

inleiding

In dit document leg ik mijn ervaringen en meningen vast. Tijdens dit project gaan we een database maken voor de Formule 1. De opdrachtgever Jorg Jansen wil hier graag een API-systeem bij hebben. Deze informatie wil hij dan gaan verkopen per API-call. De calls worden dan getrackt via een API-Key, zodat er goed overzicht is wie er moet betalen en hoeveel er betaald moet worden. De database moet een aantal dingen laten zien:

- Alle coureurs die meedoen aan de race.
- Alle circuits waar op geracet wordt.
- Alle teams die ooit hebben meegedaan.
- Alle race uitslagen.
- De race planning.

Voor mij wordt het een uitdaging om een staging area te maken. Hiervoor moeten we namelijk met een nieuw soort database werken. Op dit vlak heb ik nog weinig ervaring. Ook lijkt mij de API-verbinding tussen de database en staging area lastig om te maken.. Dit komt doordat we hebben gekozen om het te werken met Python.

Bij dit project ga ik tevens werken aan het ontvangen van feedback op het gedrag. Ook ga ik mezelf verbeteren in het geven van feedback op het reviewen van producten.

Oordeel kwaliteit deelproduct

In dit hoofdstuk geef ik mijn mening over de kwaliteit die we hebben opgeleverd.

PvA

Ik ben zelf heel erg tevreden over ons Plan van Aanpak.

Ik vind dat we erg goed hebben beschreven wat de opdrachtgever kan verwachten. Dit hebben we beschreven in hoofdstuk 6 *op te leveren producten en kwaliteitseisen*. De informatie die we daar hebben staan is dan ook in **5xW en 1xH** geschreven of **smart**. Omdat dit zo is gedaan wordt het heel erg duidelijk voor de opdrachtgever wat hij kan verwachten en ook missen we geen details.

FO

Ik vond dat ons FO er goed uit zag en het bevatte alle onderdelen die verwacht zijn in het FO. Alleen na het gesprek met de opdrachtgever prototype oplevering kwam aardig wat feedback op ons FO. Voordat we de feedback kregen was ik tevreden met het FO. Na de feedback kwamen we achter dat we niet alles op orde hebben staan. Zo zat er een fout in ons CDM wat door getrokken werd op de rest van het document.

Oordeel eindproduct

PvA

Het doel van een PvA is om goed in kaart te brengen wat er wordt verwacht door de opdrachtgever en wat je gaat opleveren. Naar mijn mening is dit goed gelukt.

In de inleiding zou duidelijk gemaakt moeten worden waarvoor het document is en wat er in staat. Ik vind dit onderdeel goed gelukt, omdat er in wordt gezegd voor wie we het maken en wat er gevraagd wordt door de opdrachtgever.

In de achtergrond van het project hoort informatie te staan hoe dit project op stand is gekomen. Dit hebben we ook goed duidelijk gemaakt, dus hier ben ik zeer tevreden mee. We hebben bijvoorbeeld goed benoemd dat Jorg Jansen een database mét een API wil hebben om de data te kunnen verkopen.

De doelstelling, opdracht en op te leveren resultaten voor het bedrijf is een onderdeel van de PvA. Ik vind bij dit hoofdstuk dat het goed en volledig gemaakt is. Dit hoofdstuk moet namelijk goed in kaart brengen wat de opdracht precies inhoudt. Ook moet je goed kunnen zien wat er opgeleverd moet worden. Dit hebben we gedaan door een lijst van producten op te stellen die de opdrachtgever gaat krijgen.

Bij het hoofdstuk projectgrenzen moet komen te staan wat we allemaal gaan uitvoeren. Dit hebben we allemaal opgeschreven in een lijst en punt voor punt benoemd.. Hierbij hebben we goed opgelet dat we duidelijke en volledige zinnen schrijven. De duidelijkheid komt in de volgende zin bijvoorbeeld goed naar voren: "het team zal het product niet deployen". Zo weet de opdrachtgever dat het 'deployen' niet in scope is van de opdracht.

In de randvoorwaarden wordt beschreven wat we nodig hebben van de opdrachtgever om het project tot een goed einde te brengen. Ik vind dat we dit niet helemaal goed hebben beschreven. We hadden duidelijker moeten opschrijven wat we precies van de opdrachtgever verwachtten. We hebben ook dingen opgeschreven waar de opdrachtgever niet heel veel inspraak op heeft. Bijvoorbeeld in PvA 5.4 Locatie zeggen we dat hij een locatie ter beschikking moet stellen.. Helaas weten we dat een docent niet gedurende het hele project een lokaal voor ons kan reserveren.

Bij de op te leveren producten en kwaliteitseisen staat hoe we ervoor zorgen dat datgene dat we gaan opleveren ook met kwaliteit opgeleverd wordt. Ik vind dat we dit hoofdstuk heel goed hebben geschreven omdat we de eisen van het product goed SMART geschreven hebben. Ook zijn de product kwaliteitseisen goed doelgericht geschreven. We hebben we de Proces Kwaliteit beschreven met 5xW en 1xH. Dit zorgde ervoor dat het proces helemaal goed beschreven is en hoe zaken worden aangepakt om te zorgen dat de kwaliteit van het product wordt geborgd.

In het hoofdstuk Ontwikkelmethode behoort te staan hoe we het proces gaan aanpakken en welke ontwikkelmethode we daarbij gebruiken. Persoonlijk vind ik dat dit hoofdstuk goed en volledig is en dat we duidelijk hebben gemaakt welke hulpmiddelen we gebruiken. Ook hebben we RUP goed uitgelegd en verteld wat dat inhoudt. Verder hebben we goed naar voren gebracht dat we zaken uit de scrum hebben gebruikt. Deze scrum artefacten hebben we goed beschreven en we hebben aangegeven waar we deze voor gaan gebruiken.

Het hoofdstuk Projectorganisatie en communicatie is compleet en zorgt er dan ook voor dat de opdrachtgever goed kan zien wie bij ons waarvoor verantwoordelijk is. Hij kan dan ook makkelijk zien wat onze dagelijkse afspraken zijn. Bijvoorbeeld dat er elke ochtten om 9.15 uur een stand-up is. Hier staat ook alle contact informatie van diegenen die

betrokken zijn bij het project. Dit helpt ons heel erg met het besparen van tijd zodat we bijvoorbeeld niet naar een e-mailadres hoeven te zoeken. Daarom vind ik dat dit hoofdstuk goed in elkaar zit.

Het hoofdstuk Planning vind ik goed in elkaar zitten. Dit hoofdstuk moet namelijk goed laten zien aan de opdrachtgever wanneer we wat doen. Dit hebben we hier goed opgenomen. Daarbij hebben we ook rekening gehouden met onze vakantie. Deze dagen zouden we niet aan de slag gaan met het project. We hebben van alle projectleden de vakanties in een lijst gezet. De Gantt-chart klopt helaas niet helemaal met de data. Dat is wel erg jammer, want volgens de Gantt-chart zouden we nog een paar weken meer aan de slag zijn. Dit geeft dus een verkeerd beeld.

Risico's is het hoofdstuk waarin staat welke risico's er zijn tijdens dit project. De risico's geven aan waarom er vertraging of functionaliteiten niet zouden afkomen. Daarbij is er beschreven hoe we te werk zouden gaan als er één van de risico's die benoemd zijn, zich zouden voordoen. Dit hoofdstuk is dan ook niet helemaal volledig. Ten eerste hebben we niet alle risico's benoemd die zich konden voordoen. Ook is er niet duidelijk genoeg beschreven wat we zouden gaan doen als een risico zich zou voordoen. Dit hoofdstuk is dan ook niet helemaal goed uitgeschreven. Hier hadden we meer aandacht aan moeten besteden.

Over het hele PvA ben ik ook wel erg tevreden en denk ik ook dat het voor de opdrachtgever helder was en wat hij kon verwachten.

FO

In het FO moeten alle keuzes staan die we hebben gemaakt en ook waarom we ergens voor gekozen hebben. De inleiding geeft duidelijk aan wat er in dit document staat. Daarom vind ik dit een goede inleiding. Functionele eisen geven een duidelijk overzicht wat het systeem moet gaan doen. Dat hebben we goed beschreven, in de opsomming staat wat het systeem allemaal moet kunnen.

In het hoofdstuk Fully dressed use cases staat welke actor welke functionaliteit moet hebben. Dit hebben we al goed duidelijk gemaakt met een use case diagram. De fully dressed use cases zijn helemaal uitgewerkt op de manier zoals we dat geleerd hebben bij de Course OOAD. Persoonlijk vind ik deze fully dressed use cases overbodig omdat ze alleen maar CRUD functionaliteiten doorlopen. We hadden afgesproken dat deze eigenlijk niet meer nodig zijn nadat we een toegevoegde use case hadden. Daarom vind ik dat dit hoofdstuk niet goed opgebouwd is.

In de Entity relationship diagram staan alle verwoordingen die van toepassing zijn op het project. Deze zijn dan ook voorgelegd aan de opdrachtgever(domein expert) dit houdt in dat de zinnen allemaal kloppen met de inhoud. Deze zinnen vind ik goed in elkaar zitten omdat ze goed syntactisch geschreven zijn. Ook hebben we onder de verwoordingen onze besluiten van feittypes opgeschreven. Hierin staan alle extra keuzes die we hebben gemaakt. Deze besluiten zijn niet in de verwoordingen te vinden. Het CDM die uit de verwoordingen en besluiten is gemaakt staat ook in dit hoofdstuk. Op het CDM ben ik zelf heel erg tevreden. De entiteiten, attributen en relaties kun je ook 1 op 1 halen uit de verwoordingen en besluit keuzes. Daarom vind ik dit hoofdstuk heel erg goed en duidelijk.

Als laatste hoofdstuk van het FO hebben we nog executie rechten en rollen. In dit hoofdstuk staan welke rollen welke rechten hebben om functionaliteiten te gebruiken. Deze rechten kun je afleiden van ons use case model. In het use case model hebben we actoren staan en daar zijn de namen van de rollen ook naar vernoemd. Omdat het goed afleidbaar is vanuit de rest van het document vind ik dat dit onderdeel goed is geschreven en duidelijk maakt wie waar toegang tot heeft.

TO

In dit document staan al onze technische beslissingen en hoe we dingen hebben toegepast die we hebben gemaakt in het FO. Hier hebben we de non-functional requirements goed uitgewerkt. We hebben ze in een lijst opgeschreven. De lijst is niet zo groot en we vonden ook alleen maar performance based requirements. Ik denk zelf dat we misschien niet genoeg vragen hebben gesteld aan onze opdrachtgever om er meer te kunnen benoemen. Van de non-functional requirements hebben we wel goed beschreven hoe we ze aanpakken. Daarom vind ik dit stuk kort maar wel prima beschreven.

Het Architecture overview hoofdstuk legt uit hoe de systemen met elkaar communiceren. Dit wordt dan ook duidelijk afgebeeld in de afbeelding met het overzicht. In de afbeelding wordt ook goed beschreven wat elk component is.

Het PDM vind ik ook heel goed omdat het moet laten zien hoe de database in elkaar zit. Dit is goed omdat we deze PDM hebben laten genereren vanuit het CDM, daarom klopt deze ook helemaal vanuit het FO.

Ook hebben we Integrity rules gemaakt in het TO. Dit zijn de business rules vanuit het FO, maar dan veel meer technisch opgeschreven, zodat een andere programmeur alleen hier naar hoeft te kijken om te zien hoe ze werken. Daarnaast hebben we natuurlijk ook ons besluit geschreven waarom we ze zo hebben gemaakt. De onderbouwing van de Rules vind ik dan ook goed beschreven en laat duidelijk zien waarom we ze hebben geschreven.

In het eind van het TO hebben we nog de lastigste stukken code beschreven. Wanneer je dit leest begrijp je onmiddellijk hoe het ontwerp in elkaar zit. Ik vind dat we dit ook heel goed op orde hebben. Err worden stukken code getoond met een beschrijving erbij. Dit zorgt ervoor dat wanneer je dit leest alle functionaliteiten begrijpt en tevens begrijpt hoe je data erdoor moet sturen..

Over heel het TO ben ik tevreden. Het document doet namelijk waarvoor die bedoeld is. Met als grootste reden dat andere developers er verder op door kunnen gaan en begrijpen wat er gedaan is

Code

Ik ben zeer tevreden over de code kwaliteit die we geschreven hebben. We hebben namelijk een goede DoD opgesteld waardoor we allemaal dezelfde regels gebruikt hebben. Het DoD bevat de volgende regels over code, namelijk:

- Is traceerbaar vanaf het PDM.
- Moet in een keer uitgevoerd kunnen worden.
- Bevat een create statement.
- Bevat een drop tabel if exists.
- Bevat Keys.
- Bevat stored procedures.
- Bevat triggers.
- Bevat constraints.
- De naam van het script is in lower-camelcase geschreven.
- Gebruiken SP_ voor de naam in grote letters.
- De naam van de procedure wordt in lower camelcase geschreven.
- De stored procedures worden geunittest op succes scenario en als mogelijke een alternative flow.
- Een stored procedure die een select statement gebruikt hoort een index te gebruiken als dit de query versneld.

- De JSONS worden zo klein mogelijk gehouden door alleen de benodigde data op te slaan.
- De variabelen zijn in lower camelcase geschreven.
- Gebruiken count.
- Gebruiken TR_ voor de naam.
- De naam van de trigger wordt in lower camelcase geschreven.
- Wordt geunittest.
- Testen zijn uitgevoerd en slagen.
- De naam van de test geeft aan wat de unittest test.
- De testen testen nuttige functionaliteiten.
- Testresultaten zijn gedocumenteerd en opgeslagen in de /doc map.
- Testen zijn gereviewd door minimaal één teamlid.
- Testen zijn pas goedgekeurd na één keer feedback te hebben verwerkt.

Dit hebben we op alles van de code waarop het van toepassing is getoetst. Hierdoor vind ik dat alle code die we geschreven hebben goed is gedaan. Bijvoorbeeld is bij de pull request Pull Request toevoegen race uitlagen Hierin kun je ook feedback zien die we elkaar geven. We gebruikten daarnaast ook Jira om elkaar feedback te geven.

Evaluatie RUP

In het begin begrepen we RUP nog niet zo goed. Hierdoor was er veel verwarring wat er verwacht werd. De verwarring kwam ook door onze opdrachtgever. Die wilde graag dat we van tevoren alle Use Cases zouden opstellen. Alleen bij RUP zou dit niet horen. Zo kwamen we in discussie met de docent. We hebben toen de procesbegeleider om advies gevraagd en hij zei dat we dat wel moesten gaan doen, ook al is het niet RUP. Uiteindelijk is de klant natuurlijk koning. Nadat we de problemen hebben besproken met de professional skills docent. Ging hij ervoor zorgen dat ze met elkaar zouden gaan praten over hoe we met RUP moeten werken. Hierna is het voor ons ook een stuk duidelijker geworden wat er verwacht wordt van ons met RUP.

In het PvA hebben we beschreven dat we aspecten van Scrum zouden meenemen. Het Scrumbord is er één van. Wij hebben ervoor gekozen om deze mee te nemen, omdat dit bord altijd goed overzicht houdt met wat er nog te doen is. Dit bord hebben we online op Jira staan. Dit hebben we alleen nog niet goed gebruikt, want we liepen tegen een paar problemen aan. Zo konden we bijvoorbeeld geen backlog items aanmaken. Ook zitten we nog niet in iteratie waar we heel druk aan het maken zijn. Bij die fase is het bord heel erg handig om veel dingen erop te zetten wat gedaan moet worden.

Beschrijving rollen

Zelf heb ik de rol Software Architect gekregen. Hier heb ik zelf nog niet heel veel mee gedaan. Ik heb alleen het FO en TO goed gereviewd voor de groepsgenoten op wat ze gedaan hadden.

Voor de 2e helft van het project heb ik dan ook nog veel te leren wat er bij de rol nog allemaal komt te kijken. Zo staat er bijvoorbeeld nog nergens in het FO of TO op welk soort systeem het draait. Dit staat alleen in het onderzoek dat ik geschreven heb. Ook ga ik navragen aan docenten wat er precies wordt verwacht van deze rol. Zodat ik beter weet welke takken er horen bij deze functie.

We hebben na week 3 ook allemaal de rol 'tester' gekregen. Deze rol was eerst van Niels, maar hij is helaas uitgevallen tijdens het project door een burn-out. Daarom hebben we besloten deze gezamenlijk over te nemen. Deze rol brengt ook weer wat andere verantwoordelijkheden met zich mee. Nu moet ik opletten of alle testen slagen en of er goed getest wordt.

Hierdoor is de volgende situatie ontstaan: Toen ik de code van Jannis aan het reviewen was, was er een unittest die niet meer klopte. Ik ben toen gaan onderzoeken wat er kapot is gegaan met de unittest. Door dat te doen kwam ik erachter dat die unittest zelf helemaal niet klopte volgens ons DoD. Deze had verkeerde naamgeving voor de Stored Procedure en variabelen. Dit was niet de reden waarom die faalde, maar wel iets wat nog opnieuw gedaan moest worden met de goede benaming volgens het DoD. Ik heb toen gekeken wie die test geschreven had en tegen dat teamlid gezegd dat hij het moest gaan verbeteren.

toelichting op competenties

ISEP-02

Je maakt een analyse van de eisen en wensen voor een informatiesysteem en stelt op basis van deze analyse een dynamisch model en een statisch model (d.w.z. conceptueel datamodel inclusief business rules) op en documenteert deze in een functioneel ontwerp.

Ik heb alle verwoordingen gemaakt van de coureur, dit zijn de verwoordingen 13 tot en met 18. Dit heb ik gedaan om goed in beeld te krijgen wat er allemaal bij coureur hoort. Deze verwoordingen zorgen ook voor duidelijkheid van hoe ik de tabel coureur heb gemaakt. Na de verwoordingen gemaakt waren heb ik het ook toegepast in het CDM. Later gegenereerd in het PDM. Die dan weer omgezet wordt naar een create script. We hebben voor de verwoordingen ook gekozen om het precies hetzelfde te doen als hoe we dat deden met de course DMDD. Dit deden we dan zodat we allemaal zelfde soort verwoordingen hebben.

Natuurlijk heb ik niet alleen aan die verwoordingen gezeten. Ik heb daarom ook veel van de andere verwoordingen gereviewd. Door sommige punten van mijn feedback die ik jannis heb gegeven kwamen er betere verwoordingen uit. Deze verwoordingen zijn dan ook van cruciaal belang dat ze goed in elkaar zitten. Zonder deze verwoordingen kun je ook geen CDM, PDM en create script maken.

ISEP-03

Je stelt een technisch ontwerp op voor het te ontwikkelen informatiesysteem dat consistent is met het functioneel ontwerp en motiveert de ontwerpbeslissingen in relatie tot met de requirements.

In het TO heb ik aan een paar dingen gewerkt. Hieronder valt bijvoorbeeld het kopje tabellen. Hierin leg ik alle tabellen van de database uit. Door deze tabellen is voor toekomstige programmeurs die hier aan werken gemakkelijk om te zien wat waarvoor staat in de database. Deze namen komen dan ook overeen in de database zodat het ook heel erg makkelijk wordt gemaakt om hiermee te werken. Deze tabellen komen uit het PDM en het PDM uit het CDM die in het FO staat.

Eerder in het document heb ik een architecture overview gemaakt en de beschrijving erbij gezet. Dit zorgt voor een duidelijk overzicht hoe het systeem met elkaar verbonden is en welke systemen er gebruikt worden. Dit helpt met het opzetten van de applicaties.

ISEP-04

Je bewaakt continu de kwaliteit van het ontwerp en van het te ontwikkelen systeem.

Ik heb meerdere dingen gereviewd, zoals de verwoordingen die Jannis heeft gemaakt. Hierin checkte ik of de zinnen goed syntactisch stonden. Ook keek ik of hij alle Attributen en entiteiten goed benoemde. Daaronder staat het predicaat. Hierop had ik dan feedback, omdat hij deze niet helemaal goed geschreven had. Na de feedback was het helemaal compleet en kloppend. Dit zorgde ervoor dat de database heel makkelijk en goed in elkaar zit.

Ik heb het versturen van JSON naar de API gereviewd. Deze had Dylano gemaakt. Eerst snapte ik niet helemaal goed wat er gedaan werd in de code. Bij Dylano heb ik nagevraagd hoe het in elkaar zat. Hij kon het eerst ook niet goed uitleggen met woorden wat het precies deed. Met behulp van Paint legde hij uit hoe het zou moeten werken. Na zijn uitleg ben ik nog een keer door de code gegaan en zag dat het goed in elkaar zat.

ISEP-05

Je implementeert in teamverband het informatiesysteem in overeenstemming met het ontwerp ervan, zorgt voor traceerbaarheid tussen ontwerp en implementatie en motiveert eventuele afwijkingen.

Wij hebben in de code een goede traceerbaarheid gemaakt. Deze kun je al vinden als je een Stored Procedure inkijkt. We hebben daar comments neergezet zodat je weet aan welke usecase die gebonden is. In het Fo staan fully dressed use cases die verwijzen naar de stored procedure naam. In het TO hebben we een lijst staan welke functionaliteiten afgekomen zijn en welke we niet hebben afgekregen. Ook hebben we de lastige stukken code goed uitgelegd in het TO. Dit zorgt weer voor extra traceerbaarheid van de code. Door de uitleg kun je precies weten hoe de code werkt en waarom we het zo hebben gemaakt. Ook hebben we in het hoofdstuk Data laten opslaan of data updaten in MongoDB van het TO een schema staan hoe code wordt doorgestuurd. Dit schema is zo gemaakt om aan te duiden waarom we een executieschema hebben in ons systeem. Hierdoor kan je onze keuzes ook mooi zien. Dit zorgt ervoor dat volgende developers er makkelijker mee door kunnen gaan en zien wat we gemaakt hebben.

Werken aan leerdoelen

Leerdoel 1

Specifiek: Het verbeteren van het ontvangen van negatieve feedback op mijn gedrag. Meetbaar: Het aantal keer dat ik negatieve feedback verwerk in de volgende sprint zodat het niet terugkomt tijdens het retrospective. Acceptabel: Het verbeteren van communicatieve vaardigheden is essentieel voor een effectieve samenwerking en professionele ontwikkeling en het is belangrijk voor mij om dit doel na te streven. Realistisch: Ik zal mijn leerdoel ondersteunen door actief deel te nemen aan teamvergaderingen, feedback te vragen aan groepsgenoten. Tijdgebonden: tijdens het volgend project.

Hieraan ga ik werken om niet in de verdediging te schieten als ik feedback krijg op mijn gedrag. Dit heb ik dan ook goed gedaan bij de IPV. Toen kreeg ik feedback dat ik te snel afgeleid raak door mijn mobiel. Ook kreeg ik er een tip bij dat als ik die in mijn tas leg deze dan makkelijker te negeren zou zijn.

Situatie beschrijvingen: Bij de 2e retrospective op donderdag 30 mei heb ik feedback ontvangen van Jannis. De feedback was dat ik wel af en toe te laat was en dat het hem erg irriteerde. In de ochtend was er een meeting met Jorg. Hier was ik niet aanwezig omdat ik me verslapen had. Aan deze feedback heb ik zelf niet heel veel omdat ik er niet echt wat aan kan doen dat ik me verslapen had. Ik heb al 4 wekkers staan die afgaan in de ochtend. Het doel was ook dat ik er niet op tegen in ga en in de verdediging schiet. Dat heb ik dan daarom ook niet gedaan. Nu ik terugkijk, had ik ook moeten vragen hoe ik het zou kunnen voorkomen. De volgende keer wanneer ik nieuwe feedback krijg ga ik ook vragen hoe ik het de volgende keer beter zou kunnen aanpakken.

Bij de 3e retrospective op dinsdag 2 juni, kreeg ik feedback van Dylano dat ik eerder moest aangeven als ik er een ochtend niet bij zou zijn. Ik kreeg deze feedback omdat ik een keer een afspraak was vergeten en een melding in de ochtend kreeg dat ik die afspraak had staan. Dat vond hij erg vervelend, hier schoot ik ook niet in de verdediging, maar ik wilde wel weten hoe ik het dan moet aangeven. Dit heb ik dan ook niet teruggezegd. Hierdoor laat ik goed zien dat ik een stuk vooruit ben gekomen met mijn leerdoel.

Leerdoel 2

Specifiek: Mijn vermogen verbeteren om feedback te geven over de kwaliteit van code, met de nadruk op leesbaarheid, efficiëntie en onderhoudsgemak. Meetbaar: Ik zal mijn feedbackvaardigheden bijhouden door regelmatig feedback te geven aan mijn teamleden tijdens code reviews. Ik zal minstens één keer per week feedback geven op minimaal drie verschillende stukken code. Acceptabel: Het geven van feedback over code is van cruciaal belang voor het verbeteren van de softwarekwaliteit. Realistisch: Ik zal mijn leerdoel ondersteunen door actief deel te nemen aan code reviews, feedback te vragen aan groepsgenoten voor codekwaliteit. Tijdgebonden: tijdens het volgend project.

Dit leerdoel ga ik uitvoeren door de wat lastigere onderdelen te reviewen. Als ik er dan niet uitkom ga ik navragen wat er bedoeld wordt met de code. Dit is alleen nog niet in uitvoering geweest omdat we nog heel weinig code hebben op dit punt van het project. We zitten ook nog niet in de construction fase.

Situatie beschrijvingen: Dylano had de `sendJsonToAPI` geschreven, deze functionaliteit heb ik gereviewd. Tijdens het reviewen snapte ik een paar dingen niet. Deze heb ik nagevraagd aan Dylano. Na de uitleg snapte ik het en had ik ook als feedback meegegeven dat hij het moest documenteren. Daarnaast heb ik feedback op de naamgeving, dit is allemaal verbaal gedaan omdat we naast elkaar zaten toen hij het uitlegde. Ook hebben Dylano en ik samen de server jobs gemaakt, hier gaven we telkens feedback aan elkaar over hoe het misschien beter kon.

Verwachting Eind Oplevering

Zelf verwacht ik dat de eindoplevering goed zou gaan verlopen. Helaas doen we dit project, maar met 3 personen omdat er 1 is uitgevallen. Dit gaat er wel voor zorgen dat het project wat kleiner zou zijn dan andere projecten. Hieronder zou de kwaliteit niet achteruitgaan.

We moeten nog wel hard aan de slag om er voor te zorgen dat we het project gaan halen.

conclusie

Tijdens dit project heb ik mezelf flink ontwikkeld, zowel met technische vaardigheden als met mijn leerdoelen. Op het technische vlak heb ik heel wat dingen geleerd, bijvoorbeeld hoe ServerJobs werken in Microsoft SQL Server. Ook had ik nog nooit met mongoDB gewerkt. Toch is dit heel goed gegaan.

Professioneel ben ik ook goed vooruit gegaan. Daarnaast heb ik aan mijn leerdoelen gewerkt. Helaas heb ik hier niet zoveel aan kunnen werken als ik had gewild. Maar ook met de weinige voorhanden situaties heb ik zeker goede stappen kunnen zetten voor mijn leerdoelen. Vooral bij het reviewen van lastige taken heb ik een grote stap gezet. Dit zou ik in het vorige project nog niet hebben gedaan, omdat ik niet wilde overkomen als iemand die er niks van snapt. Daarom ga ik deze leerdoelen ook door trekken in het volgende project.