van werknemers vast te leggen. De huidige oplossing werkt met behulp van Excel.

De wens is om hier een automatisch systeem voor te hebben. Dit systeem valt onder een lijn systemen waarmee kennis binnen een bedrijf wordt beheerd. De zogenaamde *Knowledge Management Systems* (KMS). Binnen de KMS valt een subcategorie systemen waarmee vaardigheden van werknemers worden vastgelegd. Deze systemen zijn de zogenaamde *Skill Management Systems* (SMS). SMS bieden inzicht in de vaardigheden waar werknemers over beschikken en daarnaast welke *gaps* er tussen een werknemer en een taak binnen het bedrijf bestaat. Dit inzicht maakt het op managementniveau mogelijk om vervolgstappen op te stellen, bijvoorbeeld het opstellen van een opleidingsplan voor de betreffende werknemer.

Het te ontwikkelen systeem moet qua architectuur zo opgezet worden dat het later mogelijk ook buiten het CoE kan worden toegepast. Door het proces te automatiseren is het mogelijk om de kenniskaarten gemakkelijker te beheren. Verder maakt het geavanceerde toepassingen mogelijk.

De realisatie van het systeem brengt enkele problemen met zich mee. Zo ligt slechts een deel van de functionaliteiten die het systeem moet bevatten vast. Verder is het van belang dat de kenniskaarten en data die de werknemers invullen actueel zijn. Uit vooronderzoek bleek usability aan de interactie met het systeem bij te kunnen dragen De vraag is echter hoe dit het beste met usability kan worden bereikt.

De centrale vraag voor dit onderzoek luidt als volgt:

Hoe kan het kenniskaartensysteem van Sogeti's Java Centre of Excellence het beste geautomatiseerd worden, zo dat de Skill Management efficiënter verloopt?

De centrale vraag is opgemaakt uit de volgende deelvragen:

- *Hoe werkt het handmatige kenniskaartenproces van Sogeti's Java Centre of Excellence?*
- *Aan welke functionele- en niet functionele requirements moet het kenniskaartensysteem voor Sogeti's Java Centre of Excellence voldoen?*
- *Is het mogelijk om de huidige data in de kenniskaarten van Sogeti's Java Centre of Excellence in het automatische systeem mee te nemen en zo ja, op welke manier(en)?*
- *Op welke manier(en) kan usability bijdragen aan de actualiteit van de kenniskaarten?*
- *Hoe kan op architecturaal niveau uitbreidbaarheid en toekomstbestendigheid van de kenniskaartenapplicatie worden bereikt?*

Interviews zullen worden gebruikt voor het beantwoorden van de eerste twee deelvragen. Voor het beantwoorden van de eerste deelvraag zullen interviews worden gehouden met medewerkers die gebruik maken van het huidige kenniskaartenproces. Het resultaat van de tweede deelvraag is een requirementanalyse. Hiervoor zullen interviews met verschillende stakeholders worden gehouden.

Het beantwoorden van de derde deelvraag zal met behulp van literatuuronderzoek en desk research gebeuren. Verder zal er worden onderzocht hoe usability in de frontend van het systeem hoog kan liggen, aangezien dit de effectiviteit van het systeem kan verhogen. Het onderwerp usability is veelvuldig beschreven in de literatuur, wat het bij uitstek geschikt maakt voor literatuuronderzoek. Daarnaast kan er ook naar bestaande systemen worden gekeken of kunnen interne UX-experts worden geraadpleegd.

Er zal worden gezocht naar manieren om de data uit het huidige Excel proces mee te nemen in het systeem. Dit zal voornamelijk met desk research gebeuren, eventueel met gebruik van ondersteunende literatuur. Ook de deelvraag met betrekking tot de architectuur van het systeem zal worden beantwoord met gebruik van literatuur en desk research, hierbij eventueel gebruik makend van de expertise van de technisch begeleider.

Het eindresultaat is een Proof of Concept(PoC) implementatie van de kenniskaartenapplicatie. Hierin zullen de antwoorden op de deelvragen terugkomen. Dit PoC zal opgesteld worden aan de hand van de uitgevoerde requirementanalyse. Daarnaast zal het huidige proces en de bijbehorende data terug te vinden zijn in het PoC. Verder zal er met het ontwerpen frontend rekening worden gehouden met de usability van het systeem. Tot slot zal de architectuur van de backend aan de hand van de antwoorden op de bijbehorende deelvraag worden opgezet.

De kwaliteit van het systeem zal worden geborgd door verschillende vormen van testen. De functionaliteit van het systeem zal door unit testing met behulp van JUnit worden getest. De usability wordt door het afnemen van gebruikerstests geborgd. Tot slot zal de kwaliteit van de code worden geanalyseerd met SonarQube.

Inhoudsopgave

1	**Inleiding**	**1**
2	**Context**	**2**
2.1	Sogeti	2
2.2	Aanleiding	3
2.3	Theoretisch kader	3
3	**Opdracht**	**7**
3.1	Probleemstelling	7
3.2	Doelstelling	7
3.3	Opdrachtomschrijving	8
4	**Vraagstelling**	**11**
5	**Methoden en Aanpak**	**12**
5.1	Huidig kenniskaartenproces	12
5.2	Requirementanalyse	13
5.3	Legacy data	14
5.4	Actualiteit Kenniskaarten	14
5.5	Architectuur	14
5.6	Proof of Concept	15
5.7	Communicatie	15
5.8	Softwareontwikkelmethode	16
5.9	Kwaliteitsbewaking	18
6	**Risico's**	**20**
7	**Planning**	**21**
7.1	Takenplanning	21
7.2	Planning deliverables	22
8	**Bronvermelding**	**23**
9	**Contactpersonen**	**24**

1 Inleiding

Sogeti maakt in hun Java Centre of Excellence (CoE) gebruik van zogenaamde kenniskaarten. Deze kenniskaarten trachten de kennis van medewerkers in kaart te brengen. Dit proces maakt het bijvoorbeeld mogelijk om aan de hand van expertise gemakkelijk werknemers te selecteren voor openstaande projecten. Verder kunnen er analyses plaatsvinden resulterend in bijvoorbeeld een ontwikkelingsplan voor de betreffende werknemer. Momenteel vindt dit proces handmatig met behulp van een Excel bestand plaats.

Dit document beschrijft de afstudeeropdracht met betrekking tot het implementeren van een geautomatiseerd kenniskaartensysteem bij Sogeti. Het systeem zal de bestaande oplossing in Excel vervangen en skill management efficiënter maken. De opdracht zal tot in detail worden toegelicht en er zal een theoretisch kader worden afgebakend. Verder zal de organisatie beknopt worden beschreven.

Tijdens het afstudeerproject wordt onderzoek uitgevoerd dat bepaalde vragen beantwoord omtrent het systeem dat gerealiseerd zal worden. Deze vraagstelling zal behandeld en toegelicht worden. De onderzoeksvragen zullen worden beantwoord met behulp van methoden die beschreven zullen worden in het hoofdstuk methoden en aanpak.

Tot slot volgt een planning die het afstudeertraject beslaat. Hierin worden de belangrijkste werkzaamheden en deliverables uiteengezet. Naast deze planning zal ook een risicoanalyse worden getoond. Deze analyse bevat risico's die een succesvolle uitvoering van het afstudeerproject kunnen tegenwerken.

2 Context

In dit hoofdstuk zal de context van het project ter sprake komen. Onder andere de organisatie waar het afstuderen plaatsvindt zal hier worden toegelicht. Daarnaast zal de aanleiding voor het project worden beschreven, hiermee zal duidelijk de ontstaansreden van het project worden toegelicht. Tot slot volgt een kort theoretisch kader waarin de centrale begrippen van dit traject worden besproken.

2.1 Sogeti

De afstudeeropdracht bij het bedrijf Sogeti plaatsvinden. Sogeti (2015) is een IT-dienstverlener die wereldwijd actief is in vijftien landen. Eén van de landen waarin Sogeti actief is, is Nederland. Sogeti Nederland bestaat uit ongeveer vijfentwintighonderd medewerkers en beschikt over kantoren in Amersfoort, Amsterdam, Eindhoven, Groningen en het hoofdkantoor in Vianen.

Naast het feit dat Sogeti geografisch op veel fronten actief is, geldt dit ook voor de verschillende takken van de markt. Zo zijn er verschillende afdelingen voor onder andere Java, Microsoft, Mobile en Oracle. Sogeti staat verder bekend op de markt vanwege hun capabiliteit omtrent softwaretesten en architectuur. Enkele klanten aan wie Sogeti diensten levert zijn ING, KPN, KLM en ProRail.

Zoals eerder genoemd is een groot deel van de business van Sogeti het detacheren van werknemers. Dit proces beslaat verschillende afdelingen, of zoals ze intern worden genoemd: *business lines* (in figuur 1 afgebeeld als grijze blokken). De business line Sales gebruikt het zogenoemde Business Informatie Systeem (BIS) om binnenkomende aanvragen samen met de afdeling planning te vertalen naar mogelijke matches. Een match is het vinden van een passende werknemer bij de openstaande aanvraag. Eén van de belangrijke criteria bij deze matchingprocedure is het afspiegelen van de vereiste kennis voor een project ten opzichte van de kennis van de beschikbare werknemers. De huidige bedrijfsstructuur volgens Sogeti (2015, januari) is in figuur 1 weergegeven.

Figuur 1. Bedrijfsstructuur Sogeti januari 2015

Binnen de business line Java bevindt zich het Java Centre of Excellence (CoE). Dit is een groep Javaspecialisten die volgens E. de Gier (persoonlijke communicatie, 6 maart, 2015) voornamelijk applicaties ontwikkelen voor intern gebruik en beschikbaar zijn voor consultancy. Het bevat werknemers die niet gedetacheerd zijn. Binnen dit CoE zal het project plaatsvinden en de proof of concept worden ontwikkeld. De resulterende applicatie zal in eerste instantie voor intern gebruik van het CoE dienen met mogelijk perspectief op toepassing bij andere business lines en integratie met andere systemen. De student zal deel uitmaken van het CoE. Hierin zal het ontwerpen, ontwikkelen en testen van de kenniskaartenapplicatie de verantwoordelijkheid zijn van de student.

2.2 Aanleiding

Kennis is belangrijk binnen Sogeti. Een groot gedeelte van de omzet komt namelijk van het detacheren van werknemers bij externe bedrijven. Inzicht in de vaardigheden waar werknemers over beschikken is dus van groot belang. Dit zorgt er voor dat werknemers gevonden kunnen worden voor openstaande aanvragen van externe partijen. Te realiseren projecten hebben mensen nodig die over de vereiste kennis beschikken. Echter is de vraag die hieruit voortkomt: hoe weet je welke mensen over die kennis beschikken?

De huidige oplossing binnen het CoE is het gebruik van een zogenaamde kenniskaart. Deze kenniskaart wordt gebruikt om kennis van werknemers op bepaalde gebieden te inventariseren en inzichtelijk te maken. Als de kennis in kaart is gebracht, is het gemakkelijk om de beschikbare kennis tegen de vereiste kennis uiteen te zetten. Dit stelt het bedrijf onder andere in staat om de juiste persoon voor de juiste taak te vinden.

De kenniskaart wordt momenteel bijgehouden in Excel. Het toevoegen van categorieën en kennisgebieden aan de kenniskaart vind handmatig plaats. De wens van het CoE is om hier een geautomatiseerd systeem voor te hebben. Hier moet het bestaande, handmatige proces in verwerkt worden. Echter zou het automatische systeem geavanceerde bewerkingen mogelijk maken zoals bijvoorbeeld het reflecteren op projecten of het automatisch bijwerken van de lijst met vaardigheden in de kenniskaarten.

Het systeem kan verder uitkomst gaan bieden voor het gehele matchingproces van het plaatsen van een aanvraag door een klant, tot het uitzoeken van een geschikte medewerker die aan de eisen van de aanvraag voldoet. Momenteel wordt de kennis van werknemers in verschillende systemen bijgehouden waardoor het onderhouden van deze data wordt bemoeilijkt. Een uniform, centraal systeem zou hier uitkomst bieden.

2.3 Theoretisch kader

Kennis

Sogeti is een kennisbedrijf, maar wat is kennis eigenlijk? Hoe verhoudt kennis zich tot informatie en data? Alavi & Leidner (2001) verzamelden verschillende definities van kennis, voornamelijk met toespitsing op de IT. Hierbij wordt de veronderstelling gemaakt dat kennis, informatie en data van elkaar verschillen. Dit aangezien er anders niets nieuws is aan het beheren van kennis ten opzichte van het beheren van data en informatie. Een toepasbare definitie van deze drie begrippen in de context van dit project werd gemaakt door Maglitta (geciteerd in Alavi & Leidner, 2001). Hij definieerde data als ruwe getallen en

feiten. Informatie werd omschreven als verwerkte data. Tot slot werd kennis beschreven als informatie die toepasbaar is gemaakt.

In de huidige situatie bij Sogeti worden de vaardigheden van werknemers in kaart gebracht met een skillsmatrix. In essentie zijn de categorieën die hierin vallen en de waarden die de werknemers die hier invullen een kennisinventarisatie. De matrix geeft weer tot in welke mate de werknemers kennis hebben over een specifiek onderwerp. Als hierbij de definitie van kennis door Maglitta wordt aangehouden is dit dus informatie over een onderwerp die werknemers in de praktijk kunnen toepassen.

Vaardigheden en competenties

Centrale begrippen tijdens dit project zijn vaardigheden (skills) en competenties (competencies). Hoewel deze begrippen veel van elkaar weg hebben zijn ze niet geheel te substitueren. Volgens TalentAlign (2012, december) is een vaardigheid iets wat is geleerd om een taak uit te kunnen voeren. Een competentie is daarentegen breder. Competenties kunnen bestaan uit meerdere vaardigheden en zijn dus overkoepelend. Ze bevatten ook mogelijkheden, gedrag en houding.

Een voorbeeld van het verschil tussen deze twee begrippen is gedefinieerd op TalentAlign (2012, december). Gebruik makend van een IT-voorbeeld kan een vaardigheid worden omschreven als bijvoorbeeld het kunnen coderen in Java of C#. Een voorbeeld van een competentie zou in deze analogie analytisch vermogen kunnen zijn. Elke vaardigheid in het spectrum vereist diezelfde competentie. Zo zou het opdoen van de vaardigheid coderen in C++ dezelfde competentie(s) kunnen vereisen.

Knowledge Management

Het beheren van kennis is sinds de jaren negentig een belangrijk aspect van bedrijven geworden. Nonaka (1991) inspireerde velen met zijn opvatting:
"*In an economy where the only certainty is uncertainty, the one sure source of lasting competive advantage is knowledge*".

Het artikel van Nonaka was aanleiding voor het ontstaan van het begrip *Knowledge Management* (Beck, 2003). Kennis werd gezien als een resource van de organisatie (Alavi & Leidner, 2001). Deze resource was minder vatbaar voor veranderingen van de markt in tegenstelling tot vluchtige resources zoals bijvoorbeeld kapitaal. Men zag in dat informatie van belang is voor het innoveren en het voorblijven van de concurrentie. Het beheren van deze resource werd een belangrijk onderwerp. Dit proces van het beheren van informatie en kennis werd ondergebracht onder de term Knowledge Management.

Knowledge Management werd door de groeiende interesse in het onderwerp geautomatiseerd in informatiesystemen. Deze lijn met systemen werden *Knowledge Management Systems* (KMS) genoemd. Het doel van deze systemen is het maken, verzamelen, organiseren en verspreiden van kennis binnen een bedrijf (Alavi & Leidner, 1999). Deze systemen zouden bedrijven in staat stellen om sneller te reageren op veranderingen in de markt en het zou een bedrijf flexibeler maken. Dit zou er volgens Harris en Stata (geciteerd in Alavi & Leidner, 1999) voor zorgen dat het bedrijf innovatiever en productiever wordt. Enkele voorbeelden van deze KMS zijn bijvoorbeeld intranet of een tool zoals Confluence (2015).

Skill Management en Competency Management

Onderdeel van het voorgaand benoemde KM is Skill Management (SM). SM houdt volgens Beck (2003) het beheren van kwalificaties, ervaringen en kennis van medewerkers in. Door deze kennis met een bepaald medium vast te leggen is het voor het bedrijf inzichtelijk welke medewerkers bij welke projecten inzetbaar zijn. Dit is gelijk één van de belangrijke toepassingen van SM. Het kan worden gebruikt in KM;

Werknemerprofielen kunnen tegenover opdrachtprofielen worden gezet. Verder wordt het mogelijk om te zoeken naar een specifiek profiel.

Overigens benadrukt Beck (2003) het belang van de scheiding tussen projecten/taken en personen. Ten eerste zorgt dit er voor dat de vraag gesteld moet worden wat een project vereist en wat een werknemer aan expertise biedt. Verder wordt het mogelijk om een scheiding te maken tussen de "cans" en "cannots" van een persoon. Deze verschillen kunnen geanalyseerd worden en dit kan informatie bieden die bijvoorbeeld gebruikt kan worden bij evaluatiegesprekken.

Ook voor het automatiseren van SM zijn systemen in het leven geroepen. Deze systemen worden ook wel *Skill Management Systems* (SMS) genoemd. Beck (2003) omschrijft de ontwikkeling van een dergelijk systeem. Hierbij wordt gebruik gemaakt van een nummerwaardering van nul tot vijf bij elke vaardigheid in de lijst. Deze lijst met vaardigheden wordt vervolgens door de werknemer ingevuld. Dit resulteert in een profiel van de werknemer waarin zijn vaardigheden en zijn beheersing van elke vaardigheid zijn vastlegt.

Gedurende de ontwikkeling van het systeem identificeerde Beck (2003) enkele belangrijke condities waaraan voldaan moet worden om een dergelijk systeem een succes te maken. Zo is het risico aanwezig dat werknemers het confronterend vinden om een dergelijke lijst met vaardigheden in te vullen. Dit resulteert uit het feit dat een vaardighedenlijst eerder de gebreken van een werknemer benadrukt dan de vaardigheden die hij wel beheerst. Daarnaast werd ook het belang van een goede usability voor het systeem benadrukt.

Garro en Palopoli (2003) beschrijven een vergelijkbare aanpak voor de opzet van een dergelijk systeem. Ook in dit systeem wordt gebruik gemaakt van een zogenaamde *skillmap.* In het beschreven systeem is het beheren van deze skillmaps een van de belangrijke bewerkingen die het systeem moet kunnen uitvoeren. Daarnaast wordt ook de functionaliteit van het analyseren van de *"gap"* tussen de vereiste en beschikbare vaardigheden van een werknemer genoemd. Er wordt zelfs een stap bovenop gedaan door een geavanceerde feature voor te stellen. Het betreft het aanbieden van verwijzingen naar *Learning Object*s die het geanalyseerde gap zouden kunnen dichten. Met een learning object wordt een bron van kennis bedoeld waar de benodigde vaardigheid mee kan worden opgedaan. In het artikel wordt een Multi-Agent oplossing voorgesteld voor het automatiseren van deze functionaliteiten.

In essentie is het gebruik van een skillmap dezelfde aanpak als die binnen het CoE wordt toegepast, waar het kenniskaarten worden genoemd. Verder zijn er ook commerciële tools beschikbaar voor SM. Een voorbeeld van een dergelijk systeem is de skills tracking in TribeHR (2015).

Tot slot bestaan er ook systemen voor het managen van competenties (*Competency Management Systems*). In essentie doen deze systemen iets vergelijkbaars als SMS echter betreft het dan competenties in plaats van vaardigheden. Competency Management Systems vallen buiten de scope van het project.

Usability

Eerder werd genoemd dat voor het succes van een SMS usability hoog dient te liggen. Een hoge usability maakt het voor de gebruiker fijner om gebruik te maken van de applicatie. Het is van belang om met het ontwerpen van de applicatie voor Sogeti rekening te houden met usability. Er dient rekening te worden gehouden met usability in het systeem omdat van invloed kan zijn op de effectiviteit van het SMS, zowel positief als negatief.

Usability wordt vaak geassocieerd met gebruiksvriendelijkheid van een interface. Nielsen Norman Group (2012) definieert usability als zijnde een kwaliteitsattribuut voor het gemak waarmee een interface gebruikt kan worden. Verder bestaat usability op zich volgens de Nielsen Norman Group uit de volgende kwaliteitscomponenten:

1. Learnability
2. Efficiency
3. Memorability
4. Errors
5. Satisfaction

Hierin is *learnability* de maatstaaf voor het gemak waarmee een gebruiker een taak voor de eerste keer uitvoert. *Efficiency* geeft weer hoe snel een gebruiker een taak kan uitvoeren wanneer ze het ontwerp van de applicatie hebben geleerd. Met *memorability* wordt het gemak waarmee een terugkerende gebruiker de applicatie weer onder de knie kan krijgen bedoelt. *Errors* beschrijft het aantal fouten dat gebruikers maken en hoe gemakkelijk ze hier van herstellen. Tot slot geeft *satisfaction* weer hoe fijn het ontwerp voor de gebruikers is om te gebruiken.

Naast een definitie van usability geeft de Nielsen Norman Group (2012) ook redenen waarom usabillity van belang is voor websites. Een slechte usability zorgt er voor dat bezoekers van websites de site verlaten. Voor intranet en dus ook interne applicaties ligt dit net anders. Hier betreft het vooral productiviteit van de gebruikers. Een hogere usability zorgt er voor de er minder tijd nodig is voor de gebruiker om zijn taak te voltooien.

3 Opdracht

De daadwerkelijke opdracht zal hier worden toegelicht. Allereerst zal de probleemstelling worden beschreven. Hierin worden de problemen van de huidige situatie uitgelicht. Vervolgens zal de doelstelling die Sogeti wilt bereiken met het project worden beschreven. Tot slot zal een concrete opdrachtomschrijving worden aangeleverd waarin de opdracht tot in detail zal worden beschreven.

3.1 Probleemstelling

Het probleem dat de applicatie op dient te lossen is meervoudig. Hierin zijn de volgende aspecten te herkennen:

- Het handmatig toevoegen en beheren van kenniskaarten
- Dezelfde data bevindt zich dubbel in meerdere systemen
- De requirements van de applicatie zijn niet duidelijk gedefinieerd

Ten eerste is het handmatig toevoegen en beheren van kenniskaarten een overduidelijk probleem. Dit proces is inefficiënt en de data is moeilijk te onderhouden. Dit maakt het lastig om er voor te zorgen dat de data bruikbaar is. Het systeem zal effectiever zijn als de data actueel is. Hierdoor wordt de meest accurate representatie mogelijk van de kennis waar werknemers over beschikken.

Het tweede probleem is het hebben van de data in meerdere systemen. De data die in de verschillende systemen, zoals het BIS staan is vrij overeenkomstig. De medewerkers worden gepresenteerd met daarbij een lijst kennisgebieden en een score die de werknemer hieraan heeft verbonden. Omdat deze data in verschillende systemen staat, maakt dit het wederom moeilijk om de data te onderhouden. Dit kan verder tot gevolg hebben dat data inconsistent is. Op zijn beurt kan dit problemen opleveren voor entiteiten die assumpties maken die gebaseerd zijn op de betreffende gegevens. De business line planning zou aan de hand van niet actuele gegevens een werknemer bijvoorbeeld niet kunnen plaatsen bij een opdracht. Dit kan als consequentie hebben dat het bedrijf geld misloopt.

Tot slot doet zich nog een probleem voor. Dit is een gevolg van het feit dat het systeem uiteindelijk mogelijk meerdere business lines kan beslaan (dus bijvoorbeeld ook Mobile of Microsoft). De requirements voor het systeem zijn slechts gedeeltelijk concreet. Aangezien het systeem zo een groot deel van het totale matchingproces zou kunnen vormen moeten er duidelijke, concrete requirements komen. Wanneer dit niet gebeurt, zou het systeem mogelijk niet op meerdere business lines inzetbaar kunnen zijn.

3.2 Doelstelling

De belangrijkste doelstelling die Sogeti met dit project wil bereiken is het krijgen van inzicht in de expertise van werknemers. Het gehele proces van deze inventarisatie zou geautomatiseerd en centraal geregeld moeten worden. Het systeem moet het makkelijker maken om nieuwe kennisgebieden toe te kunnen voegen en dus het geheel flexibeler maken. Wanneer deze aspecten gerealiseerd zijn is het voor Sogeti mogelijk om het systeem buiten puur en alleen het CoE toe te passen. Dit maakt op zijn beurt verdere integratie met andere gebruikte systemen van Sogeti mogelijk en relevant.

Uit de beschrijving van de opdracht (zie hoofdstuk 4.2) is af te leiden dat het een productopdracht betreft. Uiteindelijk is het project voor Sogeti een succes als er aan verschillende condities wordt voldaan. Eén van de eerste condities is het hebben van een accurate en grondige requirementanalyse. Vervolgens dienen de met de product owner afgestemde functionaliteiten geïmplementeerd te worden in een applicatie. De correcte werking van het systeem moet aantoonbaar zijn en ook dient de architectuur van het systeem uitbreidbaarheid mogelijk te maken. Per slot van rekening levert de opdracht een stuk software op waar op doorontwikkeld kan gaan worden.

3.3 Opdrachtomschrijving

De opdracht die tijdens het afstudeertraject zal worden uitgevoerd heeft betrekking tot het automatiseren van het kennisinventarisatieproces en dit proces uit te breiden. Dit proces verloopt zoals eerder vernoemd handmatig. Het systeem wat gerealiseerd zal worden, moet de huidige procesgang automatiseren. De wensen vanuit het CoE zijn om dit systeem in de vorm van een webapplicatie of Confluence-plugin te hebben. In eerste instantie zal een webapplicatie worden opgezet waarna uitbreiding naar plugin mogelijk is.

Confluence is KMS wat wordt gebruikt binnen Sogeti. Het is een applicatie waarmee één centrale plek wordt gecreëerd voor het delen van informatie binnen een bedrijf. Verdere informatie over Confluence is te vinden op de officiële website (https://www.atlassian.com/software/confluence).

Skillsmatrix

In Figuur 2 is een voorbeeld te zien van de huidige kenniskaart of *skillsmatrix* zoals het ook wel wordt genoemd. De vaardigheden zijn aan de hand van categorieën geordend. Dit is in het voorbeeld terug te zijn aan het kopje API's. Onder elke categorie hangen individuele skills, in het voorbeeld onder andere SOAP, AJAX, JAX-WS. Vervolgens is per skill een waardering toe te kennen. Dit is waar de werknemer aangeeft tot hoever zijn kennis op het betreffende gebied reikt. Dit is bereikt door een nummer van nul tot vier toe te kennen aan een skill. Des te hoger het toegekende nummer, des te meer kennis heeft de werknemer over het onderwerp.

Actualiteit kenniskaarten

Het is voor de effectiviteit van het systeem van belang dat de kenniskaarten van werknemers up to date en accuraat zijn. Op dit gebied zal onderzoek uitgevoerd moeten worden. Momenteel heeft het bedrijf geen oplossing voor het vraagstuk hoe ze de toepasbaarheid en relevantie van de kenniskaarten gaan waarborgen. Uit vooronderzoek bleek usability een grote rol spelen bij dit SMS, het kan de interactie met het systeem positief beïnvloeden. Met usability kan tijdens het ontwerpen van het systeem al rekening worden gehouden.

Privacy

Aangezien het hier data van werknemers betreft die niet per se iedereen moet kunnen inzien is privacy een punt van aandacht. Het is belangrijk om van te voren een duidelijke definitie te hebben van welke data wel of niet voor andere werknemers inzichtelijk moet zijn. Dit zal afgestemd moeten worden met de opdrachtgever. De opdrachtgever heeft de duidelijkste visie bij het product en dit maakt hem dan ook de aangewezen persoon om verheldering te verschaffen over de privacy van de werknemers.

Functionaliteiten

Naast het automatiseren van het huidige proces moet het systeem het ook nieuwe functionaliteiten bieden die zich momenteel niet in het handmatige proces bevinden. Een voorstel vanuit de organisatie was bijvoorbeeld het kunnen maken van projectevaluaties. De kern van functionaliteiten voor het systeem zien er uit als volgt:

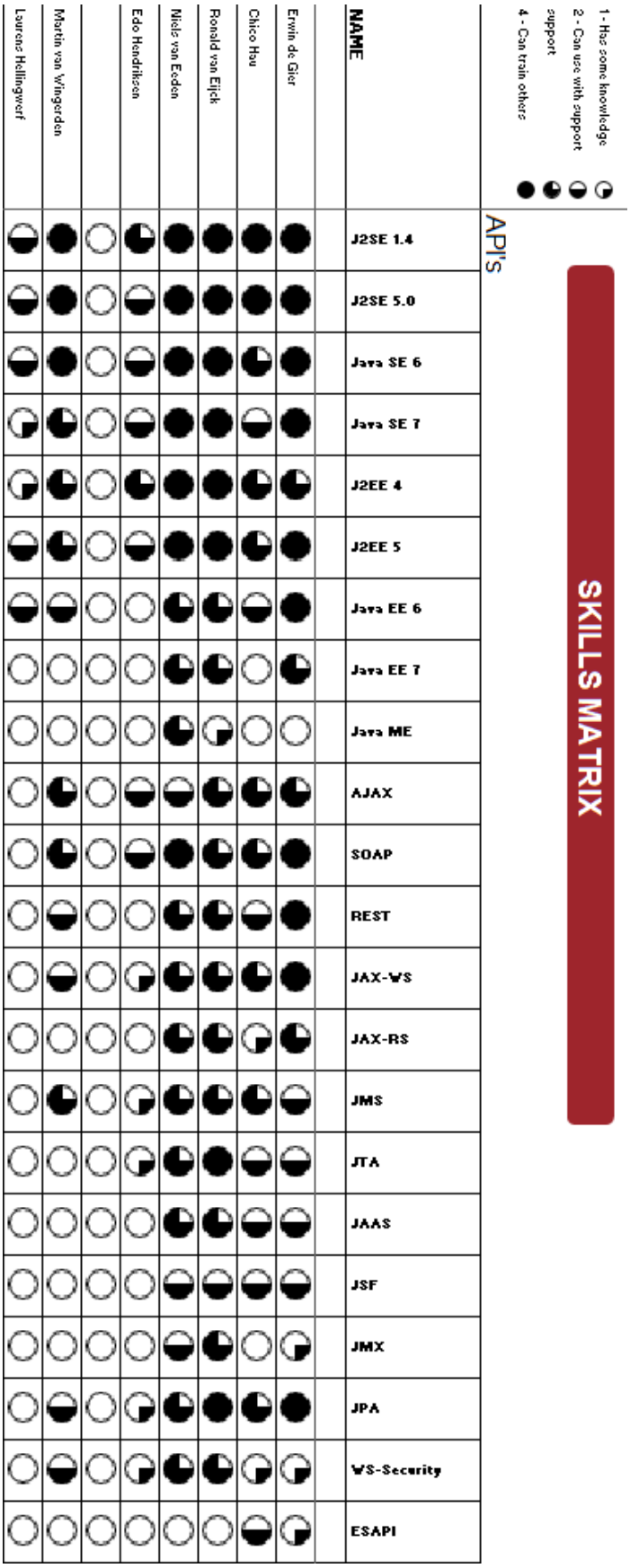
- Kenniskaarten definiëren
- Projecten definiëren
- Werknemers aan project koppelen

Ten eerste de functionaliteit "kenniskaarten definiëren". Deze functionaliteit is duidelijk te herkennen als kern van het systeem. Dit heeft dan ook hoge prioriteit. Het definiëren van kenniskaarten houdt in dat er nieuwe kenniskaarten aan het systeem kunnen worden toegevoegd die vervolgens door de werknemers kunnen worden ingevuld. Door deze kenniskaarten in te laten vullen door werknemers is het mogelijk om een helder beeld te krijgen van de expertise van de betreffende werknemer. Bij het definiëren van een nieuwe kenniskaart kan bijvoorbeeld worden gedacht aan het toevoegen van een nieuw framework aan de bekende vaardigheden in de skillsmatrix.

Ten tweede is de functionaliteit "projecten definiëren" een belangrijk onderdeel van het systeem. Kennis hebben over de expertise van werknemers is namelijk één ding, maar deze kennis moet ook effectief worden toegepast. Door projecten aan het systeem toe te kunnen voegen is het mogelijk om hier bepaalde eisen met betrekking tot kennis aan te hangen. Deze eisen zijn te vertalen naar de individuele skills in de skillsmatrix.

De derde belangrijke functionaliteit is het koppelen van werknemers aan projecten. Door deze functionaliteit is het mogelijk om te reflecteren op projecten. Zo kan in één oogopslag gezien worden of een project de juiste mensen met de juiste expertise toegekend heeft gekregen. Het moet dus mogelijk zijn om een overzicht te genereren tussen de benodigde kennis voor een project en de kennis die de projectleden met zich mee brengen of hebben opgedaan gedurende het project.

Deze drie functionaliteiten zijn reeds vastgesteld, echter is de rest van het systeem open voor interpretatie. Een deel van de opdracht zal uit het maken van een requirementanalyse bestaan. Het bedrijf heeft nog geen volledig antwoord op de vraag wat voor functionaliteit het systeem moet gaan bieden om de gewenste rol te vervullen. Hierin zullen de stakeholders van het project geïdentificeerd moeten worden. Wanneer de stakeholders geïdentificeerd zijn is het mogelijk om in overleg tot een concrete lijst requirements te komen voor het systeem. Deze requirements zijn in te delen aan de hand van een lijst waarin de prioriteit van elke functionaliteit kan worden vastgelegd. Door deze analyse op te stellen zijn de eisen en functionaliteiten van het systeem concreet te maken.



Figuur 2. Skillsmatrix

4 Vraagstelling

Dit hoofdstuk zal de vragen benoemen en omschrijven die tijdens het onderzoek leidend zijn. Allereerst zal de centrale vraag worden benoemd en toegelicht. Vervolgens zullen de bijbehorende deelvragen worden beschreven.

Uit de probleemstelling is een centrale vraag af te leiden. De vraag dient het gehele probleem te omvatten en alle aspecten te dekken. Deze centrale vraag luidt als volgt:

Op welke manier kan het kenniskaartensysteem van Sogeti's Java Centre of Excellence het beste geautomatiseerd worden, zodat Skill Management efficiënter verloopt?

Deze vraag vangt de doelstelling die Sogeti heeft voor het te ontwikkelen systeem. Het huidige proces is tijdrovend en de data staat verspreid over verschillende systemen. Een efficiënter systeem betekent in deze context dat het handmatige proces wordt geautomatiseerd. Daarnaast is de wens van het krijgen van een uniforme oplossing voor kennisinventarisatie hierin verwoord.

De centrale vraag is beantwoorden met gebruik van antwoorden op verscheidene deelvragen. Deze deelvragen dekken elk een deel van het probleem. De deelvragen die samen de centrale vraag opmaken zijn als volgt gedefinieerd:

- *Hoe werkt het handmatige kenniskaartenproces van Sogeti's Java Centre of Excellence?*
- *Aan welke functionele- en niet functionele requirements moet het kenniskaartensysteem voor Sogeti's Java Centre of Excellence voldoen?*
- *Is het mogelijk om de huidige data in de kenniskaarten van Sogeti's Java Centre of Excellence in het automatische systeem mee te nemen en zo ja, op welke manier(en)?*
- *Op welke manier(en) kan usability bijdragen aan de relevantie van de kenniskaarten?*
- *Hoe kan op architecturaal niveau uitbreidbaarheid en toekomstbestendigheid van de kenniskaartenapplicatie worden bereikt?*

5 Methoden en Aanpak

Dit hoofdstuk beschrijft de methoden en aanpak die tijdens het afstudeerproject toegepast zullen worden. Dit omvat onder andere de onderzoeksmethoden die toegepast zullen worden voor het beantwoorden van de vraagstelling bij het onderzoek. Daarnaast zullen randzaken met betrekking tot communicatie en ontwikkelmethoden in dit hoofdstuk worden besproken. De globale aanpak voor het onderzoek is weergegeven in tabel 1.

Deelvraag	Vraagtype	Aanpak	Paragraaf
Hoe werkt het handmatige kenniskaartenproces van Sogeti's Java Centre of Excellence?	Beschrijvend	Interviews	5.1
Aan welke functionele- en niet functionele requirements moet het kenniskaartensysteem voor Sogeti's Java Centre of Excellence voldoen?	Beschrijvend	Interviews	5.2
Is het mogelijk om de huidige data in de kenniskaarten van Sogeti's Java Centre of Excellence in het automatische systeem mee te nemen en zo ja, op welke manier(en)?	Vergelijkend, Ontwerpend	Desk research Literatuur	5.3
Op welke manier(en) kan usability bijdragen aan de relevantie van de kenniskaarten?	Ontwerpend	Literatuur Desk research Field research	5.4
Hoe kan op architecturaal niveau uitbreidbaarheid en toekomstbestendigheid van de kenniskaartenapplicatie worden bereikt?	Ontwerpend	Literatuur Desk research Field research	5.5

Tabel 1. Vraag en aanpak

5.1 Huidig kenniskaartenproces

Voor het in kaart brengen van het huidige kenniskaartenproces zullen interviews worden toegepast. Er zullen gedurende het onderzoek meerdere interviews gehouden worden. Interviews zijn een onderzoeksmethode die geschikt is voor het verkrijgen van kwalitatieve informatie (Steehouwer et al., 2006).

Deze interviews hebben als doel om het proces tot in detail in kaart te brengen. Dit is vereist omdat ditzelfde proces geautomatiseerd zal worden. Het is dus van belang dat dit proces helder is gemaakt. Om heldere en bruikbare data te verkrijgen is het van belang dat tijdens de interviews de juiste vragen aan de juiste personen worden gesteld. Van te voren zal dus een concrete vragenlijst en doelgroep moeten worden opgesteld.

Na het uitvoeren van de interviews zal de verkregen data verwerkt worden. De verwerkte informatie zal worden gevalideerd met persoon die is geïnterviewd. Deze informatie zal een procesgang

opleveren van het huidige kenniskaartenproces. Dit betreft niet alleen huidige ondersteunde functionaliteiten, maar ook hoe deze zich onderling verhouden. Verder zal een inzicht in de business zijn verkregen waar in het ontwerpen van het systeem de voordelen zich zullen voordoen. Hier kan bijvoorbeeld worden gedacht aan aspecten als: welke data moet er in het systeem komen? Of welke entiteiten moeten deze data gaan bevatten?

5.2 Requirementanalyse

De requirementanalyse zal uit drie stappen bestaan. Deze drie stappen volgen elkaar logischerwijs op aangezien ze afhankelijk zijn van elkaar. De drie stappen zien er als volgt uit:

- Identificeren stakeholders
- Interviewen/ondervragen stakeholders
- Verkregen data inventariseren en verwerken

De eerste stap, het identificeren van de stakeholders, is cruciaal. Stakeholders zijn in deze context belanghebbenden bij het te ontwikkelen systeem. De lijst met stakeholders zal worden opgesteld in overleg met de product owner (Edo Hendriksen) en de technisch begeleider (Erwin de Gier). De product owner heeft een duidelijke visie van de richting waar het systeem heen zal gaan. Verder weet de product owner veel van de interne bedrijfsstructuur af wat hem de aangewezen persoon maakt om stakeholders te identificeren.

De tweede stap, die opeenvolgend op de eerste stap loopt, is het interviewen van de stakeholders. Zoals eerder genoemd is het voor interviews van belang de juiste mensen de juiste vragen te stellen. Aan de eerste criteria is in dit stadium al voldaan. Dan rest dus slechts het opstellen van de juiste vragen. Wanneer dit is gebeurd kunnen de stakeholders geïnterviewd worden. Deze interviews zullen als belangrijkste resultaat een lijst met requirements voor het systeem opleveren. Wederom zal de verwerkte informatie in samenwerking met de geïnterviewden worden gevalideerd.

De laatste stap in het opstellen van de requirementanalyse is het inventariseren en verwerken van de verkregen data uit de interviews. De functionaliteiten zullen in overleg met de product owner worden ingedeeld naar prioriteit. Dit maakt het plannen van de realisering van het systeem in een later stadium gemakkelijker.

Het indelen naar prioriteit zal gebeuren met behulp van de MoSCoW methodiek. Dit een relatief simpele methodiek die niet tijdsintensief is (Verma, 2010). De methodiek is ontwikkeld voor het gebruik in een van de voorlopers van hedendaagse Agile ontwikkelmethodes: Dynamic Systems Development Method (Ramsin & Paige, 2008). Dit aangezien het prioriteren van functionaliteiten helpt bij het indelen van iteraties. Deze techniek kenmerkt zich door de functionaliteiten in te delen in vier categorieën die beginnen met een medeklinker uit het woord (**M**o**SC**o**W**). De categorieën zijn luiden als volgt:

- Must-haves
- Should-haves
- Could-haves
- Won't-haves

De vier categorieën lopen in aflopende mate van prioriteit van functionaliteiten; Must-have heeft de hoogste prioriteit, won't have de laagste.

5.3 Legacy data

In de tijd waarin het kenniskaartenproces handmatig is uitgevoerd is veel data vergaard. Het zou zonde zijn om deze data niet in het geautomatiseerde systeem mee te nemen. De huidige data staat in de Excel sheets die door de jaren heen zijn opgebouwd. Dit vereist dat er gezocht moet worden naar manieren om deze data te importeren in het systeem.

De technieken waarmee dit bereikt kan worden zullen aan het licht worden gebracht door digitale bronnen te raadplegen en eventueel ook gebruik te maken van literatuur. Hier zal voornamelijk worden gezocht naar het importeren van Excel data en eventueel naar beschikbare libraries. Digitale bronnen zullen bij uitstek geschikt zijn aangezien daar vaak handleidingen en/of opsommingen van verschillende libraries te vinden zijn. Vervolgens kunnen de resultaten worden uiteengezet en worden vergeleken met behulp van vooraf opgestelde criteria.

5.4 Actualiteit Kenniskaarten

Het beantwoorden van de deelvraag met betrekking tot het up to date houden van de kenniskaarten zal bereikt worden door literatuur te raadplegen. Dit is bij uitstek geschikt voor een onderwerp als usability, aangezien hier reeds veel over is geschreven. De uitkomst van deze literatuurstudie is een set met best practices of design patterns. Deze ontwerptechnieken kunnen in het proof of concept worden toegepast om de usability in het product hoog te krijgen.

Overigens is het een mogelijkheid om gebruik te maken van het expertise van de uX-werknemers binnen Sogeti. Deze zouden de usability van de applicatie kunnen testen of feedback verschaffen. Ze zouden echter ook tijdens het ontwerpen al betrokken kunnen worden wanneer de middelen hiervoor aanwezig zijn.

5.5 Architectuur

Voor het correct opzetten van de architectuur zal uitvoerig onderzoek moeten worden uitgevoerd. De doelstelling (zie hoofdstuk 3.2) beschrijft duidelijk dat het systeem uitbreidbaar moet zijn. Dit vereist dat de architectuur zodanig moet zijn opgezet dat dit gemakkelijk te bereiken is.

Het antwoord op de vraag hoe deze uitbreidbaarheid op architecturaal niveau te bereiken is zal met behulp van verschillende onderzoeksmethoden worden beantwoord. De onderzoeksmethoden die gebruikt zullen worden zijn een stuk literatuurstudie, desk research en field research.

Literatuur kan worden gebruikt om bijvoorbeeld bewezen oplossingen te zoeken voor het probleem. Dit zal waarschijnlijk een set met bewezen oplossingen voor een probleem opleveren in de vorm van een *design pattern.* Er zijn design patterns beschikbaar voor veel problemen en deze zijn in literatuur uitvoerig beschreven en vastgelegd.

Daarnaast kan er uiteraard naar bestaande applicaties en de structuur daarvan worden gekeken. Deze vorm van deskresearch kan voornamelijk mogelijke herbruikbare structuren of oplossingen op leveren. De oplossingen zouden vervolgens in het systeem kunnen worden toegepast.

Tot slot kan er een deel fieldresearch plaatsvinden. De technisch begeleider van de student is een software architect en bevindt zich op dezelfde locatie. Aangezien hij bedreven is in het ontwerpen van

systemen kan hij worden benaderd voor input. Verder kunnen ideeën en uitwerkingen bij de technisch begeleider worden gevalideerd.

5.6 Proof of Concept

De afstudeeropdracht zal als eindresultaat een proof of concept (PoC) opleveren. Het proof of concept zal de kennis die tijdens het beantwoorden van de deelvragen is verzameld toepassen. Het doel van het PoC is uiteindelijk om alle aspecten samen te brengen in een werkend stuk software. Deze software dient om aan te tonen dat het toepasbaar is voor de doeleinden die Sogeti bij de applicatie heeft.

Het PoC zal iteratief ontwikkeld worden (zie hoofdstuk 8.5: softwareontwikkelmethode) en dus zullen er steeds kleine brokken van het systeem worden opgeleverd. Belangrijk voor het systeem is dat er met het ontwerpen rekening gehouden wordt met de uitbreidbaarheid. Dit in verband met het mogelijk breder inzetten dan puur en alleen in het CoE. Daarnaast dient de correcte werking en de usability van het PoC getest en aangetoond te worden (zie hoofdstuk 8.6: kwaliteitsbewaking).

Vanuit Sogeti zijn er enkele technieken waarmee gewerkt dient te worden. Zo is het gebruikelijk dat de applicaties een REST backend hebben. Deze backend wordt ontwikkeld met het J2EE ontwikkelplatform in de Eclipse IDE. Voor de frontend wenst Sogeti dat er wordt ontwikkeld met HTML, CSS en JavaScript (JS). Specifieker met *AngularJS*. Angular is een JS framework wat veel gebruikelijke handelingen abstraheert en vergemakkelijkt. Zo maakt het onder andere bidirectional databinding mogelijk waarmee een view gemakkelijk een model kan updaten. Meer informatie is te vinden op de site van AngularJS (https://angularjs.org/). Tot slot zal het geheel gebruik maken van *Maven* voor buildautomatisering. Maven maakt het onder andere gemakkelijk om projectdependencies te beheren. Voor meer informatie over Maven kan de officiële site worden geraadpleegd (http://maven.apache.org/).

Al met al is het PoC een belangrijk resultaat van de opdracht. Dit schept voor de organisatie een helder beeld van wat de mogelijkheden zijn en hoe de geleerde concepten toegepast kunnen worden in de praktijk. Tot slot zou het PoC naderhand verder doorontwikkeld kunnen worden tot een daadwerkelijke in de praktijk te gebruiken applicatie.

5.7 Communicatie

Communicatie is van groot belang tijdens het ontwikkelen van een systeem. Dit project kent vier personen waarmee door de student regelmatig communicatie plaats dient te vinden. Dit zijn de docentbegeleider, de bedrijfsbegeleider, de technisch begeleider en de opdrachtgever.

Esther van der Stappen vervult de rol van docentbegeleider. Er dient voornamelijk gecommuniceerd te worden over deliverables die vereist zijn vanuit het opleidingsinstituut. De docentbegeleider is degene die beslist of de afstudeeropdracht haalbaar en van voldoende niveau is. Communicatie zal dan ook voornamelijk plaatsvinden via de mail. Verder zal de docentbegeleider (minimaal) één keer op bedrijfsbezoek komen bij de organisatie.

Willem van de Griendt zal tijdens het afstudeertraject de bedrijfsbegeleider zijn vanuit Sogeti. De bedrijfsbegeleider zal regelmatig op de hoogte moeten worden gesteld omtrent de voortgang van de student. Op deze manier is het mogelijk, indien noodzakelijk, om het project tijdig bij te sturen. De

afspraken die zijn gemaakt met de bedrijfsbegeleider omvatten een tweewekelijkse bilateraal en een wekelijkse update via de mail.

Erwin de Gier is de technisch begeleider van het project. Communicatie met Erwin zal voornamelijk tijdens de ontwerp en implementatie van het systeem plaatsvinden. Dit zal voornamelijk ontwerpkeuzes of verheldering met betrekking tot implementatievraagstukken betreffen. Communicatie zal voornamelijk in den lijve plaatsvinden aangezien de student in de aanvangsfase zich op dezelfde locatie zal bevinden.

Tijdens de aanvang van het project bleek dat de eisen en stakeholders nog onduidelijk waren. In eerste instantie zou Erwin de opdrachtgever zijn voor het project. Er bleek echter dat Erwin niet zoveel met de applicatie te maken had als veronderstelt was. Dit leidde er toe dat in overeenstemming met de technisch begeleider en bedrijfsbegeleider een andere opdrachtgever is gekozen. De opdrachtgever van het project zal Edo Hendriksen zijn. Met de opdrachtgever zal vooral in het begin veel afgestemd moeten worden. Dit zal in de vorm van meetings en mail verlopen. Verder zal de voortgang van elke timebox worden gecommuniceerd met de opdrachtgever.

5.8 Softwareontwikkelmethode

Het ontwikkelen van het PoC zal gestructureerd plaatsvinden volgens een zogenaamde softwareontwikkelmethode. Deze methoden structureren verschillende stadia van het softwareontwikkelproces. Afhankelijk van de bekende criteria kan een keuze worden gemaakt tussen verschillende methodieken.

Van de criteria die hierbij bekend zijn, is er één verstrekt door de opdrachtgever. De opdrachtgever heeft aangegeven dat de implementatie iteratief dient te verlopen. Daarnaast zal het project voornamelijk door één persoon worden uitgevoerd. De gekozen methodiek dient flexibel genoeg te zijn om voor een klein ontwikkelteam toepasbaar te zijn.

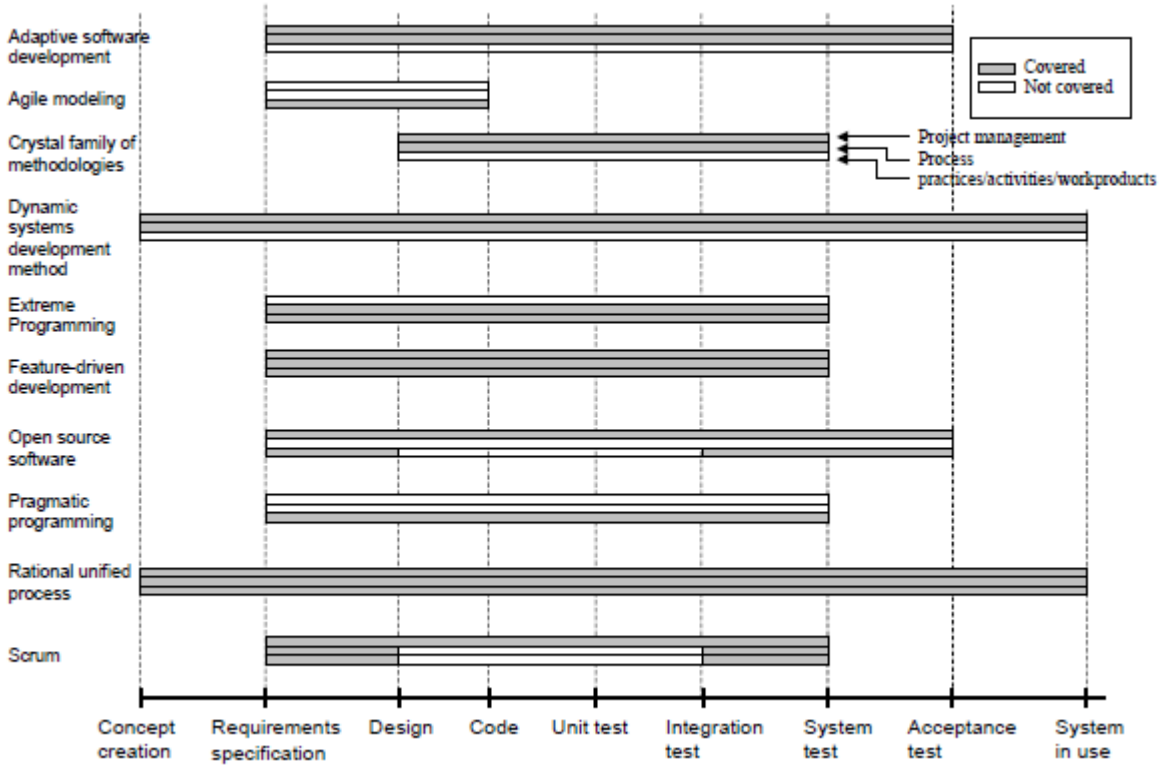
Aangezien de software iteratief dient te verlopen leidt dit al gauw tot een softwareontwikkelmethode in de categorie *Agile* methodieken. Volgens Abrahamsson, Salo, Ronkainen en Warsta (2002) wordt Agile gekenmerkt door vier kernwaarden. Ook al zijn het eerste en het derde punt meer bedrijfsgericht, punt twee en punt vier zijn daarbuiten ook goed toepasbaar voor de doeleinden van het huidige project. De kernwaarden luiden als volgt:

1. Individuen en interacties gaan voor processen en tools
2. Werkende software gaat voor uitgebreide documentatie
3. Samenwerking met de klant gaat boven contractonderhandelingen
4. Reageren op verandering gaat boven het volgen van een plan

Punt twee benadrukt in essentie het belang van het opleveren van werkende stukken software, hiermee krijgt de klant een beeld van de software. Regelmatige opleveringen van software maken tijdige bijsturing mogelijk. Dit sluit aan bij de wens van de opdrachtgever om ontwikkeling iteratief te laten verlopen.

Het vierde punt beschrijft het feit dat er op verandering ingespeeld moet kunnen worden. Dit is een punt dat ook aansluit bij de context van het project. De verandering in de huidige context betreft dan echter het ontbreken een concrete lijst requirements voor het te ontwikkelen systeem. Daarnaast zou het in theorie mogelijk zijn om op verandering in prioriteit van functionaliteiten in te kunnen spelen. De iteratieve denkwijze van Agile maakt het mogelijk om hier snel op in te spelen.

De Agile denkwijze wordt toegepast in verschillende methodieken. Deze methodieken geven elk een eigen invulling aan toepassing binnen projecten. Tussen methodieken kunnen bijvoorbeeld regels, processen, documentatie en scope verschillen. Sommige methodieken definiëren een zeer uitgebreide set met regels en procesgang zoals RUP (Rational Unified Process) of Scrum en anderen zoals OSS (Open Source Software) dienen meer als guideline (Abrahamsson et al., 2002). Abrahamsson et al. (2002) zette tien Agile methodieken uiteen en gaf de verschillen met betrekking tot scope grafisch weer. Deze uiteenzetting is terug te vinden in figuur 3.



Figuur 3. Uiteenzetting Agile methoden

Als softwareontwikkelingmethode zal OpenUP worden gebruikt. OpenUP is een opensource versie van het door IBM beheerde RUP . Deze methodiek beschrijft een framework voor het gehele proces van softwareontwikkeling. Hieronder valt alles van verdieping tot daadwerkelijke integratie van de software. Het framework is aanpasbaar waar nodig en dit biedt flexibiliteit in de toepassing van de methodiek. Zo kan er voor worden gekozen om bepaalde rollen niet toe te passen of om invulling te geven aan de implementatie van de gedefinieerde fasen binnen OpenUP. RUP wordt niet als echt Agile gezien (Abrahamsson et al., 2002) echter kan de Construction fase iteratief plaatsvinden waarmee de criteria van de opdrachtgever is vervuld.

Naast het feit dat de methodiek aanpasbaar is biedt het nog twee pluspunten ten opzichte van de andere methodieken. Allereerst wordt OpenUP beheerd door de Eclipse Foundation. Er zijn dan ook integratietools beschikbaar voor de IDE die gebruikt zal worden tijdens het project: Eclipse. Ten tweede is er reeds met OpenUP gewerkt, aangezien het onderdeel is van het curriculum van de opleiding Software Engineering aan het opleidingsinstituut.

OpenUP zal binnen het project toegepast worden door middel van het toepassen van de vier gedefinieerde fasen: Inception, Elaboration, Construction en Transition. De nadruk zal liggen bij de eerste drie fasen. Inception is hier een globale oriëntatie in de business. Elaboration is het verdiepen in de eisen van het te ontwikkelen systeem. Construction is het daadwerkelijke implementeren van het systeem. De Construction zal iteratief plaatsvinden met behulp van opgestelde tijdseenheden genaamd "Time-boxes". Aan de hand van de projectrequirements die gedurende de Elaboration-fase op tafel komen krijgen de time-boxes invulling. Tijdens het project zullen timeboxes van twee weken worden aangehouden.

5.9 Kwaliteitsbewaking

Kwaliteitsbewaking is een belangrijk aspect van het projectverloop. Tijdens het afstudeertraject zal kwaliteitsbewaking op verschillende manieren terugkomen. Deze luiden als volgt:

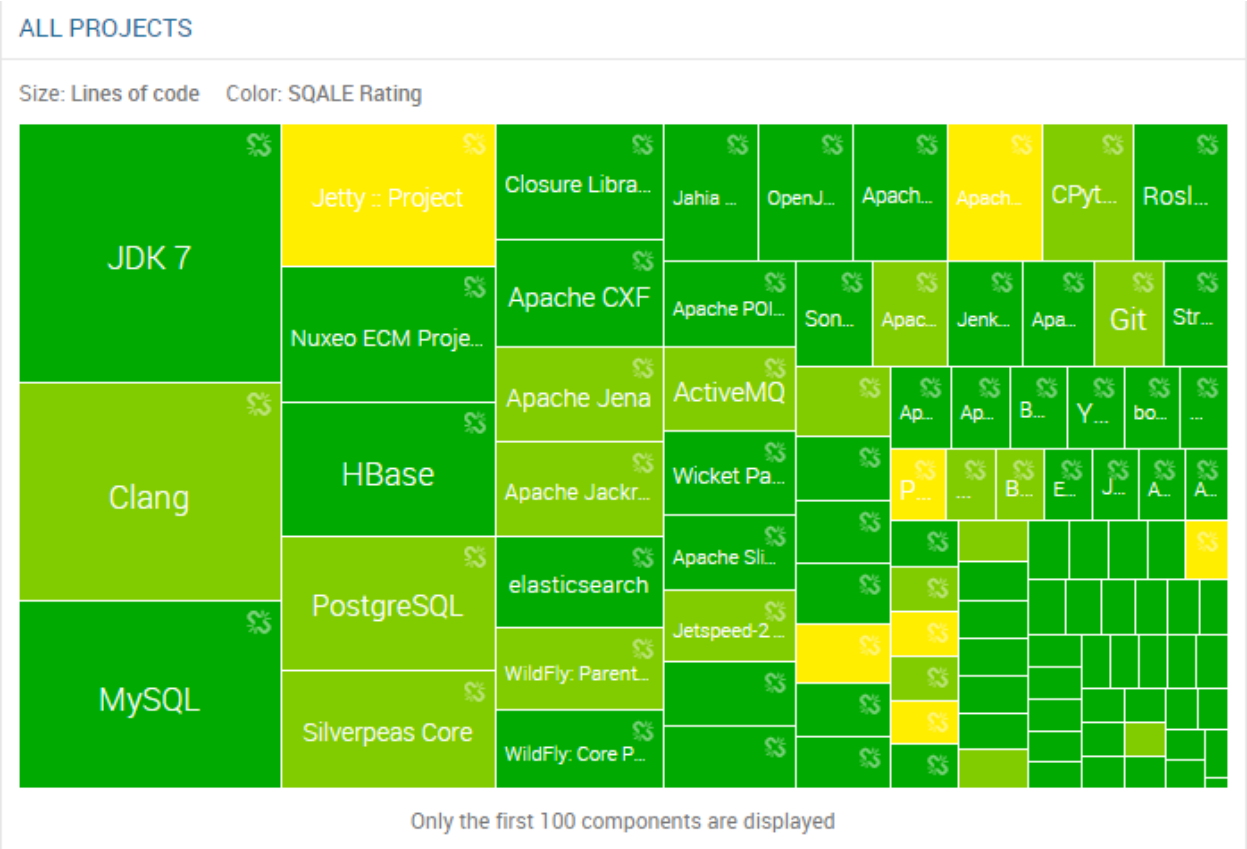
- Functionaliteitstesten
- Usabilitytesten
- Codequality analyse

De functionaliteiten van het systeem dienen getest te worden. Er moet kunnen worden aangetoond dat het systeem correct werkt. Hiervoor zullen white box testen worden uitgevoerd. Dit zal worden gerealiseerd met *JUnit*. De keuze voor JUnit is gemaakt vanwege de hoge mate van integratie in de Java SDK, wat het gebruik vergemakkelijkt. Aangezien het implementeren van het systeem iteratief zal verlopen kan ook het testen in een vroeg stadium beginnen. Technisch begeleider E. de Gier (persoonlijke communicatie, 6 maart, 2015) heeft aangegeven dat het streven een code coverage van rond de tachtig procent is.

Naast de waarborging van de functionaliteiten van het systeem, zal ook de usability van het systeem getest moeten worden. Nogmaals, het hebben van relevante data zal van groot belang zijn voor de effectiviteit van het systeem. Een applicatie met een goede usability zal de bruikbaarheid van het systeem ten goede komen. Het is dan voor werknemers prettiger om gebruik te maken van het systeem.

Om de usability te testen zal er gebruik worden gemaakt van gebruikerstests. Gebruikerstesten worden door de Nielsen Norman Group (2012) benoemd als manier om de usability op de proef te stellen. De testen zullen kortweg inhouden dat er een testgroep opgesteld zal worden die met het systeem enkele interacties zullen uitvoeren. Vervolgens kan de gebruikers om hun ervaring en feedback worden gevraagd. De testgroep zal zoveel mogelijk bestaan uit mensen die het huidige systeem gebruiken en de applicatie zullen gaan toepassen. Door de gebruikerstesten gedurende de ontwikkeling te laten plaatsvinden kan op de feedback worden ingespeeld.

Tot slot zal de kwaliteit van de code worden geanalyseerd met behulp van de tool *SonarQube* (http://www.sonarqube.org/). Dit genereert een overzicht van de codebase die wordt geanalyseerd. Hiermee kunnen onder andere issues worden gedetecteerd aan de hand van een verzameling regels. Voorbeelden van deze regels zijn bijvoorbeeld naming conventions. Er zijn standaard regelsets beschikbaar voor verschillende talen. Een voorbeeld van een dashboard in SonarQube is weergegeven in figuur 4.



Figuur 4. Fragment SonarQube dashboard

De dashboards die gegenereerd worden geven een indruk van de algemene kwaliteit van de code en verbinden hier een *SQALE* (Software Quality Assessment based on Lifecycle Expectations) rating aan. Deze rating wordt gebruikt om de toekomstbestendigheid en onderhoudbaarheid van code uit te drukken. De methode definieert een aantal indices waar de code op geïnspecteerd moet worden. Bij deze indices kan bijvoorbeeld worden gedacht aan een index voor security, portability en efficiency. Meer over SQALE is te vinden op de officiële website (http://www.sqale.org/).

6 Risico's

In dit hoofdstuk zullen de belangrijkste risico's die zich tijdens het afstudeerproject kunnen voordoen worden toegelicht. De risico's zullen worden beschreven door te omschrijven hoe ze kunnen ontstaan tijdens het project. Daarnaast zullen er mitigerende maatregelen per risico worden omschreven.

Risico omschrijving	Ontstaansreden	Mitigerende maatregel
Producteisen onduidelijk/inaccuraat	Slechte communicatie en/of afstemming met de begeleiding vanuit de organisatie.	Regelmatige contactmomenten met begeleiding vanuit de organisatie. Mogelijkheid bieden tot bijsturing van de opdracht. Grondige requirementanalyse uitvoeren
Onvoldoende tijd ontwerp en/of implementatie	• Onvoldoende ruimte in planning toegewezen. • Onduidelijke/ontoepasselijke softwareontwikkelingsmethode	Duidelijke, ruime en realistische planning opstellen. Toepasselijke softwareontwikkelingsmethode gebruiken met regelmatige oplevering van resultaten. Heldere definitie op te leveren functionaliteiten, prioriteiten en scope.
Ontoepasbaarheid systeem buiten CoE	Onjuiste systeemarchitectuur	Tijdens het ontwerp rekening houden met uitbreidbaarheid op hoog niveau. Waar nodig hulp inschakelen van technisch begeleider.
Onheldere conclusie gebruikerstests	Slechte afbakening testgroep	Duidelijke/realistische gebruikersgroep opstellen. Mocht er zich alsnog onenigheid voordoen, dan zal de keuze bij de opdrachtgever worden voorgelegd.
Terughoudendheid invullen data bij werknemers	Kenniskaart benadrukt het ontbreken van kennis.	Positieve aspecten benadrukken in de applicatie. Bijv. dat het t.b.h.v. zelfontwikkeling is.

Tabel 2. Risico's

7 Planning

Dit hoofdstuk beschrijft de planning gedurende het afstudeerproject. Hierin zullen de taken worden gespecificeerd. Vervolgens zullen de taken ingedeeld worden en zal er een tijdsschatting aan worden verbonden. De taken zullen aan de hand van startdatum worden ingedeeld. Naast de takenplanning zullen de deliverables tijdens dit project ook worden ingedeeld. Dit verschaft een grove chronologische volgorde voor het afstudeertraject.

7.1 Takenplanning

Het afstudeerproject bestaat uit diverse taken. Taken die zijn opgesteld aan de hand van eisen van het opleidingsinstituut en taken die betrekking hebben tot de daadwerkelijke opdracht. Deze taken zijn in de onderstaande tabel (zie tabel 2) zoveel mogelijk in kaart gebracht. De taken zijn ingedeeld op geplande startdatum, waarbij de omschrijving en het geschatte aantal uren is gegeven. Vervolgens wordt er een verwachte einddatum bij genoemd. De taken met lichtrode arcering zijn hierbij iteratieve handelingen.

Geplande startdatum	Taak	Geschatte tijdsduur(in dagen)	Geplande einddatum
2-2-15	Start afstuderen/OCJP certificaat behalen	10	13-2-15
16-2-15	Plan van aanpak opstellen	15	20-3-15
20-3-15	Scriptie opstellen	15	2-6-15
20-3-15	Interviews kenniskaartenproces afnemen en verwerken	3	20-3-15
23-3-15	Requirementanalyse uitvoeren	10	3-4-15
6-4-15	Literatuurstudie usability en architectuur uitvoeren	3	8-4-15
9-4-15	Timebox 1. Bestaande uit de onderstaande iteratieve onderdelen.	10	24-4-15
9-4-15	Functioneel ontwerpen	5	2-6-15
9-4-15	Technisch ontwerpen	10	2-6-15
9-4-15	Implementeren systeem	15	2-6-15
9-4-15	Testen functionaliteiten	5	2-6-15
9-4-15	Testen usability	3	2-6-15
27-4-15	Timebox 2. Bestaande uit de bovenstaande iteratieve onderdelen.	10	8-5-15

11-5-15	Timebox 3. Bestaande uit de bovenstaande iteratieve onderdelen.	10	22-5-15
25-5-15	Uitloop, afronding en overdracht.	7	2-6-15
3-6-15	Presentatie afstudeerzitting maken	3	26-6-15

Tabel 3. Takenplanning

7.2 Planning deliverables

De taken die worden verricht hebben resultaten tot gevolg. Sommige van deze resultaten hebben bijbehorende deadlines. Deze deadlines hebben voornamelijk betrekking tot de resultaten die vereist zijn vanuit het opleidingsinstituut. Deze resultaten of deliverables zijn in tabel 3 terug te vinden. De deliverables zijn hier kort beschreven, verder bevindt zich hierbij de deadline en tot slot zijn de personen genoemd bij wie het moet worden opgeleverd.

Deadline	Deliverable	Relevante personen en/of entiteiten
28-2-15	Plan van aanpak: concept	Esther van der Stappen
20-3-15	Plan van aanpak: definitief	Esther van der Stappen Willem van de Griendt Afstudeeradministratie
20-3-15	Ondertekend afstudeercontract	Esther van der Stappen Willem van de Griendt Afstudeeradministratie Roy Straub
2-4-15	Scriptie: conceptversie één	Esther van der Stappen
12-5-15	Scriptie: conceptversie twee	Esther van der Stappen
2-6-15	Scriptie: definitief	Esther van der Stappen Willem van de Griendt Afstudeeradministratie

Tabel 4. Planning deliverables

8 Bronvermelding

Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J. (2002). *Agile software development methods: Review and analysis.* Oulu: VTT Electronics.

Alavi, M., Leidner, D. E. (1999). Knowledge management systems: issues, challenges, and benefits. *Communications of the AIS*, *1*(2es), 1.

Alavi, M., Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly*, 107-136.

Beck, S. (2003). Skill and competence management as a base of an Integrated Personnel Development (IPD). *Journal of Universal Computer Science*, 9(12), 1381-1387.

Confluence (2015). Confluence. *Overview*. Geraadpleegd op 12 maart 2015 via https://www.atlassian.com/software/confluence.

Garro, A., Palopoli, L. (2003). An xml multi-agent system for e-learning and skill management. In *Agent Technologies, Infrastructures, Tools, and Applications for E-Services* (pp. 283-294). Springer Berlin Heidelberg.

Nielsen Norman Group (2012). Usability. *Usability 101: Introduction to usability.* Geraadpleegd op http://www.nngroup.com/articles/usability-101-introduction-to-usability/.

Nonaka, I. (1991). The knowledge-creating company. *Harvard business review*, 69(6), 96-104.

Raman, R., Paige, R. F. (2008). Process-centered review of object oriented software development methodologies. *ACM Computing Surveys, 40(1),* 3:1-3:89. doi: 10.1145/1322432.1322435

Sogeti (2015). Sogeti Nederland. *Facts en Figures.* Geraadpleegd op 24 februari 2015 via https://www.sogeti.nl/over-sogeti/facts-en-figures.

Sogeti (2015, januari). Bedrijfsstructuur Sogeti Nederland. *Introductieprogramma 2015.* Geraadpleegd op https://sharenet.sogeti.nl/.

Steehouwer, M., Jansen, C., Maat, K., Staak, J. van der, Vet, D. de, Witteveen, M., & Woudstra, E. (2006). *Leren communiceren: Handboek voor mondelinge en schriftelijke communicatie* (5e dr.). Groningen/Houten: Wolters-Noordhoff.

TalentAlign (2012, december). Skills and competencies. *Skills vs. Competencies. What's the difference?* Geraadpleegd op http://www.talentalign.com/skills-vs-competencies-whats-the-difference/.

TribeHR (2015). Talent management. *Skills tracking*. Geraadpleegd op 12 maart 2015 via http://tribehr.com/social-hr-software/talent-management/skills-tracking.

Verma, A. (2010). *Requirements analysis phase reengineering: Decision making & requirements prioritisation among stakeholders for a new software project.* Geraadpleegd op http://undergraduate.csse.uwa.edu.au/year4/Current/Students/Files/2010/AbhijeetVerma/CorrectedDissertation.pdf

9 Contactpersonen

Willem van de Griendt

Telefoon:	+31(0)88 660 66 00 (nakiesnummer 2781)
Mobiel:	+31(0)6 50 60 13 76
Email:	willem.vande.griendt@sogeti.nl

Erwin de Gier

Telefoon:	+31(0)88 660 66 00 (nakiesnummer 2625)
Mobiel:	-
Email:	erwin.de.gier@sogeti.nl

Edo Hendriksen

Telefoon:	-
Mobiel:	+31 (0)6 51 82 67 06
Email:	edo.hendriksen@sogeti.nl

Esther van der Stappen

Telefoon:	088 481 82 83
Mobiel:	-
Email:	esther.vanderstappen@hu.nl

Roy Straub

Telefoon:	-
Mobiel:	+31(0)6 83 67 81 90
Email:	roy.straub@sogeti.nl

B. Zelfreflectie

Gedurende het afstudeertraject heb ik veel nieuwe kennis op moeten doen. Deze kennis behoorde tot verschillende aspecten. Dit waren voornamelijk de volgende aspecten:

- Technisch
- Communicatief
- Schriftelijk

Op het technische aspect heb ik voornamelijk veel nieuwe kennis opgedaan op het gebied van webontwikkeling. Tot voor mijn afstuderen had ik hier zeer weinig ervaring mee. Dit was dan ook één van de grootste initiële uitdagingen aangezien het eindproduct een webapplicatie betrof. Ik heb in een korte tijd veel geleerd van collega's en door het implementeren van het eindproduct. Echter zou ik mij nog meer willen ontwikkelen op dit gebied, vooral op het gebied van AngularJS. Ook al heb ik de basis eigen gemaakt, ben ik me er van bewust dat de library veel meer mogelijkheden biedt dan waar gebruik van is gemaakt. Dit kan in de toekomst voor mij leidden tot de ontwikkeling van betere applicaties. Door trainingen te volgen bij het bedrijf, vragen om inzicht bij collega's en projecten uit te voeren hoop ik mij hierin verder te kunnen ontwikkelen.

Op communicatief gebied heb ik vooral in het begin moeten wennen aan het feit dat ik deel maakte van een grote organisatie. Tot de aanvang van mijn afstuderen had ik hier geen ervaring mee. Gedurende het afstudeertraject moest ik leren om tussen vele verschillende partijen te zitten en hiermee te overleggen. Een aspect hiervan was ook het uitvoeren van een requirementanalyse. Dit uitvoeren in een bedrijfscontext was compleet nieuw voor mij. Vanwege de sterke afhankelijkheid van het hebben van goede, heldere requirements en het afleveren van een product waar de klant tevreden mee is, is dit een aspect waar ik me meer op wil ontwikkelen. Communicatie met opdrachtgevers en stakeholders vormt een groot deel van het softwareontwikkelproces en is van cruciaal belang. Ik denk mij hier in te kunnen ontwikkelen door deel te nemen aan grootschalige projecten in het bedrijfsleven.

Tot slot heb ik mij wederom moeten ontwikkelen op schriftelijk gebied. Het schrijven van een scriptie was een uitdaging. Mede vanwege veelvuldige feedback heb ik mijzelf kunnen ontwikkelen ten opzichte van de vorige keer dat ik een scriptie moest indienen. Ditmaal heb ik getracht meer te denken vanuit het perspectief van de lezer om de scriptie helder en leesbaar te kunnen maken. Dit is een vaardigheid die in alle schriftelijke communicatie van belang is.

C. Antwoorden interviews kenniskaartenproces

Interview Christiaan Bennik (16-04-15)

1. Wat moet er gebeuren als er een kenniskaart voor een project moet worden opgesteld?

Allereerst naar de klant toespitsen. Dit is het kijken welke kennis er specifiek nodig is voor deze klant. Daarnaast afspraken maken met de klant. Kijken wanneer welke expertiseniveaus van toepassing zijn en hierin een overeenstemming bereiken. Vervolgens wordt algemene kennis voor het project wordt vastgelegd (Java, Oracle etc.). Daarna laat ik de kenniskaart individueel door werknemers invullen. Tot slot voeg ik dit samen in een overall sheet. Een nieuwe werknemer krijgt binnen de eerste maand de kenniskaart voor zijn kiezen om deze in te vullen. Eén keer per half jaar laat ik de kenniskaart updaten en deel ik deze met de klant.

2. Wat moet er gebeuren als er de kennis van een werknemer in de kenniskaart vastgelegd moet worden?

De medewerker allereerst de kenniskaart laten invullen. Soms gebeurt dit samen, in de meeste gevallen maakt de medewerker zelf een inschatting van zijn niveau. Ikzelf controleer of dit correct is ingevuld. Meestal is dit wel correct, de niveaus kunnen breed genomen worden.

3. Welke gebruikersgroep/personen vullen de vaardigheden van werknemers in?

Dat doen ze in principe zelf. Alle leden die aan een project meewerken vullen dit in. Heel soms een uitzondering waarop de medewerker het niet zelf invult. Dit is een gevolg van het feit dat deze medewerker niet onder hetzelfde contract valt als de rest van de projectleden.

4. Welke gebruikersgroep/personen vullen de projecteisen in?

Ikzelf als service coördinator zijnde vul deze in samen met de architect, hij kan bepalen welke technische eigenschappen nodig zijn. De architect helpt mij met het vastleggen van de initiële technische aspecten. Verder stem ik de projecteisen af met de klant.

5. Hoe worden de categorieën en expertises bepaald waarvan het kennisniveau vastgelegd moet worden?

De technische expertises komen voornamelijk van de kant van de architect. De functionele expertises stem ik verder zelf af met de klant. Verdere competenties staan niet op de kenniskaart.

6. Wanneer en aan de hand waarvan worden deze categorieën en expertises bijgewerkt?

Qua extra expertises is er niet gek veel bijgekomen. Dit moet nu wel gebeuren aangezien er over wordt gegaan naar een nieuw platform. Dit is dus het resultaat van veranderingen op het technische vlak.

7. Welke gebruikersgroep/personen raadplegen de kenniskaart?

Ikzelf. Daarnaast ook de klant.

8. Met welke doeleinden en frequentie doen zij dit?

Voor mijzelf voor de borging van de expertise richting de klant. Dit om te laten zien hoe de kennis geborgd is. Voor de klant is dit een contracteis. Elke keer als hij wordt geüpdate wordt de kenniskaart gedeeld met de klant. Dit gebeurd gebruikelijk eens per half jaar. Veel vaker is dit niet nodig aangezien er niet van zoveel fluctuatie sprake is.

9. **Hoe vaak en met welke aanleiding worden de projecteisen in de kenniskaarten aangepast?**

Dit gebeurt wanneer er technische aspecten van het project veranderen. Dit kan bijvoorbeeld komen door het toevoegen van nieuwe functionaliteiten. Vaak gebeurt dit alleen wanneer er een grote wijziging plaatsvindt in de applicatie zelf.

10. **Hoe vaak en met welke aanleiding worden de kennisniveaus van werknemers in de kenniskaarten aangepast?**

Na een half jaar heb ik zelf gevraagd of de medewerkers hun expertiseniveaus wilden bijwerken. Soms is dit ook een gevolg van het feit dat een medewerker zelf een technische cursus of iets dergelijks heeft gevolgd. Dit moeten ze dan zelf aangeven.

D. Antwoorden interviews requirementanalyse

Interview Edo Hendriksen (09-04-15)

1. **Wie moeten er van het kenniskaartensysteem gebruik gaan maken?**

In eerste instantie voor gebruikers binnen het CoE Java. Het zal hoofdzakelijk drie soorten gebruikers betreffen: managers, medewerkers en beheerders. Managers zijn leidinggevenden in projecten of bijvoorbeeld opdrachten. Tevens zijn zij hoofdgebruiker van het systeem. De medewerkers kunnen worden gezien als de teamleden tijdens een project. Dit zijn de personen waar de manager de expertiseniveaus inzichtelijk van wilt hebben. Beheerders zijn de personen die de applicatie beheren.

2. **Met welke doeleinden doen zij dit?**

In eerste instantie voor gebruikers binnen het CoE Java. Het zal hoofdzakelijk drie soorten gebruikers betreffen: managers, medewerkers en beheerders. Managers zijn leidinggevenden in projecten of bijvoorbeeld opdrachten. Tevens zijn zij hoofdgebruiker van het systeem. De medewerkers kunnen worden gezien als de teamleden tijdens een project. Dit zijn de personen waar de manager de expertiseniveaus inzichtelijk van wilt hebben. Beheerders zijn de personen die de applicatie beheren.

3. **Wat zou je idealiter met het kenniskaartensysteem bereiken?**

Het doel van het systeem is gelijk aan die van de huidige kenniskaart in Excel. Het belangrijkste voordeel van de applicatie is dat het centraal bereikbaar is en dat er één versie van de kenniskaarten aanwezig is. Daarnaast is uitbreidbaarheid belangrijk. Een applicatie maakt geavanceerde functionaliteiten mogelijk.

4. **Wat moet het kenniskaartensysteem kunnen om dat doel te bereiken?**

Er is een lijst opgesteld met 24 user stories. Hiervan zijn de eerste acht functionaliteiten die reeds in de Excel sheets zitten. Vanaf user story negen volgen de meer geavanceerde features.

- Als gebruiker wil ik kunnen inloggen met mijn Sogeti gebruikersnaam en wachtwoord, zodat ik geen aparte gebruikersnaam en wachtwoord hoef te onthouden. Ik wil ook weer kunnen uitloggen.
- Als beheerder wil ik gebruikers kunnen toevoegen en ze rechten kunnen geven (beheerder, manager, medewerker), zodat ze de applicatie kunnen gebruiken. Ik wil ook gebruikers die nergens aan zijn gekoppeld kunnen verwijderen.
- Als manager wil ik kenniskaarten kunnen definiëren zodat ik kan bepalen voor welk project of welke opdracht ik de benodigde en aanwezige expertise wil vastleggen. Ik wil alleen mijn eigen kenniskaarten zien en openen en ook mijn kenniskaarten kunnen verwijderen.
- Als manager wil ik in een kenniskaart de benodigde expertises en bijbehorende kennisniveaus voor een project of opdracht kunnen invullen of wijzigen, zodat ik kan vastleggen voor welke expertises welke kennisniveaus nodig zijn.
- Als manager wil ik medewerkers kunnen toevoegen, zodat ik ze aan een kenniskaart kan koppelen.

- Als manager wil ik medewerkers aan een kenniskaart kunnen koppelen, zodat ik kan bepalen welke medewerkers de kenniskaart moeten invullen.
- Als medewerker wil ik een kenniskaart waar ik aan gekoppeld ben kunnen invullen of wijzigen, zodat ik voor elke expertise kenbaar kan maken welk kennisniveau ik heb.
- Als manager wil ik in een kenniskaart de aanwezige kennisniveaus kunnen vergelijken met de benodigde kennisniveaus zodat ik maatregelen kan nemen indien expertise ontbreekt of onvoldoende is. Ik wil in één oogopslag kunnen zien welke expertise ontbreekt of in welke mate onvoldoende is. Daarnaast wil ik de kennisniveaus van alle afzonderlijke teamleden kunnen zien.
- Als manager wil ik andere managers kunnen toevoegen aan de kenniskaart, zodat zij ook de kenniskaart kunnen beheren.
- Als manager wil ik teams van medewerkers kunnen beheren, zodat ik specifieke groepen medewerkers heb om aan een kenniskaart te koppelen. Ik wil een team ook kunnen wijzigen of verwijderen.
- Als manager wil ik een team kunnen koppelen aan een kenniskaart zodat ik een specifieke groep medewerkers in de kenniskaart heb. Ik wil een team ook kunnen ontkoppelen.
- Als manager wil ik expertiseblokken kunnen definiëren, zodat ik blokken van expertises in meerdere kenniskaarten kan toepassen. Ik wil een expertiseblok ook kunnen wijzigen of verwijderen.
- Als manager wil ik expertiseblokken kunnen koppelen aan een kenniskaart zodat ik een specifiek blok van expertises kan toekennen aan een kenniskaart. Ik wil een expertiseblok ook kunnen ontkoppelen.
- Als manager wil ik een kenniskaart kunnen kopiëren, zodat ik snel een template heb voor nieuwe kenniskaart (zonder medewerkers) voor een ander project of andere opdracht.
- Als manager wil een kenniskaart kunnen maken die logisch hoort bij een andere kenniskaart, zodat ik voor deelprojecten of -opdrachten, of voor specifieke rollen binnen een project, een eigen kenniskaart kan maken. In het overzicht van kenniskaarten wil ik bij de kenniskaart van een project of opdracht alle eventuele aanvullende kenniskaarten zien.
- Als manager wil ik binnen de expertises onderscheid kunnen maken tussen hoofdexpertises en subexpertises, tot maximaal 3 niveaus diep, zodat ik een hiërarchisch expertisemodel kan hanteren; ik wil dat informatie kan worden ingevuld op ieder niveau en dat als bij subexpertises niveau kennisniveaus worden ingevuld deze automatisch resulteren in een niet-wijzigbare gemiddelde waarde op de bovenliggende niveaus.
- Als gebruiker wil ik de kenniskaartenapplicatie kunnen openen in Confluence, zodat ik geen aparte applicatie hoef te openen.
- Als manager wil ik kunnen kiezen uit alle expertises die eerder zijn ingevuld in deze of andere kenniskaarten, zodat ik een expertise niet onnodig zelf hoef in te voeren en expertises consistent worden ingevoerd over alle kenniskaarten. Ik wil hiervoor een slim zoekveld kunnen gebruiken, die suggesties doet op basis van mijn invoer in een vrij tekstveld.
- Als medewerker wil ik dat als ik een kennisniveau bij een expertise invul in een kenniskaart deze waar van toepassing ook wordt ingevuld of gewijzigd in andere kenniskaarten, zodat ik geen waardes dubbel hoef in te vullen of wijzigen.
- Als beheerder wil ik een manager kunnen verwijderen, zodat ik de database kan opschonen als een manager de applicatie niet meer gebruikt. De bij deze manager behorende kenniskaarten die niet ook aan een andere manager zijn gekoppeld wil ik ofwel mee kunnen verwijderen, ofwel aan een andere manager kunnen koppelen.
- Als beheerder wil ik dubbel ingevoerde expertises en medewerkers kunnen samenvoegen, zodat geen dubbelingen optreden in het systeem. Waar verschillende kennisniveaus zijn ingevuld voor de te ontdubbelen expertises wil ik dat de meest recente behouden blijft.
- Als manager wil ik expertiseniveaus van medewerkers kunnen aanpassen en algemene niveaucorrecties over alle medewerkers heen kunnen uitvoeren, zodat ik tot betere (gewogen) gemiddeldes kom.
- Als manager wil ik ook kunnen kiezen uit expertises uit de CV's (via BIS), zodat ik deze niet zelf hoef in te voeren.
- Als medewerker wil ik bij een uit BIS geïmporteerde expertise default de bijbehorende kennisniveau ingevuld hebben, zodat ik deze niet zelf hoef in te vullen.

5. **Geordend van hoge naar lage prioriteit: wat is prioriteit van deze functionaliteiten?**

De user stories zijn logisch opvolgend in afnemende prioriteit. Alle functionaliteiten die in het huidige systeem zitten hebben hoge prioriteit. Dit zijn de eerste acht user stories.

6. **Op welke manier moet het kenniskaartensysteem deze functionaliteiten bieden? (Hierbij kan aan snelheid of security worden gedacht)**

De veiligheid dient te voldoen aan de eisen die gelden voor andere vergelijkbare systemen binnen het CoE Java. Verder is het gewenst dat alle gebruikers het systeem gelijktijdig kunnen gebruiken.

Interview Christiaan (16-04-15)

1. **Wie moeten er van het kenniskaartensysteem gebruik gaan maken?**

Ikzelf, dus de projectmanager. Verder zou het handig zijn als de klant toegang heeft, of in ieder geval de kenniskaart extern raad te plegen is. De medewerkers moeten zelf hun kenniskaart kunnen wijzigen. Eventueel nog de review manager vanuit de klant.

2. **Met welke doeleinden doen zij dit?**

Voor mij voornamelijk om te zien of er voldoende expertise aanwezig is. Verder om te zien hoeveel projectleden er op een project zitten en welke kennis zij meebrengen. Ik zou verder graag zelf de mogelijkheid willen hebben om op mensen kunnen zoeken die bepaalde expertises bezitten. Op deze manier zou ik de juiste mensen voor mijn project kunnen vinden.

3. **Wat zou je idealiter met het kenniskaartensysteem bereiken?**

Dat het vanuit meerdere oogpunten te gebruiken is. Ik wil het bijvoorbeeld aan de klant kunnen tonen. Ik wil op expertisegebieden kunnen zoeken. Daarnaast wil ik een overview van welke expertise in mijn project aanwezig is. Ik wil ook een expertiseniveau kunnen escaleren, dus hogere eisen toekennen. Daarnaast wil ik het als "kaartenbak" kunnen gebruiken en zoeken naar mensen die de benodigde expertise bieden. Momenteel moet ik hiervoor mensen bellen om te vragen of ze werknemers kennen die de benodigde expertise bieden. Wanneer ik op expertise kan filteren kan ik in ieder geval gerichter zoeken.

4. **Wat moet het kenniskaartensysteem kunnen om dat doel te bereiken?**

Een dashboard dat ik aan de klant kan tonen. Een zoekfunctionaliteit waarmee ik kan filteren op verschillende criteria bij het zoeken naar een medewerker. Daarnaast wil ik graag dat de kenniskaart van een medewerker meegenomen kan worden naar verschillende projecten. Dat zou vanuit de medewerker gezien handig zijn.

5. **Geordend van hoge naar lage prioriteit: wat is prioriteit van deze functionaliteiten?**

Voor mijzelf is het voornamelijk belangrijk dat de expertises naar de klant toe aan te tonen zijn. Daarnaast is het voor mij van belang dat ik een overview heb. Als derde vind ik het zoeken naar medewerkers met bepaalde expertises belangrijk.

6. **Op welke manier moet het kenniskaartensysteem deze functionaliteiten bieden? (Hierbij kan aan snelheid of security worden gedacht)**

Snelheid is niet zo belangrijk aangezien de kenniskaarten niet gek veel worden gebruikt. Het is wel belangrijk dat het buiten het Sogeti-netwerk beschikbaar is. Verder is veiligheid en toegankelijkheid belangrijk.

E. Inception Report

Inception Report

Kenniskaartenapplicatie

Roy Straub
27/03/15

INHOUD

1 Inleiding 3

2 Strategy 4
- 2.1 Stakeholders 4
- 2.2 Users 4
- 2.3 Tasks 4
- 2.4 Persona's 5

3 Scope 7
- 3.1 Functionele requirements 7
- 3.2 Niet-functionele requirements 8
- 3.3 Content requirements 8
- 3.4 Scenario's 8

4 Structure 9
- 4.1 Interaction 9
- 4.2 Information 10

5 Skeleton 11
- 5.1 Wireframes 11

6 Architecture 13
- 6.1 Layer model 13
- 6.2 Component model 14
- 6.3 Domain Model 15

1 Inleiding

In dit document zal de inception fase van het project omtrent de kenniskaartenapplicatie bij Sogeti worden beschreven. Het betrof in deze fase voornamelijk het vastleggen van concepten op functioneel niveau. De structuur volgt voornamelijk de indeling op oppervlakken van het vijf lagenmodel van the Elements of User Experience. De Strategy-, Scope-, Structure- en Skeleton-laag zullen worden behandeld. Daarnaast zal een stuk functioneel ontwerp volgen in het hoofdstuk Architecture.

2 Strategy

Het hoofdstuk Strategy omvat het beschrijven van de stakeholders en gebruikers van het project. Bij de geïdentificeerde stakeholders worden de strategische doeleinden beschreven. Voor de vastgestelde Users zullen de User Needs worden toegelicht.

2.1 Stakeholders

Klanten Sogeti

De klanten van Sogeti willen inzichtelijk hebben of de juiste mensen met adequate kennis aan het project toegekend zijn. De kenniskaarten worden reeds gebruikt om dit inzichtelijk te maken. Het voornaamste voor deze stakeholders is dus het tastbaar hebben van de expertisedekking bij een project.

Sogeti als bedrijf

Met een kenniskaartensysteem kan het stukje kwaliteitsborging soepeler verlopen en makkelijker aantoonbaar worden gemaakt aan klanten. Daarbij heeft een hogere efficiëntie het bijkomende voordeel dat het manuren kan besparen wat op zijn beurt minder kosten kan betekenen.

Projectmanagers

Projecten kunnen eisen hebben wat betreft de kwaliteit van een project en het kunnen aantonen hiervan. De expertisedekking bij een project inzichtelijk maken is hier één van. Hiervoor wordt gebruik gemaakt van de kenniskaarten. Het kenniskaartensysteem is voor de projectmanagers een belangrijk gereedschap waarmee ze hun werk mogelijk efficiënter kunnen uitvoeren. Het zou daarnaast meer functies mogelijk maken.

2.2 Users

Projectmanagers

Een projectmanager wil te allen tijden kunnen zien of de bezetting bij een project wel of niet voldoet. Dit wil hij omdat hiermee gereageerd kan worden op verschuivingen in de bezetting die de verloop van een project kunnen beïnvloeden. Hij wil zien waar de tekorten en overschotten liggen in kennis. Daarnaast wil hij de kenniskaarten kunnen gebruiken om aan opgestelde kwaliteitseisen bij een project te voldoen. De expertisedekking wil hij eenvoudig aan de klant toonbaar kunnen maken.

Projectleden

Leden van projecten waar expertisedekking inzichtelijk gemaakt moet worden dienen de kenniskaart in te vullen. Hiermee maken zij inzichtelijk in welke mate zij bepaalde expertises beheersen. Een projectlid wil dan ook op een eenvoudige manier de expertises kunnen invullen die voor hem relevant zijn. Dit betekent dat het expertises zijn waarvoor hij bij projecten wordt ingezet. Hij of zij wil dit het liefst zo snel mogelijk doen en hier geen overbodige tijd aan verspillen.

Beheerders

De beheerder wil voornamelijk op een gemakkelijke en gebruiksvriendelijke manier de applicatie kunnen configureren. Hij dient de data in de applicatie te kunnen beheren. Hij wil hiervoor een eenvoudig te gebruiken interface hebben.

2.3 Tasks

Projectmanagers

De projectmanager voert de meeste taken uit in het kenniskaartenproces. Deze taken zullen dan ook voor een groot deel in de applicatie bij de projectmanager terecht komen. De taken die de projectmanager met de applicatie wil voltooien zijn de volgende:

- Kenniskaart opstellen voor een project
- Vereiste expertises vastleggen
- Projectleden koppelen aan de kenniskaart

- Skillgap toonbaar maken naar de klant

Projectleden

De projectleden hoeven weinig taken uit te voeren met de applicatie. Zij dienen enkel en alleen hun kennisniveaus te beheren. De taken die een projectlid uit dient te voeren zijn als volgt:

- Beheerste expertises kenbaar maken en bijhouden

Beheerders

Een beheerder heeft voornamelijk taken omtrent het beheren van data in de applicatie. De taken die de beheerder uitvoert zijn de volgende:

- Gebruikers van de applicatie beheren
- Kenniskaarten beheren

2.4 Persona's

De Persona's die hier geïntroduceerd zullen worden hebben als doel het bieden van een gebruikersperspectief. Door de Persona's op te stellen krijgen gebruikers van de applicatie context en een gezicht. Dit maakt het gemakkelijker om dit perspectief in gedachten te houden bij het maken van keuzes in het ontwerp- en ontwikkelproces van de applicatie.

Naam	**Harry de Vries**
Foto	
Leeftijd	38
Beroep	Projectmanager Business Line Java
Burg. Staat	Gehuwd
Kinderen	2
Hobby	Tv-series
Kleur	Rood
Auto	Volkswagen Polo Bluemotion
Kledingmerk	Boss
Tijdschrift	Top Gear
Telefoon	iPhone
PC	HP Elitebook 15"

Winkel	Apple
Vakantiebestemming	New York
Overige merken	Asus, Samsung
Overig	n.v.t.

Naam	**Henriëtte Janssen**
Foto	
Leeftijd	45
Beroep	Senior Microsoft Developer
Burg. Staat	Gehuwd
Kinderen	4
Hobby	Hardlopen
Kleur	Blauw
Auto	Peugeot 508sw
Kledingmerk	Polo
Tijdschrift	Linda
Telefoon	Galaxy S6
PC	Lenovo T520
Winkel	H&M
Vakantiebestemming	Egypte
Overige merken	Samsung
Overig	

3 Scope

In dit hoofdstuk wordt de stap van Strategy naar Scope gemaakt. Dit omvat de vertaling naar concretere productspecificaties. Allereerst zullen de functionele- en niet-functionele requirements worden gepresenteerd waarna de content requirements worden gegeven. Tot slot volgen enkele scenario's waarin de functionaliteiten in context worden geplaatst met gebruik van de opgestelde persona's.

3.1 Functionele requirements

Must have

- Inloggen met Sogeti gegevens
- Gebruikers toevoegen en rechten geven
- Kenniskaarten definiëren
- Benodigde expertises en expertiseniveaus vastleggen
- Medewerkers toevoegen
- Medewerker aan kenniskaart koppelen
- Medewerkerexpertise vastleggen
- Skillgapanalyse uitvoeren
- Read-Only rapport genereren

Should have

- Kenniskaarten delen
- Teams van medewerkers beheren
- Teams aan kenniskaarten koppelen
- Expertiseblokken beheren
- Expertiseblokken aan kenniskaarten koppelen
- Kenniskaarttemplate kopiëren

Could have

- Kenniskaarten hiërarchisch definiëren
- Expertises hiërarchisch definiëren
- Confluence plugin
- Pregedefinieerde expertises selecteren
- Wijzigingen in expertiseniveau werknemer laten doorwerken in andere kenniskaarten
- Manager verwijderen

Would have

- Expertises en medewerkers samenvoegen
- Expertiseniveaus medewerkers aanpassen en niveaucorrectie toepassen
- Expertises uit BIS importeren
- Expertiseniveaus uit BIS importeren

3.2 Niet-functionele requirements

De niet-functionele requirements bestaan voornamelijk uit Security en Concurrency. Het systeem moet voldoen aan dezelfde veiligheidsstandaarden als andere systemen binnen het CoE Java. Daarnaast moet het mogelijk zijn voor meerdere werknemers om tegelijk gebruik te maken van het systeem.

3.3 Content requirements

Expertisevereisten bij projecten

Deze informatie wordt door de projectmanager verstrekt. Dit wordt bij aanvang van het project vastgelegd of aangepast wanneer eisen aan het project veranderen. Het doel van deze content is om inzicht te bieden in de vereiste kennis van een project. Het format hiervan dient te bestaan uit vaardigheden, eventueel ondergebracht in categorieën. Elke vaardigheid dient verschillende niveaus van beheersing te tonen. Daarnaast dient het aantal werknemers per beheersingniveau vastgelegd te kunnen worden.

Expertises projectlid

Deze content wordt door de medewerker of projectmanager verstrekt. Ongeveer halfjaarlijks wordt deze informatie bijgewerkt. Het kan voorkomen dat deze content op initiatief van de medewerker wordt aangepast. Het doel van deze content is om inzicht te bieden in de kennis van een werknemer. Het format hiervan is vergelijkbaar aan dat van de criteria van projecten. Het betreft vaardigheden waarbij per vaardigheid een niveau van beheersing wordt gespecificeerd. De invoerdatum van deze informatie wordt bijgehouden.

3.4 Scenario's

"Harry"

Harry heeft zojuist te horen gekregen dat hij een groot nieuw project mag managen bij ING. Uit gesprekken met ING zijn de projectlijnen duidelijk. Het project betreft acht projectleden en het gaat om een product wat ING miljoenen kan besparen. Harry heeft ontzettend veel zin in deze nieuwe uitdaging. Aangezien hij zijn zaken op orde wilt hebben besluit Harry dat hij een overzicht moet hebben van de actieve projectleden in het project. Hij heeft te horen gekregen van zijn unitmanager dat hier kenniskaarten voor beschikbaar zijn. Nadat hij deze heeft bemachtigd begint hij met het toevoegen van een project en probeert hij zo accuraat mogelijk de vereiste kennisvelden in te voeren naar aanleiding van zijn gesprek met ING. Na het toevoegen van het project, wil Harry zijn bezetting vastleggen. Hij laat elk van zijn projectleden de kenniskaart invullen. Op het overzicht van het project zoekt hij naar de werknemers en voegt ze toe aan het project. Hij kan zien dat alle kennisvereisten worden gedekt door de toegekende, actieve projectleden.

"Henriette"

Henriette maakt sinds 2014 deel uit van een project bij de NS. Bij het project zijn vier andere projectleden betrokken. Haar manager heeft haar verteld over de verandering in de kwaliteitseisen van het project. Er moet aangetoond worden dat de juiste kennis bij het project aanwezig is. Haar manager praat haar bij over een zogenaamde "kenniskaart". Ze krijgt deze toegestuurd met de vraag of ze haar expertises hierin wilt vastleggen. Henriette vindt het allemaal omslachtig, maar vult het toch in. Haar manager verteld haar vervolgens dat hun projectgroep de benodigde kennis blijkt te bezitten. De klant laat weten zeer tevreden te zijn met de verloop van het project en laat een positieve reactie horen.

4 Structure

Het hoofdstuk Structure dient om de resultaten van de scope tastbaar te maken. De interacties met het systeem zullen hier duidelijk worden gemaakt in de sectie "Interaction". De Information Architecture wordt in de sectie "Information" besproken.

4.1 Interaction

Inloggen met Sogeti gegevens

Als medewerker wil ik wanneer ik de applicatie start een inlogscherm zien waar ik bij gebruikersnaam mijn Sogeti gebruikersnaam invul. Bij wachtwoord vul ik het bijbehorende wachtwoord in. Ik verwacht vervolgens een knop waarmee ik kan inloggen op mijn account met bijbehorende rechten.

Gebruikers toevoegen en rechten geven

Als beheerder wil ik gebruikers kunnen toevoegen aan het systeem. Ik wil hiervoor een eenvoudig te herkennen component. Wanneer ik een nieuwe gebruiker toevoeg wil ik hem een rol kunnen toekennen.

Kenniskaarten definiëren

Als manager wil ik dat ik op een eenvoudige, herkenbare manier een component vindt waarmee ik een nieuw project kan aanmaken. Wanneer ik deze component heb aangeklikt wil ik een naam aan het project kunnen geven. Ik wil dit bevestigen waarop het systeem het nieuwe project toegevoegd dient te hebben. Wanneer dit is gebeurd wil ik als gebruiker naar het overzichtscherm van het toegevoegde project worden gebracht.

Benodigde expertises en kennisniveau vastleggen

Als manager wil ik wanneer bij een kenniskaart de mogelijkheid hebben om vereiste expertises toe te voegen. Ik wil hiervoor een eenvoudig te herkennen component hebben. Wanneer ik deze selecteer wil ik de mogelijkheid hebben om uit bekende expertises te kiezen. Als ik een expertise heb geselecteerd wil ik het aantal personen per kennisniveau vast kunnen leggen en dit bevestigen. Ik verwacht vervolgens dat deze expertisevereisten in het overzicht van de kenniskaart terug zijn te vinden.

Medewerker toevoegen

Als manager wil ik dat ik op een eenvoudige, herkenbare manier een component vind waarmee ik een nieuw projectlid kan aanmaken in het systeem. Ik wil hierbij de naam van de werknemer kunnen vastleggen. Daarnaast wil ik een functiebeschrijving vastleggen. Ik verwacht na het toevoegen van het projectlid dat hij te zien is in het medewerkeroverzicht.

Medewerker aan kenniskaart koppelen

Als manager wil ik reeds toegevoegde medewerkers kunnen koppelen aan een project. Ik wil een eenvoudig herkenbare knop zien waarmee ik een medewerker kan selecteren. Vervolgens wil ik deze medewerker als projectlid kunnen toevoegen aan een project. Ik verwacht deze projectleden te kunnen zien bij het overzicht van de kenniskaart.

Medewerkerexpertise vastleggen

Als medewerker wil ik mijn expertiseniveau vast kunnen leggen. Ik wil hiervoor gemakkelijk voor mij relevante expertises kunnen selecteren en toevoegen. Vervolgens wil ik dat de expertiseniveaus gemakkelijk in te vullen zijn waarna ik ze kan opslaan. Ik verwacht deze bij het overzicht van mijn profiel te zien.

Skillgapanalyse uitvoeren

Als manager wil ik wanneer ik een kenniskaart heb geselecteerd de skillgap zien. Wanneer de expertises zijn ingevuld en er gekoppelde werknemers zijn met een ingevuld verwacht ik dit. Ik wil per expertise zien of de aanwezige kennis wel of niet toereikend is. Verder wil ik kunnen zien hoeveel personen per niveau er te veel of te kort is. Dit moet te allen tijde up to date zijn.

4.2 Information

Expertisevereisten project

Bij een kenniskaart moeten de vereiste expertises zichtbaar zijn. Bij elke expertise moeten de verschillende niveaus te zien zijn. Per niveau dient het aantal vereiste personen te zien te zijn. Deze expertises zouden eventueel gekoppeld kunnen zijn aan categorieën waarmee doorzoeken gemakkelijker wordt.

Expertises projectlid

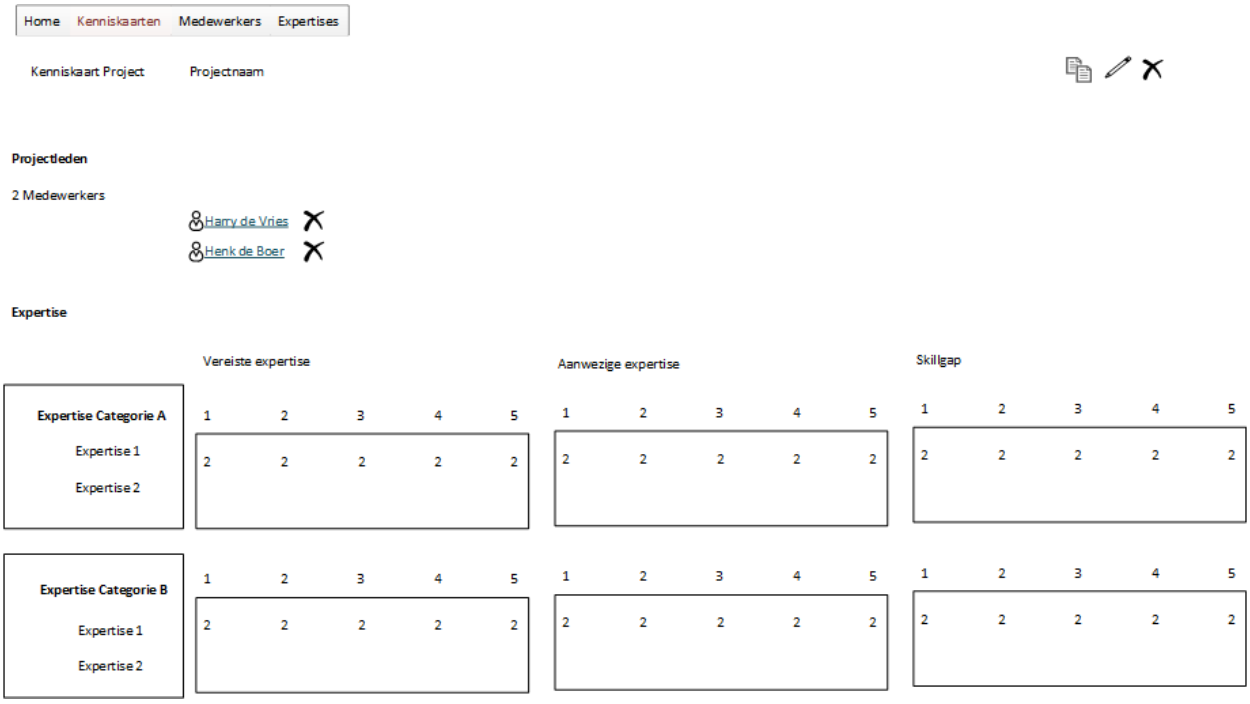
Bij elk projectlid in het systeem dienen zijn gekoppelde expertises inzichtelijk te zijn. Alle gekoppelde expertises representeren een vaardigheid waar de werknemer een niveau van beheersing in heeft. Dit niveau van beheersing dient bij elke vaardigheid te worden vastgelegd. De niveaus rangeren van een zeer lage vorm van beheersing tot een extreem hoge vorm van beheersing.

5 Skeleton

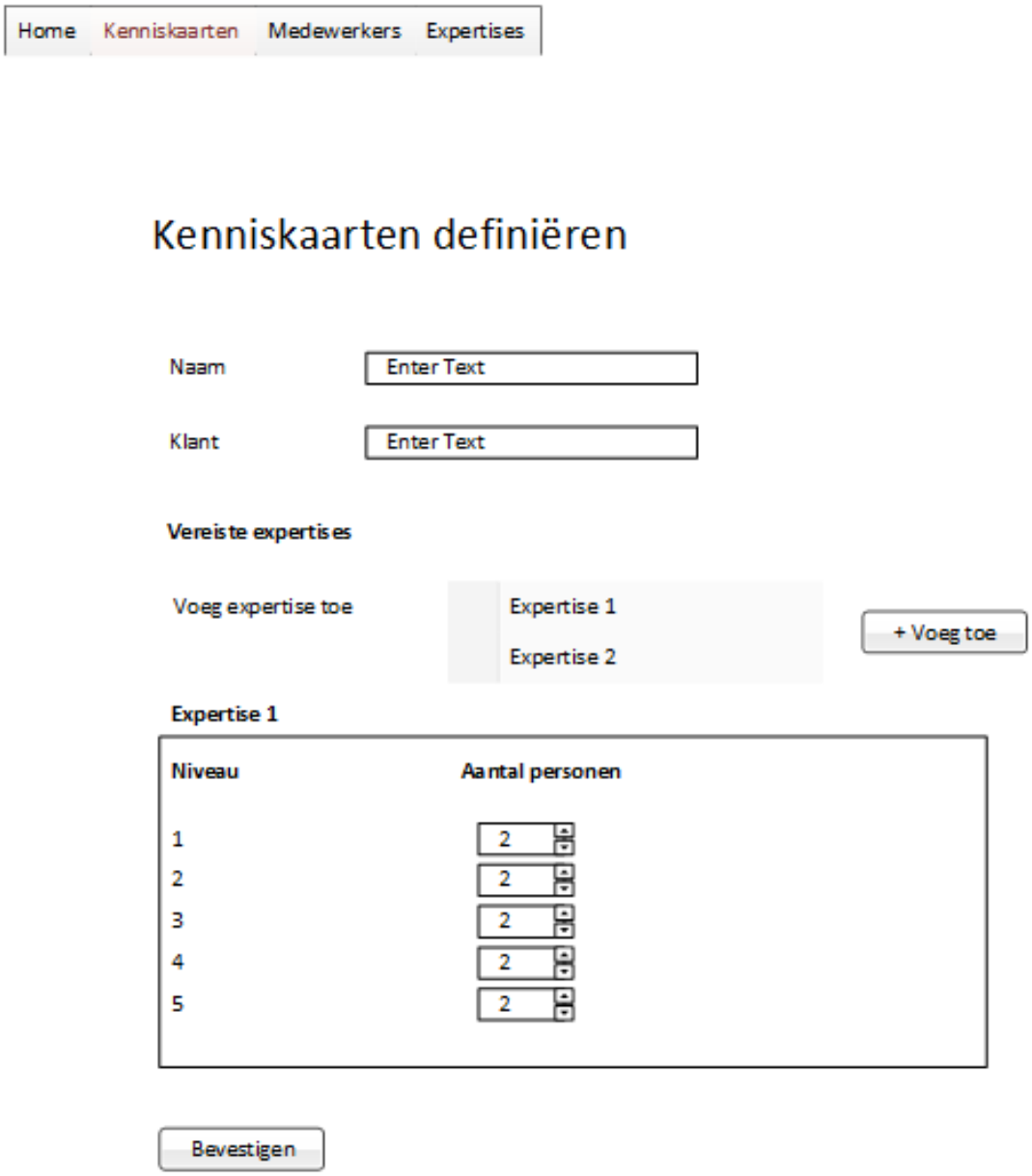
In het hoofdstuk Skeleton wordt de stap van conceptuele vormen en ideeën naar een tastbare representatie gemaakt. Hiervoor zijn Wireframes opgesteld voor de drie voornaamste views: de overzichtpagina, de detailpagina en de aanmaakpagina. Deze pagina's keren bij elke entiteit in het systeem terug. Het betreft in dit geval de meest uitgebreide entiteit: de kenniskaart.

5.1 Wireframes

Figuur 1. Overzicht kenniskaarten



Figuur 2. Detail kenniskaart



Figuur 3. Kenniskaart aanmaken

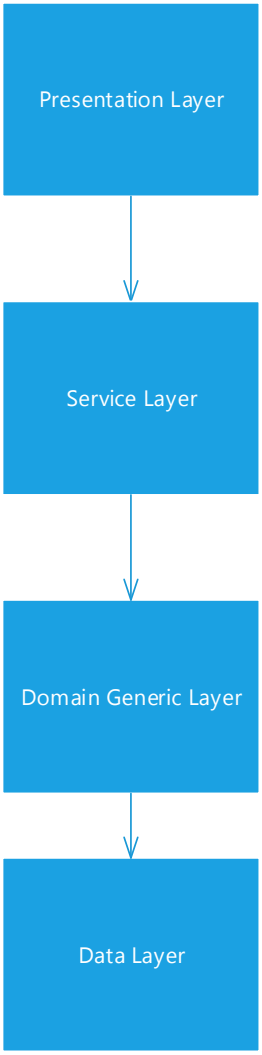
6 Architecture

Het hoofdstuk Architecture dient om de hoofdlijnen van de architectuur vast te leggen. Dit zal dienen als een conceptueel houvast, een "outline" voor de architectuur gedurende de verloop van het project. Er wordt allereerst een lagenmodel gepresenteerd waarna de componenten van het systeem in een Component Model worden weergegeven. Tot slot volgt een functioneel Domain Model.

6.1 Layer model

In dit model bestaat de Presentation Layer uit de UI en de task specific logic. De Service Layer bestaat uit alles wat betrokken is bij het beschikbaar maken van de logica als web service. Daarop volgend bestaat de Domain Generic Layer uit business logica en het vastgestelde domein. Tot slot bestaat de datalaag uit persistentielogica en operaties.

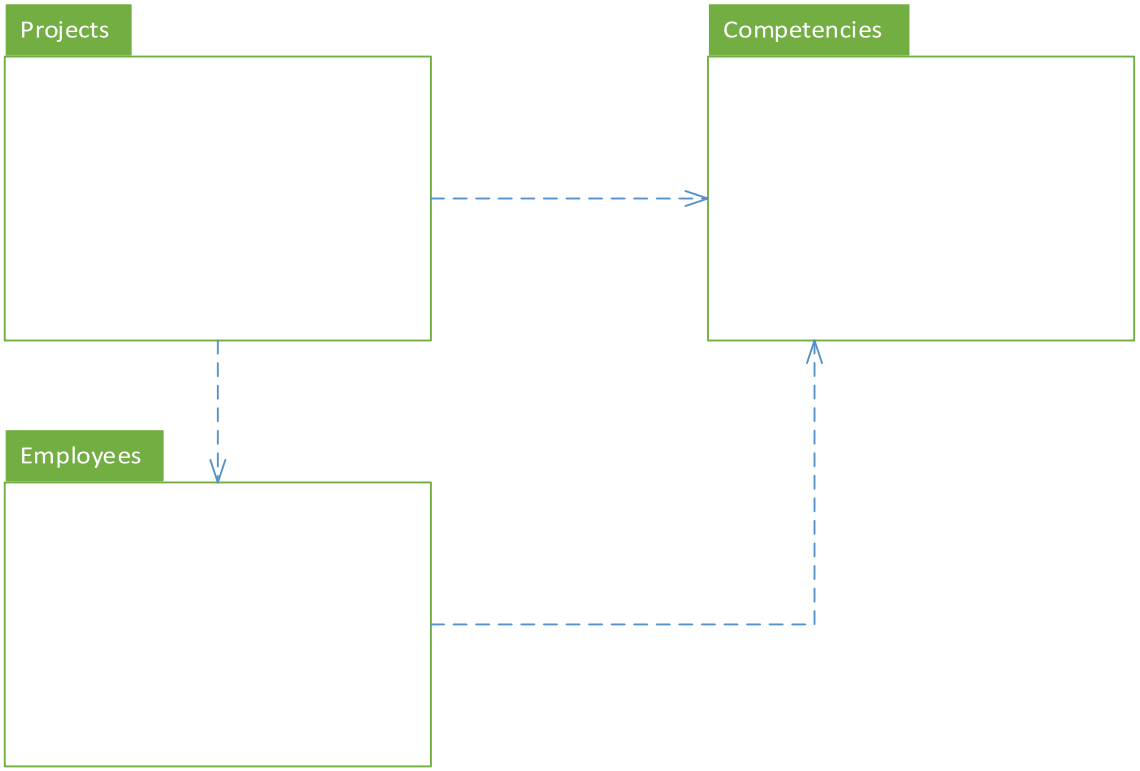
De communicatieregels voor het toepassen van dit model zijn als volgt. Alle operaties die een waarde terug verwachten dienen slechts naar beneden te mogen verlopen. Er dienen geen lagen overgeslagen te worden ten behoeve van validatie. Tot slot dient de elke laag zo min mogelijk van de lagen erboven af te weten.



Figuur 4. Layer Model

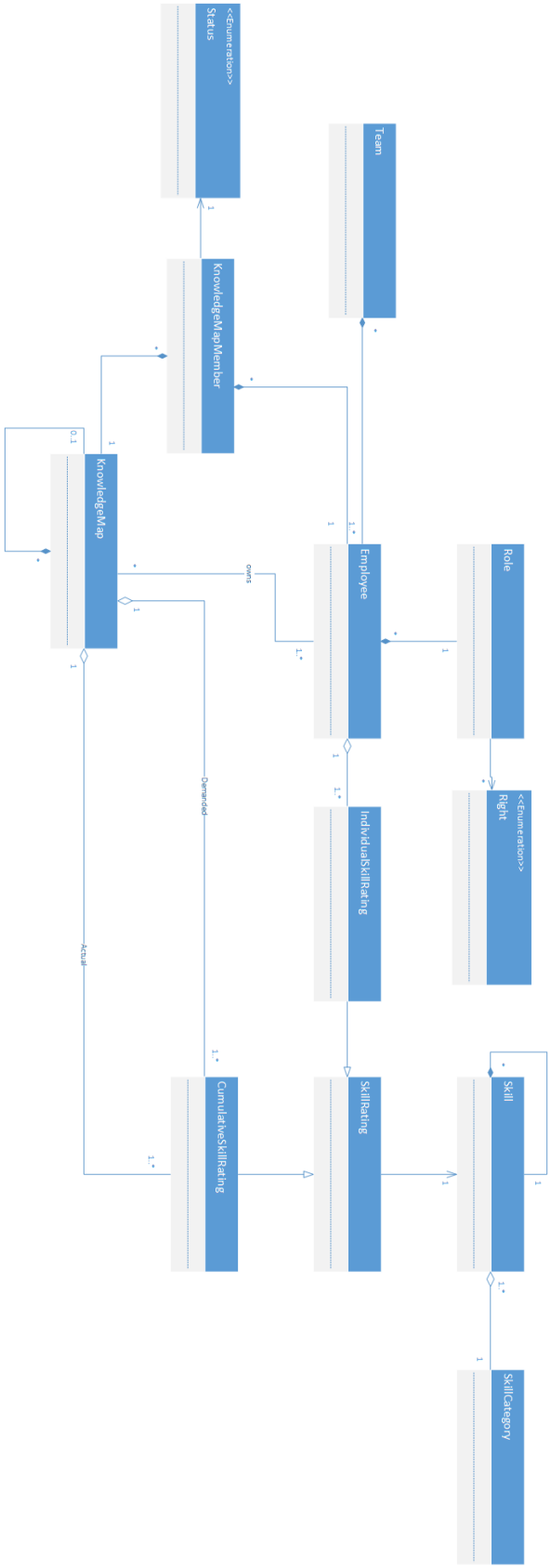
6.2 Component model

De componenten staan voor grote onafhankelijke onderdelen in het systeem. De gedetecteerde componenten in zijn weergegeven in figuur 5.



Figuur 5. Component model

6.3 Domain Model



Roy Straub
27/03/15

F. Elaboration Report

Elaboration Report

Kenniskaartenapplicatie

Roy Straub
06/04/15

INHOUD

1 Inleiding 3

2 Domein Model 4

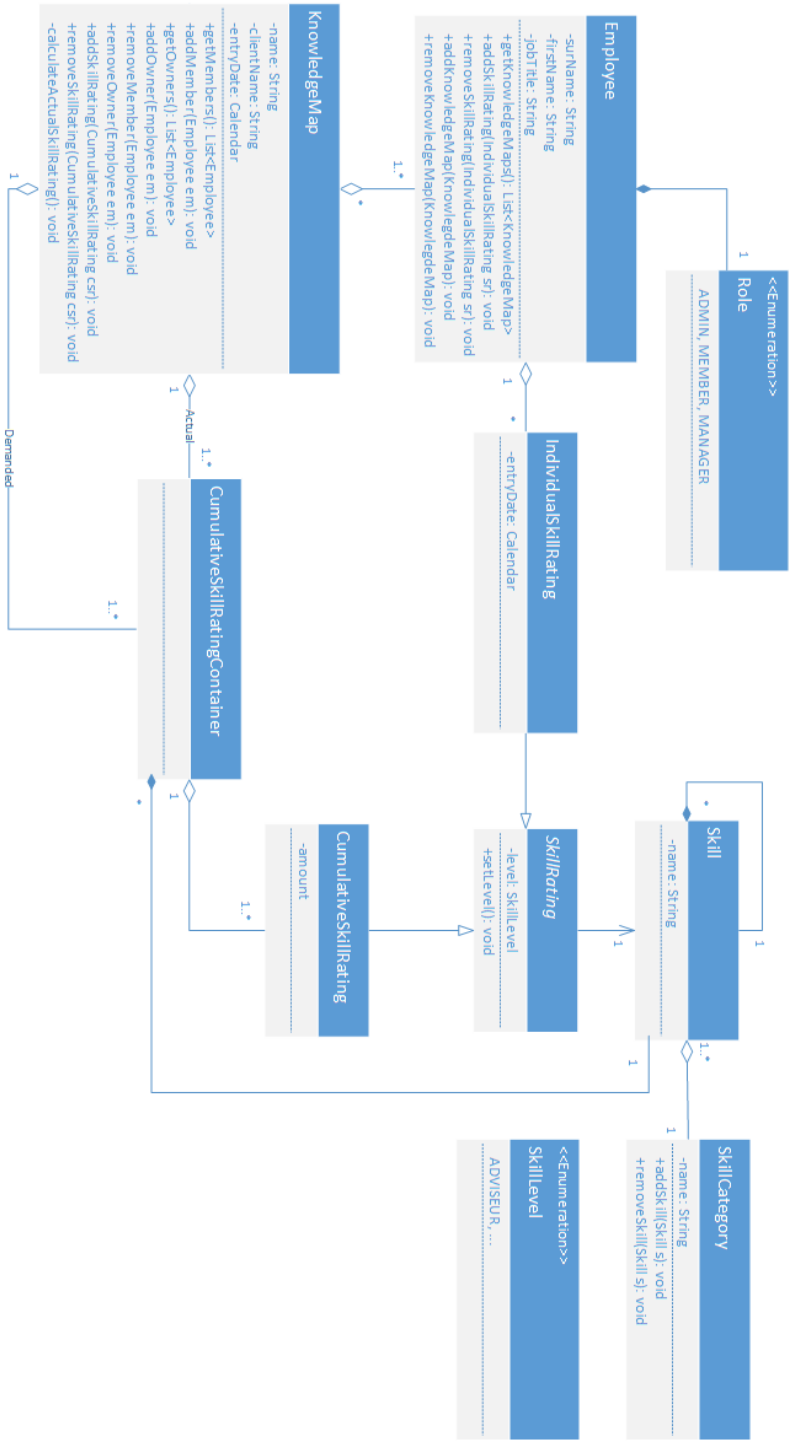
3 Implementatiemodel 5

1 Inleiding

In dit document zal de Elaboration fase van het project omtrent de kenniskaartenapplicatie bij Sogeti worden vastgelegd. Het betreft hierin voornamelijk het vastleggen van concepten op technisch niveau. Allereerst zal een technisch Domain Model worden gepresenteerd. Vervolgens zal een Implementation Model worden getoond. De keuzes in dit rapport zijn gebaseerd op eerder gemaakte keuzes in de Inception fase.

2 Domein Model

Dit model bouwt voort op het model uit de Inception fase. Het technische niveau zal onder andere belangrijke members van de klassen weergeven. Daarnaast worden relaties geëxpliciteerd. Dit model is aan de hand van de time-boxes en hun work items bijgewerkt.



Figuur 1. Domain Model

3 Implementatiemodel

Hier zal het implementatiemodel worden getoond. Het implementatiemodel zal de klassen bevatten die niet in het domein vallen, maar wel onder de business logica van de applicatie. Hiermee zal de werking van de applicatie verhelderd worden. Dit is incrementeel bijgewerkt, afhankelijk van de work items in de time-boxes tijdens de constructiefase.

Figuur 2. Implementation Model

G. Construction Report

Construction Report

Kenniskaartenapplicatie

Roy Straub
13/04/15

INHOUD

1 Inleiding 3

2 Timebox 1 4
2.1 Work items 4
2.2 User Tests 5

3 Timebox 2 7
3.1 Work items 7
3.2 User Tests 7
3.3 SonarQube Code Quality Analysis 8

1 Inleiding

In dit document zal de Construction fase van het project omtrent de kenniskaartenapplicatie bij Sogeti worden vastgelegd. Hierin worden de timeboxes beschreven met hun bijbehorende work items. Elk van de work items zal worden toegelicht. Daarnaast zijn de User Tests en code quality analyse in dit document beschreven.

2 Timebox 1

2.1 Work items

1. **Configuratie.**

Het project is opgezet in eclipse met gebruik van een sample project. Vervolgens is het sample project aangepast aan de huidige context.

2. **Overzicht kenniskaarten**

De functies voor het opvragen van kenniskaarten in het systeem zijn geïmplementeerd. Bij de cliëntapplicatie is de overzichtpagina gerealiseerd .

3. **Kenniskaart toevoegen**

In de backend is de functionaliteit om nieuwe kenniskaarten aan te maken toegevoegd. Vervolgens is een aanmaakpagina gecreëerd in de cliëntapplicatie.

4. **Kenniskaart detail**

Functionaliteit voor het opvragen van de data van een enkele kenniskaart is toegevoegd aan het systeem. In de cliëntapplicatie is een detailpagina toegevoegd waarin alle data van een kenniskaart te zien is.

5. **Kenniskaart verwijderen**

Het verwijderen van een kenniskaart aan de hand van zijn id is mogelijk gemaakt. Verder kan deze functie worden aangeroepen vanaf de detailpagina van een kenniskaart.

6. **Kenniskaart bewerken**

Een bestaande kenniskaart kan worden aangepast. Hierbij kan data van een kenniskaart worden bewerkt. Deze functie is in de frontend te gebruiken gemaakt.

7. **Overzicht medewerkers**

De medewerkers in het systeem zijn op te vragen. Een overzichtpagina is opgesteld.

8. **Medewerker toevoegen**

Medewerkers kunnen worden aangemaakt. Er is in de cliëntapplicatie een pagina gemaakt waarmee nieuwe medewerkers kunnen worden aangemaakt.

9. **Medewerker detail**

De data van een enkele medewerker is opvraagbaar. Daarnaast kan dit in de cliëntapplicatie worden getoond.

10. **Medewerker verwijderen**

Een medewerker kan worden verwijderd. In de cliëntpagina is een deze functie aanroepbaar gemaakt.

11. **Medewerker bewerken**

De gegevens van een medewerker kunnen worden bewerkt. Deze wijzigingen worden in de backend verwerkt.

12. **Overzicht expertises**

Alle expertises in het systeem kunnen worden opgehaald. In de cliëntapplicatie is het mogelijk gemaakt om dit overzicht op een overzichtpagina te tonen.

13. **Expertise toevoegen**

Nieuwe expertises kunnen aan het systeem worden toegevoegd. De cliënt is uitgebreid met een pagina waar een nieuwe expertise kan worden aangemaakt.

14. **Expertise verwijderen**

Een bestaande expertise kan uit het systeem worden verwijderd. Deze functionaliteit is in de cliënt bereikbaar gemaakt.

15. **Expertise bewerken**

Bestaande expertises kunnen worden aangepast in de backend. De frontend is uitgebreid met deze functie aanroepen.

16. **Expertises aan kenniskaarten toevoegen**

De cliëntapplicatie is uitgebreid met de mogelijkheid om bestaande expertises in het systeem aan kenniskaarten te kunnen koppelen. Hiermee worden vereiste expertises, hun niveaus en het aantal benodigde medewerkers per niveau vastgelegd.

17. Unit testing domein timebox 1
Het testen van de logica in de domeinlaag betreffende de user stories uit timebox 1.

2.2 User Tests

De functionaliteiten uit de eerste timebox zijn geïmplementeerd in een webapplicatie. Deze applicatie is voorgelegd bij personen die nog niet bekend zijn met de applicatie. Vervolgens is deze mensen gevraagd enkele handelingen met de applicatie uit te voeren. Er werd voornamelijk gekeken of interacties te vinden waren en logisch in elkaar zaten. Verder werd aan de subjecten gevraagd om hun ervaringen gedurende de handelingen te verwoorden.

Het testscript

De handelingen die uitgevoerd moesten worden door de subjecten volgden elkaar zoveel mogelijk op. Het beginpunt van het script was het overzicht van kenniskaarten. Het subject moest vervolgens plaatsnemen achter de laptop. Wanneer het subject klaar was werd begonnen met het stellen van een paar vragen en moesten de subjecten de interacties uitvoeren. De vragen die aan de subjecten werden gesteld waren als volgt:

1. *Wat valt je als eerste op in dit scherm?*
2. *Wat moet je doen om een kenniskaart toe te voegen?*
3. *Hoe kan je de details van de zojuist toegevoegde kenniskaart zien?*
4. *Hoe zou je de gegevens van deze kenniskaart kunnen aanpassen?*
5. *Op welke manier zou je deze kenniskaart kunnen verwijderen?*

Resultaten user tests

Test 1 – Tiffany Straub (07-05-15)

Wat valt je als eerste op in dit scherm?
"Vooral de groene knop waar maak aan op staat. Verder valt het zoekveld me op. Ik vind het overzichtelijk en kan zien wat waar voor staat." (Uit toelichting bleek hiermee bedoeld te worden dat het duidelijk was welke waardes voor welke eigenschap staat.)

Wat moet je doen om een kenniskaart toe te voegen?
"Ik verwacht dat ik op de groene knop moet drukken." Het subject vond het niet duidelijk dat op het aanmaakscherm gedrukt kon worden op een expertise om hier expertiseniveaus en hun vereisten vast te leggen.

Hoe kan je de details van de zojuist toegevoegde kenniskaart zien?
"Ik verwacht op de naam van de kenniskaart te moeten klikken, deze is blauwgekleurd zoals een link."

Hoe zou je de gegevens van deze kenniskaart kunnen aanpassen?
"Daarvoor moet ik op de gele knop drukken waar bewerken op staat." Tijdens het uitvoeren verliep alles goed. Het subject lukt het in één keer goed te doen.

Op welke manier zou je deze kenniskaart kunnen verwijderen?
"Ik moet daarvoor de op rode knop waar verwijderen op staat klikken." Ook dit werd foutloos uitgevoerd.

Test 2 – Karin Wong (12-05-2015)

Wat valt je als eerste op in dit scherm?
De titel, omdat deze groter is dan de rest. Verder viel de knop en het logo op vanwege de kleur.

Wat moet je doen om een kenniskaart toe te voegen?
"Drukken op de knop maak aan." Het aanmaken was volgens het subject een logisch proces. Het was niet 100% duidelijk waar geklikt moest worden in de accordion die voor de expertises wordt gebruikt.

Hoe kan je de details van de zojuist toegevoegde kenniskaart zien?

Deze stap voerde het subject bij voorbaat al uit. Ze vond dit logisch omdat de naam blauw was weergegeven. Dit was uit ervaring een logische associatie met een link.

Hoe zou je de gegevens van deze kenniskaart kunnen aanpassen?
"Ik moet op de knop bewerken klikken." Het subject vond dit een logische stap. Ze voerde dit correct uit zonder aanwijzingen.

Op welke manier zou je deze kenniskaart kunnen verwijderen?
"De knop verwijderen aanklikken". Wederom waren geen aanwijzingen vereist en kon het subject dit correct uitvoeren.

Aanpassingen
Naar aanleiding van de eerste user test is er een aanpassing gedaan bij het aanmaakscherm van kenniskaarten. Hier is het duidelijker gemaakt dat de expertisenaam aangeklikt kan worden.

3 Timebox 2

3.1 Work items

1. Medewerkersexpertise invullen.
Expertises kunnen aan medewerkers worden gekoppeld. Bij elke expertise kan een beheersingsniveau worden vastgelegd. De aanmaakpagina voor medewerkers is uitgebreid om deze koppeling in de webapplicatie mogelijk te maken.
2. Expertise medewerkers bijwerken
De expertise van een medewerker kan aangepast worden. In het bewerkscherm van een medewerker kan het niveau bij een expertise worden aangepast. Daarnaast is het mogelijk gemaakt om expertises te kunnen ontkoppelen. Bij een bestaande medewerker kunnen expertises worden gekoppeld die nog niet in zijn bekende vaardigheden staan.
3. Medewerkers koppelen aan kenniskaart.
Het is in de cliënt mogelijk gemaakt om bestaande medewerkers in het systeem te koppelen aan een kenniskaart. Deze gekoppelde medewerkers zorgen voor de aangeleverde kennis bij een project.
4. Medewerkers bij een kenniskaart aanpassen
Bij een bestaande kenniskaart kunnen de gekoppelde medewerkers worden aangepast. Nieuwe medewerkers kunnen worden gekoppeld en reeds gekoppelde medewerkers kunnen worden ontkoppeld.
5. Algoritme berekenen cumulatieve expertises medewerkers
Het algoritme kan berekenen welke expertises een groep medewerkers bezit, met daarbij de niveaus en hoeveelheid personen die het niveau beheersen. Hiermee kan de aanwezige kennis bij een kenniskaart worden berekend.
6. Algoritme berekenen expertisedekking kenniskaart
Dit algoritme berekent aan de hand van de vereiste kennis en de aanwezige kennis hoeveel personen met een specifiek niveau een kenniskaart netto per expertise heeft. Dit wordt ook wel de skillgap genoemd.

3.2 User Tests

Het testscript
Voor de tweede set user tests is het testscript meer gefocused op het koppelen van de entiteiten uit de eerste timebox. Daarnaast zal de skillgapanalyse worden voorgelegd. Dit keer was in de uitgangssituatie dan ook reeds een set medewerkers, expertises en een kenniskaart beschikbaar. Het script bestaat wederom uit een verzameling vragen die aan de subjecten worden gesteld. Vervolgens dienden ze een handeling uit te voeren. De vragen voor de tweede set user tests waren als volgt:

1. *Hoe moet je een vaardigheid koppelen aan een bestaande medewerker?*
2. *Op welke manier zou je het niveau van de toegevoegde vaardigheid kunnen verhogen?*
3. *Wat moet je doen om de toegevoegde vaardigheid te ontkoppelen van de medewerker?*
4. *Hoe kan je een medewerker koppelen aan een kenniskaart?*
5. *Wat zou er moeten gebeuren om een medewerker te kunnen ontkoppelen van een kenniskaart?*
6. *Is er genoeg kennis bij alle expertises aanwezig?*
7. *Wat zie je op het scherm en welke conclusies trek je aan de hand daarvan?*

Resultaten user tests
Test 1 – Jarno Bats (29-05-15)

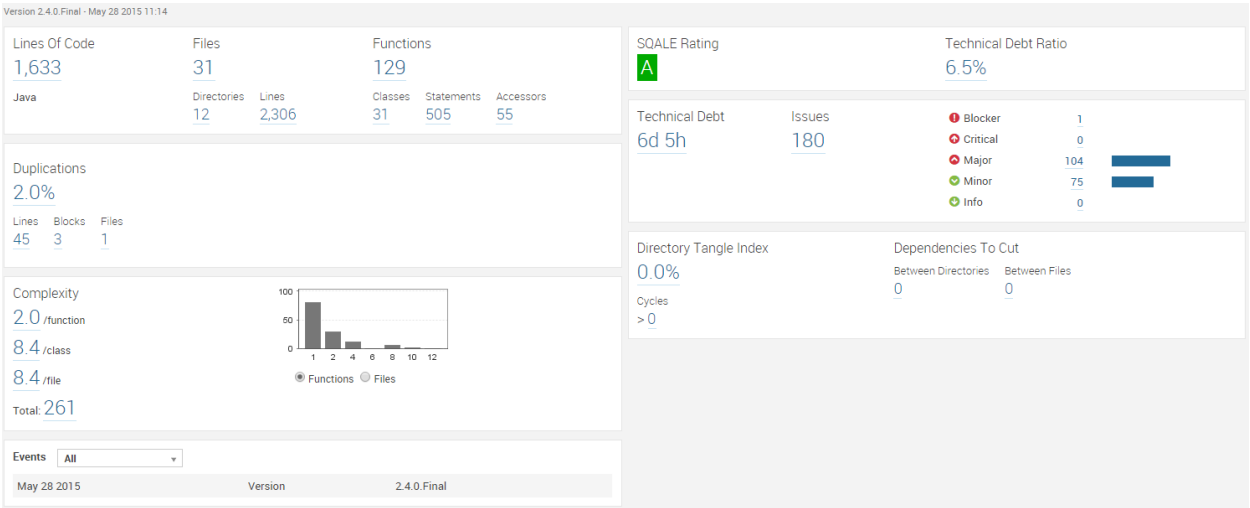
Hoe moet je een vaardigheid koppelen aan een bestaande medewerker?
De subject klikte vrijwel gelijk op de naam van een medewerker en op de knop bewerken. Vervolgens had hij de expertisesectie gevonden en selecteerde foutloos een expertise. Hij zag de control voor het selecteren van een niveau verschijnen en voerde dit correct in. “Misschien dat het selecteren van het niveau iets meer aanwezig kan zijn”.

Op welke manier zou je het niveau van de toegevoegde vaardigheid kunnen verhogen?
De subject klikte op bewerken bij de detailpagina van een medewerker. Hierna selecteerde hij al snel de vaardigheid die hij wilde aanpassen en verhoogde dit zonder complicaties.
Wat moet je doen om de toegevoegde vaardigheid te ontkoppelen van de medewerker?
Wederom verliep dit foutloos. Het subject drukte op bewerken en verwijderde de vaardigheid zonder problemen.
Hoe kan je een medewerker koppelen aan een kenniskaart?
Het subject dacht dat vanaf de medewerkerpagina te kunnen bereiken. Na enige uitleg vond hij de stap vanuit het detail van de kenniskaart logisch en voerde hij het koppelen goed uit.
Wat zou er moeten gebeuren om een medewerker te kunnen ontkoppelen van een kenniskaart?
Geen complicaties, dit verliep in één keer goed.
Is er genoeg kennis bij alle expertises aanwezig?
De subject moest voornamelijk kijken wat alles precies inhield. De kleuren hielpen hem bij het onderscheiden van positief en negatief en trokken zijn aandacht. Hij kon zien dat er tekorten en overschotten waren op sommige niveaus.
Wat zie je op het scherm en welke conclusies trek je aan de hand daarvan?
Het subject vond de hoeveelheid informatie een beetje verwarrend aangezien hij geen voorkennis had. Na enige toelichting was dit verholpen.

Aanpassingen
De terugknop in de detailviews moeten naar de vorige pagina verwijzen, niet het overzicht. De heading van de accordion helemaal klikbaar maken en het kennisniveau selecteren prominenter aanwezig maken.

3.3 SonarQube Code Quality Analysis

De code quality van de applicatie is getest met behulp van SonarQube. Na het analyseren is het mogelijk een dashboard raad te plegen waarin de resultaten zijn in te zien. Hier zijn onder andere complexiteit, regels code, duplicatie, technical debt en andere metrieken inzichtbaar per module in de applicatie. Met behulp van deze metrieken wordt een SQALE rating verbonden aan de applicatie die loopt van E (het slechts) tot A (het beste). Het resultaat van de SonarQube-analyse is te zien in figuur 1.



Figuur 1. SonarQube-analyse

De resultaten laten zien dat de applicatie de beste SQALE-rating heeft die mogelijk is. Verder laat het zien dat er twee procent code duplicatie is. Daarnaast zijn enkele issues in de applicatie opgespoord die de technical debt kunnen verhogen. Het oplossen van deze issues zou leiden tot een beter onderhoudbaar en uitbreidbaar geheel.