

Macaw Assistant Bot

AFSTUDEERVERSLAG SOFTWARE ENGINEERING

MARTIJN VAN OUDENHOVE

15099997

Voorwoord

Mijn afstudeerstage heeft plaatsgevonden in de periode van 1 oktober tot en met 1 maart. In die periode heb ik een chatbot gemaakt voor intern gebruik bij het bedrijf Macaw. De totale doorlooptijd van de stage bij Macaw was achttien weken, waarvan er vier weken niet gewerkt zijn doordat ik ziek geweest ben en door de onbeschikbaarheid van de stagebegeleider/product owner. Hierna heb ik nog ruim drie weken gewerkt aan dit verslag. Ik wil graag alle mensen die ik heb gesproken bij Macaw bedanken en in het bijzonder mijn begeleider Peter Heibrink.

Inhoudsopgave

Voorwoord	1
Inleiding.....	4
Achtergrondinformatie Macaw	5
De opdracht	6
Aanleiding en context	6
Probleemanalyse en probleemstelling	6
Doelstelling en eindresultaat	7
Aanpak	8
Gekozen oplossingsrichting	8
Keuze van de methode	9
Afspraken SCRUM-methode	10
Requirements analyse.....	10
Functional Requirements.....	11
Non-Functional Requirements.....	12
Risicoanalyse.....	12
Planning	13
Tools.....	14
Projectfase	16
Sprint 0: Planning en randvoorwaarden	17
Planning.....	17
Realisatie.....	17
Review	18
Sprint 1: De Echo Bot	19
Planning.....	19
Ontwerp	19
Realisatie.....	23
Test.....	24
Review	24
Sprint 2: De QnA Bot	26
Planning.....	26
Ontwerp	26
Realisatie.....	26

Test.....	27
Review.....	28
Sprint 3: De Service Bot	29
Planning.....	29
Ontwerp	29
Realisatie.....	29
Test.....	30
Review.....	30
Sprint 4: Productie	31
Planning.....	31
Ontwerp	31
Realisatie.....	31
Test.....	31
Review.....	32
Evaluatie.....	33
Afwijking afstudeerplan	33
Proces.....	33
Product.....	33
Beroepstaken	34
A1: Analyseren probleemdomein & opstellen probleemstelling	34
Gc: Kritisch en methodisch werken	34
Gf: Leren, leren	34
A3: Vergaren en analyseren van requirements	34
D14: Realiseren van software	34
D15: Testen	36
Appendix	37
Bijlage 1: Afstudeerplan	37
Bijlage 2: Plan van Aanpak	37
Bijlage 3: Manual Knowledge Base Management	37
Bijlage 4: Deployment Manual.....	37
Bijlage 5: Technical Design.....	37

Inleiding

Dit verslag betreft mijn afstudeerstage bij het IT bedrijf Macaw in Hoofddorp. Als ontwikkelaar heb ik gewerkt aan een chatbot applicatie binnen Microsoft Teams, een belangrijk samenwerkingsplatform binnen Macaw. Ik heb gewerkt aan het analyseren van het probleem, het inventariseren van de requirements en het ontwerpen, bouwen, testen, packagen en documenteren van deze applicatie.

Het doel van dit verslag is aantonen dat ik mijn afstudeerstage heb afgerond op het niveau zoals mij dat geleerd is op de Haagse Hogeschool. In dit verslag reflecteer ik op mijn stageperiode door het proces en het product te beschrijven aan de hand van een aantal vooraf opgestelde beroepstaken. Dit zijn de beroepstaken die ik heb opgenomen in mijn afstudeerplan. De beroepstaken die zijn opgenomen in dit verslag zijn:

- **A1: Analyseren probleemdomein & opstellen probleemstelling**
- **A3: Vergaren en analyseren van requirements**
- **D14: Realiseren van software**
- **D15: Testen**
- **Gc: Kritisch en methodisch werken**
- **Gf: Leren leren**

De volgende twee beroepstaken heb ik niet vooraf in mijn afstudeerplan genoemd:

- **C6: Ontwerpen software**
- **D17: Configureren**

Om een compleet beeld te geven van alle beroepstaken die ik tijdens het project heb uitgevoerd, heb ik ook deze opgenomen in dit verslag. Bij elk van de sprints heb ik onder de paragraaf review, naast de review bij Macaw, ook mijn reflectie op deze taken opgenomen.

Dit verslag begint na deze inleiding met een stuk achtergrondinformatie over Macaw als bedrijf. Hierop volgend beschrijf ik de opdracht die ik uit heb gevoerd bij Macaw. Daarna beschrijf ik de aanpak van het project en onderbouw ik de keuzes die zijn gemaakt voor de aanpak. Dan wordt de projectfase beschreven, de uitvoering van alle werkzaamheden. Het hoofdstuk daarna is een evaluatie van het afstudeerproject. Tenslotte een appendix met alle geproduceerde documenten. De gemaakte code lever ik in een aparte zip-file aan.

Achtergrondinformatie Macaw

Macaw is een profit bedrijf binnen de IT-branche. De doelstellingen van het bedrijf zijn het creëren van positieve klantervaringen en het versterken van medewerkersproductiviteit en samenwerking. Macaw biedt een viertal diensten aan: **Digital Marketing & Commerce, Data, Analytics & AI, Business Applications** en **Collaboration & Modern Workplace**. De organisatie van Macaw is gebaseerd op deze diensten. Op dit moment heeft Macaw ongeveer 350 medewerkers in dienst en de omzet van 2018 bedroeg 30 miljoen euro. Macaw bestaat al 25 jaar en zet op het moment in op internationale groei.

De dienst waar deze afstudeeropdracht betrekking op heeft is de dienst **Collaboration & Modern Workplace**. Onder deze dienst vallen drie afdelingen, die zich elk richten op een specifiek aspect: hoe maakt de gebruiker gebruik van de nieuwe oplossing; **User Adoption**, op welke manier wordt er samengewerkt: **Smart Collaboration** en welke middelen maken dit mogelijk: **Modern Workplace**.

De opdracht zal worden uitgevoerd op de afdeling **Workplace Solutions**. Deze afdeling heeft ongeveer 25 medewerkers. Iedereen op deze afdeling is deskundig op het gebied van Office365 en heeft daarnaast nog hun eigen specialisatie binnen het werkgebied. De afdeling fungeert als ondersteuning voor bedrijven op het gebied van interne collaboratie- en communicatieplatformen. Deze platformen maken altijd gebruik van Office365. De afdeling ondersteunt bedrijven enerzijds door uit te leggen hoe zij gebruik moeten maken van Office365 en anderzijds ontwikkelt deze afdeling ook zelf deze samenwerkingsplatformen naar de wens van de klant.

Mijn rol als afstudeerder binnen deze afdeling was de rol van ontwikkelaar. Ik heb gewerkt aan een applicatie binnen een van deze samenwerkingsplatformen, namelijk Microsoft Teams. Ik heb gewerkt aan het analyseren van het probleem, het inventariseren van de requirements en het ontwerpen, bouwen en testen van deze applicatie. Ook heb ik gezorgd voor het opleveren van het project met bijbehorende documentatie.

De opdracht

De opdracht heeft geleid tot het ontwikkelen van een chatbot, de Macaw Assistant, die intern gebruikt wordt door Macaw. Deze chatbot moet dienen als een ondersteuning voor het personeel van Macaw door het beantwoorden van vragen en het aanbieden van bepaalde services (bijvoorbeeld het aanmelden van een bezoeker). Hieronder meer over hoe de opdracht is ontstaan.

Aanleiding en context

Macaw is een zakelijk dienstverlener met een informele sfeer. Door de informele sfeer worden veel vragen beantwoord bij de koffieautomaat. Deze informatie is niet altijd vast gelegd, maar leeft in de hoofden van medewerkers die er al een tijd werken.

Doordat Macaw de laatste jaren enorm gegroeid is en aan het uitbreiden is naar het buitenland (momenteel Litouwen en Duitsland) wordt het steeds belangrijker om op een uniforme manier vast te leggen en te communiceren hoe Macaw eigenlijk werkt. Binnen Nederland werkt een groot gedeelte van de medewerkers een aantal dagen per week op locatie bij de klanten, waardoor veel nieuwe medewerkers met vragen blijven zitten, die niet meer bij de koffiemachine beantwoord kunnen worden. Hier is een oplossing voor nodig.

Probleemanalyse en probleemstelling

De informatie voor medewerkers is momenteel heel erg verspreid. Nieuwe medewerkers kost het veel tijd om erachter te komen bij wie ze voor bepaalde zaken moeten zijn. Intern kan dat bij een groot aantal verschillende afdelingen zijn, zoals HR, Recruitment, Finance, Operation Office, Service Desk, Facilities, Receptie.

Daarbij komen de vragen niet altijd direct bij de juiste afdeling of persoon. Op dit moment is daar nog redelijk mee om te gaan, omdat medewerkers direct doorverwezen worden naar de juiste afdeling en dan antwoord kunnen krijgen, maar dit kost de relatief kleine stafdelingen veel tijd. Met de huidige groei en de groeiplannen gaat dit een probleem worden.

Er is een oplossing nodig welke op een eenvoudige manier medewerkers helpt met vragen over hoe binnen Macaw zaken geregeld zijn. Daarbij moet het voor de medewerker één loket zijn waar hij of zij haar vraag kan stellen en geholpen wordt. Dit gebeurt idealiter zonder menselijke tussenkomst, zodat het enerzijds geen tijd kost, maar anderzijds ook buiten werktijden gebruikt kan worden.

Aangezien de medewerkers van Macaw veel werken met Microsoft Teams zou dit een goed platform kunnen zijn voor de oplossing. Deze applicatie is zowel op de laptops als de mobiele apparaten van medewerkers te gebruiken, wat voor de beschikbaarheid ideaal is.

Doelstelling en eindresultaat

Zoals hiervoor beschreven, is de uitdaging bij Macaw dat er geen eenduidige manier is om op te zoeken hoe zaken binnen Macaw geregeld zijn, anders dan medewerkers welke langer in dienst zijn te vragen. Met de huidige groei en toekomstige groeiplannen is dit geen houdbare optie. De doelstelling van dit project luidt dan ook:

Lever een oplossing om medewerkers op een laagdrempelige manier informatie te verstrekken over hoe zaken binnen Macaw geregeld zijn.

Hierbij zijn nog een aantal secundaire doelstellingen benoemd:

1. Ontlasten van de interne stafafdelingen.
2. Een oplossing welke gedemonstreerd kan worden bij klanten van Macaw, aangezien deze problematiek daar ook leeft.

Aanpak

In dit hoofdstuk beschrijf ik de aanpak van het project. Dit zijn alle zaken die aan bod zijn gekomen voordat de eerste sprint (sprint 0) van start ging.

Gekozen oplossingsrichting

Binnen Macaw zijn er een aantal mogelijke oplossingen voor dit probleem bedacht:

- **Uitbreiding introductiedagen:**
Macaw heeft een aantal keren per jaar introductiedagen voor nieuwe medewerkers. Er kan langer tijd besteed worden aan het in meer detail uitleggen hoe Macaw werkt. Daarvoor zijn medewerkers nodig die deze presentaties/workshops geven.
- **Opzetten internationale HR-helpdesk**
Ter ontlasting van de interne stafafdelingen. Deze helpdesk dient gedurende de service uren bemenst te zijn.
- **Verbetering informatievoorziening intranet:**
Een eventuele verbetering van het intranet zou ook een alternatieve oplossing kunnen zijn. Het gaat hier om het vastleggen, onderhouden en doorzoekbaar maken van informatie
- **Maken van een chatbot:**
Een chatbot in de standaard werkomgeving van de medewerkers die vragen kan beantwoorden. Het vastleggen, onderhouden en doorzoekbaar maken van informatie speelt ook hier een rol.

We hebben de volgende selectiecriteria gebruikt:

- **Eenmalige investering**
Het maken van een oplossing is een investering die zou moeten leiden tot beter geïnformeerde medewerkers zonder verhoging van de reguliere operationele kosten.
- **Laagdrempelig, makkelijk toegankelijk**
De oplossing is makkelijk toegankelijk en vindbaar.
- **Moderne flexibele technologie:**
Het is voor een IT-bedrijf altijd interessant om een nieuwe/opkomende technologie onder de loep te nemen.
- **Microsoft gebaseerd**
Macaw is een gecertificeerd partner van Microsoft en ontvangt allerlei benefits. Hier gaat het om kortingen en bijvoorbeeld voorrang bij het opvragen van support bij de helpdesk van Microsoft. Tevens omdat Macaw een bedrijf is dat zich richt op Microsoft technologieën is er veel kennis in huis over het gebruik van Microsoft producten.
- **Commercieel (her)bruikbaar:**
Specifiek de technologie van het Bot Framework is commercieel interessant voor Macaw, omdat de chatbots ook ingezet kunnen worden bij de klanten van Macaw.

Het resultaat van het toepassen van de criteria op de oplossing is hieronder aangeven:

Criterium	Introductiedagen	HR helpdesk	Intranet	Chatbot
Eenmalige investering	Nee	Nee	Ja	Ja
Laagdrempelig, makkelijk toegankelijk	Nee, alleen bij start	Neutraal	Ja	Ja
Moderne flexibele technologie	N.v.t.	N.v.t	Redelijk flexibel, niet heel modern	Ja
Microsoft gebaseerd	Nee.	Nee	Ja	Ja
Commercieel bruikbaar	Nee	Nee	Nee	Ja

Zowel bij het betere intranet als bij de chatbot blijft natuurlijk dat de basis juiste en volledige informatie is. Deze informatie dient ook bijgehouden te worden. Deze kosten zijn buiten de afweging gehouden. De nieuwe technologie heeft duidelijk de doorslag gegeven voor de chatbot.

Keuze van de methode

Er zijn drie methoden overwogen aan het begin van het project. Deze methoden waren: **waterval**, **SCRUM** en **KANBAN**.

De waterval-methode viel al snel af, omdat er verwacht werd dat er veel wijzigingen in de requirements tijdens het project plaats zouden vinden. Een meer agile approach zou dus beter passen in het project.

KANBAN is een agile methode die gebruik maakt van het KANBAN-board. Dit board heeft een aantal swimlanes voor de verschillende fases (develop, test, etc.) waarin een work item zich bevindt. Het grootste voordeel van de KANBAN methode is dat er een limiet wordt opgelegd aan het aantal work items dat zich tegelijkertijd mag bevinden in de verschillende swimlanes. Dit zorgt ervoor dat het werk dat wordt uitgevoerd voor iedereen overzichtelijk blijft. Een nadeel van KANBAN is dat er geen vaste tijdstippen zijn waarop een review wordt uitgevoerd. Ook ligt er geen nadruk op het maken van tussenproducten. Uiteraard zou kunnen worden gekozen om deze tijdstippen van review en de nadruk op tussenproducten op te nemen in de manier van aanpak, maar dan begint het al erg veel op SCRUM te lijken.

Er is dus uiteindelijk gekozen voor de SCRUM-methode, omdat de product owner beperkt beschikbaar was. Door gebruik te maken van SCRUM konden alle vaste sessies (daily stand-up, planning en review) voorhand vastgelegd worden, zodat de product owner deze tijdstippen in zijn agenda kon blokkeren. Een andere reden waarom er is gekozen voor SCRUM is dat SCRUM de meest voorkomende manier van

projectmethodiek is binnen Macaw. De laatste reden is dat ik veel ervaring heb vanuit de Haagse Hogeschool met het werken met SCRUM. Een nadeel van SCRUM is dat, omdat ik de enige ontwikkelaar zou zijn, er geen sprake is van een team. SCRUM kijkt ook naar het verbeteren van de werkwijze van het team wat in dit project minder relevant is door het kleine team.

Afspraken SCRUM-methode

Scrum Agile zal worden gebruikt om de Macaw Assistant te realiseren. Het project zal worden opgestart met een sprint 0 met een duur van twee weken. Deze sprint 0 zal worden gebruikt om te zorgen dat de randvoorwaarden voor het project vastgelegd worden. Hierna zal de Macaw Assistant worden gerealiseerd in 4 sprints van elk 3 weken. Na deze 14 weken zou er 3 weken worden opgenomen in de planning om mij de ruimte te geven om dit stageverslag te schrijven. Er is gekozen voor een sprintduur van 3 weken, omdat er elke sprint genoeg tijd moet worden genomen om iets te onderzoeken, iets op te leveren dat werkt en iets dat in gebruik kan worden genomen. Het team heeft geen ervaring met de nieuwe technologie van het Microsoft Bot Framework, dus er moet genoeg tijd worden genomen voor onderzoekswerk en het oplossen van eventuele onvoorziene problemen. Mocht de sprintduur te lang zijn, dan kan deze altijd, in overleg, bijgesteld worden. Azure DevOps zal als tool ingezet worden om de processen van SCRUM inzichtelijk te maken.

Elke werkdag wordt er begonnen met een daily stand-up waarin drie vragen worden beantwoord: "Welk werk is er gisteren verricht?", "Ben je tijdens het verrichten van dit werk tegen problemen aangelopen?" en "Welk werk wordt er vandaag opgepakt?". De aanwezigen bij deze stand-up zijn, in ieder geval, de ontwikkelaar en de product owner.

Een sprint zal altijd beginnen met een sprintplanning waarin duidelijk wordt gemaakt wat het doel is van de huidige sprint. Om dit doel te bereiken worden de relevante user stories uit de MoSCoW geprioriteerde productbacklog verplaatst naar de sprintbacklog. Tevens wordt er, waar nodig, meer detail toegevoegd aan de user stories zodat de user stories helder zijn voor het team. Er worden acceptatiecriteria opgesteld voor elke user story en elke user story krijgt een set van taken. De taken krijgen een schatting van het aantal uren nodig om uit te voeren. Dit wordt allemaal vastgelegd in de **Boards** omgeving van Azure DevOps.

Het uitvoeren van de work items binnen een sprint gebeurt volgens een kleine waterval. Het work item begint met het analyseren. Hierbij worden diverse bronnen geraadpleegd. Hierna wordt er waar nodig het ontwerp gemaakt of aangepast. Na het schrijven van de code worden de functionaliteiten getest. Aan het einde van de sprint vindt er een demonstratie van de gerealiseerde functionaliteit plaats. Bij deze demonstratie zijn de stakeholders (in dit geval het complete SCRUM-team) aanwezig. Na feedback te hebben ontvangen over het gemaakte product wordt er de tijd genomen om, waar nodig, het project bij te sturen.

Requirements analyse

De requirements heb ik opgesplitst in twee tabellen, namelijk de **Functional Requirements** en de **Non-Functional Requirements**. Tevens heb ik ook aangegeven waar de requirement op gebaseerd is, ofwel met wie ik heb gesproken om de requirement vast te stellen. Ook verwijs ik in de tabellen waar in het verslag te lezen is over de uitwerking van de requirement. Elk van deze requirements is omgezet naar work items voor de backlog van het project. Door gebruik te maken van acceptatiecriteria voor de work items zijn de requirements SMART gemaakt. Voor de duidelijkheid heb ik de laatste versie van de lijst

met requirements genoteerd. Gedurende het project zijn uiteraard hier en daar wat requirements veranderd, maar dit geeft een goed overzicht van de high level eisen van het project.

Functional Requirements

ID	Omschrijving	Prioriteit	Bron	Aan voldaan?
F1	De settings van de chatbot moeten configureerbaar zijn	Must Have	Product Owner	Ja, zie Sprint 1: De Echo Bot
F2	De chatbot moet de vragen van de medewerkers kunnen beantwoorden	Must Have	Product Owner	Ja, zie Sprint 2: De QnA Bot
F3	De chatbot moet gebruiksvriendelijk zijn	Should Have	Product Owner	Geen directe focus op gehad, zie Evaluatie
F4	De chatbot moet services kunnen aanbieden	Must Have	Product Owner	Ja, zie Sprint 3: De Service Bot
F5	De chatbot moet te gebruiken zijn in andere landen waar Macaw actief is	Should Have	HR-Manager	Ja, zie Sprint 2: De QnA Bot
F6	De database van vraag en antwoorden moet te onderhouden zijn	Must Have	HR-Manager	Ja, zie Sprint 2: De QnA Bot
F7	De database moet te updaten zijn via SharePoint	Would Have	HR-Manager	Nee, zie Sprint 2: De QnA Bot
F8	De chatbot moet logging bevatten	Must Have	Product Owner	Ja, zie Sprint 1: De Echo Bot

Non-Functional Requirements

<u>ID</u>	<u>Omschrijving</u>	<u>Prioriteit</u>	<u>Bron</u>	<u>Aan voldaan?</u>
NF1	De chatbot moet draaien in Microsoft Teams	Must Have	Product Owner	Ja, zie Projectfase
NF2	Er moeten drie omgevingen worden gebruikt tijdens het project, namelijk development, test en production	Must Have	Product Owner	Ja, zie Projectfase
NF3	De chatbot moet gehost worden op Azure	Must Have	Product Owner	Ja, zie Sprint 1: De Echo Bot
NF4	De code van de chatbot wordt geschreven in C#	Must Have	Product Owner	Ja, zie Projectfase
NF5	Het project moet overdraagbaar zijn	Must Have	Product Owner	Ja, zie Sprint 4: Productie
NF6	Het project moet gedocumenteerd zijn	Must Have	Product Owner	Ja, zie Sprint 4: Productie
NF7	De chatbot moet te allen tijde beschikbaar zijn	Should Have	Product Owner	Ja, door gebruik te maken van de clouddienst van Microsoft Azure is de uptime 99,9%

Risicoanalyse

Binnen het project is er sprake van een aantal technische randvoorwaarden. De voorwaarden zijn:

- Er moet een laptop geregeld worden met Visual Studio en een licentie.
- Er moet een account beschikbaar worden gesteld die toegang levert tot het Macaw netwerk en de services, onder andere Microsoft Teams.
- Er is een geldige Azure subscription nodig om mee te werken.

De mogelijke risico's voor het project zijn:

- De product owner heeft naast dit project nog veel andere projecten die tijd vergen, waardoor hij mogelijk onvoldoende beschikbaar is voor dit project.
- De content voor de invulling van de chatbot om te kunnen testen moet tijdig worden aangeleverd.
- Voor de productieve versie is er veel potentieel op te nemen informatie beschikbaar bij de afdelingen en er zijn veel services die ontsloten kunnen worden. Keuzes dienen nog gemaakt te worden tijdens het project. Dit is een beperkt risico, omdat dit buiten de scope van het project ligt.
- Er is nog geen vergelijkbare oplossing in gebruik binnen Macaw, waardoor er beperkt kennis is van deze technologie.

De maatregelen die worden genomen om deze risico's te minimaliseren zijn:

- Alle SCRUM-meetings zijn voor aanvang van het project ingepland en doorgesproken. Ook is er een back-up voorzien, mocht de product owner toch niet beschikbaar zijn.
- Tijdig overleg hebben met HR-afdeling voor aanleveren initiële testset van vragen en antwoorden.
- De randvoorwaarden voor de content van de chatbot worden ruim van tevoren aangevraagd. Hierbij worden alternatieve afdelingen/services bepaald om ervoor te zorgen dat er geen stilstand komt.
- De backlog van het project is ingericht volgens het principe van **fail early**. Het principe van fail early is gebaseerd op het idee dat de functionaliteiten waarvan verwacht wordt dat deze het moeilijkst zijn om te realiseren, de hoogste prioriteit krijgen. Dit voorkomt dat er, wanneer er nog maar relatief weinig tijd is om het project af te ronden, er niet (tot geen) vertraging wordt opgelopen door complexe stukken functionaliteit. Het hele proces van de chatbot beschikbaar stellen wordt al in sprint 1 opgepakt en waar nodig kan er overlegd worden met specialisten binnen Macaw.

Planning

Hieronder nog een beknopt overzicht van de sprints met datum en sprintdoel:

Sprint	Geplande Periode	Daadwerkelijke Periode	Doel
0	02-10 t/m 11-10	02-10 t/m 11-10	Planning en randvoorwaarden
1	14-10 t/m 01-11	14-10 t/m 01-11	Echo Bot
2	04-11 t/m 22-11	04-11 t/m 22-11	QnA Bot

3	25-11 t/m 13-12	25-11 t/m 13-12	Service Bot
4	16-12 t/m 03-01	16-01 t/m 03-02	Production

Zoals al eerder aangegeven in het voorwoord van dit document is door griep en onbeschikbaarheid van de stagebegeleider/product owner de laatste sprint een maand vertraagd.

Het maken van dit verslag heeft daarna plaatsgevonden.

Tools

Hieronder volgt een beschrijving van de gebruikte tools binnen het project en ook een korte beschrijving van deze tools. Doordat Macaw een gecertificeerd Microsoft partner is, was de voorkeur om zoveel mogelijk tools van Microsoft te gebruiken. Dit omdat er kennis in het bedrijf is over de tools, maar ook omdat er licenties vrij zijn voor het gebruik van deze tools.

De gebruikte tools zijn:

- **TFS/Azure DevOps**

Team Foundation Server (TFS) of Azure DevOps is een tool die gebruikt werd voor meerdere zaken. Allereerst werden hier de SCRUM-artifacts vastgelegd. Denk hier bijvoorbeeld aan de productbacklog, de sprintbacklog en het verloop van de sprint. Ook werd deze tool gebruikt voor het testen van de chatbot. De testscenario's om uit te voeren na elke push werden hier opgenomen. Als laatste punt werd deze tool gebruikt voor het regelen van de continuous-development en deployment. Door gebruik te maken van meerdere pipelines werden de nieuw ontwikkelde functionaliteiten uitgerold naar de verschillende omgevingen.

Er is gekozen om te werken met TFS/Azure DevOps om een aantal redenen:

1. TFS/Azure DevOps is de standaard tool die gebruikt wordt door Macaw tijdens het uitvoeren van de projecten. Het gebruik van deze tool zal dus de overdraagbaarheid bevorderen, omdat de medewerkers van Macaw weten waar ze bepaalde informatie over het project kunnen vinden binnen deze tool.
2. TFS/Azure DevOps geeft goede mogelijkheden om hand in hand te werken met een SCRUM-methodiek. Een voorbeeld is het ingebouwde SCRUM-board binnen de tool.
3. Ik heb al eerder gewerkt met deze tool bij verschillende projecten die ik heb doorlopen tijdens mijn opleiding aan de Haagse Hogeschool.

- **SonarQube**

SonarQube is gebruikt om automatisch een code analysis uit te voeren op het moment dat er source code werd gepushed. Deze tool was ingebouwd in de build pipeline in TFS.

Er is gekozen om te werken met SonarQube om twee redenen:

1. Idem aan de laatste reden van het gebruik van TFS/Azure DevOps is dat ik deze tool eerder heb gebruikt tijdens de projecten aan de Haagse Hogeschool. Deze tool was fijn om mee te werken en mede om deze reden heb ik gekozen om de tool opnieuw in te zetten voor dit project.
2. SonarQube is een tool die de kwaliteit van de code analyseert en suggesties doet waar de code kan worden verbeterd. Een voorbeeld hierbij is dat de tool aangeeft

waar **bad smells**, zoals bijvoorbeeld een **large class** of **duplicate code**, plaatsvinden en doet suggesties om dit op te lossen.

- **Bot Framework Emulator**

Bot Framework Emulator is een tool die gebruikt is om de chatbot te testen. Deze tool zorgde er dus voor dat de code gedebugd kon worden zonder dat deze gedeployed hoefde te worden naar Microsoft Teams. Er is gekozen om deze emulator te gebruiken, omdat deze voorgeschreven wordt in alle Microsoft documentatie over het werken met hun Bot Framework.

- **Microsoft Teams**

Microsoft Teams is gebruikt als tool voor het SCRUM-team om te communiceren. Binnen Microsoft Teams is een channel (team) aangemaakt specifiek voor dit project waar kon worden overlegd over de chatbot. Ook hebben veel daily stand-ups plaatsgevonden via de meeting functie van Microsoft Teams, omdat lang niet altijd iedereen op dezelfde locatie zat.

Ook is dit de tool die Macaw gebruikt voor communicatie. Tevens moest de chatbot gekoppeld worden met deze interface, dus voor het product was dit goed om hier al kennis mee te maken.

- **Visual Studio 2019**

Visual Studio 2019 is gebruikt als IDE voor dit project. De keuze viel voor het gebruik van deze IDE, omdat Visual Studio de IDE is van Microsoft, dus de integratie met de andere tools en programmatuur van Microsoft werkt goed. Tevens heb ik ook positieve ervaring met deze IDE, want tijdens de colleges van de Haagse Hogeschool was dit vaak de voorgeschreven IDE.

- **Microsoft Visio**

Microsoft Visio is gebruikt om de diagrammen in de ontwerpfases vast te leggen. Er is gekozen om dit programma te gebruiken, omdat het wederom een Microsoft programma is, dus Macaw had hier licenties voor. Ook heb ik eerder met Microsoft Visio ontworpen tijdens mijn projecten aan de Haagse Hogeschool en het programma is mij altijd positief bevallen.

Projectfase

In dit hoofdstuk heb ik voor elke sprint, m.u.v. sprint 0, de volgende indeling gehanteerd:

- **Planning:**
Voor de overzichtelijkheid heb ik de planningsfase als eerste fase van elke sprint beschreven, ondanks dat de sprintplanning gelijktijdig met de review van de vorige sprint werd uitgevoerd.
- **Ontwerp:**
Het in iedere sprint selecteren van de te gebruiken componenten met onderbouwing.
- **Realisatie:**
Hieronder vallen het doen van onderzoek, het schrijven van code, het configureren van componenten en het schrijven van documentatie.
- **Test:**
De geschreven tests zijn vastgelegd op de DevOps omgeving. Elke sprint zijn er tests toegevoegd voor de nieuwe functionaliteiten en zijn de tests die al opgenomen waren opnieuw uitgevoerd om er zeker van te zijn dat de oude functionaliteit ook nog werkt. In overleg met de product owner is ervoor gekozen om deze tests vooralsnog nog niet te automatiseren. De tijd die hiervoor nodig zou zijn kan niet besteed worden om nieuwe functionaliteit te realiseren. De mogelijkheid bestaat wel om de tests te kunnen automatiseren binnen DevOps, dus in een vervolgproject kan dit worden toegevoegd.
- **Review:**
Het demonstreren van de gerealiseerde functionaliteit en het doorspreken van het proces. Daarnaast voeg ik in deze paragraaf de gebruikte beroepstaken toe.

Er zijn totaal 5 sprints uitgevoerd die elk een specifiek thema/naam gekregen hebben:

Sprint	Titel/focus gebied
0	Planning en randvoorwaarden
1	Echo bot
2	QnA Bot
3	Service Bot
4	Productie

Sprint 0: Planning en randvoorwaarden

Sprint 0 had als thema **planning en randvoorwaarden**. Deze sprint was gericht op het opstellen van een Plan van Aanpak, het inrichten van de ontwikkelomgeving, het aanvragen van licenties voor benodigde software en het opvullen van de initiële productbacklog. Tevens werd deze sprint ook gebruikt om een eerste opzet te doen over de mogelijkheden van chatbots in het algemeen.

Planning

De sprintbacklog voor sprint 0 zag er als volgt uit:

- Set up laptop environment to start development
- Create **Definition of Ready**
- Create **Definition of Done**
- Write **Plan van Aanpak** document
- Determine approach for bots

Realisatie

Het eerste work item dat is opgepakt in deze sprint is het opzetten van de laptop environment. **Visual Studio** 2019 is geïnstalleerd en bij de service desk van Macaw is een licentie hiervoor aangevraagd. Andere software die is geïnstalleerd waren bijvoorbeeld **Bot Framework Emulator** en **Microsoft Teams** voor de communicatie.

Nadat alle software draaiende was op de laptop is de **Definition of Ready** opgesteld en opgenomen in de DevOps omgeving. De **Definition of Ready** luidt als volgt:

A User Story is ready for planning when:

- The User Story's dependencies are identified
- The User Story is sized by Delivery Team
- The SCRUM-team accepts the User Experience artifacts
- The Performance criteria are identified, where appropriate
- The Acceptance criteria are identified
- The Team has a good idea what it will mean to Demo the User Story

Ook de **Definition of Done** is opgesteld en opgenomen in de DevOps omgeving. De **Definition of Done** luidt als volgt:

A User Story is ready for review when:

- All tasks are completed
- All code is completed
- Code is tested, where needed
- All code is refactored
- All code is documented
- All functionality tested
- Code is peer reviewed
- Documentation is written, when needed

De Definition of Ready en de Definition of Done heb ik opgesteld in overleg met mijn begeleider. Bij de Definition of Ready was vooral het eerste criterium belangrijk tijdens de sprintplanning. Het was belangrijk om goed duidelijk te krijgen dat er mogelijk eerst specifieke user stories moesten worden afgerond voordat er aan andere stories kon worden gewerkt. Als we als voorbeeld bijvoorbeeld de stories van sprint 1 nemen, dan kan de logging pas geïmplementeerd worden wanneer er een initiële versie van de bot draaiende is, anders valt er niets te loggen. Omdat het SCRUM team klein was is er niet erg intensief gewerkt met de Definition of Ready en de Definition of Done. Deze Definitions zijn opgesteld als bewustwording om te zorgen dat de work items van een project overzichtelijk blijven, maar voor een klein team is het overzicht van het project goed te behouden.

Het **Plan van Aanpak** document is geschreven tijdens deze sprint en is te vinden in de bijlagen onder Bijlage 2: Plan van Aanpak.

Tenslotte is er onderzoek gedaan naar de initiële aanpak voor het ontwikkelen van een chatbot. In eerste instantie is er gekeken welke versie van het Bot Framework SDK zou worden gebruikt.

SDK v3	SDK v4
Geen active development, alleen nog bugfixes	Active development
Nog onduidelijk of support doorgezet wordt na 2019	Support aanwezig
Meer kans op ervaring binnen Macaw	Nieuwe technologie moet geleerd worden
Verouderde documentatie	(Nieuwste) Quickstarts gebruiken deze SDK

Op grond van bovenstaand overzicht is voor versie 4 gekozen.

Ook is er tijdens het onderzoek gekeken naar eventuele relevante Quickstart Tutorials van Microsoft om niet vanaf scratch te hoeven beginnen met programmeren. Dit had als voordeel dat ik niet het wiel opnieuw hoefde uit te vinden met betrekking tot standaardvraag-en-antwoord functionaliteit van de chatbot.

Review

Na afloop van sprint 0 zijn de uitkomsten van het onderzoek doorgenomen met de product owner en is de initiële aanpak goedgekeurd door de product owner. De review was belangrijk om vast te stellen hoe het team zou beginnen met het ontwikkelen van de chatbot.

Tijdens deze sprint heb ik de volgende beroepstaken uitgevoerd:

- **A1: Analyseren probleemdomein & opstellen probleemstelling**
Deze beroepstaak komt terug in het schrijven van het Plan van Aanpak document. In dit document heb ik de huidige situatie geanalyseerd en het probleem zo concreet mogelijk gemaakt.

- **A3: Vergaren en analyseren van requirements**

Tevens ben ik ook bezig geweest met het vergaren en het analyseren van requirements. De initiële aanpak voor het ontwikkelen van de eerste versie van de chatbot vergde onderzoek naar de eisen van het systeem.

- **Gf: Leren leren**

Ik heb deze sprint onderzoek gedaan om advies te geven over de aanpak voor het opzetten van de initiële chatbot. Ik heb dit onderzoek gedaan door zoveel mogelijk relevant studiemateriaal te vinden vanuit Microsoft. Het gaat hier niet alleen om documentatie vanuit Microsoft, maar ook over code examples, namelijk de Quickstart Tutorials. Ik ben zoveel mogelijk uitgegaan van Microsoft documentatie, omdat het Bot Framework gemaakt is door Microsoft.

Sprint 1: De Echo Bot

Sprint 1 had als thema de **Echo Bot**. De bedoeling van deze sprint was om een chatbot te hebben draaien in Microsoft Teams die berichten van de gebruiker kon ontvangen en berichten naar de gebruiker toe kon sturen. Het herhalen van het verstuurd bericht door de gebruiker (vandaar de term **Echo**) was genoeg functionaliteit voor deze sprint.

Planning

De sprintbacklog voor sprint 1 zag er als volgt uit:

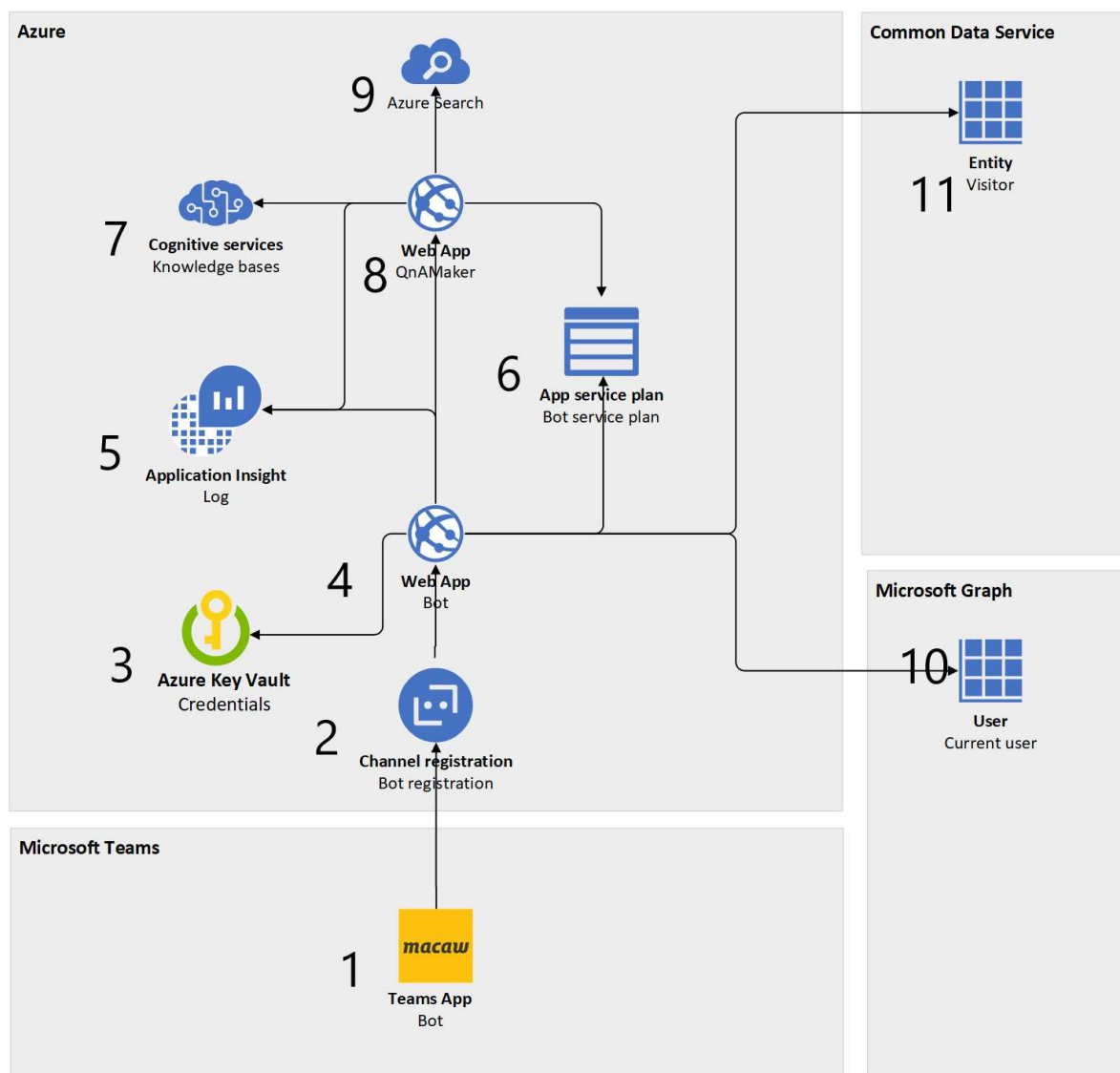
- Create design diagram
- Create Initial version of chatbot
- Setup Azure DevOps environment
- Deployment to Microsoft Azure
- Integration Microsoft Teams
- Implement logging
- Research Adaptive Cards for chatbot

Ontwerp

In de ontwerpfase van deze sprint heb ik een context diagram opgesteld. Dit diagram geeft op een hoog niveau aan hoe de onderlinge systemen communiceren.

In de figuur hieronder de finale versie van het diagram. Het diagram is tijdens de sprints aangepast. Ik heb ervoor gekozen om alleen de laatste versie te tonen.

Het grootste verschil met het initiële diagram is dat er eerst ook vanuit werd gegaan om een link te leggen met SharePoint. Dit zou nodig zijn geweest om vanuit SharePoint een lijst te kunnen exporteren en die te gebruiken als database voor de vraag-en-antwoordparen van de chatbot. Doordat dit requirement een lage prioriteit had, is deze functionaliteit niet geïmplementeerd en dus ook niet meer terug te vinden in het diagram.



De oplossing maakt gebruik van vier verschillende subsystemen. Azure is het middelpunt van de oplossing. Hier worden alle componenten aangemaakt die nodig zijn voor de chatbot.

Onze chatbot draait op Azure. Dit diagram maakt ook duidelijk dat er vanuit Azure links worden gelegd naar andere systemen.

De chatbot moet communiceren met Microsoft Graph (10) en met de Common Data Service (11). Microsoft Graph wordt gebruikt om de gegevens van de huidige gebruiker op te halen, bijvoorbeeld het land waar de gebruiker momenteel werkzaam is. De Common Data Service wordt gebruikt om te communiceren met de verschillende applicaties die worden gebruikt voor het beheren van services, voor deze versie van de chatbot gebruiken we de Macaw Visitors-applicatie om de registratie van bezoekers via de chatbot op te kunnen slaan. Tenslotte zullen de gebruikers van de chatbot beschikbaar stellen via Microsoft Teams (1).

De basic flow van de chatbot is: Er wordt een bericht vanuit Microsoft Teams (1) verstuurd naar het Azure systeem. Door middel van de Channel Registration (2) wordt de Web Applicatie: Bot (4) met onze custom code aangeroepen. De instellingen voor deze chatbot en de benodigde authenticatie wordt opgehaald vanuit de Azure Key Vault (3). Ook wordt de huidige gebruiker opgehaald doordat de Microsoft Graph API (10) wordt aangeroepen. De Web Applicatie: Bot (4) verstuurt een API request naar de Web Applicatie: QnAMaker (8) en deze raadpleegt de verschillende componenten (7 & 9) om zo een antwoord terug te kunnen sturen naar de gebruiker. Mocht er een service worden aangeroepen dan wordt er een API request verstuurd naar het Common Data Service systeem (11). De resources die in de cloud worden gebruikt worden gereserveerd door gebruik te maken van de Application Service Plan (6) en de logberichten die vanuit onze custom code komen, maar ook de logberichten die al automatisch in de Microsoft code zitten van de QnAMaker worden verstuurd en opgeslagen in de Application Insight (5). Binnen Azure is er weinig ruimte voor alternatieven voor de bepaalde componenten. Componenten zijn in Azure ontwikkeld voor één bepaald doel, dus in de meeste gevallen zijn er geen alternatieven. Hieronder volgt een korte omschrijving van de gebruikte Azure componenten.

2 Bot Channel Registration

Dit component wordt gebruikt om de Macaw Assistant in staat te stellen om te kunnen werken in Microsoft Teams. Deze instelling worden beheerd op de Azure Portal. Een alternatief voor dit component zou een Azure Bot kunnen zijn. Dit is een andere versie van deze component. Er is voor gekozen om hier geen gebruik van te maken, omdat dit component niet specifiek gericht is op de Microsoft Teams omgeving. Dit type component zou bijvoorbeeld gebruikt kunnen worden als je een chatbot toe wil voegen aan een website.

3 Azure Key Vault

De Azure Key Vault wordt gebruikt om gevoelige configuratie op te slaan. Dit is nodig zodat er niet vanaf buitenaf API requests kunnen worden uitgevoerd op onze oplossing. Dit zou ervoor kunnen zorgen dat informatie die alleen voor medewerkers bestemd is in handen zou kunnen komen van mensen die daar geen rechten toe hebben. Denk hier bijvoorbeeld aan autorisatie keys.

4 Web Application: Bot

Deze Web Application bevat de gerealiseerde code voor de Macaw Assistant Bot.

5 Application Insight

Logging is een van de ondersteunende functionaliteiten van de Macaw Assistant. Met dit component kunnen developers de logberichten van de Macaw Assistant ophalen op de Azure Portal.

6 Application Service Plan

Een Application Service Plan is nodig vanuit Azure om een of meerdere Web Applications te laten runnen. Dit component maakt resources vrij om de applicatie uit te voeren. Op de Azure Portal kan dit component worden gewijzigd om meer resources vrij te geven. Dit maakt deze applicatie schaalbaar, zoals verwacht wordt van cloudoplossingen.

7 Cognitive Services

De Azure Cognitive Services is de component dat de Knowledge Bases van de Macaw Assistant bevat. Dit onderdeel bevat twee verschillende endpoints voor de API requests:

1. Endpoint voor het beheer van de kennisbanken
2. Endpoint voor het opvragen van informatie uit de kennisbanken tijdens runtime

Het eerste eindpunt wordt niet gebruikt binnen de code van de oplossing. Voor het beheer van de Knowledge Bases wordt de web-interface gebruikt die beschikbaar is gesteld door Microsoft. Om samen te werken aan Knowledge Bases, moet de gebruiker worden toegevoegd aan de lijst met machtigingen op de Azure Portal. Dit wordt gedaan bij de Cognitive Services-component en vereist dat de gebruiker wordt toegevoegd als een bijdrage aan dit component.

De Macaw Assistant benadert tijdens runtime het tweede endpoint door middel van API requests om de vragen van de gebruiker te kunnen beantwoorden.

8 Web Application: QnAMaker

Deze Web Application component verwerkt de API requests van het Cognitive Services component. Dit component bevat vaste code toegereikt door Microsoft genaamd de QnAMaker. Dit component wordt automatisch aangemaakt als er een Cognitive Service component wordt aangemaakt op Azure.

De QnAMaker werkt op basis van een algoritme dat niet wordt vrijgegeven door Microsoft. Dit algoritme bepaald in tot hoeverre een bericht dat verstuurd wordt door een gebruiker overeenkomt met vragen binnen de kennisbanken. Als een bericht één-op-één overeenkomt met een vraag in de kennisbank dan krijgt deze een score van 1, ofwel 100%.

Ook heeft de QnAMaker een **threshold** setting. Deze setting bepaalt vanaf welke score de vragen als juist beschouwd worden en dus de antwoorden op de vragen verstuurd worden naar de gebruikers. Als uit het algoritme blijkt dat de score van een bericht tegenover een vraag in de kennisbank bijvoorbeeld 0.7 is en de threshold staat op 0.8, dan wordt het antwoord op deze vraag dus niet teruggestuurd naar de gebruiker.

9 Azure Search

Azure Search is een component dat gebruikt wordt door de Cognitive Services om de indexering van de vraag-en-antwoordparen in de kennisbanken te regelen.

De volgende tabel geeft aan welke component in welke sprint is toegevoegd:

Sprint	Toegevoegde Componenten
1	1 Microsoft Teams App 2 Bot Channel Registration 4 Web Application: Bot 5 Application Insight 6 Application Service Plan 7 Cognitive Services 8 Web Application: QnAMaker 9 Azure Search
2	3 Azure Key Vault 10 Microsoft Graph
3	11 Common Data Service
4	Geen componenten toegevoegd

Realisatie

Nadat ik de globale architectuur had gedefinieerd, ben ik begonnen met het ontwikkelen van de eerste versie van de chatbot. Zoals al eerder genoemd hoefde de eerste versie van de chatbot alleen maar berichten te kunnen ontvangen en te kunnen versturen. Het herhalen van het verstuurd bericht door de gebruiker was genoeg functionaliteit voor deze eerste versie. Als een gebruiker een bericht "Hallo" stuurde naar de chatbot, dan zou de chatbot een bericht "Hallo" terugsturen. In sprint 0 is de keuze gemaakt om een Quickstart Tutorial van Microsoft te gebruiken voor de initiële versie. Deze Quickstart Tutorial bevatte ook al functionaliteit voor de QnA Bot. In overleg met mijn begeleider is er gekozen om gelijk voor deze versie te gaan, omdat dit de overgang naar de volgende versie van de chatbot (sprint 2) zou versnellen. In eerste instantie zou deze versie van de chatbot lokaal draaien.

Toen de chatbot eenmaal lokaal draaide is de DevOps omgeving ingericht en zijn de verschillende benodigde componenten voor Azure aangevraagd en aangemaakt. Er is een repository voor de code aangemaakt op DevOps, zodat hier de code van de chatbot kon worden opgeslagen en door kon worden gestuurd naar de web applicatie draaiende in Azure. Deze deployment naar Azure gebeurde in deze sprint nog handmatig.

De volgende stap was om de chatbot te koppelen met een Microsoft Teams applicatie. De link van Azure naar Microsoft Teams wordt geregeld in het Channel Registration component op Azure. Hier wordt aangegeven dat de chatbot berichten kan ontvangen en versturen via Microsoft Teams. Ook moest er een application package worden aangemaakt voor Microsoft Teams met het endpoint van de chatbot.

Een vereiste van deze sprint was ook het implementeren van logging. De logging is opgezet binnen de code van de chatbot en de logs kunnen worden uitgelezen via de Azure Portal in het Application Insight component. Belangrijk hierbij was dat de benodigde keys goed gekoppeld waren in de code van de chatbot. Deze keys waren belangrijk, omdat anders de logberichten niet zichtbaar werden binnen de Application Insight draaiende in de cloud. Een voorbeeld van een simpel logbericht was bijvoorbeeld het loggen van een nieuwe gebruiker. Ook was configureerbaarheid een van de acceptatiecriteria voor deze versie van de chatbot. In eerste instantie is de configuratie opgeslagen in de appsettings.json van de source code. Later is deze verplaatst naar een Azure Key Vault, zodat gevoelige waarden afgeschermd konden worden en niet iedereen de configuratie kon aanpassen. Hier moeten namelijk rechten voor worden gegeven op de Azure Portal. Om te voorkomen dat er constant tijdens het ontwikkelen naar de Azure Portal moest worden gegaan om settings aan te passen, is deze Key Vault pas later geïmplementeerd in het systeem. Het is belangrijk om de gevoelige waarden af te schermen, omdat er anders direct naar de API requests konden worden verstuurd. Hiervoor is wel een authenticatie key nodig en deze moet dus afgeschermd worden. Dit beschermt eventuele gevoelige informatie die Macaw alleen aan de medewerkers wil verstrekken.

Tenslotte heb ik de rest van de sprint gebruikt om onderzoek te doen naar het gebruik van Adaptive Cards in Microsoft Teams. Het versturen van formulieren of knoppen zou belangrijk worden in de komende sprints en dus was het belangrijk dat er duidelijkheid kwam over de mogelijkheden van deze Adaptive Cards.

Test

In de eerste sprint is gekeken naar de mogelijkheden voor het testen van de chatbot. In eerste instantie is er gekeken naar het unittesten van de verschillende onderdelen van de code van de chatbot. Dit bleek echter niet efficiënt, omdat in dat geval de verstuurde of ontvangen berichten gemoockt moesten worden. Elk object dat een bericht bevat, heeft ontzettend veel parameters (meer dan 50) die elk ook weer uit objecten met parameters bestaan. Het mocken van zo'n object wordt dus heel erg complex. Uitgebreid testen op gebruikersniveau is daarom een veel zinvollere manier van testen dan het proberen te unittesten. In overleg met mijn begeleider is besloten om gebruik te gaan maken van gebruikerstestcases die opgenomen worden in de DevOps omgeving.

Het testen in de DevOps omgeving werkt als volgt:

1. Er wordt een test aangemaakt voor een bepaalde functionaliteit
2. De stappen die moeten worden ondernomen om de functionaliteit te testen worden beschreven met de verwachte uitkomsten van iedere stap
3. De test wordt gestart en handmatig worden de stappen uitgevoerd
4. Wanneer alle stappen doorlopen zijn is er de mogelijkheid om of feedback te geven bij bepaalde stappen waar een andere uitkomst werd ondervonden dan de verwachte uitkomst of alle stappen zijn goed doorlopen en de test is dus geslaagd

Review

Er zijn twee belangrijke beslissingen genomen tijdens de review van sprint 1. Allereerst is ervoor gekozen om de koppeling met SharePoint te laten vallen. In overleg met de HR-manager die, in eerste instantie, de kennisbanken (database) van de chatbot zou beheren is duidelijk geworden dat het onderhouden van deze kennisbanken goed te doen was via de web-interface die aangeboden wordt

vanuit Microsoft. Echter was het wel belangrijk dat hier een goede handleiding voor werd geschreven die alle functionaliteiten zou bevatten. Dit nieuwe work item is opgenomen in de sprintbacklog van de volgende sprint.

Ten tweede is er besloten om gebruik te gaan maken van een ARM-template voor het deployen van de benodigde componenten, code en configuratie voor de verschillende omgevingen. We hebben het hier dan vooral over een testomgeving en de productieomgeving. Een Azure Resource Manager template is dus een template dat gebruikt wordt voor het aanmaken van nieuwe componenten binnen Azure of het updaten van bestaande componenten. Het ARM-template automatiseert een groot deel van de deployment. Tevens is het gebruik van ARM-templates erg standaard voor de werkwijze van Macaw. De templates verbeteren namelijk de overdraagbaarheid van de projecten, doordat er minder onderzoek hoeft plaats te vinden naar de verschillende Azure componenten bij het overdragen van het project. De templates in combinatie met de DevOps pipelines zorgen voor de **Continuous Deployment** en **Continuous Integration** bij de projecten van Macaw.

Een openstaande vraag na deze review was de vraag of de QnA chatbot kon worden uitgebreid met de services functionaliteit of dat er uiteindelijk twee chatbot zouden moeten worden ontwikkeld.

Onderzoek naar deze vraag is opgenomen voor de volgende sprint.

Tijdens deze sprint heb ik de volgende beroepstaken uitgevoerd:

- **C6: Ontwerpen software**
De initiële versie van het systeem is ontworpen in deze sprint.
- **D14: Realiseren van software**
Ik ben bezig geweest met het implementeren van verschillende logberichten in de code, zodat er meer informatie beschikbaar werd over het proces van de gemaakte oplossing.
- **D15: Testen**
Ik heb nagedacht over hoe ik het testen van deze applicatie wilde aan gaan pakken. Zoals al eerder genoemd is er besloten om geen unittests te schrijven, maar op een hoger niveau te testen, zodat er meer tijd beschikbaar blijft voor het implementeren van de benodigde functionaliteiten.
Tevens is er een testcase opgesteld om te controleren of de echo functionaliteit werkte. Door te testen of deze functionaliteit naar behoren werkte is er gecontroleerd of de berichten die de gebruiker stuurde naar de chatbot aankwamen, maar ook of de berichten die de chatbot terugstuurde naar de gebruiker aankwamen. Later is deze testcase verwijderd, omdat de echo functionaliteit niet behoorde tot de gewenste functionaliteit. Deze functionaliteit is in deze sprint alleen toegevoegd om te kijken of het sturen en ontvangen van berichten vanuit de chatbot in eerste instantie werkte.
- **D17: Configureren**
Het opzetten van de Azure omgeving met bijbehorende settings is opgepakt in deze sprint. Ook moesten de settings voor de initiële chatbot goed gezet worden, zodat er mee gewerkt kon worden.
- **Gf: Leren leren**
Tijdens het onderzoek in deze sprint ben ik voornamelijk uitgegaan van Microsoft documentatie. Ik heb bijvoorbeeld documentatie geraadpleegd over hoe de logging werkt via het component

van Application Insight. De technologie van Adaptive Cards is ook van Microsoft, dus ik daarvoor heb ik ook de bijbehorende documentatie geraadpleegd.

Sprint 2: De QnA Bot

Deze sprint bestaat uit het ontwikkelen van gewenste functionaliteiten die niet standaard vanuit de Quickstart aangeboden werden zoals de multi-turn conversation en het gebruik van locatie.

Planning

De sprintbacklog voor sprint 2 zag er als volgt uit:

- Implement multi-turn conversations
- Implement location usage
- Write **Usage Manual Knowledge Bases** for Knowledge Base owners
- Train Knowledge Base owner
- Configure Active Learning
- Set up Test environment
- Research QnA and Service Bot combination

Ontwerp

Tijdens de ontwerpfase van deze sprint is het SharePoint systeem verwijderd uit het eerder getoonde diagram. Dit op basis van de review van sprint 1. De functionaliteit om de kennisbanken van de chatbot te beheren via SharePoint is niet verwijderd van de productbacklog, maar heeft uiteindelijk de laagste prioriteit gekregen. Mogelijk dat in de toekomst deze functionaliteit verwezenlijkt kan worden.

Realisatie

De uitvoering van deze sprint is begonnen met het implementeren van de multi-turn conversation functionaliteit. Multi-turn conversation wordt toegepast als de chatbot meer informatie nodig heeft om een vraag te beantwoorden. Wanneer dit optreedt dan stelt de chatbot de gebruiker een vervolgvraag om de benodigde informatie op te halen. Om te illustreren een voorbeeldscenario. De gebruiker wil graag weten wat het adres is van het kantoor van Macaw te Barendrecht. De gebruiker stelt de vraag: **What is the office's address?** De chatbot reageert met de vraag: **Which office are we talking about?** en geeft een aantal keuzes, bijvoorbeeld **Hoofddorp, Barendrecht, Düsseldorf**. Deze opties worden als Adaptive Cards verstuurd naar de gebruiker, zodat dit klikbare knoppen zijn. Wanneer de gebruiker klikt op de knop **Barendrecht** wordt het adres van het kantoor van Macaw in Barendrecht verzonden naar de gebruiker. Er is voor gekozen om de opties als klikbare knoppen te versturen naar de gebruiker, zodat het antwoord op de vervolgvraag altijd correct is. Dit voorkomt foutieve antwoorden op de vraag van de chatbot.

De HR-manager kaartte terecht aan in de review van de vorige sprint dat sommige antwoorden verschillen per land. Als een gebruiker bijvoorbeeld de lunchkosten wil opvragen voor zijn of haar kantoor, dan is het vervelend om constant de procedure van een multi-turn conversation te moeten doorlopen. Daarom is ervoor gekozen om aan relevante vragen **metadata** te koppelen. Deze metadata bevat het land waar het antwoord van de vraag betrekking op heeft. Een vraag die alleen relevant is in Nederland zal dus als metadata **location:netherlands** bevatten. Doordat er in de code van de chatbot een koppeling is gelegd aan de Microsoft Graph API kan de locatie van een ingelogde gebruiker in Microsoft Teams opgehaald worden. Hierna kunnen niet relevante vragen dus gefilterd worden door de

locatie van de gebruiker te vergelijken met de metadata aangegeven voor de vraag. Tevens kan door deze functionaliteit ook gespeeld worden met de talen van de verschillende landen. Vragen en antwoorden die bijvoorbeeld alleen relevant zijn voor de medewerkers van Macaw Litouwen zouden eventueel in het Litouws kunnen worden getoond.

De volgende stap was het schrijven van de handleiding voor de HR-manager die initieel de kennisbanken zou beheren. Deze handleiding is te vinden in de bijlagen onder Bijlage 3: Manual Knowledge Base Management. Na het afronden van de handleiding is deze ook doorgesproken met de HR-manager en is de feedback op dit document verwerkt. De HR-manager is nu in staat de content aan te leveren voor de chatbot draaiende in de productieomgeving.

Tijdens deze sprint heb ik ook onderzoek gedaan naar een functionaliteit vanuit Microsoft, namelijk Active Learning. Door het bestuderen van verschillende documentatie van Microsoft over hun Bot Framework ben ik achter deze zeer bruikbare functionaliteit gekomen. Active Learning is een techniek die kennisbanken automatisch analyseert en verbeteringen suggereert waar relevant. Het is erg makkelijk om deze functionaliteit aan te zetten via de web-interface waar de kennisbanken beheerd worden.

In de review van de vorige sprint besloot de product owner om gebruik te gaan maken van een ARM-template. Dit template is uitgewerkt in deze sprint en gebruikt om de testomgeving op te zetten. Tevens is de Azure Key Vault gelijktijdig geïntegreerd door middel van het gebruik van dit template. Het template heeft direct geholpen bij de continuous development en deployment van dit project. Vanaf dit moment wanneer er vanuit de repository een pull request werd aangemaakt voor de master branch van de repository werd er gelijk gedeployed en gereleased naar de testomgeving.

Uit het onderzoek naar de mogelijkheid om de QnA Bot met een Service Bot te combineren, is gebleken dat dat inderdaad mogelijk is. Het combineren van deze bots is wenselijk, zodat de gebruikers van Macaw niet meerdere chatbots zouden hoeven te gebruiken, wat de oplossing minder laagdrempelig zou maken.

De functionaliteit werkt als volgt:

De functie die de Adaptive Cards aanmaakt heeft als parameter een file path naar de resource.json. Als antwoord op de vraag in de kennisbank kan zo een file path geretourneerd worden waarmee deze methode wordt aangeroepen. Zo blijft de Adaptive Card goed aanpasbaar (omdat deze nog in de source code staat) en kan zo elke Adaptive Card verstuurd worden naar de gebruiker.

Tevens wordt er als metadata aan het antwoord (de file path) een metadata tag meegegeven. Deze tag heeft een "Name": "extension" en een "Value": "json". Zo kunnen er condities in de code van de Bot worden aangemaakt die kijken of de methode om de Adaptive Cards aan moet worden geroepen of dat er gewoon een regulier antwoord (in plain text) wordt gegeven en deze verstuurd wordt naar de gebruiker.

Test

Tijdens de testfase van deze sprint zijn een aantal testscenario's wederom toegevoegd aan de DevOps omgeving. Ook zijn er testaccounts met ieder een licentie voor Microsoft Teams aangemaakt, zodat de locatie metadata van de QnA chatbot kon worden getest op de nieuw aangemaakte testomgeving.

Omdat er nu met meerdere omgevingen en dus meerdere kennisbanken werden gebruikt zijn er ook een aantal testsets met vraag-en-antwoordparen opgesteld.

Review

Tijdens de review van sprint 2 is er besloten om twee nieuwe functionaliteiten op te nemen in de aankomende sprint. Deze functionaliteiten hadden betrekking tot het QnA gedeelte van de chatbot. Er is besloten om een lijst met gerelateerde vragen te versturen naar de gebruiker. Dit zou de gebruiker helpen om sneller tot zijn of haar antwoord te komen. Ook is er besloten om een knop toe te voegen aan antwoorden die gebaseerd waren op bronnen. Dit zou handig zijn als een gebruiker meer informatie wilde hebben over een specifiek onderwerp. Deze work items zijn opgenomen in de sprintbacklog van de aankomende sprint.

De onderzoeksresultaten over een eventuele combinatie van de QnA Bot en de Services Bot zijn ook doorgesproken. De mogelijkheid was aanwezig om beide versies van de chatbot te combineren en hier is ook voor gekozen, zodat de gebruiker niet meerdere bots zou hoeven te gebruiken. Het is makkelijker voor de gebruiker om alles via één chatbot te kunnen regelen.

Er is ook gekozen om voor de te implementeren services de focus te leggen op het aanmelden van een bezoeker. Het concentreren op één service in plaats van op meerdere zou bijdragen aan het leveren van een compleet tussenproduct door de beperkte tijd.

Tenslotte is er ook gevraagd aan de HR-manager voor content voor de kennisbanken van de chatbot. De HR-manager verwees ons in eerste instantie door naar de website van Macaw waar een lijst met veel gestelde vragen stond. Echter bleek al gauw in de volgende sprint dat deze lijst onvoldoende compleet was om een goede kennisbank op te kunnen stellen. De daadwerkelijke lijst met vragen en antwoorden liet dus nog op zich wachten. Deze kennisbank is nodig op het moment dat de chatbot in productie wordt genomen en de chatbot is niet in productie genomen tijdens de periode van mijn stage, dus dit heeft niet voor vertraging gezorgd.

Tijdens deze sprint heb ik de volgende beroepstaken uitgevoerd:

- **D14: Realiseren van software**

Er zijn twee grote functionaliteiten opgepakt met betrekking op het schrijven van code. Dit zijn de functionaliteiten van de multi-turn conversations en de implementatie van het gebruik van locatie metadata.

- **D15: Testen**

De bovenstaande functionaliteiten moesten uiteraard ook getest worden.

- **Gf: Leren leren**

Ook deze sprint heb ik voornamelijk Microsoft documentatie geraadpleegd voor het uitzoeken van de mogelijkheden om de verschillende chatbots te combineren en voor het implementeren van de functionaliteiten.

Sprint 3: De Service Bot

De derde sprint had als doel services aan te bieden. Gekozen werd het implementeren van de service van het aanmelden van een bezoeker. Ook waren er nieuwe functionaliteiten (source button en related questions) toegevoegd vanuit de review van de vorige sprint die gerealiseerd moesten worden.

Planning

De sprintbacklog voor sprint 3 zag er als volgt uit:

- Configure HR content
- Create source button
- Related or suggested questions
- Implement visitor registration service

Ontwerp

Tijdens de ontwerpfase van deze sprint is het Common Data Service systeem opgenomen in het diagram. Het CDS is waar de data van het aanmelden van een bezoeker wordt opgeslagen. Dit systeem wordt daarna gebruikt door de receptie van Macaw om de planning door te kunnen nemen. Het aanmelden van een bezoeker vond voor de chatbot plaats door een mail met de informatie door te sturen naar de receptie die deze data verwerkte in het CDS. Wanneer de bezoeker arriveerde bij Macaw werd de medewerker opgebeld. Het proces van het mailen naar de receptie is nu uit handen genomen door de chatbot.

Realisatie

De realisatie is begonnen met het overzetten van de veel gestelde vragen die op het moment aangegeven waren op de website van Macaw. De beperkte set van vragen en antwoorden zijn toegevoegd aan de kennisbanken van de chatbot. Ook is er besloten om een aparte kennisbank aan te maken waarin de services van de chatbot werden opgeslagen. Deze keuze is gemaakt, omdat de HR-manager niet verantwoordelijk is voor het onderhouden van de services. Ondanks dat dit work item geen betrekking heeft tot de Service Bot is dit work item wel opgenomen in deze sprint. Dit komt omdat dit work item in de review van de voorafgaande sprint is toegevoegd aan de backlog.

Daarop volgend heb ik de functionaliteiten toegevoegd die aan bod kwamen bij de vorige review. Ik ben begonnen met het implementeren van een source knop onder relevante vragen, zodat de medewerkers meer informatie konden krijgen over specifieke onderwerpen. Deze functionaliteit is gerealiseerd door een **metadata** veld toe te voegen genaamd **source:url** waar de URL van het relevante brondocument werd geplaatst. De code van de chatbot controleert of een vraag-en-antwoordpaar bron metadata bevat en stuurt deze bron als een knop onder het gegeven antwoord. Klikken op de knop zorgt ervoor dat de URL geopend wordt in een browser.

Hierna ben ik aan de slag gegaan met de functionaliteit van de gerelateerde vragen. Zoals al eerder genoemd bepaalt het algoritme van de chatbot of vragen een match zijn of niet. Wat ik heb toegevoegd om gerelateerde vragen ook op te nemen is dat vragen die een score hebben die volgens de **threshold** setting van de **QnAMaker** worden opgenomen in een lijst met mogelijke antwoorden. De vraag met de hoogste score wordt beantwoord en dus verstuurd naar de gebruiker en hierna wordt er voor elke vraag die nog in de lijst zit een button verstuurd per gerelateerde vraag. De gebruiker kan dus gewoon doorklikken en zo antwoorden krijgen op gerelateerde vragen. Om te zorgen dat er niet te veel

gerelateerde vragen worden verstuurd heb ik ook een nieuwe setting opgenomen genaamd **MaximumNumberOfAnswersToReturn**. Standaard hebben we deze op de value **3** gezet, zodat er maximaal twee gerelateerde vragen worden meegestuurd met het antwoord (het antwoord en de twee gerelateerde vragen zijn drie antwoorden in totaal).

De sprint is afgesloten met de implementatie van het aanmelden van een bezoeker. Tijdens de implementatie waren er nog wat problemen met het zorgen dat de chatbot rechten had om records aan te maken binnen het CDS. Samen met een andere medewerker van Macaw die veel verstand had van het CDS heb ik dit weten op te lossen. De chatbot verstuurt nu een formulier naar de gebruiker waarin alle informatie over het bezoek wordt ingevuld. De data van het formulier wordt uitgelezen en hier wordt een JSON object voor aangemaakt die vervolgens met een POST request naar het CDS wordt toegestuurd. Er is ook rekening gehouden met bijvoorbeeld dubbele records of formulieren waar data foutief is ingevoerd.

Test

Tijdens de testfase van deze sprint zijn een aantal relevante testscenario's opgesteld met betrekking tot het aanmelden van een bezoeker. Ook zijn er een aantal scenario's toegevoegd om te testen of de knop voor het brondocument werkt naar behoren en scenario's over het versturen van gerelateerde vragen.

Review

Tijdens de review van sprint 3 kwam er een grote bug naar voren bij de functionaliteit van het aanmelden van een bezoeker. Het object dat werd aangemaakt tijdens de aanmelding was een globaal object. Dit had als gevolg dat als er twee gebruikers tegelijkertijd bezig waren met het aanmelden van een bezoeker dat dit object werd overschreven door de laatste gebruiker. Een typisch geval van een concurrency issue en deze bug is dus opgenomen in de sprintbacklog van sprint 4.

Tijdens deze sprint heb ik de volgende beroepstaken uitgevoerd:

- **D14: Realiseren van software**
Tijdens deze sprint heb ik code geschreven voor de functionaliteiten van het toevoegen van een source knop, het versturen van gerelateerde vragen en het aanmelden van een bezoeker via het CDS.
- **D15: Testen**
Deze functionaliteiten moesten getest worden. Ook heb ik de testcases van de vorige sprint opnieuw doorlopen om er zeker van te zijn dat deze nog steeds werkten naar behoren.
- **Gf: Leren leren**
Tijdens deze sprint heb ik veel overleg gehad met een collega bij Macaw. Deze collega had veel ervaring met het Common Data Service systeem. Doordat hij mij heeft uitgelegd hoe het CDS werkt heb ik de problemen bij de koppeling met dit systeem kunnen oplossen.

Sprint 4: Productie

Sprint 4 was bedoeld om de laatste loodjes af te werken. De deployment naar de productieomgeving moest worden geregeld in deze sprint. Tijdens deze sprint zijn de technische documenten gemaakt zodat de overdraagbaarheid van de Macaw Assistant geregeld werd. Het werken aan de documenten was een iteratief proces. Hiermee bedoel ik dat de documenten regelmatig gereviewd moesten worden om te zorgen dat de documenten kwalitatief in orde waren en er opgeschreven werd wat belangrijk was voor Macaw.

Planning

De sprintbacklog voor sprint 4 zag er als volgt uit:

- Fix concurrency bug
- Write **Deployment Manual** document
- Write **Technical Design** document
- Create Teams package
- Deploy to production environment

Ontwerp

Er heeft geen ontwerpfase plaatsgevonden tijdens deze sprint.

Realisatie

De sprint begon met het oplossen van de concurrency bug die geconstateerd was tijdens de review van de vorige sprint. De bug is uiteindelijk verholpen door geen gebruik meer te maken van een globaal object waarin de data werd opgeslagen en het object opnieuw uit te lezen door een bevestigingsbericht met alle informatie die nodig was om het aanmelden te voltooien en dit nieuwe object te gebruiken in de POST request naar het CDS.

Tijdens deze sprint heb ik ook een tweetal documenten geschreven. Het eerste document is de **Deployment Manual**. Dit document beschrijft welke stappen er moeten worden genomen om de complete solution werkende te krijgen. Dit document is nodig als de solution opnieuw moet worden aangemaakt of als de chatbot ingezet moet worden bij andere bedrijven. Het document is te vinden in de bijlagen onder Bijlage 4: Deployment ManualBijlage 4: Deployment Manual en bevat 26 pagina's.

Ook heb ik het **Technical Design** document geschreven. Dit document beschrijft de functionaliteiten van de chatbot, de werking van de componenten in Azure en gaat bijvoorbeeld in op zaken zoals security en non-functional requirements. Ook dit document is te vinden in de bijlagen onder Bijlage 5: Technical Design en bevat 22 pagina's.

Tenslotte zijn alle zaken omtrent het deployen naar de productieomgeving geregeld. Alle permissions voor de verschillende systemen (CDS, Microsoft Graph, etc.) zijn aangevraagd bij de service desk van Macaw. Tevens is er een nieuwe Microsoft Teams applicatie aangemaakt voor de medewerkers van Macaw. Het gaat hier om de applicatie die zal worden gebruikt in de productieomgeving.

Test

De testfase van deze laatste sprint bestond uit het nog eenmalig doorlopen van alle opgestelde testscenario's op DevOps, maar dan in de productieomgeving van Macaw. Dit zorgde ervoor dat we

zeker wisten dat alle rechten met betrekking tot systemen van buiten Azure (bijvoorbeeld het CDS) goed functioneerde.

Review

Naast het bespreken van de oplossing voor de concurrency bug zijn de finale versies van de verschillende documenten doorgenomen met de product owner om te kijken of deze voldoende zijn voor de overdraagbaarheid van het project.

- **D14: Realiseren van software**

Tijdens deze sprint heb ik een bug opgelost om het concurrency probleem op te lossen tijdens het aanmelden van een bezoeker.

Ook heb ik twee documenten geschreven met betrekking op het gemaakte systeem. Deze documenten bevorderen de overdraagbaarheid van het project.

- **D15: Testen**

Tijdens deze sprint was het vooral belangrijk om de regressietesten te doorlopen, aangezien er wijzigingen plaats vonden in de code van het aanmelden van een bezoeker. Het was belangrijk om te zorgen dat de bug verholpen werd, maar dat de standaard functionaliteit nog steeds werkte.

Evaluatie

In dit hoofdstuk volgt de evaluatie van het gehele project.

Afwijking afstudeerplan

Er zijn twee afwijkingen ten opzichte van het afstudeerplan. Allereerst de onverwachte langere doorlooptijd door ziekte en onbeschikbaarheid van de product owner. Ten tweede was in het afstudeerplan aangegeven dat de laatste sprint gewijd zou worden aan de integratie met Microsoft Teams. Tijdens het afstuderen is ervoor gekozen om de integratie met Microsoft Teams naar voren te halen en in elke sprint te zorgen dat er een versie van de chatbot kan draaien binnen Microsoft Teams. We hebben hiervoor gekozen, omdat er zo elke sprint een geheel afgerond tussenproduct kon worden geleverd die, in theorie, gedeployed kon worden naar de productieomgeving van Macaw. Deze wijziging zorgde er dus ook voor dat het thema voor de laatste sprint is gewijzigd van **Microsoft Teams** naar **Production**.

Proces

Het proces van het ontwikkelen van de Macaw Assistant verliep goed. De gekozen elementen uit de SCRUM-methode werkten goed. De SCRUM-meetings (daily stand-ups, sprintplannings en sprintreviews) waren vooraf ingepland en stonden dus al gelijk vast.

Iets wat ik in een volgend project anders zou aanpakken is het gebruik van retrospectives. Er was gekozen om deze meetings niet in te plannen en dit resulteerde erin dat deze meetings eigenlijk niet plaatsvonden. Tijdens dit project was dit niet heel erg, omdat het team eigenlijk alleen maar uit mijzelf en mijn begeleider bestond en er tijdens de daily stand-ups werd regelmatig gereflecteerd over het proces. Toch denk ik dat het nuttig was geweest om bij een sprint retrospective meer high level feedback te krijgen over het proces en het functioneren van het team.

Ook heb ik als begeleider iemand gehad die heel erg druk bezig was bij Macaw zelf. Dit is natuurlijk niet raar, maar het zorgde er af en toe wel voor dat ik een relatief lange tijd met vragen bleef zitten.

Product

Peter maar ook ik zijn erg tevreden over de uiteindelijke versie van de Macaw Assistant. De chatbot bevat alle functionaliteit die vanuit de requirements naar voren kwam. Ik vind het alleen jammer dat de chatbot nog niet in gebruik is door Macaw. Op het moment zijn we nog aan het wachten tot er iemand vanuit HR tijd heeft om de juist vraag-en-antwoordparen in te voeren. Hierna zal er met de Marketing afdeling gecommuniceerd worden en zal de chatbot uitgerold worden naar alle medewerkers van Macaw.

Ook denk ik dat de chatbot nog wel wat gebruiksvriendelijker kan worden gemaakt. Sommige berichten die de chatbot verstuurt kunnen nog wel voor wat onduidelijkheid zorgen bij de medewerkers. Als de chatbot bijvoorbeeld een vraag niet herkent dan zou het mooi zijn als de chatbot aangeeft naar de gebruiker dat hij of zij feedback kan geven. Op het moment wordt de lijst met onbeantwoorde vragen wel gelogd, maar wij geven dit bijvoorbeeld niet door aan de gebruiker.

Peter is ook blij met hoe overdraagbaar het project is. De geschreven documenten zijn duidelijk en de code van de chatbot is volledig voorzien van commentaar.

Uiteraard ben ik blij met de positieve feedback die ik van mijn begeleider heb gekregen.

Beroepstaken

A1: Analyseren probleemdomein & opstellen probleemstelling

Ik heb voldaan aan deze beroepstaak tijdens het schrijven van het **Plan van Aanpak** document. In dit document heb ik helder beschreven wat de huidige situatie was vanuit Macaw en hoe de chatbot deze zou moeten verbeteren. In het document is de probleemstelling helder geformuleerd.

Gc: Kritisch en methodisch werken

De methode die gebruikt is tijdens dit project is SCRUM. Door gebruik te maken van de artifacts van SCRUM (daily stand-ups, sprintplanning en sprintreview) heb ik voldaan aan het methodisch werken. Tijdens het onderzoeken van verschillende onderwerpen ben ik ook kritisch gebleven. Ik heb waar zover mogelijk gebruik gemaakt van documenten die aangeleverd werden door Microsoft. Doordat het Bot Framework opgezet is door Microsoft acht ik deze informatie als een feitelijke, en dus betrouwbare, bron.

Gf: Leren, leren

Ik heb een hoop geleerd van deze stage. Allereerst heb ik geleerd over het Bot Framework en dus over het ontwikkelen van een chatbot. Ook heb ik meer geleerd over het koppelen van systemen met betrekking tot authenticatie en autorisatie, het omgaan met API's vanuit code, het werken met ARM-templates en het gebruik van Adaptive Cards via json files.

Het leren over deze verschillende onderwerpen is voornamelijk gedaan door relevante documentatie op te zoeken. In eerste instantie waren dit vooral documenten verstrekt door Microsoft, omdat ik met hun framework aan de slag ben gegaan. Als ik tegen bepaalde problemen aanliep tijdens het project was het ook altijd goed om ruggespraak te houden met verschillende collega's die ervaring hadden met de betreffende onderwerpen. Ik ben erachter gekomen dat het erg goed is om problemen door te spreken, omdat ik als programmeur vaak gericht ben op één bepaalde oplossing voor een probleem. Het doorspreken met collega's helpt om je uit deze tunnelvisie te halen, want er zijn vaak nog veel meer oplossingen voor het bepaalde probleem.

A3: Vergaren en analyseren van requirements

Ik heb voldaan aan deze beroepstaak door te beginnen met mijn requirements analyse. De analyse is te vinden onder de kop Requirements analyse. Ik heb de belanghebbenden van dit project gesproken om de requirements te achterhalen en daarna samen met de product owner de requirements geprioriteerd.

D14: Realiseren van software

Ik heb veel code geschreven tijdens dit project. In eerste instantie heb ik gebruik gemaakt van een Quickstart Tutorial van Microsoft om initiële versie van de chatbot op te zetten. Hierna zijn alle functionaliteiten die beschreven zijn in dit verslag als custom code toegevoegd aan deze initiële versie. In totaal heeft de solution 3700 regels code waarvan er ongeveer 200 gebruikt zijn van de code verstrekt door Microsoft in de Quickstart Tutorial.

Tijdens het programmeren heb ik mij gericht op een aantal principes van het ontwikkelen van software. Deze principes waren:

- **Single Responsibility principle**

Ik heb mij erg gericht op object-oriented programming. De solution bestaat uit veel klassen die allemaal hun eigen, enkele taak hebben. Zo blijft de code van de solution overzichtelijk en daarmee onderhoudbaar.

- **Reusability**

Ook heb ik mij gericht op reusability. Een secundaire doelstelling van dit project was dat de technologie van chatbots ook kon worden gebruikt bij klanten van Macaw die daar belangstelling voor hebben. Deze solution is uitstekend geschikt om te implementeren bij andere projecten. Het enige dat moet worden aangepast is het aanpassen van de verschillende settings en keys. Doordat dit dynamisch wordt opgehaald vanuit de Azure Key Vault is dit gemakkelijk aan te passen. Ook is er een **Deployment Manual** toegevoegd die te vinden is in de bijlagen. Dit document legt stapsgewijs uit hoe de solution moet worden geconfigureerd om te werken in een andere omgeving.

- **Maintainability**

Ik heb mij ook gericht op de maintainability van het project. Dit heb ik gedaan door veel commentaar in de code te schrijven. Elke functie, field, parameter en class bevat commentaar, zodat de code makkelijk te lezen is. Tevens heb ik het **Technical Design** document geschreven, zodat er op hoog niveau kan worden gelezen welke beslissingen er zijn genomen tijdens het project betreffende de solution.

- **Quality Assurance**

Als laatste heb ik door middel van het gebruik van SonarQube gezorgd voor een hoge kwaliteit van de code. Door na elke sprint deze tool te laten draaien en de code te analyseren op **bad smells** is de code regelmatig gerefactored om de kwaliteit van het project te bewaken.

Het schrijven van alle code ging uiteraard niet altijd vlekkeloos. Vooral het koppelen van het gehele Azure systeem met de twee andere systemen (Microsoft Graph en CDS) verliep moeizaam. De autorisatie en authenticatie van de chatbot met deze systemen heeft veel tijd in beslag genomen. Dit komt omdat ik gewend was vanuit de projecten op de Haagse Hogeschool om, wanneer er werd gewerkt met web applicaties, gebruik te maken van bijvoorbeeld Google voor autorisatie en authenticatie. Dit gebeurde altijd op gebruikersniveau, ofwel een persoon logt in met zijn of haar Google-account en krijgt zo rechten op de applicatie. Tijdens dit project moest niet een persoon inloggen om rechten te krijgen, maar juist de applicatie zelf. Uiteindelijk heb ik deze problemen opgelost door het goed bestuderen van documentatie (niet alleen van Microsoft, maar ook van de betreffende subsystemen) en het inschakelen van hulp van collega's bij Macaw. Het Bot Framework was weliswaar nieuw voor de collega's van Macaw, maar het integreren van de subsystemen met Azure uiteraard niet. Vaak was een gesprek met een kop koffie genoeg om duidelijkheid te krijgen over hoe de systemen gekoppeld moesten worden. Ook werd er vaak door de collega's doorverwezen naar projecten waarin de systemen gekoppeld waren. Door deze solutions te bestuderen heb ik alle problemen kunnen verhelpen.

D15: Testen

Ik heb ook voldaan aan de beroepstaak testen. Ik heb 17 testscenario's geschreven in Microsoft DevOps die de hele chatbot te controleren.

Tijdens de testfases van de sprints heb ik mij beziggehouden met twee zaken:

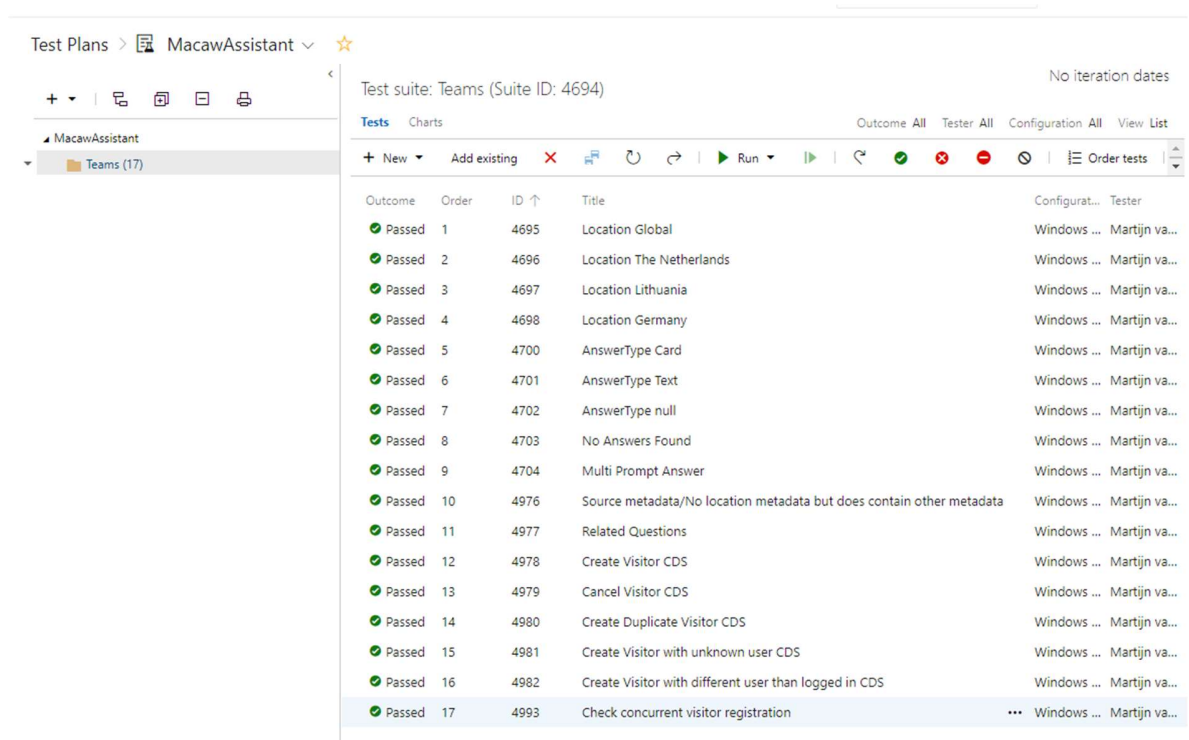
- **Test Driven Development**

Voorafgaande aan het programmeren van elke functionaliteit heb ik nagedacht over welke mogelijke uitkomsten de functionaliteit zou kunnen opleveren. Ik maakte op de DevOps omgeving op dat moment lege testcases aan met een dekkende titel, zodat ik, nadat de functionaliteit was geïmplementeerd, invulling kon geven aan deze testcases.

- **Regressie testen**

Door aan het einde van elke sprint ook de testcases uit te voeren van de voorafgaande sprints ben ik ook regelmatig bezig geweest met regressie testen. Deze werkwijze zorgde ervoor dat, ondanks het implementeren van nieuwe functionaliteiten, de oude functionaliteiten nog steeds werkten zoals bedoeld.

Hieronder een overzicht van alle opgestelde testcases.



Test Plans > MacawAssistant

Test suite: Teams (Suite ID: 4694)

No iteration dates

Outcome	Order	ID	Title	Configurat...	Tester
Passed	1	4695	Location Global	Windows ...	Martijn va...
Passed	2	4696	Location The Netherlands	Windows ...	Martijn va...
Passed	3	4697	Location Lithuania	Windows ...	Martijn va...
Passed	4	4698	Location Germany	Windows ...	Martijn va...
Passed	5	4700	AnswerType Card	Windows ...	Martijn va...
Passed	6	4701	AnswerType Text	Windows ...	Martijn va...
Passed	7	4702	AnswerType null	Windows ...	Martijn va...
Passed	8	4703	No Answers Found	Windows ...	Martijn va...
Passed	9	4704	Multi Prompt Answer	Windows ...	Martijn va...
Passed	10	4976	Source metadata/No location metadata but does contain other metadata	Windows ...	Martijn va...
Passed	11	4977	Related Questions	Windows ...	Martijn va...
Passed	12	4978	Create Visitor CDS	Windows ...	Martijn va...
Passed	13	4979	Cancel Visitor CDS	Windows ...	Martijn va...
Passed	14	4980	Create Duplicate Visitor CDS	Windows ...	Martijn va...
Passed	15	4981	Create Visitor with unknown user CDS	Windows ...	Martijn va...
Passed	16	4982	Create Visitor with different user than logged in CDS	Windows ...	Martijn va...
Passed	17	4993	Check concurrent visitor registration	Windows ...	Martijn va...

Appendix

Bijlage 1: Afstudeerplan



Bijlage 1
Afstudeerplan.pdf

Bijlage 2: Plan van Aanpak



Plan van
Aanpak.pdf

Bijlage 3: Manual Knowledge Base Management



Manual Knowledge
Base Management -

Bijlage 4: Deployment Manual



Bijlage 4
Deployment Manual

Bijlage 5: Technical Design



Bijlage 5 Technical
Design - Macaw Ass