

**Auteur**

Abdel Hadi Saad

Datum

30/01/2026

Versie

1.0

© Hogeschool Utrecht,
Utrecht, 8 februari 2026

AI Analytic agent

Plan van aanpak

Toelichting

De tekst op deze pagina en alle geel gearceerde teksten zijn toelichtingen die je helpen bij het schrijven van het voorstel. **Verwijder deze pagina en de gele toelichtingen vóórdat je het voorstel inlevert in OnStage** Mocht je het onprettig vinden om in dit template te werken, dan mag je deze uiteraard loslaten en het Plan van Aanpak (PvA) in je eigen stijl schrijven, mits je wel alle onderdelen behandelt.

Op basis van de afstudeeropdracht die jij en de organisatie met elkaar hebben besproken, ga je een plan van aanpak (PvA) schrijven. In dit PvA beschrijf je uitgebreider je afstudeeropdracht. In het afstudeervoorstel heb je hier al een eerste versie van beschreven. In dit PvA pas je het voorstel aan en breid je het uit aan de hand van de kennis en inzichten die je afgelopen tijd hebt opgedaan.

Het PvA moet voor alle betrokkenen een goed beeld geven van de geplande uitvoering van de opdracht. Dat betekent dat ze te begrijpen en te beoordelen moet zijn door de docentbegeleider, die wel algemene ICT-kennis heeft, maar geen kennis heeft van het bedrijf en de specifieke context van de opdracht. In het PvA geef je aan hoe je je opdracht gestructureerd en methodisch gaat aanpakken.

Zorg ervoor dat alle aangegeven (deel)onderwerpen aan bod komen in het voorstel. Onvolledige voorstellen worden niet beoordeeld.

Verwijder deze pagina met de toelichting, maar:

- Vul de titelpagina in
- Let op correct, verzorgd en professioneel taalgebruik:
- Gebruik geen spreektaal, geen 'ik'
- Gebruik de spellingscontrole (of laat het door iemand nalezen)
- Zorg voor een uniforme opmaak (plak geen tekst uit andere documenten zonder opmaak)
- Werk na afloop de inhoudsopgave bij
- Voeg eventuele relevante extra informatie toe in een bijlage en verwijst op de juiste manier volgens de APA stijl regels
- In geval van afstuderen in het buitenland of een afstudeeropdracht bij een bedrijf in Nederland waar de voertaal Engels is kan het PvA ook in het Engels opgesteld worden
- Het voorstel is kort maar volledig, max 10-15 pagina's (exclusief bijlagen)

Inhoudsopgave

1. Inleiding	4
1.1. Achtergrond	4
1.2. Organisatorische context	4
1.3. Kwestie	4
1.4. Doel.....	4
1.5. Afbakening	5
1.6. Onderzoeksvragen.....	6
1.7. Producten.....	6
1.8. Beschrijving van uit te voeren beroepstaken en professional skills	7
1.9. Randvoorwaarden.....	7
2. Vooronderzoek	8
3. Aanpak	10
3.1. Inleiding.....	Error!
Bookmark not defined.	
3.2. Paragraaf	Error!
Bookmark not defined.	
4. Risico's	15
4.1. Risico's nader toegelicht	15
5. Planning en communicatie	15
5.1. Fasering activiteiten	Error!
Bookmark not defined.	
5.2. Benodigde mensen en middelen opdrachtgever	Error!
Bookmark not defined.	
5.3 Communicatie	Error!
Bookmark not defined.	
Literatuur	16
Bijlage A - <bijlagetitel>	19
A.1 Inleiding	19
A.2 <kopje>.....	19

1. Inleiding

1.1. Achtergrond

De snelle ontwikkeling van AI-technologieën biedt nieuwe mogelijkheden voor het toegankelijk maken en analyseren van data binnen organisaties. Tegelijkertijd brengt het toepassen van AI binnen softwareontwikkeling nieuwe uitdagingen met zich mee, met name op het gebied van betrouwbaarheid, controleerbaarheid en verantwoord gebruik.

Binnen softwarebedrijven wordt steeds vaker gewerkt met databases waarin waardevolle bedrijfsinformatie is opgeslagen. Het benutten van deze data vereist echter technische kennis, waardoor niet-technische gebruikers afhankelijk blijven van ontwikkelaars om inzichten te verkrijgen. Dit belemmert efficiënte besluitvorming en het optimaal benutten van beschikbare data.

Deze afstudeeropdracht richt zich op het ontwerpen en realiseren van een softwaresysteem waarin een AI-agent natuurlijke taal vertaalt naar database-analyses. De nadruk ligt daarbij op de software-engineeringuitdagingen die ontstaan bij het integreren van een niet-deterministische AI-component binnen een betrouwbaar en beheersbaar softwaresysteem.

1.2. Organisatorische context

De afstudeeropdracht wordt uitgevoerd binnen Thrinx, een klein softwarebedrijf dat zich richt op webontwikkeling, maatwerksoftware en datagedreven oplossingen. De organisatie bestaat uit een klein team en voert projecten uit voor diverse opdrachtgevers, waarbij flexibiliteit en technische diepgang centraal staan.

De student voert de opdracht uit binnen zijn eigen onderneming, in de rol van software developer en ontwerper. De werkzaamheden bestaan uit het zelfstandig uitvoeren van onderzoek, het ontwerpen van een software-architectuur en het realiseren van een werkend prototype. De student wordt inhoudelijk begeleid door een externe bedrijfsbegeleider, werkzaam op hbo+ niveau, en door een docentbegeleider vanuit de opleiding.

De opdracht staat los van commerciële klantprojecten en richt zich op het ontwikkelen van een generiek prototype, waarmee de context geschikt is voor een afstudeeropdracht op hbo-niveau Software Development.

1.3. Kwestie

Binnen veel organisaties is een grote hoeveelheid data beschikbaar in databases. In de praktijk blijkt echter dat deze data vaak onvoldoende wordt benut, omdat het analyseren ervan technische kennis vereist, zoals het schrijven van SQL-queries en inzicht in databasestructuren. Niet-technische gebruikers zijn hierdoor afhankelijk van ontwikkelaars of data-specialisten om antwoorden te krijgen op relatief eenvoudige vragen.

Deze situatie leidt tot vertraging in besluitvorming en beperkt het effectief gebruik van data. Tegelijkertijd biedt dit een kans om met behulp van AI een brug te slaan tussen gebruikers en data, door natuurlijke taal te vertalen naar database-analyses. De kern van de kwestie zit niet alleen in het toepassen van AI, maar vooral in het betrouwbaar en verantwoord integreren van een niet-deterministische AI-component binnen een softwaresysteem.

Zonder een gecontroleerde aanpak bestaat het risico op onjuiste analyses, onveilige database-interacties en onbetrouwbare uitkomsten, wat het gebruik van AI in een bedrijfscontext problematisch maakt.

1.4. Doel

Het doel van dit afstudeerproject is het ontwerpen en realiseren van een softwaresysteem waarin een zelfgebouwde AI-agent gebruikers in staat stelt om in natuurlijke taal vragen te stellen over data in een database en op basis daarvan betrouwbare analyses en inzichten te verkrijgen.

Na afronding van het project is een werkend prototype opgeleverd waarin de AI-agent is geïntegreerd binnen een backend-architectuur. Het systeem vertaalt natuurlijke taalvragen naar valide, read-only databasequeries en presenteert de resultaten op een begrijpelijke manier.

Het doel is behaald wanneer het prototype binnen de vastgestelde afstudeerperiode is gerealiseerd en aantoonbaar voldoet aan de vooraf vastgestelde functionele en kwalitatieve eisen. De beoordeling hiervan vindt plaats in overleg met de externe bedrijfsbegeleider en de docentbegeleider.

1.5. Afbakening

Dit afstudeerproject kent een duidelijke afbakening om de haalbaarheid, focus en beoordeelbaarheid te waarborgen.

Organisatorische afbakening

Het project wordt uitgevoerd binnen de context van Thrinx en is gericht op het ontwikkelen van een generiek prototype. De opdracht staat los van lopende of toekomstige commerciële klantprojecten. Externe eindgebruikers of klanten zijn niet direct betrokken bij de uitvoering of validatie van het prototype.

Functionele afbakening

Het systeem ondersteunt het stellen van natuurlijke taalvragen over data en het genereren van analytische resultaten. De functionaliteit is beperkt tot analyse en interpretatie van bestaande data. Het systeem neemt geen autonome beslissingen en voert geen acties uit die directe gevolgen hebben voor bedrijfsprocessen.

Technische afbakening

De AI-agent genereert uitsluitend read-only databasequeries. Schrijfbewerkingen, mutaties of automatische optimalisaties van de database vallen buiten de scope. De focus ligt op backend softwareontwikkeling en architectuur. Uitgebreide frontendontwikkeling, productieklare beveiligingsmaatregelen en schaalbare cloudinfrastructuur maken geen onderdeel uit van dit project.

AI-afbakening

Binnen dit project wordt een eigen AI-agent ontworpen en gerealiseerd. De student is verantwoordelijk voor de architectuur, logica en besturingsmechanismen van deze agent, waaronder het interpreteren van natuurlijke taal, het vertalen naar database-analyses, het toepassen van validatie- en veiligheidscontroles en het afhandelen van fouten en uitzonderingen.

De AI-agent maakt gebruik van een bestaand taalmodel als onderliggende inferentiecomponent. Het trainen, finetunen of aanpassen van het taalmodel zelf valt buiten de scope. De focus van het project ligt op software engineering: het gecontroleerd, uitlegbaar en beheersbaar inzetten van AI binnen een backend-architectuur.

Tijd en projectfasen

Het project wordt uitgevoerd binnen de vastgestelde afstudeerperiode. De scope omvat de fasen analyse, ontwerp, realisatie en validatie van een prototype. Doorontwikkeling naar een

productieomgeving, langdurige gebruikersadoptie en beheer na oplevering zijn geen onderdeel van deze opdracht.

1.6. Onderzoeksvragen

De onderzoeksvragen zijn opgesteld om het ontwerpen en realiseren van het beroepsproduct te onderbouwen en richting te geven. De vragen zijn afgebakend en direct gekoppeld aan de software-engineeringuitdagingen van het project.

Hoofdvraag

Hoe kan een eigen AI-agent softwarematig worden ontworpen en gerealiseerd zodat gebruikers op een gecontroleerde, uitlegbare en veilige manier in natuurlijke taal analyses kunnen uitvoeren op databasegegevens?

Deelvragen

- Welke architectuurprincipes en ontwerpkeuzes zijn geschikt voor het integreren van een niet-deterministische AI-agent binnen een betrouwbare backend-architectuur?
- Hoe kan natuurlijke taal op een gecontroleerde manier worden vertaald naar valide, read-only databasequeries zonder risico op onveilige of onjuiste database-interacties?
- Welke validatie-, beveiligings- en foutafhandelingsmechanismen zijn noodzakelijk om de betrouwbaarheid en uitlegbaarheid van de AI-agent te waarborgen?
- Hoe kan de output van de AI-agent zodanig worden gepresenteerd dat de gegenereerde analyses begrijpelijk en controleerbaar zijn voor gebruikers?
- Op welke wijze kan het gerealiseerde prototype worden gevalideerd om aan te tonen dat het voldoet aan de vastgestelde functionele en kwalitatieve eisen?

1.7. Producten

Binnen dit afstudeerproject worden meerdere beroepsproducten opgeleverd die gezamenlijk aantonen dat de doelstelling is behaald en de onderzoeksvragen zijn beantwoord.

1. Werkend softwareprototype (AI-agent)

Het belangrijkste beroepsproduct is een werkend prototype van een AI-agent die natuurlijke taalvragen vertaalt naar gecontroleerde, read-only database-analyses. Het prototype omvat de backend-architectuur, de agentlogica, validatie- en veiligheidsmechanismen en een eenvoudige interface voor het tonen van resultaten.

Kwaliteitscriteria:

- De AI-agent genereert uitsluitend valide read-only databasequeries;
- Onveilige of ongeldige queries worden gedetecteerd en geblokkeerd;
- De gegenereerde analyses zijn reproduceerbaar en uitlegbaar;
- Het prototype functioneert stabiel bij vooraf gedefinieerde testscenario's.

2. Software-architectuur en ontwerpdocumentatie

Er wordt documentatie opgeleverd waarin de architectuur en ontwerpkeuzes van het systeem zijn vastgelegd. Deze documentatie beschrijft de opbouw van de backend, de positionering van de AI-agent en de wijze waarop controle en validatie zijn ingericht.

Kwaliteitscriteria:

- Architectuurkeuzes zijn onderbouwd op basis van onderzoek en best practices;
- Diagrammen (zoals component- en sequentiediagrammen) zijn consistent en begrijpelijk;

- De documentatie maakt inzichtelijk hoe betrouwbaarheid en beheersbaarheid zijn geborgd.

3. Onderzoeks- en verantwoordingdocumentatie

Er wordt een onderzoeksdocument opgeleverd waarin de beantwoording van de onderzoeksvragen is vastgelegd. Hierin wordt onderbouwd hoe keuzes tot stand zijn gekomen en hoe het beroepsproduct hieruit voortvloeit.

Kwaliteitscriteria:

- Alle onderzoeksvragen zijn expliciet beantwoord;
- Gekozen oplossingen zijn onderbouwd met literatuur en analyse;
- De relatie tussen onderzoek, ontwerp en realisatie is duidelijk aantoonbaar.

4. Test- en validatierapport

Tot slot wordt een test- en validatierapport opgeleverd waarin is vastgelegd hoe het prototype is gevalideerd en in hoeverre het voldoet aan de vastgestelde eisen.

Kwaliteitscriteria:

- Testscenario's zijn herleidbaar naar functionele en kwalitatieve eisen;
- Testresultaten zijn reproduceerbaar en transparant;
- Eventuele beperkingen van het prototype zijn expliciet benoemd.

1.8. Beschrijving van uit te voeren beroepstaken en professional skills

Binnen deze afstudeeropdracht worden beroepstaken uitgevoerd die aansluiten bij de afstudeerrichting Software Development en de architectuurlaag Software. De opdracht omvat minimaal één beroepstaak op niveau 3 en één beroepstaak op niveau 2, conform de eisen van HBO-ICT.

Beroepstaak 1 – Ontwerpen (niveau 2)

Taak

Ontwerpen

Architectuurlaag

Software

Complexiteit van de context

De context van deze beroepstaak bestaat uit het ontwerpen van een software-oplossing binnen een relatief afgebakende en voorspelbare omgeving. Het project richt zich op één organisatie en één duidelijk gedefinieerd probleemgebied. Hoewel AI een rol speelt binnen het systeem, is de scope beperkt tot een prototype, waardoor de context beheersbaar blijft.

Complexiteit van de taak

De taak omvat het opstellen van een software-architectuur en het maken van ontwerpkeuzes voor de positionering van de AI-agent, datastromen en validatiemechanismen. Hierbij worden bekende ontwerpprincipes en methodieken toegepast, passend bij een afstudeeropdracht op niveau 2.

Zelfstandigheid

De student voert deze beroepstaak zelfstandig uit op basis van bestaande richtlijnen en best practices. Ontwerpkeuzes worden besproken met de bedrijfsbegeleider en docentbegeleider, waarbij feedback wordt verwerkt in het ontwerp.

Beheersingsniveau

Op basis van de complexiteit van taak en zelfstandigheid wordt beroepstaak 1 uitgevoerd op niveau 2.

Beroepstaak 2 – Realiseren (niveau 3)

Taak

Realiseren

Architectuurlaag

Software

Complexiteit van de context

De context van deze beroepstaak is complex doordat het realiseren van het systeem plaatsvindt binnen een deels onvoorspelbare technische omgeving. De integratie van een niet-deterministische AI-agent binnen een software-architectuur vereist continue afwegingen rondom betrouwbaarheid, veiligheid en foutafhandeling.

Complexiteit van de taak

De taak omvat het zelfstandig realiseren van een werkend softwareprototype op basis van het ontwerp, inclusief de implementatie van de AI-agentlogica, validatie- en veiligheidsmechanismen en testscenario's. Tijdens de realisatie moeten ontwerpkeuzes waar nodig worden aangepast op basis van bevindingen, wat de taakinhoud complex maakt.

Zelfstandigheid

De student voert de realisatie grotendeels zelfstandig uit en neemt zelf initiatief bij het oplossen van technische problemen. Begeleiding is beperkt tot sparring en evaluatie, passend bij een beroepstaak op niveau 3.

Beheersingsniveau

Op basis van de complexiteit van context en taak wordt beroepstaak 2 uitgevoerd op niveau 3.

Professional skills

Binnen deze afstudeeropdracht worden de volgende professional skills ontwikkeld en toegepast:

Persoonlijk leiderschap: de student plant en bewaakt zelfstandig de voortgang van het project en stuurt bij waar nodig.

Onderzoekend probleem oplossen: de student onderzoekt meerdere oplossingsrichtingen en onderbouwt ontwerp- en implementatiekeuzes.

Doelgericht interacteren: de student stemt regelmatig af met begeleiders en verwerkt feedback doelgericht.

Toekomstgericht organiseren: de student ontwerpt een oplossing die uitbreidbaar is en rekening houdt met toekomstige doorontwikkeling.

1.9. Randvoorwaarden

Voor de succesvolle uitvoering van dit afstudeerproject zijn de onderstaande randvoorwaarden vastgesteld.

Technische randvoorwaarden

- Beschikbaarheid van een ontwikkelomgeving voor backend softwareontwikkeling (bijv. lokale ontwikkelomgeving en versiebeheer);
- Toegang tot een relationele database met testdata voor analyse- en validatiedoeleinden;
- Toegang tot een bestaand taalmodel via een API of SDK als onderliggende inferentiecomponent voor de AI-agent;
- Beschikbaarheid van relevante ontwikkeltools en programmeertalen die nodig zijn voor de realisatie van het prototype.

Organisatorische randvoorwaarden

- Inhoudelijke begeleiding door een externe bedrijfsbegeleider op hbo+ niveau;
- Regelmatige afstemming met een docentbegeleider vanuit de opleiding;
- Voldoende ruimte en tijd om het project zelfstandig uit te voeren binnen de vastgestelde afstudeerperiode.

Persoonlijke randvoorwaarden

- De student is verantwoordelijk voor planning, voortgangsbewaking en documentatie;
- De student heeft de vrijheid om zelfstandig technische keuzes te maken binnen de vastgestelde scope;
- Feedback van begeleiders wordt tijdig verwerkt in ontwerp en realisatie.
- Deze randvoorwaarden vormen de basis voor een gecontroleerde en haalbare uitvoering van het afstudeerproject.

2. Vooronderzoek

Dit hoofdstuk beschrijft het vooronderzoek dat wordt uitgevoerd ter ondersteuning van het ontwerp en de realisatie van het beroepsproduct. Het vooronderzoek heeft als doel inzicht te verkrijgen in de centrale en gerelateerde begrippen die bepalend zijn voor het functioneren van een AI-agent binnen een softwaresysteem. De samenhang tussen deze begrippen vormt de basis voor onderbouwde ontwerp- en implementatiekeuzes. Het betreft een verkennend onderzoek; diepgaande uitwerking en conclusies vinden plaats in het afstudeerverslag.

2.1. AI-agenten binnen softwareontwikkeling

Het centrale begrip binnen dit project is de AI-agent. Binnen moderne softwareontwikkeling worden AI-agenten ingezet om complexe interacties tussen gebruikers en systemen te vereenvoudigen. (Russell & Norvig, 2021) Een AI-agent kan worden beschouwd als een softwarecomponent die input interpreteert en analyses uitvoert binnen vooraf gedefinieerde kaders.

De AI-agent vormt de kern van het systeem, maar kan niet los worden gezien van de omgeving waarin hij opereert. Het functioneren van de agent is direct afhankelijk van de wijze waarop invoer wordt geïnterpreteerd, hoe interactie met databronnen plaatsvindt en hoe controlemechanismen zijn ingericht. In het vooronderzoek wordt daarom gekeken naar bestaande agent-architecturen en orkestratieprincipes die het gecontroleerd inzetten van AI binnen informatiesystemen ondersteunen.

2.2. Natuurlijke taalverwerking en database-interactie

Om de AI-agent bruikbaar te maken voor eindgebruikers is natuurlijke taalverwerking noodzakelijk. De vertaling van natuurlijke taal naar database-analyses vormt een essentiële schakel tussen gebruiker en data. (Jurafsky & Martin, 2023) Deze stap introduceert echter risico's, aangezien onjuiste interpretatie kan leiden tot foutieve of onveilige databasequeries.

De relatie tussen natuurlijke taalverwerking en database-interactie vraagt om aanvullende validatie- en begrenzingsmechanismen. In het vooronderzoek wordt verkend welke benaderingen geschikt zijn om natuurlijke taalvragen te structureren, te controleren en te beperken tot read-only databasequeries. Hierbij wordt aansluiting gezocht bij bestaande oplossingen voor tekst-naar-querytransformatie en query-validatie.

2.3. Software-architectuur en beheersbaarheid

De inzet van een AI-agent en natuurlijke taalverwerking vereist een software-architectuur die deze componenten op een beheersbare manier samenbrengt. Architectuur fungeert hierbij als verbindend concept tussen de AI-agent, de database-interactie en de validatie- en controlelagen.

In het vooronderzoek wordt aandacht besteed aan architectuurprincipes zoals scheiding van verantwoordelijkheden, modulaire opbouw en expliciete validatie- en foutafhandelingslagen. (Bass, Clements, & Kazman, 2012) Deze principes maken het mogelijk om het gedrag van de AI-agent te begrenzen en inzichtelijk te houden, wat essentieel is voor betrouwbaarheid en onderhoudbaarheid van het systeem.

2.4. Validatie en teststrategieën

De voorgaande begrippen komen samen in de validatie van het systeem. Validatie en testen vormen het middel om aan te tonen dat de AI-agent, de natuurlijke taalverwerking en de architectuur gezamenlijk functioneren zoals beoogd. (ISO/IEC, 2011)

In het vooronderzoek wordt verkend welke test- en validatiestrategieën passend zijn bij een systeem met een niet-deterministische component. Hierbij wordt gekeken naar functionele tests, scenario-gebaseerde validatie en het testen van randgevallen. Deze strategieën maken het mogelijk om de betrouwbaarheid en controleerbaarheid van het prototype aantoonbaar te maken.

3. Aanpak

3.1. Inleiding

In dit hoofdstuk wordt beschreven wat er concreet wordt gedaan tijdens het afstudeerproject en op welke wijze de activiteiten bijdragen aan het beantwoorden van de onderzoeksvragen en het realiseren van de beroepsproducten. Voor de theoretische onderbouwing van gemaakte keuzes wordt verwezen naar het vooronderzoek (hoofdstuk 2).

Het project wordt uitgevoerd in vier fasen: analyse en vooronderzoek, ontwerp, realisatie en testen & validatie. In Tabel 3.1 is een overzicht opgenomen van de methoden die per deelvraag worden toegepast, inclusief de beoogde deelproducten en kwaliteitscriteria. In de daaropvolgende paragrafen (3.2 t/m 3.5) worden deze methoden nader toegelicht.

Deelvraag	Methoden	Beoogd resultaat / deelproduct	Kwaliteitscriteria
Architectuur en ontwerp van de AI-agent	Literatuuronderzoek, architectuuranalyse, modelleren	Architectuur- en ontwerpdocumentatie	Consistentie met ontwerp-principes, onderbouwing met literatuur, akkoord begeleider
Vertaling natuurlijke taal naar database-analyses	Analyse bestaande benaderingen, prototyping, validatie	Werkende NL→query vertaal-component	Alleen read-only queries, correcte validatie, reproduceerbare resultaten
Beheersbaarheid en veiligheid van de AI-agent	Ontwerpanalyse, risico-analyse, implementatie van guardrails	Validatie- en beveiligingsmechanismen	Geen onveilige queries, foutafhandeling aantoonbaar effectief
Begrijpelijkheid en controleerbaarheid van output	Scenario-analyse, evaluatie	Presentatie van analyse-uitkomsten	Output is uitlegbaar en consistent
Validatie van het prototype	Testontwerp, scenario-gebaseerd testen	Test- en validatierapport	Testscenario's herleidbaar naar eisen, transparante resultaten

3.2. Fase 1 – Analyse en vooronderzoek

Analyse en vooronderzoek In deze fase wordt het probleemgebied verder geanalyseerd en wordt het vooronderzoek uit hoofdstuk 2 verdiept. De focus ligt op het concretiseren van eisen en randvoorwaarden voor het systeem.

Activiteiten:

- Verfijnen van de onderzoeksvragen op basis van voortschrijdend inzicht;
- Inventariseren van functionele en niet-functionele eisen;
- Bestuderen van relevante literatuur en best practices;
- Analyseren van risico's met betrekking tot AI-gebruik en database-interactie.

Methoden:

- Literatuuronderzoek;
- Analyse van vergelijkbare oplossingsrichtingen;
- Reflectie en afstemming met begeleiders.

Resultaat:

- Overzicht van eisen en randvoorwaarden;
- Aangescherpte uitgangspunten voor ontwerp en realisatie.

3.3. Fase 2 – Ontwerp

In de ontwerpfase wordt een software-architectuur opgesteld die de AI-agent, database-interactie en validatiemechanismen op een beheersbare manier samenbrengt.

Activiteiten:

- Ontwerpen van de globale software-architectuur;
- Uitwerken van componenten en verantwoordelijkheden;
- Vastleggen van datastromen en interacties;
- Ontwerpen van validatie- en foutafhandelingsmechanismen.

Methoden:

- Modelleren met architectuur- en ontwerpschema's;
- Toepassen van bekende software-ontwerpprincipes;
- Review van ontwerpkeuzes met begeleiders.

Resultaat:

- Architectuur- en ontwerpdocumentatie;
- Onderbouwde ontwerpkeuzes als basis voor realisatie.

3.4. Fase 3 – Realisatie

In deze fase wordt het ontworpen systeem gerealiseerd. De focus ligt op het bouwen van een werkend prototype waarin de AI-agent geïntegreerd is binnen de backend-architectuur.

Activiteiten:

- Implementeren van de backend-architectuur;
- Ontwikkelen van de AI-agentlogica;
- Implementeren van validatie- en veiligheidscontroles;
- Uitvoeren van tussentijdse tests.

Methoden:

- Iteratieve softwareontwikkeling;
- Toepassen van programmeerprincipes en best practices;
- Versiebeheer en incrementele oplevering.

Resultaat:

- Werkend softwareprototype;
- Technische implementatie conform het ontwerp.

3.5. Fase 4 – Testen en validatie

In deze fase wordt het prototype getest en gevalideerd om aan te tonen dat het voldoet aan de vastgestelde eisen en kwaliteitscriteria.

Activiteiten:

- Opstellen van testscenario's;
- Uitvoeren van functionele tests;
- Valideren van AI-output aan de hand van scenario's;
- Documenteren van testresultaten en bevindingen.

Methoden:

- Scenario-gebaseerd testen;
- Analyse van testresultaten;
- Evaluatie met begeleiders.

Resultaat:

- Test- en validatierapport;
- Inzicht in sterke punten en beperkingen van het prototype.

4. Risico's

4.1. Risico's nader toegelicht

In dit hoofdstuk worden de belangrijkste risico's binnen het afstudeerproject beschreven, inclusief de bijbehorende beheersmaatregelen. Het doel van deze risicoanalyse is het tijdig identificeren van mogelijke knelpunten die de voortgang, kwaliteit of haalbaarheid van het project kunnen beïnvloeden, en het expliciet maken van maatregelen om deze risico's te beperken.

4.1 Risicoanalyse

Risico	Omschrijving	Kans	Impact	Beheersmaatregelen
Onvoldoende beheersbaarheid van AI-output	De AI-agent genereert output die moeilijk te controleren of te verklaren is, wat de betrouwbaarheid van het systeem aantast.	Middel	Hoog	Ontwerpen van expliciete validatie- en controlelagen; beperken tot read-only database-interacties; scenario-gebaseerde validatie.
Onjuiste of onveilige databasequeries	Natuurlijke taal wordt verkeerd geïnterpreteerd, waardoor ongeldige of potentieel onveilige queries ontstaan.	Middel	Hoog	Implementeren van query-validatie, restricties op querytype en uitgebreide testscenario's.
Scope-uitbreiding tijdens realisatie	Tijdens de realisatie ontstaat de neiging om extra functionaliteiten toe te voegen buiten de afgesproken scope.	Laag	Middel	Strikte afbakening hanteren zoals beschreven in hoofdstuk 1.5; wijzigingen alleen doorvoeren na afstemming met begeleiders.
Tijdsoverschrijding	Onderdelen van het project kosten meer tijd dan vooraf ingeschat.	Middel	Middel	Iteratieve aanpak; prioriteren van kernfunctionaliteit; periodieke voortgangscontrolle.
Afhankelijkheid van externe AI-dienst	Beschikbaarheid of gedrag van het gebruikte taalmodel via API verandert.	Laag	Middel	Ontwerpen van een abstractielaag; testen met verschillende scenario's; documenteren van aannames en beperkingen.
Onvoldoende validatie van resultaten	Testen zijn niet representatief genoeg om conclusies te trekken over betrouwbaarheid.	Laag	Hoog	Opstellen van expliciete test- en validatiecriteria; herleidbaarheid van tests naar eisen.
Beperkte of niet-representatieve testdata	De beschikbare testdata weerspiegelt onvoldoende realistische situaties, waardoor validatie beperkt blijft.	Middel	Middel	Gebruik van meerdere testscenario's en randgevallen; expliciet documenteren van aannames en beperkingen; evaluatie met begeleiders.

De genoemde risico's en beheersmaatregelen worden gedurende het project periodiek geëvalueerd en waar nodig bijgesteld in overleg met de begeleiders.

5. Planning en communicatie

In dit hoofdstuk wordt de planning en communicatiestructuur van het afstudeerproject gedetailleerd beschreven. De planning is opgesteld voor een periode van twintig weken met een gemiddelde tijdsbesteding van twintig uur per week. De uitwerking is bewust concreet en specifiek, zodat inzichtelijk is hoe de verschillende activiteiten bijdragen aan het behalen van de projectdoelstelling en de beroepsproducten.

5.1. Fasering activiteiten

Het afstudeerproject is opgedeeld in vijf samenhangende fasen. Elke fase kent duidelijke activiteiten, tussenresultaten en beslismomenten. De fasering sluit aan bij de aanpak zoals beschreven in hoofdstuk 3 en maakt het mogelijk om de voortgang te monitoren en tijdig bij te sturen.

Fase	Activiteiten	Periode	Resultaat / mijlpalen
Analyse & vooronderzoek	Uitwerken vooronderzoek; concretiseren onderzoeksvragen; vaststellen functionele en niet-functionele eisen; identificeren risico's en randvoorwaarden	Week 1–4	Definitieve eisenlijst; aangescherpte onderzoeksvragen; analyseverslag
Ontwerp	Ontwerpen software-architectuur; uitwerken AI-agentstructuur; definiëren datastromen en validatiemechanismen; review met begeleiders	Week 5–8	Architectuur- en ontwerpdocumentatie; goedgekeurd ontwerp
Realisatie (iteratief)	Implementeren backend-architectuur; ontwikkelen AI-agentlogica; implementeren validatie en beveiligingslagen; tussentijdse tests en refactoring	Week 9–15	Werkend prototype; tussentijdse versies; technische documentatie
Testen & validatie	Opstellen testscenario's; uitvoeren functionele en scenario-gebaseerde tests; analyseren testresultaten; vastleggen beperkingen	Week 16–18	Test- en validatierapport; onderbouwde conclusies
Afronding & oplevering	Verwerken feedback; afronden documentatie; voorbereiden eindpresentatie en inlevermomenten	Week 19–20	Definitieve oplevering beroepsproducten

Door de realisatiefase iteratief in te richten, is het mogelijk om tijdens de ontwikkeling nieuwe inzichten te verwerken zonder de globale planning te overschrijden.

5.2. Benodigde mensen en middelen opdrachtgever

Voor de uitvoering van het afstudeerproject zijn specifieke betrokkenen en middelen noodzakelijk. Dit onderdeel beschrijft uitsluitend wie betrokken zijn en welke middelen benodigd zijn.

Betrokkenen:

- Externe bedrijfsbegeleider: inhoudelijke begeleiding bij software-architectuur en realisatiekeuzes.
- Docentbegeleider: begeleiding op methodologisch en academisch niveau.
- Student: uitvoering van het project, planning en documentatie.

Middelen:

- Laptop en lokale ontwikkelomgeving;
- Versiebeheersysteem;
- Toegang tot een relationele database
- Toegang tot een taalmodel via API.
- Deze middelen zijn gedurende de volledige looptijd van het project beschikbaar.

5.3. Communicatie en voortgangsbewaking

Dit onderdeel beschrijft uitsluitend hoe de communicatie en voortgangsbewaking plaatsvinden.

- Wekelijkse of tweewekelijkse voortgangsafstemming met de externe bedrijfsbegeleider;
- Periodieke afstemming met de docentbegeleider conform opleidingsrichtlijnen
- Vastlegging van voortgang, beslissingen en feedback in projectdocumentatie;
- Evaluatiemomenten na afronding van iedere projectfase;
- Tijdige signalering en bespreking van risico's en knelpunten.

6. Literatuur

Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice*. Addison-Wesley.

ISO/IEC. (2011). *ISO/IEC 25010: Systems and software Quality Requirements and Evaluation*.

Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson.

Jurafsky, D., & Martin, J. (2023). *Speech and Language Processing*. Pearson.

Bijlage A - <bijlagetitel>

Gebruik voor bijlagetitels de stijl **Kop 1 Bijlage** en **Kop 2 Bijlage**

A.1 Inleiding

Geef in een paar zinnen een toelichting op de inhoud van de bijlage.

A.2 <kopje>