

Scriptie v2.0

Onderzoek naar de voor- en nadelen van het gebruik van low-code/no-code in projecten bij Netcompany.

Datum van publicatie: 04-06-2021

Afstudeergegevens	
Student	Patricia Marie Moscoso
Studentnummer	
e-mail	
Onderwijsinstelling	De Haagse Hogeschool Johanna Westerdijkplein 75 2521 EN, Den Haag
Opleiding	HBO-ICT deeltijd Software Engineering
Bedrijf	Netcompany
Adres	Kanaalweg 3b 2628EB Delft
Opdrachtgever	Ing. Azad Imamoglu
Functie	Managing Architect
Bedrijfsbegeleider	Ing. Azad Imamoglu
Begeleidende examiner	Dr. Peter Becker
Expert examiner	Ir. Job Habraken

Versiebeheer

Versienummer	Datum	Opmerkingen
1.0	30-04-2021	Versie beoordeeld tijdens het tussentijdse assesment en feedback verwerkt.
2.0	04-06-2021	Laatste versie afgerond en voorbereid voor inlevering

Voorwoord

Voor u ligt de scriptie 'Onderzoek naar de voor- en nadelen van low-code in projecten bij Netcompany'.

Deze scriptie is in het kader van mijn afstuderen aan de opleiding HBO ICT Deeltijd aan de Haagse Hogeschool Den Haag en in opdracht van mijn bedrijf Netcompany. Van februari 2021 tot en met juni 2021 ben ik bezig geweest met het onderzoek en het schrijven van de scriptie.

Samen met mijn bedrijfsbegeleider, ing. Azad Imamoglu heb ik de onderzoeksvraag voor deze scriptie bedacht. Na uitvoerig kwalitatief onderzoek heb ik de onderzoeksvraag kunnen beantwoorden. Tijdens mijn traject kon ik altijd terecht bij mijn bedrijfsbegeleider Azad Imamoglu en mijn begeleider vanuit mijn opleiding, dr. Peter Becker.

Bij dezen wil ik graag mijn begeleiders bedanken voor hun ondersteuning tijdens dit traject. Naast hen wil ik ook mijn collega's, de low-code leveranciers en de ontwikkelaars die ik mocht interviewen bedanken. Zonder hen had ik dit onderzoek nooit zo gedetailleerd kunnen uitwerken.

Robin Bastiaan mijn klasgenoot dank je dat je elke vrijdagochtend een half uur tijd had om te sparren. Dit heeft mij enorm geholpen!

Ook wil ik mijn familie en vrienden bedanken voor hun steun. Expliciet wil ik mijn lieve ate (zus in het filipijs) Ravenna Moscoso en haar vriend Mark van Reusel bedanken voor het ondersteunen tijdens mijn schrijfproces. Tot slot wil ik mijn beste vriend en partner Maarten Griffioen bedanken. Zijn coaching tijdens mijn hele schoolperiode heeft mij geholpen om deze scriptie tot een goed einde te brengen.

Patricia Marie Moscoso

Pijnacker, 04 juli 2021

Samenvatting

De ontwikkeling van software met traditioneel development is specialistisch werk en neemt veel tijd in beslag. In de commerciële wereld is het daarom erg interessant om een opdracht sneller en goedkoper op te kunnen leveren. Een manier om dit te doen, is low code. Low code is een vorm van software development waarmee visueel applicaties kunnen worden ontwikkeld. Er is echter te weinig kennis van low-Code binnen Netcompany om te onderbouwen waarom dit wel of niet zou werken bij projecten voor Netcompany.

In deze scriptie staat de daarom de volgende vraag centraal: “Kan door middel van een low-code platform sneller en efficiënter software worden ontwikkeld zonder kwaliteitsverlies ten opzichte van traditioneel development en welk low-code platform past het beste bij Netcompany?”

Om een antwoord te kunnen geven op de hoofdvraag heb ik mijn onderzoek in tweeën verdeeld. Het eerste gedeelte is het zoeken naar het geschikte platform voor Netcompany. Hiervoor is eerst een literatuuronderzoek gedaan, waarin generieke eisen worden opgesteld waaraan een moderne applicatie moet voldoen. Vervolgens is deze lijst uitgebreid met specifieke wensen en eisen van Netcompany zelf. Deze specifieke eisen zijn voortgekomen uit interviews met een software architect van Netcompany, en een interview met de sales afdeling. Aan de hand van deze lijst is er met behulp van verschillende beslissingsmethodieken gekeken welk low-code platform het beste aan deze eisen voldoet. Uit het onderzoek kwam het platform Mendix als beste kandidaat naar voren.

Het tweede gedeelte van mijn onderzoek richt zich op de vraag of met low-code sneller en efficiënter ontwikkeld kan worden zonder kwaliteitsverlies. Er is een applicatie, die ontwikkeld is met traditioneel development, nagebouwd in Mendix om de ontwikkeltijd te vergelijken. De applicatie ontwikkeld met low-code heeft in totaal 32 uur gekost, terwijl de traditionele respectievelijk 160 uur nodig had. Hieruit blijkt dat deze applicatie sneller en efficiënter ontwikkeld kan worden met low-code. Daarnaast is er een literatuuronderzoek gedaan naar de kwaliteit van low-code. Hierin is ingegaan op performance, testen van op maat gemaakte componenten, monitoring, en vendor lockin. Uit dit onderzoek is gebleken dat Mendix vergelijkbare kwaliteit levert in vergelijking met traditioneel development.

Uit dit kwalitatieve onderzoek is gebleken dat low-code een oplossing zou kunnen zijn, omdat projecten potentieel sneller ontwikkeld kunnen worden, zonder kwaliteitsverlies. Er is echter wel verder onderzoek nodig, wanneer er projecten worden ontwikkeld met low-code, van hoge complexiteit. In deze projecten zijn er veel opmaat gemaakte componenten nodig en is het de vraag of ontwikkel-snelheid en ontwikkel-complexiteit lager zijn bij low-code of bij traditioneel development.

Inhoud

1. Inleiding	1
2. Organisatieomschrijving	2
3. Opdrachtomschrijving	3
3.1. Projectomgeving	3
3.2. Doelstelling project	3
3.3. Opdrachtformulering	3
3.3.1. Projectopdracht	3
3.3.2. Opgeleverde producten en diensten	3
4. Aanpak	4
4.1. Hoe	4
4.2. Methodes	4
5. Keuze low-codeplatform	6
5.1. Soort onderzoek	6
5.2. Literatuuronderzoek	6
5.2.1. Zoekplan	6
5.2.2. Keuze uit de gevonden artikelen en samenvatten van artikelen	7
5.2.3. Model-driven development en low-code	7
5.2.4. Generieke lijst met eisen	8
5.3. Gekozen beslissingsmethodieken	9
5.3.1. Beslissingsmodel	9
5.3.2. Beslissingsboom	9
5.3.3. SFA-matrix	11
5.4. Eisen van Netcompany voor software	12
5.4.1. Eisen uit interviews/ prioriteren eisen	12
5.4.2. Low-codeplatform voorkeuren	13
5.4.3. Selectie pakket eisen	13
5.5. Invullen beslissingsmethodieken	15
5.5.1. Beslissingsmodel (Farshidi, Jansen, & Fortuin, 2021)	15
5.5.2. Beslissingsboom (Sattar, 2018)	15
5.5.3. SFA-matrix	16
5.6. Conclusie keuze low-codeplatform	17
5.6.1. Mendix	18
5.6.2. Bijdrage Mendix	18
5.7. Verantwoording en reflectie keuze van low-codeplatform	18
5.7.1. Onderzoek	18

5.7.2.	Gekozen beslissingsmethodieken.....	19
5.7.3.	Opstellen van het adviesrapport	19
6.	Low-code ontwikkelsnelheid en kwaliteit	20
6.1.	Soort onderzoek.....	20
6.2.	Proof of Concept.....	20
6.2.1.	Requirements en omschrijving van Proof of Concept.....	21
6.2.2.	Bouwen van de proof of concept	21
6.2.3.	Tijdsverschil tussen beide Proof of Concepts	24
6.2.4.	Discussie Proof of Concept.....	24
6.3.	Interviews met ervaren ontwikkelaars	24
6.4.	Literatuuronderzoek naar bekende problemen van low-code	25
6.5.	Roadmap.....	27
6.6.	Verantwoording en reflectie ontwikkelsnelheid en kwaliteit	28
7.	Conclusie & discussie	29
8.	Evaluatie	30
8.1.	Product reflectie	30
8.2.	Verantwoording beroepstaken	31
8.2.1.	A1 Analyseren probleemdomain & opstellen probleemstelling	31
8.2.2.	A2 Informatie vergaren, analyseren, beoordelen en verwerken	31
8.2.3.	B1: Gemotiveerd selecteren van ICT gerelateerde oplossingen	32
8.2.4.	B3 Adviseren over inrichting van ICT gerelateerde oplossingen & processen	32
8.2.5.	C1 Ontwerpen software	32
8.2.6.	D1 Realiseren van software.....	32
8.2.7.	Gc Kritisch, onderzoekend & methodisch werken	33
8.2.8.	Gf leren: voorbereiden op de volgende studiefase & beroep.....	33
	Bibliography.....	34
	Bijlagen.....	i
I.	Afstudeerplan	i
II.	Plan van Aanpak.....	ii
III.	Zoekplan	iii
IV.	Samenvatting literatuuronderzoek	iv
V.	Productowner generieke lijst	v
VI.	Selectie pakket Netcompany.....	vi
VII.	Rapport beslissingsmodel (Farshidi, Jansen, & Fortuin, 2021)	vii
VIII.	Scorematrix.....	viii
IX.	Voor- en nadelen van low-codeplatformen	ix

X.	Adviesrapport: onderzoek low-codeplatform die het beste bij Netcompany past	xi
XI.	Requirements Proof of Concept	xii
XII.	Uren van traditionele development PoC	xiii
XIII.	Snapshots traditionele PoC	xiv
XIV.	Snapshots Mendix PoC.....	xvi
XV.	Tijdsbesteding Low-code en Traditioneel development.....	xvii
XVI.	Expert interviews met ontwikkelaars	xviii
XVII.	Gesprekken met leverancier Mendix	xix
XVIII.	Certificaat Udemy	xx
XIX.	Domein model	xxi
XX.	Voortgangsrapportage maand februari.....	xxii

1. Inleiding

De relatief snelle groei van het aantal vacatures én de technologische ontwikkelingen in de ICT veroorzaken een grote druk op deze specialistische arbeidsmarkt. De vraag naar goed geschoold personeel was al groot maar is nu zeer groot. Niet alleen in de sector zelf, maar in elke branche zijn IT-professionals nodig. Zo hebben de grotere werkgevers allemaal een ICT-servicedesk en deze moet bemand worden met deskundig personeel (Economische Verkenning Rotterdam, 2020). Ook zijn er veel bedrijven die applicaties uitbesteden aan externe partijen. Deze business consultancy bedrijven proberen zo goedkoop mogelijk dit soort services aan te bieden. Doordat traditioneel development veel tijd en specialisme vereist zou low-code een oplossing kunnen zijn om projecten goedkoper aan te bieden.

“Low-code is een vorm van softwareontwikkeling die is gericht op het visueel ontwerpen van applicaties. Bij low code maakt men gebruik van een grafische gebruikersomgeving en het instellen van configuraties in plaats van het schrijven van traditionele computercode.” (Low-code, 2020). Low-code ontwikkeling is het grafisch ontwikkelen van software met in het proces zo min mogelijk code schrijven en het hergebruiken van componenten. Met low-code wordt er gebruik gemaakt van Business Process Management Suite. Dit betekent dat in het platform een Business Proces Model Notation gerealiseerd wordt, een Entity-Relation Diagram en de business regels. Dit allemaal is visueel. Op deze manier kan ongeveer 80% van de applicatieontwikkeling worden gedaan door een junior programmeur of een bedrijfsanalist. De overige 20% betreft proces- en gegevensarchitectuur, specifieke gegevensverwerking en geavanceerde visuele controles waarvoor ervaren professionals nodig zou zijn. (Rossi, 2015)

Low-code wordt gevraagd bij huidige aanbestedingen. Zo is gebleken uit het interview met de Sales afdeling dat er een opkomst is bij toekomstige klanten die specifiek om low-code vragen. Dit roept de vraag op of de organisatie hiervan gebruik kan maken om haar ontwikkelactiviteiten te versterken. Er is echter te weinig kennis van low-Code om te onderbouwen waarom dit wel/niet zou werken bij projecten. Het gewenste resultaat is dat Netcompany weet of de low-code oplossingen ingezet kan worden bij projecten. Netcompany kan daarbij haar afwegingen baseren op de resultaten van het onderzoeksrapport. De hoofdonderzoeksvraag luidt:

“Kan door middel van een low-code platform sneller en efficiënter software worden ontwikkeld zonder kwaliteitsverlies ten opzichte van traditioneel development en welk low-code platform past het beste bij Netcompany?”

In hoofdstuk 2 beschrijf ik hoe de organisatie van Netcompany eruitziet en voor welke sectoren zij werk verrichten.

In hoofdstuk 3 zet ik de opdrachtomschrijving uiteen.

In hoofdstuk 4 vertel ik de aanpak van het onderzoek en beschrijf ik de methodes die ik heb gebruikt.

In hoofdstuk 5 wordt beschreven hoe ik een geschikt low-code platform voor Netcompany heb uitgekozen.

In hoofdstuk 6 ga ik in op de ontwikkelnelheid van low-code en hoe deze vergelijkt met traditioneel development. Daarnaast wordt in dit hoofdstuk een literatuuronderzoek beschreven over de kwaliteit.

In hoofdstuk 7 behandel de conclusie en discussie van het onderzoek.

In hoofdstuk 8 evalueer ik hoe het onderzoek was gegaan.

2. Organisatieomschrijving

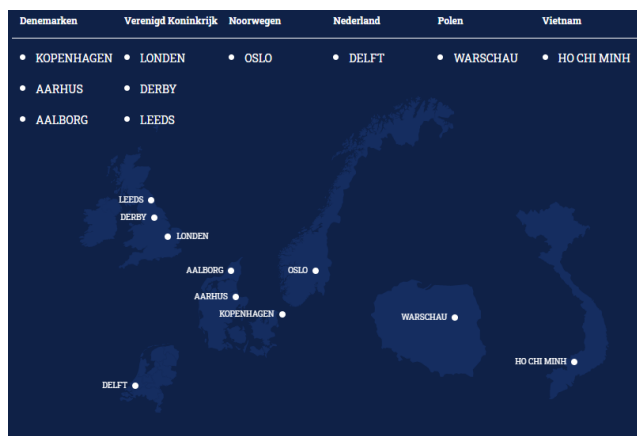
Netcompany is een IT en Business consulting bedrijf, dat end-to-end IT-diensten van ontwikkeling tot beheer, onderhoud en hosting levert. Het bedrijf bestaat uit meer dan 2500 werknemers, met in totaal 10 vestigingen in 6 landen.

De volgende diensten/ activiteiten worden door Netcompany geleverd:

- Applicatieontwikkeling;
- Design en user experience;
- Data en inzicht;
- Case- en documentmanagement;
- Applicatiebeheer;
- Exploitatie en infrastructuur;
- Digitale web- en selfserviceoplossingen;
- Apps en mobiel.

De soorten klanten, of ook wel sectoren genoemd, van Netcompany zijn:

- Overheid;
- Finance;
- Industrie;
- Gemeenten;
- Ledenorganisaties;
- Groot- en detailhandel;
- Dienstverlenende bedrijven;
- Vakbonden;
- Telecommunicatie en media;
- Energie en nutsbedrijven



Figuur 1: Netcompany vestigingen

Netcompany groeit per jaar buitengewoon. In 2019 werkten er 2000 mensen, en in 2020 is dit aantal met 500 toegenomen. Bij de vestiging in Nederland bestaat Netcompany nu in totaal met 146 werknemers. Het doel is om in 2021 de 200 werknemers te passeren. Waardoor groeit Netcompany zo snel? Netcompany probeert zich te verbeteren door ervaringen op te doen met hun projecten. Deze ervaringen worden aan de Netcompany methodologie toegevoegd als richtlijnen en best 'practices'. De methodologie is dus een model wat door de werknemers wordt onderhouden. De 'deliver methodology' bestaat uit een agile ontwikkelingsmodel in combinatie met een strikte projectmanagementmethode en tools om het model en de methode te ondersteunen.

Met opmerkingen [PM1]: strikt? in Agile?

3. Opdrachtschrijving

3.1. Projectomgeving

Netcompany maakt projecten alleen met traditioneel development. Traditioneel development is een handmatig proces, dat betekent dat een persoon de software creëert. Alle applicaties worden gerealiseerd met de programmeertaal. Net in combinatie met een programmeertaal om de gebruikersinterface te bouwen.

Netcompany is continu op zoek naar nieuwe manieren om projecten te realiseren. Als de projecten sneller en met kwaliteit gebouwd kunnen worden door middel van andere methodes, betekent het dat er meer projecten binnengehaald kunnen worden. Een keuze maken om apps te bouwen met een low-code platform kan een organisatie zoals Netcompany de middelen bieden om in een korte tijd complexe en effectieve software te produceren zonder daarvoor een volledige staf van software engineers te gebruiken (Reilly, 2019).

De vraag van de opdrachtgever luidt: “Kan door middel van een low-code platform sneller en efficiënter worden ontwikkeld zonder kwaliteitsverlies ten opzichte van traditioneel development?” Low-code is gericht op het visueel ontwerpen van applicaties. Er zijn veel verschillende platforms beschikbaar (Ngih, 2020). Voor dit onderzoek zal er gekeken worden naar welk platform het beste past conform de Netcompany methodologie.

Om hierachter te komen heb ik het volgende als hoofdonderzoeksvraag gesteld:

“Kan door middel van een low-code platform sneller en efficiënter software worden ontwikkeld zonder kwaliteitsverlies ten opzichte van traditioneel development en welk low-code platform past het beste bij Netcompany?”

3.2. Doelstelling project

Het doel van het onderzoek is om erachter te komen of low-code beter aansluit bij Netcompany dan traditioneel development. Kan het low-code platform sneller zijn met het ontwikkelen van software en efficiënter zijn, zonder kwaliteit te verliezen ten opzichte van het traditioneel development platform?

Het zou voor Netcompany onder andere aan de volgende verwachtingen moeten voldoen:

- Sneller met ontwikkelen van software, dit betekent dat er in een kortere tijd een applicatie gerealiseerd kan worden (Studio61, 2020);
- Het voorkomen van ‘secure vulnerabilities’, die vaak door softwareontwikkelaars worden gecreëerd (onbewust en bewust). (Software development security, 2020);
- Leercurve minder stijl, waardoor een junior meer zelfstandig kan ontwikkelen;

3.3. Opdrachtformulering

3.3.1. Projectopdracht

Voor de applicatie TenderNed is er een POC gerealiseerd met behulp van traditionele development technieken. Dit heb ik volledig nagebootst in een low-codeplatform. Voordat ik hiermee kon beginnen heb ik eerst onderzocht welk type bij Netcompany past. Voor het kiezen van zo’n platform heb ik een adviesrapportage gemaakt. Het verschil tussen traditioneel development en het gekozen low-codeplatform wordt onderbouwd door mijn bevindingen.

3.3.2. Opgeleverde producten en diensten

- Adviesrapport: Het low-code platform wat geschikt is conform de eisen van Netcompany. Zie bijlage X;
- Bevindingen: low-code ontwikkelsnelheid en kwaliteit. Zie hoofdstuk 6.

4. Aanpak

In dit hoofdstuk vertel ik in grote lijnen wat de aanpak was voor mijn onderzoek. In paragraaf 4.1 bespreek ik de werkwijze voor de twee grote stappen die ik moest zetten om mijn hoofdvraag te beantwoorden. Het ging om het geschikte low-codeplatform zoeken voor Netcompany en wat de verschillen waren tussen low-code en traditionele coderen. Bij de volgende paragraaf leg ik de methodes uit die ik heb gebruikt tijdens mijn research conform de methodenkaart (Methods, 2018). Zij bieden onderzoeksmethodieken aan, wat gerelateerd is aan ICT-projecten. Ik ben van mening dat dit een goed overzicht schept voor het onderzoek.

4.1. Hoe

De laatste 17 weken ben ik bezig geweest met de beantwoording van mijn hoofdvraag:

“Kan door middel van een low-code platform sneller en efficiënter software worden ontwikkeld zonder kwaliteitsverlies ten opzichte van traditioneel development en welk low-code platform past het beste bij Netcompany?”

Om een antwoord te kunnen geven op de hoofdvraag heb ik mijn onderzoek in tweeën verdeeld. Het eerste gedeelte is het zoeken naar het geschikte platform voor Netcompany. Hiervoor is eerst een literatuuronderzoek gedaan, waarin generieke eisen worden opgesteld waaraan een moderne applicatie moet voldoen. Vervolgens is deze lijst uitgebreid met specifieke wensen en eisen van Netcompany zelf. Deze specifieke eisen zijn voortgekomen uit interviews met een software architect van Netcompany, en een interview met de sales afdeling. Aan de hand van deze lijst is er met behulp van verschillende beslissingsmethodieken gekeken, welk low-code platform het beste aan deze eisen voldoet. Uit het onderzoek kwam het platform Mendix als beste kandidaat naar voren.

Het tweede gedeelte van mijn onderzoek richt zich op de vraag of met low-code sneller en efficiënter ontwikkeld kan worden zonder kwaliteitsverlies. Er is een applicatie, die ontwikkeld is met traditioneel development, nagebouwd in Mendix om de ontwikkeltijd te vergelijken. De applicatie ontwikkeld met low-code heeft in totaal 32 uur gekost, terwijl de traditionele respectievelijk 160 uur nodig had. Hieruit blijkt dat deze applicatie sneller en efficiënter ontwikkeld kan worden met low-code. Daarnaast is er een literatuuronderzoek gedaan naar de kwaliteit van low-code. Hierin is ingegaan op performance, testen van op maat gemaakte componenten, monitoring, en vendor lockin. Uit dit onderzoek is gebleken dat Mendix vergelijkbare kwaliteit levert in vergelijking met traditioneel development.

4.2. Methodes

De volgende methodes heb ik toegepast aan mijn traject conform de methodenkaart (Methods, 2018) en bestaan uit de volgende:

Methodes	Omschrijving	Gebruikt bij welke stap?
Literature studie	- Het maken van een zoekplan over low-code; - Samenvatting literatuuronderzoek aantekeningen van de gevonden artikelen; - Wetenschappelijke artikelen en scripties gezocht via Google Scholar, HHS Simultaan; - Rapportage van marketing bedrijven zoals Gartner en Forrester; - Informatie verzamelen over kwaliteitsborging bij ISO/IEC.	<i>Adviesrapport het kiezen van een low-codeplatform</i>
Available product analysis	Het analyseren van de applicatie van het ‘Proof of Concept’ die met traditioneel development ontwikkeld is. Hier bevinden zich de ‘requirements’ voor het low-code ‘Proof of Concept’.	<i>Low-code efficiëntie</i>

Expertinterview	▪ Interview met de productowner om achter te halen welke eisen Netcompany heeft; ▪ Interview met collega van afdeling Sales;	<i>Adviesrapport het kiezen van een low-codeplatform</i>
	▪ Interview met 2 ontwikkelaars die met low-code hebben gewerkt of bij het platform hebben gewerkt; ▪ Gesprekken met Mendix.	<i>Low-code efficiëntie</i>
Explore user-requirements	Het invullen van 'user stories' in het Mendix programma	<i>Low-code efficiëntie</i>
IT Architecture sketching	Domain modeleren in het programma Mendix	<i>Low-code efficiëntie</i>
Prototype	'Proof of Concept' in Mendix	<i>Low-code efficiëntie</i>
Product review	Voortgangsrapport	<i>Algemeen</i>

Tabel 1: Methodieken uit het methodenkaart die voor dit onderzoek toegepast waren.

5. Keuze low-codeplatform

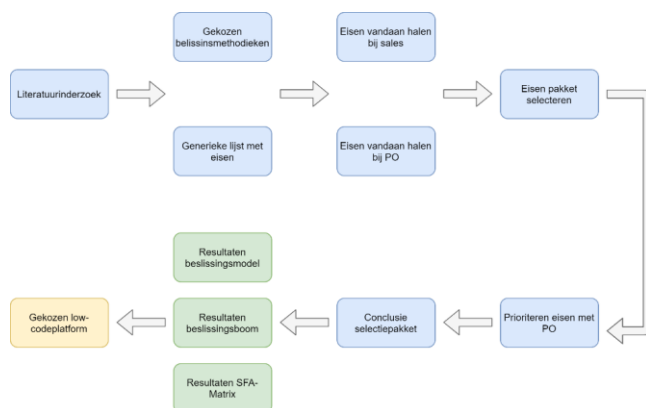
In dit hoofdstuk wordt beschreven hoe het onderzoek naar een geschikt low-codeplatform voor NetCompany is aangepakt. In paragraaf 5.1 geef ik aan wat voor soort onderzoek het is. Daarna behandel ik mijn literatuuronderzoek en hoe ik aan informatie ben gekomen. Er zal ook onderbouwd worden wat de connectie is tussen model-driven development en low-code. Tijdens mijn zoektocht naar betrouwbare bronnen, vond ik een generieke lijst met kenmerken voor model-driven development platformen, wat in paragraaf 5.2 wordt behandeld. Ook zijn er een aantal methodieken die ik voor het onderzoek heb toegepast. De keuzes van deze methodieken onderbouw ik in paragraaf 5.3. Hierna was mijn volgende stap de eisen van Netcompany samenstellen en prioriteren. Hier worden de interviews met de afdeling Sales en de PO besproken. Als product zal de selectie pakket aan eisen gepresenteerd worden. De producten die ter sprake komen zijn:

- Generieke lijst geprioriteerd conform de MoSCoW methodiek;
- Uit eindelijk selectie pakket eisen Netcompany.

In paragraaf 5.5 heb ik een samenvatting van de resultaten wat uit de beslissingsmethodieken zijn gekomen. Met behulp van de uitgekomen resultaten van de beslissingsmethodieken, heb ik mijn oordeel gegeven over het gekozen low-codeplatform. Als laatste kijk ik terug naar het proces dat ik heb gevolgd om een low-codeplatform te kiezen. Er wordt ook terug gekeken naar mijn afstudeerplan en waarom ik voor bepaalde onderdelen ben uitgeweken.

5.1. Soort onderzoek

Dit onderzoek is een kwalitatief onderzoek, waar de beste keuze op basis van literatuur en interviews bepaald wordt. Dit is een praktijkgericht onderzoek en bekijk ik naar de concrete situaties en welke oplossing het beste past. In Figuur 2 wordt in grote lijnen de route aangetoond die ik heb bewandeld om het gekozen low-codeplatform te bereiken.



Figuur 2: Onderzoeksproces voor het kiezen van een low-codeplatform voor Netcompany

5.2. Literatuuronderzoek

In deze paragraaf beschrijf ik in wat mijn literatuuronderzoek inhield. Als eerst heb ik een zoekplan opgesteld met mijn eigen vereisten waar mijn zoekplan aan zou moeten voldoen. Na het verzamelen van data heb ik gekozen om een aantal artikelen samen te vatten. Uit mijn literatuurstudie ben ik erachter gekomen dat er een connectie is tussen model-driven development en low-code. Tevens heb ik een generieke lijst ontdekt en geef ik aan waarom ik dit heb gebruikt voor mijn onderzoek.

5.2.1. Zoekplan

Mijn eerste taak was het bijeengaren van de juiste literatuur, omdat het onderwerp low-code voor mij onbekend was. Met behulp van een zoekplan probeerde ik mijzelf te bakenen. De volgende vereisten zijn daarom opgesteld:

- Wetenschappelijke artikelen, scripties en rapportages mogen niet ouder dan het jaar 2015 zijn;

- Voor de kwaliteitsnormen zal de laatste versie (jaar 2011) opgenomen worden in mijn verslag;
- Er worden geen blogs of andere internetwebsites gebruikt die niet betrouwbaar genoeg is;

Naast deze vereisten vond ik het belangrijk om te beginnen met de volgende termen en ben ik met een sneeuwbalmethode verdergegaan:

- Low-code;
- Low-code ecosystems;
- Low-codeplatform;
- Visual tools;
- Advantages and disadvantages low code programming;
- Organisation;
- Top 10 low-codeplatforms;
- Code kwaliteitsnormen;
- ISO-norm.

Door de sneeuwbalmethode ging het zoeken naar betrouwbare bronnen eenvoudiger en kon ik blogs en onbetrouwbare internetwebsites vermijden. Hiermee associeer ik de websites van low-codeleveranciers. Feiten, dus cijfers en rapporten zijn betrouwbaar, maar overige informatie van de leverancier zelf zijn niet betrouwbaar genoeg. De wetenschappelijke artikelen, en rapportages heb ik toegevoegd in mijn zoekplan met de volgende informatie:

- Soort publicatie;
- URL;
- Titel;
- Auteur;
- Organisatie;
- Jaar.

Voor het definitieve zoekplan zie bijlage III.

5.2.2. Keuze uit de gevonden artikelen en samenvatten van artikelen

De volgende stap na het opstellen van mijn zoekplan is het ordenen van mijn verzamelde artikelen.

Deze bestonden uit de volgende data:

Informatie	Aantal
Rapporten van onderzoeks- en adviesbedrijven	4
Scripties	2
Wetenschappelijke artikelen	3
Vakbladen	1
Standaarden	2

Tabel 2: Gevonden informatie:

Voor het samenvatten van al deze artikelen kwam ik tijd tekort. Ik heb er daarom voor gekozen voor de volgende selectie: Mijn eigen aantekeningen van de rapporten van de onderzoeks- en adviesbedrijven, de scripties, 2 wetenschappelijke artikelen, het vakblad en de ISO standaard. Zie bijlage IV.

5.2.3. Model-driven development en low-code

In (Farshidi, Jansen, & Fortuin, 2021) wordt er gesproken over Model-driven development (later genoemd MDD). Dit is een software development methodologie dat zich concentreert op het creëren en exploiteren van domein modellen. Het doel is om productiviteit te verhogen door de compatibiliteit tussen systemen te maximaliseren.

Voorbeelden hiervan zijn:

- Het hergebruiken van gestandaardiseerde modellen;
- Het ontwerpproces te vereenvoudigen;
- De communicatie tussen individuen en teams die aan het systeem werken te bevorderen (Wikipedia Contributors, 2021).

Maar welke connectie is er tussen low-codeplatformen en MDD? Low-codeplatformen maken gebruik van dezelfde technologie.

1. De low-codeplatformen bieden een manier de productiviteit te verhogen met behulp van het hergebruiken van componenten;
2. Het ontwerpproces wordt vereenvoudigd door terugkerende design patterns (gebeurt allemaal onderwater);
3. Het bevordert de communicatie met behulp van het integreren (denk aan best practices wat gebruikt wordt in het domein model.)

5.2.4. Generieke lijst met eisen

Uit het artikel van (Farshidi, Jansen, & Fortuin, 2021) wordt een generieke lijst gepresenteerd. Deze is samengesteld met behulp van experts die veel ervaring hebben met MDD. Dat maakt dit een betrouwbare lijst met features. Het prioriteren van de kenmerken is conform de MoSCoW-methodiek. De lijst wordt in de volgende onderdelen verdeeld:

Onderdeel	Omschrijving
Modeling spectrum	Het model dat gebruikt wordt om software te ontwikkelen. (Low-code en no-code)
Application types	Welk type applicatie het bedrijf wenst om te bouwen.
Platform types	Wat voor low-codeapplicatieplatform het bedrijf wenst
Security and Protection	Welk beveiliging en bescherming aanwezig moet zijn bij het platform
Data modeling	Hoe data gemodelleerd wordt. Denk aan de ER-diagram of ook wel EAR genoemd (Entity-Attribute-relationship)
Model transformation	Is een geautomatiseerde manier om modellen te wijzigen en te creëren. Het doel is om moeite te besparen en fouten te verminderen door het bouwen aanpassen van modellen waar mogelijk geautomatiseerd kan worden.
Business rule modeling	Een manier om bedrijfsregels formeel te definiëren, dit is inclusief de manier waarop ze worden gerangschikt en de domeinelementen waarmee ze verband houden. Met het model bereidt zich voor op het automatisch generen van applicatie (programmeer) code die uiteindelijk zo worden gebruikt om de regels te implementeren.
User Interface	Gebruikersomgeving van het platform.
Expression modeling	Manier hoe ontwikkelaars code gaat implementeren. Dit kan een programmeertaal gebruiken zijn, maar ook een iets wat dichtbij programmeren valt. Denk aan een taal in Excel.
Business Process Automation	Het automatiseren van bedrijfsprocessen.
Developer Citizen	Wie zal gebruik maken van het platform. De professionele ontwikkelaars of juist de bedrijfsanalist.
Qualitative Analysis	Kwalitatieve analyse over de leverancier. Hoe volwassen in het bedrijf. Microsoft is een groot bedrijf en is daardoor volwassen.
Development	Wat het platform allemaal kan ontwikkelen. Denk aan deployment, het visuele IDE van het platform, datamanagement en integratie (SOAP, REST).

Tabel 3: Onderdelen uit het generieke lijst

5.3. Gekozen beslissingsmethodieken

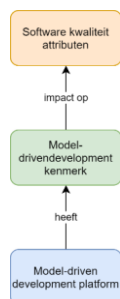
Tijdens het literatuuronderzoek kwam ik tot de conclusie, dat ik twee beslissingsmethodieken kan gebruiken van reeds geleverde wetenschappelijke artikel en een scriptie. Maar alsnog zou dit onvoldoende zijn voor het onderbouwen van mijn keuze. Daarom besloot ik om nog een SFA-matrix toe te passen.

5.3.1. Beslissingsmodel

Het beslissingsmodel van (Farshidi, Jansen, & Fortuin, 2021), bestaat uit een 'framework' en een decision support system (later genoemd DSS) voor het ondersteunen van software-engineers met hun MCDM-problemen bij softwareproductie. Voor deze problemen biedt het framework een richtlijn voor het bouwen van beslissingsmodellen. Deze bestaat uit de volgende zes stappen:

1. Identificeren van het object;
2. Selectie van kenmerken;
3. Selectie van alternatieven;
4. Selectie van de wegingsmethode;
5. Het toepassen van de aggregatiemethode;
6. Een beslissing maken gebaseerd op het aggregatieresultaat.

In Figuur 1 wordt er getoond hoe het beslissingsmodel gedeeltelijk eruitziet van (Farshidi, Jansen, & Fortuin, 2021) om een beter beeld te krijgen hoe het beslissingsmodel werkt. Model-driven development platform zijn platformen zoals Microsoft PowerApps, Mendix, Outsystems en Salesforce. Hiernaast wordt er gekeken welke model-driven development kenmerken het platform heeft. En als laatste zal er gekeken worden welke impact het heeft als het om softwarekwaliteit attributen gaat. Softwarekwaliteit attributen bestaan uit de ISO/IEC 25010. De ISO is een internationale organisatie voor standaardisatie. Zij stellen de normen vast voor systemen en softwarekwaliteit modellen. Hier beschrijft de organisatie het model voor softwareproductkwaliteit en softwarekwaliteit. Zie hieronder de kenmerken en subkenmerken.



Figuur 3: Het beslissingsmodel voor het selecteren van het Model-drivenplatform

5.3.2. Beslissingsboom

Volgens (Rymer J. R., Mines, Vizgaitis, & Reese, 2017) is het moeilijk om op tijd aan de zakelijke vereisten te voldoen en is dit de grootste uitdaging van bedrijven. Met een low-codeplatform bouw je sneller een applicatie vergeleken met traditioneel development (Virta, 2018). De beslissingsboom van (Sattar, 2018) selecteert een low-codeplatform gebaseerd op organisatie en de applicatie type die zij realiseren. Hiernaast worden de low-codeplatform leveranciers beoordeeld per criteria. De evaluatiegegevens bevatten meerdere vragen om de beoordeling van het Low-Code Platform te vormen. Het gaat om de volgende criteria:

- Startup Experience
- User Interface development
- Data mangement
- Digital process automation
- Reporting and analysis
- AD&D support and governance tools

- Cloud platform attributes
- Vision
- Road Map
- Market Approach

Deze beslissingsboom is samengesteld op basis van de gegevens uit het literatuuronderzoek van (Sattar, 2018). In Figuur 4 wordt de beslissingsboom getoond.

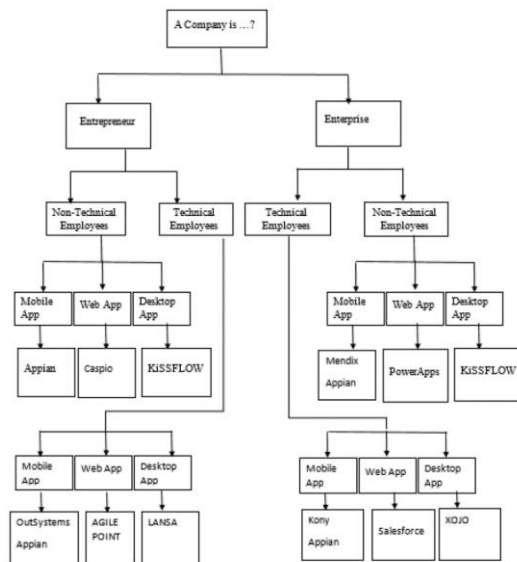
Type organisatie	Definitie
Entrepreneur	Een ondernemer is een persoon die een klein bedrijf opricht en runt en alle risico's en beloningen van de onderneming op zich neemt; Iemand die een bedrijf of bedrijven opricht en financiële risico's neemt in de hoop op winst; Een ondernemer is een persoon die, in plaats van als werknemer te werken, een klein bedrijf opricht en runt, waarbij hij alle risico's en voordelen van de onderneming op zich neemt. De ondernemer wordt vaak gezien als een innovator, een bron van nieuwe ideeën, goederen, diensten en zaken/ of procedures.
Enterprise	Een bedrijf dat is opgericht voor commerciële doeleinden; Het leveren van goederen en diensten waaraan financiële, commerciële en industriële aspecten zijn verbonden.

Tabel 4: Definitie van type organisaties volgens Sattar

Type applicatie	Definitie
Mobiel applicatie	Een mobiele app is een softwareapplicatie die speciaal is ontwikkeld voor gebruik op kleine, draadloze computerapparaten, zoals smartphones en tablets, in plaats van desktop- of laptopcomputers;
Webapplicatie	Een webapplicatie is een computerprogramma dat webbrowsers en webtechnologie gebruikt om taken via internet uit te voeren; Een webapplicatie (webapp) is een applicatieprogramma dat is opgeslagen op een externe server en via internet wordt geleverd via een browserinterface.
Desktopapplicatie	Een applicatie die stand-alone draait op een desktop of laptop; De term 'desktopsoftware' verwijst naar software die wordt uitgevoerd op een Mac-, Windows- of Linux-desktopcomputer die is ingesteld voor gebruik op één plek, zoals een bureau.

Tabel 5: Definities van type applicatie volgens Sattar

De uitkomst qua keuze van de low-code platformen is summier (11 low-codeplatformen in de beslissingsboom) vergeleken met het beslissingsmodel (30 low-codeplatformen) wat in paragraaf 5.3.1. is beschreven. De reden hiervoor is dat Sattar voor de low-codeplatformen heeft gekozen die in het Forrester rapport is opgesteld in 2017 (Rymer J. , Mines, Vizgaitis, & Dostie, 2017). Het resultaat en mijn onderbouwing wordt in hoofdstuk 5.5.2 beschreven.



Figuur 4: Beslissingsboom van Satar

5.3.3. SFA-matrix

Tijdens mijn literatuurstudie ben ik erachter gekomen dat het noodzakelijk is om ook mijn eigen inbreng te leveren voor dit onderzoek en niet alleen methodieken gebruiken die door anderen zijn gerealiseerd. Dit komt meer betrouwbaar over en kan ik mijn conclusie ook onderbouwen met mijn eigen inbreng. Voor ik aan mijn SFA-matrix ben begonnen heb ik aan de hand van het generieke lijst wat in paragraaf 5.2.4, een scorematrix gerealiseerd en heb ik aan de MoSCoW-methode een wegingsfactor aangehangen (Tabel 6). Per low-codeplatform ben ik gaan kijken of de kenmerken aanwezig waren.

Tijdens mijn zoektocht naar het juiste template voor mijn adviesrapport kwam ik een artikel tegen hoe ik een perfect adviesrapport kon schrijven (Benders, 2020). In dit artikel kwam het FOETSJ-model ter sprake. Aangezien ik een poging wou wagen om een perfect adviesrapport te schrijven heb ik besloten om niet alleen dit model, maar het gehele SFA-matrix te gebruiken aangezien de FOETSJ-model een onderdeel is van deze matrix. Een SFA-matrix wordt vaak gebruikt voor marketing. Eén van de definities van marketing is "alles wat een bedrijf doet om de verkoop van producten of diensten te bevorderen". Dit betekent dat dit onderzoek Netcompanies diensten kan bevorderen door de juiste low-codeplatform te kiezen.

MoSCoW methodiek	Wel aanwezig	Niet aanwezig
Must have	5	-5
Should have	3	-3
Could have	1	-1
Won't have	0	0

Tabel 6: Wegingsfactor voor het generieke lijst

De SFA-matrix bestaat uit de volgende onderdelen:

- Suitability
Wordt er vastgelegd in hoeverre de optie aansluit bij de kansen die zich in de markt voordoen.
- Feasibility
In hoeverre de optie aansluit bij de mogelijkheden van de organisatie.
- Acceptability
In hoeverre de winst, en het bijbehorende risicoprofiel van de verschillende opties aansluiten bij de verwachtingen van de belanghebbenden.

Tijdens een voortgangsgesprek met de productowner komen de onderwerpen te vervallen wat onder de 'Feasability' vallen, omdat dit niet van toepassing zou zijn met het onderzoek. Het gaat om de volgende vragen:

- Sociaal verantwoord?
- Ethisch verantwoord?

Het SFA-Matrix is in overleg gegaan met de PO, echter was het ook mijn verantwoordelijkheid om aan te geven waarom ik dit zou willen toepassen. De punten in mijn SFA-wegingsfactor zijn gerealiseerd met behulp van bijlage IX en de interview met Sales. (Hoofdstuk 5.4.1)

5.4. Eisen van Netcompany voor software

In deze paragraaf wordt beschreven hoe ik aan de eisen van Netcompany kom. In paragraaf 5.4.1 zijn de interviews met collega's van Netcompany behandeld en hoe ik de eisen heb geprioriteerd. Tijdens het interview met mijn collega werd er een aantal low-codeplatformen genoemd die zijn voorkeur hadden. Deze worden in paragraaf 5.4.2 besproken. Als laats heb ik een selectie pakket aan eisen opgesteld en behandel ik deze in paragraaf 5.4.3. Door het onderzoek weet ik nu hoe ik tussen verschillende platformen kan kiezen, aan welke eisen een platform zou moeten voldoen en hoe deze geprioriteerd dienen te worden. En zo heb ik bepaald welke generieke eisen voor Netcompany gelden.

5.4.1. Eisen uit interviews/ prioriteren eisen

5.4.1.1. Interview met salesafdeling

Dit is een half gestructureerd interview met een Sales collega voor het achterhalen of er projecten aanwezig zijn in TenderNed die expliciet om low-code vragen. Dit betekent dat ik een lijst met onderwerpen heb gehanteerd met de termen low-code, populair en de volgende vraag: 'Is er meer vraag naar low-code in aanbestedingen?'

Het gesprek met mijn collega van Sales heeft ongeveer 15 minuten geduurd. Het doel was om achter te halen of er wel degelijk interesse is voor applicaties die gebouwd worden in low-code bij de klanten. De conclusie is dat low-code grond wint bij klanten. De klanten van Netcompany richten zich op projecten die overheid gerelateerd zijn.

Als oplossing probeert Netcompany andere opties voor te dragen aan de klant, zoals een combinatie met Dynamics 365, een .Net backend en het gebruiken van Microsoft PowerApps (is low-codeplatform) als frontend. Echter als feedback van een klant kregen ze de volgende reactie terug:

"Microsoft onderkent dit ook en heeft in 2017 MS Powerapps als no-code (low-codeplatform) uitgebracht. Echter, ook deze tooling (waar de gegadigde in haar referentie niet naar refereert) is geen oplossing voor de klant waar deze enkel geschikt is voor het automatiseren van kleinere en dagelijkse herhalingstaken. Tevens blijkt uit de aangeleverde referentie dat ASP.NET (object georiënteerde programmeertaal) benodigd is om functionaliteiten te verwezenlijken. Ook dit voldoet niet aan de wens van de klant dat toekomstige software niet meer wordt ontwikkeld, maar wordt geconfigureerd."

Er is een stijgende lijn te zien in het vragen naar software dat in low-code gebouwd wordt. Netcompany heeft geen projecten met low-code, waardoor ze genoodzaakt zijn om andere opties voor te stellen aan nieuwe projecten. Het doel voor Netcompany is om meer informatie te weten over low-code en wat zijn de voor- en nadelen van low-code als Netcompany het gaat gebruiken.

Hiernaast wordt er afgevraagd of de klant bewust is wat low-code echt inhoudt. Het is een term wat pas in 2014 is gebruikt (Virta, 2018) en is op dit moment populair/ hip. Low-codeplatformen geven als voordeel dat je tijd kan winnen. Een applicatie is bijvoorbeeld in 2 maanden gebouwd in plaats van 6 maanden. Wat betekent dit voor de voorbereiding? Betekent dit voor Netcompany dat er in het analysefase extra werk is vergeleken met de huidige stand van zaken (ontwikkelen in traditioneel development)? Een grote zorg van het gebruiken van low-codeplatformen is de kans van vendor-lock in. Hoe kan je dit voorkomen?

Met opmerkingen [ra2]: Is dit een vraag? of een statement?

Met opmerkingen [pm3R2]: Het was inderdaad een vraag :)

5.4.1.2. Interview met de 'productowner'

Tijdens de voortgangsgesprekken met de productowner kwamen er een paar eisen waar een low-codeplatform aan zou moeten voldoen:

- Het voorkomen van vendor-lock in;
- Het moet mogelijk zijn om de applicatie compleet zelf te beheren (on-premises);
- De eigenaar van de code moet de klant zijn en niet de low-codeplatform leverancier;
- Een mogelijkheid hebben om een .Net backend te combineren met een frontend low-code applicatie;
- Performance moet niet eronder lijden.

5.4.1.3. Interview prioriteren van eisen door de productowner

De generieke lijst wat in paragraaf 5.2.4 staat beschreven bestaat uit 84 kenmerken waar een low-codeplatform aan voldoen kan. Tijdens dit interview met de productowner heb ik deze lijst doorgenomen, wat 1 uur duurde. De kenmerken worden geprioriteerd conform de MoSCoW-methodiek, maar er is ook de keuze om er geen toepassing van te maken (none). Dit heb ik niet zelf omschreven, maar dit stond al vastgelegd in de literatuur zelf. De reden hiervoor is voor mijzelf onbekend gebleven en stond niet beschreven in (Farshidi, Jansen, & Fortuin, 2021). De volledige lijst wat ik met de productowner heb geprioriteerd staat beschreven in bijlage V.

Tijdens het interview was er aangegeven dat de kans groot is dat deze generieke lijst aangepast zou worden. De conclusie van dit gesprek is dat Netcompany zich nu bewust kan maken aan welke kenmerken een low-codeplatform kan voldoen.

5.4.2. Low-codeplatform voorkeuren

Met de productowner is er een afspraak gemaakt om 4 low-codeplatformen te bestuderen. Op basis van zijn eigen prioriteiten heeft de productowner voor 3 grote leiders van low-codeplatformen gekozen. Deze leiders zijn: *Mendix*, *Outsystems* en *Microsoft PowerApps*. Hiernaast heb ik geadviseerd om ook te kijken naar *Salesforce*. Hiernaast heb ik geadviseerd om ook naar *Salesforce* te kijken. Dit platform is namelijk een aantal keren ter sprake gekomen bij Netcompany.

5.4.3. Selectie pakket eisen

Na alle eisen in kaart te brengen en geprioriteerd te hebben, kwam er een selectie pakket uit conform de eisen van Netcompany. In Tabel 7 worden de belangrijkste onderdelen benoemd en omschreven. Echter moet ik erop attenderen dat het lijst wat ik hier presenter in de toekomst veranderd kan worden. Voor het complete lijst zie bijlage VI.

Must Haves	Omschrijving
Access Control	
Single Sign-on	Een gebruiker logt in met een enkele ID en wachtwoord om toegang te krijgen tot een van de verschillende gerelateerde systemen
OAuth	OAuth is een open standaard voor toegangsdelegatie, die vaak wordt gebruikt als een manier voor internetgebruikers om websites of applicaties toegang te verlenen tot hun informatie op andere websites, maar zonder hen de wachtwoorden te geven.
Multi-factor authentication	Een authenticatiemethode waarbij een computergebruiker alleen toegang krijgt nadat hij met succes twee of meer bewijsstukken gepresenteerd heeft aan een authenticatiemechanisme. Denk aan een tweefactor-authenticatie.
Identity and permissions management	Identiteits- en toegangsbeheer beheert gebruikers door zowel authenticatie als autorisatie af te dwingen. Authenticatie verifieert de identiteit van een gebruiker. Eenmaal geauthentiseerd, kan de geverifieerde gebruiker alle bronnen gebruiken waarvoor zijn account toegang heeft.
Applicatie type	

Web-portals	Een web portaal is een speciaal ontworpen website die informatie uit verschillende bronnen, zoals e-mails, online forums en zoekmachines, op een uniforme manier samenbrengt. Gewoonlijk krijgt elke informatiebron zijn eigen gebied op de pagina voor het weergeven van informatie; vaak kan de gebruiker configureren welke worden weergegeven.
Data modelleren Entity-Attribute-relationship	Manier van data modelleren. Diagram maakt representeert attributen en entiteiten. Wat ook wel de EAR diagram genoemd. Verwar dit niet met een ER-diagram. Het verschil tussen deze twee is dat een EAR-diagram ook het begrip attribuuotype bevat dat lost staat van een relatietype.
Object-role modelling	Een object-rolmodel gebruikt grafische symbolen die zijn gebaseerd op eerste orde predicatenlogica en verzamelingenleer om de modelleur in staat te stellen een ondubbelzinnige definitie te creëren van een willekeurig universum van discours.
Developers citizen Professional developers	Ontwikkelaars zijn mensen die weten hoe ze applicaties moeten schrijven in een programmeertaal (inclusief SQL).
Kwalitatieve analyse Maturity level of company	Het volwassenheidsniveau specificeert bepaalde kenmerken voor processen, waarbij hogere volwassenheidsniveaus meer geavanceerde kenmerken hebben en is een stap in de richting van het bereiken van een volwassen proces, dat een reeks doelen biedt die, wanneer voldaan is, een organisatie op het volgende volwassenheidsniveau plaatsen.
Model Spectrum Model-centric (low-code)	Ontwikkelaars gebruiken modellen om software te ontwikkelen, de code wordt automatisch gegenereerd.
Platform type General purpose platform	Platforms voor algemene doeleinden richten zich op een breed scala aan web- en mobiele applicaties met uitgebreide collecties declaratieve tools die betrekking hebben op het maken, integreren, implementeren, levenscyclusbeheer en distributie van applicaties (via marktplaatsen).
Security GDPR	De Algemene Verordening Gegevensbescherming 2016/679 is een verordening in de EU-wetgeving inzake gegevensbescherming en privacy in de Europese Unie en de Europese Economische Ruimte.
ISO 27001-2013 certification	ISO/ IEC 27001: 2013 is de toonaangevende norm die de processen specificeert voor het opzetten, implementeren, herzien, bewaken, beheren en onderhouden van een effectief informatiebeveiligingsbeheersysteem (ISMS).
Overige Customer must be the owner of the code when application is in production. Performance of application should be scalable Prevent vendor lock-in	Ook al maken wij gebruik van het low-codeplatform uiteindelijk is de klant de eigenaar van de code. De mogelijkheid om applicatie op te schalen, zowel horizontaal als verticaal. Vendor lock-in maakt een klant afhankelijk van een leverancier voor producten en diensten, omdat hij niet in staat is om van leverancier te veranderen zonder substantiële omschakelingskosten of ongemak.

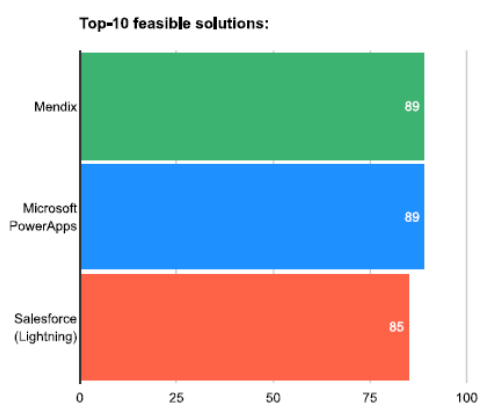
Tabel 7: Een gedeelte van de Must Have's uit het lijst van belangrijke eisen van Netcompany

5.5. Invullen beslissingsmethodieken

In deze paragraaf bespreek ik de resultaten van de beslissingsmethodieken en wat mijn bevindingen waren.

5.5.1. Beslissingsmodel (Farshidi, Jansen, & Fortuin, 2021)

Uit het beslissingsmodel kwam als uitslag 3 low-codeplatformen uit, terwijl ik er 10 had verwacht (Figuur 5). Het verschilt zit in het kiezen hoe volwassen een leverancier is. Er is duidelijk een verschil tussen het low-codeplatform Mendix en Microsoft. Volgens het beslissingsmodel is Microsoft een leverancier met een hoge volwassenheid (*maturity of the company*.) Dit geldt ook voor Salesforce, echter krijgt Mendix de stempel van medium. Voor het volledige rapport van het beslissingsmodel zie bijlage VII. Waarom Outsystems niet werd meegenomen in de top-10 is op dit moment niet te verklaren aangezien het model alleen het resultaat aantoont.



Figuur 5: Resultaat beslissingsmodel

5.5.2. Beslissingsboom (Sattar, 2018)

Om het resultaat te zien moeten de volgende vragen beantwoord worden:

1. Wat voor bedrijf is Netcompany?
Netcompany is een bedrijf georganiseerd voor een commercieel doel. Zij leveren goederen en diensten waaraan financiële, commerciële en industriële aspecten zijn verbonden.
2. Wat voor werknemers zullen aan de slag gaan met het low-codeplatform?
De werknemers die in het bedrijf werken zijn grotendeels technische werknemers. Het zijn analisten, programmeurs, testers etc.
3. Wat voor applicaties worden er mee gebouwd?
De applicaties die zij bouwen zijn webapplicaties, dit zijn applicatieprogramma's dat zijn opgeslagen op een externe server en via internet wordt geleverd via een browserinterface.

Na het beantwoorden van deze vragen komt er als resultaat Salesforce uit. Echter ben ik hier niet mee eens, want Salesforce App Cloud is een platform dat zich richt op Customer Relationship Management (CRM.) De productowner verwacht meer dan alleen een CRM. In (Sattar, 2018) wordt ook expliciet beschreven dat Salesforce een CRM is, en zijn er platformen beschreven die ook voor Netcompany waarde kunnen hebben. Met Outsystems kunnen er ook webapplicaties gebouwd worden en niet alleen mobiele applicaties. Outsystems wordt in de beslissingsboom beschreven als een product dat het beste bij een Enterprise past met technische werknemers. In (Sattar, 2018) wordt er ook beschreven dat Outsystems zowel bij Enterprises als Entrepreneurs past. Dit is een artikel uit 2018. Omdat de leveranciers in 3 jaar drastisch veranderd zijn, beschouw ik dit artikel als verouderd.

5.5.3. SFA-matrix

Naast de beslissingsboom en -model, is er ook een SFA-matrix aanwezig. Het doel is om zelf uit te zoeken welk van de 4 low-codeplatformen het beste bij Netcompany past ter vervanging van de bestaande hulpmiddelen. Een beslissingsmodel kan je op weg helpen, maar neemt niet weg dat men zelf een keuze moet gaan maken. Denk maar aan de verkiezing van de Tweede Kamer. Een beslissingsmodel als Stemmijzer kan je helpen, maar men gaat alsnog voor zijn eigen intuïtie. Vandaar is er voor dit onderzoek een scorematrix aanwezig. Een beslissingsboom of -model is niet genoeg om te bepalen welk low-codeplatform het beste bij Netcompany past.

Er werd gekeken naar de generieke lijst die geprioriteerd werd door de productowner. De volgende uitslag kwam eruit:

MoSCoW Methodiek	Mendix	Microsoft PowerApps	Outsystems	Salesforce
Aanwezig				
Must Have	120	120	115	120
Should Have	51	51	51	48
Could Have	18	16	19	15
Niet aanwezig				
Must Have	0	0	-5	0
Should Have	-6	-6	-6	-9
Could Have	-3	-5	-2	-6
Totaal	180	176	172	168

Tabel 8: Totale punten scorematrix van Mendix, Microsoft PowerApps, Outsystems, Salesforce

Voor de berekening van per low-codeplatform en de volledige scorematrix zie bijlage VIII. In de volledige scorematrix wordt er ook getoond welke onderdelen niet aanwezig waren. Bij alle 4 de low-codeplatformen komen platformtypes 'Process app platform' en 'Request-handling platform' niet in voor, maar ik vraag me af of dit nodig is aangezien de platformen allemaal het type 'general-purpose platform' hebben. Dit betekent dat het platform ook alles zou kunnen wat de andere types kunnen.

	Omschrijving
Natural language rules (Should have)	Onderdeel van 'expression modelleren': Het gebruik van natuurlijke taal versus reguliere expressie-elementen. Met andere woorden, u kunt regels in natuurlijke taal uitdrukken. In plaats van operatoren zoals <, > en = te gebruiken, kan een modelleur bijvoorbeeld zinnen gebruiken zoals er minder zijn dan, er zijn tenminste, alle volgende voorwaarden waar zijn enz.
Process app platform (Could have)	Onderdeel van 'platform type': Low-code platforms voor proces-apps zijn gericht op applicaties die coördinatie en samenwerking tussen verschillende werknemers- en klantrollen vereisen. Deze platforms bieden een rijke mix van procesautomatisering, casemanagement en sociale interactiefuncties die kunnen worden geconfigureerd om gestructureerde en ongestructureerde bedrijfsprocessen te beheren. Low-code platforms voor proces-apps putten hun kracht uit visuele proces- en case-modelleringstools waarmee teams processtroomlogica en ad-hoc taakbeheer visueel kunnen configureren voor een reeks interne en externe rollen, applicaties en databasesystemen. Deze leveranciers bieden ook ingebouwde statistieken, analyses en audittrails voor het bewaken van proces- en casusprestaties.
Request-handling platform (Could have)	Onderdeel van 'platform type': Low-code platforms voor apps voor het afhandelen van verzoeken zijn gericht op applicaties die verzoeken accepteren, verwerken en volgen. Deze platforms komen uit de wereld van IT-servicemanagementoplossingen (itsM) en presenteren aan ontwikkelaars het kerntoepassingspatroon van de afhandeling van verzoeken als

	een raamwerk dat kan worden aangepast met declaratieve tools. De declaratieve tools die deze platforms bieden, bieden het maken van adrestoepassingen, de werkstroom, de implementatie, het beheer van de servicelevenscyclus en de distributie (via marktplaatsen).
--	--

Tabel 9: De afwezige onderdelen van de kenmerken waar een low-codeplatform aan zou moeten/kunnen voldoen

Niet alle onderdelen van de FOETSJ-methode is van toepassing voor dit rapport. In Figuur 6 is het resultaat te zien van het SFA-matrix.

	Gewicht	optie 1: Mendix	optie 2: Outsystems	optie 3: Microsoft platform	optie 4: Salesforce
Suitability					
Goed inspelen op de markt	0,3	(8) 2,4	(8) 2,4	(8) 2,4	(8) 2,4
Creëert onderscheiden vermogen?	0,3	(7) 2,1	(7) 2,1	(7) 2,1	(7) 2,1
Oplossen van centraal probleem?	0,4	(8) 3,2	(7) 2,8	(4) 1,6	(5) 2,0
Feasibility					
Financieel haalbaar	0,1	(7) 0,7	(7) 0,7	(5) 0,5	(5) 0,5
Organisatorisch uitvoerbaar?	0,2	(7) 1,4	(8) 1,6	(9) 1,8	(6) 1,2
Economisch verantwoord?	0,1	(8) 0,8	(8) 0,8	(8) 0,8	(8) 0,8
Technologisch haalbaar?	0,2	(8) 1,6	(8) 1,6	(5) 1,0	(8) 1,6
Sociaal verantwoord?	0,0	0	0	0	0
Juridisch zeker?	0,1	(8) 0,8	(8) 0,8	(8) 0,8	(8) 0,8
Ethisch verantwoord?	0,0	0	0	0	0
Acceptability					
Verwachte winst?	0,3	(8) 2,4	(8) 2,4	(8) 2,4	(8) 2,4
Aanvaardbaar risico?	0,2	(8) 1,6	(8) 1,6	(6) 1,2	(6) 1,2
Acceptatie belanghebbend?	0,5	(8) 4	(8) 4	(8) 4	(8) 4
Totaal		21	20,8	18,6	19

Figuur 6: SFA-matrix low-codeplatform

5.6. Conclusie keuze low-codeplatform

In dit hoofdstuk wordt het gekozen low-codeplatform behandeld. Ik licht toe waarom ik hiervoor heb gekozen en zal mijn vervolgstappen bekend maken.

Aangezien de vraag naar low-code stijgt, is het economisch verantwoord, om te investeren in low-code. De huidige low-code is niet zoals de GL talen wat in de jaren '80 bestond en verdween. In die tijd had men een visie met de GL-talen. De visie "een manier voor ontwikkelaars om meer gedaan te krijgen" is een visie van 4GL wat low-code ook belangrijk vindt. Het grote verschil tussen 4GL en low-code is dat 4GL op een gegeven moment is verdwenen. Bij aanbestedingen vragen zij juist meer naar projecten wat in low-code gemaakt moet worden.

Netcompany wilt graag een platform voor algemeen doeleinden die zich richten op een bredere scala aan web- en mobiele applicaties met uitgebreide collecties declaratieve tools. Zij hebben betrekking op het maken, integreren, implementeren, levenscyclusbeheer en distributie van applicaties (via marktplaatsen). Low-codeplatformen o.a.: Outsystems, Mendix, Salesforce en Microsoft PowerApps.

De belangrijkste eis voor Netcompany is wie de eigenaar zal zijn van de code op het moment dat de applicatie in productie gaat. Voor zowel Mendix, Outsystems en MS PowerApps zal de klant uiteindelijk de eigenaar zijn. Echter is er nog geen onderzoek gedaan voor Salesforce.

Zowel uit de beslissingsmodel en de SFA-matrix komt als resultaat Mendix uit. Bij de SFA-matrix heb ik een scorematrix toegepast (hoofdstuk 5.3.3) en hier scoorde Mendix het beste, omdat deze het meest overeenkomst had met het geprioriteerde generieke lijst. Hiernaast geven zij ook aan dat de klant uiteindelijk de eigenaar is van de code, en proberen zij vendor lock-in te voorkomen. Toch heb ik nog mijn vragen over het voorkomen van vendor lock-in. Ook bieden zij twee

verschillende modi aan, zodat zowel de citizen ontwikkelaar als de professionele ontwikkelaar aan de slag kan met het low-codeplatform.

5.6.1. Mendix

Mendix is één van de leiders van low-code-applicatieplatformen en staat bekend als de concurrent van Outsystems. Het grootste verschil tussen deze twee platformen is dat Mendix zich zowel op citizen developer als de professional developer concentreert. Mendix heeft de mogelijkheid gecreëerd om zowel met Entity-Attribute-relationship als met Object-Role-modelling (data modelleren) te werken. Het Object-role modelling is niet aanwezig bij Outsystems. Mendix was ook de uitkomst van zowel de scorematrix en de SFA-matrix (paragraaf 5.5.3) als met het beslissingsmodel (paragraaf 5.5.1).

5.6.2. Bijdrage Mendix

In het vervolgonderzoek is het de bedoeling om te achterhalen of je sneller een applicatie kan bouwen vergeleken met traditioneel development. Als uitkomst zal er bekeken worden naar de snelheid van bouwen, maar ook de leercurve van het gebruiken van Mendix. Met low-code geven zij aan dat de leercurve korter is vergeleken met traditioneel development. De tijd die ik aan het 'Proof of Concept' zal besteden, zal ik vergelijken met de tijd wat er bij het traditioneel development aan is besteed.

5.7. Verantwoording en reflectie keuze van low-codeplatform

In deze paragraaf beschrijf ik mijn verantwoording en wordt er gereflecteerd over mijn proces voor het zoeken naar een geschikte low-codeplatform voor Netcompany. Als eerst besteed ik aandacht aan het verzamelen, analyseren en verwerken van literatuur. In paragraaf 5.7.2 geef ik aan of ik tevreden was met mijn gekozen beslissingsmethodieken, maar geef ik ook aan of ik dit anders had kunnen doen. In de laatste paragraaf bespreek ik mijn proces van het opstellen van het adviesrapport. Hier had ik tijdens mijn onderzoek moeite mee.

5.7.1. Onderzoek

Tijdens het zoeken naar literatuur kwam ik tegen dat het niet makkelijk was om artikelen te vinden. Dit kwam omdat het term low-code pas in 2014 gepresenteerd werd. Hierdoor werd ik geneigd om naar internetwebsites te zoeken zoals blogs, wat ik niet volwaardig vond om in mijn adviesrapport te plaatsen. Om kwaliteit te leveren verwachtte ik van mezelf om ook wetenschappelijke artikelen, scripties en rapporten te gebruiken. Na 3 weken zoeken heb ik het artikel van (Farshidi, Jansen, & Fortuin, 2021) gevonden en heb ik dit gebruikt als richtlijn voor het interview met de productowner om zo de eisen in kaart te brengen. Zij hadden namelijk een gedetailleerde lijst met kenmerken die zij door middel van expertinterviews hebben opgesteld.

Hiernaast ben ik proactief bezig geweest door low-codeleveranciers te benaderen. Dit zat niet in mijn oorspronkelijke planning, maar ik had meer informatie nodig over de prijzen en abonnementen. Tijdens deze gesprekken kwam ik erachter dat het leren van hun low-codeplatform niet zo makkelijk was als ik zou verwachten. Ik heb daarom demo's bijgewoond voor mijn voor de volgende stap: het maken van een 'Proof of Concept'.

Het oriënteren van verschillende low-codeplatformen had veel gedaan voor mijn onderzoek. Maar achteraf gezien was dit een onderzoek waar nog veel meer in verdiept kon worden. Hiermee bedoel ik dat dit onderdeel al een volledig onderzoek had moeten zijn. Helemaal als men meer waarde zou hechten aan kwaliteit.

Netcompany heeft de belangrijke eisen goed in kaart gebracht. Denk hier aan het voorkomen van vendor-lock in, wie de eigenaar is van de code en on-premises (zit in lijst van kenmerken). Hiernaast heb ik een lijst van kenmerken gepresenteerd waardoor de productowner bewuster werd welke eigenschappen er aanwezig waren voor het kiezen van een low-codeplatform. Dit is een levend document. Hiermee bedoel ik dat dit in de toekomst aangepast kan worden.

5.7.2. Gekozen beslissingsmethodieken

Het beslissingsmodel heeft mij geholpen met het onderbouwen van mijn conclusie, omdat het uitslag drie platformen aantoonde waar ook de voorkeur van de productowner naar toe ging. Ik begrijp alleen nog steeds niet waarom Outsystems niet als resultaat naar voren kwam. Het beslissingsmodel zou ik in de toekomst nog een keer kunnen gebruiken wanneer de eisen van Netcompany zouden veranderen. Het beslissingsmodel was een passende keuze, omdat zij aan onderdelen dachten die ik graag zou willen gebruiken om uit te zoeken welk platform het beste bij Netcompany zou passen. Hier gaat om het kwaliteitsnorm ISO/IC 25010, het kijken naar de kenmerken van een low-codeplatform en de MoSCoW methodiek.

De beslissingsboom was uiteindelijk niet van toepassing voor de conclusie in het rapport. Dit toont aan dat je niet blindelings moet gaan vertrouwen op één methode. Als ik alleen voor de boom had gekozen zou dit gevolgen hebben voor mijn conclusie. Ik kan alleen speculeren waarom de boom niet zou werken. De reden hiervoor zou zijn dat het artikel verouderd is. Het proces dat de low-codeplatformen ervaren gaat zo snel. In 3 jaar is hier veel in veranderd.

De SFA-matrix heeft wel degelijk invloed gehad op mijn besluit. Als resultaat kwam Mendix naar voren, maar de andere low-codeplatformen waren erg dichtbij. De SFA-matrix wordt veel gebruikt bij marketing en zou niet bij dit opleiding passen. Echter om een passende optie te maken was het wel degelijk een methodiek wat goed zou werken met geschikte literatuur en de juiste vragen stellen bij expertinterviews van Netcompany.

Om de matrix nog nauwkeuriger te maken had ik meer moeten expertinterviews moeten houden bij Netcompany. Het gaat over het financiële kosten voor de organisatie. Tijdens mijn gesprek met Mendix en Outsystems merkte ik dat zij niet transparant zijn als het gaat om de gehele kosten. De reden was niet duidelijk. Als gemiddelde gaven zij beide aan dat het ongeveer €10.000,- zou gaan kosten. Maar dat was slechts een gemiddelde. Het bedrag werd ook niet gespecificeerd. En ik kon hier dus niet veel uit concluderen.

Als ik terugkijk naar de keuzes van mijn beslissingsmethodieken ben ik trots op wat ik heb bereikt. Het toepassen van meerdere beslissingsmethodieken heeft enorm veel tijd gekost en heeft mij wat stress opgeleverd. Ik was bang voor tijdnood.

5.7.3. Opstellen van het adviesrapport

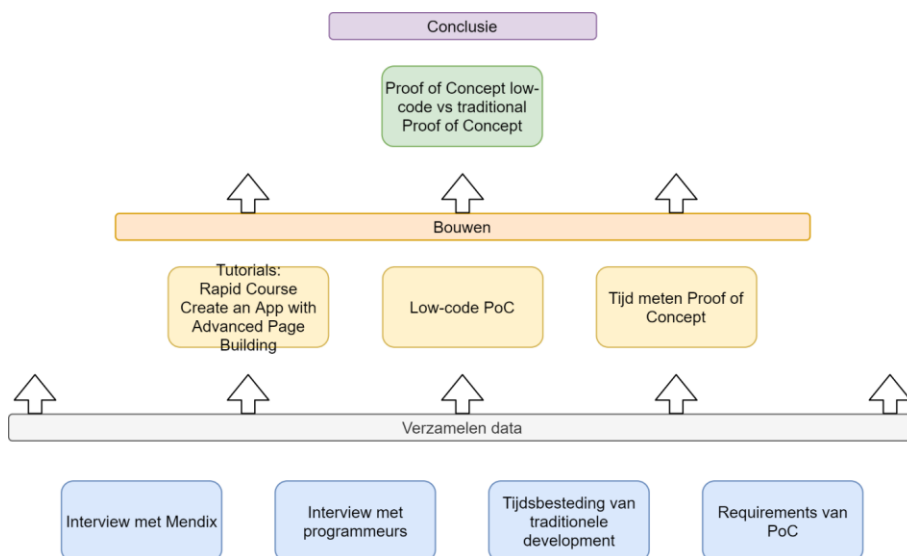
Tijdens het opstellen van mijn adviesrapport ben ik erachter gekozen dat mijn onderzoek een veel te grote rol zou spelen en dat ik er veel meer tijd aan zou besteden dan gepland. In mijn planning gaf ik aan dat het 6 weken zou duren. Als eerst had ik een week achterstand vanwege het opstellen van een plan van aanpak. Het maken van mijn adviesrapport heeft mij heel veel tijd gekost omdat ik een hoogkwalitatief product wilde aanleveren. Dit kwam grotendeels door mijn Nederlands, maar ook door het verzamelen van informatie. Voor het opstellen van het adviesrapport had ik 6 weken nodig, met alle werkzaamheden er naast (literatuuronderzoek, interviews etc.). Ik zou mijn adviesrapport klaar hebben in week 11, maar dit werd pas in week 16 definitief. Ik had problemen met de structuur van het rapport en Nederlands. Qua inhoud was het in week 14 goed zichtbaar. Het kiezen van een low-codeplatform was een heel onderzoek op zich. Het proces had korter gekund, maar daar zou kwaliteit onder leiden. Ik heb uiteindelijk voor kwaliteit gekozen, omdat ik het ontzettend belangrijk vond om mijn keuze goed te onderbouwen met betrouwbare informatie. Ook al zou dit langer duren.

6. Low-code ontwikkelsnelheid en kwaliteit

In dit hoofdstuk vergelijk ik traditioneel development met Mendix. In paragraaf 6.1 beschrijf ik het soort onderzoek. Daarna behandel ik hoe ik aan mijn informatie ben gekomen. Hier gaat het om de requirements waar de Proof of Concept aan moet voldoen. De expertinterviews die ik met ontwikkelaars heb gehouden zullen ook aan bod komen. Ook behandel ik de gesprekken met de leverancier. Eveneens moest ik achterhalen hoeveel tijd er aan de traditionele Proof of Concept is besteed. Tijdens het bouwproces stuitte ik nog op een aantal andere onderwerpen die in paragraaf 6.3 bespreek. De vergelijkingen tussen beide methodes worden benoemd in deel 6.4. Hierna kijk ik terug op het proces van dit onderdeel van mijn gehele onderzoek.

6.1. Soort onderzoek

Dit gedeelte van mijn onderzoek is een kwalitatief onderzoek met o.a. een literatuuronderzoek en expertinterviews over de kwaliteit van low-code. Het is een praktijkgericht onderzoek en als resultaat komen de verschillen tussen traditioneel development en het low-codeplatform Mendix naar voren. In Figuur 7 wordt in grote lijnen de route aangetoond die ik heb bewandeld om aan mijn resultaat te komen.



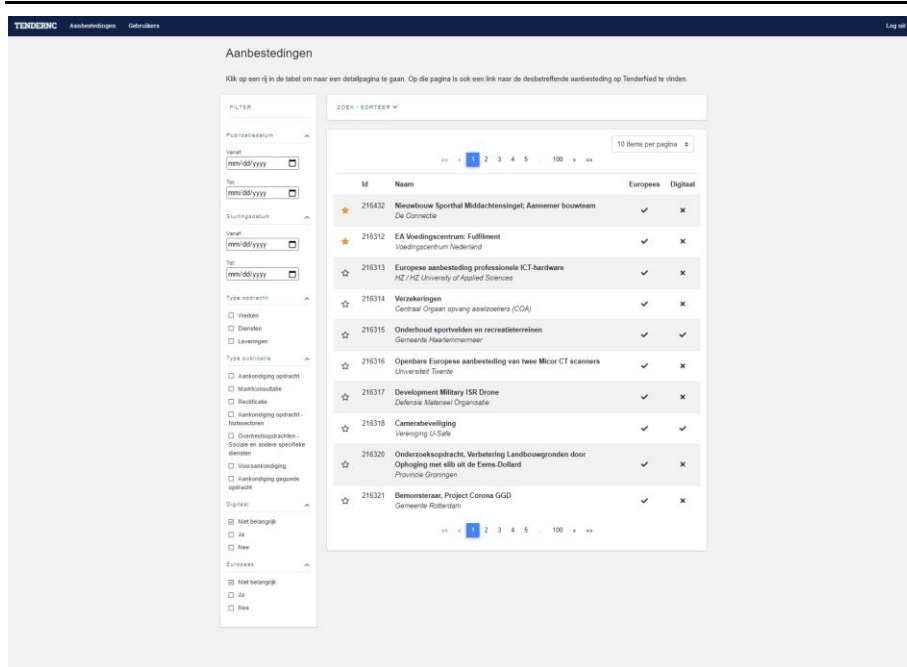
Figuur 7: Onderzoeksproces van het uitzoeken van de verschillen tussen Traditioneel development vs. low-code

6.2. Proof of Concept

In het verleden is er een traditioneel Proof of Concept gerealiseerd waar Netcompany de aanbestedingen van de website TenderNed ontvangt. In deze applicatie bevindt er een dashboard waar je deze aanbestedingen kan bekijken. Er is een mogelijkheid om deze te filteren en te zoeken op aanbestedingsnaam. Hiernaast is er een detailpagina waar nog meer informatie te vinden is. Op deze detailpagina is het mogelijk om naar de TenderNed pagina te gaan.

Als gebruiker is het mogelijk om een aanbesteding te kiezen als favoriet op de dashboard pagina. Je kunt je registreren met een gebruikersnaam en wachtwoord, zodat je op deze pagina kan inloggen.

In Figuur 8 is de dashboard pagina te zien. Voor de andere screenshots zie bijlage XIII.



Figuur 8: Dashboard pagina van het traditionele Proof of Concept

6.2.1. Requirements en omschrijving van Proof of Concept

Voordat ik kon beginnen met het realiseren van de low-code Proof of Concept moest ik eerst de eisen prioriteren. In overleg met mijn bedrijfsleider heb ik de requirements tot stand gebracht door naar het traditionele development Proof of Concept te kijken. Er waren namelijk geen requirements aanwezig van de originele applicatie.

De belangrijkste requirements zijn:

- Het kunnen inloggen;
- Het kunnen zien van de aanbestedingen;
- Het kunnen zoeken naar bijvoorbeeld aanbestedingsnaam;
- Een detailpagina tonen van een aanbesteding;

In bijlage XI zijn de requirements opgesteld die van toepassing waren voor de Proof of Concept.

Deze applicatie behoort simpel te zijn en moet overeenkomen met het originele. Dit betekent geen connectie met Active Directory of een koppeling met Azure DevOps voor de CI/CD pipelines. Dat maakt de tijd dat ik eraan heb besteed goed meetbaar voor hoofdstuk 6.2.3.

6.2.2. Bouwen van de proof of concept

In deze paragraaf besteed ik aandacht aan het platform Mendix. Ik behandel de stappen die ik heb gezet voor het realiseren van de applicatie en omschrijf de problemen die ik tegenkwam.

Domein model

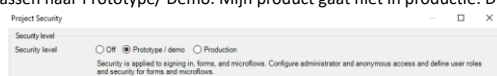
Als eerst heb ik de entiteit aanbesteding in het domeinmodel gebouwd. Ik heb gekozen om dit in dezelfde stijl te houden als traditionele table in de MSSQL-database. De validaties heb ik nog niet toegepast, omdat ik eerst de pagina's moest realiseren, zodat dit zou eventueel in de toekomst toegevoegd kunnen worden.

Genereren van de pagina's

Na het aanmaken van de aanbestedingsentiteit kon ik de pagina's genereren via het domein model. Dit is in principe een knop drukken. Mendix maakte dan automatisch een 'overview' folder aan met een pagina om een heel lijst aanbestedingen te zien. Dat werd vervolgens de Dashboard pagina. Nog een onderdeel was een dataview van één aanbesteding met een knop waarmee je een nieuwe aanbesteding kon maken. Deze pagina gebruik ik later als detail pagina. Op dit moment maakte ik alleen gebruik van het dashboard en heb ik de onnodige attributen verwijderd van de gegenereerde pagina.

Inloggen en gebruikersrollen

Mijn volgende stap was het inloggen met gebruikersrollen. Mendix heeft zijn eigen default login pagina. Om hier te komen moet de beveiliging van het project aangepast worden naar Prototype/ demo of Production. In Mendix wordt er met modules gewerkt. Dit houdt in dat je functionaliteiten kan splitsen in aparte delen. Per Module is het mogelijk om de beveiliging aan te passen. Zo is het mogelijk d een gebruikersrol toegang te weigeren voor een bepaalde module. Ik heb gekozen om het beveiligingsniveau aan te passen naar Prototype/ Demo. Mijn product gaat niet in productie. Dit is slechts een prototype.



Figuur 9: Beveiligingsniveau Proof of Concept in Mendix

Het is mogelijk om als demogebruiker in te loggen met een gegenereerd wachtwoord. Op het moment dat het project lokaal draait, is het mogelijk om hiermee in te loggen.

Gebruikersrollen

Gebruikersrollen worden toegepast op het moment dat het beveiligingsniveau op Prototype/ Demo of Productie staat. Wanneer het niveau werd aangepast, moest ik op een aantal plekken de beveiliging aanpassen. Het gaat om de volgende plekken:

- Module rollen
- Pagina toegang;
- Microflow toegang.

Ik heb gekozen om naast de default rollen mijn eigen rollen toe te voegen wat in tabel Tabel 10 is te zien.

Gebruikersrollen	Omschrijving	Onderbouwing
(Default) Administrator	Aangemaakt door Mendix	Nvt.
(Default) User	Aangemaakt door Mendix	Nvt.
NCUser	Heeft de mogelijkheid om de aanbestedingen te zien en alle pagina's. Ook is het mogelijk om Excel bestanden te importeren.	Niet iedereen hoeft data te importeren. Ik heb ervoor gekozen om mijzelf deze rol te geven, aangezien ik aan deze applicatie werk.
TenderNC	Heeft de mogelijkheid om alleen de aanbestedingen pagina's te zien.	Rol voor de product owner.

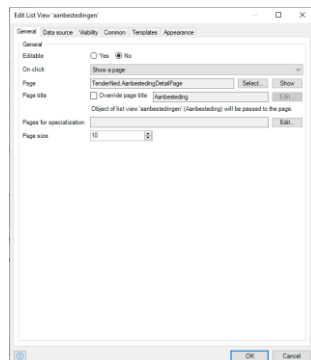
Tabel 10: Gebruikersrollen in Mendix

Importeren van data

Om data te tonen heb ik gekozen om eerst met mock data te werken, zodat ik kan zien hoe de informatie er uit ziet op de pagina. Ik heb gekozen om een Excel sheet te gebruiken van Tenedernd. Via een externe module kon ik dit importeren in mijn database. Met behulp van de gids (Mendix, 2021) heb ik deze module toegepast. Het standaardversie van Excel Importer heeft een maximale grootte van 5 MB en was genoeg voor deze situatie.

Doorgeven van data van pagina naar pagina

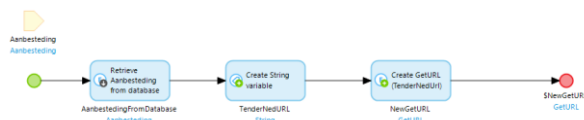
Na het importeren van de gegevens en dit in de Dashboard pagina te verwerken, heb ik gekozen om van een lijst naar één object te gaan. Dit kan met een *on click* event. Het is mogelijk om in Mendix de algemene eigenschappen aan te passen. In Figuur 10 is te zien hoe je een *on click* event kan toevoegen. Er zijn verschillende keuzes en de beste keuze op dit moment is het tonen van een pagina. De *show on page* toont een object wat in dat pagina geopend kan worden.



Figuur 10: Algemene eigenschappen van lijst aanbestedingen

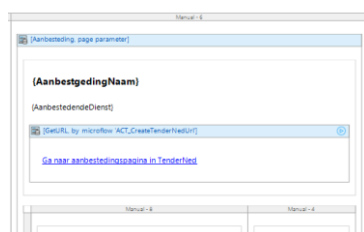
Externe website

In de detail pagina bevindt zich een link, waarmee de gebruiker naar de TenderNed pagina kan gaan van de aanbesteding. Hiervoor heb ik een *microflow* voor moeten gebruiken. Met een microflow wordt de logica van de applicatie gebouwd m.b.v. het uitvoeren van acties en het tonen van pagina's of het aanpassen van objecten. In Figuur 11 wordt het microflow in een happy flow getoond die ik heb gerealiseerd voor mijn Proof of Concept. Hier wordt getoond dat ik als parameter het object aanbesteding meegeef, wat de microflow ontvangt van het detail pagina.



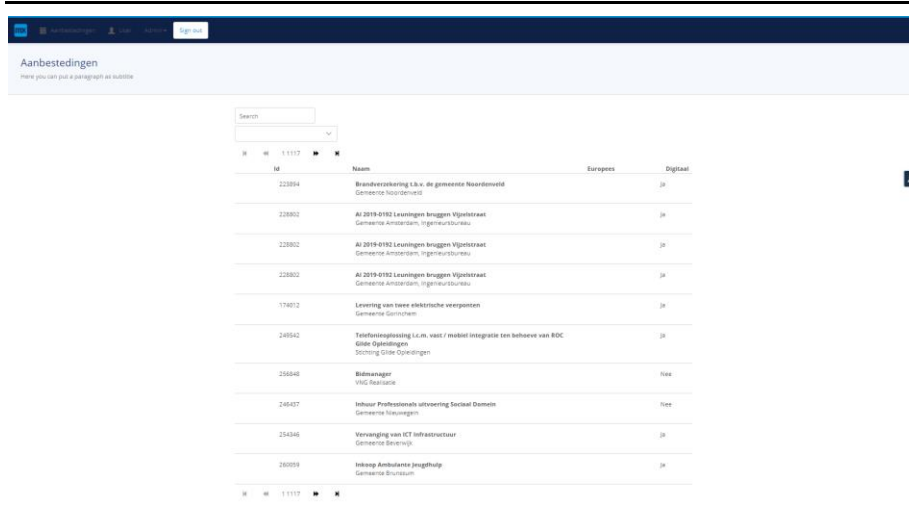
Figuur 11: Microflow voor het creëren van het TenderNed url

Dit betekent dat in de *dataview* van de aanbesteding een nieuwe *dataview* is toegevoegd met het ophalen van het goede URL (Figuur 12). In deze *GetUrl* data view heb ik een link knop toegevoegd die een nieuwe webbrowser opent met de TenderNed pagina. In de microflow zou ik nog een *if statement* toepassen op het moment dat er geen aanbesteding Id aanwezig zou zijn.



Figuur 12: Data view Aanbesteding heeft een data view genaamd GetUrl

In Figuur 13 is het uiteindelijke dashboard in Mendix te zien. Voor de rest zie bijlage XIV.



ID	Naam	Europees	Digitaal
22894	Brandverzekering i.o.v. de gemeente Noordenveld Gemeente Noordenveld		Ja
22892	Af 2019-0102 Levenslagen bruggen Vijverstraat Gemeente Amstelveen, Ingenieursbureau		Ja
22892	Af 2019-0102 Levenslagen bruggen Vijverstraat Gemeente Amstelveen, Ingenieursbureau		Ja
22892	Af 2019-0102 Levenslagen bruggen Vijverstraat Gemeente Amstelveen, Ingenieursbureau		Ja
174012	Levering van twee elektrische voertuigen Gemeente Groningen		Ja
24942	Telnettoegankelijk L.A. web 7 mobiel integratie een behouwe van BDC Globe Opleidingen Sociale Opleidingen		Ja
25548	Bidrukker VHG Realisatie		Nee
24547	Inhuur Professionalisme uitvoering Sociaal Domein Gemeente Nieuwegein		Nee
25436	Vervanging van ICT Infrastructuur Gemeente Breda		Ja
26059	Inhuur Ambulance Jeugdhulp Gemeente Breda		Ja

Figuur 13: Dashboord pagina in Mendix

6.2.3. Tijdsverschil tussen beide Proof of Concepts

De tijdsbesteding aan het traditioneel development is in onderdelen verdeeld. In bijlage XV wordt er per onderdeel beschreven wat het inhoudt en hoeveel uur eraan is besteed. Samen met mijn bedrijfsbegeleider heb ik de uren kunnen realiseren. Echter moet ik aangeven dat dit een schatting is. De start van de originele applicatie begon eerst als een uitprobersel, wat uit monde in een Proof of Concept en heeft een simpele opzet. Dit betekent geen Active Directory, complexe algoritmes en etc.

Low-code totaal (in uren)	Traditioneel development (in uren)
32 uur	160

Tabel 11: Totale besteding aan de proof of concepts

6.2.4. Discussie Proof of Concept

Het resultaat van het Proof of Concept is niet representeerbaar met gewone Netcompany projecten. Een project van Netcompany heeft meer complexiteit zoals koppelingen met o.a. Microsoft Active Directory, Azure DevOps en Microsoft SQL Server of PostgreSQL. Ik was daardoor genooddaakt te kijken naar ervaren ontwikkelaars die met low-codeplatformen hebben gewerkt.

6.3. Interviews met ervaren ontwikkelaars

Het interview met programmeurs was als doel om achter te halen hoe zij over low-code denken en traditioneel development. De sessie per persoon duurde ongeveer 30 minuten. Voor deze gesprekken heb ik gekozen voor een gestructureerde interview (bijlage XVI), omdat ik meer wilde weten over low-code en hun ervaringen. Achteraf gezien had ik ook meer vragen moeten stellen over traditioneel development, maar ik kon de gesprekken niet al te lang maken en moest mijn vragen prioriteren.

In totaal heb ik twee ontwikkelaars kunnen interviewen. Zelf had ik gehoopt dat ik meer gesprekken, maar daar stond de rest niet open voor. Een ontwikkelaar had het niveau senior en de andere junior/medior. Ik heb hiervoor gekozen om ook na te gaan of er tussen de niveaus verschillen aanwezig waren. In de volgende tabellen heb ik de conclusies beschreven per onderwerp:

- Conclusie met perspectief klant (Tabel 12);
- Conclusie met perspectief organisatie (Tabel 13);
- Conclusie over low-code in het algemeen (Tabel 14);
- Toekomst low-code (Tabel 15);

Conclusie interviews met perspectief klant:

Low-code is wel degelijk een product om de klant te helpen. Visueel is er gelijk veel te zien voor mensen die geen code kent, dus voor de klant.

Low-code kan goed werken op het moment dat dit gebouwd kan worden met als belangrijkste rol de analist. Dit betekent dat er bij de klant iemand aanwezig zou moeten zijn die heel goed weet wat zij willen en hoe het proces van de applicatie zou moeten werken. Dan zal het bouwen van de applicatie in low-code heel snel gaan.

Als een klant een project zou willen wat simpel is zonder complexiteit is het goed te creëren in low-code.

Tabel 12: Conclusie interviews met de perspectief klant

Conclusie interviews met perspectief organisatie:

Als een project maar 3 ontwikkelaars nodig heeft voor een project, in plaats van 6 zou dit in kosten schelen als het project draait in een happy flow. Dit betekent dat je minder ontwikkelaars nodig zou hebben om een applicatie te bouwen in low-code.

Een bedrijf als Netcompany zou hier wel degelijk een kans kunnen grijpen. Als een klein bedrijf graag met low-code zou willen werken is het i.v.m. de kosten geen oplossing.

Dit wil zeggen dat low-code werkt voor grote projecten en niet voor kleine projecten.

Tabel 13: Conclusie interviews met de perspectief organisatie

Conclusie interviews over low-codeplatformen algemeen

Als het complex is zoals met machine learning, wordt er geadviseerd om dit te doen met traditioneel development.

Low-code is gericht voor analisten en niet zozeer voor de ontwikkelaar.

Op het moment dat de werknemer niet uitkomt tijdens het realiseren van een applicatie is het niet mogelijk om dit makkelijk op internet te zoeken. Je moet naar de leverancier gaan voor de oplossing.

Op het moment dat je een vraag hebt voor de leverancier, bestaat er een kans dat je een paar uur moet wachten voor antwoord. Hier moet men rekening mee houden.

Je hebt weinig vrijheid, omdat de opmaat gemaakte componenten in de syntax moet zijn van het low-codeplatform.

Een mix met low-code en traditioneel development is mogelijk. Dat wil zeggen dat een frontend in een low-codeplatform en een .Net backend mogelijk is. Echter gebruik je het low-codeplatform niet optimaal op het moment dat men hiervoor gaat kiezen.

Tabel 14: Conclusie interviews over low-codeplatformen in het algemeen

Toekomst low-code in de interviews

Het is een programma wat ook in de toekomst aanwezig zal zijn, omdat het goed werkt voor de klant en is sneller vergeleken traditioneel development.

Ook is dit een oplossing voor bedrijven die moeite hebben met het vinden van ontwikkelaars. Bedrijfsanalisten kunnen namelijk ook aan de slag met het low-codeplatform.

Tabel 15: Toekomst van low-code in de interviews

6.4. Literatuuronderzoek naar bekende problemen van low-code

In (Virta, 2018) waren er wat nadelen aanwezig voor low-code in 2018, waar Mendix bewust van was. Waarschijnlijk andere leveranciers ook. Wel is te zien dat ze in 3 jaar degelijk hebben gewerkt aan hun zwaktes.

In de interviews uit hoofdstuk 6.3 wordt er aangegeven dat low-code wel degelijk een product is wat men kan gebruiken. Dit is mogelijk wanneer het een eenvoudige applicatie is, zonder enige complexiteit, wat overeenkomt met (Virta, 2018). Een voorbeeld van complexiteit is het gebruik van machine learning in het project. Echter is er geen ruimte en zicht om welke complexiteiten het allemaal gaat, dit moet nog worden geanalyseerd.

Volgens de interviews met ontwikkelaars werkt low-code werkt Low-code omde o.a de volgende redenen, (Virta, 2018):

- Snelheid in ontwikkelen door de tools te gebruiken;
- Visueel kan er eerder iets getoond worden;
- Minder ontwikkelaars nodig voor het bouwen van een applicatie;
- Simple procesautomatisering Figuur 11.

Dit te hebben gezegd zijn de voordelen die benoemd worden een feit. Zowel in literatuur als voor de interviewers in het artikel.

Wat ook ten tijde ter sprake kwam in 2018 was het voordeel van gebrek aan unit testen (Virta, 2018). Ter verduidelijking: dit is een voordeel wanneer je zelf geen workflow aanmaakt. Mendix biedt het volgende aan wanneer testen cruciaal wordt voor het project:

- Microflows testen met Unit Testing Module
Om te verifiëren dat de geïmplementeerde flow werken is het mogelijk om unit testen te creëren. De module maakt gebruik van JUnit.
- Application Test Suite (ATS)
Een hulpmiddel voor het testen van geïntegreerde systemen (elektronisch systeem)
- Het testen van Web Services met SoapUI
Dit kan op het moment dat er webservices aanwezig zijn in je project. Hiermee maak je (geautomatiseerde) testen.
- Selenium IDE
Selenium IDE is een Firefox-plugin die gebruikersinteracties met de browser registreert en afspeelt.
- Automated testen met TestNG
Is een Java test framework dat kan worden gebruikt om Selenium aan te sturen.

Volgens (Virta, 2018) zijn dit de meest genoemde nadelen van low code:

- Inefficiëntie van de solutions;
- Grote prestatieproblemen bij grote datavolumes;
 - Het gebrek om database-aanroepen te bulken (bulkify);
- Onderhoudsproblemen met complexe solutions;
- De onbekendheid van de aangeboden tools.

(Virta, 2018)De grootste nadeel van low-code is de prestatie. Het gaat om het leveren van een groot hoeveelheid data, wat hand in hand gaat met onderhoudsproblemen. Dit kan ook gerelateerd zijn met het feit dat de gebruiker het platform gebruikt zonder enige algemene begrip van het woord software-engineers.

In Mendix is het mogelijk om de performance van de applicatie te monitoren en te beheren. Op het moment dat men met Mendix Cloud werkt, is het mogelijk om de standaard statistieken te gebruiken. Ook is het mogelijk om andere deployment opties te gebruiken. Zij zijn bewust waar de problemen aanwezig zijn tijdens het ontwikkelen. Het gaat om de volgende punten:

- UI-centric;
Dit betekent een langzame gebruikersinterface, wat een langzame microflow kan zijn aangeroepen of aan te veel groot aantal oproepen door de gebruikersinterface.
- Microflow-centric.
Bij dit probleem zijn er een aantal handelingen zoals het gebruiken van *breakpoints* zodat je de microflow kan debuggen.

Hiernaast biedt de leverancier verticale en horizontale schalen aan. Schaalbaarheid zorgt ervoor dat de prestaties constant zijn, ongeacht hoeveel mensen binnenkomen. In bijlage XVII worden specificaties getoond wat Mendix levert. Zij leveren verschillende containers per omgeving. Dit kan een grootte zijn van XS tot een XL. In Tabel 16 worden de belangrijkste specificaties getoond van het grootte XS.

XS	
Mx Memory	1 GB
Mx vCPU	0,25
Database Memory	1 GB
Database vCPU	2
DB Storage	5 GB
File storage	10 GB
Network	Onbeperkt
Encryption at rest	✓

Tabel 16: Container XS die Mendix aanbied voor het schalen

Tijdens een voortgangsgesprek met de productowner kwam het onderwerp vendor lock-in voor. Dit maakt een klant afhankelijk van een leverancier voor producten en diensten, omdat hij niet in staat is om van leverancier te veranderen zonder substantiële omschakelingskosten of ongemak.

Als oplossing biedt Mendix de volgende oplossing aan: het exporteren van je applicatiemodellen en je data. Door het Mendix software development kit (SDK) te gebruiken is het mogelijk om je applicatie modellen naar een alternatieve low-code platform te migreren of een platform zoals Java en Hibernate. Deze mogelijkheid heb ik zelf niet kunnen uitoefenen. De data kan je met behulp van een backup downloaden inclusief de bestanden die de applicatie vormen. Dit betekent dat je een nieuwe database zou kunnen creëren met het gedownloade data of converteren naar een ander database.

Het probleem met low-code is dat je de leverancier wel degelijk nodig hebt tijdens het ontwikkelen van je applicatie. Low-code is moeilijk te vinden op internet. Leveranciers bieden een gratis gemeenschap aan waar je je vragen kan stellen. Echter komt erop neer dat je afhankelijk bent van hen bent. Het is mogelijk om via tutorials de basis van het programma te leren. Maar wanneer je als bedrijf meer geavanceerde technieken nodig hebt dan is de kans groot dat je de hulp van de leverancier moet inschakelen. Een mogelijke oplossing is je medewerkers de tutorials laten volgen om certificaten te halen die Mendix overhandigen. Als bedrijf moet je je afvragen of je vendor lock in kan accepteren.

6.5. Roadmap

Voordat men met low-code aan de slag kan gaan zijn er wat aandachtspunten waar een organisatie over na moet denken. Als eerst moeten ze zich afvragen voor welk doel zij low code willen gebruiken. Een belangrijke reden om het te gebruiken, vanuit het oogpunt van de klant, is de betaalbaarheid op lange termijn en de onderhoudbaarheid volgens (Virta, 2018). De klant ziet namelijk een snellere softwareontwikkeling. Ook zijn de kosten goedkoper als het probleem van het klant opgelost moet worden.

Ten tweede moet er gekeken worden of low-code geschikt is voor de applicatie door in kaart te brengen hoe complex de applicatie moet zijn. Denk aan te veel opmaat gemaakte componenten en veel gebruik van machine learning.

Ten derde moet er nagedacht worden welk vorm van cloud computing de applicatie gaat gebruiken. Mendix biedt de volgende versies aan: public cloud, *private* cloud, on premises. Wanneer er voor on premises wordt gekozen, zal het one-click deployment wegvallen. De one-click deployment betekent dat Mendix het gedeelte deployment pipeline en application operation geautomatiseerde processen zijn. Jij als klant hebt hier geen omkijk naar.

En ten vierde moet er in het begin van het project goed nagedacht worden over het ontwerp van het systeem. Dit vereist de 'best practices' van software engineering. Ook moet er rekening worden gehouden hoe je low-code gebruikt. Volgens (Rymer J. , 2018) is het proces van de applicatie een cruciaal. De bedrijfsanalist wordt namelijk de belangrijkste speler, want zij beschrijven hoe de applicatie zou moeten werken.

6.6. Verantwoording en reflectie ontwikkelsnelheid en kwaliteit

Het bouwen van het Proof of Concept was complex vanwege tijdsdruk.

Als eerst heb ik de evaluation guide gescand. Dit is een gids met evaluatie criteria wat mij zou kunnen helpen om te kiezen van een platform. Dit heb ik van de platform leverancier gekregen. Daarentegen zocht ik meer naar een hands-on tutorial en kwam ik uit bij de learningpaths van Mendix. Dit zijn tutorials waar je stapsgewijs Mendix kan leren. De learningpaths zijn ingedeeld per niveau, rollen en onderwerpen. Als een beginner is het mogelijk om te focussen op de basis. Er is ook niveau rapid aanwezig en dat is bedoelt voor developers die zo snel mogelijk aan de slag wilt gaan.

Ik heb de Crash Course van Mendix tot de helft gevolgd en mijn aanpak was de cursus naast mijn proof of concept realiseren. Dit betekent dat ik alles door elkaar deed, wat chaos en stress opleverde. Hierdoor kreeg ik o.a problemen bij het inloggen en registreren van gebruikers, omdat dit niet beschreven stond in de Crash Course.

Op 14 april 2021 heb ik een kort wekelijkse demo van Mendix bijgewoond. Het was een korte presentatie over Mendix en het manier van werken. Zelf had ik verwacht dat het meer zou gaan over hoe je in Mendix een applicatie moest bouwen, maar daar was een half uur veel te kort voor.

Uiteindelijk ben ik van aanpak gaan veranderen. Ik heb via Udemy een cursus gekocht met de naam Mendix: Low-Code Application Development Course. Mendix zelf biedt ook cursussen aan, maar ik moest handelen omdat ik tijd tekort kwam. Het was al 30 april 2021 en als ik zo door zou gaan, redde ik het niet. Hierdoor had ik besloten om de hele cursus te volgen en even de Proof of Concept lost te laten. In bijlage XVIII wordt er getoond dat ik het certificaat op 8 mei 2021 heb behaald. De cursus gaf een basis om zo aan mijn Proof of Concept te werken.

7. Conclusie & discussie

In dit onderzoek is gezocht naar een antwoord op de hoofdvraag: *‘Kan door middel van een low-code platform sneller en efficiënter worden ontwikkeld zonder kwaliteitsverlies ten opzichte van traditioneel development en welk low-codeplatform past het beste bij Netcompany?’* De resultaten van de Proof of Concept tonen aan dat met low-code efficiënter ontwikkeld kan worden en met dezelfde kwaliteit als het concept ontwikkeld met traditioneel development. Daarnaast blijkt uit dit onderzoek dat Mendix het beste aansluit bij de wensen en eisen van Netcompany.

Voor het kiezen van het low-codeplatform heb ik de SFA-Matrix gebruikt. Een SFA-matrix wordt vaak gebruikt voor marketing. Eén van de definities van marketing is “alles wat een bedrijf doet om de verkoop van producten of diensten te bevorderen”. Dit betekent dat dit onderzoek Netcompanies diensten kan bevorderen door het juiste low-codeplatform te kiezen. Door de kaders van mijn onderzoek heb ik slechts 1 interview gehouden, er zou meer werknemers voor van Netcompany geïnterviewd moeten worden. De SFA-matrix wordt representatiever op het moment, dat er meer interviews worden gehouden.

Ik heb gemeten dat het bouwen van de applicatie sneller is in Mendix. Hoewel dit een goede indicatie is, zijn er omwille van de tijd ook concessies gedaan. De Proof of Concept is niet een volledig representatieve applicatie voor Netcompany. Er missen bepaalde functionaliteiten, die veel voorkomen in projecten van Netcompany. Hier gaat het bijvoorbeeld om de koppelingen met externe database, Azure DevOps en Microsoft Active Directory. Er is een kort onderzoek verricht en deze componenten bestaan, maar het is onbekend hoelang het implementeren duurt. Ook heb ik geen custom componenten toegevoegd, omdat dit niet van toepassing was voor de applicatie. Dit is echter wel een belangrijk aspect in softwareontwikkeling en het is interessant om hier vervolgonderzoek naar te doen.

Low-code heeft veel voordelen volgens literatuuronderzoek en ontwikkelaars. De snelheid in ontwikkelen door de tools te gebruiken is een groot verschil. Het resultaat tussen de beide Proof of Concepten toont aan dat in 32 uur je een applicatie kan tonen in low-code, terwijl de traditionele een respectievelijke 160 uur nodig heeft (hoofdstuk 6.2). Volgens ontwikkelaars en literatuur is low-code een product dat voor de klant goed werkt, omdat er visueel snel wat getoond kan worden. Ook zijn er minder ontwikkelaars nodig voor een project. Voor een low-code project zou de helft van ontwikkelaars nodig zijn vergeleken bij een traditionele development project (hoofdstuk 6.4).

In 2018 was het grootste nadeel van low-code de prestatieproblemen, waar leveranciers bewust van zijn. Zij leveren informatie hoe je het beste deze problemen kan opsporen, maar deze prestatieproblemen kunnen ook ontstaan doordat de gebruikers geen kennis hebben van de 'best practices' van het algemene begrip software engineering.

De term vendor lock-in is ook bekend bij low-code leveranciers. Zij bieden een oplossing ervoor, maar er komt meer bij kijken bij deze dilemma (zie hoofdstuk 6.4). Tijdens geavanceerde technieken bij het platform, is de kans groot dat men de leverancier moet inschakelen voor ondersteuning. De mogelijkheid om de juiste oplossing te vinden op internet is klein. Echter is de steun van de leverancier aanwezig bij verschillende abonnementen (hoofdstuk 6.3).

Uit dit onderzoek is gebleken dat Mendix en low-code geschikt zijn voor gebruik binnen Netcompany. Dit hoeft echter niet te betekenen dat Netcompany traditioneel development uit hoeft te sluiten. Dit zullen waarschijnlijk projecten zijn waar low-code beter tot zijn recht komt en projecten waar traditioneel development noodzakelijk is.

8. Evaluatie

Het proces vanaf de eerste 3 weken gingen heel vloeiend, vooral door mijn motivatie en enthousiasme om dit voor het bedrijf te onderzoeken. Ik stuitte op problemen bij het literatuuronderzoek, en ook omdat ik kwaliteit wou leveren. Dit betekende dat ik meer tijd moest besteden om de juiste artikelen te zoeken, zodat ik dit voor mijn onderzoek kon gebruiken. Hiernaast had ik niet geanticipeerd, dat het maken van een adviesrapport over het kiezen van een low-codeplatform veel tijd zou kosten. Al vanaf het bedrijfsbezoek liep ik achter op mijn schema en had ik mijn proces van het beantwoorden van de deelvragen sneller verwacht welke ik voor het adviesrapport zou gebruiken. Het adviesrapport dat ik heb opgeleverd is van kwaliteit en hier ben ik ook erg tevreden over. Het proces van het maken van het low-code 'Proof of Concept' heeft mij veel moeite gekost. Men zegt dat de leercurve van low-code sneller is, maar daarvoor heb je wel de juiste tools voor nodig. Zoals Mendix aan gaf: "iedereen kan leren autorijden, maar je hebt de juiste tools voor nodig om het goed te doen.". Deze tools miste ik in het begin een beetje en tijdens het ontwikkelen stuitte ik ook op het probleem dat er een veel minder grote community is. Dit maakt het zoeken naar oplossing voor problemen soms lastiger, vergeleken met traditioneel development.

Ondanks mijn overall-planning heeft het thuiswerken de situatie sterk beïnvloed, maar ook het feit dat problemen die je tegenkomt in het low-codeplatform lastig op te lossen zijn. De low-codeleverancier zorgt ervoor dat bij het bouwen van uitdagende applicaties hun support nodig heb. Dit betekent dat ik hen MOET vragen om iets op te lossen, of juist betaalde cursussen moet volgen. Ondanks dat de leercurve steiler was dan ik dacht, ben ik wel trot op het feit dat ik in een erg korte tijd, een Proof of Concept heb kunnen opleveren. Het is soms erg moeilijk om binnen de kaders van het onderzoek te blijven, en ik wil altijd graag meer doen. Toch vind ik dat de opgeleverde applicatie een erg goede indruk geeft over de ontwikkelsnelheid in low-code

Aan het begin was ik erg gefocust op het gebruiken van mijn dagelijkse planning, echter op het moment dat ik aan mijn adviesrapport werkte, viel dit weg. Dit was ook het moment dat ik begon te stressen, vooral omdat ik geneigd was om op kleine details te concentreren.

Al met al, heb ik genoten van mijn afstudeeropdracht. Ik ben erg tevreden over het eindresultaat van het adviesrapport en vooral de samenwerking met de collega's.

8.1. Product reflectie

Voor dit onderzoek moest ik antwoord hebben op de hoofdvraag: 'Kan door middel van een low-code platform sneller en efficiënter worden ontwikkeld zonder kwaliteitsverlies ten opzichte van traditioneel development en welk low-codeplatform past het beste bij Netcompany?' De resultaten van de Proof of Concept tonen aan dat met low-code efficiënter ontwikkeld kan worden en dat het dezelfde kwaliteit behoudt als een concept dat ontwikkeld is met traditioneel development. Daarnaast blijkt uit dit onderzoek dat Mendix het meest voldoet aan de wensen en eisen van Netcompany.

Zoals ik in hoofdstuk 7 aangeef heb ik een aantal opmerkingen over de kwaliteit van mijn product:

- Om de SFA-matrix representatiever te maken zou er eventueel meer werknemers van Netcompany geïnterviewd worden;
- Ook is de Proof of Concept niet een volledig representatief applicatie voor Netcompany. Er missen namelijk een aantal functionaliteiten die veel voorkomen bij Netcompany die niet aanwezig waren bij de Proof of Concept. Er is een kort onderzoek naar verricht naar deze functionaliteiten en deze functionaliteiten zijn wel aanwezig bij Mendix.

Als ik naar de behaalde resultaten en alle bijlages kijk die ik heb toegevoegd, vind ik dat ik veel werk heb verricht. Toch mis ik een aantal onderdelen die ik in mijn afstudeerplan had beschreven. Het gaat om de volgende onderdelen:

- Testplan;
- Adviesrapport over de low-code ontwikkelsnelheid en efficiëntie.

Voor het kiezen van de juiste low-codeplatform heb ik in totaal 15 weken besteedt. Dit te zeggen, was het niet mogelijk om nog een adviesrapport te realiseren voor Netcompany. In overleg met mijn bedrijfsbegeleider mocht

ik dit vervallen. Alsnog waren de onderbouwingen die ik nu mijn scriptie heb geplaatst zeer essentieel. Hiermee kon ik aantonen dat ik wel degelijk een goed product over low-code ontwikkelsnelheid en efficiëntie (hoofdstuk 6) heb geleverd.

Het onderzoek kon ik zo breed maken als ik wilde, maar ik had tijdnoed. Het afbakenen was lastig, omdat ik gelijk alles zou willen doen. Hiermee bedoel ik een heel project opzetten met een testplan en een connectie maken met alle externe software. In theorie wat ik afstudeerplan beschreef was het mogelijk om alle producten te leveren, maar in praktijk ging het toch anders.

8.2. Verantwoording beroepstaken

8.2.1. A1 Analyseren probleemdomein & opstellen probleemstelling

Deze taak was van belang tijdens het oriënteren van de opdracht. Hiervoor had ik een plan van aanpak gemaakt en dit besproken met mijn begeleider van werk en school. Hier werd een lijstje met belangrijke punten opgesteld om zo het project duidelijk neer te zetten en mijn aanpak kon bepalen. In het plan van aanpak geef ik de projectopdracht aan, toon ik mijn werkwijze en de opgeleverde voorwaarden van het project zodra het afgerond is.

A1 Analyseren probleemdomein & opstellen probleemstelling	
Plan van aanpak	Bijlage II

8.2.2. A2 Informatie vergaren, analyseren, beoordelen en verwerken

Deze taak heb ik gedurende mijn hele scriptie uitgevoerd. Het gaat om de volgende taken:

- Tutorials;
Om Mendix te begrijpen heb ik een Udemy tutorial gevolgd en tutorials die Mendix heeft aangeboden.
- Literatuuronderzoek;
Voor het kiezen van het low-code platform heb ik naar wetenschappelijke artikelen, scripties en rapporten gezocht. Hiernaast heb ik literatuur moeten zoeken over de oplossingen van Mendix en de voor- en nadelen van low-code.
- Architect overleg;
Dit gaat over het eisenpakket. Dit gesprek heb ik met de architect manager besproken.
- PO overleg;
Het wekelijkse overleg met de productowner.
- Interview met Sales.
Het achterhalen welke eisen zij hebben voor low-code en of low-code een eis is bij aanbestedingen.
- Interviews senior/ junior software developers;
Informatie winnen over low-code en traditioneel development.
- Bedrijfs informatie over Mendix, korte demo;
Gesprekken gevoerd met een medewerker van Mendix hoe ik het beste zou kunnen beginnen met Mendix en extra informatie zoals het horizontale en verticale schaling.
- Bedrijfsinformatie over Outsystems inclusief een demo dag
Tijdens mijn onderzoek was ik bezig met het vergelijken van prijzen van verschillende leveranciers en had ik besloten om ze te bellen. Dit leidde naar twee gesprekken met Outsystems en een demo middag.

A2 Informatie vergaren, analyseren & verwerken	
Het uitvoeren van brononderzoek	Hoofdstuk 5.2 en zie bijlage III
Het kiezen van de juiste informatie	Hoofdstuk 5.2, 6.4, bijlage IV
Het analyseren welk beslissingsmethodieken van toepassing is voor dit onderzoek	Hoofdstuk 5.3 en bijlage X
Het houden van gebruikers-/ stakeholdersinterviews	Hoofdstuk 5.4.1
Het interviewen van ontwikkelaars	Hoofdstuk 6.3 en bijlage XVI
Het vergaren van informatie voor low-code ontwikkelsnelheid en kwaliteit	Hoofdstuk 6.4, bijlage III, IV

Informatie van Mendix	Bijlage XVII
Tutorial certificaat	Bijlage XVIII

8.2.3. B1: Gemotiveerd selecteren van ICT gerelateerde oplossingen

Tijdens het vergaren van informatie en analyseren met behulp van interviews en artikelen is er een lijst opgesteld van kenmerken. De lijst van kenmerken is geprioriteerd met de eisen conform de MoSCoW-methode. Om de juiste low-codeplatformen te kiezen heb ik 4 low-codeplatformen gekozen om te onderzoeken. Dit heb ik voorgelegd aan de productowner. Het waren de 4 leiders, volgens het rapport van Gartner. Oorspronkelijk waren dit er 3, maar ik heb low-codeplatform Salesforce toegevoegd. De reden hiervoor is omdat Netcompany hier graag mee wil werken. Dit betekent dat de vestiging in Denemarken zich aan het verdiepen is naar low-code en expliciet naar het platform Salesforce. Als oplossingen heb ik drie alternatieven gepresenteerd. Dit waren het resultaat uit het beslissingsmodel, -boom en de scorematrix in combinatie met de SFA-matrix.

B1 Gemotiveerd selecteren van ICT gerelateerde oplossingen	
Het gebruiken van de MoSCoW methode	Hoofdstuk 5.4.3, 5.5.1, bijlage VI, VII.
Mijn selectie pakket van eisen van Netcompany	Hoofdstuk 5.4.3, bijlage VI, X
Het SFA-matrix	Hoofdstuk 5.3.3, bijlage XIX
De scorematrix	Hoofdstuk 5.3.3, bijlage VIII, X
Het beslissingsmodel	Hoofdstuk 5.3.1, bijlage X
De beslissingsboom	Hoofdstuk 5.3.2, bijlage X

8.2.4. B3 Adviseren over inrichting van ICT gerelateerde oplossingen & processen

Deze beroepstaak is toegepast tijdens het zoeken naar de juiste low-code. Hiervoor heb ik een adviesrapportage voor gemaakt. Hier wordt in mijn conclusie beschreven dat Mendix het juiste platform is voor Netcompany. Hiernaast heb ik een conclusie getrokken in mijn onderzoek of low-code wel degelijk een optie is voor Netcompany.

B2 Adviseren over inrichting van ICT gerelateerde oplossingen & processen	
Het opstellen van een adviesrapport	Hoofdstuk 5, bijlage X
Mijn conclusie of low-code gebruikt kan worden	Hoofdstuk 7

8.2.5. C1 Ontwerpen software

Deze taak heb ik toegepast tijdens het bouwen van het 'Proof of Concept'. Het low-codeplatform maakt gebruik van applicatie logica. Deze bestaat uit microflows welke gebaseerd zijn op een BPMN-model.

C1 Ontwerpen software	
Domein model in Mendix	Hoofdstuk 6.2, bijlage XIX

8.2.6. D1 Realiseren van software

Het realiseren van software is toegepast tijdens bouwen van het Proof of Concept en het tutorial die ik heb gevolgd. Dit heeft een externe module gebruikt voor het importeren van een Excel bestand. Hiernaast heb ik opmaat gemaakte microflows en pagina's gerealiseerd. Door de tutorial weet ik hoe ik een for-loop en een if-statement kan gebruiken, en gebruikersrollen instelt. In het volgende tabel toon ik alle taken aan die ik gebruikt om software te realiseren.

D1 Realiseren van software	
Custom CSS	Hoofdstuk 6.2
If-statement implementeren	
Een webapplicatie bouwen	
Het gebruiken van een externen module	
Het gebruiken van de Mendix IDE	

8.2.7. Gc Kritisch, onderzoekend & methodisch werken

Deze taak was vooral van belang bij mijn voorbereidingsfase voor het zoeken naar het juiste low-codeplatform. Vooraf heb ik een zoekplan opgesteld hoe ik dit zou aanpakken. Hiervoor heb ik requirements opgesteld voor artikelen, scripties en rapportages. Tot slot heb ik een aantal artikelen moeten laten vervallen en heb ik de belangrijkste gekozen.

Gc Kritisch, onderzoekend & methodisch werken	
Zoekplan	Hoofdstuk 5.2.1, bijlage III
Kiezen van artikelen	Hoofdstuk 5.2.2, bijlage IV

8.2.8. Gf leren: voorbereiden op de volgende studiefase & beroep

Deze beroepstaak was toegepast bij mijn werkzaamheden en keuzes die ik heb ondergaan tijdens dit traject. Dit heeft mijn goed voorbereid op de volgende fase in mijn studieveld en toekomstige carrière. In de volgende tabel worden mijn werkzaamheden getoond.

Het volgen van een tutorial over Mendix in Udemy	Bijlage XVIII
Het opstellen van een plan van aanpak en het afbakenen er van	Bijlage II
Gesprekken houden met Mendix en demo volgen	Bijlage XVII
Het zoeken naar de juiste artikelen voor mijn onderzoek	Bijlage III

Bibliography

- Benders, L. (2020, Januari 20). *Een perfect adviesrapport (met template en voorbeeld)*. Retrieved from Scribbr: <https://www.scribbr.nl/stage/schrijf-een-perfect-adviesrapport/>
- Bijdragers aan Wikimedia projecten. (2004, Januari 24). *internationaal samenwerkingsverband van nationale organisaties die standaardnormen vaststelt in meer dan 160 landen*. Retrieved from Wikipedia.org: https://nl.wikipedia.org/wiki/Internationale_Organisatie_voor_Standaardisatie
- Farshidi, S., Jansen, S., & Fortuin, S. (2021). Model-driven development platform selection: four industry case studies. *Springer*, 1-27.
- ISO. (n.d.). *The ISO/IEC 25000 series of standards*. Retrieved from ISO 25000: iso25000.com
- Kumar, V. (2021, Januari 07). *Wat is Power apps? - Power Apps*. Retrieved from Microsoft.com: <https://docs.microsoft.com/nl-nl/powerapps/powerapps-overview>
- Methods*. (2018, Februari 9). Retrieved from ICTResearchMethods: <http://ictresearchmethods.nl/Methods>
- Ngih, M. (2020, Maart 6). *Complete Introduction to the Low-Code Ecosystem*. Retrieved from Medium: <https://medium.com/swlh/complete-introduction-to-the-low-code-ecosystem-defb9b260d20>
- Reilly, K. (2019, Augustus 24). *What is Low-Code Development?* Retrieved from Medium: <https://medium.com/swlh/what-is-low-code-development-f45550c3243d>
- Rymer, J. R. (2017). *The Forrester Wave™: Low-Code Development*. Cambridge: Forrester.
- Rymer, J. R., Mines, C., Vizgaitis, A., & Reese, A. (2017, Oktober 23). *The Forrester Wave™: Low-Code Development Platforms For AD&D Pros, Q4 2017*. Cambridge: Forrester. Retrieved from Forrester.
- Rymer, J., Mines, C., Vizgaitis, A., & Dostie, P. (2017, Oktober 23). *The Forrester New Wave™: Low-Code Platforms For Business Developers*. Retrieved from Forrester: <https://www.forrester.com/report/The+Forrester+New+Wave+LowCode+Platforms+For+Business+Developers+Q4+2017/-/E-RES140271>
- Sattar, N. (2018). Selection of low-code platforms based on organization and application type. *Selection of low-code platforms based on organization and application type*. Lappeenranta, Finland: Technische Universiteit Lappeenranta .
- Software development security*. (2020, Oktober 21). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Software_development_security
- Studio61. (2020, Maart 20). *Why low code development is accelerating business success and driving digital disruption*. Retrieved from ZDNet: <https://www.zdnet.com/paid-content/article/why-low-code-development-is-accelerating-business-success-and-driving-digital-disruption/#:~:text=Low%20code%20enables%20the%20rapid,agile%20approach%20to%20software%20development.>
- Swaen, B. (2013, Oktober 10). *Wat is kwalitatief en kwantitatief onderzoek?* Retrieved from Scribbr: <https://www.scribbr.nl/onderzoeksmethoden/kwalitatief-vs-kwantitatief-onderzoek/>

-
- Vincent, P., Natis, Y., Iijima, K., Wong, J., Ray, S., Jain, A., & Leow, A. (2020). *Magic Quadrant for Enterprise low-code applications platforms*. 30: September.
- Virta, T. (2018). Relation of low-code development to standard software development. *Relation of low-code development to standard software development*. Lappeenranta, Finland: Lappeenranta University of Technology.
- What is Low-Code?* (2020, Oktober 29). Retrieved from Mendix: <https://www.mendix.com/low-code-guide/>
- Wikipedia Contributors. (2021, Januari 19). *Model-driven engineering*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Model-driven_engineering
- Wikipedia Contributors. (2021, Januari 19). *Model-driven engineering*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Model-driven_engineering

Bijlagen

I. Afstudeerplan



Bijlage_1_Afstudeer
plan_PMM_1710098

II. Plan van Aanpak



Bijlage_2_PlanVanA
anpakV0.4.docx

III. Zoekplan



ZoekplanV0.3.docx

IV. Samenvatting literatuuronderzoek



Bijlage_4_Samenvat
ting literatuuronder

V. Productowner generieke lijst



Bijlage_5_Azad_Eisen
Per MDD platform

VI. Selectie pakket Netcompany



ConceptKenmerken
LCAP.xlsx

VII. Rapport beslissingsmodel (Farshidi, Jansen, & Fortuin, 2021)



Bijlage_7_Descision
Model1Concept.pdf

VIII. Scorematrix



Bijlage_8_Eisen Per
MDD platform.xlsx

IX. Voor- en nadelen van low-codeplatformen

Internal	Strengths <u>Mendix Gartner Report 2020</u> - Product: Mendix biedt robuuste mogelijkheden om tegemoet te komen aan behoeften op het gebied van integratie, workflow, gebeurtenisverwerking en het gebruik van AI om ontwikkeling te ondersteunen. - Innovatie: de onlangs uitgebrachte Mendix Data Hub-component stelt zijn ontwikkelaars in staat om gemakkelijker gegevens en zakelijke mogelijkheden te ontdekken, samen te stellen en te orkestreren, met behulp van dezelfde Mendix visuele modelleringstools die het gebruikt om bedrijfslogica en applicatiedatamodelen te bouwen. - Marktbegrip: het LCAP van Mendix ondersteunt visuele en modelgestuurde ontwikkeling in een breed spectrum van ontwikkelaarsprofielen. Het platform biedt een aparte design-time omgeving voor citizen developers (Mendix Studio) en een robuustere ontwikkelomgeving voor professionele developers (Mendix Studio Pro).	Weaknesses <u>Mendix Gartner Report 2020</u> - Verkoopuitvoering en prijsstelling: Mendix verlegde zijn verkoop naar het segment van de grote ondernemingen. Het heeft het daar goed gedaan, maar de algehele klantengroei is niet zo hoog als die van andere leveranciers in deze snelgroeiende markt. - Industriestrategie: Mendix is laat met het aanbieden van branchespecifieke oplossingen met zijn LCAP, met slechts een recente introductie van een Industry Solutions-team. De leverancier vertrouwt meestal op zijn moederbedrijf Siemens voor specifieke gerichte oplossingen en op partner-ISV's voor industriële toepassingen op zijn markt. - Marketinguitvoering: hoewel de overname door Siemens de markt meer bekendheid heeft gegeven met Mendix, heeft Gartner niet gezien dat Mendix een zo breed publiek bereikte als andere leiders in dit Magic Quadrant, met name buiten Europa en Noord-Amerika.
Internal	Strengths <u>Outsystems Gartner Report 2020</u> - Applicatie gebruikerservaring: OutSystems UI is een ingebouwd gebruikersinterfaceframework met meerdere sjablonen en UI-patronen. Heeft een uitgebreide set van aanpasbare componenten voor het bouwen van apps. Je kan ook zelf je UI-patronen bouwen - Ondersteuning voor ontwikkelaarspersona en software-ontwikkelingslevenscyclus (SDLC): de ervaring van Multipersona-ontwikkelaars biedt een op standaarden gebaseerde architectuur die veilig en schaalbaar is. - Architectuur: applicaties kunnen werken met verschillende services en gegevensbronnen, zowel van OutSystems als van een reeks on-premise of cloudgebaseerde services van derden.	Weaknesses <u>Outsystems Gartner Report 2020</u> - Applicatieontwikkeling: hoewel OutSystems Workflow Builder heeft ontwikkeld om het gebruik van burgerontwikkeling aan te pakken, is het nog steeds niet algemeen beschikbaar. - Proces- en bedrijfslogica: OutSystems 'LCAP wordt steeds vaker gebruikt voor automatisering van bedrijfsprocessen en ondersteunt accelerators voor data-ontwerp, inclusief Excel-import. Het is echter nog niet concurrerend in alle aspecten van procesgerichte applicatieontwikkeling, vooral met betrekking tot aspecten als complexe besluitvormingsmodellering en procesoptimalisatie. - Productstrategie: OutSystems heeft de neiging om de voorkeur te krijgen van professionele ontwikkelaars in plaats van burgerontwikkelaars, terwijl de meeste LCAP's zich richten op een meer gemengd ontwikkelaarspersonage.
Internal	Strengths <u>Microsoft Gartner Report 2020</u> - Productstrategie: Microsoft stelt fusieteams in staat zonder code en geavanceerde pro-ontwikkelaarsmogelijkheden. Microsoft biedt een van de eenvoudigere ontwerp tools in Power Apps, een die wordt gebruikt voor het ontwerpen van canvas-apps en die gebruikmaakt van slepen en neerzetten met een expressietaal die aanvoelt als Microsoft Excel. - API- en integratieservices: Power Apps en Common Data Service (CDS) hebben een uitgebreide set API's en OData-eindpunten beschikbaar om volledige CRUD-bewerkingen (maken, lezen, bijwerken en verwijderen) uit te voeren op gegevens ter ondersteuning van integratie met iPaaS van derden. - Innovatie: gebruiksklare AI-modellen stellen ontwikkelaars in staat om rijke AI-mogelijkheden in hun apps en stromen te gebruiken zonder een model te configureren en te trainen.	Weaknesses <u>Microsoft PowerApps Gartner Report 2020</u> - In 2019 werd PowerApps veel gebruikt. In oktober 2019 heeft Microsoft deze gebruiksvoorwaarden gewijzigd, een wijziging die veel klanten en zakenpartners in verwarring leek te brengen. Met licentie rechten for klanten heeft voor sommige situaties verholpen. - Verkoopuitvoering en prijsstelling: PowerApps heeft een complex prijsmodel en machtigingsregels. Er is kans op licentiewijzigingen in de toekomst. - Proces- en bedrijfslogica: Power Apps ondersteunt werkstromen, maar de interface voor bedrijfsprocesstromen is afzonderlijk en alleen beschikbaar in modelgestuurde apps. Soms zijn producten vereist, zoals Microsoft Azure Logic Apps. Bovendien wordt de BPMN-notatienorm in Power Automate alleen ondersteund door nauwe integratie met Microsoft Visio.

Internal	Strengths <u>SalesForce Gartner Report 2020</u> • Algemene levensvatbaarheid: De aanhoudende sterke groei van de algehele Salesforce-activiteiten en het toonaangevende ecosysteem van partners vormen een sterke drijfveer voor de acceptatie van het LCAP-aanbod van het bedrijf • Bedrijfsmodel: De aanhoudende populariteit van Salesforce SaaS-aanbiedingen bij zowel bestaande als nieuwe klanten heeft een sterk kanaal gevormd voor de platformdiensten van het bedrijf, waaronder het LCAP. • Innovatie: Salesforce bouwt aan een uniforme ontwikkel- en beheeromgeving die meerdere Customer 360 Platform-technologiestacks (Heroku en Apex) overbrugt.	Weaknesses <u>SalesForce Gartner Report 2020</u> • Architectuur: Het architectuur is in zijn nieuwe vorm anders. Dit zal veel klanten beïnvloeden. • Integratie: Het LCAP van Salesforce mist nog steeds low-code ondersteuning voor integratie en API-beheer, en het procesbeheer is relatief beperkt. • Markt strategie: Salesforce wordt alleen gebruikt voor CRM systemen. Het probleem is dat het platform als SaaS gebruikt wordt.

X. Adviesrapport: onderzoek low-codeplatform die het beste bij Netcompany past



Bijlage_10_Adviesra
pportage_KeuzeLow

XI. Requirements Proof of Concept

User Stories	MoSCoW
Inloggen via Mendix	Must Have
Het definiëren van gebruikersrollen	
Het gebruiken van demo users	Ik heb hiervoor gekozen omdat dit een Prototype/Demo is;
Uitloggen via Mendix	Must Have
Aanbestedingen dashboard pagina	Must Have
In het domein model van TenderNed een entiteit	
Aanbesteding aanmaken	
Generate overview pages	
Importeren van data TenderNed eerst als excel template;	Must Have
Aanbestedingen data tonen in het aanbestedingen dashboard pagina;	Must Have
Op het moment dat er op een aanbesteding wordt geklikt op het dashboard pagina moet ik een detail pagina aantonen met meer informatie;	Must Have
Het kunnen zoeken naar een bepaald Aanbesteding door een aanbestedingsnaam in te tikken;	Must Have
Het kunnen sorteren van Aanbestedingen	Must Have
Het klikken op een link dat mij naar de TenderNed website stuurt.	Must Have
Een gebruiker kan een aanbesteding kiezen als favoriet	Must Have
Het tonen van alleen de favorieten van een gebruiker in het Dashboard pagina.	Should Have
Het filteren van de aanbestedingen in het Dashboard.	Should Have
CSS stylesheet conform traditioneel development	Should Have
Het kunnen registreren van gebruikers via Mendix	Could Have
Het registreren van gebruikers via Active Directory	Could Have
Koppeling maken met Azure	Could Have
Koppeling maken met Netcompany database SQL Server	Could Have
Koppeling maken met TenderNed voor het ophalen van de gegevens	Could Have

XII. Uren van traditionele development PoC

Onderdeel	Omschrijving	Aantal uren
Het creëren van de pagina's	Inlog scherm; Dashboard pagina aanbestedingen; Detail pagina; User pagina;	40 uur
Simpele backend	Het ophalen van alle aanbestedingen; Het detail pagina weergeven van een aanbesteding; Authenticatie als gebruiker inlogt; Registreren van gebruiker; Opzetten database in .Net; Ophalen van data via URL TenderNed;	40 uur
Filter, sorteren en zoeken	In de frontend is het mogelijk om de aanbestedingen te filteren, sorteren en te zoeken via Aanbestedingsnaam	40 uur
Bugs	Mogelijke bugs verhelpen;	40 uur

XIII. Snapshots traditionele PoC



netcompany

Log in

Gebruikersnaam

Wachtwoord

Inloggen Registreren

TENDERBIC Aanbestedingen Offertes Log uit

Nieuwbouw Sporthal Middachtensingel; Aannemer bouwteam
De Connectie

[Link naar deze aanbesteding op Tenderbid](#)

Type publicatie: Aankondiging van een opdracht

Type opdracht: Werven

Omschrijving: De opdracht omvat de bouwkundige en civiele werken en de werkruimtebouwkundige en elektrotechnische installaties. Na gunning wordt in bouwtent-verband het door de gemeente Arnhem goedkeurende DO (gezamenlijk) uitgevoerd naar een TO en vervolgens tot een plan geres voor uitvoering (UO). De uitvoering van het werk moet binnen een vooraf gesteld plannibudget blijven. De gemeente is voortvarend vervolgende de uitvoering aan de aannemer op te dragen.

Procedure: Niet-openbaar

Publicatiedatum: 11-01-2021

Sluitingsdatum: 01-03-2021 10:00

Aanvangsdatum opdracht: 13-07-2021

Dagen tot sluitingsdatum: Sluitingsdatum is al gemeest

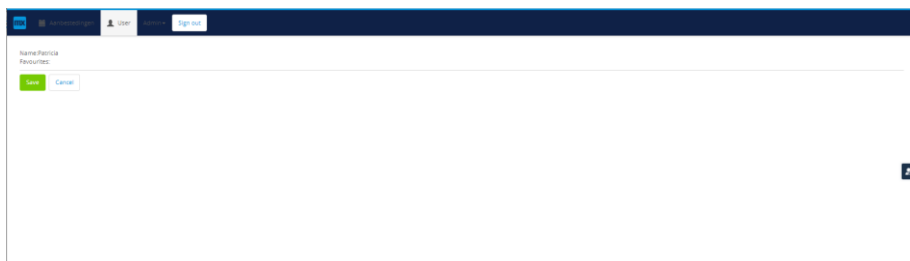
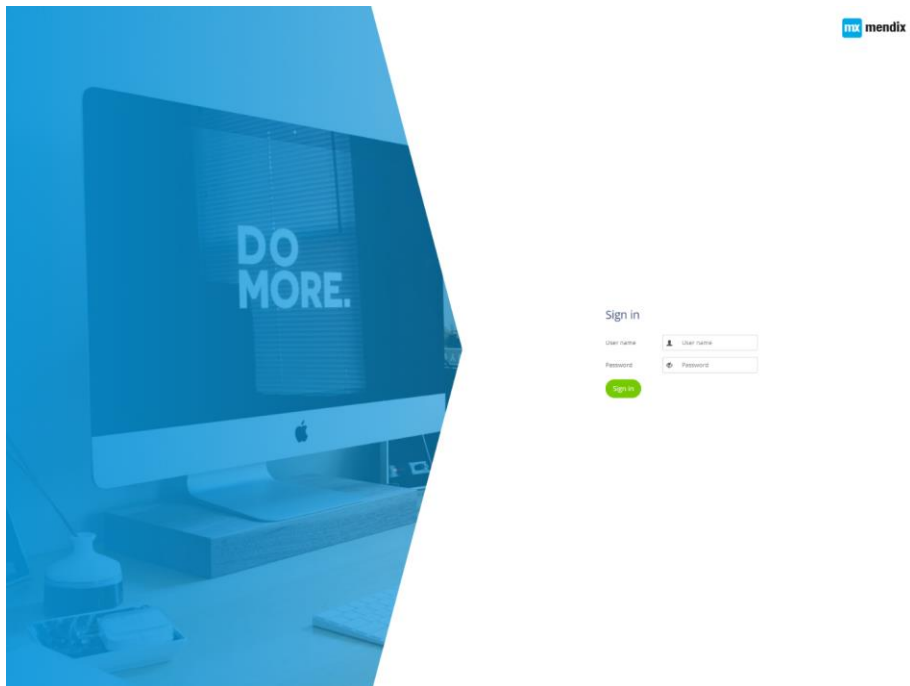
TENZING
Aanbevelingen
Gebruikers
Log uit

Gebruikers

Gebruikers toevoegen

Voornaam	Achternaam	Gebruikersnaam	
Azad	Imamoglu	aim	AANPASSEN
Chris	van den Broeck	cbr	AANPASSEN
L.	P	LP	AANPASSEN
Azad	Imamoglu	azad@hiqua.net	AANPASSEN
Patricia	Moscato	pma	AANPASSEN
Peggy	Kampman	pka	AANPASSEN
Ger	Senden	gsa	AANPASSEN
P	Mos	patriciamoscato	AANPASSEN

XIV. Snapshots Mendix PoC



XV. Tijdsbesteding Low-code en Traditioneel development

		Traditioneel	Low-code
Onderdeel	Omschrijving	Aantal uren	Onderdeel
Het creëren van de pagina's	Inlog scherm; Dashboard pagina aanbestedingen; Detail pagina; User pagina;	40 uur	8
Simpele backend	Het ophalen van alle aanbestedingen; Het detail pagina weergeven van een aanbesteding; Authenticatie als gebruiker inlogt; Registreren van gebruiker; Opzetten database in .Net; Ophalen van data via URL TenderNed;	40 uur	16
Filter, sorteren en zoeken	In de frontend is het mogelijk om de aanbestedingen te filteren, sorteren en te zoeken via Aanbestedingsnaam	40 uur	8
Bugs	Mogelijke bugs verhelpen;	40 uur	0

XVI. Expert interviews met ontwikkelaars



Bijlage_16_Questio
ns.docx

XVII. Gesprekken met leverancier Mendix



Bijlage_17_1Memo_
Mendix_Contact_me



Bijlage_17_2Memo_
Mendix_Contact_Tip

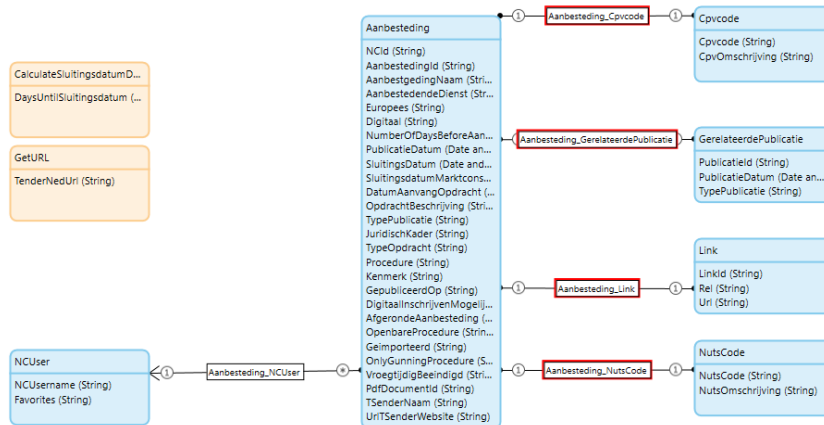


Bijlage_17_3Memo_
Mendix_Contact_Sch

XVIII. Certificaat Udemy



XIX. Domein model



XX. Voortgangsrapportage maand februari



Bijlage_20_Logboek
.docx