



Drone mesh netwerk simulatie

Plan van aanpak

HAN Arnhem

561399

MWJ.Berentsen@student.han.nl

Versie 2

Alten Nederland B.V.

Docent: J. Visch, MSc

Assessor: ir. C.G.R. van Uffelen

M.W.J. Berentsen

19 maart 2019

Inhoudsopgave

1	Inleiding	3
1.1	Leeswijzer	4
2	Achtergrond van het project	5
3	Doelstelling, opdracht en op te leveren resultaten voor het bedrijf en school	6
3.1	Probleem	6
3.2	Doelstelling	6
3.3	Opdracht	7
3.3.1	Casus	7
3.4	Op te leveren resultaten voor het bedrijf en school	8
4	Projectgrenzen	9
4.1	Organisatorische grenzen	9
4.2	Inhoudelijke grenzen	9
5	Randvoorwaarden	11
5.1	Organisatorische randvoorwaarden	11
5.2	Inhoudelijke randvoorwaarden	11
6	Op te leveren producten en kwaliteitseisen	12
7	Ontwikkelmethoden	14
7.1	Rational Unified Process	14
7.1.1	De vier fases van RUP	14
7.1.2	Risicovermindering	15
7.1.3	Templates	15
7.2	De aspecten van scrum	16

8	Projectorganisatie en communicatie	17
8.1	Begeleiders	17
8.2	Beschikbaarheid	17
8.3	Vrije dagen	18
8.4	Overige afspraken	18
9	Planning	19
10	Risico's	21
10.1	Interne risico's	21
10.2	Externe risico's	22
10.3	Afgevangen risico's	22
	Literatuur	23
A	Code guideline	24
A.1	Algemeen	24
A.2	Class, Struct en Enum	24
A.3	Functies	25
A.4	Variabelen	25
A.5	Naamgeving	25
A.6	Commentaar	26
B	Iteratie plan template	27
C	Iteratie assessment template	28
D	Usecase template	30
E	Definition of Done	31

1 — Inleiding

Het volgende verslag betreft het plan van aanpak voor de afstudeerstage van Maurice Berentsen (hierna: student). Dit plan van aanpak is gebaseerd op het document *"Toelichting op PvA 3.0"* (ICA Praktijkbureau, 2018b)

Het beschrijft de aanpak van de student waarin hij omschrijft wat hij gaat onderzoeken, hoe dit uitgevoerd wordt en wat de planning van het onderzoek is.

Drones zijn onbemande luchtvaartuigen die kunnen navigeren door de lucht en daarom toepasbaar om bepaalde taken uit te voeren voor de mens. Alten wil gebruik maken van drones om een netwerk te kunnen creëren over een groot gebied. Daarbij wil Alten dat dit netwerk zichzelf kan herstellen door drones te kunnen herpositioneren bij uitval of een slechte verbinding. De opdracht die uitgevoerd gaat worden in dit project luidt als volgt:

Maak een netwerkmodule voor een drone die het mogelijk maakt om meerdere drones onderling te voorzien van een zelf herstellend meshnetwerk. Maak deze module na in een simulatie die geschikt is om meer dan 100 drones te simuleren waarvan het netwerk zich gelijk gedraagt zodat er verdeelalgoritmes getest kunnen worden.

Dit project heeft een tweedelig doel. Het doel voor de student is het aantonen dat hij competent in zijn vak als embedded software developer en het behalen van zijn bachelor diploma Technische Informatica. Het doel van de organisatie Alten Nederland BV. is het zetten van een eerste stap naar bouwen van een drone netwerk door een prototype van een netwerkmodule voor een drone op te leveren die het meshnetwerk opzet en wanneer nodig de drones kan herverdelen.

Omdat de uitvoerende student geen vergunning heeft om te vliegen met drones gebruikt hij een simulatie. De simulatie heeft als doel om de verdeling van de drones te visualiseren, het gedrag te testen bij uitval of storing in het netwerk en simuleren van de drones zelf.

Er wordt voor de communicatie een fysiek prototype gemaakt. Dit prototype zal bestaan uit microcontrollers die draadloos met elkaar communiceren via een onderling meshnetwerk. Dit prototype wordt niet alleen gemaakt voor het aantonen van de communicatie maar ook voor het extraheren van realistisch communicatie gedrag in de simulatie.

Een samenvatting van het plan is:

- Onderzoek de keuze naar het gebruik van simulatiesoftware, hardware en software meshnetwerk.
- Implementeer een abstracte drone de simulatiesoftware; Bouw een prototype van het mesh netwerk en simuleer deze; Gebruik het netwerk om meerdere drones aan te sturen.

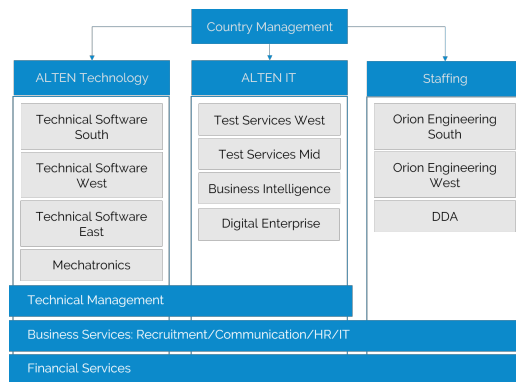
- Lever een onderzoeksrapport op waarin aangetoond wordt dat de simulatie geschikt is voor de wens van Alten om verdeelalgoritmes te testen met meer dan 100 drones.

1.1 Leeswijzer

Dit document gaat eerst in op de [Achtergrond van het project](#) waar het bedrijf kort omschreven wordt en de aanleiding voor dit project. Vervolgens gaat [Hoofdstuk 3](#) in op het doel, de opdracht en het resultaat van het project. Daarna is het project afgebakend in het hoofdstuk [Projectgrenzen](#). Direct daarop zijn de [Randvoorwaarden](#) naar zowel het bedrijf als de school vastgelegd. Hierna is in [Hoofdstuk 6](#) per product, die de student wil opleveren, opgesteld wat de kwaliteitseisen zijn en hoe hij daar aan wil voldoen. [Hoofdstuk 7](#) verduidelijkt welke ontwikkelmethode de student gebruikt tijdens het project. Een praktisch hoofdstuk wat betreft contactgegevens en onderlinge communicatie volgt hierop in [Projectorganisatie en communicatie](#). De [Planning](#) is in het daarop volgende hoofdstuk uitgewerkt. Tenslotte eindigt het plan van aanpak met een risicoanalyse in [Hoofdstuk 10](#).

2 — Achtergrond van het project

Alten Nederland B.V. is een bedrijf met omstreeks 650 medewerkers in dienst. Overkoepe-
lend is Alten Group die in totaal wereldwijd meer dan 25.000 engineers in dienst heeft. De
afdeling oost waar de stage plaats vindt telt ongeveer 50 medewerkers. Dit zijn grotendeels
consulenten die meewerken aan de productontwikkeling van de partners van Alten op het
gebied van technische software engineering. Het overige deel is technisch management en
human resource management.



Figuur 2.1: Organogram Alten IT. Afkomstig van *intranet.alten.nl*, 5 februari 2019

In figuur 2.1 is het organogram van Alten IT zichtbaar. De student voert zijn onderzoek uit binnen het onderdeel Technical Software East.

De omzet van Alten haalt zij uit het delen van kennis door onder andere het detacheren van specialisten, uitvoeren van projecten of het geven van workshops. Daarmee is kennis dus van groot belang voor Alten en is zij altijd bezig deze uit te breiden.

Alten is in het bezit van drones en is bezig haar kennis hierin uitbreiden. Deze kennis breidt zij uit door te investeren in onderzoeken naar dit onderwerp.

Het gebruik van drones neemt toe in Nederland. Volgens het luchtvaartregister in de publicatie van De Inspectie Leefomgeving en Transport (2018) zijn er op 03-10-2018 1806 drones geregistreerd. Daarmee is het aantal drones met een totaal aantal luchtvaart voertuigen van 4445 goed voor 40.6 procent. Op 1 juni 2017 heeft De Jager (2017) van het mediabedrijf Dronewatch een zelfde berekening gedaan waarbij het aantal nog 755 drones was en goed voor 22 procent. Dit is dus een verviervoudiging in iets meer dan een jaar tijd.

Alten springt in op deze snelle groei in toename van drones door zo snel mogelijk meer kennis op te doen naar dit onderwerp. Het is daarom urgent voor Alten om te investeren in onderzoeken naar toepassingen van drones.

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

Het doel van Alten is om onderling verbonden drones te kunnen verdelen om zo een netwerk op te kunnen bouwen over een gebied. Alten wil dat dit netwerk zichzelf kan onderhouden door te reageren op uitval of een slechte verbinding door de één of meerdere drones te herverdelen.

3.1 Probleem

Op dit moment is Alten in het bezit van drones maar deze kunnen niet met elkaar communiceren. Om te kunnen communiceren moet er een netwerkmodule toegevoegd worden aan elke drone. Alten wil grote netwerken kunnen opbouwen die robuust zijn en daarom wil zij gebruik maken van een zelf herstellend meshnetwerk. Hoewel er al meerdere oplossingen bestaan in het gebruik van meshnetwerken wil Alten het zelf herstellend vermogen vergroten. Daarom wil Alten dat de focus van de netwerkmodule ligt op het snel detecteren van uitval van punten in het netwerk zodat daar adequaat op gereageerd kan worden. Adequaat reageren kan op twee manieren volgens Alten. De netwerkmodule kan een ander netwerkpunt zoeken om via dat punt te communiceren, als deze niet beschikbaar is het alternatief om de drones opnieuw te verdelen. Het autonoom fysiek kunnen verplaatsten van de netwerkpunten is dan ook de uitbreiding die Alten wil toevoegen aan het zelf herstellend vermogen. Alleen wat is een efficiënte manier van herverdelen? Het makkelijkste is om alle drones terug naar hun start punt laten vliegen maar dit zou betekenen dat het netwerk op dat moment niet beschikbaar is en het zou ook nog eens onnodig veel stroom verbruiken.

Om algoritmes te testen voor het verdelen van de drones is een simulatie de oplossing. Hierbij is het dus ook van belang dat een simulatieomgeving zich realistisch gedraagt. Om realistisch gedrag na te bootsen moeten de virtuele netwerk modules zich in de simulatie zich net zo gedragen als in het echt.

3.2 Doelstelling

Het doel van dit project is het zetten van de eerste stap in de ontwikkeling van het dronenetwerk. De eerste stap is ontwikkelen van een netwerkmodule voor het onderling verbinden van drones. Het is van belang dat het meshnetwerk van de drones snel kan reageren op uitval van netwerkpunten. Deze netwerkmodule moet zowel virtueel als fysiek gerealiseerd worden in dit project.

Voor het virtueel realiseren van de netwerkmodule moet een simulatie gebruikt worden. Omdat er nog geen simulatiesoftware beschikbaar is wil Alten dat de student uitzoekt welke

simulatiesoftware geschikt is voor het simuleren van meer dan 100 drones. Vervolgens kan de student deze software gebruiken om de virtuele netwerkmodules te testen.

3.3 Opdracht

De opdracht van dit project is de ontwikkeling van meerdere producten die bijdragen aan het behalen van het doel.

De student moet een prototype maken van een netwerkmodule die het onderlinge mesh-netwerk verzorgt en zichzelf kan herstellen bij uitval of slecht signaal. Wanneer er geen alternatieve communicatie route mogelijk is moet de module de drone een instructie uitsenden om zich te herpositioneren.

De student moet een simulatie opzetten waarin de netwerkmodules verplaatst kunnen worden in een vrije ruimte. In de simulatie moet het mogelijk zijn om de netwerkmodules te verplaatsen aan de hand van hun X , Y en Z as zodat er verdeelalgoritmes getest kunnen worden. De netwerkmodules moeten in de simulatie kunnen uitvallen en de onderlinge afstand heeft effect op de virtuele signaalsterkte. De simulatie moet aan de hand van een script herhaalbaar zijn met eenzelfde resultaat.

Alten wil graag dat de simulatiesoftware gebruik maakt van Robot Operating System (ROS) als middleware. Dit wil Alten omdat zij al kennis heeft in ROS maar nog niet in combinatie met drones en mesh netwerken en door het begeleiden hier ervaring in opdoet. Het ervaring opdoen in ROS is geen doel van dit project en de student hoeft zich hier niet op te concentreren.

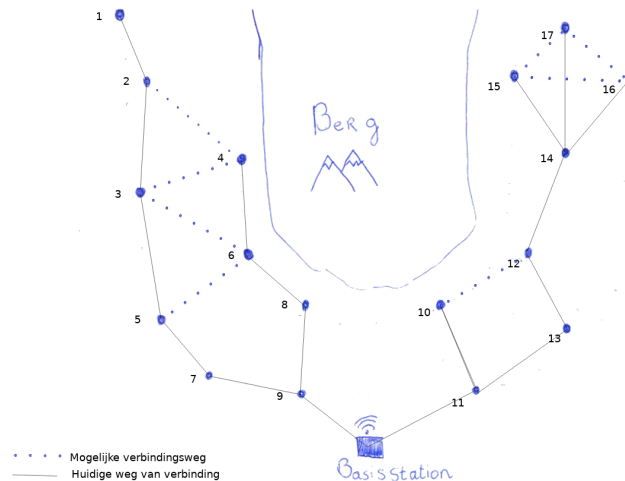
Hoewel de keuze van de simulatiesoftware zich concentreert op het simuleren van een dronenetwerk zal de student alleen de netwerkmodule toevoegen aan de simulatie. De student zal zich dus niet specialiseren in het realistisch simuleren van drones.

Het project is geslaagd als een prototype van het meshnetwerk is opgeleverd samen met een simulatie waarin het mesh netwerk getest kan worden met verschillende verdelingsalgoritmes.

3.3.1 Casus

Om het doel waar naartoe gewerkt moet worden te voorzien van een richting heeft Alten een casus bedacht. Deze casus wordt hieronder toegelicht in een schets en is voorzien van individuele situaties. Er wordt gedrag omschreven voor een simpel algoritme van het herstel van het meshnetwerk om aan te tonen dat dit getest kan worden in de simulatie. In de casus blijft het vlieggedrag buiten de scope.

In de schets van [Figuur 3.1](#) wordt er vanuit gegaan dat elke drone een directe vluchtroute heeft met zijn verbonden netwerkpunt. De verbindingen zijn gemaakt om de casus uit te leggen en volgen dus niet de realistische verbindingen die ze zouden hebben op basis van locatie.



Figuur 3.1: Schets van casus netwerk

In de casus komen de volgende situaties apart voor:

- Punt 2 valt uit waardoor punt 1 geen verbinding meer heeft met het basisstation. Het verwachte resultaat is dat drone 1 zich verplaatst naar de positie van drone 2 en zo het netwerk herstelt.
- De bovenstaande situatie gebeurt. Tijdens het verplaatsen van punt 1 herstelt punt 2 zich weer. Het verwachte resultaat is dat punt 1 weer terug keert naar zijn oude positie.
- Punt 3 valt uit waardoor punt 1 en 2 geen verbinding meer hebben met het basisstation. Het verwachte resultaat is dat drone 2 een nieuwe verbinding aangaat met drone 4.
- Punt 14 valt uit waardoor punt 15, 16 en 17 geen verbinding meer hebben met het basisstation. Het verwachte resultaat is dat de drones 15, 16 en 17 eerst met elkaar verbinden, vervolgens zal één van de drones zich verplaatsen naar punt 14.

3.4 Op te leveren resultaten voor het bedrijf en school

Aan het einde van het project levert de student de volgende producten op aan het bedrijf:

- Simulatiesoftware met netwerksimulatie
- Prototype meshnetwerk module.
- Broncode van de bovenstaande producten.
- SRS (Software Requirement Specification)
- SDD (Software Design Document)
- Onderzoeksverslag

Aan het einde van het project levert de student het bovenstaande plus een projectverslag op aan de Hogeschool van Arnhem en Nijmegen.

4 — Projectgrenzen

Om duidelijkheid te scheppen in dit project zullen hier de grenzen van het project besproken worden. Dit is om voor beide partijen duidelijkheid te creëren welke zaken er niet uitgevoerd gaan worden.

Deze grenzen zijn onderverdeelt in:

- Organisatorische grenzen.
- Inhoudelijke grenzen.

4.1 Organisatorische grenzen

Hier staan alle grenzen die organisatorische invloed hebben.

- Het project start op 1 februari 2019.
- Het project stopt op 28 juni 2019.
- De student werkt 5 dagen per week aan het project wat een totaal van 40 uur per week geeft.
- Het project wordt door één student uitgevoerd.
- De locatie waar de student werkt is vrij waarbij de werkplek van Alten voorkeur heeft.
- Het schoolritme dicteert gedurende het project, ofwel tijdens de schoolvakanties is de student vrij.

4.2 Inhoudelijke grenzen

Hier staan alle grenzen die inhoudelijk over het project gaan. In hoofdstuk 3 is te vinden welke producten worden opgeleverd. Daarnaast zullen ook de volgende punten onder de inhoudelijke project grenzen vallen.

- In het project wordt niet met fysieke drones gevlogen.
- Als middleware voor de simulatie moet ROS gebruikt worden op wens van Alten.
- De student zal zich niet verdiepen in het realistisch simuleren van drones.

- Er is geen beschikbaar budget afgesproken met de Alten. De student maakt geen eigen kosten ten behoeve van de uitvoer van het project.
- Versiebeheer voor code maar ook documenten vindt plaats op de repository van het stage aanbiedende bedrijf.
- Na de einddatum en wanneer alle resultaten opgeleverd zijn zal er geen nazorg meer geleverd worden door de afstudeerder aan het project binnen het huidige dienstverband.
- Code wordt geschreven in C++/C omdat dit hoort bij de specialisatie van de student als embedded software developer.
- Alten heeft geen eigen softwareproducten op de markt. Het bedrijf heeft daarom op dit moment ook geen interesse om dit product op de markt te zetten.

5 — Randvoorwaarden

Bij randvoorwaarden geeft de student aan welke zaken er door de opdrachtgever geregeld moeten worden voordat hij aan het project kan beginnen. De student heeft randvoorwaarden opgedeeld in organisatorische en inhoudelijke randvoorwaarden.

5.1 Organisatorische randvoorwaarden

- Beschikbaarheid van de HAN procesbegeleider: Alle werkdagen per mail waarop binnen 72 uur geantwoord wordt, bereidt om naar Alten te reizen op de feedback-momenten.
- Alten stelt vijf dagen per week een werkplek met een internetverbinding beschikbaar.
- Contactgegevens van alle betrokken partijen zijn bekend bij elkaar.
- Begeleiders zijn in het bezit van een GitHub account en maken deze ook bekend aan de student.
- De student krijgt de ruimte om aan zijn verslagen voor school te werken binnen de 40 uur durende werkweek.

5.2 Inhoudelijke randvoorwaarden

- Beschikbaarheid van de bedrijfsbegeleider: Alle werkdagen per mail waarop binnen 48 uur geantwoord wordt, eenmaal per week overleg/reflectie met de student.
- Alten verzorgt een laptop krachtig genoeg voor simulaties voor de student.
- De HAN procesbegeleider beoordeelt documentatie op ISAS binnen het door het systeem opgeven termijn. Zodat de student genoeg tijd heeft om de feedback te verwerken.
- Afstudeerder moet de kans krijgen om tijdens het project de competenties gesteld vanuit de opleiding te kunnen behalen, en aan te kunnen tonen.

6 — Op te leveren producten en kwaliteitseisen

In dit hoofdstuk worden de op te leveren producten, kwaliteitseisen en de uit te voeren activiteiten besproken. Het gaat hier om zowel de producten die aan de opdrachtgever worden opgeleverd, als om de producten die voor school worden opgeleverd. Hierbij worden de resultaten, zoals beschreven in [Hoofdstuk 3](#), nader uitgewerkt. Voor alle producten worden kwaliteitseisen opgesteld, waaraan de producten moeten voldoen. Daarnaast worden ook alle overige activiteiten aangehaald in dit hoofdstuk om te komen tot het product.

Alle documentatie die wordt geschreven voldoet aan de kwaliteitseisen van de ICA controlekaart.

Product	Productkwaliteit eisen	Benodigde activiteiten om te komen tot het product	Proceskwaliteit
Broncode	Komt overeen met SRS en SDD; Alle publieke functies zijn voorzien van Engels commentaar; Kritische functies hebben unit tests; Voldoet aan de code standaard van Appendix A ; De Definition of Done is voldaan	Schrijven code; Opstellen code standaard; Unit testen schrijven; Commentaar in code schrijven.	Wekelijks code reviews; Tussentijdse beoordeling door begeleiders; Het gebruik van code analysetool Cppcheck . Invullen van de Definition of Done
Presentatie / verdediging	Ondersteunt het projectverslag in het aantonen van de vijf beoordelingscriteria; (ICA Praktijkbureau, 2018a) Geeft een indruk van het opgeleverde product; Duurt in totaal niet langer dan 30 minuten.	Presentatie maken; Demonstratie van het product maken.	Oefenen met bedrijfsbegeleiders /collega's.
Plan van aanpak	Omschrijft wat het plan van de student is; Voldoet aan het feedbackformulier projectplan van het ICA Praktijkbureau (2017) .	Verdiepen in het stage aanbiedende bedrijf; Verduidelijken stageopdracht; Het plan schrijven.	Feedback/goedkeuring van bedrijfsbegeleider; Feedback van procesbegeleider en assessor;

Product	Productkwaliteit eisen	Benodigde activiteiten om te komen tot het product	Proceskwaliteit
Projectverslag	Toont de vijf beoordelingscriteria met bijhorende prestatie criterium (ICA Praktijkbureau, 2018a) voldoende aan; Onderbouwt de keuzes van handelen.	Keuzes tijdens het project documenteren; Feedback gesprekken houden; Het schrijven van het verslag	Tussentijdse feedbackrondes; Feedback van de procesbegeleider; Wekelijkse feedbacksessie Alten.
Prototype mesh netwerk	Geschikt om gedrag te extraheren voor de simulatie; Voldoet aan de opgestelde requirements van het dronenetwerk.	Keuze maken in hardware gebruik; Software voor onderzoek schrijven; Onderzoek naar gedrag van het netwerk.	Code reviews; feedback op onderzoek van begeleiding.
Onderzoeksverslag	Bevat een relevante onderzoeksvraag; Geeft antwoord op de onderzoeksvraag; Is gebaseerd op de ICA-onderzoek kaart. (Van Turnhout et al., 2014)	Zowel literatuur- als labonderzoek uitvoeren; Software bouwen voor het bewijzen van het antwoord.	Wekelijks reflecteren met de stagebegeleider op het proces van het onderzoek.
Simulatie software met meerdere drones en netwerk simulatie	Realistisch en valide; Geschikt om de onderzoeksvraag te beantwoorden; Is voorzien van documentatie.	Schrijven van code; Onderzoek doen wanneer de simulatie realistisch en valide is; Gedrag van het netwerk overzetten tot gesimuleerd gedrag; Een drone simuleren.	Code reviews; Terugkoppeling van het product naar Alten.
Software Design Document	Komt overeen met code; Toont alle relevante ontwerpen; Ontwerpkeuzes staan vastgelegd.	Usecases vertalen tot activity diagrammen; Domein model omzetten tot class diagrammen; component diagram opstellen.	Ontwerpkeuzes bespreken met begeleiding; Ontwerp baseren op ontwerp patronen.
Software Requirement Specification	De requirements komen overeen met de wensen van de opdrachtgever; Bevat genoeg informatie voor het opstellen van een SDD.	Eisen van het product vaststellen; usecases opstellen; furps++ opstellen.	Producteisen en usecases terugkoppelen naar de opdrachtgever.

Tabel 6.1: Op te leveren producten met gestelde eisen m.b.t. activiteit en kwaliteit

7 — Ontwikkelmethoden

De genoemde op te leveren producten in het voorgaande hoofdstuk worden gerealiseerd door middel van de ontwikkelmethoden RUP in combinatie met scrum.

Deze ontwikkelmethode geeft de student de mogelijkheid om voldoende onderzoek te doen naar de problemen die deze opdracht met zich mee brengt. Binnen RUP wordt er ook aan het einde van iedere iteratie een retrospectieve en een review gehouden. Tijdens deze retrospectieven en reviews wordt de voortgang van het proces en het product besproken. Hierdoor krijgt de student niet alleen een duidelijk beeld van de staat van het te ontwikkelen product, maar ook van de afgeronde competenties. Hierdoor kan de student vervolgens bijsturen wanneer nodig om de benodigde competenties te halen.

7.1 Rational Unified Process

RUP (Rational Unified Process) is een iteratieve ontwikkelmethode. Dit houdt in dat het project wordt gerealiseerd in verschillende, elkaar opvolgende iteraties en dat ervaringen uit voorgaande iteraties in volgende iteraties worden meegenomen. (Collaris & Dekker, 2011)

7.1.1 De vier fases van RUP

RUP beschrijft vier fasen waarin de nadruk ligt op verschillende disciplines van RUP (Van 't Einde, z. j.). Hieronder de vier fasen, hoelang ze duren en een korte beschrijving.

Inceptionfase

De inceptiefase loopt tot en met 15 maart en duurt 5 weken.

In deze periode gaat de student de scope, doelen, projectgrenzen, risico's en tegenmaatregelen vastleggen. Deze worden vastgelegd in het plan van aanpak. Aan het einde van de inceptie fase is er een analyse gemaakt en zijn de eisen bekend. Ook zijn er ideeën bedacht over hoe het systeem geïmplementeerd moet worden.

Elaborationfase

Deze fase eindigt op 17 mei en duurt 8 weken.

In deze fase worden de bedachten ideeën uit de inceptiefase uitgewerkt. Er worden prototypes gemaakt en onderzoeken gedaan om te kijken of bepaalde dingen mogelijk zijn. Aan het einde van de elaboratie fase is het bekend hoe het systeem wordt gerealiseerd en

bestaat er een stabiele architectuur. Met stabiele architectuur wordt bedoelt dat er een versie bestaat van het systeem, waarbij er door op ontwikkeld kan worden. De elaboration fase is opgedeeld in vier iteraties van twee weken.

Constructionfase

Deze fase duurt 4 weken en is klaar op 14 juni.

In deze fase wordt het systeem gerealiseerd. De constructie fase bestaat uit iteraties van een week. Volgens het onderwijs moet op vrijdag 14 juni alles op ISAS ingeleverd zijn, tot die tijd wordt alles afgerond. Aan het einde van deze fase is het proof of concept aanwezig en voldoet deze aan de gestelde kwaliteitseisen, zoals gesteld in [Hoofdstuk 6](#). Tevens wordt er een onderzoeksrapport opgeleverd waarin de eerder genoemde gestelde vragen beantwoord worden.

Transistionfase

Deze fase loopt van 17 juni tot 27 juni.

In de transitiefase wordt de presentatie en demo voorbereid. In deze fase wordt het systeem gepresenteerd aan de opdrachtgever en school.

7.1.2 Risicovermindering

Aked(2003) beschijft dat RUP een algemene benadering heeft wat betreft risico's. De risico's worden geïdentificeerd en er worden uitwijkstrategieën opgesteld. De risico's worden gebruikt bij het prioriteren van de use cases en de risico's worden meegenomen als doelen voor het beoordelen van een iteratie. Tijdens de inceptionfase worden de grootste risico's beschreven. In de elboratiefase worden risico's afgedekt door middel van proof of concepts.

7.1.3 Templates

Templates helpen met het stroomlijnen van het proces tijdens het project. RUP heeft een ruim aanbod van templates. Niet iedere template is relevant voor dit project. In dit hoofdstuk worden de templates behandeld die de student wil gebruiken in dit project.

De volgende templates worden gebruikt tijdens het project.

Iteratieplan template

De iteratie template is handig te gebruiken, omdat hierin wordt vastgelegd wat de doelen van de iteratie zijn. De iteratie template is uit RUP gehaald.

[Appendix B Iteratie plan template](#)

Iteration Assessment template

Aan het einde van een iteratie geëvalueerd hoe de iteratie is gegaan en wat er verbeterd kan worden. Om dit proces te structureren beschrijft RUP een 'Iteration Assessment template' hier worden actiepunten opgesteld en opgeschreven wat er gehaald is en wat niet.

[Appendix C Iteratie assessment template](#)

Use Case template

Voor use cases gebruiken wordt er een zelfgemaakt template gebruikt. Dit template is gebaseerd op kennis uit het OSM semester en beschrijft voor een use case verschillende zaken zoals de actors, stakeholders en de flow.

[Appendix D Usecase template](#)

7.2 De aspecten van scrum

Net als RUP is scrum een agile framework die op een iteratieve manier te werk gaat. Zo hebben beide ontwikkelmethoden iteraties. Scrum is meer gericht op dagelijkse taken in tegenstelling to RUP waarbij dit niet het geval is. Scrum vereist dat het op te leveren product ten alle tijden potentially shippable is (James, 2016). Dit betekent dat bij scrum het product continu goed geïntegreerd en getest moet zijn bij iedere iteratie.

Omdat dit project door één student wordt uitgevoerd zijn niet alle best practices van scrum van toegevoegde waarde en zijn daarom weggelaten. In de onderstaande [Tabel 7.1](#) worden de aspecten van scrum aangekaart welke gebruikt worden in dit project.

Element	Wat houdt het in	Waarom?
Sprint Review	Presentatie van product en voortgang aan de opdrachtgever & andere belanghebbenden	Om feedback te krijgen van de opdrachtgever en de prioriteiten te controleren
Retrospective	Bespreking van proces. Wat ging goed/fout en wat gaat de student volgende keer beter doen?	Om het proces van het project te verbeteren. Proces heeft invloed op het product.
Product backlog	Lijst met gewenste functionaliteit.	
Sprint backlog	Scrum board met taken bestaande uit de kolommen: Todo - busy - for review - done	Overzicht houden in de voortgang van taken
Burn down chart	Grafiek die weergeeft wat er nog gedaan moet worden.	Toont in een oogopslag hoe een sprint er voor staat en de planning gehaald wordt.
Sprint planning	Een detail planning voor de sprint.	Geeft overzicht en focus over de te behalen doelen van de sprint.
Definition of Done	Een checklist om te controleren of een programmeer taak klaar is.	Voorkomt dat taken vergeten worden.

Tabel 7.1: Overzicht elementen meegenomen uit scrum

8 — Projectorganisatie en communicatie

Het volgende hoofdstuk gaat in op praktische informatie als contactinformatie en vakantiedagen. Daarnaast worden overige afspraken vastgelegd waar in staat hoe er in bepaalde situaties gehandeld moet worden.

8.1 Begeleiders

Gedurende het project zijn er twee docenten van de HAN betrokken en twee medewerkers van Alten Nederland B.V..

Organistatie	Naam	Rol	Contact
Hogeschool van Arnhem en Nijmegen	Chris van Uffelen	Assessor	Chris.vanUffelen@han.nl
Hogeschool van Arnhem en Nijmegen	Jorg Visch	Processbegeleider	Jorg.Visch@han.nl
HAN / Alten	Maurice Berentsen	Afstudeerder	mauriceberentsen@live.nl
Alten Nederland B.V.	Hugo Logmans	Technisch Manager	hugo.logmans@alten.nl
Alten Nederland B.V.	Hugo Heutinck	Bedrijfsbegeleider	hugo.heutinck@alten.nl

Tabel 8.1: Contactinformatie alle betrokkenen bij dit project

8.2 Beschikbaarheid

In [Tabel 8.2](#) staat beschreven wanneer de student aanwezig dient te zijn.

Dag	Aanwezig	Locatie
Maandag t/m Vrijdag	9:00 - 17:00	Linie 544, 7325 DZ Apeldoorn
Zaterdag en Zondag	Vrij	N.V.T.

Tabel 8.2: Beschikbaarheid student gedurende het project

8.3 Vrije dagen

[Tabel 8.3](#) beschrijft welke dagen de student vrij heeft vanwege verplichte vrije dagen of schoolvakanties.

Toelichting	Datum
Voorjaarsvakantie	04-03-2019 - 10-03-2019
Goede Vrijdag	19-04-2019
2e Paasdag	22-04-2019
Meivakantie	29-04-2019 - 05-05-2019
Hemelvaartsdag	30-05-2019
Dag na Hemelvaartsdag	31-05-2019
2e Pinksterdag	10-06-2019

Tabel 8.3: Schoolvakanties en verplichte vrije dagen

Deze afwezigheid door verplichte vrije dagen telt bij elkaar een volledige werkweek op die de student afwezig is. Omdat deze dagen goed zijn verdeelt over de RUP fases hebben ze weinig impact en wordt er niet speciaal op gepland.

8.4 Overige afspraken

Om de student zo goed als mogelijk te kunnen begeleiden zijn de overige afspraken opgesteld in [Tabel 8.4](#).

Onderwerp	Afspraak
Ziekteverzuim	Bij ziekte maakt de student dit voor 09:00 telefonisch bekend aan zijn manager. Bij langdurige ziekte maakt de student dit ook bekend aan alle begeleiders
Langdurige afwezig of ziekte van begeleiding	Bij langdurige afwezigheid maakt de begeleider van de school of het bedrijf dit bekend aan de student per mail. Bij afwezigheid langer dan twee weken moet de organisatie een tijdelijke vervanger aanbieden.
Iteratie start	Iedere iteratie start op maandagochtend om 09:00
Werkplek	Er wordt van de student verwacht dat hij werkt op zijn werkplek bij Alten in Apeldoorn. Op het moment dat dit afwijkt om bijvoorbeeld thuis te werken laat hij dit minimaal een werkdag van tevoren weten aan de stage aanbiedende organisatie
Workshops en terugkomdag	De student is vrij om de terugkomdag en workshops verzorgd door de HAN bij te wonen. Wel laat hij dit tijdig weten aan de stage aanbiedende organisatie.
Code	De student gebruikt in het project C++ als programmeertaal. Dit omdat dit past bij zijn specialisatie als embedded software developer
IDE	De student is vrij in zijn keuze van IDE.

Tabel 8.4: Overige afspraken voor de studenten en begeleiders

9 — Planning

In dit hoofdstuk wordt de planning voor het project behandeld. Het gaat hier over een globale planning waarin projectweken worden gebruikt. De planning blijft globaal omdat de student op dit moment nog niet genoeg details weet van de onderwerpen. Aan het begin van iedere iteratie zal de student de planning gedetailleerder maken door het opstellen van een [Iteratie plan template](#). In totaal telt het project vanaf de eerste week tot aan de inleverdeadline 17 weken. Daarna zijn er nog twee weken waarin de presentatie zal plaats vinden. Bij de week nummers staat een letter welke aangeeft welke RUP fase de week is terug te vinden in [De vier fases van RUP](#).

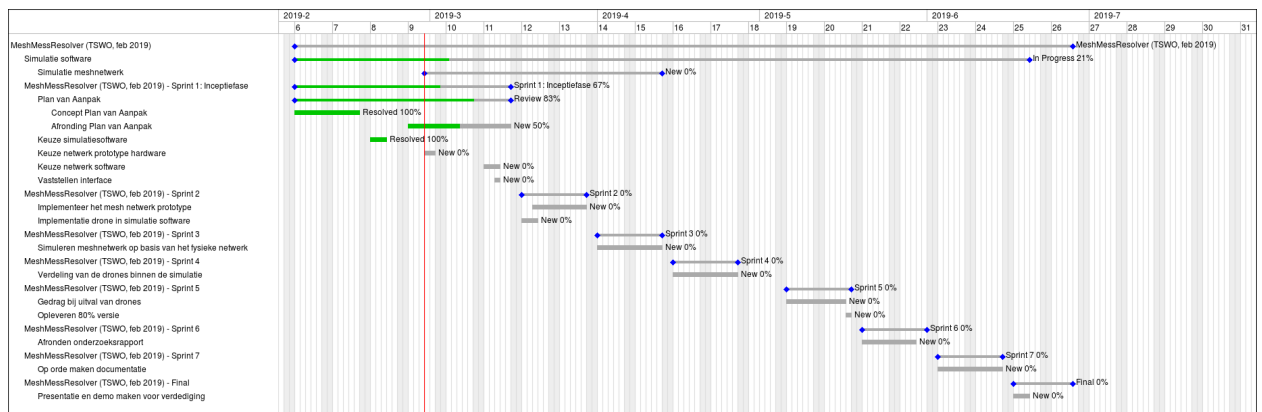
Week nummer	Geplande activiteiten
1 - I-1 4 feb - 8 feb	Opstellen van het plan van aanpak. Defineren onderzoeksvraag. Joinersdag Alten.
2 - I-2 11 feb - 15 feb	15 feb: Inleveren Concept Projectplan. Vaststellen requirements onderzoek. Afronden concept projectplan.
3 - I-3 18 feb - 22 feb	Onderzoek keuze simulatiesoftware. Bijwerken projectplan a.d.h.v. feedback assessor en begeleider.
4 - I-4 25 feb - 1 mrt	Onderzoek keuze netwerk prototype hardware.
5 - I-5 11 mrt - 15 mrt	15 mrt: Inleveren Definitief Projectplan. Onderzoek keuze netwerk prototype. Vaststellen interfaces tussen componenten. Afronden projectplan.
6 - E-1 18 mrt - 22 mrt	Implementatie abstracte drone in simulatie software. Implementatie mesh netwerk.
7 - E-2 25 mrt - 29 mrt	29 maart: sprint review.
8 - E-3 1 apr - 5 apr	Implementatie simulatie mesh netwerk.
9 - E-4 8 apr - 12 apr	12 april sprint review.
10 - E-5 15 apr - 19 apr	19 april: Goede vrijdag. Implementatie gedrag verdeling drones.
11 - E-6 22 apr - 26 apr	22 april: 2e paasdag. 26 april: sprint review.
12 - E-7 6 mei - 10 mei	Onderzoek gedrag mesh netwerk bij slecht signaal. Implementatie gedrag netwerk bij slecht of geen signaal.

Week nummer	Geplande activiteiten
13 - E-8 13 mei - 17 mei	17 mei: Inleveren Definitief 80 procent versie. 17 mei: 29 sprint review. Roosteren afstudeerpresentatie.
14 - C-1 20 mei - 24 mei	Afronden onderzoeksrapport.
15 - C-2 27 mei - 31 mei	29 mei: sprint review 30 mei: hemelvaartsdag 31 mei: dag na hemelvaart
16 - C-3 3 jun - 7 jun	Feedback verwerken op onderzoek. Op orde maken van alle documentatie
17 - C-4 10 jun - 14 jun	10 juni: 2de pinksterdag 14 juni: Inleveren Definitief Eindverslag
18 - T-1 17 jun - 21 jun	Voorbereiden demo en afstudeerpresentatie Verdediging
19 - T-2 24 jun - 28 jun	

Tabel 9.1: globale planning

De bovenstaande planning is uitgewerkt in taken op de project management omgeving redmine. Alten is de aanbieder van deze omgeving. Redmine heeft ondersteuning voor het genereren van een Gantt chart. Deze grafiek geeft goed inzicht in de voortgang van het project en de planning.

Doordat deze omgeving alleen voor medewerkers van Alten beschikbaar is de grafiek gedeeld in dit verslag aan de hand van een geëxporteerde afbeelding uit redmine.



Figuur 9.1: Gantt chart

10 — Risico's

In dit hoofdstuk worden risico's van het project beschreven die niet afgevangen kunnen worden door de student. Er zal een scheiding gemaakt worden tussen risico's die intern kunnen optreden en welke van buitenaf komen. Onderkende risico's die wel afgevangen kunnen worden staan ook genoteerd.

10.1 Interne risico's

Risico	Kans	Impact	Tegenmaatregel	Uitwijkstrategie
Planning voor het onderzoek wordt niet behaald.	Groot	Groot	De planning laten reviewen door de begeleiders of naar haalbaarheid. MoSCoW planning gebruiken voor de requirements. Tijdens het project een Gantt chart gebruiken voor om zo vroeg mogelijk achter dit probleem te komen. Sprints gebruiken i.c.m. een burndown chart	Requirements die in de could planning vallen schrappen.
Simulatiesoftware is niet toereikend voor het doel.	Klein	Groot	Requirements van uit het onderzoek halen waar de simulatiesoftware aan moet voldoen. De Inceptionfase moet dit voorkomen.	Heronderzoeken welke simulatiesoftware wel toereikend is. Wanneer de beschikbare tijd het toestaat overstappen. Anders moeten de requirements opnieuw overwogen worden.

Tabel 10.1: Interne risico's

10.2 Externe risico's

Risico	Kans	Impact	Tegenmaatregel	Uitwijkstrategie
Hardware voor prototype mesh netwerk defect.	Klein	Groot	Hardware gebruiken die snel leverbaar is. Reserve hardware bestellen als het budget het toestaat.	Andere hardware die aan de requirements voldoet gebruiken.

Tabel 10.2: Externe risico's

10.3 Afgevangen risico's

Onderstaande risico's zijn afgevangen aangezien de student tegenmaatregelen heeft getroffen voor de risico's. Als een van onderstaande risico optreedt weet de student wat er van hem verwacht wordt en wordt de impact van het risico geminimaliseerd.

Risico	Kans	Impact	Tegenmaatregel	Uitwijkstrategie
Kwaliteit van op te leveren producten wordt niet gewaarborgd.	Klein	Middel	Kwaliteitseisen opstellen in het plan van aanpak. Daarbij wordt ieder product gereviewed door de bedrijfsbegeleider. Het wordt op een later moment nogmaals gereviewed door de student om te kijken of de code voldoet aan de gestelde eisen.	Als er producten zijn die alsnog niet voldoen aan de kwaliteitseisen zal de bedrijfsbegeleider actie ondernemen. Hierdoor voldoen alle producten aan de gestelde kwaliteitseisen.
Langdurige afwezigheid begeleiding.	Klein	Middel	Na een week van geen gehoor contacteert de student de organisatie.	De organisatie van de begeleider biedt een tijdelijke vervanger aan.
Plan van aanpak onderdeel blijkt niet goed geformuleerd of niet (goed) omvattend te zijn.	Middel	Middel	Laten reviewen door begeleiders en docenten of onderdelen correct worden afgevangen. Daarbij ieder hoofdstuk naast het document <i>Toelichting op PvA 3.0</i> houden en dit document als een checklist gebruiken.	Plan van Aanpak onderdeel aanpassen zodat het onderdeel correct wordt geformuleerd. Dit onderdeel wordt dan ook nogmaals gereviewd en wordt naast de checklist gehouden, zoals beschreven in de tegenmaatregel.

Tabel 10.3: Afgevangen risico's

Literatuur

- Aked, M. (2003, 25 november). *Risk reduction with the rup phase plan*. Op 12 februari 2019 verkregen van <https://www.ibm.com/developerworks/rational/library/1826.html>
- Collaris, R.-A. & Dekker, E. (2011). *Rup op maat- agile met rp, scrum en prince2* (3e dr.). Den Haag, Nederland: Sdu Uitgevers bv.
- De Inspectie Leefomgeving en Transport. (2018, 3 oktober). *2018-12-03 aircraft registrations*. Op 4 februari 2019 verkregen van <https://www.ilent.nl/documenten/publicaties/2018/07/05/luchtvaartuigregister>
- Van 't Einde, W. (z. j.). *Rational unified process (rup)*. Op 7 februari 2019 verkregen van http://www.wimvanteinde.nl/index.php?option=com_content&task=view&id=413
- ICA Praktijkbureau. (2017, 6 juli). *Feedbackformulier projectplan afstuderen 2017-2018 profielenmodel*. Op 4 februari 2019 verkregen van <https://onderwijsonline.han.nl/elearning/lesson/1NXw0pzq>
- ICA Praktijkbureau. (2018a, 16 augustus). *Beoordelingsformulier afstuderen ict; profielenmodel 2018 - 2019*. Op 4 februari 2019 verkregen van <https://onderwijsonline.han.nl/elearning/lesson/1NXw0pzq>
- ICA Praktijkbureau. (2018b, augustus). *Toelichting op PvA 3.0*. Op 4 februari 2019 verkregen van <https://onderwijsonline.han.nl/elearning/lesson/1NXw0pzq>
- De Jager, W. (2017, 1 juni). *Aantal zakelijke drones in nederland in half jaar verdubbeld*. Op 4 februari 2019 verkregen van <https://www.dronewatch.nl/2017/06/01/aantal-zakelijke-drones-in-nederland-in-half-jaar-verdubbeld/>
- James, M. (2016, 21 november). *Scrum effort estimation and story points*. Op 7 februari 2019 verkregen van <http://scrummethodology.com/scrum-effort-estimation-and-story-points/>
- Van Turnhout, K., Craenmehr, S., Holwerda, R., Menijn, M., Zwart, J. & Bakker, R. (2014, september). *De informatieprofessional 3.0*. In (p. 163-174). Den Haag, Nederland: Academic Service BV. Op 11 februari 2019 verkregen van https://www.researchgate.net/publication/272421605_De_methodekaart_praktijkonderzoek

A — Code guideline

In deze bijlage is de code guideline opgesteld, deze is gebaseerd op de google C++ Styleguide ([link](#)) en die van Lockheed Martin ([link](#)).

A.1 Algemeen

- Het bestand waar de main functie in zit mag alleen main.cpp genoemd worden en geen andere functies bevatten.
- Alle source files hebben een header file met uitzondering van main.cpp. Dit zijn .hpp bestanden.
- Headers alleen includeren in het header/source bestand waar het niet zonder kan.
- Zowel commentaar als code zijn in het Engels geschreven.
- Volgorde van includeren: system, library, local. Wordt gescheiden met een witregel.
- Het gebruik van verouderde .h libraries is verboden. Gebruik hiervoor de c++ variant. Bijvoorbeeld <cmath> ipv <math.h>
- Bij if, else, while, do en for instructies altijd accolades gebruiken. Accolade openen op de volgende regel.
- Geen onnodig witregels. Binnen een codeblok is er maximaal één opeenvolgend witregel toegestaan. Daarbuiten zijn er maximaal twee toegestaan.
- Code die niet wordt gebruikt of is uitgecommentarieerd wordt verwijderd.
- Maximale aantal tekens per rij is 80 regels. Uitzonderingen hierop als er gebruik wordt gemaakt van raw String data, deze zullen niet meetellen in het aantal karakters.
- Foutmeldingen alleen in de vorm van exceptions/runtime errors. Het gebruik van return fouten (illegale Int & bool) is verboden.

A.2 Class, Struct en Enum

- Accolades op een nieuwe regel.
- Een class representeert niet meer dan één samenstelling van bij elkaar horende eigenschappen.

- De constructor van een class mag naast het zetten van variabelen geen uitgebreid rekenwerk uitvoeren. Dit wordt gedaan in een aparte functie bijvoorbeeld void init.
- Struct mag voor passieve objecten die alleen data dragen, anders wordt het een class.
- Maak alle waarden en functies Private wanneer dit kan.

A.3 Functies

- Commentaar over wat een functie doet staat in de header. Uitleg over de werking ervan staat in de source file.
- Een functie heeft niet meer dan één doel. Aan de hand van het commentaar is dit doel duidelijk.
- Accolades op een nieuwe regel.
- De scope van een functie is zo klein mogelijk.
- Parameters die niet gewijzigd worden binnen de functie zijn als const aangeduid alleen wanneer er geen sprake is van "pass by value".
- Probeer forward declarations zoveel mogelijk te vermijden.
- Inline functies mogen niet groter zijn dan 1 regel.
- Voor elke functie, die wat berekend of wat retourneerd, worden unittests voor geschreven. Getters en setters hoeven niet.
- Mocht de functie moeten werken in een thread, heeft deze zijn eigen try en catch.

A.4 Variabelen

- Bij een pointer declaratie wordt de asterisk tegenaan de datatype gezet.
- Bij een reference declaratie wordt de ampersand tegenaan de datatype gezet.
- De scope van een variabele is zo klein mogelijk.
- Alle variabelen worden geïnitieerd voor gebruik.

A.5 Naamgeving

- Naamgeving in lowerCamelCase, tenzij expliciet anders vermeld.
- Constante waarden (#define, const declaratie en enum members) in hoofdletters met een laag liggend streepje tussen de woorden.
- Class, struct, enum, union en typedef namen UpperCamelCase.

A.6 Commentaar

- Commentaar wordt geschreven in Doxygen format.
- In de tag @brief wordt beschreven welk doel de class behaald.
- In de tag @author wordt gezet wie de functie heeft gemaakt.
- In de tag @param worden de parameters beschreven van de functie.
- In de tag @return wordt beschreven wat de functie retourneert.
- Commentaar voor een functie hoeft niet in een @brief tag worden beschreven.
- In de beschrijving van de functie een pre en post conditie van de functie benoemen. Bij get en set is dit niet verplicht.
- Commentaar moet enkel gaan over code en niet over ontwerpen of andere externe bronnen.
- Commentaar wordt geschreven boven de functie. Deze is alleen nodig wanneer het toegevoegde waarde heeft.
- Commentaar wordt geschreven boven of naast de variabele. Deze is alleen nodig wanneer het toegevoegde waarde heeft.

B — Iteratie plan template

Iteratie plan template

M.W.J. Berentsen

12 februari 2019

1 Inleiding

Dit Iteratie Plan heeft betrekking op iteratie___ van het project automated mesh drone network simulation van Alten Nederland B.V.. Het geeft een gedetailleerd overzicht van te halen mijlpalen en te realiseren iteratiedoelen.

2 Belangrijke mijlpalen

[Geef hier een gedetailleerd overzicht van de in deze iteratie te halen mijlpalen.]

Mijlpaal	Datum Planning	Eigenaar
[Denk aan af te ronden werkproducten, start van belangrijke activiteiten zoals testen, demomomenten en opleveringen.]	[deadline]	[Naam van de verantwoordelijke]
Start iteratie		
[Keur oplevering iteratie___goed]		
[Keur specificatie Use Case___goed]		
[Oplevering product_____]		
Eind iteratie (evaluatieoverleg)		

3 Iteratiedoelen

[Geef hier een gedetailleerd overzicht van de binnen deze iteratie te halen doelen en te verrichten taken en welke persoon voor de realisatie ervan verantwoordelijk is.]

Doel/taak	Prioriteit	Eigenaar
[Denk aan maken of verder uitwerken van werkproducten,maatregelen tegen risico's en het uitvoeren van ondersteunende taken]	[MoSCoW prioriteit]	[Naam van de verantwoordelijke]
[Bouw Use Case 1: Autorisatie. Alleen Basisscenario en Scenario 2.]	[M]	[Naam Programmeur]
[Bouw Use Case 1: Autorisatie. Scenario 3.]	[S]	[Naam Programmeur]
[Keur specificatie Use Case 2 goed]		
[Aanvullen]		
Plan de volgende iteratie	M	Student

C — Iteratie assessment template

Iteration Assessment template

M.W.J. Berentsen

12 februari 2019

1 Inleiding

[De inleiding van het Iteration Assessment zegt iets over het doel van dit document en waar het betrekking op heeft. Besteed, indien van toepassing, ook aandacht aan het hoofddoel van deze iteratie].

Deze Iteration Assessment heeft betrekking op iteratie ____ van het project automated mesh drone network simulation van Alten Nederland B.V.. Doel van de assessment sessie is te kijken naar wat er in deze iteratie is gebeurd, wat er is bereikt, wat er niet is bereikt en waarom niet, en we ervan kunnen leren. De beoordeling van de iteratie leidt tot een kwalificatie ervan.

2 Deelnemers

[Geef weer welke personen deelnemen aan de assessment sessie.]

3 Belangrijke mijlpalen

[Geef hier een gedetailleerd overzicht van de in deze iteratie geplande mijlpalen.]

Mijlpaal	Behaalde datum	Eigenaar
[Denk aan af te ronden werkproducten, start van belangrijke activiteiten zoals testen, demomomenten en opleveringen.]	[deadline]	[Indicatie ja of nee]
Start iteratie		
[Keur oplevering iteratie ____goed]		
[Keur specificatie Use Case ____goed]		
[Oplevering product ____]		
Eind iteratie (evaluatieoverleg)		

4 Iteratiedoelen

[Geef hier een gedetailleerd overzicht van de binnen deze iteratie te halen doelen en te verrichten taken en welke persoon voor de realisatie ervan verantwoordelijk is.]

Doel/taak	Prioriteit	Behaald
[Denk aan maken of verder uitwerken van werkproducten, maatregelen tegen risico's en het uitvoeren van ondersteunende taken]	[MoSCoW prioriteit]	[Indicatie ja of nee]
[Bouw Use Case 1: Autorisatie. Alleen Basisscenario en Scenario 2.]	[M]	
[Bouw Use Case 1: Autorisatie. Scenario 3.]	[S]	
[Keur specificatie Use Case 2 goed]		
[Aanvullen]		
Plan de volgende iteratie	M	

5 Wat ging er deze iteratie goed

[Geef hier puntsgewijs de zaken die in positieve zin vermeldenswaard zijn. Denk daarbij ook aan de aanpak en best practices.]

6 Wat is voor verbetering vatbaar

[Geef hier puntsgewijs de zaken die voor verbetering in aanmerking komen. Documenteer de oorzaak en bepaal te nemen tegenmaatregelen of stel ze bij.]

7 Actielijst

[Geef hier de afgesproken acties voor de volgende iteratie(s) aan.]

8 Kwalificatie van de iteratie

[Geef hier de kwalificatie van deze iteratie: uitzonderlijk, gehaald, gehaald met risico, onvoltooid, mislukt, gestopt. Deze kwalificaties hebben de volgende betekenis:]

- **Uitzonderlijk:** de iteratie is vlekkeloos verlopen. Alle milestones en doelen zijn gehaald.
- **Gehaald:** alle Must have milestones en doelen zijn gehaald.
- **Gehaald met risico:** Bijna alle Must have milestones en doelen zijn gehaald en het is mogelijk om door te gaan zonder dat er extra iteraties nodig zijn.
- **Onvoltooid:** Er zijn niet genoeg Must have milestones en doelen gehaald maar de juiste dingen zijn gedaan gedurende de iteratie.
- **Mislukt:** Er zijn weinig Must have milestones en doelen gehaald; bijsturing noodzakelijk.
- **Gestopt:** De iteratie is stopgezet.

D — Usecase template

Usecase template

M.W.J. Berentsen

12 februari 2019

1 Fully-dressed use case description

Use Case: <use case naam>
Purpose: <doel use case>
Description of use case: <wat kan je met de use case doen>
Primary actor: <actor>
Preconditions: <pre conditie>
Postconditions: <post conditie>

2 Basic Flow (Main Success Scenario)

[Vul deze tabel in met de happy flow van het systeem]

Actor action	System responsibility
Actie actor 1	systeem repsonse /actie

3 Alternative Flows

[Geef hier een de alternative flow weer van het systeem. Per alternatieve flow een tabel]

Actor action	System responsibility
Actie actor 1	systeem repsonse /actie

E — **Definition of Done**

Voor elke programmeer taak is een gelijke definition of done opgesteld. Dit is een simpele checklist om de programmeur scherp te houden of het opgeleverde werk ook echt af is. Deze lijst zal in de taken op redmine terug te vinden zijn.

- Documentatie voldoet aan gestelde eisen.
- Ontwerpen voldoen aan gestelde eisen.
- Code getest.
- Code voldoet aan Code style guideline.
- Code gedocumenteerd (Doxygen).
- Code compileert.
- Code doet wat het moet doen.