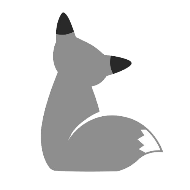
Plan van aanpak

Gibas en het mysterie van de festo drive

Versie 1.0



Student: Matthijs Vos (1651350)

Stagedocent: Jorn Bunk

Stagebegeleider: Sjoerd Radstok

Inhoud

[2 Inleiding 3](#_Toc7097528)

[3 Organisatorische Context 4](#_Toc7097529)

[3.1 Gibas Automation 4](#_Toc7097530)

[3.2 Bedrijfs- en Informatiesystemen 4](#_Toc7097531)

[3.2.1 Universal Robots 4](#_Toc7097532)

[3.2.2 Product Transportation System 4](#_Toc7097533)

[3.2.3 Festo Drive 5](#_Toc7097534)

[3.3 Bedrijfscultuur 5](#_Toc7097535)

[3.4 Taken en Positie van Stagiair 6](#_Toc7097536)

[4 De Opdracht 7](#_Toc7097537)

[4.1 Stageopdracht 7](#_Toc7097538)

[4.2 Businessdoel 7](#_Toc7097539)

[4.3 Doelstelling Opdracht 7](#_Toc7097540)

[4.4 Eindproduct 7](#_Toc7097541)

[4.5 Tussenproducten 7](#_Toc7097542)

[4.5.1 Gemaakte keuzes en onderbouwing 7](#_Toc7097543)

[4.5.2 Solution Architecture 7](#_Toc7097544)

[4.5.3 Vertical Slice 8](#_Toc7097545)

[4.6 Projectmethode 8](#_Toc7097546)

[4.7 Hoofd en Deelvragen 9](#_Toc7097547)

[5 Theoretisch Kader 10](#_Toc7097548)

[5.1 Universal Robots 10](#_Toc7097549)

[5.1.1 PolyScope 10](#_Toc7097550)

[5.1.2 URScript 11](#_Toc7097551)

[5.1.3 URCaps 11](#_Toc7097552)

[5.1.4 URControl 11](#_Toc7097553)

[5.2 Festo Drive 11](#_Toc7097554)

[6 Kwaliteitscriteria 12](#_Toc7097555)

[6.1 ISO25010 12](#_Toc7097556)

[6.1.1 Functionele geschiktheid 12](#_Toc7097557)

[6.1.2 Bruikbaarheid 12](#_Toc7097558)

[6.1.3 Betrouwbaarheid 12](#_Toc7097559)

[6.2 Code Quality standaard 12](#_Toc7097560)

[6.3 Toetsing 12](#_Toc7097561)

[7 Randvoorwaarden 13](#_Toc7097562)

[8 Planning en Organisatie 13](#_Toc7097563)

[8.1 Projectorganisatie 13](#_Toc7097564)

[8.2 Product Back log 13](#_Toc7097565)

[8.2.1 Must Have 14](#_Toc7097566)

[8.2.2 Could Have 14](#_Toc7097567)

[8.2.3 Should Have 14](#_Toc7097568)

[8.2.4 Wont Have 14](#_Toc7097569)

[8.3 Risico’s 14](#_Toc7097570)

[8.4 Ethische Aspecten 14](#_Toc7097571)

[9 Communicatie 15](#_Toc7097572)

[10 Bibliografie 16](#_Toc7097573)

[11 Bijlage 17](#_Toc7097574)

[11.1 Gibas Organigram 17](#_Toc7097575)

# Inleiding

In dit document wordt de stageopdracht bij Gibas besproken. Het doel van dit document is om alle betrokkenen een goed beeld te geven van de geplande uitvoering van de opdracht. Zo worden alle zaken behandeld met betrekking tot de stage en indien nodig vooronderzoek. Het bedrijf zal duidelijk beschreven worden met haar diensten en producten, en de rol van de student binnen het bedrijf. Vervolgens wordt de opdracht gedefinieerd en de benodigde kennis vastgesteld om de opdracht uit te kunnen voeren. Ten slotte komt de planning aan bod; waarin het project wordt opgedeeld over de vastgestelde stageperiode.

# Organisatorische Context

Gibas is een bedrijf dat meerdere dienstenen oplossingen voor bedrijven aanbiedt. Van Draai en freesbanken tot Industriële robots. Daarom is het ook opgedeeld in vier verschillende branches; Numeriek, Automation, Tools en Logistiek. De stageopdracht wordt uitgevoerd bij Gibas Automation. Gibas Automation beschrijft haarzelf als:

De system integrator, gevestigd in Nijkerk, heeft een solide positie in de productie industrie. Bouwend vanuit een jarenlange expertise in de metaalverwerkende sector is een brede know-how ontwikkeld in meerdere branches. Gibas verstevigt zo haar positie als uw leverancier van totaal oplossingen.

Van gerobotiseerde machinebeladingen tot aan assemblage automatiseringen wordt maatwerk equipment vanuit een krachtige engineering geleverd. De interne service organisatie met meer dan 20 technici draagt verder zorg voor een jarenlange gegarandeerde support.

De klant stellen wij centraal en de oplossingen zijn breed gedragen. De integratie met topmerken als Fanuc Robotics, Universal Robots, Siemens ...... maken het geheel betrouwbaar en stabiel. (Gibas Automation, sd)

## Gibas Automation

Gibas Automation biedt verschillende diensten aan, zoals robotoplossingen, machinebeladingen, engineering, speciaal machinebouw en productieautomatisering. Het verspreidt zich dan ook over verschillende branches, van automotive & aerospace tot medical & laboratoria.

## Bedrijfs- en Informatiesystemen

Binnen Gibas komen een aantal systemen voor die betrekking hebben tot de stageopdracht. De Universal Robots(UR) producten worden gebruikt, deze worden in 3.1.1 beschreven. Ook wordt er gebruik gemaakt van URCaps, een systeem dat het mogelijk maakt randapparatuur en accessoires toe te voegen aan een UR product. Communicatie tussen de systemen is een belangrijk aspect in de automatisering, dit wordt vooral afgehandeld door Gibas het eigen Product Transportation System(PTS).

### Universal Robots

Een groot deel van de robotoplossingen van Gibas levert maakt gebruik van Universal Robots. Universal Robots levert robotarmen die toepasbaar zijn op alle productieniveaus met als doel de productiviteit te verhogen, letsel te verminderen en de werksfeer positief te houden. De Universal Robots zijn multi-inzetbaar en ook zeer geschikt voor het beladen van pallets. Vooruitstrevend door haar mobiliteit, de gemakkelijke programmering en interactie tussen mens en robot maakt dat er nieuwe mogelijkheden ontstaan met deze robots.

### Product Transportation System

Het PTS is software ontwikkeld door Gibas, wat het mogelijk maakt via een interface opdrachten in te voeren en deze opdrachten door het programma uit te laten voeren. De software wordt modulair opgebouwd zodat verschillende versies van het programma mogelijk zijn afhankelijk van de wensen van de klant.

Het PTS kan gebruikt worden om met verschillende type robotarmen, mobile robots of opslagsystemen te communiceren. Echter blijft het hoofdprogramma die de opdrachten verwerkt hetzelfde. Het PTS bevindt zich nog in bèta maar zal in de loop van de tijd uitgebreid worden met nieuwe functies.

### Festo Drive

Festo pneumatische actuatoren zijn normaal gesproken cilinders die bedoeld zijn voor lineaire bewegingen. Deze kunnen worden geleverd als standaard cilinder met zuigerstang (ISO 6432, ISO 15552, ISO 21287 of industriële standaard zoals bijvoorbeeld UNITOP) of met/ zonder zuigerstang volgens de specificaties van de fabrikant. De meeste cilinders kunnen eenvoudig worden geïnstalleerd in een bestaande machine met geschikt bevestigingsmateriaal / bevestigingsonderdelen. Ook worden deze geleverd met een motorcontroller en de benodigde documentatie voor de besturing van deze motoren.

## Bedrijfscultuur

Een organisatiecultuur kan beschreven worden met verschillende modellen. Het model van Harrison onderscheidt vier types: de machtsgerichte cultuur, de rollencultuur, de taakcultuur en de personeelscultuur:



Figuur 1 Model van Harrison

Binnen Gibas heerst er een Taakcultuur. Er is sprake van hoge machtsspreiding en een hoge samenwerkingsgraad, dit is terug te zien in het organogram (Bijlage 10.1) van het bedrijf maar ook op de werkvloer. De klanten die aangenomen worden hebben specifieke verzoeken en worden in aparte opdrachten aangenomen. Er zijn meerdere specialisten nodig om een opdracht voor een klant te voltooien. Tevens is er veel contact met de klant nodig om een op maat gemaakte oplossingen te leveren. (Toolshero, sd)

## Taken en Positie van Stagiair

Binnen de organisatie heeft de student een van tevoren gedefinieerde opdracht om te voltooien. Deze opdracht is een uitbreiding op een systeem dat al toegepast wordt op de producten die Gibas levert. De student wordt als software engineer aangenomen onder begeleiding van een software engineer.

# De Opdracht

## Stageopdracht

De stageopdracht is een productopdracht. De opdracht omvat het toevoegen van een festo drive aan de workflow van een Universal Robot(UR) Robotarm. Deze festo drive is in essentie een motor die het mogelijk maakt om een product op zijn platform te bewegen door de UR, zodat de UR niet verplaatst hoeft te worden of rare bewegingen moet maken om bij een bepaald punt te komen. Het UR systeem maakt gebruik van URCaps, een platform om randapparatuur en accessoires toe te voegen. Gibas heeft hier al software geproduceerd alleen vertoond deze een aantal kuren volgens de opdrachtgever.

## Businessdoel

Gibas levert op maat gemaakte oplossingen voor het automatiseren van de productie bij bedrijven. Het resultaat van de opdracht zal meerdere mogelijkheden toevoegen aan het assortiment en diensten van Gibas. De toevoeging van het product is een volgende stap in de automatisering en optimalisering van de UR producten die Gibas levert.

## Doelstelling Opdracht

De opdracht is op te delen in een onderzoek, analyse en productie. Allereerst zal er onderzoek gedaan worden naar de UR, URCaps ,festo drive en de communicatie tussen deze systemen, gevolgd door het analyseren van de geleverde code. Er zal een keuze gemaakt worden of er verder aan de bestaande code gewerkt wordt of dat er een nieuwe codebase wordt opgezet. Vervolgens zal er aan de productie van een systeem gewerkt worden, afhankelijk van het resultaat van het onderzoek en de analyse van de code zal er gewerkt worden aan de bestaande code of aan een geheel nieuwe codebase.

## Eindproduct

Het uiteindelijke eindproduct betreft een werkend systeem dat op een UR product gebruik maakt van de URCaps om een festo drive te kunnen besturen.

## Tussenproducten

Omdat het project een agile methode toepast zal het resulteren in meerdere tussenproducten gedurende het ontwikkeltraject. De volgende tussenproducten zijn vastgesteld om op te leveren tijdens de stage.

### Gemaakte keuzes en onderbouwing

Voordat de daadwerkelijke ontwikkeling van het systeem zal plaatvinden moet eerst het benodigde onderzoek gedaan worden naar de het bestaande systeem en alle randsystemen die er mee communiceren. Dit resulteert in een document met het onderzoek naar de verschillende systemen. Uiteindelijk zal er een onderbouwde keuze voortgebracht worden om of de code uit te breiden, of opnieuw te beginnen.

### Solution Architecture

Een duidelijk solution architecture document creëert een roadmap die helpt met het visualiseren van een systeem en met het elimineren van problemen die je zonder solution architecture pas tijdens het ontwikkelen was tegengekomen.  
Het voorlopige solution architecture document bevat de volgende diagrammen:

* Use Case Diagram

Een Use-Case Diagram modelleert de functionaliteit van een systeem door het gebruik van actoren en Use cases. Use cases zijn een set van acties, services en functies die een systeem kan uitvoeren.

* Activity Diagram

Een Activity Diagram is een flowchart die de flow van een activiteit naar de ander representeert, de activiteit is in essentie een operatie van het systeem.

* Class Diagram

Een Class Diagram beschrijft de structuur van het systeem door de klassen van een system en hun attributen en methodes te laten zien.

* State Transition Diagram

Een State Transition Diagram beschrijft alle states die een object kan hebben, de events die de state van een object kunnen veranderen en de condities waaraan voldaan moet worden voordat de states kunnen veranderen.

### Vertical Slice

Om een makkelijke agile aanpak van het systeem te behalen zullen er vertical slices van het systeem opgeleverd worden voor feedback van de opdrachtgever. Dit is een “bare-bones” versie van het systeem met minimale functionaliteit. Zodra deze is goedgekeurd zullen de functionaliteiten uitgebreid worden tot een daadwerkelijk inzetbaar systeem.

## Projectmethode

Er is gekozen voor een agile aanpak door de lastig te bepalen projectplanning en doordat er veel wijzigingen kunnen komen in de scope. Ook is het waardevol om net werkbare tussenresultaat te kunnen tonen aan de opdrachtgever. Agile software ontwikkeling heeft verschillende kenmerken die van toepassing zijn aan de eisen van zowel het project, de opdrachtgevers en de student (met betrekking tot het persoonlijk ontwikkelplan). Binnen agile development wordt er gebruik gemaakt van iteraties om risico’s te verminderen. Het doel van elke iteratie is om iets bruikbaars op te leveren. Hierdoor is goed vast te stellen met de stagebegeleider of het project correct verloopt en of er mogelijke veranderingen nodig zijn.

De stageperiode wordt opgedeeld in verschillende sprints, waarin elke sprint een afgebakende periode is met vaste onderdelen. Elke sprint duurt een week, en de sprints volgen elkaar steeds direct op. Aan het begin van de sprint is er een sprint planning waarin wordt vastgesteld wat er komende sprint wordt gedaan. Verder is er dagelijks sprake van een Daily Stand-up waarin de stagiaires voorleggen wat er gedaan is, wat er goed ging, wat er beter kan en wat er die dag gedaan wordt. Aan het eind van de sprint zal er een review en een retrospective plaatsvinden waar wordt teruggekeken op de sprint.

## Hoofd en Deelvragen

De hoofdvraag voor deze stageopdracht is als volgt geformuleerd:

Hoe kan door middel van een URCap randapparatuur aangestuurd worden voor het gebruik met een UR systeem **voor Gibas**?

Om deze vraag te kunnen beantwoorden zijn de volgende deelvragen opgesteld :

Is de huidige ontwikkel omgeving geschikt voor het produceren van URCaps?

Over welke functionaliteiten beschikt de URCap API om randapparatuur te besturen?

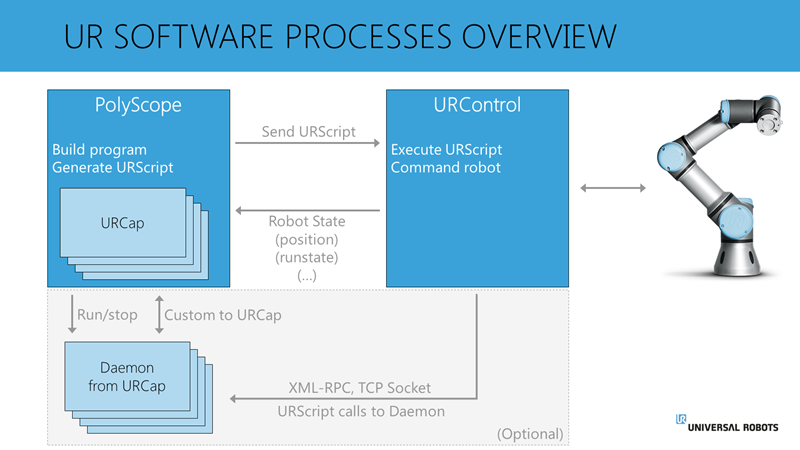
Hoe werkt het communicatieprotocol van het UR systeem met randapparatuur?

**Waar moet een gebruikersinterface van een URCap minimaal aan voldoen?**

# Theoretisch Kader

Het doel van het theoretisch kader, vanuit de stageleidraad genomen, is (voor)onderzoek doen. Het opstellen van een onderzoek om de benodigde theorie, begrippen en concepten te kunnen definiëren en onderbouwen voor de voltooiing van de stageopdracht. Dit wordt gebaseerd op betrouwbare literatuur met correcte literatuurverwijzingen. Ook wordt de belangrijkste literatuur die nog onderzocht moet worden voor de uitwerking en beantwoording van de deelvragen vastgesteld.

Er worden verschillende systemen gebruikt bij de stageopdracht. Om een duidelijk beeld te creëren van het UR-systeem en het onderbouwen hiervan zullen deze omschreven worden. Voor een beter beeld van het UR-systeem is het volgende diagram opgesteld door Universal Robots (Figuur 2).



Figuur 2: UR Software Processes Overview (Universal Robots)

## Universal Robots

Bij de stageopdracht wordt er gewerkt met de systemen van Universal Robots. Het onderzoek van deze systemen benut het ontwikkelproces van een URCap. Een Universal Robot kan op twee niveaus bestuurd worden op GUI niveau, door middel van de PolyScope (of eigen software) en op Script niveau, door middel van URScript.

### PolyScope

PolyScope is de grafische programmeer interface van Universal Robots. Het is de interface die de programmeur gebruikt om een UR-programma te bouwen en besturen. URCaps zijn plug-ins binnen PolyScope. Er zijn momenteel twee versies van PolyScope die geleverd worden door Gibas. Welke versie wordt gebruikt zal geen verschil zijn in de back-end van het systeem. Alleen de user interface is wel anders per versie, er zal tijdens het project onderzocht worden hoe deze verschillen en er zal uiteindelijk een user interface voor beide versies van het systeem komen. Een groot deel van het systeem wordt uitgevoerd in PolyScope, er zal genoeg vooronderzoek gedaan moeten worden naar het optimaal gebruik van deze software. Universal Robots stelt veel hulpmiddelen beschikbaar voor programmeurs , ‘Getting Started’-tutorials en fora voor verdere vragen.

### URScript

Op Script niveau wordt URScript gebruikt om de UR te besturen. URScript is een programmeertaal die gebruik maakt van variabelen, types en flow control statemens zoals “if” en “else”. Verder zijn er ook variabelen en functies beschikbaar voor het monitoren en controleren van de I/O en de bewegingen van de robot. Dit is van toepassing bij het testen het systeem en de communicatie vanuit het UR systeem (Universal Robots, 2018)

### URCaps

Een URCap is een Java-based plugin voor PolyScope. Deze kan gebruikt worden om functionaliteit toe te voegen aan PolyScope, bijvoorbeeld specifieke hardware configuraties of nieuwe templates voor operaties of communicatie. De URCaps komen in het algemeen in vier soorten : Installation Node, Program node, Toolbar en Daemon (Universal Robots)

#### Installation Node

Een Installation Node is een extensie op het installatie domein van het UR systeem. Hierin kan een URCap aanpassingen aan de user interface of extra functionaliteiten toevoegen aan de configuratie van het PolyScope systeem.

#### Program Node

Een Program Node is een extensie op het programmeer domein van het UR systeem. Hier kan de URCap nieuwe programma’s toevoegen aan PolyScope .

#### Toolbar

Een Toolbar is een toevoeging aan de user interface in de vorm van een pop-up menu dat vanuit het gehele programma beschikbaar is. Deze is vaak beschikbaar als er sprake is van randapparatuur als mogelijke manual override die altijd te bereiken is.

#### Daemon

Een Daemon is een extension die er voor zorgt dat er een service in het OS van de UR uitgevoerd kan worden, bijvoorbeeld de communicatie met externe apparatuur. Deze wordt parallel uitgevoerd naast PolyScope en URControl. URScript kan deze Daemon aanroepen met communicatie interfaces zoals XML-RPC of TCP sockets. (Universal Robots, sd)

### URControl

URControl is de real-time controller, deze bestuurt de robot en voert de programma’s uit. In essentie krijgt URControl een programma van PolyScope en zet deze om naar individuele machine instructies. Het is een low-level robot controller op een PC in de UR. Wanneer de UR opstart zal deze als een service(daemon) opstarten en verbinden met PolyScope via TCP/IP connectie (Universal Robots, 2018)

## Festo Drive

Het project moet uiteindelijk een motor van Festo aansturen. Deze worden geleverd met motor controller en een Festo Handling and Positioning Profile(FHPP). In dit FHPP wordt grondig uitgelegd hoe met de motor controller gecommuniceerd wordt. Om de functionaliteiten van de festo drive beter vast te stellen voor een bruikbare interface zal er testsoftware geschreven worden voor het aanroepen van alle benodigde functionaliteit. (Festo)

# Kwaliteitscriteria

Om de kwaliteitscriteria van het project vast te stellen zijn de te gebruiken standaarden, methodieken en tools vastgesteld. Ook zal de methodiek voor het testen van de eindresultaten beschreven worden.

## ISO25010

Een van belangrijkste standaarden voor de kwaliteit van softwareproducten is ISO25010 (ISO 25000, 2019). Deze bevat een uitgebreide lijst met verschillende onderdelen waarop een softwareproduct gescoord kan worden om de kwaliteit hiervan te bepalen. Voor deze opdracht zijn een paar van deze criteria uitgekozen die duidelijk van toepassing zijn. Deze criteria zijn (in willekeurige volgorde):

* Functionele geschiktheid (Functional suitability)
* Bruikbaarheid (Usability)
* Betrouwbaarheid (Reliability)

Dit is een deel van de criteria uit de ISO25010 aangezien alle criteria in verschillende mate of niet van toepassing zijn op dit project. Security en Portability vallen bijvoorbeeld buiten de scope van dit project. Hieronder staat van de gekozen criteria een uitleg waarom dit belangrijk is voor dit project.

### Functionele geschiktheid

Dit is de mate waarin een softwareproduct of computersysteem functies levert die voldoen aan de uitgesproken en veronderstelde behoeften, bij gebruik onder gespecifieerde condities. Het draait hier om compleetheid, correctheid en toepasselijkheid. Omdat het systeem bij voltooiing direct ingezet zal worden door Gibas, speelt de functionele geschiktheid een grote rol. Het systeem zal geleverd worden aan klanten en direct in gebruik genomen worden. (ISO 25000, 2019)

### Bruikbaarheid

De mate waarin een product of systeem gebruikt kan worden door gespecificeerde gebruikers om effectief, efficiënt en naar tevredenheid gespecificeerde doelen te bereiken in een gespecificeerde gebruikscontext. Dit komt vooral naar voren bij de GUI van de URCap. (ISO 25000, 2019)

### Betrouwbaarheid

Aangezien het systeem gelijk toegepast wordt is het van belang dat het systeem erg betrouwbaar werkt. Dit betekent dus dat het zo foutloos mogelijk moet werken en eventuele andere problemen moet kunnen doorgeven. (ISO 25000, 2019)

## Code Quality standaard

De code die geschreven wordt voor deze stage zal naast de bovengenoemde kwaliteitscriteria ook aan een codestandaard moeten voldoen. Binnen het project zal er vooral met Python en Java gewerkt worden. Voor Python zal de codestandaard PEP 8 (Python, sd) aangehouden worden. Voor Java wordt de Google Java Style Guide gebruikt. (Google, sd)

## Toetsing

Doordat, zoals aangeduid in hoofdstuk 4, er tijdens de ontwikkeling van de software met een agile projectmethode gewerkt wordt, wordt er bij iedere iteratie controle gehouden op de hierboven gemelde kwaliteitscriteria. Daarnaast worden alle kwaliteitscriteria tijdens de ontwerpfase mee genomen. Hierdoor wordt er vanaf het begin al zicht gehouden op de kwaliteit van de software. De gekozen code standaard wordt gecontroleerd door een set aan software tools. Met deze tools kan gecontroleerd worden dat deze standaard gevolgd wordt. Daarnaast is door het gebruik van software tools het controleren te automatiseren. Ook wordt er gebruik gemaakt van Code Reviews binnen het team van stagiairs.

# Randvoorwaarden

Tijdens deze stage zal de stagiair zich houden aan de geldende huisregels binnen Gibas. Ook zal de stagiair zorg dragen voor de veiligheid van zijn werkplek en werkzaamheden. Er is sprake van een fulltime stage, dit komt neer op 40 uur per week. Ziekte wordt gecommuniceerd met de stagebegeleider en de officemanager, evenals verlofdagen. Mocht er minder gewerkt worden dan de afgesproken uren, moet dit in overleg met de stagebegeleider en officemanager ingehaald worden op een ander moment. Het is ook een vereiste dat de stagiair niet meer dan 20% van de stagetijd aan andere dingen dan de stageopdracht mag besteden.

# Planning en Organisatie

## Projectorganisatie

De planning is vastgesteld in het volgende diagram (Figuur 4). Omdat er met een Agile methode gewerkt wordt zijn de taken nog redelijk groot vastgesteld, maar zullen gedefinieerd worden in sprints van een week.

Figuur 4 Gantt Diagram van Projectactiviteiten tegenover Projectduur

## Product Back log

Er wordt gebruikt gemaakt van een Trello board om een Product Backlog vast te stellen. Hierin staat alles wat er voorlopig bekend is om het systeem te maken. De requirements zijn verwerkt volgens een MoSCoW methode, onderverdeeld in Must Have, Should Have, Could Have, Won’t Have. Alle onderdelen zijn items in de Product Backlog. Voor sommige van de onderdelen zal eerst onderzoek gedaan moeten worden. Dit zal bestaan uit een combinatie van literatuur onderzoek en tests.

### Must Have

* Solution Architecture
* Festo program interface
* Daemon URCap
* Verbinding Daemon met Festo interface

### Could Have

* Toolbar URCap

### Should Have

* Gebruiksaanwijzing
* User Interface URCap
* Ondersteuning meerdere Festo’s

### Wont Have

* Licensing Software
* Gebruikerstests

## Risico’s

Een mogelijk risico is dat de geschatte tijd per requirements binnen het project uitloopt. Deze zijn op te lossen door middel van juiste planning, voorbereiding en duidelijk vooronderzoek. Omdat het systeem uit verschillende communicatielagen bestaat kan de ontwikkeling ook langer duren dan gepland omdat er mogelijke communicatieproblemen kunnen ontstaan. Deze problemen moeten vroegtijdig geïdentificeerd worden en de planning moet hierop aangepast worden.

## Ethische Aspecten

Er wordt weinig nagedacht over de gevolgen van de dingen die ICT mogelijk maakt. Als het gebouwd kan worden, wordt het snel genoeg gebouwd door programmeurs die niet nadenken over een veiligere, betere manier om het te ontwikkelen. Automatisering zorgt er ook voor dat er banen verdwijnen, er zijn meerdere studies beschikbaar die claimen dat ongeveer 47% van de banen in de Verenigde Staten een risico loopt geautomatiseerd te worden. Deze getallen worden aangevochten maar het blijft een factor. (Carl Benedikt Frey, 2018) Maar tegelijkertijd kan robotisering ook extra banen creëren; er wordt geschat dat robots 75 miljoen banen doen verdwijnen in 2022, maar ook 133 miljoen banen creëert (Centre for the New Economy and Society, 2018).

Hierbij moet ook nagedacht worden over wat voor banen er nou precies verdwijnen met de invoering van dit systeem. Om het simpel te beschrijven zal het systeem een beweging wegnemen die anders een werknemer zou moeten uitvoeren. Om het voorbeeldsysteem van Gibas te gebruiken: om de slijtage van een tandwiel te meten wordt met een sensor op een UR robotarm elke tand gemeten. Hierbij is het een kwestie van precisie waarbij het tandwiel elke keer correct gepositioneerd moeten worden om de juiste uitslag te krijgen. Tevens zal dit niet een volledige taak zijn van een werknemer omdat het draaien enkele secondes zou duren gevolgd door een automatische meting van de slijtage van ongeveer dertig seconden.

Natuurlijk heeft dit systeem oneindig andere toepassingen, maar deze toepassingen zullen vaak voor minimale taken zijn en nooit een volledige taak van een werknemer laten verdwijnen. Ook creëert het de taak van het onderhoud van het systeem, hiervoor is wel juiste kennis nodig. In essentie zal het een simpele taak vervangen door eentje waar meer kennis voor vereist is.

Ethische verantwoordelijkheid is een gekozen onderwerp van het persoonlijk ontwikkelplan van de student, er zal een sectie toegekend worden aan de ethische afwegingen en gevolgen van het systeem in het eindverslag.

# Communicatie

Binnen het bedrijf is over het algemeen ieders deur open. Wanneer er sprake is van officiële communicatie zal dit via de mail gaan. De communicatie met de stagedocent verloopt grotendeels via de mail of telefonisch, verder is er maar sprake van een enkel bezoek op locatie om het plan van aanpak te reviewen. Er is de kans , indien nodig, dat de bezoekmomenten van de andere stagiairs kunnen worden gebruikt worden om een gesprek in te plannen.

Wanneer het project anders verloopt dan is overeengekomen zal dit in overleg met de betrokkenen overlegd worden wat de verdere stappen zijn om het project goed te laten verlopen.

# Bibliografie

Carl Benedikt Frey, M. O. (2018, 04 13). *Automation and the future of work - understanding the numbers*. Opgeroepen op 04 23, 2019, van Oxford Martin: https://www.oxfordmartin.ox.ac.uk/opinion/view/404/

Centre for the New Economy and Society. (2018). *The Future of Jobs Report.* Switzerland: World Economic Forum. Opgeroepen op 04 23, 2019

Festo. (sd). *Support Portal.* Opgeroepen op 04 16, 2019, van Festo: https://www.festo.com/net/SupportPortal/Files/428408/CMMP-AS-M3\_M0-C-HP\_2015-12b\_8046788g1.pdf

Gibas Automation. (sd). *Gibas Automation*. Opgeroepen op 04 8, 2019, van Gibas: http://www.gibas.nl/automatisering

Google. (sd). *Java Style Guide*. Opgeroepen op 04 17, 2019, van https://google.github.io/styleguide/javaguide.html

ISO 25000. (2019). *ISO/IEC25010*. Opgeroepen op 04 17, 2019, van ISO 25000: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

Python. (sd). *PEP 8 -- Style Guide for Python Code*. Opgeroepen op 04 17, 2019, van https://www.youtube.com/watch?v=gSiJPNxuTj8

Toolshero. (sd). *Management Modellen*. Opgeroepen op 04 14, 2019, van https://www.toolshero.nl/management-modellen/handy-model/

Universal Robots. (2018, 04 12). The URScript Programming Language. (Version 3.5.4). Opgeroepen op 04 16, 2019, van http://me.umn.edu/courses/me5286/robotlab/Resources/scriptManual-3.5.4.pdf

Universal Robots. (sd). *Overview of UR software architecture*. Opgeroepen op 04 16, 2019, van Universal Robots+: https://plus.universal-robots.com/getting-started/ur-software-architecture/

Universal Robots. (sd). Principle of URCaps intigration. Opgeroepen op 04 16, 2019, van https://plus.universal-robots.com/getting-started/principle-of-urcaps-integration-in-polyscope/

Universal Robots. (sd). UR SOFTWARE PROCESSES OVERVIEW. Opgeroepen op 04 16, 2019, van https://plus.universal-robots.com/getting-started/ur-software-architecture/

# Bijlage

## Gibas Organigram

