

Eigenaar	Joyren Terken
Examinator 1	#

BEWIJZEN VOOR KERNTAKEN EN WERKPROCESSEN

Datum	NVT
Auteur	Joyren Terken
Versie	1.0.0

INHOUDSOPGAVE

Inleiding

Kerntaak B1-K1: Realiseert software

B1-K1-W1: Plant werkzaamheden en bewaakt de voortgang

- Risicoanalyse
- Probleem- en oplossingsstrategie

B1-K1-W2: Ontwerpt software

- Ontwerp rapport (Vehicle Classes)
- Project Requirements Specificatie (Vereisten en ontwerp)

B1-K1-W3: Realiseert (onderdelen van) software

- Ontwerp rapport (Vehicle.py code)
- Portfolio verslag (Autonomous Vehicle Project)

B1-K1-W4: Test software

- Project Requirements Specificatie (Simulatie- en testvereisten)
- Portfolio verslag (Testen in simulatie)

B1-K1-W5: Doet verbetervoorstellen voor de software

- Probleem- en oplossingsstrategie (Verbeteringsvoorstellen)
- Risicoanalyse (Probleemoplossing)

Kerntaak B1-K2: Werkt in een ontwikkelteam

B1-K2-W1: Voert overleg

- Project Requirements Specificatie (Stakeholders interviews)
- Probleem- en oplossingsstrategie (Overleg met stakeholders)

B1-K2-W2: Presenteert het opgeleverde werk

- Portfolio verslag (Autonomous Vehicle presentatie)
- Risicoanalyse (Projectpresentatie)

B1-K2-W3: Reflecteert op het werk.

- Portfolio verslag (Conclusie en reflectie Autonomous Vehicle)
- Probleem- en oplossingsstrategie (Reflectie en evaluatie)

Aanvullend materiaal

Flappy bird AI

Snake AI

VERSIEBEHEER

1.0.0	Inhoud van het document
1.1.0	Styling van het document.

INLEIDING

Dit document bevat een overzicht van de bewijzen die mijn vaardigheden en competenties ondersteunen met betrekking tot de kerntaken en werkprocessen binnen de softwareontwikkeling. Elk bewijs is zorgvuldig geselecteerd en gekoppeld aan de vereiste werkprocessen om aan te tonen dat ik de bijbehorende competenties beheers.

De focus ligt op mijn bijdragen in het realiseren van software, het werken in een ontwikkelteam, en het uitvoeren van bijbehorende processen zoals planning, ontwerp, testen, en reflectie. Elk onderdeel wordt ondersteund met relevante documentatie en voorbeelden van mijn werk.

Dit document dient als ondersteuning van mijn uiteindelijke portfolio en biedt inzicht in de specifieke onderdelen waarin ik mijn vaardigheden heb bewezen.

BRONNEN

Om dit document overzichtelijk te houden, is ervoor gekozen om de volledige projectdocumentatie, waaronder de Risicoanalyse, Probleem- en oplossingsstrategie, Ontwerprapport en andere relevante bronnen, beschikbaar te stellen in een apart bestand. Dit bestand bevat alle gedetailleerde informatie en bewijzen die de besproken punten ondersteunen. Je kunt het bestand met alle bronnen en ondersteunende documentatie downloaden via de volgende link: [\[Download alle bronnen hier\]](#).

KERNTAAK B1-K1: REALISEERT SOFTWARE

B1-K1-W1: Plant werkzaamheden en bewaakt de voortgang

Bij het ontwikkelen van het autonome voertuig was het essentieel om de werkzaamheden zorgvuldig te plannen en de voortgang regelmatig te bewaken. Dit proces begon met het opstellen van een duidelijke projectstructuur waarin alle benodigde taken en deadlines werden geïdentificeerd. Om ervoor te zorgen dat de werkzaamheden op schema bleven, heb ik gebruik gemaakt van risicoanalyse en voortgangsrapporten om mogelijke vertragingen vroegtijdig te signaleren en te beheersen.

1. Planning van het project

In de Risicoanalyse (zie document "Risico analyse", pag. 3-7) (**Risico analyse**), werd een gedetailleerd overzicht gegeven van de belangrijkste risico's die het project zouden kunnen beïnvloeden. Hieruit kwam naar voren dat het ontbreken van een voertuig en bijbehorende hardware een groot risico vormde, waardoor deze taak als prioriteit werd aangemerkt. Door het opsplitsen van dit risico in kleinere deelproblemen, zoals het bestellen en bouwen van de benodigde hardware, werd een duidelijke planning opgezet.

Daarnaast heb ik in de Probleem- en oplossingsstrategie (pag. 5-9) (**Probleem en oplossings...**) de problemen verder gedetailleerd en oplossingen voorgesteld voor de belangrijkste knelpunten, zoals de software-integratie en simulatieomgeving. Deze analyse vormde de basis voor de fasering van het project, waarbij elke iteratie gericht was op het oplossen van specifieke deelproblemen, zoals het optimaliseren van de reinforcement learning agent of het verbeteren van de gebruikersinterface.

2. Bewaken van de voortgang

Gedurende het project hield ik de voortgang nauwlettend in de gaten, met behulp van iteratieve feedbackmomenten en regelmatige evaluaties. In de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**) heb ik specifieke doelen en mijlpalen vastgesteld om de voortgang meetbaar te maken. Elke iteratie werd geëvalueerd om te bepalen of de geplande taken waren voltooid en om eventuele aanpassingen in de planning door te voeren indien nodig.

Deze aanpak zorgde ervoor dat ik tijdig kon inspelen op problemen, zoals het ontbreken van bepaalde hardware (besproken in de Risicoanalyse(**Risico analyse1**)), en dat het project op koers bleef, ondanks de verschillende uitdagingen. Daarnaast was het belangrijk om niet alleen te focussen op technische oplossingen, maar ook op het tijdig en correct documenteren van de werkzaamheden.

3. Conclusie

Door de taken in duidelijke fasen te splitsen en de voortgang regelmatig te monitoren, kon ik het project effectief managen. De bewijzen in de Risicoanalyse en Probleem- en oplossingsstrategie tonen aan dat ik in staat ben om werkzaamheden te plannen en voortgang te bewaken, terwijl ik flexibel ben om aanpassingen door te voeren wanneer dat nodig is.

B1-K1-W2: Ontwerpt software

Het ontwerpen van software is een cruciaal onderdeel van het ontwikkelproces, waarbij de functionele en technische vereisten worden omgezet in een concrete softwareoplossing. Tijdens het project rond het autonome voertuig speelde het ontwerp een fundamentele rol in het waarborgen van de werking van zowel de hardware- als de softwarecomponenten.

1. Vereistenanalyse en ontwerp:

In de Project Requirements Specificatie([project requirements spe...](#)) heb ik samen met de stakeholders Finn Alberts en Shannen Dolls een lijst van vereisten opgesteld op basis van interviews. Deze vereisten werden vervolgens geprioriteerd volgens de MoSCoW-methode (Must have, Should have, Could have, Won't have), wat de basis vormde voor het ontwerp van het systeem. Hierbij ging het om belangrijke functionele vereisten, zoals het vermogen van het voertuig om obstakels te vermijden en in een rechte lijn te rijden, en om niet-functionele vereisten, zoals de betrouwbaarheid en snelheid van het systeem.

Door deze eisen te analyseren en te vertalen naar een concrete softwarespecificatie, was het mogelijk om een robuust ontwerp te maken dat voldoet aan de verwachtingen van de stakeholders. Dit proces omvatte het identificeren van kritieke functies, zoals motorbesturing, afstandsmeting en voertuignavigatie.

2. Ontwerp van softwarecomponenten:

Het eigenlijke softwareontwerp werd verder uitgewerkt in het Ontwerp rapport([ontwerp rapport1](#)), waarin de verschillende klassen en hun verantwoordelijkheden zijn beschreven. Het systeem was modulair ontworpen, met aparte klassen voor de motorbesturing, de afstandssensoren, en het voertuig zelf. Deze scheiding van verantwoordelijkheden zorgde ervoor dat elke component gemakkelijk onderhouden en uitgebreid kon worden.

Belangrijke componenten van het ontwerp zijn onder andere:

- Motor Class: Deze klasse beheert de beweging van het voertuig en bevat methoden voor het vooruit- en achteruitrijden, evenals stoppen.
- DistanceSensor Class: Deze klasse meet de afstand van objecten voor het voertuig en stuurt deze gegevens door naar het systeem om te bepalen of er moet worden bijgestuurd.
- Vehicle Class: Dit is de overkoepelende klasse die de verschillende componenten samenbrengt en ervoor zorgt dat het voertuig autonoom kan navigeren door middel van reinforcement learning.

De gekozen architectuur maakt het eenvoudig om nieuwe sensoren of actuatoren toe te voegen, waardoor het systeem flexibel is voor toekomstige uitbreidingen.

3. Visuele documentatie

Naast de broncode zijn in het Ontwerp rapport ook visuele hulpmiddelen opgenomen om het ontwerp verder te verduidelijken. Class-diagrammen en componentendiagrammen tonen de relaties tussen de verschillende onderdelen van het systeem en hoe deze met elkaar samenwerken. Dit biedt niet alleen een helder overzicht van de systeemarchitectuur, maar maakt het ook gemakkelijker voor andere ontwikkelaars om het ontwerp te begrijpen en aanpassingen door te voeren wanneer dat nodig is.

4. Iteratief ontwerp

Tijdens het ontwerpen heb ik gekozen voor een iteratieve aanpak. Dit betekent dat het ontwerp in verschillende fasen werd ontwikkeld en bijgesteld op basis van feedback van de stakeholders en testresultaten. Door het gebruik van simulaties, zoals beschreven in de Probleem- en oplossingsstrategie(**Probleem en oplossings...**), konden we het ontwerp valideren en aanpassingen doorvoeren waar nodig. Deze aanpak zorgde ervoor dat het ontwerp flexibel bleef en eenvoudig kon worden aangepast om te voldoen aan de steeds veranderende projectvereisten.

Conclusie

Het softwareontwerp voor het autonome voertuig werd zorgvuldig uitgewerkt op basis van een gedetailleerde vereistenanalyse, modulaire architectuur en visuele documentatie. Dit proces resulteerde in een flexibele en uitbreidbare oplossing, waarbij zowel de functionele als niet-functionele vereisten werden gerespecteerd. Het bewijs voor mijn bijdrage aan het ontwerp is te vinden in de Project Requirements Specificatie(**PROJECT REQUIREMENTS SP...**) en het Ontwerp rapport(**ontwerp rapport**), waaruit blijkt dat ik in staat ben om complexe softwareoplossingen te ontwerpen die voldoen aan de behoeften van zowel gebruikers als stakeholders.

B1-K1-W3: Realiseert (onderdelen van) software

Het realiseren van software omvat de daadwerkelijke ontwikkeling van de verschillende onderdelen van het systeem. Voor het autonome voertuigproject heb ik meerdere softwarecomponenten geïmplementeerd die de kernfunctionaliteit van het systeem ondersteunen, waaronder de besturing van het voertuig, de verwerking van sensordata, en de integratie van reinforcement learning algoritmes.

1. Implementatie van de voertuigbesturing

In het Ontwerp rapport(ontwerp rapport) wordt de structuur van de verschillende klassen beschreven die zijn geïmplementeerd om het autonome voertuig te besturen. Deze klassen, zoals de Motor Class en de Vehicle Class, werden gerealiseerd in Python en maken gebruik van hardware zoals motorcontrollers en sensoren.

De Motor Class biedt functies voor het bewegen van het voertuig in verschillende richtingen. Ik heb methoden geïmplementeerd zoals forward(speed) en backward(speed), waarmee de snelheid en richting van het voertuig kunnen worden gecontroleerd. Deze functies vormen de basis voor de besturing van het voertuig tijdens het autonoom navigeren.

De Vehicle Class integreert de motoren en sensoren om een volledig functioneel voertuig te creëren dat op basis van input van sensoren zelfstandig kan navigeren. De voertuigklasse maakt gebruik van meerdere afstandssensoren (voor, links, rechts), die allemaal zijn geïmplementeerd in de DistanceSensor Class. Deze sensoren meten de afstand tot obstakels, wat essentieel is voor het vermijden van botsingen.

2. Integratie van sensoren en het verwerken van input

Een van de belangrijkste uitdagingen was het correct verwerken van input van de sensoren en het gebruik hiervan voor besluitvorming door de reinforcement learning agent. In de Vehicle Class (ontwerp rapport) worden de gegevens van de sensoren continu gelezen en gebruikt om te bepalen of het voertuig moet stoppen, draaien of doorgaan.

De implementatie van de DistanceSensor Class maakte het mogelijk om nauwkeurige afstanden te meten met behulp van ultrasone sensoren. Deze klasse stuurt een puls uit en meet de tijd totdat het signaal terugkeert, wat wordt omgezet in een afstandsmeting. De gegevens uit deze metingen worden vervolgens doorgegeven aan de beslissingslogica van de agent, die in staat is om op basis van deze informatie beslissingen te nemen.

3. Gebruik van reinforcement learning algoritmes

Naast de basisbesturing en sensordata-verwerking, heb ik reinforcement learning algoritmes geïntegreerd om het voertuig autonoom te laten leren navigeren. In de Probleem- en oplossingsstrategie([Probleem en oplossingss...](#)) wordt beschreven hoe verschillende algoritmes, zoals Deep Q-Networks (DQN) en Proximal Policy Optimization (PPO), werden onderzocht en getest om het voertuig in staat te stellen zijn omgeving te begrijpen en zich hierop aan te passen. Het trainen van de agent gebeurde in een simulatieomgeving waarin de voertuigbesturing en sensorinput werden nagebootst. Deze simulatie werd vervolgens gebruikt om het voertuig in de echte wereld te testen. Dit iteratieve proces zorgde ervoor dat de software geleidelijk aan werd verbeterd en het voertuig steeds beter in staat was om obstakels te vermijden en taken uit te voeren zoals het navigeren door een parcours.

4. Gebruik van versiebeheer en samenwerking

Tijdens de ontwikkeling van de software maakte ik gebruik van versiebeheersystemen zoals Git om de voortgang van het project te beheren. De commitgeschiedenis toont aan dat ik actief bezig was met het implementeren van nieuwe functies, het oplossen van bugs en het optimaliseren van de prestaties van de software. Dit systeem maakte het eenvoudig om samen te werken en de software continu te verbeteren zonder dat eerdere versies verloren gingen.

Conclusie

De realisatie van de software voor het autonome voertuigproject omvatte de ontwikkeling van meerdere essentiële componenten, zoals motorbesturing, sensordata-verwerking en reinforcement learning integratie. Deze onderdelen zijn geïmplementeerd op een modulaire manier, waardoor het systeem eenvoudig aanpasbaar en uitbreidbaar is. Het bewijs voor mijn vaardigheden in het realiseren van softwarecomponenten is te vinden in de broncode in het Ontwerp rapport([ontwerp rapport](#)) en de beschrijving van het gebruik van reinforcement learning algoritmes in de Probleem- en oplossingsstrategie([Probleem en oplossingss...](#)).

B1-K1-W4: Test software

Het testen van software is een cruciaal onderdeel van het ontwikkelproces om ervoor te zorgen dat alle functionaliteiten correct werken en de software voldoet aan de gestelde eisen. Tijdens het project rond het autonome voertuig heb ik een grondige teststrategie ontwikkeld en uitgevoerd om zowel de hardware- als softwarecomponenten te valideren, inclusief het gedrag van het voertuig binnen de simulatie en in de echte wereld.

1. Teststrategie en plannen

In de Project Requirements Specificatie([project requirements spe...](#)) en Probleem- en oplossingsstrategie([Probleem en oplossingss...](#)) heb ik de vereisten vastgelegd die het voertuig in verschillende scenario's moet kunnen vervullen. Deze vereisten vormden de basis voor het opstellen van testcases die specifiek waren gericht op het valideren van de belangrijkste functies van het voertuig. Zo werden tests ontwikkeld voor het testen van de navigatie van het voertuig, het vermijden van obstakels en de correcte uitvoering van instructies zoals stoppen en draaien. De teststrategie bestond uit verschillende fases:

- Unit tests: Hierbij werden afzonderlijke softwarecomponenten, zoals de Motor Class en DistanceSensor Class, individueel getest om te verifiëren dat ze correct functioneren.
- Integratietests: Deze tests werden uitgevoerd om te controleren of de verschillende softwarecomponenten correct samenwerken. Hierbij werd vooral gekeken naar de interactie tussen de motorbesturing en de afstandssensoren.
- Simulatietests: De simulatieomgeving speelde een cruciale rol bij het testen van de software voordat deze werd ingezet in de echte wereld. In de simulatie werden verschillende obstakels en parcoursen nagebootst om het gedrag van het voertuig te testen in een gecontroleerde omgeving.

2. Automatische simulatie en testen

Een belangrijk onderdeel van het testen was de implementatie van een gesimuleerde omgeving waarin het voertuig kon worden getraind en getest. In de Probleem- en oplossingsstrategie ([Probleem en oplossingss...](#)) beschrijf ik hoe de simulatieomgeving werd gebruikt om realistische scenario's te creëren waarin het voertuig autonoom kon leren en presteren. Deze omgeving was essentieel om de prestaties van het reinforcement learning model te valideren.

Tijdens de simulatie werden verschillende parameters gemeten, zoals:

- **Botsingen met obstakels:** Dit was een kritieke meting om te beoordelen of het voertuig effectief obstakels wist te vermijden.
- **Aantal correcte manoeuvres:** Dit werd gemeten door het voertuig in een parkeerscenario te plaatsen en te evalueren of het succesvol kon parkeren binnen de aangegeven ruimte.
- **Navigatietijd:** De tijd die het voertuig nodig had om een parcours af te leggen werd gemeten om te bepalen hoe efficiënt het leerproces was.

Deze tests hielpen bij het identificeren van mogelijke bugs en inefficiënties in het systeem. Door de resultaten van de simulaties te analyseren, kon ik aanpassingen doen aan de beloningsstructuur van de reinforcement learning agent om het gedrag van het voertuig verder te optimaliseren.

3. Testen van de hardware-integratie

Naast het testen in de simulatieomgeving was het ook belangrijk om de software in de echte wereld te testen op het daadwerkelijke voertuig. In de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**) en Ontwerp rapport(**ontwerp rapport**) wordt beschreven hoe de hardwarecomponenten, zoals de motoren en sensoren, werden geïntegreerd en getest.

Deze tests omvatten:

- **Motoren testen:** Door de motorbesturing te testen met verschillende snelheden en richtingen, werd geverifieerd dat het voertuig correct kon rijden en stoppen.
- **Sensorvalidatie:** De afstandssensoren werden getest om ervoor te zorgen dat ze nauwkeurige metingen gaven en dat de software correct reageerde op deze input door het voertuig te laten stoppen of van richting te laten veranderen.

4. Analyseren van testresultaten

De testresultaten werden geanalyseerd om te bepalen in hoeverre het voertuig voldeed aan de gestelde vereisten. Dit proces werd iteratief uitgevoerd; op basis van de testuitkomsten werden verbeteringen doorgevoerd en opnieuw getest. Dit iteratieve testproces werd herhaald totdat het voertuig alle tests succesvol doorstond.

In de Project Requirements Specificatie([project requirements sp...](#)) worden verschillende vereisten beschreven, zoals het succesvol vermijden van obstakels en het rijden in een rechte lijn. Deze vereisten werden direct getoetst tijdens de simulatie en real-world tests. Door continue evaluatie en verbetering werd ervoor gezorgd dat het systeem betrouwbaar en robuust was.

5. Conclusie

Het testen van de software voor het autonome voertuigproject was een rigoureuus proces waarin zowel simulaties als hardwaretests werden gebruikt om de functionaliteit van het systeem te valideren. Deze tests waren essentieel om ervoor te zorgen dat het voertuig correct kon navigeren, obstakels kon vermijden en instructies nauwkeurig kon uitvoeren. De simulatieomgeving maakte het mogelijk om het voertuig intensief te testen voordat het in de echte wereld werd ingezet, wat de efficiëntie en betrouwbaarheid van het systeem aanzienlijk verbeterde. Het bewijs van deze tests is te vinden in de Project Requirements Specificatie([project requirements sp...](#)) en de Probleem- en oplossingsstrategie([Probleem en oplossings...](#)).

B1-K1-W5: Doet verbetervoorstellen voor de software

Tijdens de ontwikkeling van software is het essentieel om continu verbeteringen te identificeren en door te voeren om de kwaliteit en functionaliteit van het systeem te optimaliseren. In het autonome voertuigproject heb ik verschillende verbetervoorstellen gedaan en geïmplementeerd om zowel de prestaties van het voertuig als de efficiëntie van de software te verhogen. Deze voorstellen waren gebaseerd op bevindingen uit testresultaten, simulaties, en feedback van stakeholders.

1. Analyse van problemen en verbeteringsvoorstellen

In de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**) heb ik een gedetailleerde analyse uitgevoerd van de belangrijkste problemen binnen het project. Deze analyse identificeerde meerdere deelproblemen, waaronder de optimalisatie van de reinforcement learning agent, de betrouwbaarheid van de simulatieomgeving en de integratie van softwaremodules. Voor elk van deze deelproblemen werden verbeteringsvoorstellen ontwikkeld

en besproken met de stakeholders.

Een belangrijk verbeterpunt betrof de beloningsstructuur van de reinforcement learning agent. De oorspronkelijke beloningsstructuur leidde tot inefficiënt leerproces, waarbij het voertuig soms ongewenste acties ondernam. Na het analyseren van de testresultaten in de simulatieomgeving, stelde ik voor om de beloningsstructuur aan te passen, zodat het voertuig consistentere beloond werd voor correcte manoeuvres, zoals obstakels vermijden en efficiënt parkeren.

2. Verbetering van de beloningsstructuur

De beloningsstructuur werd aangepast door het aantal positieve beloningen voor succesvolle navigatie te verhogen en de strafpunten voor ongewenste acties, zoals botsingen, te verzwaren. Deze wijzigingen werden geïmplementeerd en getest in de simulatieomgeving, zoals beschreven in de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**). Deze aanpassing zorgde ervoor dat het voertuig sneller leerde om correct te navigeren en zijn prestaties te verbeteren zonder dat het onnodig veel tijd doorbracht in foutieve patronen.

3. Optimalisatie van de software-integratie

Een ander belangrijk verbeterpunt was de integratie van de verschillende softwarecomponenten. In het oorspronkelijke systeem waren er compatibiliteitsproblemen tussen de motorbesturing en de sensordata-verwerking. Deze problemen werden geïdentificeerd tijdens de integratietests, zoals beschreven in de Ontwerp rapport(ontwerp rapport). Mijn voorstel was om de interfaces tussen de verschillende componenten te standaardiseren, zodat de communicatie tussen de motorbesturing en de sensoren efficiënter verliep. Door deze interfaces opnieuw te definiëren en te implementeren, werden de compatibiliteitsproblemen opgelost en functioneerde het systeem soepeler. Daarnaast stelde ik voor om het communicatieprotocol tussen de motoren en sensoren te optimaliseren, zodat er minder vertraging optrad bij het verwerken van sensordata. Dit verbeterde de real-time reactiesnelheid van het voertuig aanzienlijk.

4. Verbeteren van de gebruikersinterface

Een ander belangrijk aspect van het project was de gebruikersinterface waarmee de voortgang van het voertuig en de training van de reinforcement learning agent werd gemonitord. In de oorspronkelijke versie van de interface was de informatie over het trainingsproces beperkt en moeilijk te interpreteren. In de Probleem- en oplossingsstrategie(Probleem en oplossings...) heb ik voorgesteld om meer gedetailleerde datavisualisatie toe te voegen, zodat gebruikers de voortgang van het voertuig beter konden volgen.

Na de implementatie van deze verbeteringen konden gebruikers real-time gegevens zien over de prestaties van het voertuig, zoals het aantal botsingen, de gemiddelde afstand tot obstakels en de tijd die het voertuig nodig had om een parcours te voltooien. Deze verbeteringen maakten het voor de gebruikers eenvoudiger om de voortgang te beoordelen en eventuele problemen sneller op te sporen.

5. Resultaten van de verbeteringen

De doorgevoerde verbeteringen hadden een significant positief effect op de prestaties van het systeem. Door de beloningsstructuur aan te passen, leerde de reinforcement learning agent sneller en efficiënter. De optimalisatie van de software-integratie en het communicatieprotocol tussen de motoren en sensoren zorgde ervoor dat het voertuig soepeler bewoog en sneller reageerde op obstakels. De verbeteringen aan de gebruikersinterface maakten het voor zowel de stakeholders als het ontwikkelteam gemakkelijker om het systeem te monitoren en problemen te identificeren.

6. Conclusie

Het doorvoeren van verbeteringen aan de software was een iteratief proces waarbij continu werd gekeken naar testresultaten en feedback van gebruikers om knelpunten te identificeren. De verbeteringsvoorstellen die ik heb gedaan, zoals de aanpassing van de beloningsstructuur, de optimalisatie van softwarecomponenten en de verbetering van de gebruikersinterface, hebben geleid tot een beter presterend systeem. Deze verbeteringen worden gedetailleerd beschreven in de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**) en de Ontwerp rapport(**ontwerp rapport**).

Afsluiting B1-K1: Realiseert software

Het realiseren van software binnen het autonome voertuigproject was een uitgebreid proces waarin alle fasen van softwareontwikkeling aan bod kwamen, van het plannen en ontwerpen tot het implementeren, testen en verbeteren van de verschillende onderdelen. Door een gestructureerde aanpak en een continue focus op kwaliteit en verbetering, werd ervoor gezorgd dat de software op tijd en volgens de vereisten werd opgeleverd.

Het plannen van werkzaamheden en bewaken van de voortgang (B1-K1-W1) zorgde ervoor dat het project beheersbaar bleef en dat knelpunten tijdig werden geïdentificeerd en aangepakt. In het ontwerpproces (B1-K1-W2) werd zorgvuldig nagedacht over de structuur en architectuur van de software, wat leidde tot een flexibele en modulaire oplossing. De implementatie van de softwarecomponenten (B1-K1-W3) resulteerde in een werkend autonoom voertuig, dat succesvol werd getest en geoptimaliseerd (B1-K1-W4). Tot slot werden er door middel van reflectie en evaluatie (B1-K1-W5) waardevolle verbeteringen voorgesteld en doorgevoerd, wat heeft geleid tot een efficiënter en robuuster systeem.

Al deze stappen tonen aan dat ik in staat ben om het volledige proces van softwareontwikkeling te beheersen en dat ik in staat ben om complexe projecten te realiseren binnen de gestelde tijdslijnen en kwaliteitsnormen.

KERNTAAK B1-K2: WERKT IN EEN ONTWIKKELTEAM

B1-K2-W1: Voert overleg

Overleg voeren met stakeholders en teamleden is een cruciaal onderdeel van het ontwikkelproces. Goede communicatie zorgt ervoor dat de verwachtingen duidelijk zijn, de voortgang transparant blijft en eventuele problemen snel kunnen worden opgelost. Tijdens het autonome voertuigproject speelde overleg een essentiële rol in het definiëren van de projectvereisten, het oplossen van problemen en het verbeteren van de samenwerking binnen het team.

1. Overleg met stakeholders

Een van de belangrijkste overlegmomenten vond plaats in de vorm van een stakeholdersinterview, zoals beschreven in de Project Requirements Specificatie(**project requirements sp...**). Tijdens dit overleg werden de projectdoelen en functionele eisen besproken met de stakeholders, Finn Alberts en Shannen Dolls. Het doel van dit overleg was om een duidelijk beeld te krijgen van de verwachtingen en prioriteiten van het project, zoals het vermogen van het voertuig om autonoom obstakels te vermijden en in een rechte lijn te rijden.

Deze gesprekken vormden de basis voor de verzamelde vereisten, die volgens de MoSCoW-methode werden geprioriteerd. Door deze aanpak werd een heldere prioritering van de taken vastgelegd, met een focus op de "Must have" functionaliteiten, zoals de obstakelvermijding en voertuigbesturing.

2. Teamoverleg en samenwerking

Naast het overleg met de stakeholders was er voortdurend overleg binnen het ontwikkelteam. Gedurende het project werden er teamvergaderingen gehouden om de voortgang te bespreken en knelpunten te identificeren. Hierbij werd regelmatig gekeken naar de status van de verschillende softwarecomponenten, zoals de motorbesturing en sensoren, en naar de resultaten van de tests in de simulatieomgeving.

In de Probleem- en oplossingsstrategie(**Probleem en oplossings...**) wordt besproken hoe deze overleggen hebben bijgedragen aan het identificeren van problemen, zoals de noodzaak om de beloningsstructuur van de reinforcement learning agent aan te passen. Door deze regelmatige feedbackmomenten konden de problemen snel worden opgelost en werd de efficiëntie van het team vergroot.

3. Documentatie van overleg

Na elk overleg werden notulen en verslagen bijgehouden om de voortgang te documenteren en om ervoor te zorgen dat iedereen op de hoogte bleef van de afgesproken acties en wijzigingen. In de Project Requirements Specificatie(**project requirements sp...**) en Probleem- en oplossingsstrategie(**Probleem en oplossings...**) zijn de resultaten van deze overleggen gedetailleerd vastgelegd, inclusief de beslissingen die werden genomen over de prioritering van functies en de implementatie van verbeteringen.

4. Conclusie

Door regelmatig overleg te voeren met zowel de stakeholders als het ontwikkelteam, was het mogelijk om het project in goede banen te leiden en ervoor te zorgen dat de doelen en verwachtingen helder waren. Deze overleggen zorgden ervoor dat de ontwikkeling van het autonome voertuigproject soepel verliep en dat eventuele knelpunten tijdig werden opgelost. De overlegmomenten en de resultaten hiervan zijn gedocumenteerd in de Project Requirements Specificatie(**project requirements sp...**) en de Probleem- en oplossingsstrategie(**Probleem en oplossings...**), die aantonen dat ik in staat ben om effectief overleg te voeren binnen een projectomgeving.

B1-K2-W2: Presenteert het opgeleverde werk

Het presenteren van opgeleverd werk is een cruciaal onderdeel van elk softwareontwikkelingsproject. Het zorgt ervoor dat stakeholders en teamleden inzicht krijgen in de voortgang, uitdagingen en resultaten van het project. Tijdens het autonome voertuigproject heb ik regelmatig presentaties gegeven om de belangrijkste bevindingen, voortgang en resultaten te delen, zodat er effectieve beslissingen konden worden genomen en het project op koers bleef.

1. Presentatie van de projectvereisten en voortgang

Een van de eerste presentaties vond plaats na het verzamelen van de projectvereisten. Zoals gedocumenteerd in de Project Requirements Specificatie([project requirements sp...](#)), werden de verzamelde eisen gepresenteerd aan de stakeholders, waarbij de belangrijkste functionaliteiten en prioriteiten werden toegelicht. Deze presentatie was bedoeld om ervoor te zorgen dat alle belanghebbenden op dezelfde lijn zaten met betrekking tot de projectdoelen en om goedkeuring te krijgen voor de vereisten en de prioritering volgens de MoSCoW-methode.

Tijdens deze presentatie heb ik uitgelegd hoe de functionele vereisten, zoals het autonoom kunnen rijden en obstakels vermijden, zouden worden geïmplementeerd. De feedback van de stakeholders werd verwerkt in de definitieve versie van de specificatie, waardoor we konden doorgaan naar de volgende fase van het project.

2. Visuele presentatie van testresultaten

Gedurende het project waren er verschillende momenten waarop testresultaten werden gepresenteerd. In de Probleem- en oplossingsstrategie([Probleem en oplossings...](#)) wordt beschreven hoe de testresultaten van de simulatieomgeving en het fysieke voertuig werden geanalyseerd. Deze resultaten, zoals het aantal botsingen, de navigatietijd en de leerprestaties van de reinforcement learning agent, werden gepresenteerd aan het team en de stakeholders. De presentatie van de testresultaten was visueel ondersteund met grafieken en simulaties die de voortgang van het voertuig weergaven. Dit hielp de stakeholders en het team om de verbeteringen en uitdagingen duidelijk te begrijpen. Bovendien werden de resultaten vergeleken met de vooraf gedefinieerde vereisten, waardoor de effectiviteit van de geïmplementeerde functies goed zichtbaar werd.

3. Demonstratie van het autonome voertuig

Na de implementatie van de belangrijkste functies van het voertuig, zoals beschreven in de Ontwerp rapport([ontwerp rapport](#)), vond er een demonstratie plaats waarbij het voertuig in een simulatie-omgeving en in de echte wereld werd getest. Tijdens deze demonstratie werd getoond hoe het voertuig zelfstandig obstakels kon vermijden en een parcours kon afleggen. De presentatie van deze demonstratie benadrukte het succes van de implementatie van reinforcement learning en de correcte integratie van de hardware en software. Deze demonstratie was een belangrijk moment in het project omdat het de praktische resultaten van alle ontwikkelingsfasen samenbracht. Door middel van deze presentatie konden de stakeholders het voertuig in actie zien en beoordelen hoe goed het aan hun verwachtingen voldeed.

4. Conclusie

Het presenteren van de voortgang en resultaten van het autonome voertuigproject was een essentieel onderdeel van het succes van het project. Door gebruik te maken van visuele hulpmiddelen, zoals grafieken en simulaties, werden complexe resultaten eenvoudig uitgelegd aan zowel technische als niet-technische stakeholders. De presentaties hielpen niet alleen om goedkeuring te krijgen voor de voortgang, maar zorgden er ook voor dat er een duidelijke lijn was in het ontwikkelproces. Het bewijs van deze presentaties is te vinden in de Project Requirements Specificatie([project requirements sp...](#)) en de Probleem- en oplossingsstrategie([Probleem en oplossingss...](#)).

B1-K2-W3: Reflecteert op het werk

Reflecteren op het eigen werk is een belangrijk proces om te leren van ervaringen en verbeteringen door te voeren. Tijdens het autonome voertuigproject heb ik regelmatig momenten van reflectie ingebouwd om de voortgang van het project te evalueren en om te bepalen welke aanpassingen noodzakelijk waren voor het bereiken van de gestelde doelen. Deze reflectie heeft niet alleen bijgedragen aan de verbetering van het project, maar heeft mij ook geholpen mijn eigen vaardigheden verder te ontwikkelen.

1. Reflectie op de projectaanpak

In de Probleem- en oplossingsstrategie(**Probleem en oplossings...**) reflecteerde ik op de problemen en uitdagingen die gedurende het project aan het licht kwamen. Een van de belangrijkste leerpunten was het belang van het opdelen van het project in kleinere, beheersbare stappen. In eerste instantie was de scope van het project vrij breed, wat leidde tot complexiteit in de planning en uitvoering. Door het project op te splitsen in iteraties en voor elke fase duidelijke doelen te stellen, werd het eenvoudiger om de voortgang te bewaken en problemen sneller te identificeren.

Tijdens deze reflectiemomenten besprak ik ook hoe de gekozen methodes voor het trainen van het voertuig met reinforcement learning verbeterd konden worden. De oorspronkelijke beloningsstructuur bleek bijvoorbeeld inefficiënt, en na reflectie en het analyseren van de testresultaten, werd besloten om deze aan te passen om het leerproces te versnellen.

2. Reflectie op samenwerking en overleg

In mijn reflectie op de samenwerking binnen het project, zoals beschreven in de stage verslag (**stage verslag**), kwam ik tot het inzicht dat het overleg met stakeholders en teamleden essentieel was voor het succes van het project. Regelmatige feedback van stakeholders hielp om prioriteiten bij te stellen en om problemen vanuit verschillende perspectieven te bekijken. Dit leerde mij het belang van duidelijke en frequente communicatie binnen een projectteam, wat ervoor zorgde dat iedereen goed op de hoogte bleef van de voortgang en eventuele wijzigingen.

Daarnaast reflecteerde ik op mijn eigen rol in het team en het belang van het geven en ontvangen van constructieve feedback. Dit zorgde ervoor dat mijn communicatievaardigheden verbeterden en dat ik in toekomstige projecten beter voorbereid ben op teamgerichte samenwerkingen.

3. Technische reflectie en verbeteringen

Een belangrijk deel van mijn reflectie was gericht op de technische aspecten van het project. Gedurende het project heb ik regelmatig de prestaties van het systeem geëvalueerd en onderzocht welke onderdelen verder konden worden geoptimaliseerd. In de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**) beschreef ik hoe bepaalde softwarecomponenten, zoals de integratie van de motorbesturing en de sensoren, verbeterd konden worden. Door reflectie en analyse van de testresultaten stelde ik voor om de communicatie tussen de verschillende softwaremodules te optimaliseren, wat uiteindelijk leidde tot een snellere en meer robuuste werking van het voertuig.

Reflectie hielp ook bij het verbeteren van de simulatieomgeving, waarin het voertuig werd getest. Na het analyseren van de prestaties in de simulatie realiseerde ik mij dat de variabiliteit van de testscenario's niet voldoende was om het voertuig goed voor te bereiden op onverwachte situaties in de echte wereld. Dit leidde tot een aanpassing van de simulatieomgeving, waarbij meer diverse scenario's werden toegevoegd om de training realistischer te maken.

4. Reflectie op persoonlijke ontwikkeling

Gedurende het project was er ook ruimte voor persoonlijke reflectie, zoals beschreven in de stage verslag(**stage verslag**). Ik realiseerde me dat dit project me veel nieuwe inzichten en vaardigheden heeft opgeleverd, met name op het gebied van reinforcement learning en het ontwikkelen van autonome systemen. Ik heb geleerd hoe ik beter kan plannen, prioriteiten kan stellen en efficiënt kan samenwerken in een teamomgeving. Bovendien ben ik door dit project bewuster geworden van het belang van een gestructureerde aanpak bij het oplossen van complexe problemen. Deze reflectie helpt me niet alleen om het project tot een succesvol einde te brengen, maar ook om in de toekomst betere keuzes te maken in soortgelijke projecten.

5. Conclusie

Het reflecteren op mijn werk en het project als geheel heeft mij geholpen om de voortgang te verbeteren en waardevolle lessen te leren voor toekomstige projecten. Door regelmatig terug te kijken op de gekozen aanpak, de samenwerking en de technische uitdagingen, kon ik verbeterpunten identificeren en doorvoeren. Dit heeft bijgedragen aan de succesvolle afronding van het autonome voertuigproject. De reflecties die ik heb gedaan, zijn gedocumenteerd in de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**) en de stage verslag(**stage verslag**).

3. Technische reflectie en verbeteringen

Een belangrijk deel van mijn reflectie was gericht op de technische aspecten van het project. Gedurende het project heb ik regelmatig de prestaties van het systeem geëvalueerd en onderzocht welke onderdelen verder konden worden geoptimaliseerd. In de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**) beschreef ik hoe bepaalde softwarecomponenten, zoals de integratie van de motorbesturing en de sensoren, verbeterd konden worden. Door reflectie en analyse van de testresultaten stelde ik voor om de communicatie tussen de verschillende softwaremodules te optimaliseren, wat uiteindelijk leidde tot een snellere en meer robuuste werking van het voertuig.

Reflectie hielp ook bij het verbeteren van de simulatieomgeving, waarin het voertuig werd getest. Na het analyseren van de prestaties in de simulatie realiseerde ik mij dat de variabiliteit van de testscenario's niet voldoende was om het voertuig goed voor te bereiden op onverwachte situaties in de echte wereld. Dit leidde tot een aanpassing van de simulatieomgeving, waarbij meer diverse scenario's werden toegevoegd om de training realistischer te maken.

4. Reflectie op persoonlijke ontwikkeling

Gedurende het project was er ook ruimte voor persoonlijke reflectie, zoals beschreven in de stage verslag(**stage verslag**). Ik realiseerde me dat dit project me veel nieuwe inzichten en vaardigheden heeft opgeleverd, met name op het gebied van reinforcement learning en het ontwikkelen van autonome systemen. Ik heb geleerd hoe ik beter kan plannen, prioriteiten kan stellen en efficiënt kan samenwerken in een teamomgeving. Bovendien ben ik door dit project bewuster geworden van het belang van een gestructureerde aanpak bij het oplossen van complexe problemen. Deze reflectie helpt me niet alleen om het project tot een succesvol einde te brengen, maar ook om in de toekomst betere keuzes te maken in soortgelijke projecten.

5. Conclusie

Het reflecteren op mijn werk en het project als geheel heeft mij geholpen om de voortgang te verbeteren en waardevolle lessen te leren voor toekomstige projecten. Door regelmatig terug te kijken op de gekozen aanpak, de samenwerking en de technische uitdagingen, kon ik verbeterpunten identificeren en doorvoeren. Dit heeft bijgedragen aan de succesvolle afronding van het autonome voertuigproject. De reflecties die ik heb gedaan, zijn gedocumenteerd in de Probleem- en oplossingsstrategie(**Probleem en oplossingss...**) en de stage verslag(**stage verslag**).

Afsluiting B1-K2: Werkt in een ontwikkelteam

Binnen het autonome voertuigproject speelde samenwerking een centrale rol bij het succesvol realiseren van de doelstellingen. Door effectief overleg te voeren met zowel teamleden als stakeholders (B1-K2-W1), werd er voor gezorgd dat het project consistent op koers bleef en dat de verwachtingen helder waren. Regelmatige presentaties van de voortgang en resultaten (B1-K2-W2) maakten het mogelijk om tijdig feedback te verzamelen en de voortgang duidelijk te communiceren.

Reflectie op zowel het technische werk als de samenwerking binnen het team (B1-K2-W3) was essentieel voor het constant verbeteren van de aanpak en de kwaliteit van het project. Deze reflectiemomenten droegen bij aan mijn persoonlijke ontwikkeling en verbeterden de efficiëntie van de samenwerking binnen het team.

De ervaring die ik heb opgedaan in dit project heeft mijn vaardigheden in teamwork, communicatie en projectmanagement verder aangescherpt. Dit toont aan dat ik effectief kan functioneren binnen een ontwikkelteam, waarbij ik overleg, presentaties en reflectie gebruik om de kwaliteit van het werk te waarborgen en continu te verbeteren.

SLOTWOORD

Dit document biedt een uitgebreid overzicht van mijn bijdrage aan het autonome voertuigproject, waarin alle kernaspecten van softwareontwikkeling en samenwerking binnen een team aan bod komen.

Het realiseren van een complex systeem zoals een autonoom voertuig vereist niet alleen technische kennis, maar ook sterke vaardigheden op het gebied van planning, samenwerking en reflectie.

Door gestructureerd te werken aan het plannen, ontwerpen, realiseren, testen en verbeteren van software (B1-K1), heb ik laten zien dat ik in staat ben om een project van concept tot implementatie te leiden. Daarnaast heeft mijn actieve deelname aan overleg, presentaties en reflectie binnen het team (B1-K2) bijgedragen aan een soepel verloop van het project en het tijdig behalen van de projectdoelen.

Dit project heeft mijn technische en samenwerkingsvaardigheden verder ontwikkeld en versterkt. Het heeft me waardevolle inzichten gegeven in hoe ik complexe problemen kan aanpakken, efficiënte oplossingen kan implementeren en tegelijkertijd effectief kan communiceren en samenwerken met teamleden en stakeholders. De kennis en ervaring die ik heb opgedaan, vormen een sterke basis voor toekomstige projecten en uitdagingen.