

DE ONTWIKKELING VAN DE “HMI-VERTAALTOOL”

*Een verbetering van het deelproces, HMI vertalen, binnen de afdeling
software engineering van Houdijk Holland.*

Eindverslag

HOUDIJK HOLLAND

B.F.J. de Jong
4 juni 2021

De ontwikkeling van de “HMI-vertaaltool”

Een verbetering van het deelproces, HMI vertalen, binnen de afdeling software engineering van Houdijk Holland.

Auteur: B.F.J. de Jong
De Haagse Hogeschool, Netwerk and System Engineering
4 juni 2021

Begeleidend examiner: R.A. van Neijhof
Expert examiner: R. Hamelink
Bedrijfsmentor: L. Vlasman

Houdijk Holland B.V.
Vlaardingen

Referaat

Jong, B.F.J. de, Eindverslag afstuderen, HMI-vertaaltool, Houdijk Holland: Vlaardingen 2021

Dit is het eindverslag ter afronding van de opleiding Networks and System Engineering aan de Haagse Hogeschool. De opdracht is uitgevoerd voor biscuitmachine fabrikant Houdijk Holland in Vlaardingen.

Binnen de stage is een HMI-vertaaltool met user interface gerealiseerd, waarmee medewerkers van Houdijk Holland een HMI in andere (natuurlijke) taal beschikbaar kunnen stellen. Hierbij is het mogelijk om als software engineer tijdens het vertalen aanpassingen aan de HMI te maken en voorkomt de tool dat reeds vertaalde teksten opnieuw naar de vertaler worden gestuurd.

DESCRIPTOREN

- FactoryTalk View
- Human Machine Interface (HMI)
- C#
- MySQL
- Window Presentation Forms (WPF)
- Model, View and ViewModel (MVVM)
- Unified Modeling Language (UML)

Gegevens

STUDENT

Naam: Britney de Jong
Email:
Studentnummer:

OPLEIDING

Onderwijsinstelling: De Haagse Hogeschool
Opleiding: Networks and System Engineering
Adres: Rotterdamseweg 137
2628 AL Delft
Telefoonnummer: 015 260 6200

BEGELEIDEND EXAMINATOR

Naam: R.A. van Neijhof
Email:

EXPERT EXAMINATOR

Naam: R. Hamelink
Email:

ORGANISATIE

Bedrijfsnaam: Houdijk Holland
Adres: Trawlerweg 2-4,
3133 KS Vlaardingen
Telefoonnummer:

BEDRIJFSMENTOR L. Vlasman
Naam:

OPDRACHTGEVER R. Clarisse
Naam:

Voorwoord

Voor u ligt het eindverslag van mijn afstudeerstage bij Houdijk Holland in Vlaardingen. Dit eindverslag beschrijft het proces van de ontwikkeling van de “HMI-vertaaltool” een verbetering van het deelproces, HMI vertalen, binnen de software engineering afdeling. Dit eindverslag is onderdeel van het afstudeerdossier ter afronding van de opleiding Network and System Engineering aan De Haagse Hogeschool.

Met plezier heb ik de afgelopen maanden bij Houdijk Holland stage gelopen. Tijdens de stage heb ik veel nieuwe mensen leren kennen en kunnen werken met (voor mij) nieuwe technologieën. Naast alle collega's bij Houdijk Holland zou ik graag Lara Vlasman expliciet willen bedanken voor het begeleiden van mijn opdracht en het altijd voor me klaar staan. Daarnaast wil ik Ruud Hogedoorn en Robin Clarisse bedanken voor het geven van hun mening over de user interface en de code kwaliteit.

Ik wil iedereen bedanken, die me tijdens deze stageperiode heeft geholpen. Met name mijn vader en mijn vriend voor het nalezen van het eindverslag.

Britney de Jong

Vlaardingen, juni 2021

Inhoudsopgave

1.	Inleiding.....	1
2.	Houdijk Holland	3
2.1	Organisatie	3
2.2	De werkomgeving.....	3
3.	Opdrachtschrijving	5
3.1	Probleemstelling.....	5
3.2	Afbakening.....	5
3.3	Doelstelling.....	5
3.4	Producten	6
4.	Aanpak	7
4.1	Methodologie	7
4.2	Ontwikkelmethode.....	9
4.3	Programmatuur	10
4.4	Risico's en maatregelen.....	11
5.	Huidige situatie	13
6.	Analyse van technologieën	15
6.1	Van onderzoek naar analyse	15
6.2	Gekozen technologieën.....	15
7.	Requirements analysis.....	17
7.1	Formuleren user stories	18
7.2	Verzamelen, groeperen en prioriteren	20
7.3	Eisen verdelen per iteratie	22
8.	User design	23
8.1	Iteratie 1 MVVM.....	24
8.2	Iteratie 2 format	29
8.3	Iteratie 3 lay-out.....	31
8.4	Iteratie 4 progressie balk.....	33
8.5	Iteratie 5 database	34
8.6	Iteratie 6 onderhoud	35
9.	Rapid construction	36
9.1	Refactoren	36
9.2	Installatie	37
10.	Cutover.....	38
10.1	Acceptatietest	38

10.2	Gebruikershandleiding	40
10.3	BPMN model nieuwe situatie	40
11.	Resultaat eindproduct	41
12.	Beroepstaken	43
13.	Product evaluatie	44
14.	Proces evaluatie	44
15.	Reflectie.....	47
	Lijst met afkortingen	48
	Literatuur.....	49
	Bijlagen.....	51

1. Inleiding

Al sinds de start van het gebruik van FactoryTalk View bij Houdijk Holland verloopt het vertalen van de Human Machine Interfaces (HMI) met veel problemen en irritaties. In januari 2016 leek hier verandering in te komen door de interne ontwikkeling van een tool in Microsoft Excel. Echter kostte deze oplossing onnodig veel geld door redundante vertalingen en was het vertalen van een HMI tijdrovend door de inflexibiliteit van FactoryTalk View. Door de ontwikkeling van een nieuwe tool, de “HMI-vertaaltool”, zijn deze problemen grotendeels verholpen. Middels deze vertaaltool kunnen medewerkers van Houdijk Holland eenvoudig een HMI vertalen.

Houdijk Holland fabriceert biscuitmachines voor grote biscuitfabrikanten. De meeste afgeleverde machines bevatten een HMI, die vanuit het Engels vertaald moet worden naar de taal van de klant.

De HMI-vertaaltool is een Windows Presentations Forms (WPF) applicatie, die gebruik maakt van de View, Model, ViewModel (MVVM) ontwerppatroon. Aan de applicatie zit een MySQL database, waar alle vertalingen worden opgeslagen. Voor de ontwikkeling is gebruik gemaakt van de Rapid Application Development (RAD) methode.

Deze stage is het laatste onderdeel van de opleiding Network and System Engineering aan De Haagse Hogeschool. De opdracht wordt uitgevoerd onder begeleiding van bedrijfsmentor L. Vlasman. De afstudeerstage wordt beoordeeld door expert examiner R. Hamelink en begeleidend examiner R.A van Neijhof.

STRUCTUUR BINNEN HET DOCUMENT

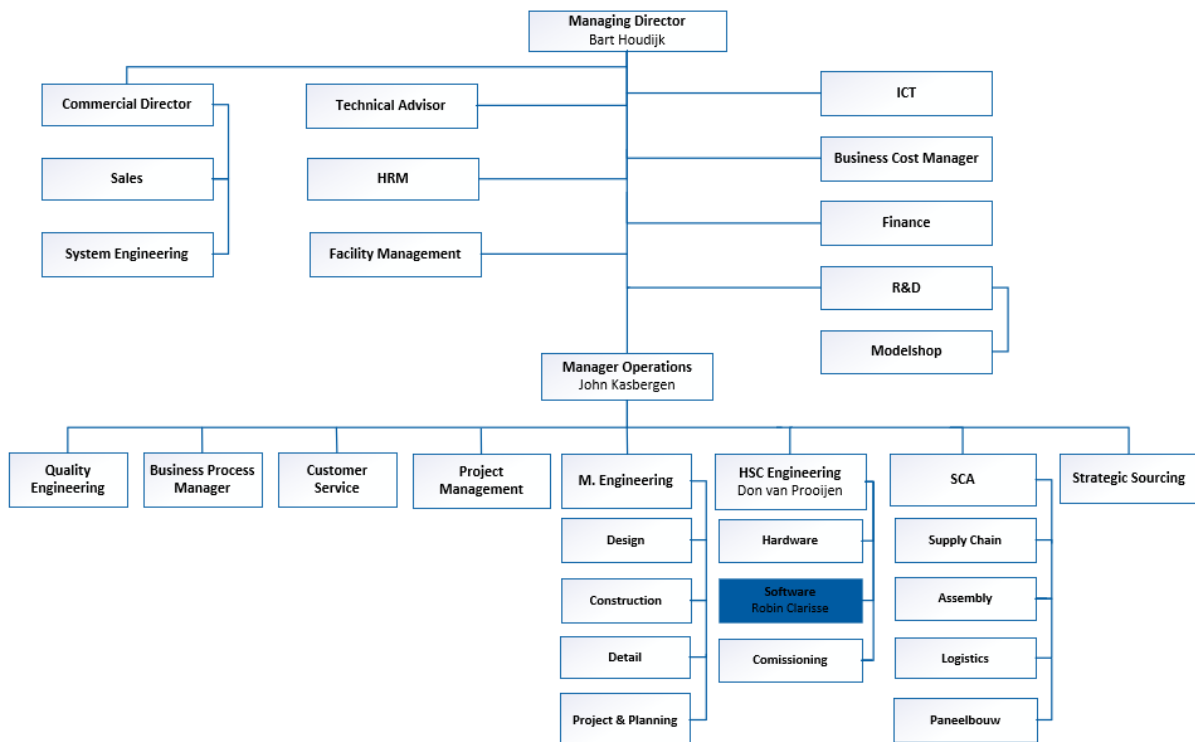
Binnen dit document wordt als eerste gesproken over het bedrijf en haar organisatie. Vervolgens wordt de opdracht en aanpak beschreven. Daarna komt de analyse van de huidige situatie en de gebruikte technologieën. Vervolgens is de uitvoering van de fases binnen de RAD-methode beschreven. Als laatste worden de beroepstaken verdedigd, het product geëvalueerd, het proces geëvalueerd en gereflecteerd op de gehele afstudeerstage.

2. Houdijk Holland

Houdijk is een commerciële machinefabrikant voor de biscuit industrie. Het in 1922 opgerichte bedrijf is gevestigd in Vlaardingen en heeft een wereldwijde klantenkring. Houdijk Holland concentreert zich voornamelijk het transporteren van biscuits binnen een fabriek, bijvoorbeeld biscuits van oven naar verpakkingsmachine brengen. Ze onderscheiden zich van concurrenten door de grote hoeveelheden koek, die ze in hoog tempo kunnen transporteren.

2.1 Organisatie

De afgelopen anderhalf jaar zijn tientallen werknemers bij Houdijk Holland aangenomen en zijn ze gegroeid tot ongeveer 200 werknemers. De afstudeeropdracht zal uitgevoerd worden op de afdeling software engineering onder leiding van teamleider en opdrachtgever Robin Clarisse. In Figuur 1 is het organogram van Houdijk Holland weergegeven en is de software afdeling donker blauw gekleurd. De functie van de software afdeling is het ontwerpen en schrijven van programmacodes voor PLC's, HMI's en vision systemen.



FIGUUR 1 ORGANOGAM HOUDIJK HOLLAND

2.2 De werkomgeving

Houdijk Holland heeft een werkplek geregeld, waar vijf dagen per week gewerkt kan worden. De werkplek bevindt zich in de kantoorruimte van de afdeling software engineering. Dit kantoor beschikt over 30 werkplekken, waar gemiddeld tussen de 2 en 12 werknemers aanwezig zijn.

Iedere maandag ochtend heeft een online voortgangvergadering met de gehele software afdeling plaats gevonden, waarin iedere deelnemer in het kort aangaf wat zijn of haar werkzaamheden voor de komende week waren. Daarnaast is in de eerste helft van de stage elke vrijdag door de stagiaire een update via de mail gegeven over de uitgevoerde werkzaamheden van die week. Voor zowel de

bedrijfsmentor als de stagiaire was de afstudeerstage een nieuwe ervaringen. Hierdoor moest in het begin gezocht worden naar een goede balans in de communicatie. Voorbij de helft van de stage zijn de wekelijkse updates per mail geleidelijk vervangen door informele mondelingen updates. Dit was mogelijk doordat de bedrijfsmentor en stagiaire elkaar hadden leren kennen, elkaar vaak op kantoor zagen en omdat er een goede verstandshouding was ontstaan.

In verband met het COVID-19 virus en voor het virus gestelde maatregelen had de stagiaire de mogelijkheid om zelf te kiezen tussen thuiswerken en op kantoor werken. Uiteindelijk is er gemiddeld een tot twee dagen per week thuisgewerkt. Tijdens en rondom de ziekteverzuim periode van de stagiaire is thuis gewerkt.

3. Opdrachtomschrijving

Om duidelijk de aanleiding van de HMI-vertaaltool te beschrijven, is gekozen om deze te verdelen in vier onderdelen. Als eerste zal de probleemstelling beschreven worden. Vervolgens zal in de paragraaf “afbakening” de focus van het project aangegeven worden. Daarna wordt de doelstelling geformuleerd. Als laatste wordt duidelijk gemaakt welke producten opgeleverd worden.

3.1 Probleemstelling

Door de recente groei van projecten bij Houdijk lopen de zeventien Software Engineers en vijf Commissioning Engineers te vaak tegen de inefficiënte werkwijze rond het vertalen van HMI teksten aan.

Regelmatig worden (delen van) bestaande HMI-applicaties opnieuw gebruikt. Deze schermen zijn dan vaak al (deels) vertaald naar de gewenste taal van de klant, maar die vertaling gaat verloren door de huidige werkwijze. Ook kunnen tijdens het vertalen geen aanpassingen worden gedaan aan de HMI, omdat dit problemen oplevert bij het importeren van de teksten. Het ontwikkelproces en vertaalproces kunnen daardoor nu niet parallel verlopen. Deze problemen komen voor bij 60 tot 80 HMI's per jaar. Dit leidt tot irritaties bij medewerkers, onnodige kosten voor de redundante vertalingen en mogelijk vertraging van projecten, doordat er niet doorontwikkeld kan worden wanneer het vertaalbureau bezig is met vertalen.

3.2 Afbakening

De HMI-vertaaltool betreft het vertalen van de natuurlijke taal van een door Houdijk Holland geprogrammeerde Rockwell HMI. Hierbij gaat het om woorden en zinnen die de klant te zien krijgt. De code van de betreffende HMI moet zijn opgebouwd in FactoryTalk View Studio. Er zal geen onderzoek gedaan worden naar pasklare oplossingen van derden. De HMI-vertaaltool zal een zelf ontwikkelde applicatie worden, met als doelgroep software en commissioning engineers.

3.3 Doelstelling

Na afloop van de opdracht is er een HMI-vertaaltool met user interface gerealiseerd, waarmee software en commissioning engineers HMI's in andere talen beschikbaar kunnen stellen.

Hierbij is het mogelijk om als software engineer tijdens het wachten op de vertaling van het vertaalbureau aanpassingen aan de HMI te maken. Ook voorkomt de tool dat er reeds vertaalde teksten opnieuw naar de vertaler worden gestuurd, wat leidt tot kostenbesparing.

3.4 Producten

Binnen de afstudeerstage zullen verschillende producten worden geleverd. De producten vormen samen het stagedossier ter beoordeling van de afstudeerstage. Een deel van deze producten zal goedgekeurd moeten worden door de opdrachtgever R. Clarisse of zijn plaatsvervanger en bedrijfsmentor L. Vlasman. In de onderstaande tabel zijn de producten beschreven en wordt er aangegeven of deze goedkeuring van de opdrachtgever nodig hebben. Wanneer een product geen goedkeuring van de opdrachtgever nodig heeft, zal deze alsnog worden geleverd aan het bedrijf.

TABEL 1 OP TE LEVEREN PRODUCTEN

Product	Beschrijving	Goedkeuring opdrachtgever verplicht
Plan van aanpak	Dit document bevat uitleg over hoe de opdracht zal worden aangepakt.	Ja
BPMN model (situatie vooraf)	Dit Business Process Model and Notation (BPMN model) laat zien hoe het proces voorafgaand aan de opdracht eruit ziet.	Nee
Analyse technologieën	Deze compacte analyse geeft inzicht in de keuze voor bepaalde technologieën en wat deze technologieën te bieden hebben.	Ja
Backlog	In de backlog staan de systeemeisen in volgorde van prioriteit.	Nee
Prototypes	In de “user design” fase wordt er elke iteratie een prototype ontwikkeld. Aan het eind van elke iteratie wordt het resultaat met een van de eindgebruikers besproken.	Nee
HMI-vertaaltool	De HMI-vertaaltool is een integratie van de gemaakte prototypes.	Ja
Testrapport	De HMI-vertaaltool zal middels een acceptatietest op een actueel project getest worden, deze resultaten zullen verwerkt worden in een testrapport.	Nee
Systeemdokumentatie	Dit document zal informatie over de werking, de installatie en het gebruik van de HMI-vertaaltool bevatten.	Nee
BPMN model (situatie achteraf)	Dit BPMN model laat zien hoe het proces er na afronding van de opdracht uit ziet.	Nee
Stage dossier	Alle bovenstaande producten in combinatie met het stageverslag vormen het stagedossier.	Nee

4. Aanpak

Voor deze opdracht is gekozen om met de Rapid Application Development (RAD) methode te werken. In de eerste paragraaf is de keuze voor de RAD-methode toegelicht en in de tweede paragraaf is de structuur en inhoud van de methode uitgelegd. Vervolgens is de binnen de opdracht gebruikte programmatuur opgesomd. In de laatste paragraaf worden de in kaart gebrachte risico's voor de opdracht gegeven met bijbehorende maatregelen en alternatief plan. Meer informatie over de beheersing, randvoorwaarden en planning is te vinden in Bijlage B "Plan van Aanpak".

4.1 Methodologie

De ontwikkelingsmethode is gekozen op basis van een vergelijkingsmatrix. In de matrix zijn de volgende methodes opgenomen: waterval, Rational Unified Process (RUP), Rapid Application Development (RAD), eXtreme Programming (XP) en scrum. Per methode is gekeken of deze de eigenschappen bezit, die van toepassing zijn op de opdracht.

De eigenschappen waar op de ontwikkelmethodes beoordeeld zijn, komen voort uit de 5W+H: wie, wat, wanneer, waar, waarom en hoe. De 5W+H helpen om gebeurtenissen of plannen te formuleren zonder dat aspecten vergeten worden. Vanuit de antwoorden op deze vragen werden eigenschappen geformuleerd.

- *Wie gaat de ontwikkelingsmethode gebruiken?* De stagiaire, een (nog) onervaren programmeur.
- *Voor wat wordt de ontwikkelingsmethode gebruikt?* Een eenpersoons-, softwareproject.
- *Wanneer vindt het project plaats?* Tussen 8 februari en 14 mei (70 werkdagen).
- *Waar vindt het project plaats?* Bij Houdijk Holland in Vlaardingen.
- *Waarom wordt het project gedaan?* Het project zorgt voor een versimpeling van het HMI vertaal proces en heeft als doel om ondersteuning te bieden software en commissioning engineers tijdens het vertalen van een HMI. Gebruikersinput is belangrijk, om de kans op ingebruikneming van de tool te verhogen.
- *Hoe zal het project verlopen?* Het project zal tijdens de ontwikkeling veel input moeten krijgen van eindgebruikers, waardoor de gestelde eisen in loop van het project kunnen veranderen.

Bij Houdijk Holland wordt niet gewerkt volgens een bestaande ontwikkelingsmethode. Ervaren software engineers maken een voorontwerp voor een project. Als het voorontwerp is goedgekeurd door de projectgroep, dan wordt deze toegewezen aan een software engineer die tijd heeft in zijn of haar planning. Deze software engineer programmeert de code, test de code en stel het programma in bedrijf bij de klant. Op kantoor werkt iedere programmeur in zijn eentje en is de sfeer rustig. Bij deze cultuur kunnen veel ontwikkelingsmethodes passen, behalve methodes die dagelijks input of grootschalige en frequente vergaderingen vereisen.

De cijfers in de matrix geven aan hoe een methode zich verhoudt tot de eigenschappen, waarbij de geschiktheid van een methode van 0 (niet geschikt) tot 3 (zeer geschikt) varieert. In tabel 2 is de vergelijkingsmatrix weergegeven.

TABEL 2 VERGELIJKINGSMATRIX ONTWIKKELMETHODES

Eigenschappen	Waterval (Balaji, 2012)	RUP (Kroll et al., 2003)	RAD (Coleman, 1998)	XP (Beck & Andres, 2004)	Scrum (McKenna, 2016)
Goede omgang met onervaren softwareontwikkelaar(s)	3	2	1	1	1
Toepasbaar op eenmansproject	3	1	3	0	1
Toepasbaar op een kleinschalig project van 70 dagen	3	0	3	3	2
Past bij cultuur Houdijk Holland	3	3	3	1	1
Feedback eindgebruikers gedurende de ontwikkeling van tool	0	2	3	3	2
Goede omgang met veranderende eisen	1	2	3	3	3
Totaal behaalde punten	13	10	16	11	10

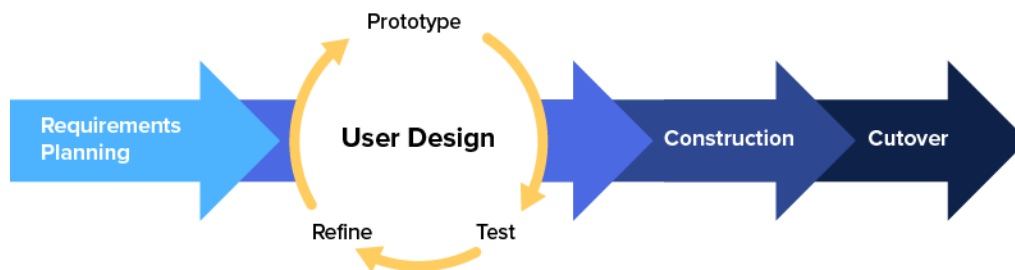
Uit de matrix blijkt dat de RAD-methode het beste aansluit op de eigenschappen van de opdracht. De RAD-methode werkt goed voor kleine teams van ervaren softwareontwikkelaars die projecten met een duur van ongeveer 90 dagen uitvoeren. De methode kan uitgevoerd worden in verschillende teams, maar is ook toe te passen op een enkel team of persoon. Binnen de stage is de HMI-vertaaltol door een enkel persoon, de stagiaire, ontwikkeld.

Een nadeel van RAD is dat er minder aandacht wordt besteed aan niet-functionele eisen (Coleman, 1998, p. 108). De methode is gefocust op het maken van prototypes om samen met de eindgebruiker te reviewen. Hierdoor kan het gebeuren dat niet-functionele eisen als bijvoorbeeld kwaliteit en robuustheid minder worden uitgewerkt, doordat de eindgebruiker dit niet kan zien. De tool zal intern gebruikt worden, dus de verwachting is dat er weinig zwaarwegende niet-functionele eisen worden gesteld. Wanneer er bewust wordt omgegaan met dit risico, kan de methode nog steeds worden toegepast.

4.2 Ontwikkelmethode

De aanpak van de ontwikkeling verliep via de Rapid Application Development (RAD) methode die gebaseerd is op hoge gebruikersinbreng en iteratief ontwikkelen. RAD wordt voornamelijk voor kleinschalige softwareprojecten gebruikt die snel afgerond moeten worden. In de deze paragraaf zal dieper worden ingegaan op de keuze voor deze methode.

RAD bestaat uit vier onderdelen: de requirements analysis, user design, rapid construction en cutover. Hierbij wordt alleen “user design” iteratief uitgevoerd. Figuur 2 geeft de onderdelen van de RAD-methode weer. Vervolgens wordt onder het figuur elk onderdeel apart toegelicht.



FIGUUR 2 RAPID APPLICATION DEVELOPMENT FASES (KISSFLOW, 2020)

Requirements analysis

De “requirements analysis” zal uitgevoerd worden volgens de story mapping methode van Jeff Patton. Deze methode staat uitgewerkt in zijn boek “User Story Mapping: Discover the Whole Story, Build the Right Product” (2014). De methode is gekozen omdat gebruikersinput een belangrijk onderdeel van de methode is. Daarnaast kan een story map goed met veranderingen van eisen en de planning omgaan. De story map is een overzicht systeemeisen per user story op volgorde van prioriteit. Dit overzicht van gewenste systeemeisen wordt gemaakt naar aanleiding van een brainstormsessie met belanghebbenden.

User design

In het onderdeel “user design” worden door middel van iteraties prototypes ontworpen, ontwikkeld en getest. Deze prototypes worden dan met een eindgebruiker bekeken en beoordeeld. Vervolgens wordt er besproken wat er in de volgende iteratie wordt uitgewerkt en of er een onderdeel verbeterd moet worden. Op deze manier blijft de applicatie dicht bij de wensen van de eindgebruiker. De prototypes kunnen per iteratie geheel verschillen, maar mogen ook op elkaar aansluiten. Aan het begin van elke iteratie wordt hier een beslissing over genomen.

De prototypes zullen worden gemaakt in de Visual Studio Community omgeving. Hierin wordt uiteindelijk ook de tool gemaakt, waardoor delen van prototypes gekopieerd kunnen worden.

Binnen de “user design” fase zal het ontwerpen van de prototypes worden gedaan aan de hand van Unified Modeling Language (UML). UML is een modeleertaal die voornamelijk wordt gebruikt bij softwareontwikkeling. UML wordt gebruikt voor verschillende programmeertalen en is daarmee niet gebonden aan een taal of techniek. Bijna alle software engineers bij Houdijk kunnen UML lezen en hebben hier enige ervaring mee. Door het gebruik van UML kan draagvlak binnen de afdeling gecreëerd worden.

In de RAD-methode worden de ontwerpen regelmatig “business, proces en datamodellen” genoemd. Bij deze opdracht wordt dat dus ingevuld door UML-modellen.

Rapid construction

Na het bouwen en reviewen van verschillende prototypes wordt in de fase “rapid construction” een werkend geheel gemaakt. Deze wordt vervolgens gepresenteerd aan de opdrachtgever. Eventueel is het mogelijk om nog terug te vallen op een extra “user design” iteratie, maar dit is niet gebruikelijk.

Cutover

De “cutover” is het moment dat de applicatie wordt geaccepteerd en geïntegreerd in het bedrijf. Bij de HMI-vertaaltool bestaat dit uit het uitvoeren van een acceptatietest en het overdragen van de tool met documentatie over de werking, de installatie en het gebruik van de tool.

Tekortkoming RAD

Werken met ervaringen softwareontwikkelaars is een van de aspecten van de RAD-methode. Dit is niet het geval bij deze opdracht, daarom is er gekozen voor een extra onderzoeksfase om dit te compenseren. Binnen deze onderzoeksfase zijn de keuze voor de programmeertaal en database gemaakt.

Naast de onderzoeksfase heeft er ook een oriënterende fase plaats gevonden om kennis te maken met het bedrijf. Verder heeft er een afsluitende fase plaats gevonden voor het opbouwen van het stagedossier. Beide fases hebben geen invloed gehad op de aanpak voor de ontwikkeling van de tool. De fases hadden wel toegevoegde waarde voor een goed en gestructureerd verloop van de afstudeerstage. In Bijlage B “Plan van Aanpak” staat een duidelijk overzicht van de fases in een Gantt-diagram, met de verwachte tijdsduur per fase.

4.3 Programmatuur

Binnen de applicatie zijn de volgende toepassingen gebruikt:

- Windows .Net 5.0
- Windows Presentation Forms (WPF)
- C#9.0
- EPPLUS versie 4.5
- Mysql.data
- MVVM-light
- Material design WPF

Platforms

- MySQL workbench 8.0
- Visual studio 2019 community
- Visual paradigm online free edition

4.4 Risico's en maatregelen

In deze paragraaf worden risico's genoemd met betrekking tot de opdracht. Naast het risico zal ook een maatregel en een alternatief genoemd worden. De maatregel dient om de kans van risico te verkleinen en het alternatief dient als uitwijkmogelijkheid. De risico's zijn ingedeeld op volgorde van hun score. De score wordt berekend door de geschatte kans in procenten te vermenigvuldigen met de geschatte impact (1-10).

Niet behalen van alle noodzakelijke functionaliteiten

Risico: De stagiaire heeft niet alle elementen vanuit de backlog binnen de gegeven tijd kunnen toevoegen aan de HMI-vertaaltool. Hierdoor bestaat de kans dat de tool voor een deel niet de benodigde functionaliteiten bevat.

Score: Kans 40% * impact 7 = 280

Maatregel: Tijdens het uitkiezen van de systeemeisen die in een iteratie worden ontwikkeld, zullen eerst de noodzakelijke systeemeisen gekozen worden.

Alternatief plan: Belangrijke elementen worden overgedragen aan een softwareontwikkelaar die deze functionaliteiten kan toevoegen. Elementen waar geen directe behoefte aan is, zullen bij het overdragen worden weggelaten.

Niet-functionele eisen worden verwaarloosd

Risico: Een risico bij de RAD-methode is dat de niet-functionele eisen verwaarloosd worden, doordat de eindgebruiker dit niet kan beoordelen.

Score: Kans 35% * impact 7 = 240

Maatregel: Tijdens iedere iteratie van de "user design" fase zal bij het ontwerpen aandacht worden besteed aan de niet-functionele eisen.

Alternatief plan: Wanneer een niet-functionele eis onvoldoende behaald is om de tool te kunnen integreren binnen het bedrijf, zal gekeken worden of met verbeteringen van of toevoegingen aan de huidige code door een ervaren software engineer dit probleem verholpen wordt. Als dat niet mogelijk is, zal opnieuw een tool ontworpen en geprogrammeerd moeten worden.

Onbruikbaarheid

Risico: Houdijk Holland wil de HMI-vertaaltool niet in gebruik nemen, omdat ze de programmeertaal, database en/of ontwikkelomgeving niet willen ondersteunen. Volgens het beleid van Houdijk Holland mogen werknemers niets op een werkstation installeren zonder toestemming. Dit beleid wordt streng nageleefd.

Score: Kans 15% * impact 8 = 120

Maatregel: De stagiaire zal bij de start van de onderzoeksfase toetsen of er programmeertalen, databases en ontwikkelomgevingen niet ondersteund zullen worden. Deze programmeertalen, databases en ontwikkelomgevingen zullen dan niet worden meegenomen in het onderzoek. Zodra bekend is welke database en welke programmeertaal gewenst is, zal hier expliciet toestemming voor gevraagd worden.

Alternatief plan: Als de gewenste programmeertaal en/of database niet ondersteund zal worden door Houdijk Holland, dan zal de tweede beste optie gekozen worden.

Te weinig kennis over databases

Risico: De stagiaire heeft te weinig kennis over databases waardoor dit gedeelte niet (tijdig) voltooid kan worden.

Score: Kans 10% * impact 9 = 90

Maatregel: Het inrichten van de database zal in de eerste iteratie worden uitgevoerd. Bij problemen is er dan nog genoeg tijd om dit op te pakken.

Alternatief plan: Bekijken of het mogelijk is om het proces te verbeteren zonder het gebruik van een database. Bijvoorbeeld door lokaal op te slaan in JSON of XML.

Geen acceptatietest

Risico: Er kan niet worden getest op een actueel project. Bijvoorbeeld door maatregelen vanuit overheid voor het COVID-19 virus of door afwezigheid van een geschikt project.

Score: Kans 20% * impact 3 = 60

Maatregel: Er zal tijdig door de stagiaire aangegeven worden in welke week de acceptatietest gewenst is, zodat de planner kan kijken of dit mogelijk is. Als dit niet mogelijk is in de gewenste week, dan heeft de stagiaire de mogelijkheid om haar eigen planning aan te passen. Bij de bekendmaking van Covid-19 maatregelen die invloed hebben op de acceptatietest, zal de stagiaire bekijken of de acceptatietest eventueel in een andere vorm mogelijk is.

Alternatief plan: Wanneer er geen actueel project beschikbaar is, zal er gekeken worden of er een extra taal met behulp van de tool in een oud project gezet kan worden. Als er niet getest kan worden door het Covid-19 virus en er geen andere optie is, zal er geen acceptatietest uitgevoerd worden. Wel zal aan systeemdokumentatie dan een hoofdstuk toegevoegd worden, waarmee in een latere periode door medewerkers alsnog een acceptatietest kan worden uitgevoerd.

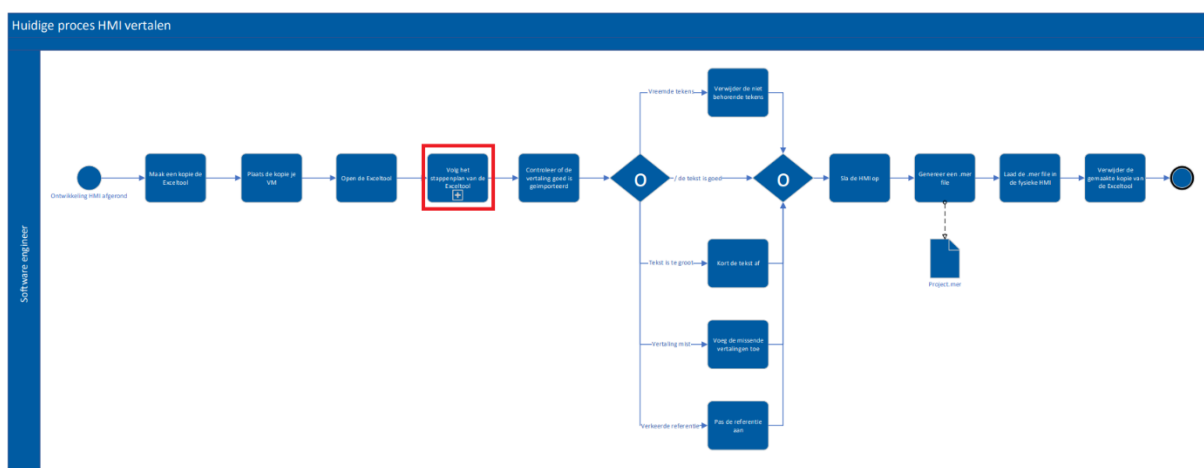
5. Huidige situatie

Houdijk verkoopt ongeveer twintig projecten per jaar. Dit zijn totaaloplossingen en daar zitten vaak meerdere sub-machines in. Die sub-machines zijn over het algemeen allemaal voorzien van een HMI. Gemiddeld zitten er drie tot vier HMI's in een project, dus dat betekent dat er zestig tot tachtig HMI applicaties worden gemaakt per jaar. Naar schatting moet 90% van de HMI's vertaald worden naar een gewenste taal. (R. Clarisse, persoonlijke communicatie, januari 2021)

Op dit moment worden de HMI's vertaald aan de hand van een Exceltool. Bij het huidige "HMI vertaal" proces worden alle teksten uit de HMI geëxporteerd en in zijn geheel naar de vertaler toegestuurd. Als het bestand terugkomt van de vertaler kunnen de vertaalde teksten worden geïmporteerd in de HMI, waarna het programma beschikbaar is in de nieuwe taal. Tussen het exporteren en importeren van de talen kunnen er geen wijzigingen gemaakt worden aan het HMI-programma. De posities van de teksten in de HMI worden via referenties bijgehouden. Eventuele wijzigingen aan de applicatie kunnen deze referenties verstoren. Als dit het geval is, zal de programmeur de teksten met de hand moeten invoeren. (L. Vlasman, persoonlijke communicatie, februari 2021)

Het proces "HMI vertalen" is weergegeven in een Business Process Model Notation (BPMN). Het BPMN-model bevat een sub-proces "volg het stappenplan in de Exceltool" die ook verder is uitgewerkt. In Figuur 3 staat het gehele hoofdproces weergegeven met het sub-proces rood omkaderd en in Figuur 4 staat het sub-proces weergegeven.

Deze twee figuren laten zien hoeveelheid stappen ongeveer nodig zijn om een HMI in de gewenste taal te krijgen. Het doel van de stageopdracht is niet om de hoeveelheid stappen te verminderen, maar de impact per stap te verlagen. Het staat de stagiaire vrij om het proces "HMI vertalen" te veranderen als dit gewenst blijkt. In Bijlage C "BPMN-model huidige situatie" staat meer informatie over de huidige situatie van het HMI vertaal proces. Ook is in deze bijlage een grotere weergave van Figuur 3 en 4 weergegeven, waarbij individuele stappen leesbaar zijn.



FIGUUR 3 BPMN MODEL HUIDIG PROCES



6. Analyse van technologieën

In dit hoofdstuk wordt in de eerste paragraaf uitleg gegeven over waarom het geplande onderzoek vervangen is door een analyse. Daarna wordt in de tweede paragraaf een korte samenvatting per gekozen techniek gegeven.

6.1 Van onderzoek naar analyse

Binnen de afstudeerstage is gekozen voor de RAD-methode. Een vereiste van de methode is dat er gewerkt wordt met ervaren softwareontwikkelaars. De stagiaire behoort (nog) niet tot deze groep en daarom was ter compensatie een onderzoeksfase gepland. Binnen de onderzoeksfase zou worden onderzocht welke programmeertaal gewenst is en in welke database de vertalingen worden opgeslagen. Vanuit Houdijk Holland wordt een stabiele oplossing verwacht die op een werkstation en in een virtuele machine geïnstalleerd kan worden. Bij alle software engineers is Windows 10 op zowel het werkstation als in de virtuele machine het besturingssysteem.

Voor de onderzoeksfase stond 15 werkdagen gepland, maar na een paar dagen was de keuze voor de programmeertaal en database al gemaakt. Op basis van gevoerde gesprekken met verschillende software engineers en de bedrijfsmentor is de programmeertaal C# en database MySQL gekozen. Deze keuze was voornamelijk gemaakt omdat al veel kennis van deze technieken binnen de afdeling beschikbaar is. Daarnaast is het meest recente innovatieproject met C#, de VisionSelector, goed verlopen en wil Houdijk Holland meer gaan investeren in deze programmeertaal.

Omdat de beslissing voor C# en MySQL binnen enkele dagen was gemaakt, is samen met de bedrijfsmentor en begeleidend examiner besloten om de onderzoeksfase om te zetten in een analyse. Dit gaf de stagiaire de vrijheid om meer inzicht en kennis te krijgen in de gekozen technieken voor de HMI-vertaaltool, zonder gebonden te zijn aan een onderzoeksvraag. Daarnaast duurde de analyse geen vijftien werkdagen, maar tien werkdagen. In de vijf werkdagen die beschikbaar zijn gekomen is een extra iteratie in de “user design” fase gepland.

6.2 Gekozen technologieën

De keuze voor de technieken C#, .NET 5.0, WPF, MVVM en MySQL wordt in Bijlage D “Analyse van technologieën” toegelicht. In deze paragraaf worden de gekozen technieken kort toegelicht.

C#

Als programmeertaal is gekozen om C# te gebruiken. Binnen Houdijk Holland is in 2018 de ontwikkeling van de “VisionSelector” gestart. De VisionSelector is een machine, waarbij een C# programma met behulp van camera's bepaalt of een biscuit voldoet aan gestelde kwaliteitseisen. De kennis van en ervaring met de programmeertaal is daardoor al aanwezig binnen de afdeling. Tevens wordt C# vaak gebruikt voor Windows desktopapplicaties en biedt het .NET platform de mogelijkheid om een user interface te maken.

.NET 5.0

C# is onderdeel van het Microsoft .NET platform. Dit platform wordt gebruikt voor het ontwikkelen en uitvoeren van applicaties. Door Microsoft worden drie implementaties van het .NET platform aangeboden: .NET Core, .NET Framework en Xamarin/Mono. De Xamarin/Mono implementatie wordt voornamelijk voor mobiele applicaties gebruikt. De implementatie .NET Core en .NET Framework worden voor websites, servers, console applicaties en desktop applicaties gebruikt. Hierbij kunnen applicaties met .NET Framework alleen op Windows systemen uitgevoerd worden. Applicaties met

.NET Core kunnen uitgevoerd worden op verschillende besturingssystemen, bijvoorbeeld Linux en macOS. (*What is .NET?*, z.d.)

November 2020 is .NET 5.0 uitgebracht, onderdeel van de .NET Core serie en wordt als toekomst gezien voor zowel .NET Core als .NET Framework (Lander, 2020). Door dit toekomst perspectief en de vrijheid betreffende het besturingssysteem is .NET 5.0 het gekozen platform.

WPF

.NET 5.0 biedt de mogelijkheid om met Windows Forms (WinForms), Windows Presentation Forms (WPF) en Universal Windows Platform (UWP) een user interface te maken. WinForms is gemakkelijk te leren maar is erg gelimiteerd op het gebied van dynamisch design. WPF is moeilijk te leren, maar biedt slimme oplossingen voor databinding. UWP is ontwikkeld voor applicaties die alleen op een Windows 10 machine draaien. (Microsoft, 2021)

Voor de HMI-vertaaltool is WPF gekozen, omdat WPF niet gelimiteerd wordt door beperkte grafische mogelijkheden zoals WinForms of maar één versie van Windows aankan zoals UWP. Daarnaast zal de tool veel te maken krijgen met data, waarbij de databindingfunctie van WPF goed aansluit.

Tijdens deze analyse is door de stagiaire een simpel programma gemaakt in WPF om voor aanvang van de RAD-methode ervaring met deze techniek te krijgen. Het voordeel is dat de start van de ontwikkeling soepel zal verlopen door deze ervaring en dat er getest kan worden of de installatie van de ontwikkelomgeving goed is verlopen.

MVVM

Het Model, View, ViewModel (MVVM) ontwerppatroon is een variant van het Model, View en Controller (MVC) ontwerppatroon. Zonder het gebruik van MVVM wordt meestal een groot aantal functionaliteiten direct achter een scherm/pagina geprogrammeerd, ook wel de “code behind” genoemd. De code behind is dan vaak een onoverzichtelijke lijst van functies waar moeilijk aanpassingen in gemaakt kunnen worden en waar functies niet herbruikt kunnen worden. Het MVVM ontwerppatroon zorgt voor overzicht, structuur en een losgekoppelde interface.

Bij MVVM is er een scheiding (ViewModel) tussen de presentatie-laag (View) en de business-laag (Model). Het ViewModel zorgt dat de juiste methodes in het Model worden aangeroepen en dat de verkregen data in het juiste format aan de View beschikbaar wordt gesteld. Het gebruik van het MVVM ontwerppatroon is een investering. De tijd en moeite die dit ontwerppatroon in het begin vergt, wordt later terug verdiend doordat de applicatie overzichtelijk en flexibel is geworden. (Sørensen & Mihailescu, 2010)

MYSQL

MySQL is een propriëitair opensource-managementsysteem voor relationele databases (MySQL, z.d.). Sinds enkele maanden is software engineer W. van der Velden bezig met een documentatiesysteem. Deze webapplicatie slaat de documentatie op in een MySQL database die door bedrijf Strato wordt beheerd. Er is voor MySQL gekozen om de mogelijkheid open te houden dat de database van de HMI-vertaaltool op dezelfde plek gehost kan worden als de database van het documentatiesysteem. Tevens verlaagd het kiezen van een al gebruikte techniek de kans dat Houdijk Holland de tool afkeurt, doordat ze deze techniek niet willen ondersteunen.

7. Requirements analysis

In de “requirements analysis” fase zijn de benodigde systeemfunctionaliteiten geanalyseerd en beschreven. Voor de requirements analyse is de methode user story mapping gebruikt. Voor deze methode is gekozen omdat het een gebruikers actieve methode is en daarom goed bij RAD-methode past.

Volgens de story mapping methode worden de systeemeisen per user story geprioriteerd. Story mapping wordt vaak gebruikt binnen agile ontwikkelingsmethodes doordat de prioritering op basis van gebruikersinput is en systeemeisen gemakkelijk kunnen verschuiven in de planning gedurende de meerdere iteraties. Daarnaast kan een story map gemakkelijk door een gebruiker of opdrachtgever gelezen en geïnterpreteerd worden. (Patton & Economy, 2014)

De story map is tot stand gekomen in vijf stappen:

- 1 Formuleer de user stories
- 2 Verzamel ideeën en activiteiten in systeemeisen voor de applicatie
- 3 Groepeer deze eisen en ideeën per user story
- 4 Prioriteer de systeemeisen en ideeën
- 5 Verdeel de activiteiten in iteraties

In dit hoofdstuk wordt aangegeven hoe de user stories zijn geformuleerd, waarbij een voorbeeld van een user story is gegeven. Vervolgens is beschreven hoe binnen een brainstormsessie de verzameling, groepering en prioritering van de systeemeisen is verlopen. Als laatste zijn de systeemeisen per iteratie verdeeld en is het resultaat van de requirements analyse weergegeven.

Meer informatie over de user stories, systeemeisen en het verloop van de fase is in Bijlage E “Requirements analysis” te vinden.

Als niet-functionele eisen is gesteld dat de applicatie onderhoudbaar moet zijn, zodat bedrijfsmentor L. Vlasman dit na overdracht kan onderhouden. Daarnaast moet de applicatie met data uit de database om kunnen gaan.

7.1 Formuleren user stories

Bij de story mapping methode worden de systeemeisen aan de user stories gekoppeld. Hiervoor zijn door de stagiaire vijf user stories gedefinieerd. Dit is gedaan op basis van haar interpretatie van het huidige “HMI vertaal” proces en gesprekken met verschillende software engineers. Na overleg met de bedrijfsmentor is de user story “Weergave status aanwezige vertalingen en ontbrekende vertalingen” toegevoegd. Deze user story levert geen document op of verandering in de database, maar is wel een goede reden om de HMI-vertaaltool op te starten. Uiteindelijk zijn de volgende zes user stories gedefinieerd:

- ApplicatieInstellen: Applicatie instellen
- StatusWeergeven: Weergave status benodigde, aanwezige en ontbrekende vertalingen
- TranslateListOntvangen: Excel met ontbrekende vertalingen ontvangen
- DatabaseAanvullen: Database aanvullen met nieuwe vertalingen
- VertalingOntvangen: Importdocument ontvangen voor de HMI met juiste vertalingen
- OnderhoudDatabaseUitvoeren: Onderhoud aan database uitvoeren

De geformuleerde user stories zijn in UML use case format geschreven. In Figuur 5 staat de use case “DatabaseAanvullen” weergegeven. De andere use cases zijn op dezelfde manier opgebouwd en te vinden in Bijlage E “Requirements analysis”.

Use case: DatabaseAanvullen

ID: 4

Korte omschrijving:

Database aanvullen met nieuwe vertalingen.

Primaire actors:

Software engineer.

Secondaire actors:

Geen.

Precondities:

1. De HMI vertaaltool is opgestart en er is verbinding met de database.
2. Aanwezigheid van Excel bestand met in kolom A originele teksten en in kolom B de vertalingen.

Hoofd leidraad:

1.
 - a. De software engineer selecteert de originele taal.
 - b. De software engineer selecteert de gewenste taal.
 - c. De software engineer knop op de knop “Upload Excel”
2. De applicatie opent een venster waar een bestand ingeladen kan worden.
3.
 - a. De software engineer laadt het Excelbestand van het vertaalbureau in.
 - b. De software engineer klikt op de knop “Add translations”.
4. De applicatie geeft aan hoeveel vertalingen aan de database zijn toegevoegd.
5. De software engineer sluit de applicatie af.

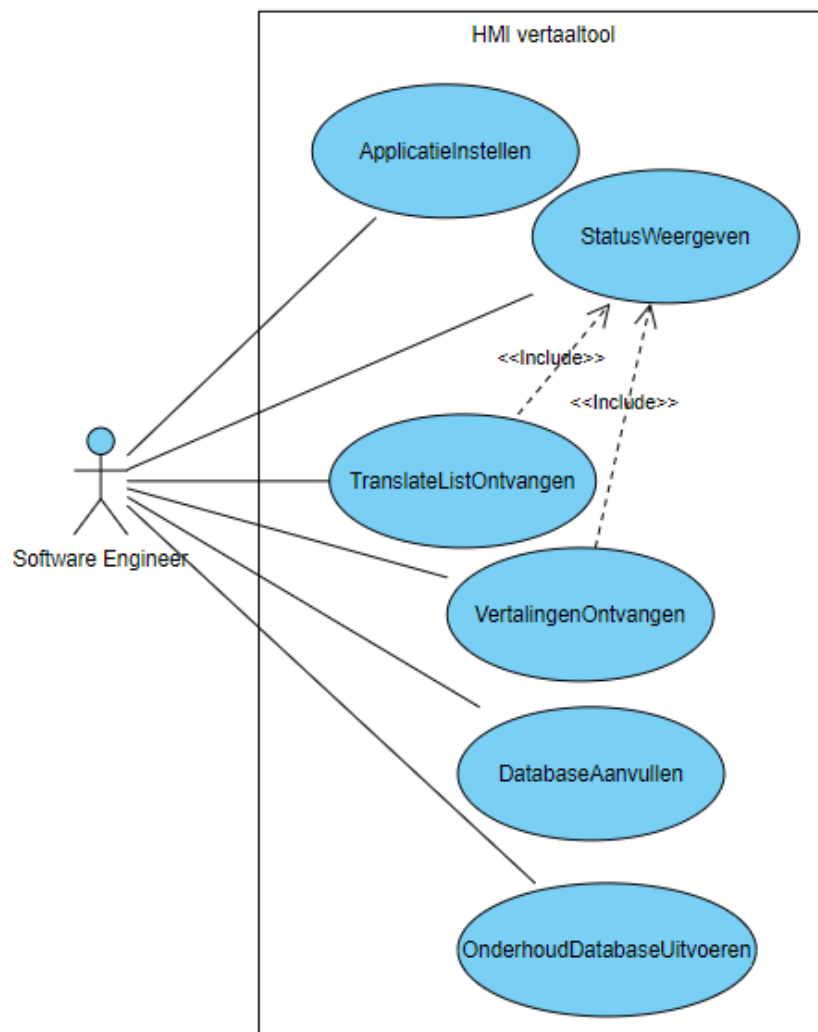
Postcondities:

De applicatie is afgesloten.

Alternatieve leidraad:
Geen.

FIGUUR 5 USE CASE 4 DATABASE AANVULLEN

De geformuleerde use cases zijn in een use case diagram in Figuur 6 weergegeven. Binnen het kader van het systeem staan de zes use cases. Om de use case TranslateListOntvangen en VertalingenOntvangen uit te kunnen voeren, moet eerst de use case StatusWeergeven worden uitgevoerd. De database is niet weergegeven als actor, omdat dit als onderdeel van het systeem wordt beschouwd en niet als buitenstaand component.



FIGUUR 6 USE CASE DIAGRAM

7.2 Verzamelen, groeperen en prioriteren

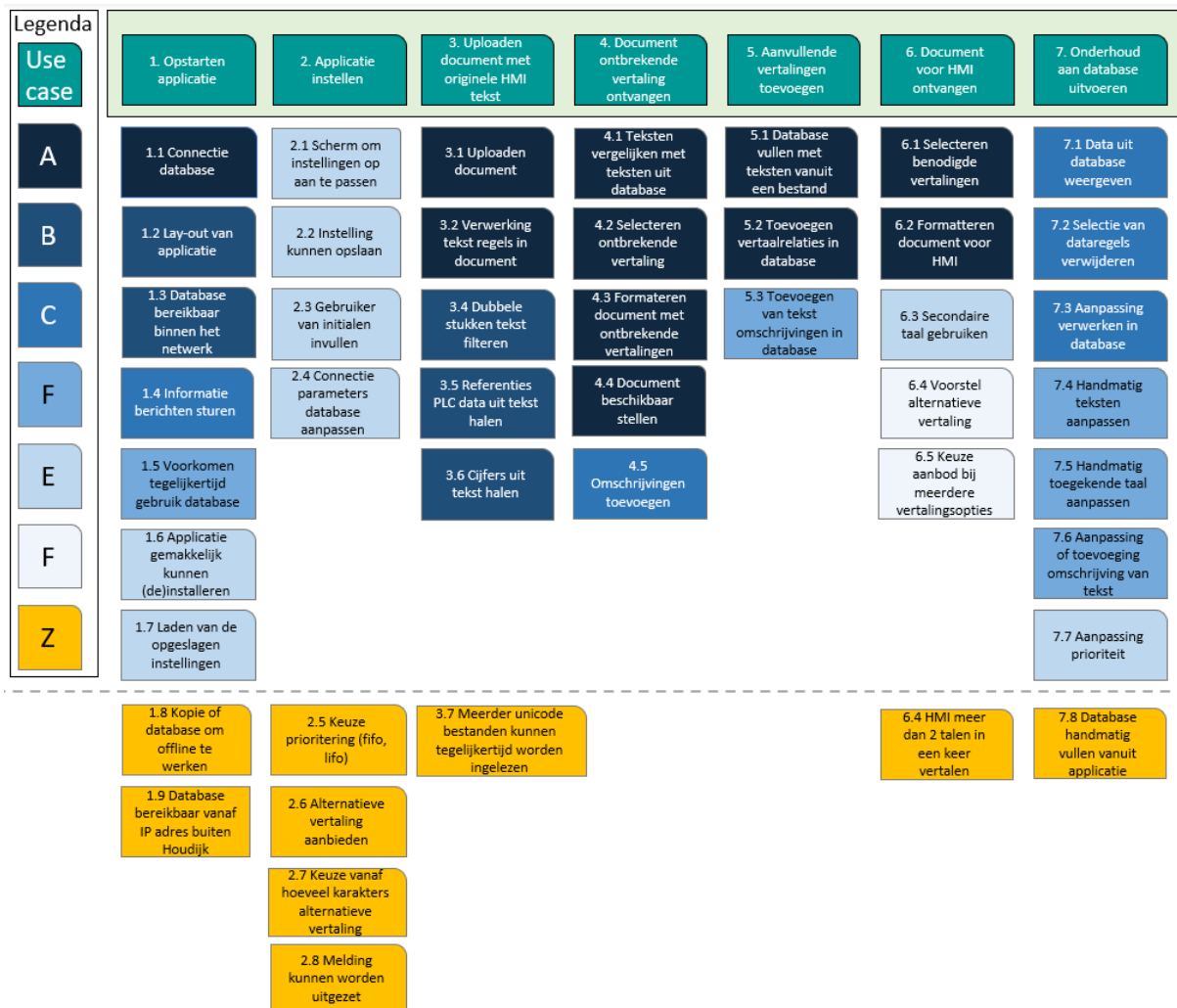
Het verzamelen van ideeën startte al op de eerste dag van de stage waarbij ideeën als aantekening werden opgeschreven. Elke opgevangen idee tijdens bijvoorbeeld een praatje bij het koffiezetapparaat werd opgeschreven. Tijdens een brainstormsessie met de bedrijfsmentor zijn deze ideeën besproken en tot concrete systeemeisen verwerkt. Het resultaat was een grote lijst met systeemeisen. Aan het eind van de brainstormsessie werden deze eisen per user story genoteerd en geprioriteerd.

De prioritering bestond uit het toekennen van een cijfer per systeemeis. Dit konden de cijfers 1 tot en met 10 zijn. Hierbij was 1 een absoluut noodzakelijke systeemeis en 10 was leuk om te hebben, maar geen prioriteit. Nadat alle systeemeisen een cijfer toegekend hadden gekregen, is door de stagiaire kritisch naar de lijst gekeken. Toen bleek dat tussen de systeemeisen met waarde 4 en 5 bijna geen verschil in prioriteit zat en dat er geen eis met waarde 7 bleek te zijn. Ook was het gebruik van cijfers niet handig, omdat de systeemeisen al met een cijfer begonnen en daardoor verwarring veroorzaakte. Daarom is besloten om groepen bij elkaar te voegen en letters in plaats van cijfers te gebruiken.

De letter A representeert de groep met eisen waaraan minimaal voldaan moet worden om een werkende HMI-vertaaltool te kunnen leveren. De eisen met de letters B tot en met F zijn systeemeisen die gewenst zijn door de bedrijfsmentor. Hierbij wordt aan categorie B de meeste en aan categorie F de minste prioriteit gegeven.

Categorie Z representeert de groep met eisen waarvan niet verwacht wordt dat deze binnen de gegeven tijd behaald zullen worden. De categorie Z kan gezien worden als backlog, een lijst van (nog) niet ingeplande eisen. De stagiaire heeft de keuze om een systeemeis vanuit een volgende iteratie naar voren te halen of een eis uit de categorie Z te verwerken als er tijd binnen een iteratie over is.

Onderstaand Figuur 7 geeft het resultaat van de verzamelde, gegroepeerde en geprioriteerde systeemeisen weer. De categorieën A tot en met F hebben een blauwtint als kleur, hoe donkerder de kleur blauw hoe hoger de prioriteit van de eis is. De geel gekleurde categorie Z wordt gezien als backlog en is gescheiden van de andere systeemeisen middels een stippellijn. Een grotere weergave van dit figuur is in Bijlage E “requirements analysis” te vinden.



FIGUUR 7 STORY MAPPING

7.3 Eisen verdelen per iteratie

Na het prioriteren van de eisen, zijn deze over vijf iteraties verdeeld. Tijdens deze verdeling zijn de eisen die enigszins bij elkaar horen in dezelfde iteratie geplaatst. Bijvoorbeeld de eisen 3.3, 3.4 en 3.5 gaan alle drie over het verwerken van tekst. Door bij elkaar horende eisen in dezelfde iteratie te plaatsen wordt verwacht dat benodigde tijd voor de realisatie per eis korter zal zijn dan wanneer de eisen worden verdeeld over verschillende iteraties.

De hoeveelheid ingeplande eisen per iteratie van vijf werkdagen is ambitieus. Dit is gedaan om direct goed te starten en de lat hoog te leggen. Aan het eind van elke iteratie is een review met een eindgebruiker en wordt met de bedrijfsmentor de planning voor de volgende iteratie besproken. Hierdoor kunnen eisen gaan verschuiven binnen de story map.

In het algemeen geldt dat de mening van de eindgebruiker zwaarder telt dan de vooraf gestelde planning. Aanpassingen op aanvraag van de eindgebruiker zullen eerst worden verwerkt en daarna de geplande systeemeisen. Mocht er onenigheid ontstaan bij het bepalen van de planning, dan bepaald uiteindelijk de bedrijfsmentor de planning voor de aankomende iteratie.

Het doel is om de eisen uit categorie A tot en met F af te ronden binnen de vijf iteraties. Het minimale wat behaald moet worden is categorie A. De verwachting is dat de verwerking van de eisen uit categorie A, B en C haalbaar zullen zijn.

In overeenstemming tussen de bedrijfsmentor en de stagiaire zou het mogelijk zijn om nog een zesde iteratie te plannen, mocht dit nodig geacht worden.

Figuur 8 geeft de user story map met verdeling per iteratie weer.

Omdat eis 7.1 goed te combineren was met eis 1.1 is deze in de eerste iteratie verwerkt, ondanks dat deze eis uit categorie C komt.

Use case	1	2	3	4	5	6	7
Iteratie 1	1.1		3.1 3.2	4.1 4.2		6.1	7.1
Iteratie 2	1.2 1.3			4.3 4.4	5.1 5.2	6.2	
Iteratie 3	1.4 1.5		3.3 3.4 3.5	4.5	5.3		7.2 7.3
Iteratie 4	1.6 1.7	2.1 2.2 2.3 2.4					7.4 7.5 7.6
Iteratie 5						6.3 6.4 6.5	7.7

FIGUUR 8 STORY MAP PER ITERATIE

8. User design

In de fase “user design” zijn in iteraties aan prototypes gewerkt. Een iteratie duurde vijf werkdagen en eindigde met een review met een eindgebruiker. De eindgebruiker was in de meeste gevallen de bedrijfsmentor. Behalve tijdens de review van de derde iteratie, toen vervulde de opdrachtgever de rol van eindgebruiker.

Een iteratie verliep volgens de volgende stappen:

- *Refine*
De refine stap gaat over de verbeterpunten uit de vorige review. Tijdens de review kan het zijn dat de aanwezige eindgebruiker een aanpassing of iets extra's wil hebben. Dit wordt als eerste tijdens de refine stap aan het prototype van de vorige iteratie toegevoegd.
- *Prototype*
Vervolgens zijn in de prototype stap nieuwe functionaliteiten toegevoegd aan het prototype van de vorige iteratie of aan een nieuw prototype. Dit was afhankelijk van de systeemeisen die verwerkt moeten worden. Alleen in de tweede iteratie is een nieuw prototype gestart, in de rest van de iteraties is doorgewerkt aan hetzelfde prototype.
Bij grote toevoegingen of aanpassingen aan het prototype wordt eerst een ontwerp gemaakt in de vorm van een klassendiagram en/of sequentiediagram.
- *Review met test*
Tijdens de review zijn eerst alle afgeronde aandachtspunten en eisen getest tijdens een demonstratie. Na het doorlopen van de testen, wordt de algemene indruk van de eindgebruiker gevraagd en of hij/zij nog bepaalde elementen anders zou willen zien. Binnen de reviewsessie is vaak de planning van de volgende iteratie gemaakt met de bedrijfsmentor.

De refine fase vind als eerste plaats in de iteratie, omdat prioriteit is gegeven aan de mening van de eindgebruiker. Als de refine fase na de review zou komen, dan had de refine fase een vooraf bepaalde duur moeten hebben. Anders zouden de iteraties in lengte van elkaar verschillen. Wanneer de refine als eerste in de iteratie plaats vindt, kan daar tot vijf werkdagen aan tijd voor gereserveerd worden.

Binnen Houdijk Holland is het gebruikelijk om de software in het Engels te maken, dit betreft zowel programmacodes als de user interfaces. Daarom zullen alle prototypes en de uiteindelijke tool in het Engels worden gemaakt.

Binnen de “user design” fase is de HMI-vertaaltool in korte tijd ontwikkeld. Tijdens deze ontwikkeling zijn verschillende ontwerpen gemaakt, besluiten genomen en oplossingen gevonden. Uitgebreide informatie over het verloop van de fase staat beschreven in Bijlage F “user design iteratie 1 t/m 6”. In deze bijlage wordt onder andere per iteratie aangegeven welke systeemeisen zijn voldaan, welke niet en waarom niet.

8.1 Iteratie 1 MVVM

In de eerste iteratie van de “user design” fase lag de focus op de noodzakelijke systeemeisen. Het onderdeel refine is overgeslagen, omdat er geen review was gehouden voorafgaand aan de eerste iteratie. Binnen deze iteratie is het eerste prototype gemaakt aan de hand van een ERD-diagram, verschillende sequentiediagrammen en een klassendiagram.

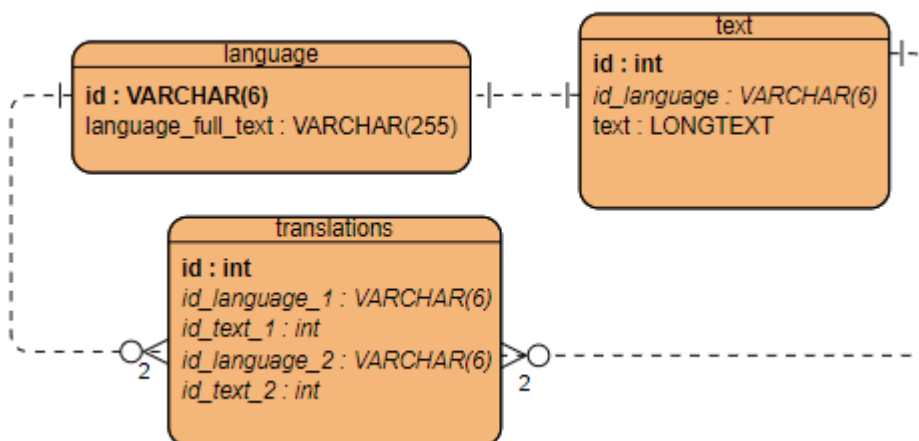
Dit prototype maakt gebruik van het Model, View, ViewModel ontwerppatroon. Dit ontwerppatroon zorgt voor een scheiding (ViewModel) tussen de user presentatie-laag (View) en de business-laag (Model), waardoor de applicatie beter onderhouden kan worden.

Prototype

Binnen dit prototype is een ERD-diagram gemaakt voor de database structuur. Vervolgens zijn drie verschillende sequentiediagrammen gemaakt: een diagram voor het maken van verbinding met de database, een voor het inladen van een unicode exportbestand van FactoryTalk View en een voor het vergelijken van tekst met de database. Vanuit deze sequentiediagrammen is een klassendiagram gemaakt. Voor de functies uit het klassendiagram is pseudocode gemaakt.

MySQL

Binnen de eerste iteratie is een ontwerp voor de databasestructuur met drie tabellen gemaakt. Dit ontwerp is gemaakt volgens een Entity Relationship Diagram (ERD), een methode om relaties tussen objecten weer te geven. In Figuur 9 is het ontwerp weergegeven.



FIGUUR 9 ONTWERP MYSQL DATABASE

De eerste tabel heet “language” en bevat de talen die gebruikt kunnen worden in de applicatie. Bij de primary key “id” wordt de taalcode opgeschreven volgens de ISO 639-1 standaard. De uitleg voor de taalcode staat in de kolom “language_full_text”.

De volgende tabel “text” bezit alle ingevoerde teksten. Elke tekst krijgt een eigen id (primary key) en id_language (foreign key van tabel language).

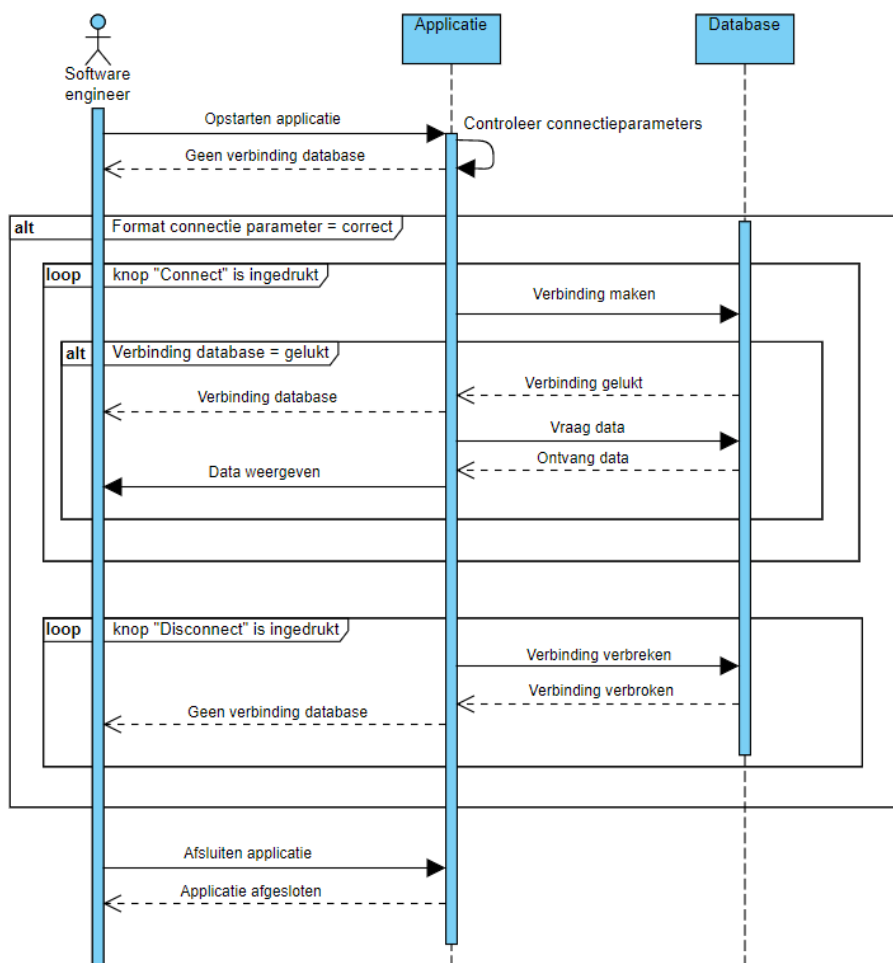
In de tabel “translations” wordt de relatie tussen twee teksten gemaakt. De kolommen id_language_1, id_text_1, id_language_2 en id_text_2 zijn foreign keys naar de tabel language of text.

Sequentiediagrammen

Naar aanleiding van de systeemeisen zijn drie sequentiediagrammen gemaakt:

- Verbinding maken met database en weergeven van data
- Exportbestand van FactoryTalk View laden in de applicatie
- Tekst uit exportbestand vergelijken met data uit database

In Figuur 10 is een voorbeeld van een sequentiediagram weergegeven. In dit diagram wordt verbinding met de database gemaakt en data weergegeven. De applicatie start zichzelf op zonder een database connectie. Zodra er op de knop "Connect" wordt gedrukt, wordt er een verbinding met de database gemaakt. Als dit gelukt is, zal er data worden opgevraagd om te laten zien op het scherm. Zodra er op de knop "Disconnect" wordt gedrukt, zal de connectie met de database verbroken worden.



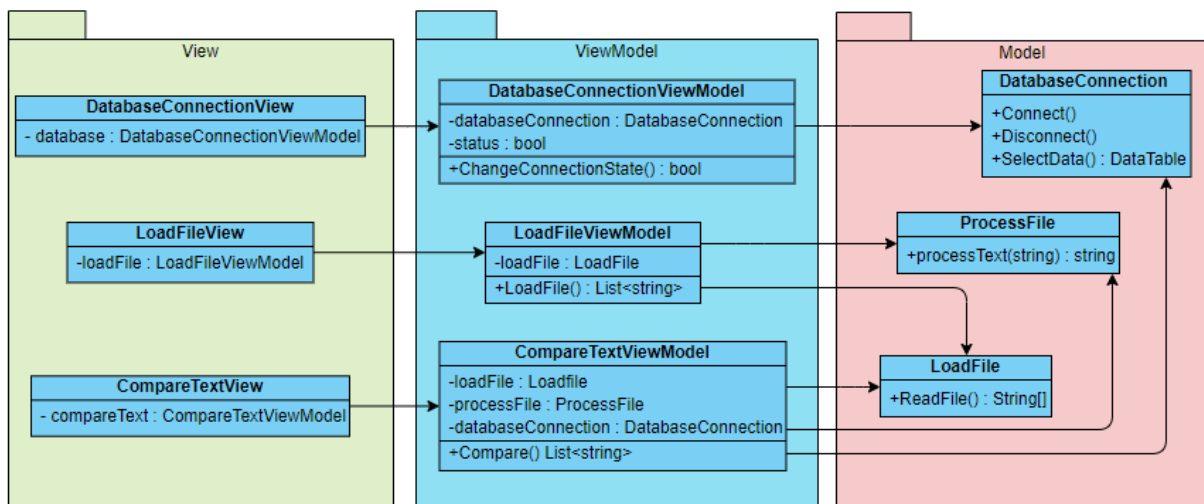
FIGUUR 10 SEQUENTIE DIAGRAM DATA WEERGEVEN

Voor het laden van het exportbestand van FactoryTalk View en het vergelijken van de teksten uit het exportbestand met data in de database, zijn ook sequentiediagrammen gemaakt. Deze twee diagrammen zijn in Bijlage F "user design iteratie 1 t/m 6" te vinden.

Klassendiagram

Na de sequentiediagrammen is een klassendiagram gemaakt. De applicatie is ontworpen volgens het Model, ModelView en View (MVVM) software ontwerppatroon. Dit ontwerppatroon maakt het onderhouden van een WPF applicatie gemakkelijk, doordat er een scheiding tussen de presentatie-laag (View) en de business-laag (Model) is gemaakt. Deze lagen worden gescheiden door het ViewModel. Het ViewModel heeft als taak de juiste processen in het Model aan te sturen en de data die hier uit voortkomt vorm te geven, zodat de View dit kan weergeven.

In Figuur 11 staat het klassendiagram met een duidelijke scheiding tussen de View, ViewModel en Model. Er is gekozen om voor elk sequentiediagram een eigen scherm in de applicatie te maken en daardoor een eigen View en ViewModel te geven.



FIGUUR 11 KLASSENDIAGRAM ONTWERP ITERATIE 1

Na het maken van het klassendiagram zijn pseudocodes gemaakt voor de functies in het ViewModel en het Model. In onderstaand kader staat een voorbeeld van de Compare functie van de CompareTextViewModel klasse. De pseudocodes voor overige functies in het klassendiagram staan in Bijlage F “user design iteraties 1 t/m 6”.

```

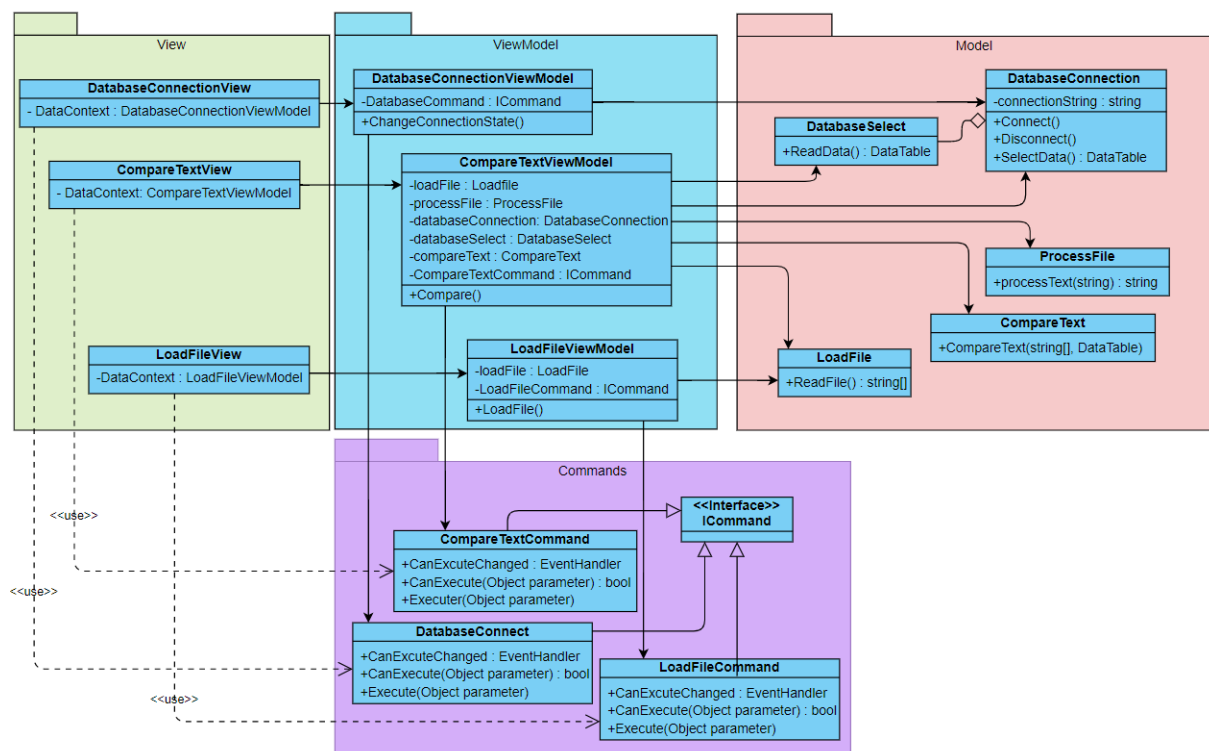
Compare()
    Open venster om bestand te selecteren
    Als bestand is geselecteerd
        Lees bestand uit in string[]
        Voor elke regel in bestand
            Verwerk regel -> processFile.ProcessText(regel)

        Vraag data aan database in DataTable-> DatabaseConnection.SelectData()
        Voor elke regel in string[]
            Als regel voorkomt in DataTable met een vertaling
                Aantal met vertaling + 1
            Anders
                Aantal zonder vertalingen + 1
    
```

Implementatie

Het programmeren van het eerste prototype verliep goed voor het grootste gedeelte. De implementatie van het MVVM ontwerppatroon bleek in het begin lastiger dan verwacht. Dit kwam omdat de aansturing van methodes vanuit de interface via commands lopen. Een command biedt de mogelijkheid om functies uit te voeren na een gebruikersactie in de user interface. Echter moet dan wel per drukknop een command worden aangemaakt, wat veel programmacode oplevert.

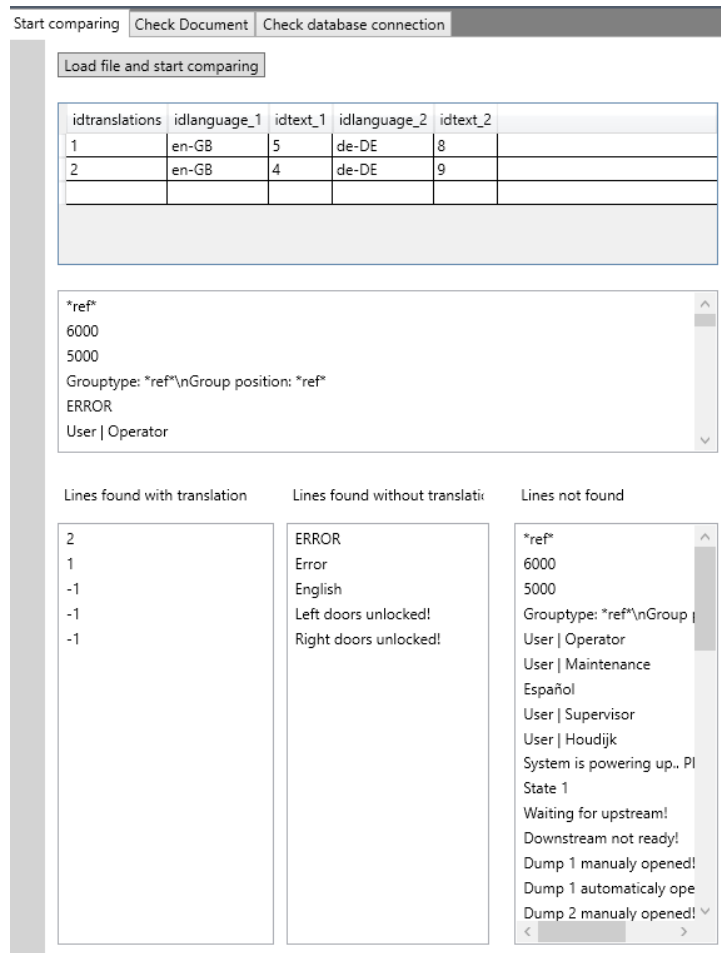
Het klassendiagram in Figuur 12 geeft de uiteindelijke implementatie weer. In het diagram is te zien dat naast de commands ook een extra klasse CompareText is toegevoegd. De code in de methode Compare() van de klasse CompareTextViewModel bleek veel uitgebreider dan verwacht. Daarom is de klasse CompareText toegevoegd die deze functionaliteiten implementeert. Verder zijn niet veel veranderingen doorgevoerd in het klassendiagram.



FIGUUR 12 KLASSENDIAGRAM IMPLEMENTATIE

De bibliotheek `MySQL.Data` van Oracle is gebruikt om vanuit de applicatie verbinding te maken met de MySQL database. Deze bibliotheek kan worden gebruikt om verschillende soorten queries op te stellen en uit te laten voeren door een MySQL database. (*MySQL.Data*, 2021)

In deze iteratie was een ooglijk user interface nog geen vereiste, maar om tijdens de review de werking te kunnen laten zien aan de eindgebruiker is een simpele user interface in elkaar gezet. De user interface CompareTextView is als voorbeeld in Figuur 13 weergegeven. Voor de gehele user interface is in de derde iteratie een ontwerp gemaakt. Deze user interface is echter alleen om de eerste functionaliteiten te testen.



FIGUUR 13 USER INTERFACE COMPARETEXTVIEW

Review

De review heeft plaatsgevonden op maandag 15 maart met eindgebruiker en bedrijfsmentor L. Vlasman. De review startte met een demonstratie van de functionaliteiten. Tijdens deze demonstratie werd actief door zowel de stagiaire als eindgebruiker naar bevindingen gevraagd en deze gedeeld. De conclusie van de demonstratie was dat de gewenste functionaliteiten waren behaald. Vanuit de eindgebruiker waren geen opmerkingen over huidige functionaliteiten.

Vervolgens is de code besproken met de bedrijfsmentor. De bedrijfsmentor merkte op dat er te weinig commentaar in de code stond. Dit was inderdaad nagenoeg niet aanwezig en is in de refine fase van volgende iteratie opgepakt. In overleg met de bedrijfsmentor is besloten om in de prototype fase van de volgende iteratie de ingeplande systeemeisen te implementeren.

8.2 Iteratie 2 format

In de tweede iteratie lag de focus op het formatteren van de documenten die door de vertaaltool worden gemaakt. Hierbij gaat het om twee documenten, de translate list die naar een vertaalbureau gestuurd kan worden en een unicode importbestand die in FactoryTalk View geladen kan worden.

Binnen deze iteratie is een nieuw prototype gemaakt. Enkele functies van de vorige iteratie zijn meegenomen in het nieuwe prototype. Voor dit prototype zijn nieuwe sequentiediagrammen gemaakt en een nieuw klassendiagram gemaakt.

Refine

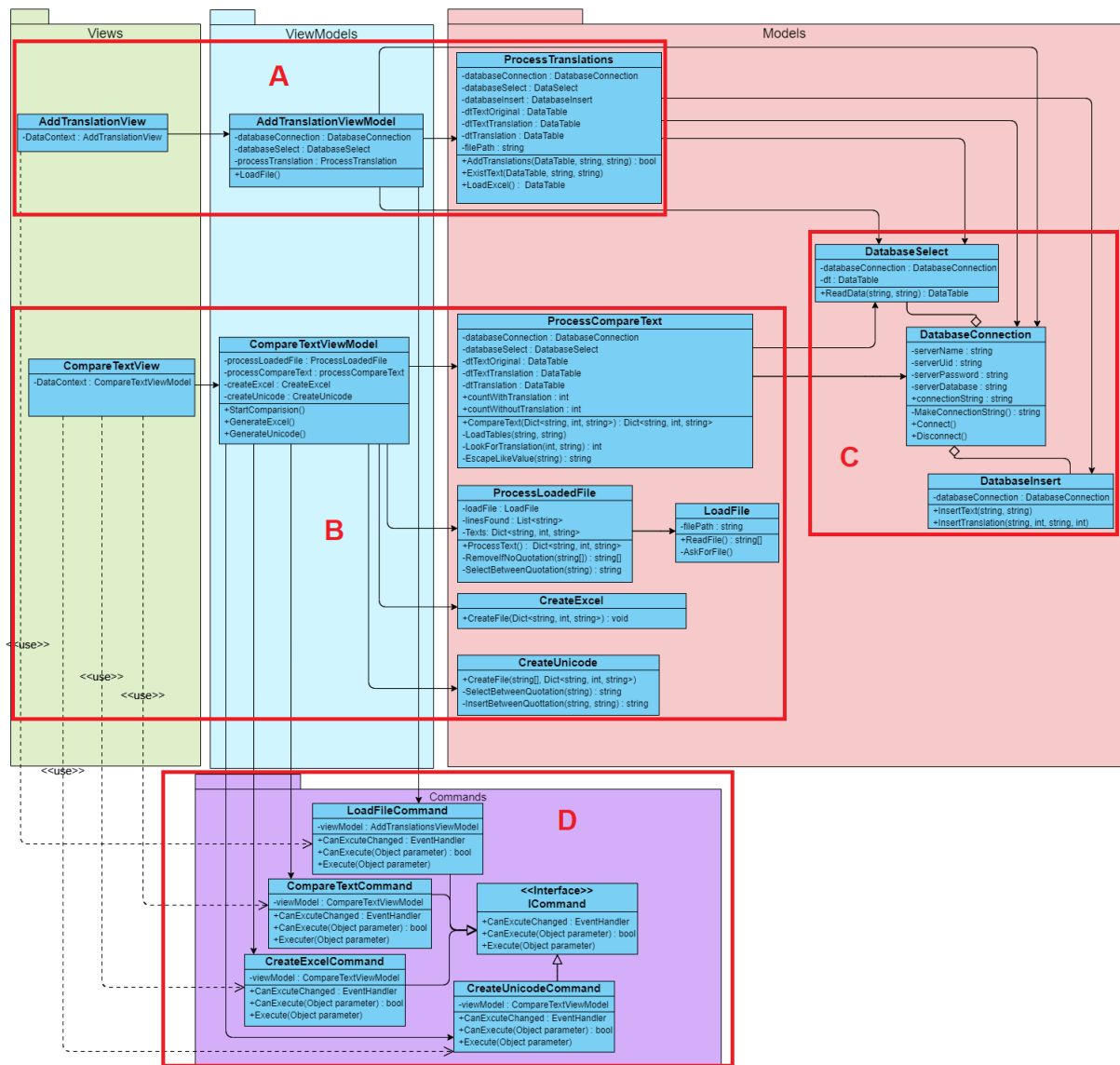
De iteratie begon met de refine stap waarbij aandachtspunten uit de vorige iteratie zijn opgepakt. Afgelopen review was alleen de opmerking dat er te weinig commentaar bij de code stond. In overleg met de bedrijfsmentor is besloten om boven elke functie een beschrijving te zetten en onduidelijkheden binnen functies toe te lichten. Vanwege tijdsbesparing is het commentaar niet toegevoegd aan het prototype van de eerste iteratie. Zoals verwacht is het prototype van de eerste iteratie niet meegenomen in de “rapid construction” fase en daarmee ook niet in de uiteindelijke tool.

Prototype

Voor het prototype van de tweede iteratie zijn twee schermen gemaakt. Het eerste scherm is voor het creëren van documenten met informatie uit de database die is vergeleken met informatie uit het exportbestand van FactoryTalk View (CompareTextView). Het tweede scherm is voor het toevoegen van vertalingen aan de database (AddTranslationsView). Beide schermen hebben een eigen sequentiediagram gekregen. Deze sequentiediagrammen zijn in Bijlage F “user design iteratie 1 t/m 6” weergegeven.

In Figuur 14 is het klassendiagram te zien, die vanuit de twee sequentiediagrammen is gemaakt. In het klassendiagram hebben de CompareTextView (deel B) en AddTranslationsView (deel A) een eigen ViewModel gekregen. In deel C wordt informatie uit de database opgevraagd en vertalingen aan de database toegevoegd. Deel D zijn de commands om de drukknoppen in de user interface aan te sturen. In de Bijlage F “user design iteratie 1 t/m 6” staat de vergrote versie van de delen A tot en met D met pseudocode voor de bedachte functies.

De implementatie van het ontwerp verliep goed. Het uitlezen en aanmaken van een Excelbestand was lastig zonder het gebruik van een externe software bibliotheek. De bibliotheek EPPLUS van J. Källman (2020) bood goede en uitgebreide mogelijkheden voor het werken met Excel. De gebruikte versie 4.5.3.3 is uitgebracht onder GNU Library General Public License (LGPL), wat in het kort betekent dat iedereen alles met de software mag doen.



FIGUUR 14 KLASSENDIAGRAM ITERATIE 2

Review

De review heeft plaatsgevonden op maandag 22 maart met eindgebruiker en bedrijfsmentor L. Vlasman. De review startte met een demonstratie van de functionaliteiten. Tijdens de demonstratie is geprobeerd om het verkregen unicode importbestand in FactoryTalk View te laden. Dit bleek niet goed te gaan doordat de verkregen unicode in format UTF-8 was aangemaakt in plaats van in de benodigde UTF-16 LE. Tevens zagen we dat teksten met alleen spaties niet werden vertaald en daardoor elke keer in de translate list terecht kwamen.

Door tijdstekort binnen de tweede iteratie is het ontwerpen van de user interface in de derde iteratie gedaan. Tevens is besloten om de afgesproken eisen volgens de story map in hoofdstuk 7.3 te verwerken in de derde iteratie.

8.3 Iteratie 3 lay-out

In de derde iteratie lag de focus op de lay-out van de applicatie. Tevens zijn de teksten die in de database geladen worden uniformer gemaakt door PLC-referenties, nummers en enters te vervangen met herkenbare tags. Naast een review met de eindgebruiker is binnen deze iteratie ook een code review gehouden. Deze code review was met C# programmeur R. Hoozeboom.

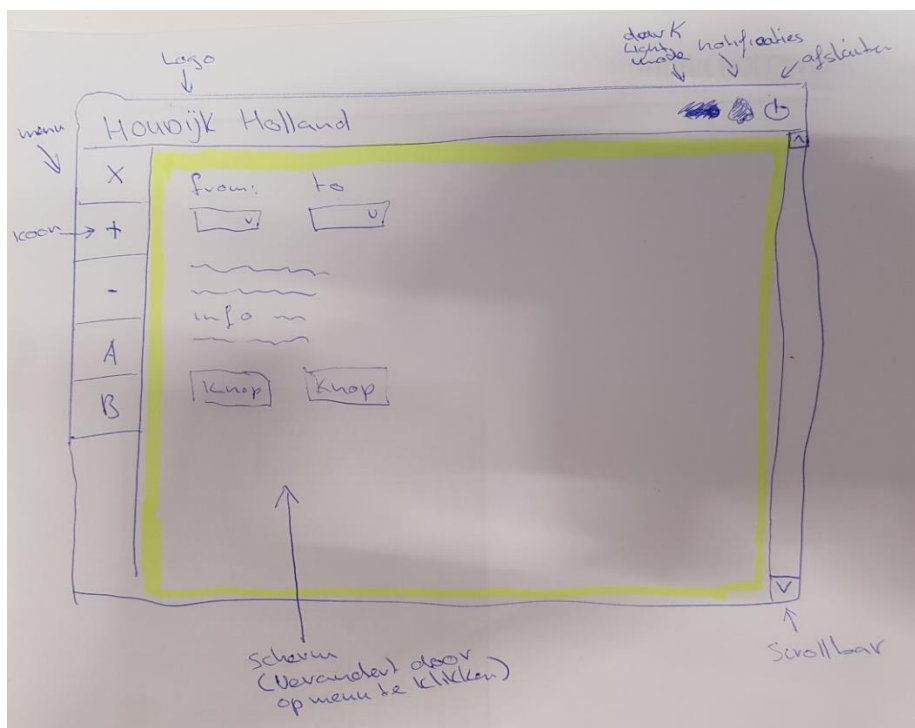
Refine

De iteratie begon met de refine fase waarbij punten uit de review van de vorige iteratie werden opgepakt. Tijdens de review van de tweede iteratie konden twee eisen niet volledig worden afgerond, omdat een paar kleine fouten optraden tijdens de demonstratie. Het verbeteren van deze fouten en het verwerken van de overige wensen van de eindgebruiker kostte minder dan een dag. Hierna kon in de prototype fase verder gewerkt worden aan de eisen vanuit de planning.

Prototype

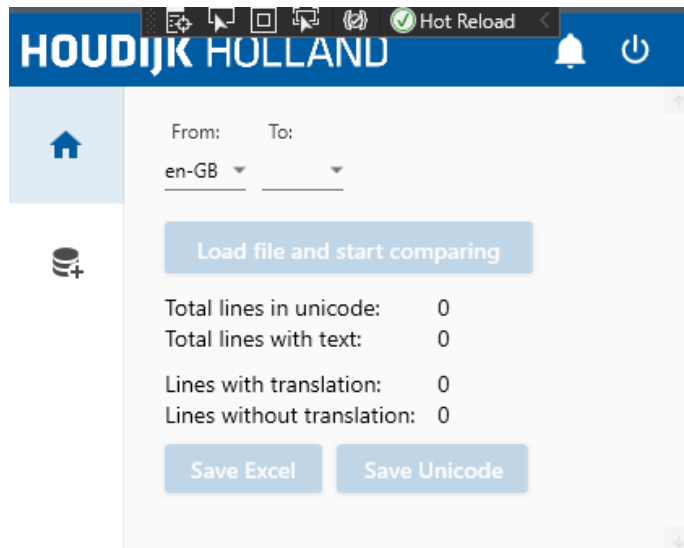
In deze prototype fase zijn de teksten die door de gebruiker worden ingeladen uniformer gemaakt. Voor het uniformer maken van teksten is pseudocode gemaakt. In Bijlage F “user design iteratie 1 t/m 6” is de pseudocode voor het uniformer maken van teksten beschreven.

Voor de lay-out van de applicatie is een ontwerp op papier gemaakt. In Figuur 15 is het ontwerp van de user interface afgebeeld. Het ontwerp bevat een navigatiemenu om tussen verschillende schermen (Views) te wisselen. Alles buiten het geel omkaderde vak wisselt niet mee met de schermen en blijft zichtbaar gedurende het gehele gebruik van de applicatie. De hoofdkleuren van de applicatie zijn de Houdijk kleuren blauw en wit. Voor accenten kunnen de Houdijk Holland kleuren donker grijs en groen gebruikt worden.



FIGUUR 15 ONTWERP USER INTERFACE

De user interface moest een simpele en moderne uitstraling krijgen. Dit is bereikt met de MaterialDesignThemes bibliotheek (Willock, 2021). Deze bibliotheek biedt flexibele stijlsjablonen aan voor de standaard WPF user interface elementen. In Figuur 16 is de uitwerking van de user interface ontwerp weergegeven.



FIGUUR 16 UITWERKING USER INTERFACE

Review

De review van de derde iteratie heeft plaatsgevonden op maandag 29 maart met eindgebruiker en opdrachtgever R. Clarisse. De review startte met een demonstratie van de functionaliteiten. De demonstratie werd vaak onderbroken om te discussiëren over de gemaakte keuzes. Meneer Clarisse heeft tijdens deze review zijn verwachtingen en ideeën kunnen delen.

De review was nuttig, omdat veel nieuwe op- en aanmerkingen werden gemaakt. Alleen is tijdens de review geen aandacht besteed aan het bepalen van de eisen die worden meegenomen in de vierde iteratie en de eisen die worden doorgeschoven in de planning.

Vrijdag 26 maart is een code review met software engineer R. Hoogeboom georganiseerd door de stagiaire. Meneer Hoogeboom is afgelopen drie jaar bezig geweest met het ontwikkelen van een applicatie in C#, die op hoge snelheid bepaald of koekjes weggegooid moeten worden met behulp van camera's. Binnen de code review is aandacht besteed aan de opbouw van het programma, MVVM, commands, exceptions en codestijl van de HMI-vertaaltool. De review was een open gesprek waarbij ieders mening van bepaalde onderdelen en technieken werden besproken. De volgende aandachtspunten voor de tool kwamen naar voren.

- Minder onnodige commentaren, in de code was nog geen juiste balans in commentaar gevonden.
- Namen van klassen veranderen naar objecten in plaats van methodes. De naam van een klasse moet vertellen wat de klasse is en niet wat de klasse doet.
- Gebruik van properties in plaats van publieke variabelen. In C# wordt vaak gebruik gemaakt van properties en vrijwel nooit van publieke variabelen .

De aandachtspunten van meneer Clarisse en meneer Hoogeboom zijn in de refine fase van de vierde iteratie opgenomen.

8.4 Iteratie 4 progressie balk

In de vierde iteratie moesten veel aandachtspunten uit de review van de derde iteratie verwerkt worden. Hierdoor is alleen de eis 1.4 “sturen van informatie berichten” vanuit de story map in hoofdstuk 7.2 verwerkt. De overige tijd is volledig geïnvesteerd in de refine fase.

Refine

Naast een aantal meerdere kleine aandachtspunten uit de review, is voornamelijk gewerkt aan de volgende drie grote onderdelen:

- Toevoeging van een progressie balk, waardoor synchrone methodes vervangen moesten worden met asynchrone methodes.
- Het gebruik van RelayCommand (Bugnion, 2018) en AsyncRelayCommand (Thiriet, 2019), om het aanmaken van nieuwe commands voor drukknoppen in de user interface efficiënter te laten verlopen.
- Toevoeging van de abstracte klasse “ViewModelBase” om variabelen en functies die in elk ViewModel voorkomen in een aparte klasse te verwerken. Hierdoor zijn de ViewModels leesbaarder en overzichtelijker geworden.

De progressiebalk in de UI liep niet geleidelijk vol, omdat tijdens het uitvoeren van methodes de UI niet werd bijgewerkt. Dit is verholpen door de betreffende methodes asynchroon te maken. Aan de hand van een voorbeeld in het boek “Concurrency in C# Cookbook” is pseudocode gemaakt (Cleary, 2014, pp. 21-22). In Bijlage F “user design iteratie 1 t/m 6” is de pseudocode voor het aansturen van de progressiebalk beschreven.

Prototype

Binnen deze iteratie is veel tijd besteed aan de refine fase. In de prototype fase is het alleen gelukt om eis 1.4, versturen van informatie berichten, toe te voegen aan het prototype. Dit is gedaan door middel van het grafische component snackbar. Een snackbar is een stuk tekst die op het scherm verschijnt en na een bepaalde tijd verdwijnt. Hierbij kunnen meerder berichten in een wachtrij gezet worden.

Review

De review heeft op dinsdag 6 april plaatsgevonden met de bedrijfsmentor. De bedrijfsmentor was enthousiast over de user interface die in de review is gedemonstreerd. Zeker de progressie balk vond ze een waardevolle toevoeging.

Maandag 29 maart is te horen gekregen dat het recent ontwikkelde documentatiesysteem van software engineer W. van der Velden gehost gaat worden op een server van Houdijk Holland. Daarbij is toestemming gegeven dat de database van de HMI-vertaaltool naast de database van de systeemdocumentatie gehost mag worden. Met de bedrijfsmentor is afgesproken om in de vijfde iteratie de database in de nieuwe omgeving te plaatsen en te starten met het scherm om onderhoud aan de database te kunnen uitvoeren.

8.5 Iteratie 5 database

In de vijfde iteratie is de lokale database op het werkstation van de stagiaire overgezet naar de server van Houdijk Holland en er is een scherm aangemaakt voor de het onderhoud aan de database (MaintenanceView). In dit scherm kunnen vertalingen uit de database verwijderd worden.

De vijfde iteratie zou eigenlijk de laatste iteratie zijn voor de start van “rapid construction” fase. In overleg met de bedrijfsmentor is besloten om een zesde iteratie te doen. Binnen deze zesde iteratie worden alleen de noodzakelijke eisen verwerkt.

Refine

Tijdens de refine fase is op verzoek van de bedrijfsmentor de prioritering van vertalingen veranderd. Voorheen werd de vertaling met het laagste ID gekozen, dit is aangepast naar de vertaling met het hoogste ID. Zodat bij dubbele vertalingen de meest recente vertaling wordt gekozen.

Prototype

Na de refine fase is de database van de HMI-vertaaltool aangemaakt op de server van Houdijk Holland. Door rechtenbeperking is het niet mogelijk om via de databaseconnectie van de tool bij de database van de systeemdokumentatie te komen. De database is alleen bereikbaar via het Houdijk Holland netwerk range. Gebruikers die zich buiten het kantoor bevinden kunnen met een VPN verbinding maken.

Het is gelukt om vanuit de applicatie vertalingen in de database te verwijderen. Om dit voor elkaar te krijgen is eerst een sequentiediagram gemaakt en een ontwerp van de verwachte lay-out op papier gemaakt. In Bijlage F “user design iteratie 1 t/m 6” is het sequentiediagram en user interface ontwerp van de MaintenanceView weergegeven.

De volgende eigenschappen en functionaliteiten zijn ook aan de MaintenanceView toegevoegd:

- Het laden van data duurt lang, hierdoor is besloten om maximaal 100 regels te laden.
- Naast taal selectie is het ook mogelijk om vertalingen te selecteren die bepaalde teksten of tekens bevatten in de originele tekst en/of de vertaling.
- Om vertalingen te kunnen selecteren en verwijderen is een selectievakje voor elke vertaling toegevoegd.
- Het verwijderen van vertalingen uit de database is een operatie die niet teruggedraaid kan worden. Daarom wordt er een extra bevestiging aan de gebruiker gevraagd.

Review

In de review op maandag 12 april wilde de bedrijfsmentor graag meer functionaliteiten zien in het onderhoudsscherm. Zoals het aanpassen van vertalingen en vertalingen selecteren op invoerdatum. In deze review is aandacht besteed aan het selecteren van de laatste systeemeisen en het maken van een planning voor de laatste iteratie. Binnen deze planning is een volgorde van de eisen gemaakt en een inschatting van de benodigde tijd per eis. Zodat aan het einde van de zesde iteratie geen eisen over zouden blijven.

8.6 Iteratie 6 onderhoud

In de zesde en laatste iteratie zijn extra functionaliteiten aan het onderhoudsscherm toegevoegd. Tijdens de review van de vijfde iteratie is de volgorde van het verwerken van de eisen bepaald. Hierdoor was er geen duidelijk scheiding tussen de refine en prototype fase.

Refine/prototype

De mogelijkheid om originele teksten, vertaalde teksten en omschrijvingen aan te passen vanuit de applicatie is toegevoegd. Aangepaste teksten worden groen gekleurd en pas aangepast in de database als de gebruiker dit bevestigt. Daarnaast is het ook mogelijk om te selecteren op datum van invoer en op ID van vertaling.

In de derde iteratie was pas bij de stagiaire bekend geworden van de alarm triggers vanuit de HMI niet vertaald mogen worden. Alarm triggers verwijzen naar een set van 32 alarmen in de HMI. Zodra alarm triggers vertaald worden, kan de vertaling van een alarm niet gevonden worden en zal het alarm dus in de originele taal weergegeven worden. Voor dit probleem is de simpele oplossing bedacht dat teksten die met “AT_” beginnen direct worden gekopieerd en niet worden vertaald.

Aan het eind van de iteratie is samen met de bedrijfsmentor een kleine test uitgevoerd om te kijken wat er gebeurt als de database door meerdere gebruikers tegelijkertijd wordt gebruikt. Tijdens deze testen bleef de applicatie de data elke keer goed afhandelen.

Review

Maandag 19 april heeft de review van de laatste iteratie plaatsgevonden met eindgebruiker en bedrijfsmentor L. Vlasman. De review was bedoeld om te kijken wat het uiteindelijke resultaat is geworden. Alle geplande werkzaamheden van de iteratie zijn behaald.

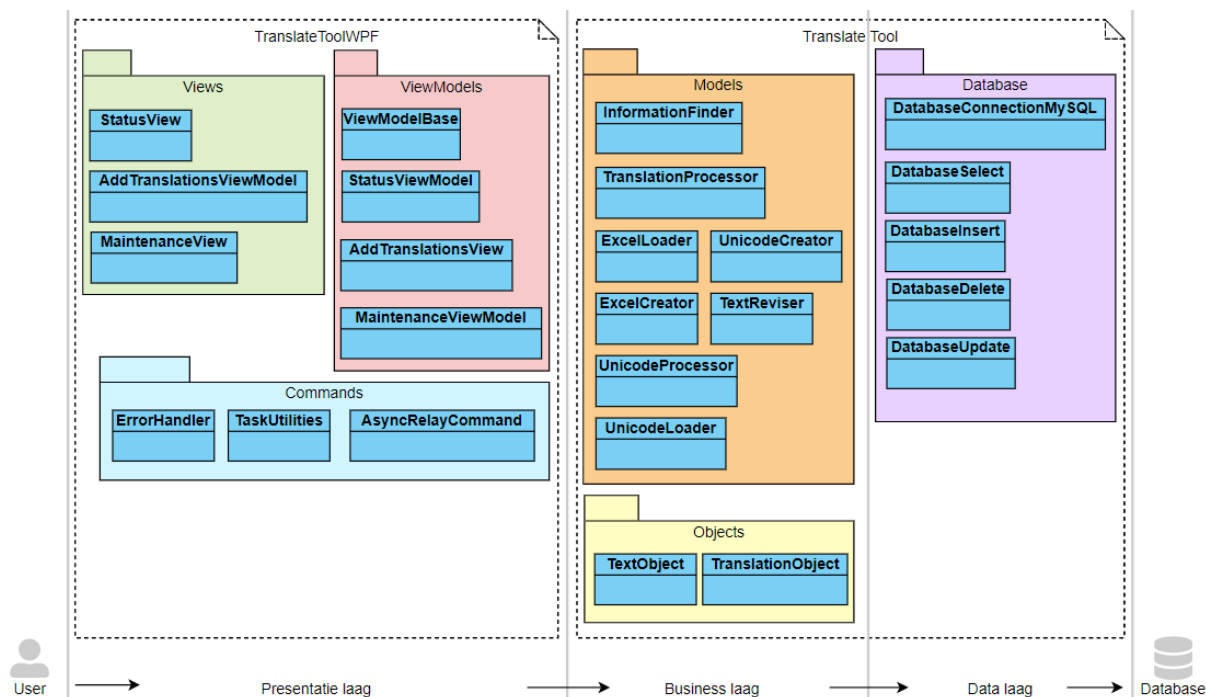
9. Rapid construction

In de fase “rapid construction” was het de bedoeling om alle gemaakte prototypes samen te voegen tot een eindproduct. Dit was voor de HMI-vertaaltool niet van toepassing, omdat incrementeel aan het prototype is gewerkt. Binnen deze fase was daardoor tijd om code te refactoren en een installatie van het programma te maken. Het refactoren bestond uit het herstructureren van de code voor het verbeteren van onderhoudbaarheid en leesbaarheid.

9.1 Refactoren

Tot aan het eind van de “user design” fases bestond de HMI-vertaaltool uit een project met hierin verschillende mappen. Voor het eindproduct wordt een verdeling tussen de presentatie-, business- en data-laag gemaakt. Tot de presentatie-laag behoort de MainWindow, Views, ViewModels en commands, omdat elk van deze onderdelen te maken heeft met de user interface. Bij de business-laag horen de Models. De klassen die queries opstellen en uitvoeren behoren tot de data-laag.

Figuur 17 geeft het ontwerp voor het herstructureren van de code weer. In het ontwerp is de presentatie, business en data-laag aangegeven. Tijdens het refactoren is een nieuw project aangemaakt voor de presentatie-laag, genaamd “TranslateToolWPF”. Het project “TranslateTool” met de overige klassen wordt als referentie bij de presentatie-laag geladen.



FIGUUR 17 ONTWERP SCHEIDING PRESENTATIE-, BUSINESS- EN DATA-LAAG

Door tijdsdruk binnen de “user design” iteraties is vervuiling in de code terecht gekomen. Tijdens het refactoren is de code in de juiste codestijl neergezet en eventuele vervuiling verwijderd.

9.2 Installatie

Om de HMI-vertaaltool te kunnen verspreiden is geprobeerd om een Windows Installer bestand met .msi extentie te maken. Dit gaf echter problemen met certificaten die goedgekeurd moesten worden door de gebruiker. Vervolgens is gezocht naar een tool om een .exe installatie te maken. De gratis tool Inno Setup van J. Russell (1997) bood hier een oplossing voor.

Met Inno Setup zijn twee installaties gemaakt. De eerste installatie (TranslateToolInstaller) is platform afhankelijk, waarbij de tool zonder .NET 5.0 referentie wordt geleverd. Deze installatie kan alleen gebruikt worden door gebruikers die .NET 5.0 op hun werkstation hebben geïnstalleerd.

De tweede installatie (TranslateToolInstallerDOTNET) is op zichzelf staand en kan door iedereen geïnstalleerd worden, ongeacht het besturingssysteem of platform van het werkstation. Het nadeel van deze installatie is de omvang. De op zich zelf staande installatie is meer dan 15 keer groter dan de platform afhankelijke installatie. Figuur 18 geeft de gegeneerde installaties weer.

⬇ TranslateToolInstaller.exe	7-5-2021 13:26	Application	5.108 KB
⬇ TranslateToolInstallerDOTNET.exe	7-5-2021 14:45	Application	85.368 KB

FIGUUR 18 INSTALLATIES VAN DE HMI-VERTAALTOOL

10. Cutover

In de fase “cutover” wordt de applicatie geaccepteerd en geïntegreerd binnen het bedrijf. Bij de HMI-vertaaltool bestaat dit uit het uitvoeren van acceptatietesten en het overdragen van de tool met documentatie over de installatie en het gebruik.

Door een ziekteperiode van bijna twee weken tegen het einde van de stage periode, zijn de werkzaamheden binnen de “cutover” fase beperkter uitgevoerd dan verwacht. Met name de documentatie van de tool richting het bedrijf

10.1 Acceptatietest

Om vertrouwen in het product te creëren bij de opdrachtgever en eindgebruikers wordt een acceptatietest gedaan. Tijdens de reviews in de “user design” fase zijn de functionaliteiten los van elkaar getest en gedemonstreerd aan de bedrijfsmentor en/of een gebruiker. Met de acceptatietest worden de use cases, die tijdens de “requirements analysis” zijn gedefinieerd, getest. Dit is gedaan door middel van black-box testen. Bij black-box testen wordt niet naar interne werking van het programma gekeken, maar naar de input en output. In Bijlage G “testrapport” zijn de acceptatietesten per use case uitgewerkt.

Binnen de acceptatietest moest de testpersoon een SlugClipper HMI van het Engels (en-GB) naar het Afrikaans (af-ZA) vertalen en vertalingen in de database aanpassen. Om dit te kunnen doen zijn er losse testen per use case geschreven.

Tijdens de “requirements analysis” zijn zes use cases gedefinieerd. Echter de use case voor het instellen van de applicatie is niet verwerkt in de HMI-vertaaltool. De volgende vijf use cases zijn wel geïmplementeerd en zijn door middel van een acceptatietest getoetst:

- Status weergeven
- Translate List ontvangen
- Database aanvullen
- Vertaling ontvangen
- Onderhoud database uitvoeren

In Tabel 4 is een voorbeeld gegeven van een acceptatietest. In deze test is het gelukt om de database aan te vullen met de benodigde Afrikaanse vertalingen.

TABEL 3 ACCEPTATIETEST DATABASE AANVULLEN

Test 3: Database aanvullen	Via de applicatie kan een gebruiker vertalingen aan de database toevoegen.
Precondities	De applicatie is opgestart, het “Home” scherm is te zien. Het werkstation is verbonden met het internet via het Houdijk netwerk. Er is een bestand aanwezig van de vertaler in kolom A de originele teksten, in kolom B de vertalingen en in kolom C eventueel omschrijvingen van de originele tekst.
Uitvoering test:	<ul style="list-style-type: none">• Navigeer naar het “Add translations” scherm door op het database icoon aan de linker zijkant te klikken.• Selecteer de taal “en-GB” bij “From: “.• Selecteer de taal “af-ZA” bij “To: “.• Druk op de knop “Upload translations”.

	<ul style="list-style-type: none"> - Er verschijnt een venster. • Navigeer in het venster naar het bestand van de vertaler. • Selecteer het bestand van de vertaler. • Druk op de knop “Open” rechtsonder in het venster. - De knop “Process translations” kleurt donker binnen enkele seconden. - Bij “Total translations found” staat hoeveel vertalingen gevonden zijn in het bestand. • Druk op de knop “Process translations”. <ul style="list-style-type: none"> - Als de progressie balk vol is, staat bij “Translations added to database” hoeveel nieuwe vertalingen zijn toegevoegd. • Klik op het witte power icoon rechtsboven in de hoek.
Postcondities:	De applicatie is afgesloten.
Conclusie:	Behaald

Op 7 mei vond de acceptatietest met de bedrijfsmentor plaats. De testen zijn uitgevoerd op haar werkstation terwijl FactoryTalk View verbonden was met een controller waar het SlugClipper PLC-programma op draaide. In Figuur 19 is een foto te zien van de testopstelling tijdens het uitvoeren van de acceptatietest. Alle testen zijn zonder problemen behaald.



FIGUUR 19 TEST OPSTELLING ACCEPTATIETEST

10.2 Gebruikershandleiding

Om de vertaaltool goed te kunnen overdragen aan Houdijk Holland is een gebruikershandleiding geschreven. De handleiding is geschreven volgens het “How to?” format van Houdijk Holland. In dit format wordt eerst een kort stappenplan per onderdeel gegeven. Vervolgens wordt elke stap uitgewerkt en mogelijk aangevuld met visuele ondersteuning. Ervaren software engineers kunnen het eerste deel als geheugensteun gebruiken. Minder ervaren software engineers kunnen in de gedetailleerde uitwerking kijken voor meer uitleg per stap en eventueel extra informatie over het onderwerp.

De onderwerpen die zijn besproken:

- A. HMI vertaaltool installeren
- B. Status en translate list verkrijgen
- C. Creëren importbestand voor HMI
- D. Vertalingen toevoegen aan database
- E. Onderhoud aan database

In Bijlage H “gebruikershandleiding” is deze handleiding te vinden.

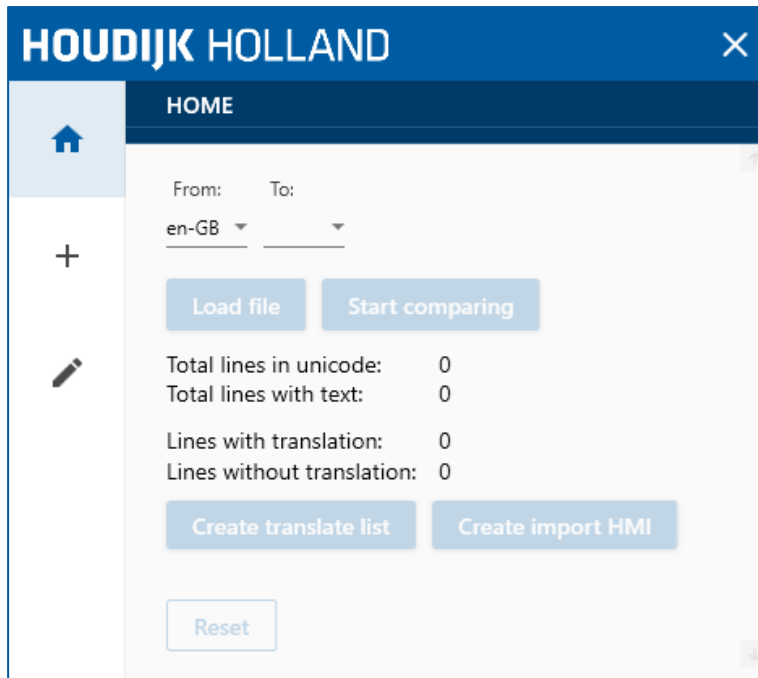
10.3 BPMN model nieuwe situatie

Aan het begin van de stage is een BPMN-model gemaakt van de huidige situatie. Dit model beschreef het proces voor het vertalen van een HMI met het gebruik van de Exceltool. Dit proces wordt door de meeste software engineers als vervelend ervaren. Na het ontwikkelen van de HMI-vertaaltool binnen deze stage is een BPMN-model gemaakt voor de nieuwe situatie. In Bijlage I “BPMN-model nieuwe situatie” is dit BPMN-model weergegeven.

Het BPMN-model bevat het hoofdproces “HMI vertalen” en drie kleine sub-processen “translate list genereren”, “ontvangen vertalingen en verwerken in database” en “importbestand voor FactoryTalk View maken”. Alleen voor het uitvoeren van de sub-processen wordt de HMI-vertaaltool gebruikt.

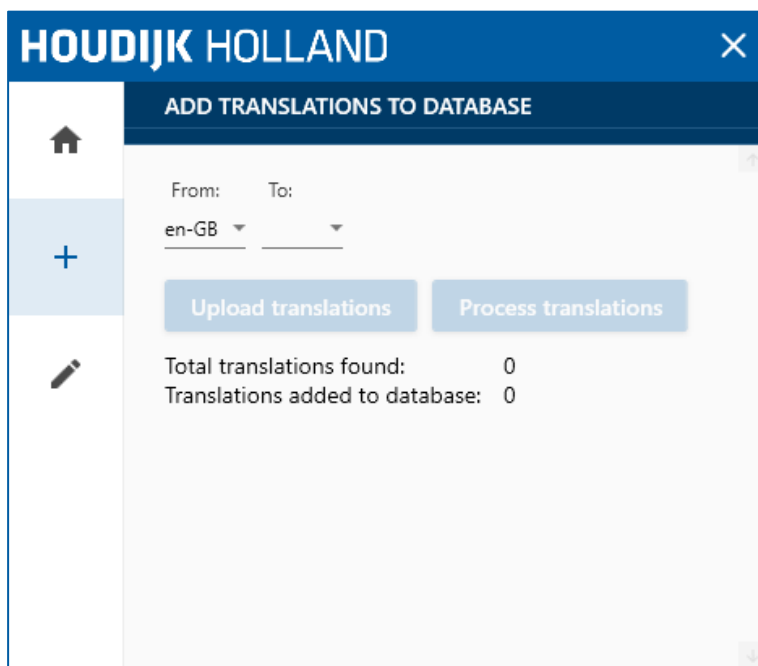
11. Resultaat eindproduct

In dit hoofdstuk worden de drie verschillende schermen van het eindproduct weergegeven. Figuur 20 geeft het begin scherm zien weer, waar een translate list en een importbestand voor FactoryTalk View gemaakt kan worden en waar de status afgelezen kan worden.



FIGUUR 20 RESULTAAT COMPARETEXTVIEW

In Figuur 21 is het scherm voor het toevoegen van vertalingen aan de database weergegeven. De vertalingen kunnen door de tool vanuit een Excelbestand aan de database toegevoegd worden.



FIGUUR 21 RESULTAAT ADDTRANSLATIONSVIEW

In Figuur 22 is het scherm voor onderhoud aan de database weergegeven. Dit scherm biedt de gebruiker de mogelijkheid om vertalingen in te zien, aan te passen en te verwijderen. De gebruiker heeft verschillende opties om te filteren.

<input type="checkbox"/>	ID	Date	Original	Original text
<input type="checkbox"/>	3	25/05/2021	en-GB	ERROR
<input type="checkbox"/>	4	25/05/2021	en-GB	User Operator

FIGUUR 22 RESULTAAT MAINTENANCEVIEW

Met de HMI-vertaaltool kunnen vertalingen worden opgeslagen in een database. Hierdoor worden reeds vertaalde teksten niet nog een keer naar een vertaalbureau gestuurd. Uiteindelijk zorgt dit voor kostenbesparing.

Door de HMI-vertaaltool te gebruiken kan een software engineer doorgaan met de ontwikkeling van de HMI, terwijl het vertaalbureau bezig is met vertalen. Dit is mogelijk omdat de tool teksten zonder vertaling filtert en in een translate list plaatst. Deze laatst aangepaste teksten kunnen dan met bijvoorbeeld Google Translate vertaald worden.

De reacties van software engineers die de tool hebben gezien, waren positief. Met name de lay-out van de applicatie kreeg veel complimenten.

12. Beroepstaken

Voor de afstudeerstage zijn de volgende beroepstaken uitgevoerd:

- A1 Analyseren probleemdomein.

Het probleem domein is geanalyseerd binnen het plan van aanpak (Bijlage B). Het BPMN-model van de huidige situatie geeft aan hoe een HMI voorheen vertaald werd. Dit proces is in Bijlage C “BPMN-model huidige situatie” toegelicht.

- C1 Ontwerpen van software

Aan de hand van de sequentiediagrammen, het klassendiagram en pseudocode is de HMI-vertaaltool ontworpen. In de tweede iteratie van de “user design” fase is een klassendiagram gemaakt. In de daaropvolgende iteraties zijn dit klassendiagram aangevuld. Voor ieder scherm binnen de tool is een eigen sequentiediagram gemaakt.

Tijdens de “rapid construction” fase is het ontwerp herzien en heeft er een kleine herstructurering plaatsgevonden. Het ontwerp voor de herstructurering is in hoofdstuk 9.1 weergegeven.

- C6 Ontwerpen van Human Computer Interaction

In de derde iteratie van de “user design” fase is de user interface ontworpen. In hoofdstuk 8.3 is een foto van het ontwerp afgebeeld. Als ontwerp is een simpele en moderne user interface gemaakt, in de stijl van Houdijk Holland.

- D2 Testen & evalueren

De opgeleverde HMI-vertaaltool is door middel van een acceptatietest op een SlugClipper project van Houdijk Holland getest. De acceptatietest is behaald en de resultaten zijn beschreven in Bijlage G “testrapport”.

- Gc Kritisch en methodisch werken

Tijdens de stage heb ik mijn eigen voortgang kunnen bewaken, planning kunnen bijstellen en proactief kunnen reageren op veranderingen. De ontwikkeling van de tool is volgens de RAD-methode verlopen, waarbij in de requirements analyse gebruik is gemaakt van de story mapping methode. In hoofdstuk 4 staat de aanpak van de opdracht beschreven. In de proces evaluatie zijn de bevindingen van de gebruikte methodes beschreven.

- Gf Leren leren

Tijdens deze stage heb ik het gehele proces van softwareontwikkeling in een professionele situatie kunnen ervaren. Hierbij heb ik voor het eerst gewerkt met C#, WPF en MySQL. Door de ervaring en de vaardigheden die ik tijdens mijn opleiding heb geleerd, heb ik het zelfvertrouwen om een nieuw taal of techniek te leren.

13. Product evaluatie

Binnen dit hoofdstuk wordt de HMI-vertaaltool geëvalueerd, het eindproduct en het doel van de stageopdracht. Onderdeel van de HMI-vertaaltool is het programma, een testrapport, een gebruikershandleiding en een BPMN-model.

HMI-VERTAALTOOL

De tool voldoet aan alle noodzakelijke systeemeisen. Waardoor de kans groot is dat het bedrijf de tool in gebruik gaat nemen.

De lay-out van de tool was belangrijk bij de acceptatie van de tool door de afdeling. Er werd vanuit Houdijk Holland een simpele en moderne tool gevraagd, wat ook geleverd is. De kleuren sluiten aan op de huisstijl van Houdijk Holland.

De programmacode van de HMI-vertaaltool is ontworpen aan de hand van UML-diagrammen. De ontwerpen geven een duidelijk beeld van het MVVM software ontwerppatroon. Dit ontwerppatroon zorgt voor leesbaarheid, herbruikbaarheid en onderhoudbaarheid.

Ik had nog het Entity Framework aan de HMI-vertaaltool willen toevoegen. Dit framework zorgt voor een eenvoudige koppeling tussen de applicatie en de database. Doordat in de analyse geen aandacht is besteed aan een efficiënte koppeling tussen de applicatie en database, is bij het programmeren voor de snelle oplossing gekozen. Achteraf denk ik dat deze snelle oplossing een zwakke schakel van het eindproduct is geworden.

MYSQL

De vertalingen worden opgeslagen in een MySQL database. De datastructuur bestaat maar uit drie tabellen en is niet ingewikkeld. Mogelijk kan in de toekomst nog gebruikersgegevens worden opgeslagen in een vierde tabel. Door in de tabel “translations” de taal van de teksten te verwijderen kan de database nog geoptimaliseerd worden. Op dit moment kunnen teksten meerdere vertalingen toegewezen krijgen, de oudste vertaling wordt dan niet meer gebruikt maar blijft wel in de database aanwezig.

TESTRAPPORT

Zoals in het afstudeerplan beschreven staat is een acceptatie test uitgevoerd. Deze testen zijn duidelijk beschreven en tonen de afgesproken functionaliteiten. Eventueel hadden nog unittesten voor de software of usability testen voor de user interface uitgevoerd kunnen worden. Door tijdsgebrek is dit niet gelukt.

GEBRUIKERSHANDLEIDING

De gebruikershandleiding is kort en duidelijk geworden in een herkenbare stijl voor de Software engineers. Het “How to?” format sluit goed aan op de afdeling, omdat dit vaker wordt gebruikt. Naast een gebruikershandleiding was het plan om systeemdokumentatie te maken voor het bedrijf. Door tijdsgebrek is mondeling uitleg gegeven over de programmacode aan de bedrijfsmentor in plaats van op papier.

BMPN-MODEL

Als laatste is voor de HMI-vertaaltool een BPMN-model gemaakt. Het model is groot, maar biedt wel inzicht in het proces. Voor een software engineer die ervaring heeft met FactoryTalk View, moet het proces gemakkelijk te volgen zijn.

14. Proces evaluatie

In dit hoofdstuk wordt geëvalueerd op het doorgelopen proces binnen de stage. Binnen de evaluatie wordt beoordeeld wat goed is gegaan en wat beter had gekund.

Bij het ontwikkelen van de HMI-vertaaltool is de RAD-methode gebruikt. Dit was een methode waar ik nog geen ervaring mee had. Vooraf wist ik dat mijn ervaring als softwareontwikkelaar niet voldoende zou zijn om vlot prototypes te kunnen ontwikkelen. Dit bleek zeker het geval toen ik besloot om in C# te programmeren, een techniek waar ik nauwelijks ervaring mee heb. Binnen de stage ben ik hier bewust mee omgegaan door een extra analyse in te plannen, wat resulteerde in vlotte iteraties.

Ik ben blij met de keuze voor de RAD-methode. De goede voorbereiding en een snelle ontwikkeling van de tool was iets wat goed bij mezelf en de opdracht paste. Helaas was bij de invulling van de RAD-methode niet meer aandacht is besteed aan het uitgebreid testen van de software. Dit is een onderdeel waar ik mezelf graag verder in zou willen verdiepen.

ANALYSE VAN TECHNOLOGIEËN

Het omzetten van de onderzoeksfase voorafgaand aan de RAD-methode in een analyse van technologieën, heeft een positieve wending aan de stage gegeven. Tijdens het onderzoek liep ik vast bij het formuleren van een onderzoeksvraag. Na enige tijd kwam ik erachter dat een onderzoek niet de juiste aanpak was om mezelf te verdiepen in de benodigde technieken. Graag had ik gewild dat ik hier eerder achter was gekomen.

REQUIREMENTS ANALYSIS

Binnen de requirements analyse is gebruikt gemaakt van de story mapping methode. De story map heb ik vervolgens als leidraad gebruikt voor de “user design” iteraties. Deze manier van systeemeisen documenteren paste bij de RAD-methode, omdat systeemeisen gemakkelijk tussen iteraties verschoven konden worden. Tijdens het verdelen van de eisen over de iteratie had ik geen rekening gehouden met de aandachtspunten vanuit de reviews met een eindgebruiker. Na de eerste twee iteraties zorgde dit voor vele systeemeisen waar geen structuur in te vinden was. In het vervolg zou ik de iteratie minder vol plannen, om meer rust te kunnen bewaren.

USER DESIGN

De user design fase is snel verlopen. Tijdens deze iteraties is in hoog tempo een applicatie neergezet. Tijdens deze drukte heb ik de voortgang bewaakt en actief gereageerd op veranderingen of tegenslagen. De codereviews met C# programmeur R. Hoogeboom hielpen met verbeteren van de tool en deze naar een hoger niveau te tillen, door functies asynchroon te maken en MVVM-Light toe te passen.

RAPID CONSTRUCTION

De “rapid construction” fase heb ik op een andere manier ingevuld dan de RAD-methode beschrijft. In plaats van verschillende prototypes samen te voegen tot een eindproduct, heb ik het laatste prototype geherstructureerd. Door hier veel tijd aan te besteden kon de HMI-vertaaltool op een nette manier afgeleverd worden aan het bedrijf.

CUTOVER

Tijdens de “Cutover” is een acceptatietest uitgevoerd, een gebruikershandleiding gemaakt en een BPMN-model van het nieuwe proces uitgewerkt. Aan het begin van deze fase kreeg ik het COVID-19 virus. De ziekte en nasleep had ik onderschat. Uiteindelijk heb ik in twee weken nauwelijks voortgang kunnen maken. Hierdoor verliep deze fase rommelig en heb ik minder tijd kunnen besteden aan de werkzaamheden dan gehoopt. Ondanks dat het niet in mijn afstudeerplan stond, had ik graag ook nog een usability test willen uitvoeren.

In overleg met bedrijfsmentor is een plan gemaakt voor de laatste weken. Door dit plan en duidelijke afspraken is het gelukt om zonder vertraging de stage te kunnen afronden.

15. Reflectie

In deze reflectie wil ik aangeven wat mijn bevindingen zijn van de stage en de opdracht. Ik kijk met trots terug op het eindproduct. De afgelopen weken heb ik veel werk kunnen verrichten en is de tijd omgevlogen. Ik heb een werkende tool aan Houdijk Holland af kunnen leveren en ik acht de kans groot dat ze deze in gebruik gaan nemen.

Voorafgaand aan de stage wist ik dat op tijd vragen stellen of om hulp vragen een uitdaging voor mij zou gaan worden. Mijn bedrijfsmentor was altijd behulpzaam en maakte tijd voor mij vrij. Nadat ik dit inzag, was de drempel om vragen te stellen een stuk lager geworden. Halverwege de derde “user design” iteratie besloot ik om software engineer R. Hoozeboom om hulp te vragen voor de codestijl van het C# programma. De vraag die ik toen van hem kreeg was: “Waarom heb je me niet eerder gevraagd?”. Na dit gesprek wist ik in welke richting ik de applicatie wilde sturen en had ik meer vertrouwen in mijn eigen werk gekregen. Ik vind het nog steeds lastig om hulp te vragen, maar dit gaat steeds beter.

Achteraf had ik graag wat vaker door de werkplaats heen gelopen. Houdijk Holland maakt hoogstaande machines met fascinerende technieken. Nu ben ik vrijwel de hele stage op kantoor geweest en maar een enkele keer in de werkplaats, terwijl daar juist hele interessante machines staan. Mocht ik ooit nog een stage lopen dan zal ik hier meer de tijd voor nemen.

De machinebouw vind ik nog steeds een boeiend vakgebied. Ik acht de kans groot dat ik in deze wereld terecht kom, zodra ik ga starten met werken.

Lijst met afkortingen

In de onderstaande tabel worden de afkortingen die in dit document zijn gebruikt weergegeven. De definitie van de afkorting is daarnaast genoteerd.

TABEL 4 LIJST MET AFKORTINGEN

BPMN	Business Process Model Notation	Een manier van een proces noteren.
HMI	Human Machine Interface	Schermen waar informatie over een machine of systeem wordt gegeven aan operators.
GPL	GNU Library General Public License	Een software licentie waarbij de gebruiker alles met de software mag en de ontwikkelaar geen verantwoordelijkheid draagt.
MVC	Model, View en Controller	Een software ontwerppatroon die scheiding van verantwoordelijkheid creëert tussen drie onderdelen.
MVVM	Model, View en ViewModel	Een software ontwerppatroon voor WPF applicaties die leesbaarheid en herbruikbaarheid bevordert.
PLC	Programmable Logic Controller	Een elektronisch apparaat met microprocessor die in- en uitgangen aanstuurt.
RAD	Rapid Application Development	Een ontwikkelingsmethode voornamelijk gebruikt bij software engineering.
RUP	Rational Unified Process	Een ontwikkelingsmethode voornamelijk gebruikt bij software engineering.
UML	Unified Model Language	Een modelleertaal om objectgeoriënteerd informatiesystemen te ontwerpen.
WPF	Windows Presentation Forms	Een extensie van het .NET raamwerk om eenvoudig user interfaces te maken en data hieraan te verbinden.
XP	eXtreme Programming:	Een ontwikkelingsmethode voornamelijk gebruikt bij software engineering.

Literatuur

- Balaji, S., & Sundararajan Murugaiyan, M. (2012). WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC. *International Journal of Information Technology and Business Management*, 2(1), 26–30.
<https://mediaweb.saintleo.edu/Courses/COM430/M2Readings/WATEERFALLVs%20V-MODEL%20Vs%20AGILE%20A%20COMPARATIVE%20STUDY%20ON%20SDLC.pdf>
- Beck, K., & Andres, C. (2004). *Extreme Programming Explained* (2nd edition). Pearson Education (Us).
- Bugnion, L. (2018). MVVM-Light (5.4.1.1) [Toolkit voor MVVM]. GalaSoft. <http://www.mvvmlight.net/>
- Cleary, S. (2014). *Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming* (1ste ed.). O'Reilly Media.
- Coleman, G. (1998). A quality software process for rapid application development. In R. Verbruggen (Red.), *Software Quality Journal* (7de editie, Vol. 1998, pp. 107–122). Kluwer Academic Publisher. <https://doi.org/10.1023/A:1008856624790>
- Geambasu, C. V. (2011). Influence factors for the choice of a software development methodology. In I. Juanu, I. Juanu, & A. Gavrilă (Reds.), *Accounting and Management Information Systems* (4de editie, Vol. 10, pp. 479–494). <https://core.ac.uk/download/pdf/6261795.pdf>
- Källman, J. (2020). EPPLUS (4.5.3.3) [Package for advanced Excel spreadsheets using .NET, without the need of interop]. NuGet. <https://github.com/JanKallman/EPPlus>
- Kissflow. (2020, 26 oktober). Rapid Application Development [Illustratie]. Rapid Application Development (RAD): Changing How Developers Work. <https://kissflow.com/wp-content/uploads/2018/07/Rapid-application-development.png>
- Kroll, P., Kruchten, P., & Booch, G. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP: A Practitioner's Guide to the RUP* (1ste ed.). Addison-Wesley Professional.
- Lander, R. (2020, 10 november). Announcing .NET 5.0. Microsoft .NET Blog. <https://devblogs.microsoft.com/dotnet/announcing-net-5-0/>
- McKenna, D. (2016). *The Art of Scrum* (Vol. 2016). Apress.
- Microsoft. (2021, 3 februari). Choose your Windows app platform - Windows applications. Microsoft Docs. <https://docs.microsoft.com/en-us/windows/apps/desktop/choose-your-platform>

MySQL. (z.d.). What is MySQL? Geraadpleegd op 13 maart 2021, van <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

MySql.Data (8.0.24). (2021). [MySqlClient .Net Core Class Library]. Oracle. <https://dev.mysql.com/>

Patton, J., & Economy, P. (2014). User Story Mapping: Discover the Whole Story, Build the Right Product (1ste editie). Van Duuren Media. <https://doi.org/10.5555/2688795>

Russell, J., & Laan, M. (1997). Inno Setup Installer (6.1.2) [Open source installer]. <https://jrsoftware.org/isinfo.php>

Sørensen, E., & Mihailescu, M. I. (2010). Model-View-ViewModel (MVVM) Design Pattern using Windows Presentation Foundation (WPF) Technology. MegaByte, 9, 1–4. http://megabyte.utm.ro/articole/2010/info/sem1/InfoStraini_Pdf/1.pdf

What is .NET? (z.d.). Microsoft .NET. Geraadpleegd op 11 maart 2021, van <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>

Willock, J. (2021). MaterialDesignThemes (4.0.0) [Templates voor WPF controls in .NET.]. Google. <https://material.io/design>

Bijlagen

Bijlage A Afstudeerplan

Bijlage B Plan van Aanpak

Bijlage C BPMN-model huidige situatie

Bijlage D Analyse technologieën

Bijlage E Requirements analysis

Bijlage F User design iteratie 1 t/m 6

Bijlage G Testrapport

Bijlage H Gebruikshandleiding

Bijlage I BPMN-model nieuwe situatie