

2018  
2019



**CORVUS**  
D R O N E S

# Plan van aanpak

BINNENSHUISE NAVIGATIESYSTEEM VOOR AUTONOME  
DRONE IN AGRARISCHE SECTOR

ARSALAN ANWARI: 1702825

**ARSALAN ANWARI: 1702825**

## INHOUDSOPGAVE

i: Inleiding.....	2
ii: Versiebeheer .....	2
1: Organisatorische context.....	3
1.1: Aanleiding en context .....	3
1.1.1: Persoonlijke verantwoordelijkheden .....	3
1.3: Bedrijfscontext.....	4
1.3.1: Bedrijfsgegevens .....	4
1.3.2: Stakeholders .....	5
1.4: Interne project relatie .....	5
2: Theoretisch kader.....	6
2.1: Literatuuronderzoek .....	6
2.2: Informatie systemen .....	6
3: Opdracht beschrijving.....	7
3.1: Probleemstelling.....	8
3.2: Doelstelling en eindresultaat .....	8
3.3: Hoofd en deelvragen.....	8
3.3.1: Hoofdvraag .....	10
3.3.2: Deelvragen .....	11
3.4: Project methode .....	12
4: Kwaliteitscontext .....	13
4.1: Standaarden en Tools.....	15
4.1.1: Tools .....	16
4.2: Test methode.....	16
5: Randvoorwaarden.....	17
6: Planning en organisatie .....	18
6.1: Projectactiviteiten.....	18
6.2: Prioriteiten en classificatie.....	20
6.2.1: Classificatie .....	22
6.3: Risicoanalyse.....	23
7: Communicatie.....	24
8: Literatuurlijst.....	24
9: Bijlages .....	25

## I: INLEIDING

Heb je wel eens een video gezien van een robot die navigeert binnen een krappe ruimte en jezelf afgevraagd hoe dit precies mogelijk is? De vraag die hierbij gesteld zal kunnen worden is dan ook: “Hoe zorg je ervoor dat een mobiel autonoom systeem kan navigeren in een binnenshuise omgeving?”. Het nadeel van deze vraag is dat hij te abstract is. Hij beschrijft niet om wat voor soort systeem het gaat, in wat voor omgeving het systeem zich verkeert (in de lucht of op de grond) of de navigatie methode. Deze hoofdvraag zou herschreven kunnen worden voor het doel van deze opdracht als volgt: “Wat is de meest effectieve en efficiënte methode om een navigatie systeem te implementeren voor een autonome drone in een binnenshuise omgeving zoals een glazen kas?”.

Het doel van dit document is dan ook om de hoofdvraag te kunnen beantwoorden en andere benodigde aspecten met betrekking de stageopdracht te kunnen toelichten. Dit document kan daarom ook gebruikt worden (door de HU en stagebegeleider) om de voortgang van het project te kunnen bewaken en de verwachtingen vast te stellen vanuit alle partijen.

In grote lijnen zal ik in de volgende hoofdstukken bespreken wie Corvus Drones is, wat de opdracht precies inhoudt en op welke manier ik de hoofdvraag met bijbehorende deelvragen zal beantwoorden.

## II: VERSIEBEHEER

Versie	Datum	Mutatie
0.1	15-09-2018	Inleiding en opstelling document
0.2	15-09-2018	Mockdocument (document structuur en kopjes)
0.3	20-09-2018	Organisatorische Context
0.4	21-09-2018	Begin Product Beschrijving
0.5	23-09-2018	Product Beschrijving en Theoretisch kader
0.6	24-09-2018	Kwaliteitscontext
0.7	25-09-2018	Randvoorwaarden , begin planning en organisatie
0.8	26-09-2018	Planning en organisatie (beta versie), literatuurlijst en bronnen
0.9	27-09-2018	Conceptversie nog niet gecontroleerd door stagebegeleider
1.0c	05-10-2018	Bruikbare versie gecontroleerd door stagebegeleider

## 1: ORGANISATORISCHE CONTEXT

Corvus Drones is een jonge startup die is opgericht door Gerhold Ten Voorde in 2016. Het bedrijf is zich gevestigd in de campus van de Universiteit Wageningen. Dit technisch bedrijf specialiseert zich in het ontwikkelen van autonome drones in de agrarische sector. Het doel hierbij is dat de drone zich volledig kan gedragen als een medewerker die tussen bepaalde tijden zelfstandig moet opstijgen van het basisstation en hier weer aan het eind van de werkdag naar moet terugkeren. De visie van het bedrijf is dat met de moderne ontwikkelen van robots en drones een autonome drone potentie heeft om de arbeidskosten in de agrarische sector te verlagen. Hierbij zullen ook nog een aantal externe factoren spelen zoals hogere productie omdat je deze unit in massa kunt aanschaffen. Hierbij heeft het bedrijf als missie om een autonome drone systeem te creëren die in zowel binnenshuise als buitenshuise omgevingen kan opereren zonder interventie van de gebruiker.

### 1.1: AANLEIDING EN CONTEXT

Momenteel is er al een prototype ontwikkeld dat al uit zichzelf kan vliegen in een buitenshuise omgeving. De drone kan opstijgen vanaf het basis station en aan de hand van computer visie verschillende analyses uitvoeren zoals het herkennen van onkruid, het lokaliseren van insecten en toestanden van planten controleren. Deze data wordt intern opgeslagen en aan het einde van de werkdag weer getransporteerd in het basisstation. Navigatie gaat via een **flight-controller**[1] en **GPS**[2]. Deze oplossing is zeer geschikt voor buitenshuise omgevingen waar het signaal niet onderdrukt wordt. Maar in een binnenshuise omgeving zoals een glazen kas is dit niet mogelijk. De aanleiding van deze opdracht is dan ook om een nieuw navigatie systeem te ontwikkelen aan de hand van **computer visie**[3] en markers voor positie bepaling. Het doel van dit navigatie systeem is om de doelgroep van Corvus Drones te vergroten naar kwekers in de glastuinbouw en agrariërs in het algemeen. Hiermee zorgt Corvus Drones dat ze sterker in de markt zijn omdat hun oplossing zowel in binnenshuise en buitenshuise omgevingen kan werken.

Als Technische Informatica student aan de Hogeschool Utrecht ben ik begonnen met deze stage op 3 september 2018 en loop ik tot 4 januari 2019 met het project mee. Na een persoonlijke gesprek met de oprichter (Gerhold) werd er afgesproken dat ik zal meehelpen aan het nieuwe navigatie systeem en me vooral bezig zal houden met de opzet fase van dit traject.

---

#### 1.1.1: PERSOONLIJKE VERANTWOORDELIJKHEDEN

Mijn persoonlijke verantwoordelijkheden bij dit project zijn dan ook om ervoor te zorgen dat meerdere camera's kunnen opereren op het systeem (Nvidia Tx1). Zodra dit in geoptimaliseerde en onafhankelijke (resources) manier geïmplementeerd, is het de bedoeling dat de camera's bepaalde markers kunnen herkennen met computer visie. Met deze markers kun je de positie van de drone bepalen wat nodig is voor het navigeren.

### 1.3: BEDRIJFSCONTEXT

Zoals ik al eerder vermeld heb, is Corvus Drones een jonge startup. Dit zorgt ervoor dat de bedrijfscultuur vrij informeel is. Op de werkplek worden zowel praktische als theoretische taken uitgevoerd. Deze zal ik verder toelichten in het organogram (figuur 1)



Het bedrijf creëert in essentie één soort product en dat is het complete autonome drone systeem met basisstation. Dit zorgt ervoor dat de soort klanten die bij het bedrijf zitten voornamelijk individuele ondernemers zijn in de agrarische sector.

#### 1.3.1: BEDRIJFSGEGEVENS

In de tabel hieronder is een lijst met bedrijfsgegevens te vinden.

Bedrijfsnaam	Corvus Drones
Adres	Vijfde Polder 1
Bezoekadres	Vijfde Polder 1, (Kamernummer 024)
Postcode	6708 WC
Plaats	Wageningen
Land	Nederland
Telefoon	+31 6 40 37 97 15
Website	<a href="http://www.corvusdrones.com">http://www.corvusdrones.com</a>

### 1.3.2: STAKEHOLDERS

In de tabel hieronder is een lijst met de stakeholder voor dit project te vinden.

<b>Naam</b>	<b>Functie</b>	<b>Email</b>	<b>Telefoon</b>	<b>Rol</b>
Gerhold ten Voorde	Directeur	<a href="mailto:gerhold@corvusdrones.com">gerhold@corvusdrones.com</a>	06-40379715	Stagebegeleider
Arie Groenenboom	Docent HU	<a href="mailto:arie.groenenboom@hu.nl">arie.groenenboom@hu.nl</a>	06-40180684	Stagecoördinator
Frits Dannenberg	Docent HU	<a href="mailto:frits.dannenberg@hu.nl">frits.dannenberg@hu.nl</a>	088-4813776	Docentbegeleider

### 1.4: INTERNE PROJECT RELATIE

Binnen het bedrijf zijn er een aantal projecten die afhankelijk zijn en verder bouwen op het product dat ik lever. Momenteel is het de bedoeling dat de verschillende camera's op de gemonteerde drone elk individueel gekalibreerd moeten worden. Dit heeft met name betrekking tot de montage fout delta die kan ontstaan bij het monteren van camera's. Deze opdracht is toegewezen aan een medestudent van de HU genaamd Stein Bout. Het is de bedoeling dat hij de kalibratie uitvoert aan de hand van de software die ik lever voor de aansturing van de camera's.

## 2: THEORETISCH KADER

Voor de functionele uitvoering van de opdracht moeten bepaalde factoren onderzocht worden. Het moet namelijk eerst duidelijk worden wat het probleem precies inhoudt en waarom dit een relevant probleem is voor het bedrijf. Dit zal verder verduidelijkt worden in het hoofdstuk [“3.1: Probleemstelling”](#). Vervolgens moet er een duidelijke schets worden gemaakt van de opdracht zelf om te bepalen wat er precies gemaakt moet worden. In het hoofdstuk [“3: opdracht beschrijving”](#) zal dit verder verduidelijkt worden. In het hoofdstuk [“3.2: Doelstelling en resultaat”](#) zal uitgelegd worden wat de doelstelling van de opdracht is en wat ervan verwacht wordt.

Het is ook belangrijk dat de hoofd en deelvragen duidelijk geschetst en beantwoord worden. Dit word gedaan in het hoofdstuk [“3.3: Hoofd en deelvragen”](#). Eventuele kwaliteitscriteria die behoren bij de opdracht zullen verduidelijkt worden in het hoofdstuk: [“4: Kwaliteitscontext”](#)

### 2.1: LITERATUURONDERZOEK

Voor de nodige kennis om de context van de opdracht te kunnen begrijpen, zullen een aantal begrippen eerst uitgelegd moeten worden. In het volgende kopje [“2.2: Informatie systemen”](#) zal ik dieper ingaan op gebruikte software en hardware begrippen en wat deze precies inhouden.

---

### 2.2: INFORMATIE SYSTEMEN

In de vorige hoofdstukken is er besproken dat het navigatie systemen aan de hand van computer visie zal werken op een systeem. In dit volgende kopje zal ik verder verduidelijken welke systemen er precies gebruikt zullen worden en wat deze precies doen.

Ten eerste zullen wij gebruik maken van een software pakket genaamd **OpenCV**[4]. Dit is een allround software pakket dat veel computer visie gerelateerde functionaliteiten al voorgeprogrammeerd heeft. Dit pakket werkt op **C++**[5] en kan ook gebruikt worden om de camera's aan te sturen. C++ is een programmeer taal die voornamelijk wordt gebruikt voor de aansturing van hardware of andere complexe systemen.

Vervolgens zullen we ook gebruik maken van **ArUco**[6]. Dit is een kleiner software pakket dat geïntegreerd kan worden met OpenCV en wordt gebruikt om zogeheten “ArucoMarkers” te detecteren. Deze markers kunnen gebruikt worden om realtime de positie van de camera te bepalen en dienen dus als een hulp middel voor het navigatie systeem.

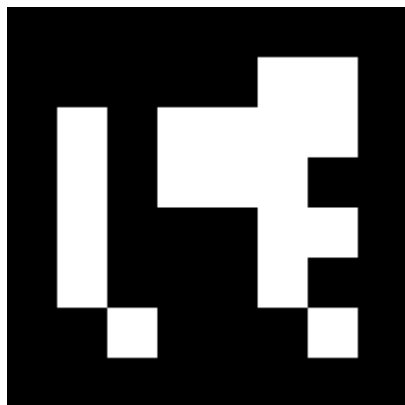
Aan de hardware kant zullen we gebruik maken van de **Nvidia Tx1**[7]. Dit is een volledige computer met ingebouwd geheugen en processor. Een aangepaste versie van **Ubuntu**[8] wordt gebruikt als besturingssysteem voor deze computer. Ubuntu is een **Linux distributie**[9] die veel gebruikt wordt voor Enterprise systemen. Vervolgens zullen we ook gebruik maken van een **Avidia J106 camera module**[10]. Dit is een feite randapparatuur die is geïntegreerd als een module die in geklikt kan worden op de computer. De module bevat 6 camera ingangen waar de camera's aan vast gemaakt kunnen worden.

### 3: OPDRACHT BESCHRIJVING

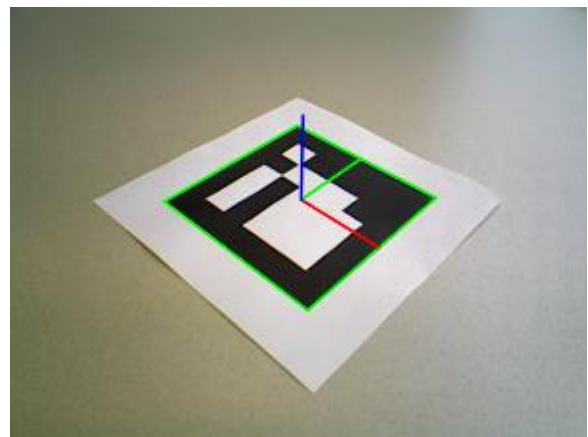
In de organisatorische context had ik al wat besproken over het te maken eindproduct van Corvus Drones. Er moet in grote lijnen een autonome drone ontwikkelt worden die observatieve taken van een werknemer in de agrarische sector moet overnemen. Dit systeem moet uit zich opereren vanaf een basisstation. Momenteel is er al een prototype ontwikkelt dat de eerste fase compleet kan uitvoeren. Dat is het autonoom vliegen vanaf het basisstation in een buitenshuise omgeving door middel van GPS.

GPS is een zeer accurate manier van locatie bepaling in een buitenshuise omgeving omdat er nauwelijks vertraging of vervalsing van het signaal plaatsvindt. Jammer genoeg is het niet mogelijk om gebruik te maken van gps in een binnenshuise omgeving zoals een glazen kas. In de vorige hoofdstukken had ik vermeld dat Corvus Drones als visie heeft om een globaal systeem te creëren dat zowel voor binnenshuise als buitenshuise doeleinden gebruikt kan worden. Uit wat persoonlijke onderzoek waar ik aan heb bijgedragen, is gebleken dat er een aantal methodes zijn om een binnenshuis navigatie systeem te creëren. Een manier was om grote RF zend palen te plaatsen in de beschikbare ruimte om zo constant positie data te versturen naar de drone. Deze methode lijkt in het eerste opzicht het meest efficiënt omdat je met een hogesnelheid data kan versturen naar de drone. Maar voor grootschalige binnenshuise omgeving financieel niet te voorzien. Dit komt voornamelijk door het aantal palen die aangeschaft moeten worden.

Een andere methode die veel gebruikt wordt is om computer visie te gebruiken om de positie tot de camera te kunnen bepalen. Dit kan namelijk met hulp van zogeheten **Aruco Markers**[11] (zie figuur 2a en 2b). Dit zijn speciale markers die erg veel lijken op een QR code. Elke van de marker bevat een unieke code die uitgelezen kan worden. Aan de hand van deze code kunnen belangrijke gegevens zoals de camera oriëntatie, positie en afstand tot de marker bepaald worden.



Figuur 2a: Aruco marker



Figuur 2b: Herkenning van de maker

Nu is het aan mij om software te schrijven die aan de hand van een aantal software pakket de markers kan herkennen en realtime de positie van de drone kan bepalen. Om ervoor te zorgen dat de drone een 360 graden blik heeft, worden er 6 camera's gebruikt die in een hexagonale manier zijn gemonteerd aan de drone.



### 3.1: PROBLEEMSTELLING

Zoals ik al eerder had vermeld is het navigeren in een binnenshuise omgeving een probleem, omdat het huidige systeem alleen werkt met GPS. Momenteel is er nog geen oplossing bedacht om toch met GPS te kunnen navigeren in deze omgevingen vanwege het feit dat dit natuurkundig onmogelijk is. Het is dus belangrijk dat dit probleem op een correcte manier wordt opgelost om te voorkomen dat dit probleem in strijd komt met de visie en missie van het bedrijf.

### 3.2: DOELSTELLING EN EINDRESULTAAT

Het businessdoel van de opdrachtgever is om een autonoom systeem te creëren die in alle omgevingen gebruikt kan worden om zo de product leader te worden op het gebied van autonome drones in de agrarische sector in Nederlands. Hiermee zorgt hij ervoor dat de klant een systeem kan aanschaffen met de garantie dat het voor al zijn doeleinden zal gaan werken.

De doelstelling van de opdracht is om een werkend navigatiesysteem te creëren voor de autonome drone in een binnenshuise omgeving. Dit navigatie systeem moet zijn input krijgen aan de hand van de positiebepaling van de drone. Deze positie wordt bepaald door het herkennen van ArUco markers. Hiermee kan de afstand en positie van de camera tot de marker bepaald worden. In totaal zullen er 6 camera's gebruikt worden om een 360 beeld te creëren. De marker worden herkend aan de hand van computer visie. Het is de bedoeling dat het detecteren van de markers en de aansturing van de camera's allemaal als een individuele taak uitgevoerd wordt. Ik ben dus verantwoordelijk voor deze twee taken en alle andere complicaties die hierbij zullen spelen.

Als eindresultaat moet er een systeem ontwikkeld worden dat autonoom kan vliegen in een binnenshuise en buitenshuise omgeving. Naast het navigeren moet de autonome drone natuurlijk ook zijn analyses kunnen uitvoeren.

### 3.3: HOOFD EN DEELVRAGEN

In de inleiding had ik al kort wat verteld over de potentiële hoofdvraag die gesteld zou moeten worden om dit systeem te kunnen schetsen. Het is hierbij belangrijk dat het duidelijk is om wat voor opdracht het gaat, en welke producten er geleverd moeten worden om de hoofd en deel vraag te kunnen opstellen. De reden dat dit belangrijk is dat de hoofd en deelvragen moeten aansluiten op wat er precies geleverd moet worden. Als er een deelvraag wordt geformuleerd die voor het onderzoek of de ontwikkelingen van het systeem irrelevant is, betekend dat dus dat hij niet goed geformuleerd is.

Laten we daarom eerst even kijken naar het soort opdracht. Je zou kunnen zeggen dat de opdracht in zijn geheel voldoet aan alle drie categorieën (product, onderzoek, ontwerp/advies opdracht) vanwege het feit dat er zowel een product geleverd moet worden, er onderzoek moet worden gedaan wat functioneel kan werken en er gekeken moet worden wat de beste oplossing voor een des betreffend probleem. Om deze reden heb ik dus besloten om de opdracht in zijn geheel op te delen in kleinere segmenten of taken waarbij ik beargumenteer waarom ik dit toewijs een bepaalde categorie.

In de onderstaande tabel kan je een overzicht van deze opdeling zien.

<b>Opdracht taak</b>	<b>Categorie</b>	<b>Reden</b>
Opzet camera's	Onderzoek	De geïntegreerde camera's met J106 module werken op een uniek manier en moeten op een bepaalde manier geconfigureerd worden tijdens de start fase van het project. Het is daarom belangrijk dat er duidelijk is hoe dit precies gedaan moeten worden
Camera's uitlezen	Product / advies	Er moet een stukje software geschreven worden om de camera's uit te kunnen lezen. Het is hierbij wel belangrijk dat er advies wordt gegeven welke methoden het best werken en welke methoden relevant zijn voor de scope en doeleinden van de opdracht gever.
Detecteren van markers	Product	Deze taak is voornamelijk functioneel en betreft het stukje software dat verantwoordelijk is voor het uitlezen van de markers aan de hand van computer visie
Camera's individueel laten werken	Onderwerp / advies	Omdat de camera's individueel zullen gaan werken moet er gebruikt worden gemaakt van een principe genaamd <b>multi-threading</b> [12]. Dit principe zal verder worden uitgelegd in het theoretische kader en bronnenlijst. Om dit goed te laten werken moet er eerst een ontwerp gemaakt worden van hoe de software zal gaan functioneren. Verder moet er ook advies gegeven worden wat de beste methode is om deze hier gebruik van te maken en welke functionaliteiten geïmplementeerd moeten worden om een accuraat systeem te creëren
Positie bepaling	Product / advies	Momenteel is er al een prototype versie van de software die de positie kan bepalen aan de hand van een aantal markers. Deze code moet echter lichtelijk verbeterd en geoptimaliseerd worden. Dit betekent dus dat er vakkundig aan het product gewerkt moet worden en eventueel gezamenlijk advies gegeven moet worden om de huidige software te verbeteren.

Nu het duidelijk is om wat voor soort opdracht het gaat is het belangrijk dat we de producten die geleverd moeten worden om de opdracht te voltooien zullen beschrijven. Deze zijn als volgt:

<b>Product</b>	<b>Uitleg</b>
<i>Camera software module</i>	Het stukje software dat de communicatie tussen het systeem en de camera's hanteert
<i>Marker detectie module</i>	Deze module zal gebruik maken van andere software pakketten in combinatie met de eigen geïmplementeerde functionaliteiten om bruikbare informatie uit de markers te kunnen halen. Denk hierbij aan positie, afstand, oriëntatie en andere navigatie benodigde gegevens
<i>Camera opzet handleiding</i>	De handleiding die beschrijft welke stappen er vooraf genomen moeten worden om te garanderen dat de camera module zal werken met het huidige systeem.
<i>Software handleiding</i>	Een algemene handleiding die beschrijft hoe de zelf geschreven software modules werken en welke functionaliteiten er beschikbaar zijn. Ook moet deze handleiding duidelijk maken hoe de modules met elkaar communiceren en werken.
<i>Multi-threading software module</i>	Dit zelf geschreven software pakket moet het mogelijk maken om individuele taken te maken die parallel naast elkaar uitgevoerd kunnen worden. Dit voornamelijk handig voor het individueel aansturen van de camera en detectie modules
<i>Nvidia Jetson Tx1 Case design</i>	Er moet een 3d tekening worden gemaakt in AutoCAD en Sketchup die gebruikt kan worden om een behuizing waar het systeem en 6 cameras in gemonteerd kan worden. Deze behuizing wordt 3D geprint

### 3.3.1: HOOFDVRAAG

Zowel de project categorieën als de geleverde producten zijn nu verduidelijkt. Aan de hand van deze twee gegevens wordt al snel duidelijk wat de hoofd een deel vragen zijn. Naar aanleiding van deze achtergrondinformatie kan de volgende hoofdvraag geformuleerd worden:

*“Wat is de meest kost effectieve en efficiënte methode om een navigatie systeem te ontwikkelen voor een autonome drone in een binnenshuise omgeving zoals een glazen kas?”.*

### 3.3.2: DEELVRAGEN

De deel vragen kunnen aan de hand van een methode matrix als volgt beschreven worden.

Deelvraag	Type	Verzamel methode	Analysemethode	Resultaat
“Welke navigatiemethodes voor autonome drones in een binnenshuise omgeving bestaan er?”	Beschrijvend	Literatuuronderzoek Deskresearch	Lezen van onderzoeken en scriptie die deze vraag analyseren.	Onderzoeksrapport
“Welk van de onderzochte navigatiemethodes is voor onze scope en budget relevant?”	Vergelijkend Verklarend	Groepsdiscussie	Opstellen van requirements en afstrepen welke methode afvalt.	Groepsbeslissing Implementatie plan
“Welke hardware modules bestaan er om 6 camera's parallel te kunnen ondersteunen voor Nvidia tx1?”	Beschrijvend	Deskresearch	Online bronnen opzoeken. Compatibiliteit gegevens onderzoeken.	Lijst van hardware modules
“Welk van deze hardware modules heeft ondersteuning voor de camera soorten die gebruikt zullen worden?”	Vergelijkend Verklaren	Deskresearch	Gebruikte camera specificaties controleren met datasheet van hardware modules.	Hardware module keuze.
“Welk van de software pakketten op het gebied van computer visie om Aruco marker te kunnen detecteren is voor onze scope relevant?”	Beschrijvend Vergelijkend Verklarend	Literatuuronderzoek Deskresearch Groepsdiscussie	Lezen van documentatie van software pakketten. Online bronnen opzoeken die bepaalde software pakketten aanraden of afraden.	Software pakket keuze
“Wat is de beste methode om meerdere camera's parallel data te laten verzamelen welk vervolgens verwerkt moet worden? ”	Beschrijvend Adviserend	Deskresearch Groepsdiscussie Experimenteel onderzoek	Beschikbare methodes bespreken met groepsleider. Onderzoeken welke methodes parrallel en consistent kunnen werken. Uitproberen wat daadwerkelijk functioneel is.	Concurrency methode Testrapport

### 3.4: PROJECT METHODE

Momenteel werkt Corvus Drones volgens de SCRUM werkmethode. Na een persoonlijk gesprek tijdens de sollicitatie met de oprichter Gerhold ten Voorde, werd al snel duidelijk dat hij van mening was dat dit de beste werkmethode is voor een klein en onafhankelijk team. Het team bij Corvus Drones bestaat uit medewerkers die elk verantwoordelijk zijn voor hun eigen specifieke taken wat onafhankelijk is van het resultaat of eindproduct van een ander. De enige uitzondering hierop is de afhankelijk van een medestudent Stein Bout die afhankelijk is van mijn software om de camera's aan te sturen.

SCRUM is bekend om de effectiviteit bij projecten waar de taken onafhankelijk van elkaar kunnen werken en is daarom naar mijn mening ook de meest geschikte keuze voor een klein team dat onafhankelijk van elkaar werkt.

## 4: KWALITEITSCONTEXT

Het eind product van Corvus Drones is een complete systeem waar alle benodigde hardware en software gebundeld is. Om ervoor te zorgen dat de prestatie van het systeem in zijn geheel niet gehinderd kan worden door een bepaalde module, moeten er een aantal standaarden en module specifiek kwaliteitseisen gehanteerd worden. In het hoofdstuk [“3.3: Hoofd en deelvragen”](#) had ik al kort besproken welke modules er ontwikkeld moeten worden voor de formulering van de deelvragen. In dit hoofdstuk zal ik bij elke module verder beschrijven aan welke kwaliteitseisen deze gebonden zijn.

Ook zal ik het kort gaan hebben over de gebruikte standaarden op software, hardware en documentatie niveau om zo een beter beeld te geven welke richtlijnen het bedrijf volgt. Verder zal ik kort vertellen welke tools (hardware/software) we zullen gebruiken. Tot slot zal ik wat vertellen over de testmethode die de kwaliteitseisen hanteert.

De kwaliteitseisen per module zijn als volgt.

Module	Kwaliteitseisen	Uitleg
Camera software module	Functionele toepasselijkheid Snelheid Koppelbaarheid Foutbestendigheid Modulariteit Vervangbaarheid	<p>De camera module bevat een aantal specifieke taken en doelen die hij moet kunnen voltooien. Een doel zoals het opvragen van frame data zonder vertraging is hier een voorbeeld van.</p> <p>Deze module moet realtime werken, dit betekent dat het binnen een bepaalde maximale tijdsduur zijn taak moet voltooien. Dit betekent dat de snelheid en verwerkingstijd van de camera moet voldoen aan de requirement van onder 5ms.</p> <p>De 6 camera's die onderdeel zijn van deze module moeten volledig parallel van elkaar kunnen werken.</p> <p>Dit betekent dat de mate van data-uitwisseling tussen de camera's en de marker detectie module altijd correct moeten verlopen. Als een camera dreigt te falen betekent dat deze bij voorkeur uitgeschakeld wordt om de veiligheid van de data transmissie te ver waarborgen.</p> <p>Zowel hardware en software fouten kunnen voorkomen bij deze module. Bij hardware fouten zoals een losse connectie moet er gedicteerd worden of de camera</p>

Marker detectie module

	<p>nog beschikbaar is, zo niet dan wordt deze uitgeschakeld. Bij software fouten is de handhaving was soepeler. Hier moet eerst worden gekeken om welke bekende fout het gaat aan de hand van exceptie handeling. Is het een onbekende fout dan wordt de camera ook afgesloten.</p> <p>De module bestaat uit een reeks camera's die software matig door een generieke controller aangestuurd worden. Hierbij zijn er al drivers geschreven voor een reeks veelgebruikte camera sensoren. Het type camera is voor dit software pakket dus niet relevant voor de totale werking van het systeem en kan met een simpele aanpassing gewijzigd worden.</p>
Functionele correctheid Snelheid Koppelbaarheid Herstelbaarheid Testbaarheid	<p>Deze module maakt gebruik van twee software pakketten namelijk OpenCV en ArUco. In beide software pakketten zijn er al bestaande functionaliteiten om de accuraatheid van meet resultaten te kunnen testen. Denk hierbij aan aspecten zoals camera afwijking, beeld vervorming/vervaging en realtime marker analysering. Dit betekend dat fout meetresultaten niet gebruikt zullen worden en herkend kunnen worden.</p> <p>Omdat deze module in feiten actief wordt nadat de camera module data verzend betekend het dat de verwerking van de marker detectie maximaal maar 20ms mag duren om een consistente en snelle frame-rate van ~30 FPS aan te kunnen houden.</p> <p>Het kan soms voorkomen dat tijdens de detectie van een marker de software pakketten een exceptie kunnen geven. De norm hierbij is dat een exceptie onafhankelijk van andere taken in het systeem gehandeld moet worden. In de meeste gevallen moet deze binnen 10ms opgelost worden of uitgeschakeld worden.</p>

Multi-threading software module		Gelukkig kunnen al deze requirements getest worden door software timers te gebruiken die naast het programma werken.
	Functionele compleetheid Volwassenheid Herbruikbaarheid	<p>Deze module werkt volgens het pre-emptive principe en bevat daarom ook alle functionaliteiten die hier normaal gebruikt worden.</p> <p>De module moet zorgvuldig worden getest door verschillende scenario's te creëren in software waar het proces fout zou kunnen gaan. Hierbij geldt dat maximaal 10% van de testen fout mag gaan.</p> <p>De module zelf is zeer herbruikbaar en kan simpelweg met een toevoeging hergebruikt worden voor een heel ander systeem.</p>

## 4.1: STANDAARDEN EN TOOLS

Om ervoor te zorgen dat de software die gemaakt worden herbruikbaar en begrijpelijk is voor iedereen in het team en eventueel nieuwe leden, moet er een bepaalde conventie aangehouden worden. Tijdens het project wordt de standaard **C++ conventie**[13] aangehouden. Hierbij wordt gebruik gemaakt van de **K&R-style**[14]. Tot slot wordt de code ook gedocumenteerd in de **Doxygen standaard**[15].

Op hardware niveau zijn er momenteel geen standaarden die gehanteerd worden vanwege het feit dat dit onmogelijk is om te hanteren. Elk hardware component werkt volgens een andere methode.

Voor de documentatie in het project gelden de volgende standaarden:

- Elke document bevat een inleiding waarin wordt verteld wat het doel van dit document is.
- Elke document moet een inhoudspagina bevatten
- Als er code of commando voorbeelden voorkomen in het document dan moeten deze in een tabel met hun formatering geplaatst worden.



---

#### 4.1.1: TOOLS

Tijdens het project zullen er een aantal tools gebruikt worden. GitLab zal voornamelijk gebruikt worden voor de versiebeheer en uploaden van de software. Gitlab is een **web-based repository manager**[16] waar de voortgang van software overzichtelijk bijgehouden kan worden. De grootste motivator voor het kiezen van deze tool in vergelijking met andere is dat het gratis mogelijk is om de repository privé te houden.

Ook zullen we gebruik maken van een **software IDE**[17] genaamd Netbeans. Met Netbeans is het mogelijk om lokaal op de host computer software te compileren en deze uit te voeren op het systeem.

Tot slot zullen we gebruik maken van trello voor de projectbeheersing en SCRUM werkmethodek. In het hoofdstuk [“6: Planning en Organisatie”](#) zal ik hier wat dieper op ingaan.

#### 4.2: TEST METHODE

In hoofdstuk 4 heb ik aantal requirements weerlegt die horen bij een bepaalde module. Elke van deze requirements waren gebaseerd op software requirements en waarden in orde van tijd. Het meten van tijd is vrij eenvoudig in software applicaties en kan al veel zeggen over de prestatie en accuraatheid van de software requirements. Om te voldoen aan deze requirements worden functionaliteiten van de software module getimed en wordt er gekeken of ze voldoen aan de requirements.

Naast de requirements moeten ook andere onderdelen van het systeem getest worden om te kijken of ze voldoen aan de kwaliteitseisen. Dit wordt gedaan aan de hand van de Unit testing standaard en de Functional Testing standaard. Hier wordt uiteraard een testrapport van gemaakt

## 5: RANDVOORWAARDEN

Tijdens het project zijn er een aantal regels die gelden voor alle teamleden in het algemeen. Ook zijn er een aantal specifieke regels voor nieuwkomers of studenten op een stage die gelden tijdens het project. In dit hoofdstuk zal ik hier wat over vertellen.

Laten we eerst de belangrijkste regel bespreken die geldt voor alle teamleden: *“Ieder teamlid is verplicht om eerlijk te zijn over de voortgang van de toegewezen taak tijdens de standup”*. Omdat teamleden onafhankelijk van elkaar werken is het vaak handig om aan de begin van de werkdag globaal te bespreken welke taken soepel verlopen en welke taken eventueel gezamenlijk gedaan moeten worden. Soms kan het voorkomen dat een ander teamlid wat meer kennis heeft bij een onderwerp dat het probleem veroorzaakt van deze toegewezen taak.

Een andere belangrijke regel voor alle teamleden is: *“In het geval dat meerdere teamleden gebruik moeten maken van een onderdeel van het systeem, moet er met die teamleden gezamenlijk besproken worden welk teamlid op welke datum mag werken aan zijn taak. In het geval dat de taken samengevoegd kunnen worden, moet er besproken worden wie welke nieuwe taak toegewezen krijgt”*. Dit is naar mijn mening een hele goede regel, want tijdens de ontwikkel fase komt het vaak voor dat meerdere teamleden van de zelfde hardware of tool gebruik willen maken tijdens hun werkweek. Door slim te bespreken wie wanneer kan werken of wat samengevoegd kan worden voorkom je dat andere teamleden hoeven te wachten op elkaar.

Voor studenten die een stage volgen, geldt er nog een extra regel: *“Het is de bedoeling dat de student minimaal 2 maal per week werkt aan de benodigde verslagen. Eventueel kan er met Gerhold besproken worden om meer tijd aan het verslag te besteden.”*. Vanwege het feit dat de werkdruk met de te opleveren documentatie aan de HU enorm hoog is, is deze beslissing wel heel fijn. Het zorgt er in feite voor dat je genoeg tijd hebt om de benodigde documentatie te kunnen maken.

## 6: PLANNING EN ORGANISATIE

In dit hoofdstuk zal toelichten wat de werkzaamheden zijn in de stageperiode bij Corvus Drones. Verder zal ik toelichten wanneer deadlines voor bepaalde taken zullen plaatsvinden.

Ook zal ik een risicoanalyse maken voor ongevallen tijdens het project, classificeren wat er opgeleverd moet worden om het project met een voldoende af te sluiten.

### 6.1: PROJECTACTIVITEITEN

In hoofdstuk [“3.4: Project methode”](#) had ik verteld dat CorvusDrones op de SCRUM werkmethode werkt. Dit betekent dat we werken in sprints. Om de twee weken hebben wij een nieuwe sprint van 80 uur. In de onderstaande tabel kan je een overzicht van het werkschema vinden.

Week	Sprint	Activiteiten	Duur
1-2	1	<p>Opzet:</p> <ul style="list-style-type: none"><li>- Maken Trello board urenverantwoording.</li><li>- Maken mockversie plan van aanpak.</li></ul> <p>Onderzoek:</p> <ul style="list-style-type: none"><li>- Documentatie ArUco library lezen.</li><li>- Documentatie J106 module lezen.</li><li>- Documentatie Tx1 lezen.</li></ul> <p>Gesprek:</p> <ul style="list-style-type: none"><li>- Introductie meeting met Gerhold.</li></ul> <p>Functioneel:</p> <ul style="list-style-type: none"><li>- J106 hardware module samenvoegen met Tx1.</li><li>- OpenCV werkend krijgen op Tx1 .</li></ul>	80
3-4	2	<p>Opzet:</p> <ul style="list-style-type: none"><li>- Handleiding schrijven J106 met Tx1.</li><li>- Opstellen concept versie Plan van aanpak en POP.</li></ul> <p>Documentatie:</p> <ul style="list-style-type: none"><li>- Handleiding afmaken Opencv met Tx1.</li><li>- Software ontwerp camera module maken.</li><li>- Software ontwerp multithreading module maken.</li></ul> <p>Gesprek:</p> <ul style="list-style-type: none"><li>- Bespreking werkwijze camera module en multithreading module.</li><li>- Telefoongesprek met docentbegeleider.</li></ul> <p>Functioneel:</p> <ul style="list-style-type: none"><li>- 1<sup>e</sup> versie camera module maken.</li><li>- Beginnen aan multithreading module .</li></ul>	80

<b>5-6</b>	<b>3</b>	<p>Documentatie:</p> <ul style="list-style-type: none"> <li>- <u>DEADLINE: Inleveren PvA en POP.</u></li> <li>- <u>DEADLINE: Handleiding J106 met Tx1 inleveren.</u></li> </ul> <p>Functioneel:</p> <ul style="list-style-type: none"> <li>- 2<sup>e</sup> versie maken camera module.</li> <li>- 1<sup>e</sup> versie multithreading module.</li> </ul> <p>Opzet:</p> <ul style="list-style-type: none"> <li>- Installeren en testen ArUco software pakket.</li> <li>- Ontwerpen van frame voor systeem op papier.</li> </ul>	<b>80</b>
<b>7-8</b>	<b>4</b>	<p>Gesprek:</p> <ul style="list-style-type: none"> <li>- Bespreking 2<sup>e</sup> versie camera module.</li> <li>- Telefoon gesprek met docentbegeleider.</li> </ul> <p>Opzet:</p> <ul style="list-style-type: none"> <li>- Mock versie maken software handleiding.</li> </ul> <p>Documentatie:</p> <ul style="list-style-type: none"> <li>- Camera module verwerken in software handleiding.</li> </ul>	<b>80</b>
<b>9-10</b>	<b>5</b>	<p>Functioneel:</p> <ul style="list-style-type: none"> <li>- <u>DEADLINE: afmaken camera module .</u></li> <li>- 2<sup>e</sup> versie maken multithreading module .</li> </ul> <p>Documentatie:</p> <ul style="list-style-type: none"> <li>- Software ontwerp maken marker detectie module.</li> </ul> <p>Opzet:</p> <ul style="list-style-type: none"> <li>- Mock versie maken Eindverslag.</li> </ul>	<b>80</b>
<b>11-12</b>	<b>6</b>	<p>Functioneel:</p> <ul style="list-style-type: none"> <li>- <u>DEADLINE: afmaken multithreading module.</u></li> <li>- 1<sup>e</sup> versie maken marker detectie module.</li> </ul> <p>Documentatie:</p> <ul style="list-style-type: none"> <li>- <u>DEADLINE: verwerken eindversie camera module in software handleiding.</u></li> <li>- Werken aan eindverslag.</li> </ul> <p>Gesprek:</p> <ul style="list-style-type: none"> <li>- Bespreking definitieve versie camera module.</li> <li>- Telefoongesprek met docentbegeleider.</li> </ul>	<b>80</b>
<b>13-14</b>	<b>7</b>	<p>Functioneel:</p> <ul style="list-style-type: none"> <li>- 2<sup>e</sup> versie maken marker detectie module.</li> <li>- Ontwerp van frame maken in Sketchup.</li> </ul>	<b>80</b>

		Documentatie: <ul style="list-style-type: none"> <li>- Verwerken marker detectie module in software handleiding.</li> <li>- Werken aan eindverslag</li> </ul>	
<b>15-16</b>	8	Functioneel: <ul style="list-style-type: none"> <li>- <u>DEADLINE: afmaken marker detectie module</u></li> <li>- Ontwerp realiseren in 3D printer en eventueel bijwerken.</li> </ul> Documentatie: <ul style="list-style-type: none"> <li>- <u>DEADLINE: verwerken eindversie multithreading module in software handleiding</u></li> <li>- <u>DEADLINE: verwerken eindversie marker detectie module in software handleiding</u></li> <li>- Werken aan eindverslag</li> </ul> Gesprek: <ul style="list-style-type: none"> <li>- Telefoongesprek met docentbegeleider</li> </ul> Opzet: <ul style="list-style-type: none"> <li>- Mockversie maken testrapport</li> </ul>	80
<b>17-18</b>	9	Testen: <ul style="list-style-type: none"> <li>- Testen Camera module</li> <li>- Testen multithreading module</li> <li>- Testen marker detectie module</li> </ul> Documentatie: <ul style="list-style-type: none"> <li>- <u>DEADLINE: camera module verwerken in testrapprt</u></li> <li>- <u>DEADLINE: multithreading module verwerken in test rapport</u></li> <li>- Werken aan eindverslag</li> </ul>	80
<b>19-20</b>	10	Documentatie: <ul style="list-style-type: none"> <li>- DEADLINE: marker detectie module verwerken in testrapport</li> <li>- DEADLINE: inleveren reflectieverslag</li> <li>- DEADLINE: inleveren eindverslag</li> </ul>	80

## 6.2: PRIORITEITEN EN CLASSIFICATIE

Tijdens het project moeten er een tal modules gemaakt worden die worden gebruikt in het navigatiesysteem. Bij elk van deze modules zijn er een aantal prioriteiten die bepalen welke taken relevanter zijn dan andere. In het onderstaande MoSCoW modellen zal ik beschrijven wat de prioriteiten zijn van elke module.

Must have	Should have	Could have
Classen diagram	Globale camera interface	Mogelijk om frame data in verschillende containers op te slaan
Frame data ophalen	Exceptie hantering	
Camera uitzetten		
Frame data verzenden aan andere module		

Figuur 3a: MoSCoW van camera software module

Must have	Should have	Could have
Classen diagram	Mogelijkheid om markers in alle dictionary formaten te herkennen	Mogelijkheid om frame data in verschillende container te kunnen ontvangen
Mogelijkheid om <b>MIP_36H[18]</b> markers te herkennen	Mogelijkheid om alle relevante gegevens van de marker te kunnen bepalen (oriëntatie, rotatie, afstand tot camera, positie, ... )	
Mogelijkheid om positie van marker te bepalen	Exceptie hantering	
Mogelijkheid om frame data te kunnen ontvangen		

Figuur 3b: MoSCoW van marker detectie module

Must have	Should have	Could have
Concurrency diagram	Globale taak interface om nieuwe taken toe te wijzen	<b>Treap logica[19]</b>
STM controller (state hantering)	Mogelijkheid om taken in pause stand te zetten	
Mogelijkheid om taken de starten en stoppen	Data communicatie tussen globale taak interfaces	
Data communicatie tussen gelijkwaardige taken		

Figuur 3c: MoSCoW van concurrency module

### 6.2.1: CLASSIFICATIE

Naast de 3 modules moeten er ook wat documentatie opgeleverde worden aan de HU. Om een overduidelijk overzicht te geven wat voor de HU relevant is en wat voor Corvus Drones, heb ik besloten om dit te verwerken in een tabel. In de onderstaande tabel vind je

Documentatie		
<b>Belanghebbende</b>	<b>Hogeschool Utrecht</b>	<b>Corvus Drones</b>
<i>Plan van Aanpak</i>	✓	
<i>Persoonlijk Ontwikkelplan</i>	✓	
<i>Software Ontwerp</i>		✓
<i>Software handleiding / code documentatie</i>		✓
<i>Testrapport</i>		✓
<i>Maandelijks rapportage</i>	✓	
<i>Stageverslag</i>	✓	
<i>Reflectierapport</i>	✓	
<i>Onderzoeksrapport</i>	✓	

Producten		
<b>Belanghebbende</b>	<b>Hogeschool Utrecht</b>	<b>Corvus Drones</b>
Source code		✓
Software modules		✓
Technische bouwtekening frame (waar systeem in gaat)		✓

## 6.3: RISICOANALYSE

Tijdens het project zullen we een aantal risico's aanwezig zijn die de voortgang kunnen belemmeren. Om dit te voorkomen is het daarom handig om de meest voorkomende risico's te benoemen en te categoriseren in een bepaalde schaal. Dit wordt gedaan in een ordinale schaal; "Laag, Gemiddeld, Hoog". Vervolgens zal ik ook toelichten hoe er gereageerd zou kunnen worden als dit incident voor zou komen. In de onderstaande tabel kan je hier een overzichtelijk beeld bij krijgen.

Risico	Reactie	Mate van risico
<i>Uitval door ziekte of letsel.</i>	Er moet contact worden opgenomen met de opdrachtgever en gezamenlijk bepaald worden of het teamlid nog in staat is om bepaalde taken thuis uit te voeren. In ernstige gevallen kunnen taken ook later ingehaald worden.	Laag
<i>Vertraging in transport</i>	Bij kleine vertraging +- 15min kan het teamlid wachten tot aankomst van het teamlid. Bij ernstigere gevallen (> 30min) zal het team verdergaan zonder het teamlid. In extreme gevallen (> 2 uur) kan het voorkomen dat het teamlid of thuis kan werken of de gemiste uren de zelfde dag moet inhalen	Laag
<i>Behoeftes kunnen veranderen en daarmee ook de eisen.</i>	Dit is afhankelijk van de situatie en behoefte. Er zal op het moment zelf een overweging gemaakt moeten worden of de behoeftes gerealiseerd kunnen worden of niet.	Gemiddeld
<i>Kennis van software blijkt inadequaar.</i>	Wanneer het blijkt dat de kennis onvoldoende is en de planning in gevaar komt, zal er hulp ingeschakeld worden van een expert.	Gemiddeld
<i>Afhankelijkheid van software van de drone en nog in ontwikkeling.</i>	Als het blijkt dat de planning hierdoor in gevaar komt dan zal er iemand moeten bijspringen om de software te schrijven. Daarnaast zou ik zelf kunnen bijspringen om een stuk software te schrijven die zijn rol zal vervullen als stub.	Gemiddeld
<i>Afhankelijkheid van beslissingen van derde partijen.</i>	Als het blijkt dat de planning hierdoor in gevaar komt dan zal er een beslissing gemaakt moeten worden zonder derde partijen.	Gemiddeld
<i>Hardware compatibiliteit met systeem is inadequaar</i>	In het geval dat een hardware component niet kan werken met het systeem moeten er urgent software geschreven om dit te ondersteunen of de zelfde dag nieuwe hardware besteld worden. Dit kan het project zeer vertragen door de levertijd of ontwikkel tijd.	Hoog
<i>Software modules blijken niet correct te functioneren of voldoen niet aan kwaliteitseisen</i>	Het grootste gevaar hierbij is dat dit de prestatie van het systeem kan verminderen. Dit probleem krijgt daarom de hoogste prioriteit.	Hoog



## 7: COMMUNICATIE

De communicatie is nauw en direct. Er wordt gewerkt in een kantoor waar we met zijn alle aan een tafel zitten. Wanneer er over bepaalde zaken vergaderd moet worden kan dat per direct gebeuren. Daarnaast wordt er gebruikt gemaakt van WhatsApp, **Discord**[20] en E-mail.

Communicatie met de stagedocent gebeurt via OnStage, E-Mail of WhatsApp

## 8: LITERATUURLIJST

- [1] Wikipedia contributors. (2018, 24 augustus). Flight controller - Wikipedia. Geraadpleegd op 25 september 2018, van [https://en.wikipedia.org/wiki/Flight\\_controller](https://en.wikipedia.org/wiki/Flight_controller)
- [2] Walla Walla, W. (z.d.). Alles over GPS - wat is gps? Geraadpleegd op 25 september 2018, van <http://www.allesovergps.nl/indexwat.html>
- [3] Wikipedia contributors. (2018, 16 septemberb). Computer vision - Wikipedia. Geraadpleegd op 25 september 2018, van [https://en.wikipedia.org/wiki/Computer\\_vision](https://en.wikipedia.org/wiki/Computer_vision)
- [4] Wikipedia contributors. (2018, 19 septemberc). OpenCV - Wikipedia. Geraadpleegd op 25 september 2018, van <https://en.wikipedia.org/wiki/OpenCV>
- [5] Wikipedia-bijdragers. (2018, 21 september). C++ - Wikipedia. Geraadpleegd op 25 september 2018, van <https://nl.wikipedia.org/wiki/C%2B%2B>
- [6] Salinas, R. M. (z.d.). ARuco: a minimal library for Augmented Reality applications based on OpenCV | Aplicaciones de la Visión Artificial. Geraadpleegd op 25 september 2018, van <https://www.uco.es/investiga/grupos/ava/node/26>
- [7] Nvidia. (2017, 18 december). Jetson TX1 Module. Geraadpleegd op 25 september 2018, van <https://developer.nvidia.com/embedded/buy/jetson-tx1>
- [8] Canonical. (2018, 20 september). The leading operating system for PCs, IoT devices, servers and the cloud | Ubuntu. Geraadpleegd op 25 september 2018, van <https://www.ubuntu.com/>
- [9] Wikipedia-bijdragers. (2018, 5 augustusb). Linuxdistributie - Wikipedia. Geraadpleegd op 25 september 2018, van <https://nl.wikipedia.org/wiki/Linuxdistributie>
- [10] Auvidia. (2017, 1 januari). J106 – Auvideo. Geraadpleegd op 25 september 2018, van <https://auvideo.com/j106/>
- [11] OpenCV. (z.d.). OpenCV: Detection of ArUco Markers. Geraadpleegd op 25 september 2018, van [https://docs.opencv.org/3.1.0/d5/dae/tutorial\\_aruco\\_detection.html](https://docs.opencv.org/3.1.0/d5/dae/tutorial_aruco_detection.html)
- [12] Wikipedia-bijdragers. (2018, 15 junid). Thread (informatica) - Wikipedia. Geraadpleegd op 25 september 2018, van [https://nl.wikipedia.org/wiki/Thread\\_\(informatica\)](https://nl.wikipedia.org/wiki/Thread_(informatica))

- [13] Google. (z.d.). Google C++ Style Guide. Geraadpleegd op 25 september 2018, van <https://google.github.io/styleguide/cppguide.html>
- [14] Wikipedia contributors. (2018, 24 septemberd). Indentation style - Wikipedia. Geraadpleegd op 25 september 2018, van [https://en.wikipedia.org/wiki/Indentation\\_style](https://en.wikipedia.org/wiki/Indentation_style)
- [15] Wikipedia contributors. (2018, 19 septembere). Doxygen - Wikipedia. Geraadpleegd op 25 september 2018, van <https://en.wikipedia.org/wiki/Doxygen>
- [16] Tim OBrien. (z.d.). What is a Repository Manager? Geraadpleegd op 25 september 2018, van <https://blog.sonatype.com/2009/04/what-is-a-repository-manager/>
- [17] Wikipedia-bijdragers. (2018, 21 julic). Integrated development environment - Wikipedia. Geraadpleegd op 25 september 2018, van [https://nl.wikipedia.org/wiki/Integrated\\_development\\_environment](https://nl.wikipedia.org/wiki/Integrated_development_environment)
- [18] JeVois, L. I. (z.d.). Aruco Dictionaries. Geraadpleegd op 25 september 2018, van <http://jevois.org/moddoc/DemoArUco/modinfo.html>
- [19] Wikipedia contributors. (2018, 22 septemberf). Treap - Wikipedia. Geraadpleegd op 25 september 2018, van <https://en.wikipedia.org/wiki/Treap>
- [20] Wikipedia contributors. (2018, 19 septemberg). Discord (software) - Wikipedia. Geraadpleegd op 25 september 2018, van [https://en.wikipedia.org/wiki/Discord\\_\(software\)](https://en.wikipedia.org/wiki/Discord_(software))

## 9: BIJLAGES

- [A] Corvus Drones. (2016). Geraadpleegd op 19 februari 2018, van [www.corvusdrones.com](http://www.corvusdrones.com)
- [B] Hogeschool Utrecht. Bijlage F Format Plan van Aanpak Stageleidraad (2017). Geraadpleegd op 19 februari 2018, van <https://onderwijsteams.sharepoint.hu.nl/fnt/stageictcluster/Gedeelde%20documenten/Documenten%20stage%20schooljaar%202017-2018/Bijlagen%20in%20Word-format%20stage%20schooljaar%202017-2018/Bijlage%20F%20Format%20Plan%20van%20Aanpak%20Stageleidraad%20schooljaar%202017-2018%20v%20170707.docx>
- [C] Trello. Boards | Trello (2018). Geraadpleegd op 20 februari 2018, van <https://trello.com/>
- [D] GitLab. The only single product for the complete DevOps lifecycle - GitLab | GitLab (2011). Geraadpleegd op 21 februari 2018, van <https://about.gitlab.com/>

[E] Onderzoek hbo, Toolkit HBO-i (2018). Geraadpleegd op 26 februari 2018, van [http://onderzoek.hbo-i.nl/index.php/Methoden Toolkit HBO-i](http://onderzoek.hbo-i.nl/index.php/Methoden_Toolkit_HBO-i)