

Een andere kijk op de lektester



JB SYSTEMS

INDUSTRIAL AUTOMATION

DE HAAGSE
HOGESCHOOL

Project: Lektester

Afstudeerder: Thijs Ritsma [07074646]

Opleiding: Technische Informatica [VT]

Bedrijfsbegeleider: Jan Coppenrath

Examinator / Begeleider: Rutger Spaans

Examinator / Expert: Ben Kuiper

Afstudeerperiode: 4-10-2010 t/m 18-02-2011

Versie: 1.0

Referaat

Dit afstudeerverslag is geschreven naar aanleiding van mijn afstudeerperiode bij JB Systems. Deze afstudeerperiode is ter afsluiting van mijn opleiding Technische Informatica in voltijd aan de Haagse Hogeschool.

© 2011 Thijs Ritsma

De titel; een andere kijk op de lektester heeft betrekking op een standalone machine genaamd de lektester. Deze machine is onderdeel van een velgenproductielijn, deze productielijn wordt gebruikt om velgen voor personenauto's te produceren. De lektester test de velgen op lekken in de lasnaad. De lektester is een bestaande machine, in de loop van dit project heb ik op een andere manier naar deze machine moeten kijken ten opzichte van de oorspronkelijke ontwerper.

Medewerkers van de Bibliotheek aan de Haagse hogeschool hebben mij doorverwezen naar de website www.trefwoordentheseausus.nl voor een lijst descriptoren. De onderstaande descriptoren zijn afkomstig van de genoemde website.

Descriptoren:

- Besturingstechniek
- Servosystemen
- Regeltechniek
- UML



Voorwoord

In de eerste plaats wil ik alle docenten aan de Haagse Hogeschool bedanken voor alle tijd en inspanning die zij in mij hebben gestoken. Zij hebben mij de basis gegeven waar ik tijdens het afstuderen verder op heb kunnen bouwen. Ik wil vooral Rutger Spaans en Robin Collard bedanken voor de afstudeer begeleiding die zij mij hebben geboden. Ik wil mijn medestudenten Robin Wijnands en Paul Buts bedanken voor hun inzicht en feedback op mijn documenten in de loop van de afstudeerperiode.

Tevens wil ik iedereen bij JB Systems bedanken die een bijdrage heeft kunnen leveren aan het afstuderen. Ik wil Jan Coppenrath bedanken voor zijn begeleiding.

Als laatste wil ik Frits de Groot bedanken voor zijn tijd en alle uitleg over de manier hoe de lektester werkt.

Vlaardingen, 18 maart 2011

Thijs Ritsma



Inhoudsopgave

1	Inleiding.....	1
2	Samenvatting.....	2
3	Achtergrondinformatie.....	3
	3.1 JB Systems en de Hoogendoorn Groep.....	3
	3.2 Project Lektester.....	5
4	Voorbereiding	12
	4.1 Opdrachtoomschrijving	12
	4.2 Ontwikkelmethode	14
4	Inception fase	20
	4.1 Een software development plan.....	21
	4.2 Iteratiestrategie.....	22
	4.3 Planning	24
5	Elaboration fase.....	25
	5.1 Analyse van de lektester in de huidige situatie.....	25
	5.2 Onderzoek naar servo drives	27
	5.2.1 Verkrijgen van informatie	27
	5.2.2 Het te volgen profiel van een velg	28
	5.2.3 Hardwareplatform	31
	5.2.4 Terug naar de opdrachtgever.....	32
	5.2.5 Een verandering in de opdrachtoomschrijving.....	33
	5.3 Systeemeisen	34
	5.4 Het modelleren van de Lektester	36
	5.5 Prototyping	36
6	Construction fase	37
	6.1 Van ontwerp naar code.....	39
	6.2 Aanroep van functieblokken in OB1.....	40
	6.3 Voorkomen van beschadigingen aan de leksensoren	41
	6.4 Proefassessment.....	43
	6.5 Het opnieuw modelleren van de Lektester	44
	6.5.1 Het bepalen van de tekentechniek	47
	6.5.2 Het opstellen van Use Cases.....	48
	6.5.3 Een model van de hardwareconfiguratie	49
	6.5.4 Programma overzicht.....	52
	6.5.5 Toestandsdiagram	53
	6.5.6 Een model van de stappenafloop.....	54
7	Evaluatie.....	56
	7.1 Productevaluatie	56
	7.2 Procesevaluatie	57
	7.3 Epiloog	58
	7.4 Beroepstaken	59
8	Literatuurlijst.....	63
9	Figuurlijst	64

1 Inleiding

Het onderwerp van dit afstudeerproject betreft de Lektester, deze machine test auto velgen op luchtlekken. De lektester is een onderdeel van de velgenproductielijn. Deze velgenproductielijn is ontwikkeld door Fontijne Grotnes.

Fontijne Grotnes is een klant van JB Systems en gespecialiseerd in het vervaardigen van machines voor de metaalomvormende industrie. JB Systems ontwikkelt de software voor de velgenproductielijn, met uitzondering van de lektester. De software van de lektester is ontwikkeld door de leverancier van de gebruikte servo drives.

Fontijne Grotnes is niet tevreden over de wijze waarop de lektester momenteel werkt en heeft JB Systems gevraagd een alternatief op de gebruikte servo drives en de software te leveren.

In dit afstudeerverslag beschrijf ik welke stappen ik in de loop van het afstudeerproject heb genomen. De motivatie achter deze stappen wordt tevens beschreven. In dit document wordt de lektester in sommige hoofdstukken in detail beschreven, dit is gedaan om de gemaakte keuzes toe te lichten.

In dit document worden zowel Nederlandse als Engelse termen gebruikt. De reden hiervoor is dat veel van de documentatie met betrekking tot het project in het Engels is. De opbouw van dit document is zoveel mogelijk in chronologische volgorde opgezet. Gebeurtenissen die laat in de afstudeerperiode zijn voorgekomen zijn dus achter in het document te vinden.

2 Samenvatting

De eerste stap die ik in dit afstudeerproject heb genomen is het bepalen van de beste ontwikkelmethode voor gebruik in dit project. Vervolgens heb ik de basis van het project opgezet volgens de gekozen ontwikkelmethode. In dit geval is de keuze gevallen op RUP, waardoor het project is opgedeeld in 4 fasen.

In de inception fase heb ik de opdrachtomschrijving vastgelegd zodat er een duidelijk beeld van de doelstelling van dit project is ontstaan. Naast de opdrachtomschrijving heb ik zaken beschreven zoals een risico analyse en de planning voor dit project.

In de elaboration fase heb ik een beeld gevormd van de werking van de lektester. Vervolgens heb ik onderzoek gedaan naar de toepassing van servo drives. In dit onderzoek heb ik alternatieve hardwareplatformen voor de aansturing van de lektester onderzocht. Ik heb een overzicht gemaakt van deze hardwareplatformen met de voor en nadelen van elk afzonderlijk platform. In dit overzicht heb ik tevens een advies uitgebracht met betrekking tot de voorkeursoplossing. Aan de hand van het overzicht heeft de opdrachtgever een keuze gemaakt.

Vervolgens heb ik ontwerpen voor de software gemaakt op basis van het gekozen hardwareplatform. Deze ontwerpen zijn te vinden in het architecture document.

In de construction fase heb ik de ontwerpen gerealiseerd in de vorm van Siemens Step 7 PLC software. Terwijl het project in deze fase verkeerde heeft het proefassessment plaats gevonden. In dit gesprek is de complexiteit van de aan te tonen beroepstaken onvoldoende gebleken.

Naar aanleiding van dit gesprek heb ik de opdrachtgever een voorstel gedaan om de testfase te laten vervallen en software opnieuw te ontwerpen waarbij de software ontwerpen ditmaal wel voldoende complexiteit bevatten. De opdrachtgever heeft dit voorstel geaccepteerd en is akkoord gegaan met het uiteindelijke ontwerp van de software.



3 Achtergrondinformatie

In dit hoofdstuk zal ik een globaal beeld schetsen van de opzet van mijn afstudeerproject bij JB Systems. In de eerste paragraaf van dit hoofdstuk zal ik de het afstudeerbedrijf beschrijven, hierbij geef ik tevens een beschrijving van mijn rol binnen het bedrijf.

In de tweede paragraaf geef ik informatie met betrekking tot het onderwerp van het afstudeerproject. Hierin geef ik een korte beschrijving van de wijze waarop een project bij JB Systems doorgaans wordt uitgevoerd. Met behulp van een afbeelding beschrijf ik de stappen die in de velgenproductielijn worden genomen om een velg te maken uit een stalen plaat. Vervolgens ga ik dieper in op het onderwerp van het afstudeerproject, namelijk de lektester. Ook in het geval van de lektester maak ik gebruik van een afbeelding om een uitleg te geven van de stappen die worden genomen om een velg te testen op lekken. Ten slotte worden de verschillende machineonderdelen die samen de lektester vormen beschreven aan de hand van een diagram.

3.1 JB Systems en de Hoogendoorn Groep

Het bedrijf waar ik mijn afstudeerproject ben gestart is JB Systems in Vlaardingen. Als afstudeerder neem ik deel binnen de organisatie als Software Engineer. JB Systems heeft zich gespecialiseerd in het realiseren van Turn-key projecten bij OEMs.

OEM staat voor Original Equipment Manufacturer, dit zijn bedrijven die onderdelen kopen, samenvoegen en vervolgens onder een eigen naam verkopen. In de industriële automatisering komt dit er vaak op neer dat de fabrikant een machine in elkaar zet, de software uitbesteedt en vervolgens het product als één geheel onder een eigen merknaam verkoopt.

Een turn-key project is een project waarbij de opdrachtgever alles aan de opdrachtnemer over laat. Aan het einde heeft de opdrachtgever een kant-en-klaar product waarbij deze slechts de sleutel hoeft om te draaien. Het concept turn-key projecten wordt door JB Systems gerealiseerd door de software op locatie te installeren en te testen.

JB Systems onderdeel is van de Hoogendoorn Groep. De Hoogendoorn Groep is zoals in de onderstaande afbeelding wordt weergegeven opgebouwd uit een vijftal business units. Deze business units opereren elk als zelfstandige bedrijven maar delen het gebruik van bedrijfsondersteunende afdelingen zonder dat ze hier eigen afdeling voor op hoeven te zetten.

Onder bedrijfsondersteunende afdelingen worden afdelingen zoals IT Services, Personeelszaken en Facilities verstaan. Door gezamenlijk gebruik te maken van deze afdelingen worden de overheadkosten geminimaliseerd.



Figuur 1: Bedrijfsconstructie

De producten van JB Systems worden doorgaans jarenlang gebruikt. Voor een klant is het belangrijk dat deze gedurende de lifetime van een product op de ondersteuning van de leverancier kan rekenen. Zaken doen met een leverancier waarvan de financiële situatie er niet goed uit ziet is dus een slecht idee omdat de ondersteuning bij faillissement kan komen te vervallen.

Bij JB Systems wordt dit risico tegengegaan doordat de Hoogendoorn Groep onderdeel is van het beursgenoteerde bedrijf Batenburg Beheer N.V. Doordat JB Systems deel uit maakt van Batenburg Beheer N.V. brengt dit JB Systems in een sterke financiële positie. Hierdoor is JB Systems in staat om tegenvallers op te vangen.



3.2 Project Lektester

In dit hoofdstuk beschrijf ik de achtergrond van het project. Een velgenproductielijn wordt door de opdrachtgever Fontijne Grotnes voor hun klanten ontworpen. Wanneer het ontwerp van de machines is vastgelegd, worden de schema's en tekeningen naar JB Systems verzonden. Het is vervolgens aan JB Systems om de benodigde hardware voor het aansturen van de machines te selecteren. Deze hardware bestaat uit sensoren en actuatoren. De actuatoren zullen gebruikt worden om de diverse onderdelen van de productielijn aan te sturen. De sensoren worden gebruikt om informatie te verzamelen. Zo wordt er bijvoorbeeld gebruik gemaakt van de fotocel om de aanwezigheid van een product te detecteren.

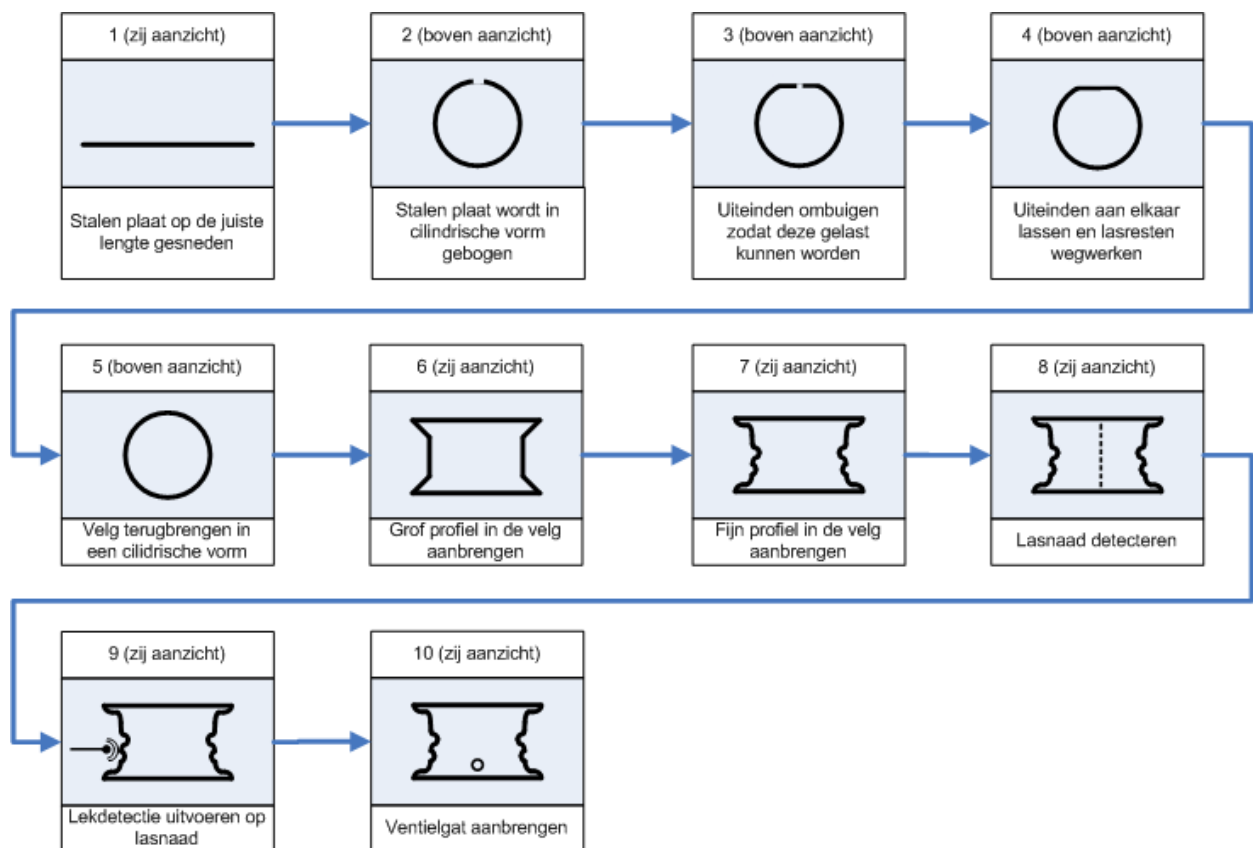
Wanneer de hardware bekend is, zal de software ontworpen en gerealiseerd worden. Tegelijkertijd wordt de productielijn in de fabriekshal van Fontijne Grotnes opgebouwd. Zodra de productielijn is opgebouwd en de software is gerealiseerd, zal de hardware en de software worden getest. In deze fase van het project kunnen er kleine aanpassingen gemaakt worden om prestatiewinst te behalen. Grote fouten zullen in deze fase tevens aan het licht komen.

Wanneer de hele productielijn is getest zal deze worden ontmanteld. De productielijn zal vervolgens in onderdelen worden verscheept naar de klantlocatie waar deze bij aankomst weer in elkaar wordt gezet. Grotere productielijnen worden in delen gestest. In dit geval is de assemblage op de klantlocatie de eerste keer dat de productielijn in zijn geheel is opgebouwd.

Op de klantlocatie zullen er wederom tests worden uitgevoerd op de productielijn om te controleren of deze nog steeds volgens de specificaties functioneert. De situatie op de klantlocatie zal namelijk verschillen ten opzichte van de assemblagehal in Vlaardingen. Hierbij kan er worden gedacht aan temperatuurverschillen waardoor de materialen zich anders gedragen.

Velgenproductielijn

Om een beeld te krijgen van hoe een stalen velg wordt geproduceerd, beschrijf ik de stappen die genomen worden om een stalen velg te produceren. De laatste stap die in het onderstaande diagram niet wordt weergegeven is het toevoegen en lassen van een schotel. Dit is het deel van de velg dat door middel van bouten aan de as van de een auto wordt bevestigd.



Figuur 2: Stappen bij de productie van een velg

Een omschrijving van de bovenstaande figuur wordt op de volgende pagina gegeven.

Het proces begint met een stapel stalen platen welke op maat zijn voorgesneden. De operator laadt de stapel met platen, de machine pakt vervolgens één plaat van de stapel. Deze plaat wordt door de “coiler” gebogen zodat deze een cilindrische vorm krijgt.

In de volgende stap worden de uiteinden omgebogen zodat het lasapparaat een plat oppervlak krijgt. Dit platte oppervlak is absoluut noodzakelijk bij het lassen. Wanneer de twee uiteinden niet goed aansluiten, is het mogelijk dat de las niet goed hecht. Na het lassen worden lasresten weggewerkt totdat de lasnaad in het metaal van velg op gaat en er een gelijkmatig oppervlak is gecreëerd.

Vervolgens wordt de velg rond gemaakt. Wanneer de velg rond is, kan het grove profiel worden aangebracht. Na het grove profiel, wordt het fijne profiel in de velg aangebracht. Het fijne profiel is zoals de velg er uiteindelijk uit komt te zien. Zodra het profiel in de velg is aangebracht, worden de randen van de velg gebogen en wordt de scherpe rand weggewerkt.

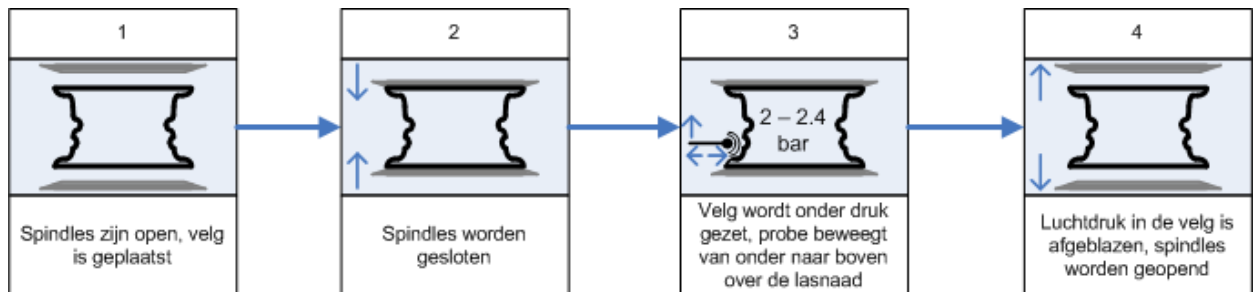
Vervolgens komt de velg in de “orienter” terecht, in dit onderdeel wordt de lasnaad opgezocht. Wanneer de lasnaad is gevonden, wordt deze positie onthouden. Deze positie wordt later gebruikt bij de lektester zodat de velg met de lasnaad in de richting van de sensoren is gepositioneerd.

In de lektester wordt de velg getest op lekken, deze test gebeurt op basis van luchtdruk. De details van de lekttest procedure worden verder in dit document op pagina 12 beschreven. Wanneer de velg is getest en is goedgekeurd, wordt het ventielgat in de velg aangebracht. Dit moet plaats vinden na de lekttest omdat de aanwezigheid van het ventielgat het op druk brengen van de velg onmogelijk maakt.

De beschreven stappen worden simultaan uitgevoerd, dit betekent dat er meerdere velgen tegelijk in de velgenproductielijn aanwezig zijn.

Lektester

In de onderstaande afbeelding staan de stappen die worden genomen bij het testen van de lasnaad van een velg beschreven.



Figuur 3: Stappen bij een lektest

Om de lektest uit te voeren, wordt de velg tussen twee rubberen seals geplaatst. Zodra de velg in de lektester aanwezig is, worden beide seals gesloten. Deze beweging van de seals wordt verzorgd door een servo drive. Deze servo drive is verantwoordelijk voor de aansturing van een servo motor welke de seal naar boven of naar onder kan bewegen. Beide seals worden afzonderlijk aangestuurd. Deze seals zorgen voor een luchtdichte verbinding tussen de seals en de velg.

Vervolgens wordt er lucht de velg in gepompt, de toevoer van deze lucht wordt mogelijk gemaakt door pneumatische solenoid kleppen in de bovenste seal. Wanneer de luchtdruk in de velg tot 2.4 bar is gebracht, worden er sensoren over de lasnaad bewogen.

Er wordt gebruik gemaakt van flowsensoren, deze sensoren meten de luchtstroom die de sensoren passeert. Bij lek in de lasnaad van de velg stroomt de lucht in de velg naar buiten omdat de luchtdruk in de velg hoger is dan de luchtdruk buiten de velg. Deze luchtstroom kan worden gedetecteerd door de gebruikte sensoren. De sensoren zijn geïntegreerd in een arm, deze arm met de sensoren wordt over de lasnaad bewogen.



Wanneer de arm de bovenkant van de velg heeft bereikt, wordt de luchtdruk in de velg opgeheven. Vervolgens gaat de arm terug naar de basispositie en worden beide seals geopend. Op dit punt kan de velg worden verwijderd en kan er een nieuwe velg worden geplaatst waarna het hele proces opnieuw begint. De lektester is in staat om iedere zes seconden één velg te testen op lekken. Bij een handmatige belading (het plaatsen van de velg in de lektester) kan het langer duren.

Zodra er een lek in de velg wordt geconstateerd, moet de arm direct worden stopgezet. Zodra de luchtdruk in de velg is opgeheven, is het veilig om de arm weer te laten bewegen. De sensoren zijn zo gevoelig dat de stroom van lucht in het geval van een lek, de sensoren kan beschadigen. Om deze reden zijn er twee soorten flow sensoren in arm opgenomen.

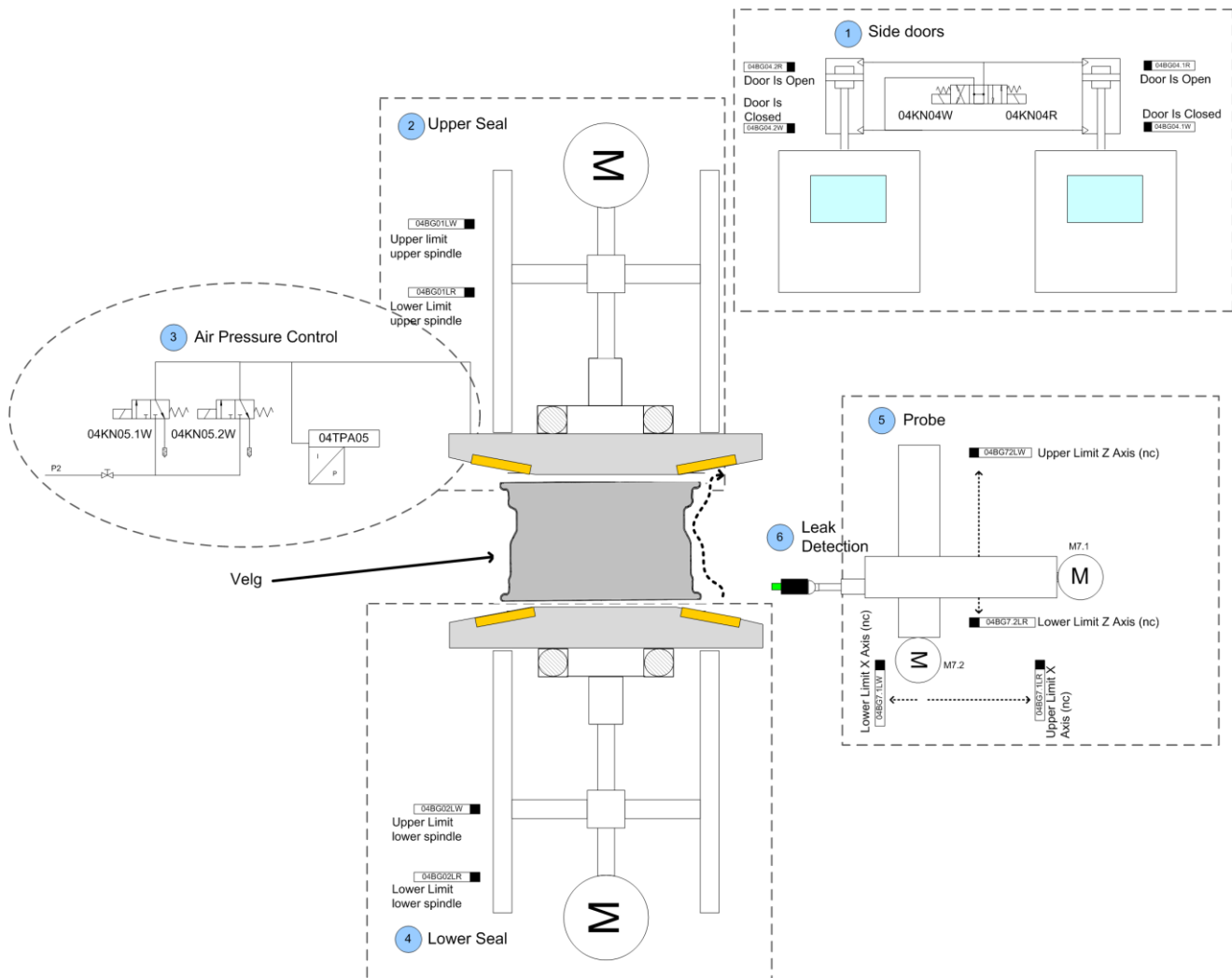
De highflow sensoren zijn aan de bovenkant van de arm geplaatst zodat deze een lek als eerste kunnen detecteren. De microflow sensoren zijn lager geplaatst ten opzichte van de highflowsensoren, deze sensoren worden gebruikt om hele kleine lekken van 10 micron (10 μ) te detecteren. De highflowsensoren zijn niet in staat om deze kleine lekken te detecteren maar dienen als bescherming voor de microflow sensoren. De lucht die vrij komt bij een lek van enkele micron is niet schadelijk voor de microflow sensoren.

Verder in dit document zullen de seals als spindles worden aangeduid. De seal is de rubberen laag op een metalen spindel. Er is hier van een rubberen seal gebruik gemaakt omdat deze ingedrukt moet worden om een luchtdichte verbinding te creëren. Om de benodigde beweging te verkrijgen dienen de spindles te worden aangestuurd.

De arm met sensoren zal verder in dit document als probe worden aangeduid. Het woord probe kan meerdere dingen omschrijven. Voor de beeldvorming is in het bovenstaande stuk gebruik gemaakt van de term arm. De arm en de flowsensoren vormen samen de probe.

Machineonderdelen binnen de lektester

Het onderstaande diagram geeft een beeld van de verschillende machineonderdelen binnen de lektester. Deze onderdelen zijn voorzien van een kader om duidelijk aan te geven welke hardware componenten samen een machineonderdeel vormen. Een omschrijving van het onderstaande diagram wordt op de volgende pagina gegeven.



Figuur 4: Aanwezige hardware onderdelen lektester



Actuatoren

Pneumatische aansturing van:

- Sliding doors (nummer 1 op tekening)
Deze sliding doors worden gebruikt om de lektester af te schermen van invloeden die de lektest kunnen verstoren.
- Luchtdruk in velg (nummer 3 op tekening)
- Cover Sensorhead (niet aanwezig op tekening)

Aansturing door middel van servodrives:

- Upper en lower spindle (nummers 2 en 4 op tekening)
- X en Z as van de probe (nummers 5 en 6 op tekening)

Sensoren

Aangesloten op de servo drives:

- Limietschakelaars boven en onder positie van beide spindles.
- Limietschakelaars uiterste posities van de X en Z as voor de probe.
- Limietschakelaars open en dicht positie van sliding doors.
- Luchtdruksensor, geeft de hoeveelheid druk binnen de velg aan.
- FlowSensor, geeft de hoeveelheid lucht die de sensors passeert aan.
- Fotocel, geeft een hoog signaal wanneer er een velg aanwezig is.

Aangesloten op een printplaat in de kast van de lektester:

- Lekdetectie sensoren op de probe.

Meer informatie met betrekking tot de aanwezige hardware in de lektester is opgenomen in het bijgevoegde Architecture document.

4 Voorbereiding

In de eerste week van het afstudeerproject tref ik voorbereidingen, met voorbereidingen wordt het vastleggen van de opdrachtomschrijving en het kiezen van een ontwikkelmethode bedoeld.

4.1 Opdrachtomschrijving

In de aanleiding wordt beschreven waarom het geformuleerde probleem, een probleem vormt voor de opdrachtgever. Vervolgens wordt in de probleemstelling beschreven wat het probleem is. In de doelstelling wordt beschreven wat het doel van het project is. In het resultaat zijn alle producten en deelproducten opgenomen.

De probleemstelling, doelstelling en resultaat zijn zo concreet mogelijk opgesteld zodat deze op één manier geïnterpreteerd kunnen worden.

Aanleiding

De opdrachtgever, Fontijne Grotnes maakt nieuwe velgen productielijnen waarbij de onderdelen van de productielijnen worden aangestuurd door Siemens apparatuur. Dit betreft Siemens PLC's en HMI panelen.

De opdrachtgever heeft aangegeven aan JB Systems dat hij de gehele productielijn door Siemens apparatuur wil aansturen. Deze beslissing is gemaakt op basis van goede ervaring met Siemens apparatuur en het feit dat Siemens apparatuur over de hele wereld leverbaar is.

Probleemstelling

Één onderdeel in de velgen productielijn wordt momenteel aangestuurd door een servo drive van Control Techniques, dit onderdeel is de lektester.

De software op deze CT drives is ontwikkeld door de mensen van Control Techniques. Wanneer er een storing is in één van de CT drives of wanneer er een aanpassing plaats moet vinden, is de opdrachtgever genoodzaakt mensen van CT in te huren.



Doelstelling

De lektester dient te worden aangestuurd door apparatuur waarmee de inhouse engineers van de opdrachtgever bekend zijn. De engineers van de opdrachtgever zijn bekend met apparatuur van Siemens, Allen-Bradley en Vector.

Het is noodzakelijk dat de lektester op dezelfde wijze blijft functioneren.

Resultaat

Het eindresultaat is een testplan welke is uitgevoerd en ondertekend door de opdrachtgever.

De handtekening van de opdrachtgever in het rapport, betekent dat de opdrachtgever bevestigt dat de software functioneert volgens de gestelde systeemeisen.

De software zelf is tevens een eindproduct en zal als zodanig worden opgeleverd.

Onder op te leveren software wordt verstaan:

- Software Development Plan
- Software Architecture Document
- Een uitgevoerd Testplan (incl. handtekening van de opdrachtgever)
- Broncode van de PLC software
- Broncode van de HMI/SCADA software.

4.2 Ontwikkelmethode

Ik gebruik een ontwikkelmethode om met de complexiteit van het project om te kunnen gaan. Ontwikkelmethoden kunnen worden toegepast om de kwaliteit van de producten en/of de productiviteit van het projectteam te waarborgen. In de loop der tijd zijn er een groot aantal ontwikkelmethoden ontwikkeld voor verschillende toepassingen, daarom is het noodzakelijk een ontwikkelmethode te kiezen die goed aansluit bij het project.

Uit navraag in de organisatie is gebleken dat er binnen JB Systems geen vaste ontwikkelmethode wordt gebruikt. De opdrachtgever heeft aangegeven dat ik de vrijheid heb om zelf te bepalen welke ontwikkelmethode er gebruikt zal worden.

Voordat er een gedegen keuze uit het aanbod van ontwikkelmethoden kan worden gemaakt, is het noodzakelijk een aantal criteria op te stellen. Deze criteria bepaal ik op basis van de eigenschappen van het project. Met eigenschappen van het project bedoel ik zaken zoals de complexiteit en de duur van het afstudeerperiode, mijn eigen kennis van de gebruikte hardware binnen het project is ook van invloed op het bepalen van de criteria.

Vervolgens zoek ik naar ontwikkelmethoden die aan ten minste één van deze criteria voldoen. De gevonden ontwikkelmethoden worden beschreven onder het kopje overzicht. Door elk van de gevonden ontwikkelmethoden te toetsen aan de criteria en deze gegevens vervolgens tegen elkaar af te zetten kan ik bepalen welke ontwikkelmethode het beste bij dit project past. Dit wordt beschreven onder het kopje conclusie.



Criteria

De volgende criteria zijn opgesteld aan de hand van de karakteristieken van het afstudeerproject en mijn eigen beeld en kennis van het project

Goede documentatie

Één van de beroepstaken die ik moet gaan verantwoorden aan het einde van dit project is de beroepstaak: *Ontwerpen van een technisch informatie systeem*.

Om deze beroepstaak te kunnen voltooien, is het van belang een ontwikkelmethode te gebruiken met een duidelijke structuur en waarbij het software ontwerp goed wordt gedocumenteerd.

Iteratief

In dit project komen er een groot aantal zaken aan bod die voor mijzelf onbekend zijn. Hierdoor is het zeer goed mogelijk dat ik in de loop van het project nieuwe inzichten krijg. Bij een iteratieve ontwikkelmethoden is het mogelijk om deze nieuwe inzichten middels een iteratie op het project toe te passen.

Klein ontwikkelteam

Om dit project uit te voeren heb ik geen ontwikkelteam tot mijn beschikking. Ik zal dit project alleen uitvoeren. Om deze reden is een ontwikkelmethode die door een klein ontwikkelteam kan worden gebruikt een pré.

Overzicht

Door middel van Wikipedia en de Google zoekmachine heb ik een overzicht van ontwikkelmethoden voor de ontwikkeling van software gemaakt. Vervolgens heb ik de eigenschappen van de gevonden ontwikkelmethoden opgezocht en hier aantekeningen van gemaakt. Ik heb deze eigenschappen getoetst aan de eerder opgestelde criteria. Door deze gegevens tegen elkaar af te zetten heb ik de keuze voor een ontwikkelmethode gemaakt.

- Agile
- Dynamic systems development method (DSDM)
- Rational Unified Process (RUP)
- SCRUM

Agile

Agile kan niet echt een ontwikkelmethode genoemd worden, het is een verzameling van principes waarmee software ontwikkeld dient te worden. Agile kan toegepast worden op bestaande ontwikkelmethoden. Agile zou bijvoorbeeld toegepast kunnen worden op RUP door de documentatie te beperken of zelfs weg te laten vallen. Dit zal de looptijd van een project kunnen verkorten. Vanwege de onduidelijke structuur binnen Agile projecten wordt het ontwikkelen aan de hand van Agile ook wel “Cowboy coding” genoemd. In Agile wordt elke iteratie gezien als een miniproject met een eigen planning, analyse, ontwerp, test en documentatie.

DSDM

De basis van DSDM ligt in de waterval methode. In deze methode wordt de focus gelegd op de meest belangrijke functionaliteiten of eisen. De eisen worden niet aan het begin van het project vastgelegd. Bij DSDM wordt er vanuit gegaan dat de eisen zullen veranderen naar mate het project vordert. DSDM kent een aantal fases welke het project structureren. Door een fase één of meerdere keren uit te voeren kunnen iteraties binnen een project worden toegepast.



RUP

RUP is ontstaan uit het vraagstuk “waarom mislukken veel projecten?”. Enkele factoren waarom veel projecten falen zijn: overmatige complexiteit en dubbelzinnige of onduidelijke eisen. RUP biedt een oplossing op deze problemen door een duidelijke structuur te creëren. Eisen kunnen in de loop van het project veranderen, door in iteraties de eisen te evalueren en waar nodig aan te passen is de kans groter dat de opdrachtgever krijgt wat hij/zij had verwacht. Het maken van prototypes geeft de opdrachtgever een beter beeld van het uiteindelijke product.

SCRUM

SCRUM is een ontwikkelmethode die heel geschikt is voor kleine ontwikkelteams. Elke dag begint met een sprint meeting in deze meeting worden 3 vragen beantwoord:

- Wat heb je gisteren gedaan?
- Wat ga je vandaag doen?
- Ben je tegen problemen aangelopen?

Doordat iedereen welkom is bij deze meeting, is iedereen op de hoogte van de status van het project. Deze methode maakt gebruik van sprints. Een sprint duurt doorgaans 2 tot 4 weken en heeft een stuk werkende en geteste software tot resultaat. De systeemeisen worden gedurende een sprint niet gewijzigd.

Aan het einde van de sprint vindt er een sprint review plaats met de opdrachtgever. In deze review wordt er een demo van de software gegeven waarbij de opdrachtgever feedback kan leveren. Door de opdrachtgever zo vroeg mogelijk te betrekken bij de ontwikkeling van het product, wordt de kans dat de opdrachtgever krijgt wat deze voor ogen heeft vergroot.

Bij JB Systems wordt er momenteel een proef gedaan door bij een aantal projecten SCRUM toe te passen. Om deze reden is er op kantoor veel informatie met betrekking tot SCRUM te vinden. Tevens heb ik de mogelijkheid om scrum meetings bij te wonen om te zien hoe scrum in de praktijk wordt gebruikt.

Conclusie

De ontwikkelmethoden die niet aan de gestelde criteria voldoen heb ik met een – geclassificeerd. De methoden die aan deze criteria voldeden heb ik een ± gegeven. De methoden die veel aandacht aan de gestelde criteria besteden heb ik met een + gemarkeerd. Vervolgens heb ik de methoden een totaalscore gegeven aan de hand van de opgestelde criteria. In het onderstaande overzicht is echter niet duidelijk te zien welke ontwikkelmethode het beste voldoet aan mijn criteria, alle methoden hebben een totaalscore van 7 punten.

Criteria	Legenda	Score
1 Goede documentatie	- Slecht	1
2 Iteratief	± Voldoende	2
3 Klein ontwikkelteam	+ Goed	3

Criteria	1	2	3	Totaal
Methode				
Agile	-	+	+	7
DSDM	+	+	-	7
RUP	+	+	-	7
SCRUM	-	+	+	7

Figuur 5: Conclusie keuze in ontwikkelmethoden

Uit het bovenstaande overzicht kan worden afgeleid dat ik moet kiezen tussen twee criteria. Ik heb geen ontwikkelmethode gevonden die aan alle criteria voldoet.

Ik sta nu voor de keuze tussen goede documentatie (kwaliteit) en klein ontwikkelteam (tijd). Wanneer ik kies voor goede documentatie, komt dit de kwaliteit van de deelproducten van het project ten goede. Wanneer ik kies voor een methode met een klein ontwikkelteam vergroot dit de kans dat ik het project binnen de gestelde tijd kan voltooien.

Wanneer de deelproducten van goede kwaliteit zijn, wordt het voor mij makkelijker om aan te tonen dat ik de gekozen beroepstaken naar behoren heb uitgevoerd. Wanneer ik de keuze maak om gebruik te maken van een methode welke is gericht op kleine ontwikkelteams, wordt de focus gelegd op het product in plaats van de documentatie. Hierdoor wordt het moeilijk om aan te tonen dat ik de beroeptaken op het gewenste niveau heb uitgevoerd.



Om te slagen voor het afstuderen is het noodzakelijk dat ik kan aantonen dat ik de beroepstaken naar behoren heb uitgevoerd. Om die reden heb ik gekozen om een weging toe te passen op de criteria. Omdat het criterium klein ontwikkelteam is voor mij minder belangrijk is, heb ik hier een weging van 50% toegepast. Dit heeft tot resultaat dat zowel DSDM en RUP evenveel punten scoren.

Vanwege de goede structuur en de grote hoeveelheid documentatie binnen de ontwikkelmethode hebben beide methoden een grote learning curve. Omdat ik een beperkte tijd heb om dit project uit te voeren is de kennis die ik reeds bezit op het gebied van ontwikkelmethoden ook van belang. Om deze reden heb ik mijn bekendheid met de verschillende ontwikkelmethoden verwerkt in het onderstaande overzicht. Door de toevoeging van het laatste criterium is er slechts één ontwikkelmethode met de hoogste score.

Criteria	Weging	Legenda	Score
1 Goede documentatie	(100%)	- Slecht	1
2 Iteratief	(100%)	± Voldoende	2
3 Klein ontwikkelteam	(50%)	+ Goed	3
4 Kennis van methode	(50%)		

Criteria	1	2	3	4	Totaal
Methode					
Agile	-	+	+	-	6
DSDM	+	+	-	-	6
RUP	+	+	-	+	8
SCRUM	-	+	+	+	7

Figuur 6: Conclusie keuze in ontwikkelmethoden met weging

Uit de bovenstaande tabel kan worden opgemaakt dat RUP de beste keuze is om toe te passen op mijn project. De documentatie met betrekking tot de ontwikkelmethode die ik heb gebruikt is het boek *The Rational Unified Process An Introduction*. De keuze voor RUP heeft ook betrekking op de indeling van mijn afstudeerverslag. Het verloop van het project is opgedeeld in verschillende fases, deze indeling is terug te zien in de indeling van dit document.

4 Inception fase

De eerste fase in RUP is de inception fase. Het doel van deze fase is het bereiken van overeenstemming met de opdrachtgever betreft de opdrachtschrijving. In deze fase worden tevens de grenzen van het project afgebakend door middel van randvoorwaarden en een afbakening. de afbakening beschrijft welke zaken buiten de grenzen van het project vallen. Onderwerpen zoals acceptatiecriteria en deelproducten worden in deze fase ook behandeld.

In dit hoofdstuk beschrijf ik welke stappen ik zal nemen bij het doorlopen van de inception fase. In de eerste paragraaf geef ik aan hoe het software development plan tot stand is gekomen. De wijze waarop de informatie is verkregen en hoe de informatie is toegepast wordt in deze paragraaf ook besproken.

In de tweede paragraaf beschrijf ik de iteratie strategie. In de iteratiestrategie wordt er beschreven in welke iteraties de verschillende fases zijn opgedeeld en welke doelen er aan deze iteraties zijn gesteld.

In de derde paragraaf wordt de globale planning weergegeven. Deze planning wordt echter niet bevroren voor de resterende duur van het project. Door de iteratieve opzet van RUP worden onderwerpen zoals de systeemeisen, risico's en de planning regelmatig bijgewerkt naar mate het project vordert en de situatie veranderd.



4.1 Een software development plan

In de tweede week van het afstudeerproject wordt de inception fase gestart. In deze fase bestudeer ik de globale werking van de lektester en leg ik de fundamenteën van het project volgens de gekozen ontwikkelmethode.

Hierbij kijk ik naar de wijze waarop de lektester volgen test, welke machineonderdelen er in de lektester aanwezig zijn en hoe de monitoring en aansturing door de operator plaats vindt. Hierbij maak ik gebruik van de informatie die ik in de voorbereiding van het project (de eerste week) heb verzameld.

Doordat de lektester een machine is die reeds in productie is genomen, is er een grote hoeveelheid aan informatie en documentatie voorhanden. Deze documentatie is echter niet op een centraal punt onder gebracht maar is verspreid over verschillende bronnen. Hierbij kan worden gedacht aan handleidingen, elektra aansluitschema's en PowerPoint presentaties.

In de gebruikershandleiding van de lektester is een afbeelding opgenomen waarop de bedieningspanelen worden weergegeven. Deze bedieningspanelen worden door een operator gebruikt om de lektester aan te sturen. De bedieningspanelen geven een goed beeld van de functionaliteiten van de lektester. Om deze functionaliteiten te beschrijven maak ik gebruik van use cases. Ik heb van elke knop op het bedieningspaneel een use case gemaakt waarin ik de reactie van de lektester op de desbetreffende knop beschrijf. Uitzonderingen hierop zijn de knoppen met een start/stop functie, in deze gevallen maak ik niet voor iedere knop een beschrijving. In dit geval beschrijf ik slechts die actie die gestart of gestopt kan worden.

Het use case diagram en de beschrijvingen zijn te vinden in het bijgevoegde software development plan. De afbeeldingen van de bedieningspanelen zijn tevens in dit document opgenomen.

4.2 Iteratiestrategie

Het iteratieplan beschrijft de indeling van de verschillende iteraties en de doelen die ik aan deze iteraties heb gesteld.

Ik houd bij het maken van de iteratiestrategie rekening met de kennis die ik op dit moment heb met betrekking tot het afstudeerproject. Omdat deze kennis redelijk beperkt is, zal ik meer tijd besteden aan het analyseren van de huidige situatie. Deze kennis is van belang omdat ik hierdoor een beter beeld krijg van de omvang van het project en de mogelijke consequenties die uit een wijziging in de hardware kunnen volgen.

Omdat mijn kennis van de lektester zal groeien naar mate het project vordert plan ik twee iteraties voor de elaboration fase. Hierdoor is het goed mogelijk dat ik in de tweede iteratie systeemeisen en ontwerpen moet aanpassen omdat deze incorrect zijn gebleken.

Voor de construction fase verwacht ik slechts één iteratie nodig te hebben. Omdat ik in deze fase gebruik kan maken van mijn eigen software ontwerpen, verwacht ik bij het realiseren van de software minder tijd nodig te hebben dan dat ik zal nodig hebben bij het ontwerpen van de software.

De transition fase heeft als doel de acceptatietest met de opdrachtgever. De test in deze fase verschilt met de tests die in de construction fase uitgevoerd zullen worden. In de construction fase wordt de werking van de software getest en in de transition fase wordt er getest of het product aan de eisen van de opdrachtgever voldoet. In deze fase wordt er bepaald of het project klaar is.



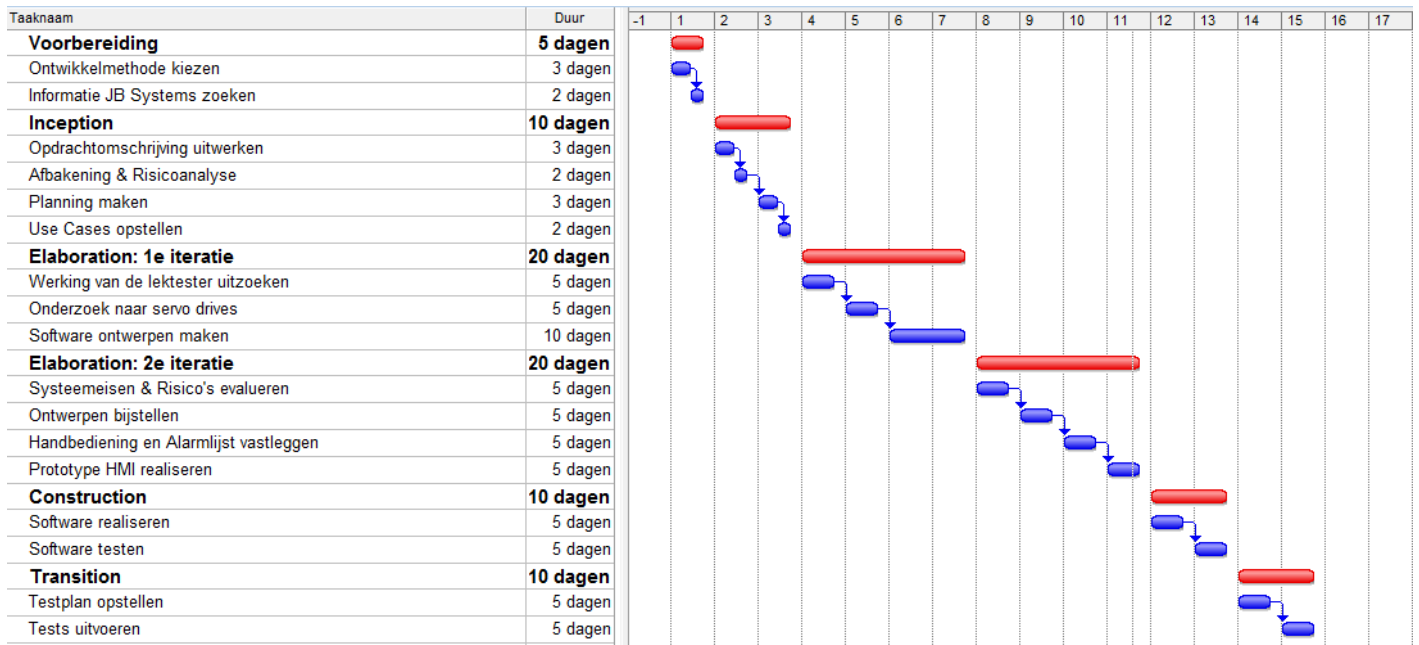
In de onderstaande tabel wordt een overzicht van de iteraties gegeven. Per iteratie is tevens het doel van de iteratie beschreven.

Fase	Iteratie	Doel
Inception (Vastgelegd in Software Development Plan)	#1 Plan van aanpak	- Opdrachtoomschrijving opstellen - Risico's opstellen - Planning maken - Use Cases opstellen
Elaboration (Vastgelegd in Architecture Document)	#1 Architectuur opstellen	- Analyse huidige situatie - Onderzoek naar servo drives - Software ontwerpen opstellen
	#2 Architectuur afronden	- Systeemeisen & Risico's evalueren - Software ontwerpen bijstellen - Handbediening & Alarmlijst vastleggen - Prototype HMI realiseren
Construction (Vastgelegd in Deployment Document)	#1 Implementatie	- Software ontwerpen realiseren - Software testen
Transition (Vastgelegd in Testplan)	#1 Testfase	- Testplan opstellen - Tests uitvoeren

Figuur 7: Iteratieplan

4.3 Planning

In de onderstaande tabel wordt de globale planning weergegeven. Dit is de planning zoals ik deze in het software development plan heb opgenomen.



Figuur 8: Eerste planning



5 Elaboration fase

In de elaboration fase wordt de architectuur van het systeem ontworpen. Deze architectuur wordt doorgaans vastgelegd in verschillende diagrammen en modellen. Bij het modelleren in RUP wordt er gebruik gemaakt van de UML standaard. In deze fase worden er ook use cases opgesteld, deze use cases zijn voor een opdrachtgever vaak makkelijker te begrijpen dan technische diagrammen. In de elaboration fase wordt tevens gebruik gemaakt van prototypes, door middel van een prototype kan er worden aangetoond dat bepaalde functionaliteiten kunnen worden gerealiseerd.

5.1 Analyse van de lektester in de huidige situatie

De doelen van de eerste iteratie in de elaboration fase zijn een analyse van de huidige situatie, het opstellen van de systeemeisen, het voltooien van de use cases en het kiezen van een hardwareplatform. Ik zal eerst de analyse van de huidige situatie uitvoeren om vervolgens de overige te doelen aan te pakken met de informatie die uit deze analyse is voortgevloeid.

Een engineer van de opdrachtgever heeft een diagram ter beschikking gesteld waarin de lektester wordt weergegeven. Dit diagram wordt afgebeeld in figuur 4 (pagina 10). Dit diagram geeft een beeld van de werking en aansturing van de verschillende sensoren en actuatoren in de lektester. Dit diagram was echter niet helemaal up to date waardoor niet alle onderdelen worden weergegeven.

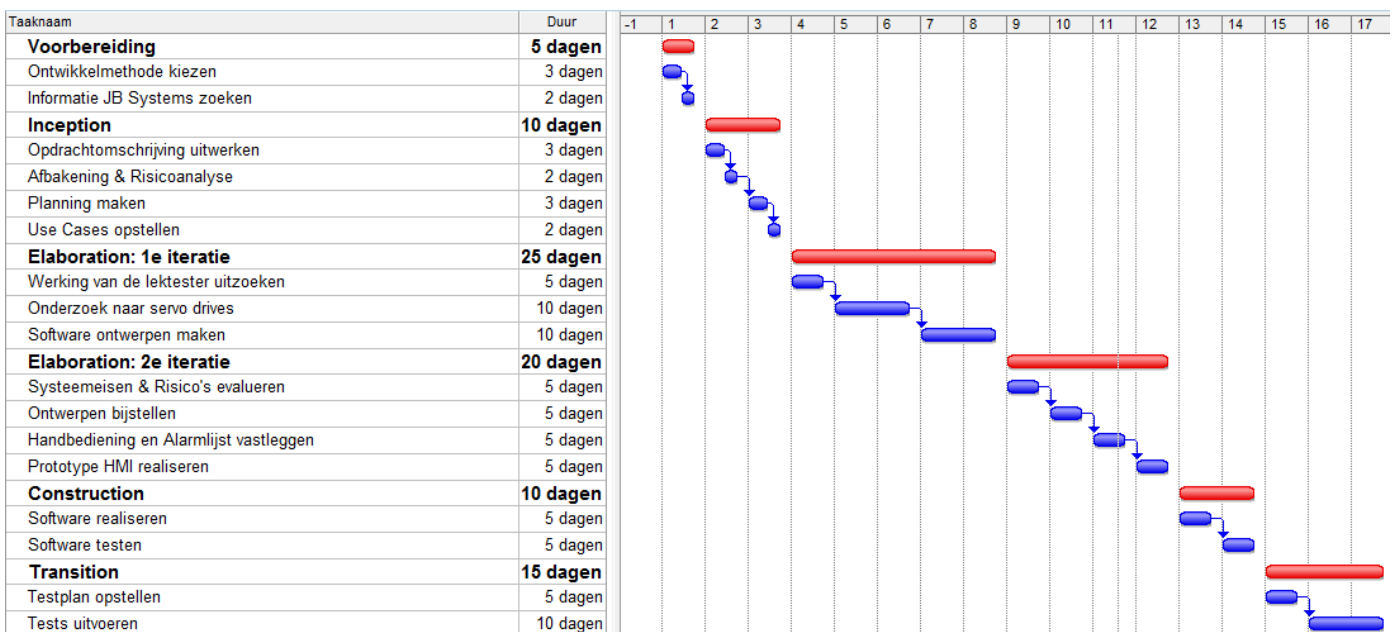
Ik heb de engineer op de ontbrekende onderdelen gewezen en heb hem om meer informatie gevraagd. Hierop antwoordde hij dat dit het enige diagram was dat hij tot zijn beschikking had. Door meer vragen te stellen en door te vragen op de antwoorden van de engineer heb ik een goed beeld gekregen ontbrekende onderdelen. Deze informatie heb ik in de beschrijving van het diagram opgenomen om het verhaal compleet te maken.

Na de analyse van de huidige situatie heb ik een beter beeld gevormd van de taken die nog uitgevoerd moeten worden en hoe lang dit ongeveer duurt. Ik heb de planning op een aantal punten gewijzigd omdat ik knelpunten voorzie. Deze punten zijn het onderzoek naar servo drives en het testen.

Vanwege de complexiteit van servo drives en mijn kennis hiervan heb ik de duur van het onderzoek met één week verlengd naar 2 weken.

In de afstudeerperiode is het niet mogelijk om over de hardware van de lektester te beschikken. Dit betekent dat de software door middel van simulatie moet worden getest. Dit kan betekenen dat er meer tijd nodig is om te testen. Voor de zekerheid heb ik hier een extra week voor uitgetrokken. Hierdoor komt de beschikbare tijd voor de tests op drie weken te liggen.

Na de wijzigingen ziet de planning er als volgt uit:



Figuur 9: Eerste wijziging in de planning



5.2 Onderzoek naar servo drives

In de lektester wordt er gebruikt gemaakt van de servo drives. Bij aanvang van dit project had ik nog nooit van het woord servo drive gehoord. Om kennis op te doen op het gebied van servo drives zal ik een onderzoek uitvoeren.

De eerste stap die ik in dit onderzoek zal nemen is het opstellen van een onderzoeksplan. Dit onderzoeksplan stel ik op in samenwerking met de opdrachtgever. In dit onderzoeksplan zal ik aangeven wat ik ga onderzoeken, hoe ik dat ga onderzoeken en waarom ik dat ga onderzoeken.

Nadat de opdrachtgever met het onderzoeksplan akkoord is, zal ik het onderzoek starten. In het eerste deel van het onderzoek zal ik uitgezocht en beschrijven wat een servo drive is en waar servo drives worden toegepast.

5.2.1 Verkrijgen van informatie

Ik ben mijn zoektocht naar informatie gestart op de website van de HHS bibliotheek. Hiervoor moet ik zoeken op basis van sleutelwoorden, om een aantal sleutelwoorden te vinden heb ik gebruik gemaakt van de zoekmachine Google. De volgende sleutelwoorden bieden een groot aantal resultaten:

- servo
- frequentieregelaar
- frequencyinverter.

Vervolgens heb ik deze sleutelwoorden ingevuld op de website van de bibliotheek, dit heeft echter geen resultaten geleverd.

Door middel van Wikipedia en de website van Vector heb ik veel informatie met betrekking tot servo drives verkregen. Ik heb gebruik gemaakt van de website van Vector omdat de huidige velgenlijn volledig is uitgerust met servo drives van Vector, met uitzondering van de lektester. Om die reden lijkt de website van Vector mij een goede bron van informatie.

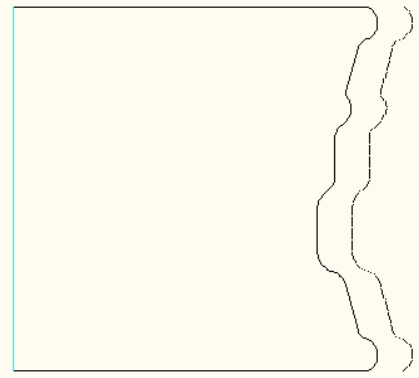
Er is veel informatie te vinden over de wijze waarop een servo drive werkt. Veel van deze informatie gaat echter veel te diep op te materie in. Zo wordt er uitgelegd hoe de gelijkspanning wordt omgezet in wisselspanning om zo een elektromotor aan te sturen om een bepaald toerental te bereiken. Wanneer ik al deze informatie zou willen verwerken en begrijpen heb ik meer tijd nodig dan de tijd die ik tot mijn beschikking heb. Om die reden heb ik gekozen om mij te richten op de toepassing van servo drives. De informatie met betrekking tot servo drives is te vinden in het onderzoeksrapport.

5.2.2 Het te volgen profiel van een velg

De servo drives voor de aansturing van de probe worden gebruikt om de probe over de lasnaad van de velg te bewegen zodat de probe lekken in de lasnaad kan detecteren.

Om deze beweging uit te voeren moet het profiel van het velgtype bekend zijn. Voor elke velgtype is er een bijbehorende AutoCad tekening van het profiel van de velg. Er wordt gebruik gemaakt van een applicatie om aan de hand van de tekening een te volgen pad te genereren.

Deze applicatie creëert een offset in het profiel zodat er een afstand tussen de velg en de probe overblijft. Een voorbeeld hiervan wordt weergegeven in figuur 9. De lijn aan de geeft het te volgen pad weer.



Figuur 10: Te volgen profiel

Het gegenereerde profiel wordt opgeslagen in een .csv (comma separated values) bestand. De opmaak van dit bestand is als volgt:

```
DB_Raw profile data.RawRadius [ 0], 292
DB_Raw profile data.RawRadius [ 1], 292
DB_Raw profile data.RawRadius [ 2], 293
DB_Raw profile data.RawRadius [ 3], 293
DB_Raw profile data.RawRadius [ 4], 294
DB_Raw profile data.RawRadius [ 5], 295
DB_Raw profile data.RawRadius [ 6], 294
DB_Raw profile data.RawRadius [ 7], 293
```

Het eerste getal representeert de Z-as (Verticaal) het tweede getal representeert de X-as (Horizontaal). Zoals in het voorbeeld wordt weergegeven gaat de waarde van de Z-as met stappen van 1 omhoog. In werkelijke eenheden betekent dit een stap van 1 millimeter.

De waarden van de X-as geven een weergave van de afstand die de probe naar de velg toe(binnen) moet bewegen. In het voorbeeld worden dus de posities van de eerste 7 millimeter van het te volgen pad van de probe weergegeven. De grootste velg die in de lektester getest kan worden is maximaal 525 millimeter hoog. Om die reden zijn er 525 posities in het profiel opgenomen. Bij een kleinere velg is de X waarde 0 aan het einde van de reeks posities, dit is te zien in het onderstaande voorbeeld bij posities 17 tot 22.

Na inspectie van de broncode van de CT drive is gebleken dat de CT drive gebruik maakt van 22 posities om het profiel van een velg van maximale grootte te volgen. Deze posities worden berekend door de posities uit het .csv bestand te halen, met intervals van 25. Hieronder wordt een voorbeeld gegeven van het gebruikte profiel door de CT drive.

1. B_Raw profile data.RawRadius[0],292
2. DB_Raw profile data.RawRadius[24],289
3. DB_Raw profile data.RawRadius[49],279
4. DB_Raw profile data.RawRadius[74],267
5. DB_Raw profile data.RawRadius[99],255
6. DB_Raw profile data.RawRadius[124],255
7. DB_Raw profile data.RawRadius[149],271
8. DB_Raw profile data.RawRadius[174],273
9. DB_Raw profile data.RawRadius[199],273
10. DB_Raw profile data.RawRadius[224],273
11. DB_Raw profile data.RawRadius[249],273
12. DB_Raw profile data.RawRadius[274],274
13. DB_Raw profile data.RawRadius[299],274
14. DB_Raw profile data.RawRadius[324],278
15. DB_Raw profile data.RawRadius[349],284
16. DB_Raw profile data.RawRadius[374],296
17. DB_Raw profile data.RawRadius[399],0
18. DB_Raw profile data.RawRadius[424],0
19. DB_Raw profile data.RawRadius[449],0
20. DB_Raw profile data.RawRadius[474],0
21. DB_Raw profile data.RawRadius[499],0
22. DB_Raw profile data.RawRadius[524],0

Doordat er een interval van 25 millimeter is gebruikt, kan het voorkomen dat er botsing tussen de probe en de velg plaats vindt. Deze botsing wordt aan de onderaan de onderstaande afbeelding geïllustreerd. De rode lijn markeert het te volgen pad voor de probe.

Door gebruik te maken van een offset tussen de velg en de probe worden dit soort botsingen voorkomen. Deze offset wordt weergegeven door de groene lijn. Het gebruik van deze offset is niet zonder nadelen. Een van deze nadelen is dat de afstand tussen de velg en de probe niet overal gelijk is. Dit verlaagt de kwaliteit van de lektest doordat kleine lekken in de moeilijk bereikbare plekken (weergegeven met een pijl) mogelijk niet worden gedetecteerd.



Figuur 11: Voorbeeld botsing probe op velg



5.2.3 Hardwareplatform

De volgende stap in het onderzoek was, het zoeken naar een geschikt hardwareplatform. Met een hardwareplatform wordt de samenstelling van de verschillende apparaten waarop besturing plaats kan vinden binnen de lektester bedoeld. Voorbeelden van deze apparaten zijn PLC's en HMI's, beide apparaten dragen bij aan de besturing van de lektester.

Uit een gesprek met de opdrachtgever is gebleken dat de CT drives moeten worden vervangen door servo drives van Vector (SEW Movidives).

Deze servo drives worden in verschillende onderdelen van de velgenlijn gebruikt. Doordat de onderdelen van de velgenlijn zijn uitgerust met hetzelfde type servo drives is een kleine voorraad spare parts voldoende om met hardware defecten om te kunnen gaan.

Een wens van de opdrachtgever is de lektest nauwkeuriger te maken ten opzichte van de huidige lektester. De kwaliteit van de lektest kan worden verbeterd door gebruik te maken van meer dan 22 posities. Het eerder benoemde .csv bestand bevat 525 posities. Dit betekent dat wanneer de lektester alle 525 posities op zou kunnen volgen, de kwaliteit van de lektest aanzienlijk is verbeterd.

Omdat de doelstelling van het project beschrijft dat de lektester door middel van Siemens apparatuur aangestuurd dient te worden, heb ik contact opgenomen met een verkoper van Siemens.

Siemens heeft een uitbreiding op een PLC genaamd een Technology CPU, dit betreft een uitbreidingsmodule. Deze CPU draait naast de standaard CPU en kan worden gebruikt om servo drives aan te sturen.

Deze verkoper heeft me uitgelegd dat er sprake is van een kleine vertraging (van 2 of 3 milliseconden) wanneer een Siemens PLC communiceert met SEW servo drives.

Gezien de dichtheid van de posities (1 millimeter van elkaar), is het mogelijk dat de aansturing van de servo drives niet snel genoeg gerealiseerd kan worden. Door deze vertraging kan de probe een punt al gepasseerd zijn terwijl de PLC de servo drives naar een eerder punt stuurt. Om deze situatie te verduidelijken heb ik praktijkvoorbeeld bijgevoegd.

Praktijkvoorbeeld:

Persoon A rijdt op de snelweg en is onderweg om persoon B op te halen. Persoon A wordt tijdens het rijden door persoon B opgebeld. In het telefoon gesprek vraagt de beller persoon A naar de huidige lokatie. Persoon A kijkt naar rechts en noemt het nummer van een zojuist gepasseerd hectometerpaaltje waarop persoon B antwoordt: daar had je moeten zijn!

Een oplossing voor het probleem is het aantal te gebruiken posities zodat de doelpositie niet zo snel voorbij wordt gegaan.

Tijdens het zoeken naar informatie over servo drives op de website van Vector is het mij opgevallen dat Vector een product aanbiedt met gelijkwaardige functionaliteiten ten opzichte van de Siemens Technology CPU. Dit product is de MOVI-PLC welke in staat is om meerdere servo drives aan te sturen. Vervolgens heb ik contact opgenomen met Vector om te kijken of zij een oplossing kunnen aanbieden waarbij alle 525 gevolgd kunnen worden. In een gesprek met een verkoper van Vector heeft deze mij uitgelegd dat de vertraging in de communicatie met Siemens apparatuur bij gebruik van de MOVI-PLC niet aan de orde is. De MOVI-PLC is tevens in staat om tot 2048 posities in een buffer op te slaan.

De conclusie die ik hier uit trek is dat de opdrachtgever een beter resultaat kan krijgen wanneer de doelstelling van het project wordt gewijzigd zodat SEW apparatuur ook tot de mogelijkheden behoort.

5.2.4 Terug naar de opdrachtgever

Ik heb de informatie met betrekking tot de Siemens Technology CPU en de MOVI-PLC aan de opdrachtgever voorgelegd, hierbij heb ik tevens een advies uitgebracht. Ik heb de opdrachtgever een keuze gegeven tussen:

- het behouden van een doelstelling en gebruik van de Siemens apparatuur.
- het aanpassen van de doelstelling en gebruik van Vector apparatuur en een verbetering in de kwaliteit van de lekttest.

De opdrachtgever heeft mijn advies opgevolgd en gekozen voor de tweede stelling. Dit heeft ertoe geleid dat de doelstelling van het project aangepast dient te worden.

Verdere resultaten van dit onderzoek zijn te lezen in het onderzoeksrapport.



5.2.5 Een verandering in de opdrachtomschrijving

De opdrachtgever heeft het voorstel om de doelstelling aan te passen zodat er een kwalitatief betere oplossing gekozen kan worden geaccepteerd.

Een wijziging in de opdrachtomschrijving heeft tot gevolg dat de onderdelen welke op de oude opdrachtomschrijving zijn gebaseerd onjuist kunnen zijn. De volgende onderdelen van het vision document zijn aangepast:

- Doelstelling
- Risicoanalyse

De oude doelstelling “*De lektester dient te worden aangestuurd door Siemens apparatuur.*” Is vervangen door de onderstaande doelstelling.

De lektester dient te worden aangestuurd door apparatuur waarmee de inhouse engineers van de opdrachtgever mee bekend zijn. De engineers van Fontijne Grotnes zijn bekend met apparatuur van Siemens, Allen-Bradley en Vector.

De wijziging in de opdrachtomschrijving heeft er toe geleid dat er een risico kan worden geëlimineerd.

Het onderzoek heeft aangetoond dat de SEW hardware de functionaliteiten van de (in de lektester aanwezige) CT drives kan evenaren en zelfs kan overtreffen.

Risico 2

Risico	De hardware van de probe is niet in staat de gewenste functionaliteit te bieden. (<u>Middel</u>) Momenteel wordt de probe beweging aangestuurd door de CT drives. Het is mogelijk dat de vervangende hardware deze aansturing minder nauwkeurig kan realiseren.
Kansverlaging	Om de kans dat dit risico zich voordoet te verminderen, zijn er eisen opgesteld waaraan de aansturing dient te voldoen.
Impactvermindering	Alternatieven zoeken
Plan B	De SEW drives aansturen met 21 posities zoals bij de CT drive ook het geval is.

5.3 Systeemeisen

In het gesprek met de opdrachtgever heb ik gesproken over systeemeisen. Omdat de lektester een bestaande machine is en reeds binnen bepaalde parameters opereert, heeft de opdrachtgever aangegeven dat er geen aanvullende systeemeisen zijn. De nieuwe uitvoering van de lektester dient dus aan dezelfde eisen te voldoen als waar de huidige lektester aan voldoet. De wens van de opdrachtgever waarin de kwaliteit van de lektester wordt verbeterd blijft echter nog steeds van kracht.

Ik heb de opdrachtgever gevraagd naar documentatie met betrekking tot de specificaties van de lektester. De opdrachtgever heeft aan deze eis helaas geen gehoor gegeven. Op dat moment zag ik twee mogelijkheden om de systeemeisen vast te stellen:

- Interview met de opdrachtgever of een engineer
- Reverse engineeren van de eisen aan de hand van de broncode

Het reverse engineeren van de software aan de hand van de broncode kan leiden tot concrete eisen. Dit is echter alleen het geval wanneer de persoon die dit uitvoert de werking van de code daadwerkelijk begrijpt. Omdat de lektester is uitgerust met hardware waarvan de werking mij onbekend is, maak ik gebruik van de eerste optie.



Het heeft twee weken geduurd voordat de opdrachtgever tijd heeft kunnen maken voor een gesprek van een half uur. In dit gesprek is gebleken dat de opdrachtgever niet op de hoogte was van de specifieke hardware configuratie van de lektester. Om antwoorden op vragen met betrekking tot de hardware van de lektester te krijgen, is het noodzakelijk met een engineer van Fontijne Grotnes te praten.

Ik heb één gesprek gehad met Frits de Groot, een engineer van Fontijne Grotnes. In dit gesprek heeft Frits mij de werking van een aantal onderdelen van de lektester uitgelegd. Aan het einde van dit gesprek hebben we afgesproken om wekelijks een soortgelijk gesprek te hebben. Op deze manier kunnen vragen met betrekking tot de lektester snel worden beantwoord. Enige dagen later heb ik een mail van Frits ontvangen waarin hij uitlegt dat hij op last van zijn manager de gesprekken moet afzeggen.

Er bleek slechts één persoon met kennis van de lektester te zijn binnen Fontijne Grotnes. Hierdoor bleef de reverse engineering optie als enige over. Ter ondersteuning bij het reverse engineeren heb ik per e-mail antwoorden op mijn vragen kunnen ontvangen. Omdat Frits vaak in het buitenland verblijft, duurt het vaak 2 a 3 weken voordat mijn vragen worden beantwoord. In de eerste iteratie stel ik de voorlopige systeemeisen op en baseer ik mijn ontwerpen op deze eisen. In de tweede iteratie verwerk ik de verkregen antwoorden op mijn vragen in uiteindelijke systeemeisen en pas ik mijn ontwerpen aan waar dit nodig is.

5.4 Het modelleren van de Lektester

Bij het modelleren van PLC software wordt er eigenlijk slechts één soort diagram gebruikt om het gedrag van de software weer te geven. Dit betreft een Sequential Function Chart. Om die reden heb ik de ontwerpen van de lektester opgesteld in SFC's.

Aan de hand van de broncode van de CT drive heb ik de software ontwerpen opgesteld. Deze broncode is verdeeld in een aantal functies. De verdeling van de functies heb ik aangehouden en is terug te zien in de ontwerpen.

Elke functie is verantwoordelijk voor de aansturing van een machineonderdeel. Een functie is opgebouwd uit case select statements. Deze statements kunnen worden weergegeven in een stappenplan omdat er slechts één stap tegelijk actief kan zijn. Een IF statement die als voorwaarde dient om naar de volgende stap over te gaan, kan worden weergegeven als een transitie tussen de stappen in een SFC.

5.5 Prototyping

Aan het einde van beide iteraties in de elaboration fase heb ik het prototype van de user interface gemaakt. Enkele voorbeelden van deze schermen zijn te vinden in het architecture document. Omdat de user interface een touchpanel betreft heb ik deze HMI (Human Machine Interface) schermen genoemd.

Deze schermen dragen bij aan de beeldvorming die een opdrachtgever krijgt van de architectuur van het product. In het prototype worden de knoppen en parameters van de lektester weergegeven. Aan de hand van deze schermen krijgt de opdrachtgever een goed beeld van de acties welke een operator met behulp van het touchpanel uit kan voeren.



6 **Construction fase**

Zowel binnen JB Systems als Fontijne Grotnes worden er bepaalde standaarden gebruikt bij het ontwikkelen van de software. Wat deze standaarden betekenen en op welke punten de software voldoet aan deze standaarden wordt beschreven in het deployment plan.

Het programmeren van de software voor de Siemens PLC wordt gedaan met behulp van het Siemens Step7 softwarepakket. In deze programmeeromgeving is het mogelijk om software te ontwikkelen in drie verschillende programmeertalen.

- LAD (Ladder Diagram)
- FBD (Function Block Diagram)
- STL (Structured Text List)

De syntax van deze programmeertalen zijn vastgelegd in de standaard IEC-61131-3.

Nadat de PLC software bij een klant is opgeleverd, wordt de software op locatie door engineers onderhouden. Doordat een andere partij de code onderhoudt, is het belangrijk om de software zo leesbaar mogelijk te maken. Deze leesbaarheid kan deels worden verkregen door commentaar in de software te verwerken. Om aanpassingen in de software te maken is het noodzakelijk om de werking van de code te begrijpen.

De mate waarin de code leesbaar is, hangt af van de voorkeur van de lezer. Mensen met een electro achtergrond geven de voorkeur aan LAD waar mensen met een electronica of informatica achtergrond de voorkeur aan FBD geven. Over het algemeen wordt STL als slechtst leesbaar beschouwd.

Bij eerdere projecten heb ik geprogrammeerd in FBD, hierdoor gaat mijn persoonlijke voorkeur uit naar FBD. Omdat ik niet met zekerheid kan zeggen dat de persoon op de klantlocatie software in FBD kan lezen zorg ik ervoor dat mijn software in zowel FBD als LAD weergegeven kan worden.

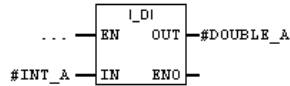
Er zijn echter situaties waarin STL een betere keuze is ten opzichte van FBD. Dit is het geval bij wiskundige berekeningen en delingen.

FB7 : Voorbeeld verschil tussen FBD en STL

Comment:

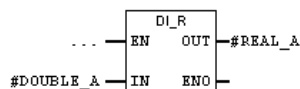
Network 1: converteer Integer_A naar Double_A

Comment:



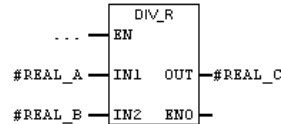
Network 2: converteer Double_A naar Real_A

Comment:



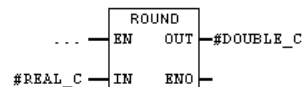
Network 3: deel Real_A door Real_B

Comment:



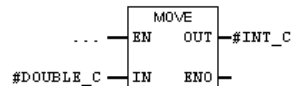
Network 4: rond Real_C af naar Double_C

Comment:



Network 5: move Double_C naar Int_C

door de move operatie wordt de double omgezet naar een int



FBD

FB7 : Voorbeeld verschil tussen FBD en STL **STL**

Comment:

Network 1: Deel INT_A door REAL_B resultaat in INT_C

Comment:

```
L    #INT_A           // Laad Integer_A in acc1
ITD                    // converteer Int naar Double en schrijft in acc1
DTR                    // converteer Double naar Real en schrijft in acc1
L    #REAL_B          // Laad Real_B in acc1 (vorige acc1 schuift naar acc2)
/R                     // Deel acc2 door acc1 en schrijft resultaat naar acc1
TRUNC                  // Trunc kapt de cijfers achter de komma af en schrijft in acc1
T    #INT_C           // schrijf acc1 naar INT_C
```

Figuur 12: Verschil tussen FBD en STL

In de bovenstaande afbeelding wordt het verschil tussen FBD en STL weergegeven bij het uitvoeren van een deling van twee integers. Wanneer er ik een deling wil uitvoeren met behulp van FBD code heb ik ten minste 5 netwerken nodig. Wanneer ik hetzelfde programmeer met behulp van STL heb ik slechts 1 netwerk met hierin 6 instructies nodig. Wanneer de instructies zijn voorzien van commentaar zoals hierboven het geval is, oogt de code in STL overzichtelijker ten opzichte van FBD.



6.1 Van ontwerp naar code

Het ontwerp van het programma overzicht is vrijwel 1 op 1 om te zetten in Step7. De uitzonderingen hierop zijn de onderdelen met een “1 op veel” relatie. Dit betreft de twee functies SendToPCB en ReceiveFromPCB. De reden waarom deze functies een 1 op veel relatie hebben is omdat de functie wordt gebruikt om één threshold te verzenden/ontvangen. Bij vier lekdetectiesensoren is het dus noodzakelijk om de functie vier maal aan te roepen. Dit kan op meerdere manieren worden geprogrammeerd. Enkele manieren zijn:

- Optie 1: Door middel van een counter bijhouden hoe veel thresholds er al zijn verzonden. Iedere keer gebruik maken van hetzelfde functieblok en een bijbehorend datablok.
- Voordeel:* Er zijn minder datablokken nodig omdat het datablok wordt hergebruikt.
- Nadeel:* Bij het monitoren van de software wordt slechts de huidige threshold weergegeven.
- Optie 2: De functie vier maal aanroepen met vier verschillende datablokken.
- Voordeel:* Bij het monitoren van de software worden alle thresholds weergegeven
- Nadeel:* Er zijn meer datablokken nodig omdat iedere threshold een eigen datablok heeft.

Een manier om te zien welke thresholds door de PCB worden gebruikt is door middel van de HMI. Bij het zoeken naar storingen in de software is het wellicht makkelijker om de thresholds die naar de PCB zijn gezonden te monitoren in Step7. In optie 1 wordt slechts de laatst verzonden threshold weergegeven bij monitoring.

Dit betekent dat optie 2 de beste optie is. Wanneer achteraf blijkt dat het programma door deze wijze van programmeren te veel vertraging op loopt kan optie 1 een betere uitkomst bieden.

6.2 Aanroep van functieblokken in OB1

De verschillende functieblokken kunnen op twee manieren worden aangeroepen. Het is mogelijk om alle functieblokken in OB1 aan te roepen. Een andere optie is om de functieblokken aan te roepen zodra deze nodig zijn en vervolgens af te sluiten zodra deze niet meer nodig zijn.

Het voordeel van de eerste optie is dat de signalen van de functieblokken te allen tijde gemonitord kunnen worden. Deze optie verbruikt echter meer resources ten opzichte van de tweede optie.

De functieblokken die in het programma overzicht als device worden geclassificeerd zijn voorzien van alarmen. Deze alarmen treden op wanneer er zich een storing voor doet. Het stappenprogramma van het functieblok main mag slechts doorlopen worden wanneer er geen alarmen actief zijn. Wanneer de functieblokken van de devices niet zijn aangeroepen kunnen deze geen alarm af geven in het geval van een storing. Om die reden is het dus noodzakelijk dat deze functieblokken te allen tijde actief zijn.



6.3 Voorkomen van beschadigingen aan de leksensoren

In het onderzoek naar servo drives heb ik de functionaliteiten van het meest tijdskritische aspect van de lektester onderzocht, namelijk het stoppen van de probe in het geval van een lek.

Omdat de lekdetectiesensoren zo gevoelig zijn, is het noodzakelijk dat de probe zo snel mogelijk wordt gestopt zodra er een lek is gedetecteerd. Om te kunnen garanderen dat de probe binnen een korte tijd wordt gestopt heb ik gebruik gemaakt van een zogeheten Cyclic Interrupt Organization Block.

Een Cyclic Interrupt OB wordt continu met een vaste interval aangeroepen en staat los van andere OB's (inclusief andere cyclic interrupt OB's). Het kortst mogelijke interval hierbij is 10ms, in dit geval wordt OB38 gebruikt. In deze OB heb ik de afhandeling van de leksensoren en het stoppen van de probe geprogrammeerd.

Doordat het stoppen van de probe in het geval van een lek los staat van de besturing van de lektester, heeft een verdere ontwikkeling van de lektester geen negatieve gevolgen op deze functie.

Om te bepalen of deze maatregel passend is, heb ik Fontijne Grotnes gevraagd naar de afstand waarop een lek wordt gedetecteerd. De highflow lekdetectiesensoren zijn in staat om de luchtstroming van een lek op een afstand van 5 millimeter te detecteren.

De bewegingssnelheid van de probe bedraagt 350mm per seconde. Hieruit kan ik opmaken dat de probe 3,5 millimeter af zal leggen in 10 milliseconde.

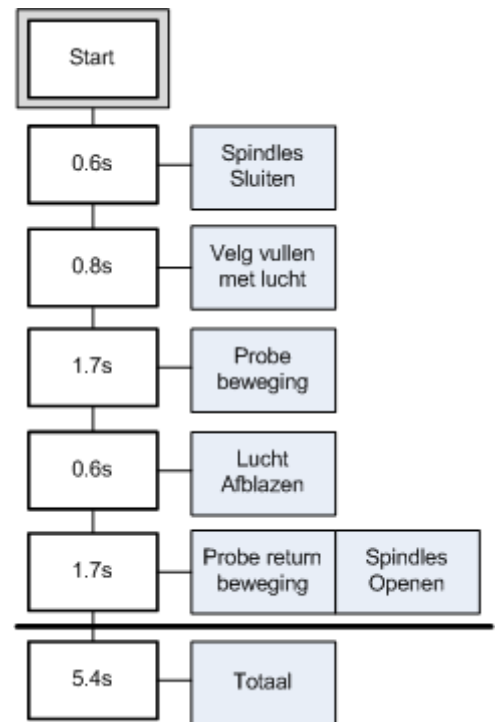
Dit betekent dat een interval van 10milliseconden voldoende is om de probe op tijd stil te kunnen zetten zodat de microflow lekdetectiesensoren niet aan het grote lek blootgesteld zullen worden.

Om te kunnen berekenen of de lektester binnen de tijds marge blijft wanneer deze is voorzien van een PLC heb ik Fontijne Grotnes naar gegevens gevraagd met betrekking tot de benodigde tijden voor het vullen. De enige gegevens die ik heb gekregen zijn de snelheden van de servo drives. De tijd die het lucht vul systeem nodig heeft, hangt af van te veel variabelen om hier een goed antwoord op te kunnen geven. Zo is de temperatuur van de velg en diens omgeving van belang bij het vullen.

In de afbeelding hiernaast is een voorbeeld te zien van de benodigde tijd om de verschillende bewegingen uit te voeren.

De maximale beweging bij het sluiten van de spindles beslaat een afstand van 198 millimeter per spindle. Met een snelheid van 350mm/s betekent dit dat de spindle maximaal 0,565714 seconden nodig heeft om te sluiten, dit heb ik naar boven afgerond op 0.6 seconden. De maximale tijd die nodig is voor het vullen van de velg heb ik af kunnen leiden van de broncode van de CT drive. Hier staat namelijk een timer met een waarde van 700ms. Wanneer deze timer is verstreken wordt de velg als zijnde lek gezien. Hieruit heb ik afgeleid dat de velg doorgaans binnen die tijd wordt gevuld.

Er wordt gebruik gemaakt van een piek filter waardoor er 60ms bij de timer opgeteld moet worden.



Figuur 13: Tijdsduur Lektester

Hiermee komt de totale vultijd op 0,8 seconden. Uitleg met betrekking tot de piek filter wordt gegeven in het Architecture document.

Bij het berekenen van de maximale tijd voor de probe beweging wordt er van het grootste velgtype uitgegaan. Dit betreft een velg voor vrachtauto's en is 595 millimeter hoog. De bewegingsnelheid van de probe is reeds eerder in dit hoofdstuk vastgesteld en bedraagt 350 millimeter per seconde. Door de af te leggen afstand te delen door de snelheid $\frac{595}{350} = 1,7$ krijg ik de tijd die nodig is om de beweging uit te voeren.

De tijd voor het afblazen van de lucht is net als het vullen afgeleid van een timer in de broncode.

De tijd die de probe nodig heeft om terug in de basispositie te komen is vrijwel net zo lang als de tijd die de probe eerder nodig had om de eindpositie te bereiken. Het openen van de spindles vindt tegelijk met het terugsturen van de probe plaats. De benodigde tijd van de probe wordt weergegeven omdat de probe beweging langer duurt t.o.v. het openen van de spindles. Dit alles komt neer op een totale tijd van 5,4 seconden. Deze tijd is ruim binnen de 6 seconden zoals die is opgenomen in de systeemeisen.



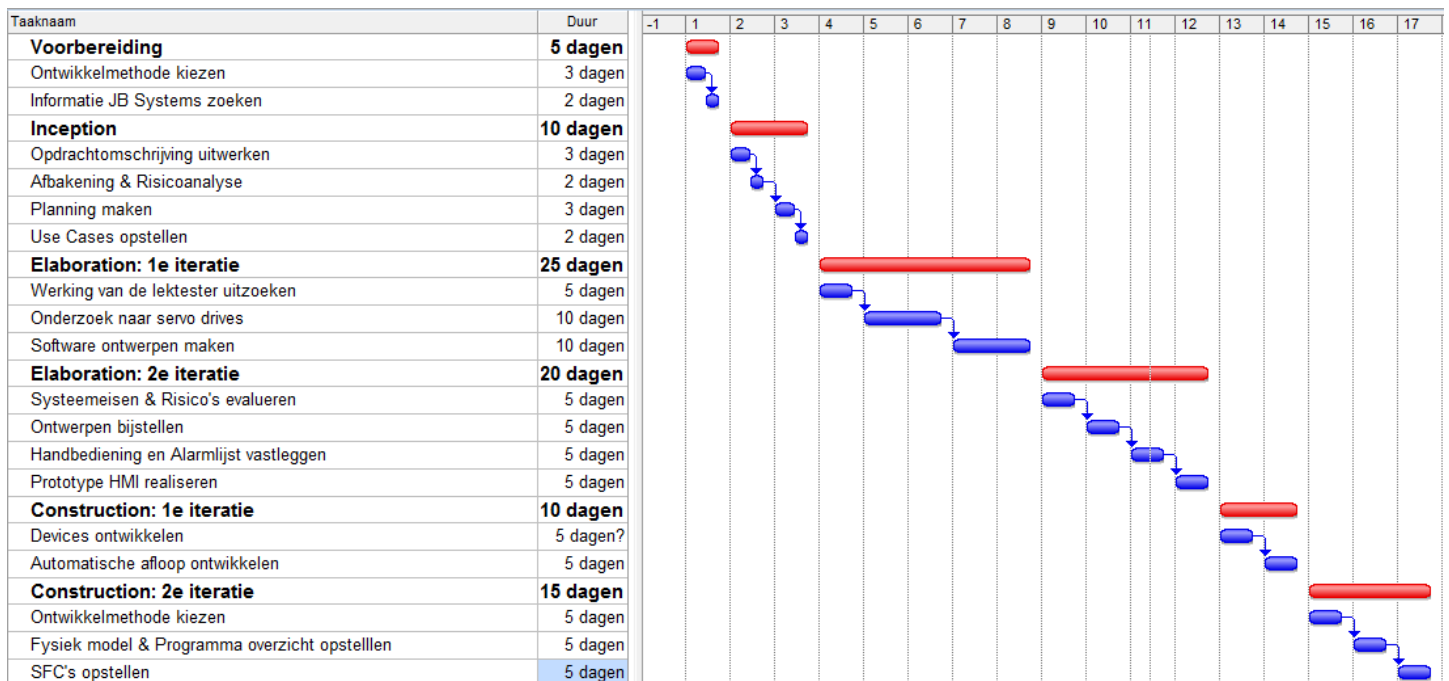
6.4 Proefassessment

In week 14 van het afstudeerproject heeft het proefassessment plaats gevonden. Op dat moment had ik de software ontwikkeling afgerond en was ik bezig met de integration test van de verschillende devices.

Uit het proefassessment is gebleken dat de aantoonbare complexiteit van de beroepstaak ontwerpen van technische informatiesystemen onvoldoende zou zijn wanneer ik hier niets aan zou veranderen.

Ik ben vervolgens met de opdrachtgever in gesprek gegaan om de situatie te bespreken. Ik heb uitgelegd dat het voor de opdrachtgever voordelen oplevert wanneer de software ontwerpen worden verbeterd. Onder andere verhoogt dit de overdraagbaarheid van het project. Omdat ik verwacht alle resterende tijd nodig te hebben om het ontwerp aan te passen, heb ik voorgesteld om de testfase (transition fase) van het project te laten vervallen. De opdrachtgever heeft hier mee ingestemd.

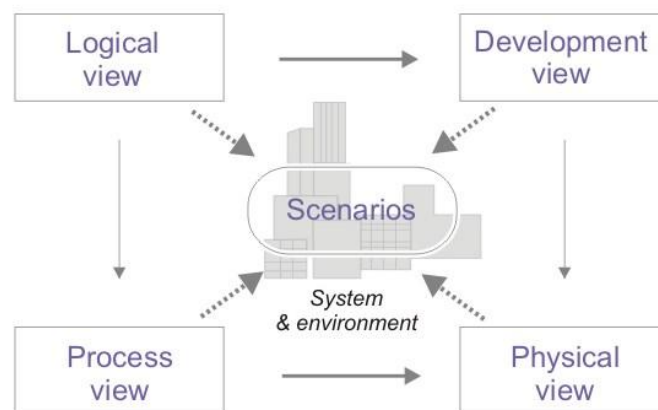
Om de complexiteit van deze beroepstaak te verhogen heb ik een extra iteratie toegevoegd in de construction fase. De onderstaande afbeelding geeft de uiteindelijke planning weer.



Figuur 14: Uiteindelijke planning

6.5 Het opnieuw modelleren van de Lektester

RUP beschrijft een bepaalde definitie van een architectuur. Ik heb hierbij gebruik gemaakt van het *4+1 Architectural view model*. De views worden gebruikt om een weergave te bieden van het beeld dat de verschillende belanghebbenden (stakeholders) bij het systeem kunnen hebben. Met belanghebbenden worden mensen zoals opdrachtgevers, ontwikkelaars en projectleiders bedoeld.



Figuur 15: 4+1 view model

Bron: Wikipedia 4+1 Architectural View Model.jpg

Uit de bovenstaande afbeelding zijn vijf verschillende views op te maken, hier volgt een beschrijving van elke view. Omdat RUP nauw is verbonden met de UML standaard worden de UML varianten genoemd als voorbeeld diagrammen.

Logical view

In de logical view worden de functionaliteiten welke het systeem de eindgebruikers biedt weergegeven. Enkele diagrammen die in deze view kunnen worden gebruikt zijn een klassendiagram, state change diagram en een sequentiediagram.



Development view

Deze view beschrijft de organisatie van statische software modules en is vooral gericht op software ontwikkelaars. Voorbeelden van deze statische software modules zijn broncode, dataopslag en libraries. Deze view wordt echter veelal gebruikt bij software bedoeld voor de PC platform. Hierbij hebben de ontwikkelaars een ruime keuze, in het geval van PLC software is deze keuze veel kleiner. Om die reden is deze view minder relevant voor dit project. Diagrammen die in deze view kunnen worden gebruikt zijn een component diagram en een package diagram.

Process view

De process view biedt een weergave van de dynamische processen binnen het systeem. Hiermee wordt het gedrag van het systeem gemodelleerd. Het diagram dat hiervoor wordt gebruikt is een activiteiten diagram. Het activiteitendiagram wordt gebruikt om de genomen stappen bij het uitvoeren van een use case weer te geven.

Physical view

De physical view wordt gebruikt om de software componenten op de hardware laag weer te geven alsmede de koppelingen tussen deze componenten. Hierbij kan worden gedacht aan een serie servers waaronder applicatie, database en opslag servers. Het diagram dat hiervoor wordt gebruikt is een deployment diagram.

Scenarios

De view scenarios geeft een weergave van de interactie tussen de eindgebruiker en het systeem. Het diagram dat hiervoor wordt gebruikt betreft een use case diagram.

Niet alle views zijn relevant bij het samenstellen van de architectuur van de lektester. Zoals hierboven beschreven is de deployment view niet van toepassing op PLC software ontwikkeling. Om die reden zal ik de diagrammen die binnen deze view vallen, niet in de architectuur op nemen.

Het activity diagram welke wordt besproken in de process view is een goede toevoeging aan de architectuur wanneer het systeem een complexe gebruikersinterface betreft. Het is niet interessant om de gebruikersinterface in een activity diagram op te nemen omdat het in de meeste gevallen slechts één druk op een knop betreft.

De diagrammen die ik ga gebruiken om de architectuur weer te geven zijn:

- Use Case diagram
- Deployment diagram
- State Change diagram
- Class diagram
- Sequence diagram

De interactie tussen de gebruiker en het systeem wordt beschreven in de use cases. Een overzicht de beschikbare use case kan worden gevonden in het use case diagram.

In het deployment diagram is de aanwezige hardware van de lektester opgenomen. De weergegeven hardware in dit diagram is gegroepeerd in onderdelen.

Het state change geeft een weergave van de standen waarin het systeem kan opereren.

De relatie tussen de verschillende software objecten zoals datablocken en functies wordt uitgebeeld in een class diagram.

De werking van de verschillende functies wordt weergegeven in de sequence diagrammen.



6.5.1 Het bepalen van de tekentechniek

UML is de huisstandaard van RUP, om die reden lijkt het logisch om UML in dit project te gebruiken. UML is echter voornamelijk gericht op gebruik bij objectgeïntendeerde software.

Een andere veel gebruikte tekentechniek bij het modelleren van PLC software is het gebruik van Sequential Function Charts. Deze tekentechniek is vastgelegd in de standaard IEC 61131-3. Deze standaard is opgesteld door de IEC, de (International Electrotechnical Commission). In deze standaard zijn ook de programmeertalen FBD (Function Block Diagram), ST (Structured Text) en Ladder opgenomen.

Omdat SFC's veel worden gebruikt bij het modelleren van PLC software lijkt dit de juiste tekentechniek voor gebruik in dit project. Er zit echter een nadeel aan het gebruik van deze techniek. Met behulp van SFC's is het alleen mogelijk om stappenplannen de modelleren. Bij het modelleren van de architectuur van het systeem zijn er meer modellen nodig dan stappenplannen alleen.

Geen van beide standaarden biedt dus een complete oplossing. Om deze reden heb ik bij elk diagram bepaald welke standaard het beste is toe te passen.

In de onderstaande afbeelding staan de mogelijkheden afgebeeld. De diagrammen die uiteindelijk zijn gebruikt, zijn in de afbeelding onderstreept. In de volgende paragrafen wordt er per diagram beschreven waarom er voor het diagram is gekozen.

UML		ISA-88 / IEC 61131-3 (SFC)
<u>Use Case diagram</u>	↔	< geen diagram beschikbaar >
Deployment diagram	↔	
Component diagram	↔	<u>Fysiek model</u>
Class diagram	↔ <u>Programma Overzicht</u> ↔	< geen diagram beschikbaar >
Sequence diagram	↔	<u>Sequential Function Chart</u>
<u>Machine State diagram</u>	↔	< geen diagram beschikbaar >

Figuur 16: Gebruikte diagrammen

6.5.2 Het opstellen van Use Cases

In de inceptionfase heb ik een aantal use cases opgesteld. Deze use cases hadden allen betrekking op de bediening van de lektester door middel van het bedieningspaneel met de drukknoppen. Naast het bedieningspaneel kan de lektester tevens worden bediend door middel van een HMI Touchscreen. De bediening met de HMI heeft toegevoegde functionaliteiten ten opzichte van het bedieningspaneel. Deze toegevoegde functionaliteiten zijn in de elaborationfase in de use cases verwerkt.

Naast de broncode heb ik tevens de bestanden gekregen van de Exor HMI die aan de lektester is verbonden. Aan de hand van de broncode van de software heb ik de use case beschrijvingen in kunnen vullen. De aannamen van de use case beschrijvingen heb ik afgeleid van de gebruikte if statements in de broncode.

De beschrijvingen heb ik afgeleid van de functies die in de broncode worden uitgevoerd. Uitzonderingen in de use cases heb ik vastgesteld door te kijken naar if statements binnen de betreffende functies.



6.5.3 Een model van de hardwareconfiguratie

De tweede stap bij het modelleren van de lektester, is het maken van een diagram waarin de hardwareconfiguratie wordt weergegeven. Hierbij heb ik gebruik gemaakt van de standaard ISA-88. Het fysieke model van ISA-88 is de tegenhanger van het deployment diagram in UML.

Een deployment diagram wordt gebruikt om de hardware configuratie weer te geven waarop de software wordt geïnstalleerd. Het diagram wordt opgebouwd uit verschillende hardware componenten, deze componenten worden vervolgens met elkaar verbonden om een representatie van de hardware configuratie te vormen. Deze componenten kunnen in UML worden vastgelegd door middel van een component diagram, dit diagram wordt echter gebruikt om het bundelen van functies weer te geven. De focus van deze diagrammen ligt echter bij het software aspect terwijl het gewenste diagram een weergave van de hardwareconfiguratie dient zijn

In het fysieke model van ISA-88 worden de verschillende hardware componenten net als in het deployment diagram met elkaar verbonden. De ISA-88 standaard beschrijft echter ook een wijze waarop de hardware gegroepeerd of ingedeeld kan worden. Er wordt binnen de standaard gebruik gemaakt van verschillende lagen in een fysiek ontwerp. Er kan een onderscheid worden gemaakt tussen vier lagen:

- Proces cell
- Unit
- Equipment
- Control module

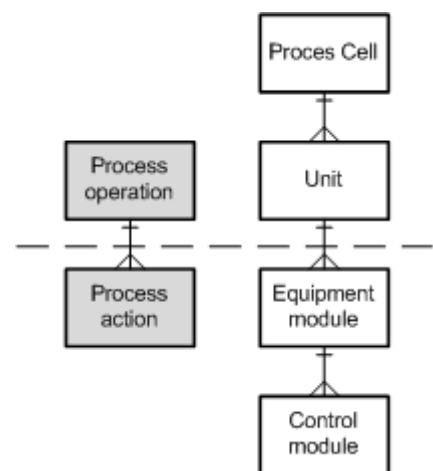
Een proces cell bevat alle benodigde equipment modules voor het maken van één batch. In het geval van de lektester kan het verwerken van één velg als batch worden gezien.

Een unit voert een actie uit op één batch of een gedeelte van één batch. Een unit is onafhankelijk van andere units.

Een equipment module kan een proces actie uitvoeren. Een control module kan een sensor of een actuator voorstellen.

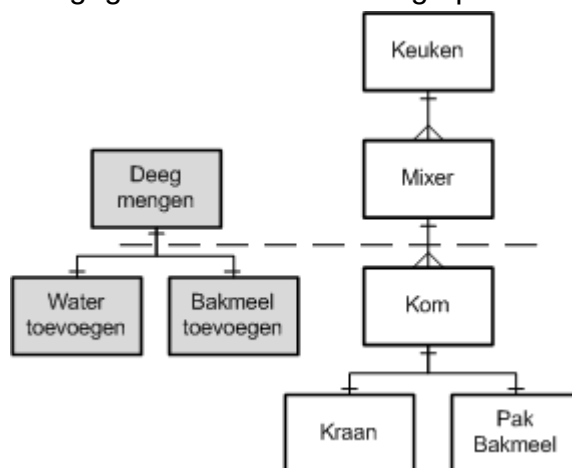
Een proces operation resulteert in een verandering van het te verwerken materiaal. Het testen van de velg resulteert niet in een fysieke verandering van de velg. De status van de velg verandert wel, na het testen is de velg namelijk wel of niet lek. Het testen van de velg wordt om die reden als proces operation beschouwd.

Een proces operation is opgebouwd uit proces acties.



Figuur 17: Process vs fysiek

Het bovenstaande diagram geeft de verhouding tussen process-zijde en de fysieke-zijde weer. De process-zijde welke aan de linkerkant wordt weergegeven heeft betrekking op de acties die worden uitgevoerd om een product te realiseren. De fysieke-zijde welke aan de rechterkant wordt weergegeven heeft betrekking op de indeling van de hardware configuratie.



Figuur 18: Voorbeeld process vs fysiek

De onderbroken lijn geeft het verschil aan van de mate waarin de het onderdeel gevolgen heeft voor de batch. De onderdelen onder de lijn hebben nauwelijks invloed op de batch. Wanneer deze onderdelen wordt gebundeld tot één van de onderdelen van boven de lijn, kan dit grotere gevolgen hebben voor het verwerken van een batch.

In vergelijking met het deployment diagram van UML biedt het fysieke model van ISA-88 meer structuur en duidelijkere regels voor het indelen van de hardware. Om deze reden heb het fysieke model gebruikt voor het modelleren van de hardware configuratie.

Er zijn twee manieren om de groepering van de hardwareconfiguratie te bepalen. Er is de top-down aanpak en de bottom-up aanpak.

Wanneer de top-down aanpak wordt gebruikt, wordt het systeem stap voor stap ontworpen. Als eerste wordt de bovenste laag bepaald. In het geval van de lektester wordt de proces cell als eerste bepaald. Vervolgens worden de onderliggende units aangewezen, daarna de equipment modules en als laatste worden de control modules aan een equipment module gekoppeld.

In het geval van de bottom-down aanpak wordt er eerst een inventarisatie gemaakt van de onderdelen welke zich in de onderste laag bevinden. Deze onderdelen worden vervolgens gegroepeerd in bovenliggende equipment modules, enzovoorts. Deze aanpak is een goede keuze wanneer het een bestaand systeem betreft omdat de kans dat er onderdelen worden vergeten kleiner is.

Om te bepalen welke van deze manieren de beste is in het geval van de lektester, kijk ik eerst naar de definities van de proces cell en de unit. Volgens de definitie van de proces cell, is de lektester als geheel een proces cell.

In de definitie van een unit staat dat de units onafhankelijk van elkaar moeten kunnen werken.

Wanneer ik kijk naar de hardware van de lektester, zie ik dat de hardware componenten van elkaar afhankelijk zijn. Zo is het lucht vulsysteem afhankelijk van de spindles om de velg luchtdicht vast te klemmen. De lekdetectie sensoren zijn weer afhankelijk van de slidingdoors om het luchtstroming van buiten de lektester te blokkeren. Dit heeft als gevolg dat het enige onderdeel dat als unit kan worden aangemerkt de lektester als geheel is.

Omdat de verschillende onderdelen zo afhankelijk van elkaar zijn, heb ik besloten om de bottom-down aanpak te gebruiken. Hierdoor is de kans dat ik een essentieel onderdeel in het fysiek ontwerp vergeet op te nemen het kleinst.

6.5.4 Programma overzicht

De relaties tussen de verschillende onderdelen van een programma worden in UML gemodelleerd door middel van een klassendiagram. Een soortgelijk diagram wordt in zowel IEC 61131-3 als ISA-88 niet genoemd.

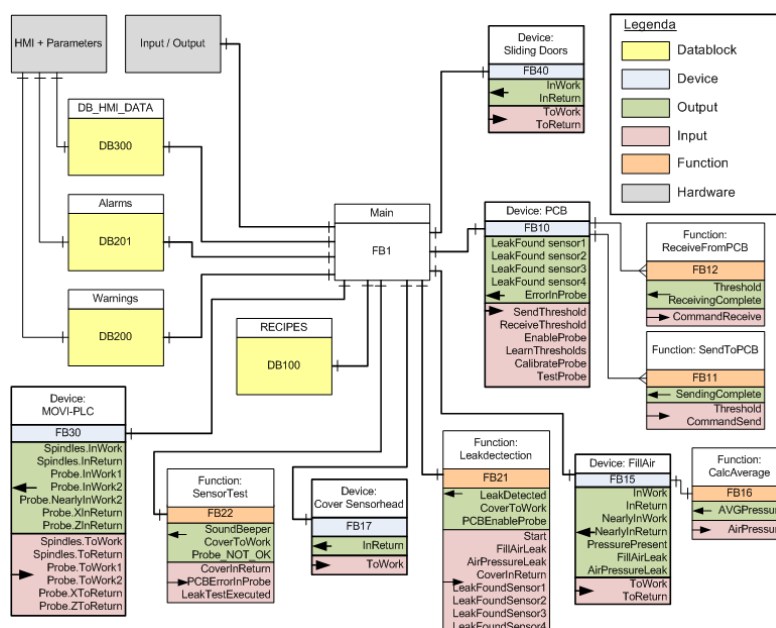
Een klassendiagram is er echter op gebouwd om de attributen en operaties binnen een klasse weer te geven. In Step7 PLC software bestaan er geen klassen met attributen of operaties, daarentegen bestaan er wel functieblokken met zowel inputs als outputs. De input signalen van een functieblok kunnen worden gebruikt om operaties door een functieblok uit te laten voeren. De output signalen kunnen de gevolgen van deze operaties terugkoppelen.



De multipliciteit van de relaties tussen de object wordt weergegeven op de manier die bij ISA-88 gebruikelijk is.

Om de koppeling tussen het fysieke model en het programmaoverzicht te behouden, is er een verschil tussen een device en een function aangebracht. De onderdelen welke in het fysieke model als equipment module worden beschreven, worden in het onderstaande diagram als device gekenmerkt. De onderdelen die als function worden gekenmerkt, dienen als ondersteuning voor een device.

De functions ReceiveFromPCB en SendToPCB worden gebruikt om één threshold af te handelen. Om deze reden wordt de relatie tussen deze functions en de device PCB als 1 op veel gemarkeerd.



Figuur 19: Programma Overzicht

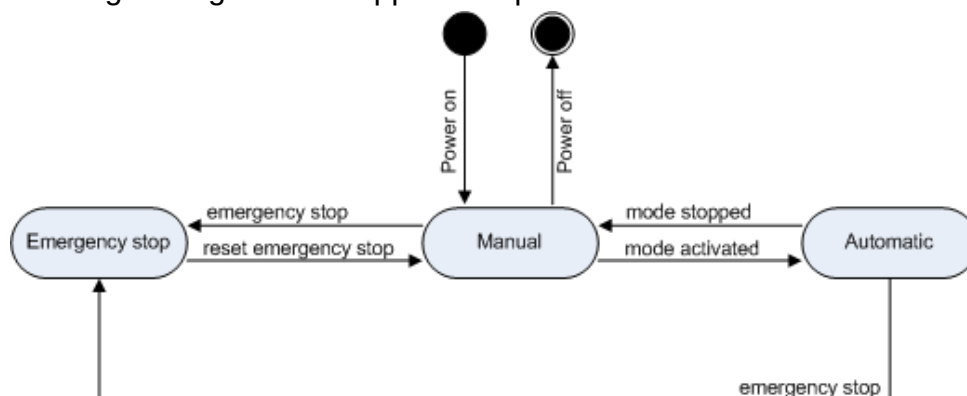


6.5.5 Toestandsdiagram

Nu de structuur van de software in de modellen is vastgelegd, is het tijd om het gedrag van de software te modelleren. Om het gedrag van de software te modelleren maak ik gebruik van twee soorten diagrammen. Een toestandsdiagram wordt gebruikt om de toestanden weer te geven waarin het systeem kan verkeren. Een ander model om het gedrag van de software te modelleren is een stappenplan. In dit stappenplan worden de inputs en outputs van het programmoverzicht in een vaste volgorde aangeroepen.

Uit de use cases is gebleken dat de lektester een aantal operationele modi kent. Deze verschillende modi hebben een grote invloed op de aansturing van de lektester. Wanneer de lektester in Manual mode staat, mogen er geen machineonderdelen bewegen wanneer de operator hier geen knop voor in drukt. Zodra de emergency stop wordt geactiveerd, mogen er helemaal geen machineonderdelen bewegen.

Wanneer de machine in automatic mode staat kunnen de machineonderdelen bewegen volgens de stappenafloop.

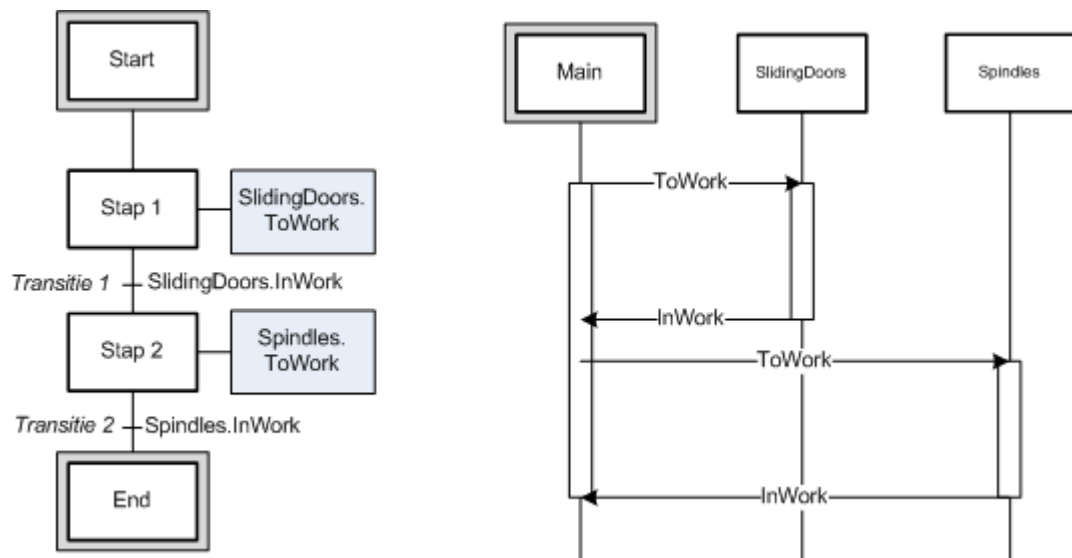


Figuur 20: Toestandsdiagram

ISA-88 kent geen diagram waarin deze modi weergegeven kunnen worden. UML biedt een toestandsdiagram waarin deze modi gemodelleerd kunnen worden.

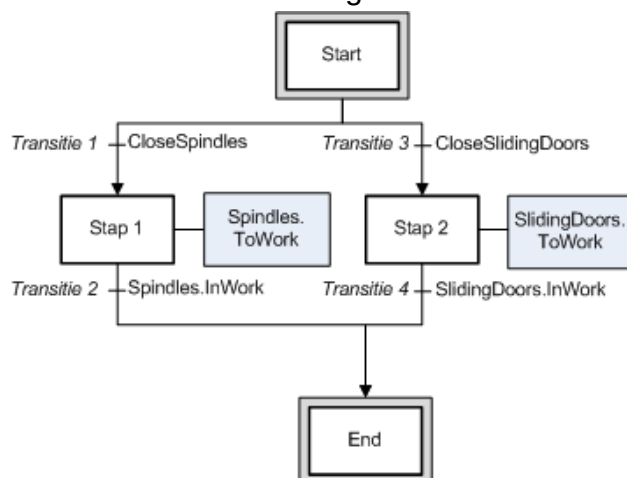
6.5.6 Een model van de stappenafloop

Zowel UML als ISA-88 bieden de mogelijkheid om het gedrag van de software te modelleren. In ISA-88 wordt dit gedaan door middel van SFC's (Sequential Function Charts). In UML wordt hiervoor gebruik gemaakt van Sequence diagrams. In het onderstaande voorbeeld wordt aan de linkerkant een SFC weergegeven, aan de rechterkant is een sequence diagram te zien.



Figuur 21: Voorbeeld SFC en Sequence diagram

In een stappenprogramma met slechts één mogelijke afloop kunnen beide diagrammen worden gebruikt. Wanneer het stappenprogramma echter meerdere mogelijke aflopen bevat, kan dit met het sequence diagram niet worden gemodelleerd. In een SFC wordt dit gemodelleerd volgens de onderstaande afbeelding.



Figuur 22: Voorbeeld SFC met meerdere mogelijke aflopen



In het geval van de lektester zal de software meerdere mogelijke aflopen bevatten. De uit te voeren acties veranderen namelijk zodra er een lek wordt gedetecteerd. Om die reden heb ik gebruik gemaakt van SFC's om de lektester te modelleren.

De lektester is verdeeld in verschillende onderdelen (volgens ISA-88) en niet alle onderdelen zullen door middel van SFC's gemodelleerd worden. Het is bijvoorbeeld niet interessant om de aansturing van een pneumatische klep in een SFC te modelleren omdat de enige stappen, het open en dicht sturen van de klep zijn. De onderdelen die wel interessant zijn om te modeleren zijn de onderdelen in de lagen van equipment modules en hoger.

In alle equipment modules worden ten minste één of meerdere control modules aangestuurd. In deze gevallen is het wel interessant om deze aansturing in een SFC te modelleren, omdat deze control modules samen de aansturing van de equipment module vormen.

In de Unit worden de verschillende equipment modules aangestuurd. Ook deze aansturing kan worden weergegeven in een SFC. Naast de equipment modules en de unit kunnen de functions ook in een SFC worden weergegeven.

De equipment modules en functions worden door middel van een SFC gemodelleerd met een stappenplan. Dit wordt ook wel een sequentieel ontwerp genoemd. Het is echter de bedoeling dat deze onderdelen volgens een conditionele wijze worden geprogrammeerd.

Wanneer de equipment modules conditioneel zijn geprogrammeerd en de spanning wordt onderbroken, zal de aansturing van de hardware door middel van de equipment module niet automatisch worden hervat. Bij gebruik van een stappenplan wordt de aansturing wel automatisch hervat. Dit besturing wordt automatisch hervat omdat de status van het stappenplan in het geheugen van de PLC wordt opgeslagen.

7 **Evaluatie**

In dit hoofdstuk wordt er eerst een beschrijving gegeven van de evaluatie van het product en de deelproducten. Daarna volgt er een evaluatie van het proces. Ten slotte worden de beroepstaken beschreven.

7.1 **Productevaluatie**

Onderzoeksrapport

Het onderzoeksrapport geeft een goed beeld waar en hoe servo drives worden toegepast. Ik heb bewust gekeken naar de toepassing van servo drives en ben niet diep ingegaan op de werking van servo drives. Ik vind dat dit goed is gelukt, doordat het onderzoeksrapport niet vol staat met technische feiten en specificaties is het document ook goed te lezen voor mensen met een minder technische achtergrond.

Ik ben tevreden over de wijze waarop de verschillende hardwareplatformen zijn beschreven. De opdrachtgever heeft aangegeven het overzicht netjes en overzichtelijk te vinden.

Zelf zou ik graag meer onderzoek gedaan hebben naar de wijze waarop een servo drive werkt. Momenteel weet ik hoe ik een servo drive kan gebruiken maar kan ik de servo drive niet ten volle benutten.

Architecture document

In het architecture document is een uitgebreide analyse van de huidige situatie beschreven. Tijdens het maken van de software ontwerpen heb ik vaak terug gebladerd naar dit hoofdstuk.

Aan de hand van het architecture document heb ik de software ontwikkeld. Aanvankelijk heb ik hier veel moeite mee gehad door het gebrek aan goede ontwerpen. Na het proefassessment heb ik een beter beeld gevormd van de wijze waarop de software ontworpen zou moeten worden. De wijzigingen die ik in de software heb doorgevoerd aan de hand van de nieuwe ontwerpen verliepen vlekkeloos. De vertaling van ontwerp naar software is goed verlopen.



Software

De software is tijdens de ontwikkeling getest door middel van unit tests en een integration test. Hiermee kan ik zeggen dat de software functioneert volgens het ontwerp. Vanwege de beperkte tijd heb ik de software niet verder kunnen testen.

In verdere tests zouden er situaties ontdekt kunnen worden waar het systeem niet op berekend is zoals meerdere knoppen tegelijk in drukken. Doordat er geen tests aan de hand van de use cases is uitgevoerd, kan ik niet zeggen of de software aan de eisen van de opdrachtgever zal voldoen.

7.2 Procesevaluatie

Planning

Bij het opstellen van het afstudeerplan heb ik een planning gemaakt op basis van mijn kennis en ervaring op het gebied van PLC projecten. Beide waren niet erg groot, wat resulteerde in een onmogelijk haalbare planning. In het vervolg zal ik ruimer plannen. Naast een ruimere planning zal ik meer meetpunten aanbrengen. Op deze meetpunten zal ik kijken naar hoe lang ik had verwacht over een activiteit te doen en hoe lang ik er daadwerkelijk over heb gedaan. Op basis van deze informatie kan ik vervolgens kiezen om gerelateerde activiteiten in te korten of te verlengen.

Gebruik van iteraties

In de loop van de Inception en de elaboration fase is mijn kennis van de lektester en de gebruikte hardware aanzienlijk gegroeid. Hierdoor ben ik in staat geweest om een beter haalbare planning op te stellen. Door iteraties toe te passen op basis van verkregen kennis en veranderende systeemeisen, is de keuze voor een iteratieve projectmode een goede keuze gebleken.

Doelstellingen

In dit project heb ik met meerdere doelstellingen gewerkt. De opdrachtgever heeft een doelstelling maar ik heb zelf ook een doelstelling. Mijn doelstelling is het kunnen verantwoorden van de beroepstaken. Na het proefassessment is gebleken dat ik mijn eigen doelstelling niet zou kunnen behalen als ik niet zou ingrijpen.

Ik ben tevreden met de wijze waarop ik in deze situatie heb gehandeld waardoor zowel de opdrachtgever tevreden gesteld kon worden en ik mijn beroepstaken heb kunnen verantwoorden.

7.3 Epiloog

Het project is begonnen als een sandbox-opdracht, ontstaan vanuit een bekende ergernis van een klant van JB Systems. De lektester in de huidige staat (met CT drives) werkt niet geheel naar wens, er komen regelmatig storingen voor.

In de laatste twee weken van het project is de ergernis veranderd in een probleem. Met andere woorden, mijn sandbox-opdracht werd een echte opdracht.

Er heeft een gesprek met de opdrachtgever plaats gevonden, hierin zijn de mogelijke oplossingen met betrekking tot het hardwareplatform opnieuw doorgesproken. Uit dit gesprek is gebleken dat de opdrachtgever bereidt is om de genoemde standaard los te laten en de voorkeur geeft aan een hardwareplatform waarbij alle onderdelen van één fabrikant afkomstig zijn.

De verkopers van JB Systems hebben een offerte uitgebracht voor de ontwikkeling van de software voor de lektester. De opdrachtgever heeft deze offerte inmiddels geaccepteerd. Dit heeft tot resultaat dat niet alleen nieuwe lektesters maar ook bestaande lektesters worden uitgerust met SEW hardware. Het project wordt door JB systems uitgevoerd met mijn project documentatie als fundament.

Deze documentatie is door ervaren software engineers van JB Systems doorgenomen. Op deze documentatie heb ik goede feedback gekregen, de documentatie bevatte volgens deze software engineers alle informatie die zij nodig hebben om het project uit te voeren. Hieruit maak ik op dat de documenten die ik in de loop van het project heb opgeleverd voldoende relevante informatie bevatten. Voldoende informatie om een vervolgproject uit te kunnen voeren met mijn documentatie als uitgangspunt.



7.4 Beroepstaken

In dit hoofdstuk beschrijf ik de beroepstaken die ik bij aanvang van het afstudeerproject heb gekozen en welke in het afstudeerplan zijn opgenomen.

Bij elke beroepstaak beschrijf ik op welke wijze aan de toetscriteria wordt voldaan.

Niveaubepaling van de beroepstaken

Ik heb het niveau van de beroepstaken bepaald met behulp van de vragenlijst in het Competentiemodel Academie ICT en Media. Het aantal vragen waarop ik “ja” kan antwoorden geven een indicatie van de complexiteit van het afstudeerproject.

Ik heb gesprekken gevoerd met de opdrachtgever, bedrijfsbegeleider en de engineer van de opdrachtgever. Uit deze gesprekken heb ik de eisen / wensen opgesteld. Doordat uit deze gesprekken is gebleken dat deze mensen verschillende eisen en wensen hebben, kan ik de vraag met nummer 3 als “ja” beantwoorden.

Op basis van het resultaat van dit afstudeerproject heeft JB Systems de klant een offerte voor de ontwikkeling van de software voor de lektester aangeboden. De klant heeft deze offerte geaccepteerd. Dit betekent dat de bestaande lektesters van SEW hardware worden voorzien. Hierdoor was er met deze opdracht veel geld gemoeid en zullen er veel mensen met mijn werk te maken krijgen. Om die reden kan ik vragen nummer 8 en 9 met “ja” beantwoorden.

In de opleiding Technische Informatica heb ik als actuatoren gebruik gemaakt van elektromotoren met een laag vermogen. In de lektester wordt er gebruik gemaakt van servo drives om servomotoren aan te sturen. Door middel van een onderzoek heb ik kennis opgedaan op het gebied van servo drives en is er een spreekwoordelijke wereld open gegaan. Om die reden ben ik van mening dat ik de vraag met nummer 15 met “ja” kan beantwoorden.

Enkele voorbeelden van de vaktechnische ingewikkeldheden die ik in de loop van dit project ben tegengekomen zijn:

- De positionering van de probe, het gekoppeld aansturen van de X en Z as.
- De communicatie van en naar de PCB.

Deze vaktechnische ingewikkeldheden heb ik geanalyseerd vanuit de CT broncode. Deze analyse heb ik uitgewerkt in diagrammen en vervolgens in Step7 geïmplementeerd. Vanwege deze vaktechnische ingewikkeldheden heb ik vraag nummer 16 met “ja” beantwoord.

De volgende vragen heb ik met “ja” beantwoord:

- 3 Je hebt meerdere (informele) opdrachtgevers die allen andere eisen/wensen kunnen hebben.
- 8 Er is veel geld gemoeid met deze opdracht.
- 9 Er zullen veel mensen te maken krijgen met mijn werk.
- 15 Goh, er gaat een wereld voor me open.
- 16 Er zitten allerlei vaktechnische ingewikkeldheden in deze opdracht.

Doordat ik meer dan vier vragen met “ja” heb beantwoord, wordt de context als lastig geclassificeerd.

Een andere factor in de niveaubepaling is de mate van zelfstandigheid binnen een project. De taakrol die ik tijdens het afstudeerproject heb vervuld is de rol zelfstandig omdat het afstudeerproject door mij alleen is uitgevoerd.

		Taakrol		
		geleid	zelfstandig	sturend
Context	simpel	1	2	3
	lastig	2	3	4
	complex	3	4	5

Figuur 23: Niveaubepaling beroepstaken

Uit de bovenstaande diagram valt op te maken dat de competenties binnen dit afstudeerproject op niveau 4 liggen.



C8 - Ontwerpen van een technisch informatie systeem

Aan de hand van de gekozen systeem ontwikkelmethode heb ik vastgesteld wat een architectuur inhoudt. Vervolgens heb ik een opsomming gemaakt van de beschikbare diagrammen om de architectuur weer te geven. Aan de hand van deze opsomming heb ik een afweging gemaakt van de typen diagrammen die op de architectuur van de lektester van toepassing zijn, hierbij heb ik tevens de motivatie achter deze afwegingen beschreven.

Zodra ik een keuze heb gemaakt in de te gebruiken diagrammen, heb ik gekeken naar de beste tekentechniek om deze diagrammen in te vullen. Hierbij heb ik de tekentechnieken UML en de ontwerp filosofie ISA-88 met elkaar vergeleken. Bij elk diagram heb ik een afweging gemaakt voor de toe te passen tekentechniek.

Door deze afwegingen te maken bij het ontwerpen van de software voor de lektester heb ik gedemonstreerd dat het ontwerp goed doordacht en doelbewust tot stand is gekomen.

C13 – Het betrekken van real-time aspecten bij een ontwerp

Bij het demonstren van deze beroepstaak heb geen optimalisatie toegepast, in plaats daarvan heb ik een beveiligingsmaatregel die afhankelijk is van real-time aspecten gerealiseerd. Deze beveiliging wordt gebruikt de probe tijdig te stoppen zodat de sensoren niet worden beschadigd door ontsnappende lucht. Door middel van een berekening heb ik aangetoond dat deze beveiligingsmaatregel treffend is en de probe op tijd laat stoppen. Hierbij heb ik rekening gehouden met de bewegingssnelheid van de probe en de afstand waarop lekken kunnen worden gedetecteerd. Deze berekening is te vinden in de beschrijving van de constructie fase.

D17 – Testen van Software systemen

Deze beroepstaak heb ik niet uit kunnen voeren. Om toch drie beroepstaken te beschrijven heb ik hier onder een extra beroepstaak beschreven.

B6 – Selecteren van een platform inclusief besturingssysteem

Tijdens het onderzoek naar servo drives heb ik een viertal hardwareplatformen samengesteld. Hiervoor heb ik contact opgenomen met zowel Siemens als Vector om de relevante specificaties met betrekking tot deze platformen te achterhalen.

Ik heb een aantal criteria opgesteld op basis van de eisen en wensen van de opdrachtgever. Op basis van deze criteria heb ik de opdrachtgever een advies gegeven met betrekking tot een voorkeursplatform.



8 Literatuurlijst

Boeken:

- The Rational Unified Process an Introduction van Philippe Kruchten (EN)
Pearson Education – 3^e druk – 2004
- Praktisch UML van Jos Warmer & Anneke Kleppe (NL)
Pearson Education – 3^e druk – 2009

Manuals:

Control Techniques

- Unidrive SP Advanced User Guide (EN)
- UD70 Manual (EN)

Siemens

- Statement List (STL) for S7-300 and S7-400 programming (EN)
- Simatic S7-300 CPU 31xT

Vector (SEW)

- MoviDrive MDX60B / 61B – Technische Handleiding (NL) – Edition 01/2010
- MoviDrive MDX60B / 61B – System Manual (EN) – Edition 09/2010
- MOVI-PLC AxisControl Sample Project (EN) - Edition 05/2008
- MOVI-PLC advanced DH.41B – Path Interpolation via the Program Module AxisGroupControl Kinematics (EN) – Edition 01/2010

Websites:

www.wikipedia.com
www.google.nl
www.vector.nu
automation.siemens.com
www.controltechniques.nl
www.dehaagsehogeschool.nl/bibliotheek
www.trefwoordenthesaurus.nl

9 **Figuurlijst**

<i>Figuur 1: Bedrijfsconstructie</i>	<i>4</i>
<i>Figuur 2: Stappen bij de productie van een velg</i>	<i>6</i>
<i>Figuur 3: Stappen bij een lectest</i>	<i>8</i>
<i>Figuur 4: Aanwezige hardware onderdelen lectester</i>	<i>10</i>
<i>Figuur 5: Conclusie keuze in ontwikkelmethoden</i>	<i>18</i>
<i>Figuur 6: Conclusie keuze in ontwikkelmethoden met weging</i>	<i>19</i>
<i>Figuur 7: Iteratieplan.....</i>	<i>23</i>
<i>Figuur 8: Eerste planning.....</i>	<i>24</i>
<i>Figuur 9: Eerste wijziging in de planning.....</i>	<i>26</i>
<i>Figuur 10: Te volgen profiel</i>	<i>28</i>
<i>Figuur 11: Voorbeeld botsing probe op velg</i>	<i>30</i>
<i>Figuur 12: Verschil tussen FBD en STL.....</i>	<i>38</i>
<i>Figuur 13: Tijdsduur Lectester</i>	<i>42</i>
<i>Figuur 14: Uiteindelijke planning.....</i>	<i>43</i>
<i>Figuur 15: 4+1 view model.....</i>	<i>44</i>
<i>Figuur 16: Gebruikte diagrammen</i>	<i>47</i>
<i>Figuur 17: Process vs fysiek.....</i>	<i>50</i>
<i>Figuur 18: Voorbeeld process vs fysiek</i>	<i>50</i>
<i>Figuur 19: Programma Overzicht.....</i>	<i>52</i>
<i>Figuur 20: Toestandsdiagram.....</i>	<i>53</i>
<i>Figuur 21: Voorbeeld SFC en Sequence diagram.....</i>	<i>54</i>
<i>Figuur 22: Voorbeeld SFC met meerdere mogelijke aflopen.....</i>	<i>54</i>
<i>Figuur 23: Niveaubepaling beroepstaken</i>	<i>60</i>

