Plan van aanpak -DWA Project

Auteurs: Steven Velderman (636657), Kachung Li (642552), Alex Cheng (636749), Ritse Huiskes (661137), Glenn Hulscher (568554)

Course: DWA Project

Projectgroep: Pardellos

Docenten: Robert Holwerda, Sander Leer, Helen Visser

Datum: 10-11-2020

Plaats: Ruitenberglaan 26, Arnhem

Inhoud

[1. Inleiding 2](#_Toc56065489)

[2. Achtergrond van het project 3](#_Toc56065490)

[3. Doelstelling, opdracht en op te leveren resultaten voor het bedrijf en school 4](#_Toc56065491)

[4. Projectgrenzen 5](#_Toc56065492)

[5. Randvoorwaarden 6](#_Toc56065493)

[6. Op te leveren producten en kwaliteitseisen 7](#_Toc56065494)

[7. Ontwikkelmethoden 9](#_Toc56065495)

[8. Projectorganisatie en communicatie 10](#_Toc56065496)

[Indeling werkdag 10](#_Toc56065497)

[Aanwezigheid 10](#_Toc56065498)

[Taken 11](#_Toc56065499)

[Communicatie 11](#_Toc56065500)

[GitHub 12](#_Toc56065501)

[Definiton of Done 12](#_Toc56065502)

[Procesbeschrijving 12](#_Toc56065503)

[9. Projectplanning 13](#_Toc56065504)

[Bibliografie 14](#_Toc56065505)

# 1. Inleiding

In dit document zit het plan van aanpak voor de opdracht LaterLezer. LaterLezer is een reader-app die artikelen van externe websites kan bewaren en deze artikelen in een vriendelijke vorm weer te geven, dus zonder advertenties en andere minder relevante elementen van de oorspronkelijke website, en ontworpen met eigen typografie. LaterLezer zal bestaan uit een browserextensie en een mobiele webapplicatie.

Dit plan van aanpak is voornamelijk bedoeld voor de Product Owner, Robert Holwerda, zodat hij inzicht krijgt op het werkproces van de opdrachtnemers. Op basis van dit document kan hij zien of het project goed is begrepen door de opdrachtnemers. Daarna kan de Product Owner het plan van aanpak naar de stakeholders sturen als dit van toepassing is. In dit geval is de Product Owner ook de opdrachtgever. Binnen het team wordt het ook duidelijker wat er van elkaar verwacht wordt.  
In het plan van aanpak komen de volgende hoofdstukken aan bod:

* **Achtergrond van het project**In dit hoofdstuk beschrijven wij de achtergrondinformatie van het project. Hierin staat voornamelijk de aanleiding van het project en waarom de opdrachtgever deze opdracht wil laten uitvoeren. Daarnaast worden alle betrokken personen en gebruikers in dit hoofdstuk benoemd.
* **Doelstelling, opdracht en op te leveren resultaten voor het bedrijf**In dit hoofdstuk formuleren wij de doelstelling van het project en de beoogde resultaten daarvan.
* **Projectgrenzen**In dit hoofdstuk beschrijven wij de projectgrenzen. Daarmee geven wij aan wat er nog onder de opdracht van LaterLezen valt en wat er net buiten valt. Hierdoor bakenen wij het project af.
* **Randvoorwaarden**In dit hoofdstuk beschrijven wij de randvoorwaarden van het project. Dit zijn alle voorwaarden die wij aan de opdrachtgever stellen om het project succesvol te laten slagen. De voorwaarden die hierin beschreven zijn, regelen de externe personen, zoals de Product Owner.
* **Op te leveren producten en kwaliteitseisen**In dit hoofdstuk beschrijven wij alle beoogde producten voor de opdrachtgever in detail en in kleinere onderdelen. Daarbij beschrijven we aan welke kwaliteiten het product en het proces aan moet voldoen. Dit hoofdstuk heeft samenhang met het hoofdstuk: “Doelstelling, opdracht en op te leveren resultaten voor het bedrijf”.
* **Ontwikkelmethoden**In dit hoofdstuk beschrijven wij welke ontwikkelmethode we gebruiken in het project van LaterLezen. Hierbij beschrijven wij wat de ontwikkelmethode inhoudt en onderbouwen wij waarom er gekozen is voor een specifieke ontwikkelmethode.
* **Projectorganisatie en communicatie**In dit hoofdstuk beschrijven we welke afspraken er gemaakt zijn met het team en de opdrachtgever. Ook staan hier afspraken in die binnen het team zijn gemaakt.
* **Planning**In dit hoofdstuk beschrijven wij planning van het project. Hierin staan voornamelijk terugkerende (scrum) afspraken met de product owner en scrum coach
* **Risico’s**In dit hoofdstuk beschrijven wij risico’s die buiten onze macht op kunnen treden in het project. Deze risico’s zijn niet af te vangen in de planning of de randvoorwaarden.

# 2. Achtergrond van het project

Er zijn al een aantal reader-apps beschikbaar, zoals Instapaper, Pocket, Raindrop.io etc. Deze reader-apps missen een aantal handige functionaliteiten die het nóg makkelijker zouden maken voor zijn gebruikers. Als een onderzoeker over een bepaald onderwerp veel artikelen aan het lezen is voor zijn/haar onderzoek, wilt diegene op een snelle en makkelijke manier de goeie artikelen opslaan om deze later eenvoudig opnieuw te kunnen raadplegen.

De Product Owner is een onderzoeker. Hij zoekt een reader-app die wetenschappelijke artikelen voor hem bewaart zodat hij deze artikelen later terug kan lezen. Hij heeft aangegeven dat de huidige reader-apps niet aan zijn wensen voldoen omdat deze apps een aantal functionaliteiten missen. Er zijn genoeg missende functies waarvan hij denkt dat het waard is om een nieuwe reader-app in elkaar te laten zetten. Daarom is hij naar ons toegekomen met deze opdracht. Deze reader-app is bedoeld om onderzoekers te ondersteunen met het opslaan en weergeven van wetenschappelijke artikelen.

De reader-app heeft niet de hoogste urgentie, het is niet dat de Product Owner niet zijn onderzoeken kan doen en geen wetenschappelijke artikelen kan opslaan als hij de Laterlezer reader-app niet heeft. Het nadeel is wel dat hij wetenschappelijke artikelen zou missen met de extra handige functies die niet in huidige reader-apps bestaan.

# 3. Doelstelling, opdracht en op te leveren resultaten voor het bedrijf en school

Het probleem dat de opdrachtgever heeft is dat er reader-apps zijn die nog geen functionaliteiten bevatten die gericht zijn op onderzoekers. Zo bevatten bestaande reader-apps bijvoorbeeld geen functionaliteit om artikelen te bewaren met hiërarchische tags.

De doelstelling van het project is om een betrouwbare reader app te ontwikkelen die extra functionaliteiten bieden voor onderzoekers die de bestaande reader-apps niet ondersteunen. Een reader app is een handige tool die gebruikt kan worden om artikelen op het internet te bewaren en later terug te lezen. Bij een standaard reader app komen de volgende zaken aan bod:

* De app moet artikelen kunnen opslaan van de gebruiker.
* De app moet deze opgeslagen artikelen kunnen ophalen en weer tonen aan de gebruiker.
* Deze reader app bevat een vriendelijke vorm, denk hierbij aan geen advertenties en ander afleidend spul dat nog op de oorspronkelijke site stond.
* De opgeslagen artikelen kunnen georganiseerd worden door middel van mappen en tags.
* De app heeft de mogelijkheid om artikelen te delen met andere gebruikers.

De extra functionaliteiten van de reader-app zijn als volgt:

* Pay-walls omzeilen. Gebruikers die abonnementen van bepaalde websites niet hebben, kunnen in Laterlezer nog steeds artikelen van die website lezen, zelfs als daar niet voor betaald is.
* De tags meer gebruikersvriendelijk maken, zodat een gebruiker een bruikbare archief heeft d.m.v. tags
* Ondersteuning bevatten voor langere teksten, behalve artikelen opslaan en later te kunnen lezen willen wij hetzelfde resultaat hebben voor bijvoorbeeld boeken.
* Ondersteuning voor het lezen van PDF’s, zodat de gebruiker artikelen op PDF formaat in de reader-app kan zetten.
* Het vermogen om goede metadata op te halen, bijvoorbeeld zoeken op specifieke auteur of tijdschrift
* Mogelijkheid om van een bestaande reader-app naar de Laterlezer reader-app over te stappen door hun archief van artikelen te importeren zodat zij van deze extra features gebruik kunnen maken.

Laterlezen zal bestaan uit een browserextensie, die artikelen van andere websites op kan slaan, en een mobiele webapplicatie, waarin de gebruiker zijn opgeslagen artikelen kan zien met de features die hierboven zijn beschreven.

# 4. Projectgrenzen

Het team werkt tijdens het project “LaterLezer” gedurende 8 weken maximaal 5 werkdagen per week gezamenlijk aan de groepsproducten van negen tot vijf.

De geschreven code wordt geplaatst op GitHub in aparte branches zodat elke functionaliteit los ontwikkeld kan worden. Uiteindelijk worden deze functies samengevoegd op de main branch.

Tijdens het project werkt het team naast de opdracht van LaterLezer ook aan zijn eigen professionele ontwikkeling aan de hand van leerdoelen.

De opdracht wordt uitsluitend gerealiseerd met Javascript, HTML en CSS-code. Er wordt gebruik gemaakt van React als framework voor het front-end. Voor de back-end wordt er gebruik gemaakt van Express. De database structuur wordt opgezet met Mongo en Mongoose.

In de sprintplanning spreekt het team met de Product Owner af aan welke User Cases per sprint gewerkt gaat worden. In de basis wordt ervoor gezorgd dat essentiële onderdelen van de “LaterLezer” app eerst ontwikkeld worden. Als er tijd is om extra functionaliteit op te pakken dan zal dit in overleg met de Product Owner gedaan worden. Een voorbeeld van extra functionaliteit is het omzetten van de webapplicatie naar een native applicatie aan de hand van PhoneGap.

Er wordt uiteindelijk een goed functionerende webapplicatie opgeleverd aan de Product Owner.

# 5. Randvoorwaarden

Tijdens het project moeten er tussen de opdrachtgever en het team een paar voorwaarden worden opgesteld. Dit zijn de eisen waaraan voldaan moet worden om een specifiek proces waar te kunnen maken. Deze randvoorwaarden geven de grenzen aan waar wij ons binnen dit project aan moeten houden.

Dit zijn onze randvoorwaarden:

* De Product Owner en coach zijn (los van elkaar) minimaal eens per week aanwezig bij overleg en ze zijn tussendoor bereikbaar voor vragen. De afspraken voor een vast bezoek worden gemaakt door de Product Owner en coach.
* Vooraf consensus over de milestones in de sprints.

# 6. Op te leveren producten en kwaliteitseisen

In dit hoofdstuk worden alle op te leveren (deel)producten van het project benoemd met bijbehorende kwaliteitseisen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Product** | **Productkwaliteit eisen** | **Benodigde activiteiten om te komen tot het product** | **ProcesKwaliteit** |
| 1 | Plan van Aanpak | * Bevat alle hoofdstukken die benoemd zijn in Toelichting op PVA[[1]](#footnote-2) | * Eventueel vragen stellen aan de product Owner over het product * Scrum Coach en eventueel professional skills docent vragen voor feedback | Het hele Plan van Aanpak na laten kijken door Product Owner, scrum coach en skills docent voor eventuele verbeteringen na het inlevermoment en bespreken via Teams |
| 2 | Software Guidebook | * Bevat alle verplichte hoofdstukken die benoemd bij het Software Guidebook[[2]](#footnote-3) | * Gebruik maken van Visual studio code plug-in – Markdown Preview Enhanced * Gebruik maken van het C4 Model * Gebruik maken van PlantUML | Feedback via Teams terugkrijgen van de Scrum Coach voor verbeteringen voordat deze naar de Product Owner wordt gestuurd. |
| 3 | Code | * Code is netjes verdeeld over meerdere bestanden * Geen onnodige herhaling van code door functies te hergebruiken * Commentaar in het Engels geschreven * Voldoet aan acceptatiecriteria * Voldoet aan Definition of Done * Wordt geplaatst op Github in een branch die losstaat van de main branch | * Code schrijven * Unit tests (back-end) * Eventueel onderzoek van minder bekende technieken | Ieder pull request wordt door minstens één teamlid gereviewd die niet aan de functie heeft gewerkt en de code voldoet aan de volgende eisen:  Kolom: “Productkwaliteit eisen”  Hoofdstuk 8: “Definition of done”  Hoofdstuk 8:  “Code kwaliteit” |
| 4 | Bouwen van een database | * Er wordt een document gebaseerde database opgesteld aan de hand van MongoDB en Mongoose. | * Opzetten van schema’s waarin beschreven wordt welke data in de database opgeslagen wordt * Maken van een diagram die inzicht geeft hoe de database eruitziet * Gebruik maken van MongoDB/Mongoose * Gebruik maken van Robo3T * Gebruik maken van Postman | Bij elke verandering aan het schema van de database wordt dit gereviewd door minstens één teamlid die niet met het schema gewerkt heeft.  Na elke doorgevoerde wijziging in de schema’s van de database moet dit worden gedocumenteerd  in het Software Guidebook. |
| 5 | Eindproduct webapplicatie | * Voldoet aan de kwaliteitseisen die beschreven zijn in het product: “Code” * Voldoet aan extra functionaliteit webapplicatie * Voldoet aan de basisfunctionaliteiten en een aantal extra functionaliteiten zoals beschreven in [Hoofdstuk 6.1](#_6.1_Productkwaliteit_eisen) aan het eind van de derde sprint | * Overleg met de Product Owner * Voldoet aan code kwaliteit zoals beschrijven in kolom: “Code” | Oplevering en  goedkeuring van  tussenproducten  door  opdrachtgever aan  het einde van  iedere sprint |
| 6 | Vormgeving van de webapplicatie en browser extensie | Er moet rekening gehouden zijn met alle eisen van de opdrachtgever bij het ontwerp | * Wireframes opstellen voor alle onderdelen van de applicatie * Documenteren welke kleuren * en lettertypes gebruikt worden voor het eindproduct | De wireframes zijn gecontroleerd door het team zodat het team een structuur heeft om de webapplicatie op te bouwen voordat het team begint met de front-end. |
| 7 | Testen van applicaties | 100% geslaagde Unit tests in de backend code | * Gebruik maken van Javascript test framework Jest | Test is uitgevoerd door minstens één teamlid die niet aan de functie heeft gewerkt om fouten op te sporen. |

## 6.1 Productkwaliteit eisen - eindproduct

* De mogelijkheid om Pay-walls te omzeilen, wij willen het mogelijk maken voor de gebruikers om de artikelen, waarvoor je moet betalen (lidmaatschap), ook te kunnen delen met gebruikers die hiervoor niet hebben betaald.
* De tags meer gebruikersvriendelijk maken, hiermee kunnen gebruikers beter hun artikelen bewaren door meer functies aan te bieden, zoals hiërarchische tags en de mogelijkheid om tags aan te vinken in plaats van deze allemaal afvinken. Deze feature zal vooral handig zijn voor de actieve gebruikers van de app die erg veel artikelen opslaan.
* Ondersteuning bevatten voor langere teksten, behalve artikelen opslaan en later te kunnen lezen willen wij hetzelfde resultaat hebben voor bijvoorbeeld boeken.
* Ondersteuning bieden voor Pdf’s, hedendaagse reader apps kunnen alleen HTML-pagina’s mooi tonen. Bij Pdf’s krijg je al snel het probleem dat het formaat niet goed past in kleine reader app schermen, of nog erger, deze Pdf’s kunnen niet eens gelezen worden door de reader app.
* Het vermogen hebben om goede metadata op te halen is ook erg belangrijk, een verbeterpunt voor de onderzoekers zou zijn om op verschillende metadata de artikelen te kunnen zoeken. Behalve op titel & URL zouden wij bijvoorbeeld ook willen zoeken op auteur(s), tijdschrift, onderwerp, datum en andere nuttige metadata.
* Als laatste wil de opdrachtgever uiteraard dat gebruikers overstappen naar onze verbeterde reader app, om dit doel te kunnen behalen willen wij de mogelijkheid hebben om te kunnen communiceren met andere reader apps, dit houdt in dat we onze reader app flexibel genoeg moeten maken om apis van andere reader apps te kunnen toepassen, hierdoor zouden wij bijvoorbeeld archieven van gebruikers kunnen overbrengen naar onze applicatie.

# 7. Ontwikkelmethoden

In dit project heeft de Product Owner de behoefte dat wij de ontwikkelmethode Scrum gebruiken om het product te ontwikkelen.

Scrum is een agile ontwikkelmethode waarin teams in sprints werken van een paar weken. De duur van elk sprint hangt af van elk project en in ons geval is dit twee weken. In elke sprint levert het team een werkend product op zodat de Product Owner op tijd feedback kan geven. De Product Owner kan de eisen van het product veranderen als hij het product een keer heeft gezien en ook later een beslissingen maken op basis van het huidige product. Dit is het sprint review. Zodra de Product Owner feedback op het product heeft gegeven, start er een nieuwe sprint en bouwt het team het product verder op. Het aantal sprints hangt af van hoe groot het project is, maar in ons geval zijn er 3 sprints.

In scrum werken alle betrokken personen binnen het project in bepaalde rollen:

* Het ontwikkelteam is verantwoordelijk om het eindproduct op te leveren.
* De scrum coach stuurt het ontwikkelteam aan zodat het ontwikkelteam goed kan presteren.
* De Product Owner zorgt voor de communicatie tussen de stakeholders en het ontwikkelteam. Hij maakt de laatste beslissingen op het eindproduct en geeft feedback op wat het team heeft opgeleverd.
* De stakeholders zijn vaak alle andere mensen die zijn betrokken bij het product, zoals klanten die het product kopen. Alle feedback die zij hebben, geven zij door aan de Product Owner. In dit project is er geen sprake van stakeholders.

Bij het begin van elk sprint houdt het team een sprintplanning met de Product Owner. Hierin bepalen zij samen het sprintdoel en wat voor product het team op moet leveren. In een sprint houdt het team elke dag een daily standup van 15 tot 20 minuten. In de daily standup bespreekt elk team lid hoe de voortgang ligt en wat elk team lid die dag gaat doen. Daarna gaat het team aan het werk. Aan het eind van elk dag houdt het team ook een daily standdown, waarin elk team lid vertelt wat diegene heeft gedaan op die dag. Vlak voor het eind van elk sprint, houdt het team een sprint review met de Product Owner. Na het ontvangen van alle feedback in de sprint review houdt het team een retrospective met de scrum coach. In de retrospective kijkt het team naar het functioneren tijdens de sprint en zorgt het team samen met de scrum coach ervoor dat het team beter presteert in de volgende sprint. Daarna start de sprintplanning van de volgende sprint, en wordt het proces weer herhaald.

# 8. Projectorganisatie en communicatie

Voor een goed verloop van dit project zijn er enkele regels opgesteld. We gaan ervan uit dat iedereen zich ook aan deze regels houdt. Zo kunnen we een fijne samenwerking creëren en een werkomgeving waar iedereen zich waardevol en gehoord voelt. Er zijn afspraken gemaakt voor de volgende punten:

* De dagindeling
* De aanwezigheid
* De taken
* De communicatie en voor
* De GitHub
* De Definition of Done
* Procesbeschrijving
* Code kwaliteit

## Indeling werkdag

|  |  |
| --- | --- |
| Tijd | Werkzaamheid |
| 09:00 | SCRUM meeting Teams |
| 09:30 | Begin werkdag |
| 12:00 | Begin lunch DND |
| 13:00 | Einde lunch |
| 16:00 | SCRUM meeting Teams |
| 16:30 | Einde werkdag |

## Aanwezigheid

De aanwezigheid is een belangrijk aspect bij het werken in groepsverband. Immers zonder elkaar te zien en spreken werk je eigenlijk voor jezelf. Daarom is het belangrijk een aantal afspraken te maken hierover.

Zo is het verplicht om bij de SCRUM momenten aanwezig te zijn. De momenten worden aan het begin van de werkdagen gebruikt om je groepsgenoten te laten weten wat je vandaag gaat doen, wat je eventueel nodig hebt van andere en waar je bang voor bent om tegenaan te lopen. Aan het eind van de werkdag gebruik je deze om de dag af te sluiten, te laten weten/zien wat vandaag gelukt is en of je ergens nog problemen hebt ondervonden die je morgen met iemand aan wilt pakken.

De werkweek loopt van maandag tot en met vrijdag. Gedurende deze dagen wordt verwacht dat je deze ook aan het project besteed. Natuurlijk zijn we hier vrij open in en is het mogelijk dat er dingen zijn waarvoor je tijd vrij wilt maken. Geef dit alleen wel op tijd aan, zo kunnen je groepsgenoten hier rekening mee houden. Daarom is de afspraak dat je dergelijke momenten 2 dagen van tevoren aangeeft.

Neem je rust. Zorg dat je tijdens de werkdag regelmatig een pauze van je werk neemt, al is het om af en toe wat te drinken te pakken, hierdoor raak je niet overbelast waardoor je aan het einde van de dag niet productief meer bent. Zit je al de hele ochtend met een probleem waar jij en je groepsleden niet uitkomen, ga dan een half uurtje wat anders doen en kom met een fris hoofd terug en plotseling lukt het wel. Vandaag ook is er een Do Not Disturb uur ingepland rond de middag tijd. Laat ieder dan even van zijn rust genieten en vraag alleen iets als je echt niet verder kan. Zo kan iedereen op zijn gemak wat eten en drinken en uitrusten om vervolgens weer goed te kunnen werken.

## Taken

Het is belangrijk dat iedereen zijn taken serieus neemt. Zolang iedereen zijn taken goed oppakt, motiveert dit andere om ook verder te gaan en hun best te blijven doen. Daarom zijn hier een aantal afspraken voor opgesteld.

Neem verantwoordelijkheid op je en wacht niet af tot een ander iets aanbiedt of vraagt. Heb je tijd over of niks te doen, kijk ook zelf of je eventueel iemand zou kunnen helpen met zijn taak of een andere taak op je kan nemen voor de dag.

Wanneer iemand niet verder kan tot jij jouw taak af hebt, is het de bedoeling om deze zo snel mogelijk af te maken. Mocht je met wat anders bezig zijn waar je lang over gedaan hebt en bijna af hebt, kan je dit natuurlijk rustig afmaken. Weet dat het voor je projectpartner fijn is wanneer diegene weer verder kan.

Rond je taken volledig af. Begin niet aan een taak om deze 90% af te maken en vervolgens iets nieuws te beginnen. Zorg dat die laatste 10% ook af komt en laat dit ook weten door middel van je taak afvinken. Zorg ook dat je eventuele deadlines haalt, mocht het een keer niet lukken, laat dit dan op tijd weten aan de rest! Zo houd je het voor jezelf overzichtelijk en weten andere ook wat ze wel en niet kunnen gebruiken. Loop je ergens tegenaan en kom je niet verder, vraag dan om hulp. Mocht er niemand beschikbaar zijn, dan kan je tijdelijk aan je volgende taak werken. Mocht het voorkomen dat we met het hele team niet uit een probleem komen, dan hebben we altijd nog de Product Owner die we kunnen benaderen.

## Communicatie

Communicatie is key wanneer je in groepsverband werkt. Zonder communicatie kom je nergens en is dit project ook niet haalbaar. Vandaar een aantal afspraken over de communicatie.

Laat weten wanneer je het ergens niet mee eens bent. Voorkom dat je gedemotiveerd raakt omdat je iets moet doen wat je eigenlijk niet wilt of dat er een afspraak wordt gemaakt waar je het niet mee eens bent. Wanneer jij niet van je laat horen, kan de rest dit ook niet weten. Samen kunnen we dan kijken naar een oplossing. Tuurlijk kan het voorkomen dat je het een keer niet eens bent met de rest, probeer dan voor jezelf na te gaan waarom je er zo’n probleem mee hebt. Luister naar de onderbouwing van de andere en sta open om van je standpunt af te stappen. Dit is niet hetzelfde als opgeven, dit is de kracht waaruit een goede samenwerking ontstaat.

Kom je ergens niet uit, of weet je niet hoe je aan iets moet beginnen? Vraag op tijd aan de andere projectleden of ze je kunnen helpen. We doen dit met z’n allen en staan ook allemaal voor elkaar klaar. Op deze manier kunnen we allemaal snel weer door en leren van elkaar.

Zorg dat je bereikbaar bent. Je hoeft niet ’s nachts om 12uur op teams berichten te reageren, maar zorg wel dat je niet alleen maar een werkdag mentaliteit opbouwt. Wanneer jouw groepsgenoot overdag niet heeft bereikt wat diegene wilde bereiken en deze persoon gaat in de avonduren nog even verder en heeft een vraag, reageer dan ook en help waar nodig. Mocht je toevallig bij je computer zitten en kan je deze persoon even in 5 a 10 minuutjes helpen, neem deze moeite dan ook. Zo staat een ander ook klaar voor jou.

## GitHub

Voor de commits hebben we de afspraak dat deze in het Engels beschreven worden. Zorg voor een duidelijke titel en beschrijving zodat duidelijk is wat er is gedaan. Commit pas wanneer het stuk ook zeker werkt, probeer ook niet voor elke regel code een nieuwe commit te gebruiken. Zorg ervoor dat je eerst uitgebreid getest hebt voor je commit. Op deze manier beperken we de onoverzichtelijkheid van de commits.

Voor branches hebben we een main branch waarop alleen een werkend product staat. Op de development branch wordt gedurende de week gewerkt. Voor elke usecase kan een nieuwe branch aangemaakt worden. Wanneer je klaar bent met je usecase en deze uitgebreid getest hebt met de laatste pull van de development branch, kan je deze mergen. Zorg dat je iemand als reviewer aanwijst die niet bezig is geweest met jouw usecase. Deze persoon kan de code dan doorlopen en een eventuele wijziging goed of afkeuren. Wanneer de reviewer zijn akkoord heeft gegeven kan je de pull request mergen.

## Definiton of Done

Onze definition of done gaat over drie verschillende onderdelen

### User Story

* Als alle tests van deze user story zijn geslaagd
* Als het project compileert zonder fouten
* Als de user story overeenkomt met de eisen van de Product Owner
* Als alle to do items van de User story zijn voldaan
* Als er wordt aan de acceptatiecriteria voldaan die bij de issue staan geschreven
* Als de software guidebook is bijgewerkt indien nodig
* Als de functionaliteit getest door programmeur die niet actief meegewerkt heeft aan de issue door het handmatig uitvoeren van de test. (code review)

### Sprint

* Als er aan elke Definition of done van de user story’s zijn voldaan
* Als alle to do’s klaar en afgevinkt zijn
* Als de product backlog is bijgewerkt
* Als alle bugs eruit gehaald zijn
* Als de sprint gemarkeerd is als ready voor productie indien deze sprint de laatste is (zie release DoD’s)

### Release

* Als de code compleet is
* Als alle unit en functionele tests gehaald zijn
* Als aan alle acceptatiecriteria is voldaan
* Als alle issues opgelost zijn
* Als het project op de juiste omgeving staat en werkt

## Procesbeschrijving

Na de meeting met de Product Owner bepalen we met de groep de taken. De status van een taak laten we zien aan de hand van de huidige kolom binnen git van deze taak. Welke User Story bij welke taak hoort laten we zien door de verschillende labels die we kunnen toewijzen. Je kan zien wie met welke taak bezig is door een gebruiker toe te wijzen aan deze taak. Daarnaast wordt hiervoor gebruik gemaakt van SCRUM om te bespreken wie wat aan het doen is en waar iemand eventueel op vast loopt. Tijdens de daily standdown kunnen we controleren hoever iedereen met zijn taken is gekomen, dan is ook het moment om te laten weten als je ergens vertraging oploopt.

## Code kwaliteit

Hieronder hebben we een aantal afspraken staan over de code kwaliteit gedurende dit project.

### Waarom code kwaliteit uitmaakt

Allereerst is het goed om te weten waarom code kwaliteit uitmaakt. Als je goede code hebt geschreven zul je zien dat de kwaliteit van je software beter is. Doordat de software kwaliteit beter is zul je zien dat het geschreven programma meer veilig, betrouwbaar en sneller is.

### Wat is nou goede code

Code die goed wordt beschouwd is:

* Werkt functioneel
* Volgt een consistente stijl
* Is makkelijk te volgen en te begrijpen
* Herbruikbaar
* Is goed gedocumenteerd
* Kan getest worden

Om ervoor te zorgen dat de code die geschreven wordt kwalitatief goed in elkaar steekt, hebben we een aantal afspraken over de code gemaakt.

1. Code review. Aangezien er vaak over eigen code makkelijker gedacht wordt, hebben we ervoor gekozen om code te reviewen die een andere projectgenoot heeft geschreven. Dit zorgt voor een kritische frisse blik.
2. Codering standaard gebruiken. Ons standaard ziet er als volgt uit: gebruik zo weinig mogelijk globale data declaraties, duidelijke variabele namen gebruiken, juiste inspringen/uitlijning van de code gebruiken en als laatst correcte error handeling.
3. Documenteer de code in het software guidebook. Mocht er een stuk code bestaan die lastig te begrijpen is, zet je er in commentaar simpel maar duidelijk neer wat er gebeurt. Dit commentaar gebeurt in het Engels.

# 9. Projectplanning

Op dit moment zijn er een aantal data al gepland met betrekking tot afspraken en inleverdata. Deze staan in de volgende tabel beschreven. Afspraken met een \* staan gepland op die datum, maar kunnen mogelijk wijzigen in verband met vakantiedagen en agenda’s van Product Owner en coach.

|  |  |
| --- | --- |
| Wanneer | Wat |
| 13-11-2020 | Inleveren PVA |
| 16-11-2020 11:00 | Sprint 1 start gesprek met Robert |
| 18-11-2020 10:00 | Coach meeting met Sander |
| 25-11-2020 10:00 | Coach meeting met Sander |
| 27-11-2020 12:30 | Sprint 1 eind gesprek met Robert |
| 30-11-2020 11:00\* | Sprint 2 start gesprek met Robert |
| 02-12-2020 10:00\* | Coach meeting met Sander |
| 09-12-2020 10:00\* | Coach meeting met Sander |
| 11-12-2020 12:30\* | Sprint 2 eind gesprek met Robert |
| 14-12-2020 11:00\* | Sprint 3 start gesprek met Robert |
| 16-12-2020 10:00\* | Coach meeting met Sander |
| 23-12-2020 10:00\* | Coach meeting met Sander |
| 25-12-2020 12:30\* | Sprint 3 eind gesprek met Robert |
| 28-12-2020 11:00\* | Sprint 4 start gesprek met Robert |
| 30-12-2020 10:00\* | Coach meeting met Sander |
| 06-12-2020 10:00\* | Coach meeting met Sander |
| 08-01-2020 12:30\* | Sprint 4 eind gesprek met Robert |

# 10. Risico’s

Er kunnen situaties ontstaan waardoor wij de doelstelling van het project niet kunnen halen. Deze situaties nemen wij hier op als risico’s. Wij beschrijven wat wij aan elk risico kunnen doen om dit risico te vermijden. Elk risico proberen we te vermijden met een tegenmaatregel, maar als dit helemaal niet lukt, dan hebben we nog een uitwijkstrategie om dit risico hopelijk alsnog te voorkomen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risico** | **Kans** | **Impact** | **Tegenmaatregel** | **Uitwijkstrategie** |
| Te weinig ervaring in het vakgebied | Middel – Groot | Middel | Scrum spike houden over het vakgebied | Overleg met de Product Owner |
| Geen mogelijkheid om fysiek bij elkaar te komen als team | Groot | Klein | Af en toe de camera aanzetten bij Teams vergaderingen | - |
| Onvoldoende kennis van de Product Owner | Klein | Middel – Groot | Zelf nog een keer onderzoek naar dat specifieke probleem doen | Navragen bij andere teams of docenten |
| Onvoldoende overlegmomenten met Product Owner | Klein | Groot | Mailbericht sturen naar de Product Owner | Product Owner bellen via de telefoon |
| Eén van de teamleden raakt gedemotiveerd / stopt met het project | Klein | Groot | In gesprek gaan met diegene | Overleg met de Product Owner over taakverdeling |

# Bibliografie

AIM, P. (2020). *Hoe kom je tot een goed plan van aanpak en wat moet erin staan?*

Boes, M., Tijsma, L., Theunissen, T., Leer, S., Haenen, P., & Holwerda, R. (2015-2016). *Software guidebook.*

1. Toelichting op PVA (AIM, 2020) [↑](#footnote-ref-2)
2. Software Guidebook (Boes, et al., 2015-2016) [↑](#footnote-ref-3)