

Project Pharmafilter

Afstudeerverslag - Versie 3.0

Joey Verseveldt

Technische Informatica
De Haagse Hogeschool
Rotterdamseweg 137
2628 AL Delft

J.P.Verseveldt@student.hhs.nl

2 juni 2016

Voorwoord

Dit document is geschreven naar aanleiding van de afstudeerperiode van 17 weken die plaats heeft gevonden van 8 februari 2016 tot en met 3 juni 2016 bij Siemens Nederland locatie Den Haag. De afstudeerstage is uitgevoerd als onderdeel van de opleiding Technische Informatica. Op deze manier zou ik graag een aantal mensen willen bedanken.

Ten eerste wil ik graag Rob Winkels bedanken voor het begeleiden en actief meehelpen met mijn afstudeerperiode. Hij heeft mij constant gesteund en aangestuurd wanneer dit nodig was. Daarnaast is hij degene die mij uiteindelijk de specifieke opdracht heeft toegewezen. Hiernaast hoort ook dank aan Jaap Bruining, die mij heeft aangenomen als afstudeerder binnen de afdeling Engineering & Commissioning binnen Siemens. Ook Wim Tiebout wil ik graag bedanken voor het zoeken binnen het bedrijf naar een afstudeerplek, na het succesvol afronden van mijn stage op zijn afdeling.

Verder zou ik graag Tony Andrioli bedanken voor het begeleiden en feedback geven tijdens mijn afstudeerperiode. Dit heeft mij geholpen om uiteindelijk dit verslag te kunnen inleveren.

Ten slotte wil ik alle andere werknemers van Siemens bedanken, waarbij in het bijzonder Lowiek van Velzen en Anton van der Lubbe, die mij tijdens het afstuderen hebben geholpen met het uitvoeren van het project. Ik wil iedereen graag bedanken voor een gezellige, leuke, maar vooral erg leerzame tijd op de afdeling.

Den Haag, 2 juni 2016
Joey Verseveldt

Inhoudsopgave

Voorwoord	i
1 Inleiding	1
2 Organisatie	2
2.1 Organisatie algemeen	2
2.2 Project	2
2.3 Plaats van afstudeerder binnen organisatie	2
3 Situatie begin afstudeerperiode	4
3.1 Aanleiding	4
3.2 Probleemstelling	4
3.3 Doelstelling	4
3.4 Resultaat	4
3.5 Afbakening	4
3.6 Analyse Pharmafilter	6
4 Methodiek	9
4.1 RUP	9
4.2 Inception fase	9
4.3 Elaboration fase	10
4.4 Construction fase	11
5 Inception fase	12
5.1 Vision Document	12
5.2 Analyse TeleControl	13
5.3 Ontwerp TeleControl	16
6 Elaboration fase	21
6.1 Aanpassingen Vision Document	21
6.2 Analyse Pharmafilter	21
6.3 Onjuist gebruik van fases RUP	21
7 Construction fase	22
7.1 Aanmaken datatypes TeleControl	22
7.2 Timestamp	22
7.3 WinCC verbinding	23
7.4 STL of SCL	23
7.5 WinCC tags	24
7.6 Aanmaken DB's	24
7.7 ShiftValue	25
7.8 Verbetering	25
8 Transition fase	26
8.1 Testplan/rapport	26
8.2 In bedrijf stellen	26
9 Extra werkzaamheden	27
9.1 WinCC	27
9.2 PLC code	29

10 Evaluatie	30
10.1 Vision Document	30
10.2 Analyse document	31
10.3 Ontwerp	31
10.4 PLC code	31
10.5 Testplan	32
10.6 Extra werkzaamheden	32
Referenties	33
Begrippenlijst	34

1 Inleiding

Op 8 februari 2016 begon de afstudeerperiode bij Siemens Nederland. Dit bedrijf staat bekend om haar producten die overal ter wereld worden gebruikt. Het bedrijf staat echter minder bekend om de service die het daarnaast verleent. Toch is dit een belangrijk onderdeel van Siemens. De afdeling Engineering & Commissioning behoort bij deze service. Zij helpen klanten met het opzetten van onder andere SCADA systemen. De engineers worden ingehuurd om projecten voor klanten uit te voeren. Zo ook het project voor de klant Pharmafilter.

De volledige afstudeerperiode is er met de engineers samengewerkt aan het project voor Pharmafilter. Een deel van dit project is volledig aan mij toebesteed. Wat er allemaal is gedaan, hoe dit is uitgevoerd en waarom, staat beschreven in dit verslag.

Om te beginnen is er een korte beschrijving gegeven over de bedrijven die betrokken waren bij het project, Siemens en Pharmafilter. Tussendoor is er uitgebreid te lezen wat er van mij werd verwacht. Vervolgens zal er te vinden zijn welke methode er gebruikt is en hoe deze is toegepast. Vervolgens zijn alle werkzaamheden beschreven. Ten slotte is er een evaluatie gemaakt om de afstudeerperiode nog eens zelf te beoordelen.

2 Organisatie

De organisatie waar ik heb afgestudeerd is Siemens Nederland NV. Prinses Beatrixlaan 800, 2595 BN Den Haag.

2.1 Organisatie algemeen

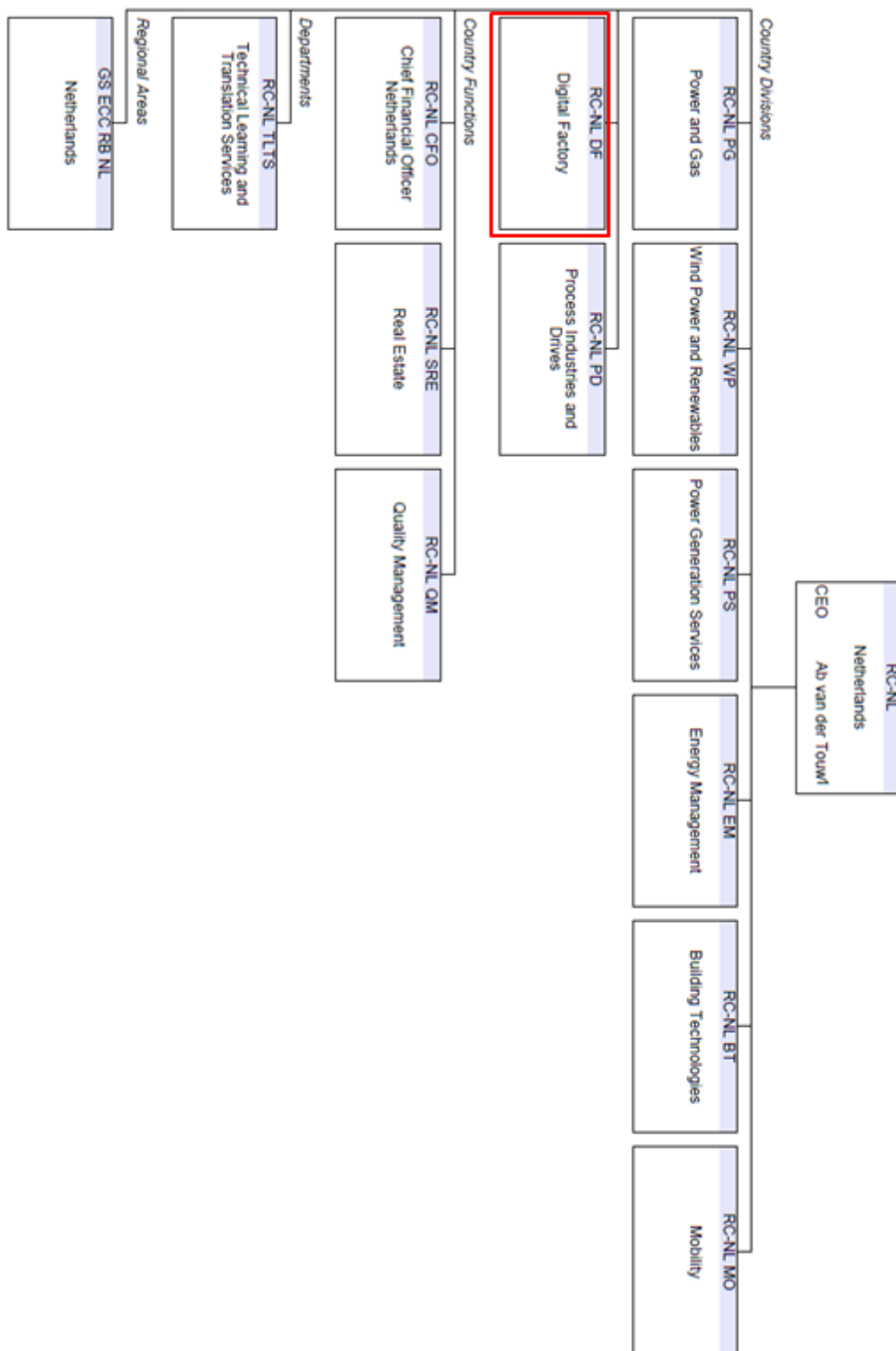
De missie van Siemens is om als pionier oplossingen te bieden voor verstedelijking, klimaatverandering, mondialisering en demografische verandering. Siemens is een bedrijf met veel verschillende soorten klanten. Dit komt door de grote variatie aan producten en diensten die zij levert. Hierbij kan gedacht worden aan PLC's en sensoren, maar ook aan grotere producten zoals treinen en gasturbines. Om dit allemaal te kunnen realiseren bestaat Siemens uit negen divisies: Power and Gas, Wind Power and Renewables, Power Generation Services, Energy Management, Building Technology, Mobility, Digital Factory, Process Industries and Drives en Healthcare. De afdeling waar ik heb afgestudeerd heet Engineering & Commissioning. Dit valt onder de divisie Digital Factory. Deze afdeling bestaat uit meerdere werknemers die elke werkdag aan projecten werken voor klanten van Siemens. Een organigram van de divisies van Siemens is te vinden in Figuur 2.1.

2.2 Project

Siemens heeft sinds januari 2016 een project lopen voor de klant Pharmafilter. Over Pharmafilter is in paragraaf 3.6 meer te lezen. Het project is bedoeld om door middel van een samenwerking van Siemens en Pharmafilter een systeem te maken om in ziekenhuizen te kunnen implementeren. Dit systeem bestaat uit een netwerk van shredders, waar S7-1200 PLC's inzitten. Meer informatie over de shredder, verder genoemd; Tonto, is te vinden in paragraaf 3.6. Verder bestaat het systeem uit een WinCC server die op afstand met het netwerk van Tonto's dient te communiceren. Hiervoor is er een S7-1500 PLC die fungeert als dataverzamelaar. Deze dataverzamelaar staat tussen de WinCC omgeving en het netwerk van Tonto's in en zorgt voor de communicatie die hiertussen moet gebeuren. Dit is ook te zien in Figuur 3.1. De samenwerking die tot stand is gekomen zorgde ervoor dat een deel van het werk voor Pharmafilter uit handen is genomen door Siemens. Mijn taak was om weer werk uit handen van de Siemens engineers te nemen.

2.3 Plaats van afstudeerder binnen organisatie

Tijdens mijn afstudeerperiode ben ik werkzaam geweest binnen de afdeling Engineering & Commissioning. Hier heb ik meegewerkt aan het hierboven beschreven project. Dit project is van start gegaan ongeveer een maand voor het begin van de afstudeerperiode. Toen de afstudeerperiode begon was hierdoor een gedeelte van het ontwerp en een deel van de code al gemaakt. Echter was er een bepaald onderdeel van het project, wat volledig aan mij is overgedragen. Dit onderdeel heeft te maken met de communicatie tussen een S7-1500 PLC en een WinCC omgeving door middel van TeleControl. Hier heb ik zelfstandig aan gewerkt, nadat een collega van de afdeling mij op de hoogte had gebracht van de stand van zaken binnen het project. Binnen de organisatie heb ik met een aantal mensen te maken gehad, waarvan de belangrijkste mijn bedrijfsbegeleider is. Met hem is er veel overleg geweest over de eisen van het project. Hij is degene die met Pharmafilter de eisen afstemde. Verder is er door mij contact geweest met een medewerker van Pharmafilter, om te overleggen en te testen of de software tot op die momenten naar wens was. Dit gebeurde tijdens contactmomenten bij Siemens, waar de medewerker van Pharmafilter af en toe op bezoek kwam.



Figuur 2.1: Organigram divisies Siemens

3 Situatie begin afstudeerperiode

Bij aanvang van de afstudeerperiode was het project binnen Siemens al een maand aan de gang. Hierdoor liep de planning zoals die door mij was bedacht niet synchroon met de planning van het project bij Siemens. De eerste week van de afstudeerperiode was vooral het regelen van spullen en overdragen van documenten en software. De week erna moest er dus direct gewerkt worden. Het project zat voor Siemens al in de ontwikkelfase. Het ontwerp was voor het grootste deel al klaar en er werd volop geprogrammeerd. Er moest snel een Vision Document gemaakt worden en wanneer deze was goedgekeurd is er gelijk begonnen met het meewerken aan het project. Hieronder zijn de onderdelen van de opdrachtschrijving van het Vision Document opgenomen.

Deze opdrachtschrijving is opgesteld aan het begin van de afstudeerperiode en zal ook op die manier geschreven zijn. Als deze opdrachtschrijving wordt vergeleken met de opdrachtschrijving zoals deze is opgesteld in het Afstudeerplan, is er een aantal verschillen te vinden. Dit wordt verder besproken in paragraaf 5.1.

3.1 Aanleiding

Via verschillende routes bereikt ziekenhuisafval het oppervlakte- en grondwater. Dit heeft bewezen nadelige effecten op waterorganismen [Van der Hoek et al., 2013]. Ook in het drinkwater wordt ziekenhuisafval aangetroffen. Bij ziekenhuisafval gaat het om onder andere: recyclebare bed-po's, plastic, papier, etensresten, injectienaalden, medicijnen enzovoorts. Om deze problemen te lijf te gaan, is het Pharmafiltersysteem bedacht [Batelaan et al., 2012]. Kern daarbij is de opvang en reiniging van afvalwater waaraan ook andere ziekenhuisafvalstromen worden toegevoegd en waarbij eenmalig te gebruiken biologisch afbreekbare producten worden toegepast.

3.2 Probleemstelling

Mijn opdracht is een onderdeel van een groter project waar Siemens mee bezig is voor de klant Pharmafilter. Pharmafilter is een groeiend bedrijf dat afvalverwerkingssystemen (TONTOS) maakt voor ziekenhuizen. Deze TONTO's zijn voorzien van een S7-1200 PLC. Deze PLC's besturen het vermaalproces van de TONTO's en houden bepaalde gegevens bij, waaronder het aantal vermalingen die in rekening zullen worden gebracht bij het ziekenhuis. Het ziekenhuis huurt de Tonto's van Pharmafilter, waar een bepaald bedrag voor wordt verrekend. Hier komen de kosten per vermaling nog bij. Verder zijn de PLC's die in de TONTO's zitten op dit moment allemaal losse units. Storingen en dergelijken worden niet centraal gemeld, maar komen alleen op de panels van de TONTO's zelf te staan. Ook wanneer er een instelling moet worden veranderd dan moet dit ter plaatse gebeuren, wat niet praktisch is voor Pharmafilter. In eerste instantie zal de Technische Dienst van het ziekenhuis proberen het probleem op te lossen. Wanneer dit niet lukt kan er een monteur van Pharmafilter worden ingeschakeld. Deze monteur moet dus naar het ziekenhuis toe om het probleem op te lossen.

3.3 Doelstelling

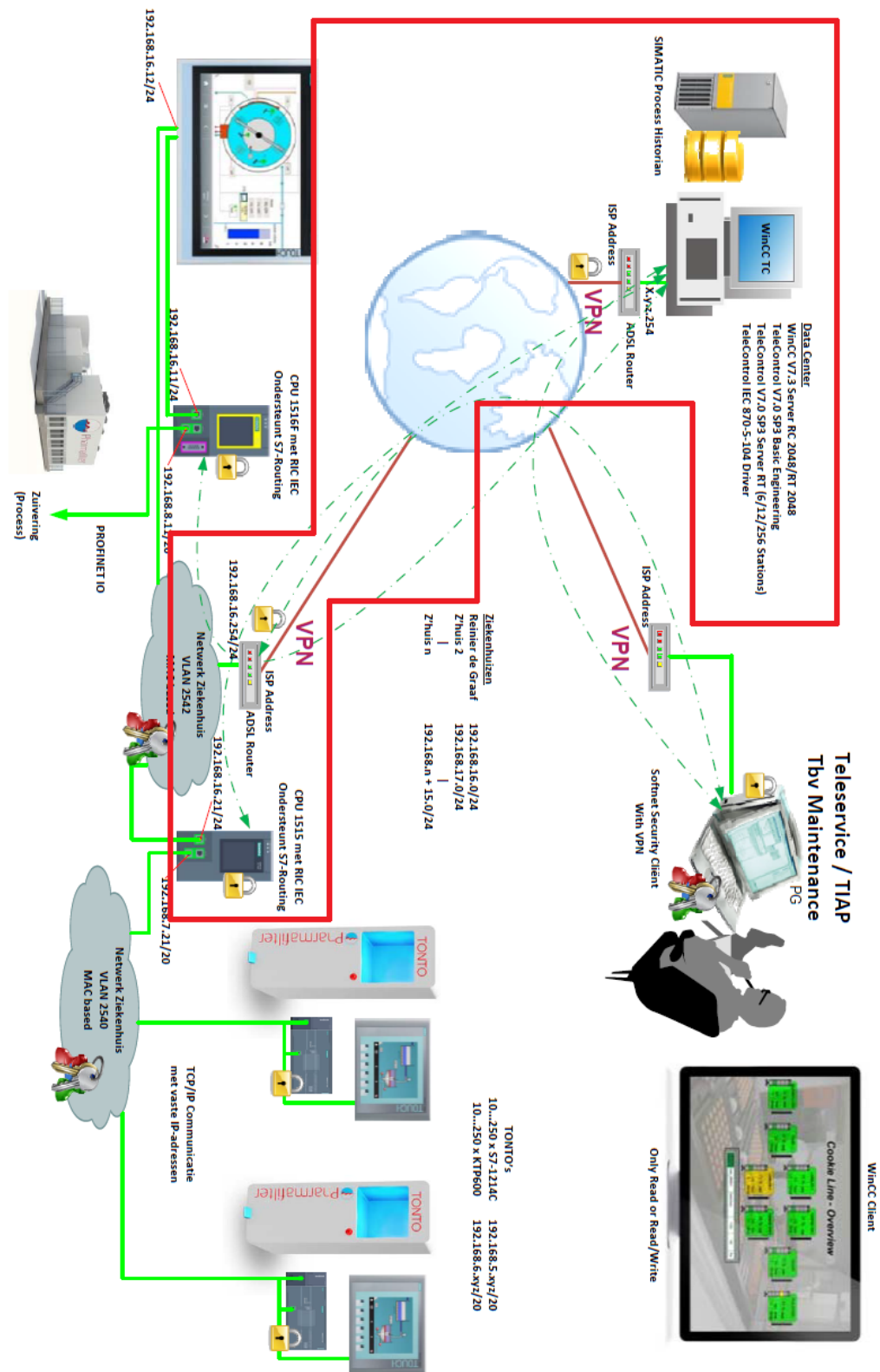
Het afstudeertraject is onderdeel van dit grotere project. Ik zal als engineer bij Siemens meewerken aan het project, waarbij een deel van het project zelfstandig uitgevoerd zal worden. Aan het einde van mijn afstudeerperiode moet er uitgezocht zijn hoe de communicatie tussen een S7-1500 PLC en een WinCC omgeving geprogrammeerd kan worden, met behulp van TeleControl. Dit is in een testopstelling geïmplementeerd.

3.4 Resultaat

Er zal een testopstelling zijn welke bestaat uit een aantal S7-1200 PLC's en een S7-1500 PLC die allemaal in een netwerk met elkaar zijn verbonden. Daarnaast communiceert de S7-1500 PLC via TeleControl met een WinCC omgeving. De software wordt ontwikkeld met behulp van TIA Portal en WinCC.

3.5 Afbakening

Het gehele communicatieplan zoals deze door de Engineers van Siemens is bedacht is te zien in Figuur 3.1. Het deel waar ik mij op ga focussen tijdens mijn afstudeerperiode is aangegeven met een rood kader. De ip-adressen en namen en dergelijken zijn in deze afbeelding niet hoe ze daadwerkelijk zullen worden geïmplementeerd.



Figuur 3.1: Communicatieplan

3.6 Analyse Pharmafilter

Aan het begin van de afstudeerperiode is er een overdracht geweest van kennis en documentatie. Hierbij hoorde het volgende document: [Batelaan et al., 2012]. In dit document staat een uitgebreide beschrijving over het Pharmafiltersysteem. Er staat in hoe de Tonto's werken en hoe de rioolwaterzuiveringsinstallatie (RWZI) werkt. Dit document is helemaal doorgelezen en geanalyseerd. Hiervoor is gescand op stukken in het document die gaan over Pharmafilter of over de Tonto's. Hieruit zijn stukken gehaald die naar mijn mening belangrijk zijn voor in dit Afstudeerverslag. Verder zijn deze stukken opgenomen in het Analyse Document. Om van waarde te zijn, moesten de teksten aan bepaalde eisen voldoen:

- Het moest gaan over het bedrijf Pharmafilter of over de Tonto's.
- Er mocht niet in worden gegaan op de details wat betreft de techniek achter het Pharmafiltersysteem.
- Er moest uiteindelijk duidelijkheid komen over wat Pharmafilter inhoudt en wat de Tonto's zijn.

De stukken tekst die aan deze voorwaarden voldeden zijn uit het document gekopieerd. Hieruit is onderstaande informatie gekomen.

Pharmafilter Pharmafilter houdt zich bezig met het zuiveren van rioolwater dat uit ziekenhuizen komt. Dit gebeurt door middel van zo genoemde Tonto's en een RWZI. Deze Tonto's staan verspreid binnen het ziekenhuis en worden gebruikt als afvalbakken. Meer informatie over de Tonto's is te vinden in hoofdstuk 3.6.

Het afval wordt vermalen en samen met het afvalwater van douches, wasbakken en toiletten via de bestaande interne riolering van het ziekenhuis afgevoerd naar de vergisting- en zuiveringsinstallatie op het terrein van het ziekenhuis [Batelaan et al., 2012]. Hier wordt het vaste en vloeibare afval gescheiden. Op deze wijze wordt in het ziekenhuis een hygiënisch, logistiek en financieel voordeel behaald.

Bacteriën zetten in een vergister het organisch afval om in biogas dat door middel van een biogasmotor de installatie van energie voorziet. Het met relatief veel medicijnen belaste afvalwater, wordt ontdaan van alle schadelijke stoffen voor mens, dier en milieu, en gezuiverd tot helder proceswater.

Het Pharmafiltersysteem is ontworpen om in de praktijk gemakkelijk geïmplementeerd te kunnen worden. De installatie is standaard uitgevoerd in modules op het formaat van een standaard zeecontainer. Deze worden inclusief leidingen in een werkplaats opgebouwd en zijn zo ingericht dat op het terrein van het ziekenhuis 'plug & play' kan worden toegepast. De tanks worden ter plaatse opgebouwd. De doorlooptijd van het totale bouwproces kan zo beperkt worden tot zes maanden vanaf het moment dat de grondplaat er ligt.

Op dit moment is het Pharmafiltersysteem in gebruik bij het Reinier De Graaf Gasthuis te Delft, zie Figuur 3.2.



Figuur 3.2: Pharmafilter bij het Reinier De Graaf Gasthuis, Delft

TONTO's De TONTO's van Pharmafilter zijn een soort vermalers. De apparaten “shredde” het afval wat erin gegooid wordt. Dit kan al het ziekenhuisafval zijn, van plastic tot injectienaalden. Dit afval wordt na het vermalen afgevoerd naar een RWZI welke naast het ziekenhuis gebouwd wordt. In elke TONTO zit een S7-1200 PLC die onder andere bijhoudt hoeveel van deze vermalingen er worden gedaan. Het ziekenhuis moet Pharmafilter namelijk per vermaling betalen. Een Tonto is een volledig nieuw elektromechanisch apparaat dat is ontwikkeld voor dit project [Batelaan et al., 2012]. In de eisen staan de volgende zaken centraal:

- Veiligheid
- Hygiëne
- Een zo breed mogelijke stroom afval verwerkbaar
- Zo min mogelijk gebruikershandelingen
- Vermaling en doorvoer naar de riolering gericht op vermijden verstoppingen
- Betrouwbaarheid ten minste vergelijkbaar met bedpanspoeler
- Geen overlast van geluid



Figuur 3.3: Eén van de tonto's zoals deze in het Reinier De Graaf Gasthuis staan

De Tonto past op de plek van de bedpanspoeler en gebruikt dezelfde aansluitingen (water, elektriciteit en riolering) en een extra UTP-aansluiting. Gebruikers dienen te worden getraind in het werken met de Tonto en de eenmalig te gebruiken bioplastics. Dit behoeft een korte instructie door leidinggevenden.

Een groot voordeel van de Tonto is dat er geen onderscheid wordt gemaakt in verschillende soorten afval en of het wel of niet besmet is. Hierdoor zullen er minder fouten worden gemaakt en wordt de hygiëne en veiligheid bevorderd. Vrijwel al het afval van een ziekenhuis kan in de Tonto worden verwerkt. Uitzonderingen zijn kantoorpapier, glas en klein chemisch afval, zoals batterijen. Dankzij de Tonto zijn er veel minder transporten en liftbewegingen voor het vervoer van afval.

De Tonto wordt met de voet bediend via een voetpedaal. Hierdoor zijn veel (ongewenste) handcontactmomenten verdwenen. De deksel van de Tonto staat in principe altijd open. Wanneer het apparaat met het pedaal in bedrijf wordt gesteld, gaat de deksel na 40 seconden weer open. Indien een Tonto in storing is en deze niet snel ter plaatse kan worden gerepareerd, vervangt de Technische Dienst van het ziekenhuis het apparaat door een reserve-Tonto. Dit gaat snel en eenvoudig, waardoor de verpleegafdelingen praktisch geen hinder ondervinden van een niet-werkende machine.

De Tonto heeft een onderdrukstelsel waardoor aerosolen niet in de lucht kunnen komen, maar via de riolering in de installatie terecht komen en daar worden verwerkt. De Tonto is aangesloten op een datanetwerk en wordt op afstand met een softwareprogramma gemonitord. Zo worden continu gegevens vanuit de Tonto verzameld. Deze gegevens zijn beschikbaar voor de Technische Dienst van het ziekenhuis en de ingenieurs van Pharmafilter: bijvoorbeeld storingsmeldingen en het aantal keren dat een Tonto heeft vermalen.

4 Methodiek

4.1 RUP

Tijdens de afstudeerperiode is er gewerkt volgens de ontwikkelmethode RUP. Door het gebruiken van RUP is het mogelijk om door middel van iteraties veranderingen in het ontwerp door te voeren (ook wanneer de requirements veranderen). Op deze manier kunnen er ook tijdens het project nieuwe technieken en tools geïntroduceerd worden in plaats van deze vooraf vast te leggen. Dit was erg handig voor deze periode, aangezien de eisen niet van tevoren zijn vastgelegd. De engineers van Siemens hebben aan het begin van het project de afspraak gemaakt met Pharmafilter, dat de engineers een ontwerp document bij zouden houden. In dit document is beschreven wat er geprogrammeerd moest worden. Elke keer als het document werd veranderd of er iets werd toegevoegd, moest Pharmafilter hiermee akkoord gaan. Hierdoor werden in dit document indirect de eisen vastgesteld. Door de controle van Pharmafilter op het document werd er verzekerd dat de klant akkoord ging met de eisen die werden opgesteld. Aangezien de engineers niet alle eisen zelf konden bedenken hebben er meerdere contactmomenten plaatsgevonden waarin Pharmafilter de eisen van hun kant door konden spelen. De engineers konden zo op hun beurt akkoord gaan met die eisen.

Door deze manier van werken konden de eisen af en toe veranderen. Door het gebruik van RUP kon hier eenvoudig mee worden gewerkt. Het gebruik van iteraties helpt hierbij. Verder liep de planning van het project voor Siemens niet gelijk met de planning van de afstudeerperiode. Hierdoor kon het voorkomen dat er taken door elkaar liepen. RUP helpt hierbij aangezien iteraties in elke fase opnieuw gepland kunnen worden.

4.2 Inception fase

In het Vision Document stond de eerste planning die gemaakt is. In Figuur 4.1 is zowel de fase- als de iteratieplanning te zien.

	08-02 / 12-02	15-02 / 19-02	22-02 / 26-02	29-02 / 04-03	07-03 / 11-03	14-03 / 18-03	21-03 / 25-03	28-03 / 01-04	04-04 / 08-04
Inception fase									
Elaboration fase									
Construction fase									
Transition fase									
	11-04 / 15-04	18-04 / 22-04	25-04 / 29-04	02-05 / 06-05	09-05 / 13-05	16-05 / 20-05	23-05 / 27-05	30-05 / 03-06	
Inception fase									
Elaboration fase									
Construction fase									
Transition fase									

Figuur 4.1: Faseplanning Vision Document

Aangezien er van tevoren aan mij is verteld dat er meegeholpen mocht worden met het in bedrijf stellen van het systeem wat gemaakt werd, is er een Transition fase ingepland. Deze zijn geschat op twee weken. De reden hiervoor is dat er zo genoeg tijd ingepland zou zijn hiervoor, maar niet teveel zodat er genoeg tijd over zou zijn voor de andere taken. Als het in bedrijf zetten minder lang zou duren, zou de tijd opgevuld kunnen worden met het werken aan het afstudeerverslag. De andere fases zijn ingepland op de hoeveelheid werk dat er ongeveer zou moeten gebeuren in die fases. Dit is gebaseerd op de opdrachtomschrijving zoals deze is opgesteld in het Vision Document.

4.3 Elaboration fase

	29-feb	1-mrt	2-mrt	3-mrt	4-mrt	7-mrt	8-mrt	9-mrt	10-mrt	11-mrt
Vision document aanpassen										
Eisen opdrachtgever opstellen										
Achtergrondinformatie project verzamelen										
Analyse document opstellen										
	14-mrt	15-mrt	16-mrt	17-mrt	18-mrt					
Vision document aanpassen										
Eisen opdrachtgever opstellen										
Achtergrondinformatie project verzamelen										
Analyse document opstellen										

Figuur 4.2: Iteratieplanning Elaboration fase

Voor de iteratieplanning voor de Elaboration fase is een aantal iteraties gekozen. Deze zijn te zien in Figuur 4.2. Aangezien het Vision Document gemaakt is in de Inception fase is gemaakt en aan het einde is gekeurd, moest er nog wat worden aangepast. Dit diende hierdoor te gebeuren in de opvolgende fase. Hierdoor is er een iteratie gepland voor het aanpassen van het Vision Document. Daarnaast was er gepland om de eisen van de opdrachtgever te achterhalen. Echter bleek hieruit dat de eisen opgesteld zouden worden via een document dat bijgehouden werd, zoals eerder beschreven is. De eisen voor mijn deel van het project zijn hier ook in beschreven. Verder is er een iteratie gepland voor het verzamelen van achtergrondinformatie over het project. Hiermee wordt bedoeld het verzamelen van informatie over Pharmafilter en de Tonto's. Hiervoor zijn drie dagen gepland. De reden hiervoor is dat er bekend was waar de informatie vandaan gehaald moest worden, door de overdracht van kennis aan het begin van de afstudeerperiode. Er was een document waar eigenlijk alle, voor mij nuttige informatie instond. Drie dagen werd zodoende ingeschat als ruim voldoende. Ten slotte moest er een Analyse Document gemaakt worden. Dit is dan ook de laatste iteratie waarvoor een week ingepland was om zo voldoende tijd hiervoor te hebben.

In dit Analyse Document is de faseplanning voor de tijd die erna volgde opnieuw opgesteld, zie Figuur 4.3. Tevens was de iteratieplanning voor de Construction fase hierin opgenomen, zie Figuur 4.4.

	04-04 / 08-04	11-04 / 15-04	18-04 / 22-04	25-04 / 29-04	02-05 / 06-05	09-05 / 13-05	16-05 / 20-05	23-05 / 27-05	30-05 / 03-06
Construction fase									
Transition fase									

Figuur 4.3: Faseplanning Analyse Document

Zoals te zien in Figuur 4.3 is de faseplanning iets aangepast ten opzichte van de faseplanning zoals deze in het Vision Document stond, zie Figuur 4.1. De oorzaak hiervan is het door elkaar lopen van de fases. Hierdoor begon de Construction fase later dan verwacht. Dit was echter niet erg, aangezien sommige taken van de Construction fase al deels in de Elaboration fase waren uitgevoerd. Daarnaast is er tijdens de Elaboration fase helaas een periode van ziekte geweest. Hierdoor hebben er ook wat verschuivingen in de planning plaatsgevonden.

4.4 Contruuction fase

	4-apr	5-apr	6-apr	7-apr	8-apr	11-apr	12-apr	13-apr	14-apr	15-apr
Ontwerp maken										
WinCC project maken										
PLC code maken										
Testplan maken										
Testen										
	18-apr	19-apr	20-apr	21-apr	22-apr	25-apr	26-apr	27-apr	28-apr	29-apr
Ontwerp maken										
WinCC project maken										
PLC code maken										
Testplan maken										
Testen										
	2-mei	3-mei	4-mei	5-mei	6-mei	9-mei	10-mei	11-mei	12-mei	13-mei
Ontwerp maken										
WinCC project maken										
PLC code maken										
Testplan maken										
Testen										
	16-mei	17-mei	18-mei	19-mei	20-mei					
Ontwerp maken										
WinCC project maken										
PLC code maken										
Testplan maken										
Testen										

Figuur 4.4: Iteratieplanning Contruuction fase

De Contruuction fase is gepland in vijf iteraties, zie Figuur 4.4. Er moest hoe dan ook een ontwerp gemaakt worden. Het document die de engineers van Siemens maakten waar de eisen ook vandaan zouden komen, was in principe een ontwerp document. Echter moest mijn deel van het project ook ontworpen worden. Dit is dan ook door mij gedaan. Aangezien niet precies duidelijk was hoelang hiervoor nodig zou zijn is hier een week voor gepland, maar doordat deze fase uit zeven weken bestaat was er nog ruimte voor verschuiving. Het maken van een WinCC project was in eerste instantie voor een deel mijn werk, waardoor er een week voor gepland was. Eigenlijk met dezelfde reden als voor het ontwerp maken. Voor het maken van de PLC code is een periode van drie weken en twee dagen gepland. Aangezien deze iteratie het meeste van de tijd nodig zou hebben is het op deze manier gepland. Voor het maken van het testplan en het testen is ongeveer anderhalve week gepland. De reden hiervoor is dat mijn eigen deel van het project door mij getest diende te worden. Door hier anderhalve week voor te plannen werd er verwacht dat hier genoeg tijd voor zou zijn. Uiteindelijk was het de bedoeling geweest om het testplan/rapport te maken en het systeem te testen in de Transition fase.

Na het Analyse Document zijn er geen andere documenten meer gemaakt. Het ontwerp wat door mij is gemaakt is opgenomen in het document van de engineers. Er is dan ook geen planning gemaakt voor de Transition fase. Dit leek mij ook niet nodig aangezien die fase inhield dat er meegewerkt zou worden aan het in bedrijf stellen van het systeem en het maken van het testplan/rapport. Daarnaast zou het testplan worden uitgevoerd. Dit zouden de enige twee iteraties zijn in die fase.

5 Inception fase

Aan het begin van de afstudeerperiode moest er direct meegewerkt worden aan het project waar Siemens mee bezig was voor de klant Pharmafilter. Er werd al sinds januari aan het project gewerkt, waardoor er al een maand werk klaar was. Het eerste wat er moest gebeuren was de overdracht van alle software die al was gemaakt. Hierbij is uitgelegd wat er nog van mij verwacht werd. Dit staat beschreven in de opdrachtschrijving in hoofdstuk 3.

5.1 Vision Document

Ondanks dat er werd verwacht dat er meteen mee werd gewerkt aan het project, werd er eerst een Vision Document gemaakt. Dit is bereikt door vragen aan de opdrachtgever te stellen. Deze vragen zijn niet formeel vastgelegd, aangezien dit niet nodig gevonden werd. Er kon zo in het Vision Document worden opgeschreven wat er werd verwacht en dit kon weer worden gekeurd door de opdrachtgever. Na de eerste versie van het Vision Document is er dus een feedback moment geweest met de opdrachtgever om het document te laten keuren. Hierna zijn er wat aanpassingen gedaan aan de opdrachtschrijving. Bij de versie die erna kwam is dit uiteindelijk goedgekeurd. Hierdoor is er overeenstemming gekomen met de opdrachtgever wat betreft de werkzaamheden van de afstudeerder. Als de opdrachtschrijving uit hoofdstuk 3 vergeleken wordt met het Afstudeerplan dat voor de afstudeerperiode is gemaakt, is er een aantal verschillen op te merken, zie Figuur 5.1. Er is namelijk in de periode tussen het opstellen van het Afstudeerplan en de aanvang van de afstudeerperiode al een deel van het project gemaakt. Hierbij is er onderling overleg geweest tussen de engineers van Siemens welk deel van het project zij niet zouden maken. Dit is uiteindelijk het deel geworden wat staat beschreven in de opdrachtschrijving.

Aan het einde van mijn afstudeerperiode moet er onderzocht zijn hoe de communicatie tussen de S7-1200 en de S7-1500 PLC's geprogrammeerd kan worden. Hieruit is een keuze gemaakt die verder is ontworpen en geprogrammeerd met behulp van TIA Portal. Dit is in de TONTO's van PharmaFilter geïmplementeerd.

(a) Afstudeerplan

Het afstudeertraject is onderdeel van dit grotere project. Ik zal als engineer bij Siemens meewerken aan het project, waarbij een deel van het project zelfstandig uitgevoerd zal worden. Aan het einde van mijn afstudeerperiode moet er uitgezocht zijn hoe de communicatie tussen een S7-1500 PLC en een WinCC omgeving geprogrammeerd kan worden, met behulp van TeleControl. Dit is in een testopstelling geïmplementeerd. Daarnaast moet er een SIMATIC Process Historian geconfigureerd zijn waar de WinCC omgeving data op kan archiveren.

(b) Vision Document

Figuur 5.1: Vergelijking doelstellingen

Er is in Figuur 5.1 ook te zien dat de doelstelling zoals beschreven in het Vision Document anders is als in hoofdstuk 3 van dit document. Het stuk over de SIMATIC Process Historian is komen te vervallen. Siemens en Pharmafilter hebben namelijk samen besloten dat dit onderdeel door Pharmafilter zelf afgehandeld zou worden.

5.2 Analyse TeleControl

In deze fase is er verder nog het een en ander uitgezocht over TeleControl. Hiervoor zijn er verschillende documenten over Telecontrol gelezen. Deze documenten zijn met de kennisoverdracht aan mij doorgespeeld. Alle documenten die over TeleControl gingen zijn er tussenuit gehaald en doorgelezen. Hieruit is informatie naar boven gekomen over de werking van TeleControl. TeleControl is namelijk de techniek die Siemens heeft geadviseerd aan Pharmafilter om te gaan gebruiken voor de communicatie tussen de WinCC omgeving en de S7-1500 PLC, zie Figuur 3.1. Deze techniek zorgt ervoor dat er gegevens zoals meetwaarden en tellers van de PLC naar de WinCC omgeving gestuurd kunnen worden. Het speciale van TeleControl is de mogelijkheid om deze gegevens te bufferen zodra de verbinding verbroken wordt. Wanneer de verbinding weer tot stand wordt gebracht zullen alle gebufferde gegevens via TeleControl alsnog verstuurd worden met de timestamp zoals deze staat aangegeven in de PLC.

TeleControl kan gebruik maken van meerdere protocollen, namelijk:

- SINAUT ST7
- IEC 60870-5
- DNP 3

Er is uiteindelijk gekozen om gebruik te maken van IEC 60870-5. De reden hiervoor is omdat er gebruik gemaakt zou worden van een S7-1500 PLC. Het DNP 3 protocol kan echter alleen gebruikt worden op de kleinere modellen zoals de S7-1200 PLC. SINAUT ST7 is een relatief oud protocol. Pharmafilter heeft ervoor gekozen om te gaan voor het “nieuwe” IEC protocol. Onder IEC vallen weer twee verschillende vormen; IEC en IEC TCP. Aangezien er gebruik gemaakt wordt van Ethernet TCP/IP en niet van serial verbindingen, wordt de IEC TCP versie gebruikt van IEC.

TeleControl heeft meerdere manieren voor het versturen van data. Hiervoor bestaan er standaard blokken die gebruikt worden voor TeleControl. In Tabel 1 hieronder is te zien uit welke blokken er een keuze gemaakt kon worden voor het verzenden van de data in dit project.

Tabel 1: TeleControl blokken

Blok	FB-nummer	Functie	Information Object Address (IOA) verdeling
SLI_SP_DP_s128	FB130	Versturen van max. 128 single (SP) of double points (DP)	1 bit per IOA, max. 128 IOAs per aanroep
SLI_ST_s8	FB131	Versturen van max. 8 step positions (ST)	1 byte per IOA, max. 8 IOAs per aanroep
SLI_BO_s8	FB132	Versturen van max. 8 bit patterns (BO)	32 bits per IOA, max. 8 IOAs per aanroep
SLI_ME_ABC_s32	FB133	Versturen van max. 32 measured values (ME)	32 bits per IOA, max. 32 IOAs per aanroep
SLI_IT_s8	FB134	Versturen van max. 8 integrated totals (IT)	32 bits per IOA, max. 8 IOAs per aanroep

In de tabel is de term IOA gebruikt. Een IOA is eigenlijk één telegram met data. Deze telegrammen worden in de PLC code aangeroepen per groep. Er kunnen bijvoorbeeld per aanroep 128 telegrammen verstuurd worden van 1 bit (SLI_SP_DP_s128), maar er kunnen per aanroep maar 8 telegrammen verstuurd worden van 32 bits (SLI_BO_s8 en SLI_ST_s8 bijvoorbeeld). Deze telegrammen worden wel per groep verstuurd, maar er wordt per telegram bepaald of de waarde van de data die erin staat daadwerkelijk veranderd is. Zodra dit niet het geval is wordt het telegram niet meegestuurd met de rest van de groep. Het kan dus ook zijn dat er een aanroep wordt gedaan van een bepaald TeleControl blok waarbij er geen enkel telegram wordt verstuurd.

Uit al deze mogelijkheden zoals in Tabel 1 is weergegeven, moest een keuze gemaakt worden. Hiervoor is er eerst bepaald wat de data was die er verstuurd moest worden. Het ging bij de data om het volgende:

- Counters vanuit de Tonto's en de RWZI;
- Alarmen vanuit de Tonto's;
- Events vanuit de Tonto's;
- Meetwaarden vanuit de RWZI;
- Processtatus vanuit de RWZI.

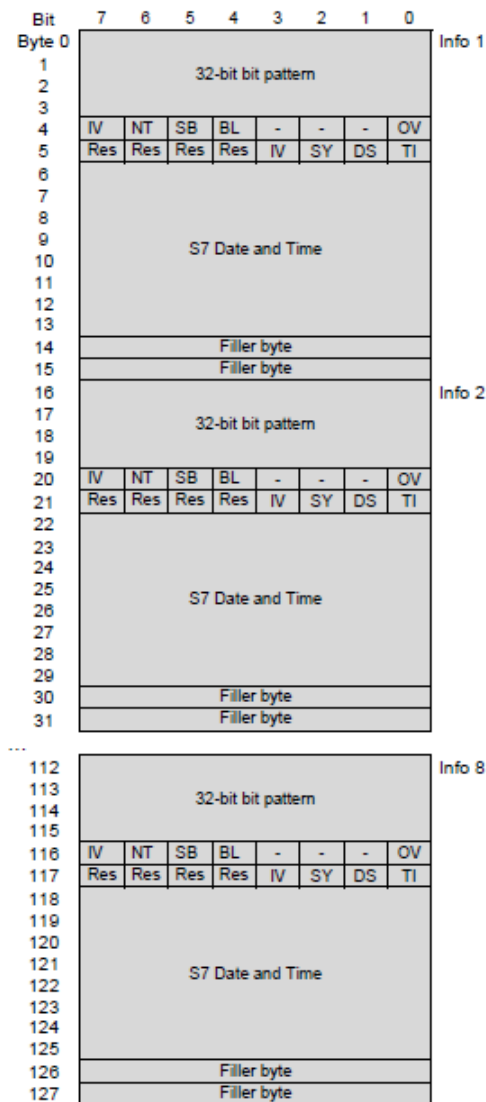
Door de engineers van Siemens is bepaald hoeveel bits voor elke soort data er nodig is. In Figuur 5.2 is een voorbeeld te zien van deze bepaling. Hierin is te zien dat de alarmen, events en counters uit de Tonto's bestaan uit respectievelijk: 32 losse bits, 32 losse bits en 1 real per counter.

Datacollector info	Byte 0...3	Datacollector info Via S7 protocol door WinCC gelezen
Locatie-ID	DBW4	
LivebestCounter	DBW6	
Profinet device name (NOS)	Byte 8...13	
MAC address	Byte 14...19	Locatie gegevens Tonto Via S7 protocol door WinCC gelezen
IP address	Byte 20...23	
subnet	Byte 24...27	
Default gateway IP	Byte 28...31	
Alarm01...Alarm32	DBD32 - 32 alarmbits	Procesgegevens Tonto Via Telecontrol door DataCollector verstuurd naar WinCC
Event01...Event32	DBD36 - 32 statusbits	
Counter01 (real)	DBD40	
Counter02 (real)	DBD44	
Counter03 (real)	DBD48	
Counter04 (real)	DBD52	
Counter05 (real)	DBD56	
Counter06 (real)	DBD60	
Counter07 (real)	DBD64	
Counter08 (real)	DBD68	
Counter09 (real)	DBD72	
Counter10 (real)	DBD76	
Counter11 (real)	DBD80	
Counter12 (real)	DBD84	
Counter13 (real)	DBD88	
Counter14 (real)	DBD92	
Counter15 (real)	DBD96	
Counter16 (real)	DBD100	
Counter17 (real)	DBD104	
Counter18 (real)	DBD108	
Counter19 (real)	DBD112	
Counter20 (real)	DBD116	
Spare Counter01 (real)	DBD120	
Spare Counter02 (real)	DBD124	
Spare Counter03 (real)	DBD128	
Spare Counter04 (real)	DBD132	
Spare Counter05 (real)	DBD136	

Figuur 5.2: Data bepaling

Door deze bepalingen kon er worden beslist op welke manier er met zo min mogelijk aanroepen, zoveel mogelijk data verstuurd kon worden. Zoals eerder genoemd kan er door middel van TeleControl met één aanroep een hele groep aan data worden verstuurd. Door zoveel mogelijk data te versturen in één aanroep, kost dit minder gereserveerde ruimte dat niet wordt gebruikt. Hierbij moest er gekeken worden naar welk TeleControl blok er gebruikt zou worden voor elk type data dat verstuurd moest worden. Door de databepaling van de engineers van Siemens kon er besloten worden hoe groot elk type data is. Stel dat, zoals in Figuur 5.2 te zien is, de alarmen uit een Tonto in totaal 32 bits nodig heeft, kunnen hiervoor de volgende blokken worden gebruikt: SLi_BO_s8, SLi_ME_ABC_s32 en SLi_IT_s8. Zoals te zien is in Tabel 1 versturen deze blokken allemaal iets anders. De alarmen van de Tonto's bestaat uit een reeks van 32 bits, waarbij door WinCC elk bit apart van elkaar wordt gebruikt. Bit 0 van de waarde geeft iets anders aan dan bit 1. Bij een bit pattern, de naam zegt het al, is het mogelijk om dit te doen. Hierdoor wordt voor de alarmen het blok SLi_BO_s8 gebruikt. Dit is zo bepaald voor elk type data dat verstuurd moest worden. De resultaten hiervan zijn te vinden in Tabel 2.

Van elk TeleControl blok zoals te zien is in Tabel 1 kunnen er meerdere types gemaakt worden. Welk type er gebruikt dient te worden hangt af van wat er gewenst is om mee te sturen met elk telegram. Een telegram kan namelijk uit meerdere gegevens bestaan. In Figuur 5.3 is te zien hoe een telegram van het SLi_BO.s8 blok eruit kan zien.



Figuur 5.3: SLi_BO.s8 blok type B#16#2D

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Res	Add_Dist			DT	Ext_State	Data_Type	

Data_Type	Data_Type is used depending on the information type – in this case, as follows: 00 (0): 4 byte bit pattern 01 (1): 4 byte bit pattern + 1 byte status, IEC assignment 10 (2): reserved 11 (3): reserved	Image spacing 4 bytes 5 bytes
------------------	---	--

Figuur 5.4: Uitleg blok type

Het type van blok SLi.BO.s8 dat te zien is in Figuur 5.3 is B#16#2D. De reden dat het hier gaat om B#16#2D is als volgt.

Het type komt voort uit hetgene dat wordt meegestuurd met een telegram. In dit geval wordt de S7 Date and Time meegestuurd. Daarnaast is bij de bovenste telegram (Info 1) bij byte 4 en 5 te zien hoe er nog twee bytes met gegevens mee worden gestuurd met dit telegram. Deze gegevens komen voort uit de IEC assignment, zie Figuur 5.4. In dit Figuur is te zien hoe het type is opgebouwd. B#16#2D is hexadecimaal voor het decimale getal 45, wat in binair weer 00101101 is. Deze bits kunnen nu ingevuld worden in de tabel van Figuur 5.4. Dit betekent dat het type B#16#2D bestaat uit bits 5, 3, 2 en 0. Hieruit kan afgeleid worden dat het Data_Type 01 is. Dus een 4 byte bit pattern + 1 byte status, IEC assignment. Verder wordt er gebruik gemaakt van Ext_State, DT en een Add_Dist bit. Ext_State kan worden gebruikt om datum en tijd eigenschappen mee te sturen die niet worden ondersteund door S7 Date and Time. DT betekent dat S7 Date and Time wordt meegestuurd. Add_Dist zorgt ervoor dat er extra filler bytes aan elk telegram wordt toegevoegd.

Er is getest hoe er een type gemaakt kon worden waarbij er zo min mogelijk wordt meegestuurd. Er was bekend dat de S7 Date and Time noodzakelijk was, voor het meesturen van de timestamp bij een telegram. Daarnaast was, in dit voorbeeld, een Data_Type nodig van alleen 4 byte bit pattern. Dit betekent dat het type uitkomt op 00001000, oftewel type B#16#08.

Hetzelfde is gedaan voor alle andere data die verstuurd moest worden. Er is van elk type data een zo minimalistisch mogelijk type gemaakt. Daarna is er getest of met dat type gewerkt kon worden, door het type te programmeren in de code en data te versturen. De resultaten zijn te vinden in Tabel 2.

Tabel 2: Bepalingen TeleControl blokken met types

Data	TeleControl blok	Blok type
Alarmen vanuit Tonto	SLi.BO.s8	B#16#08
Events vanuit Tonto	SLi.BO.s8	B#16#08
Counters vanuit Tonto	SLi.ME.ABC.s32	B#16#0A
Counters vanuit RWZI	SLi.ME.ABC.s32	B#16#0A
Processatus vanuit RWZI	SLi.ST.s8	B#16#2C
Meetwaarden vanuit RWZI	SLi.ME.ABC.s32	B#16#0A

5.3 Ontwerp TeleControl

Er is een ontwerp gemaakt een de hand van deze gegevens. In Tabel 3 is een stuk van dit ontwerp te vinden.

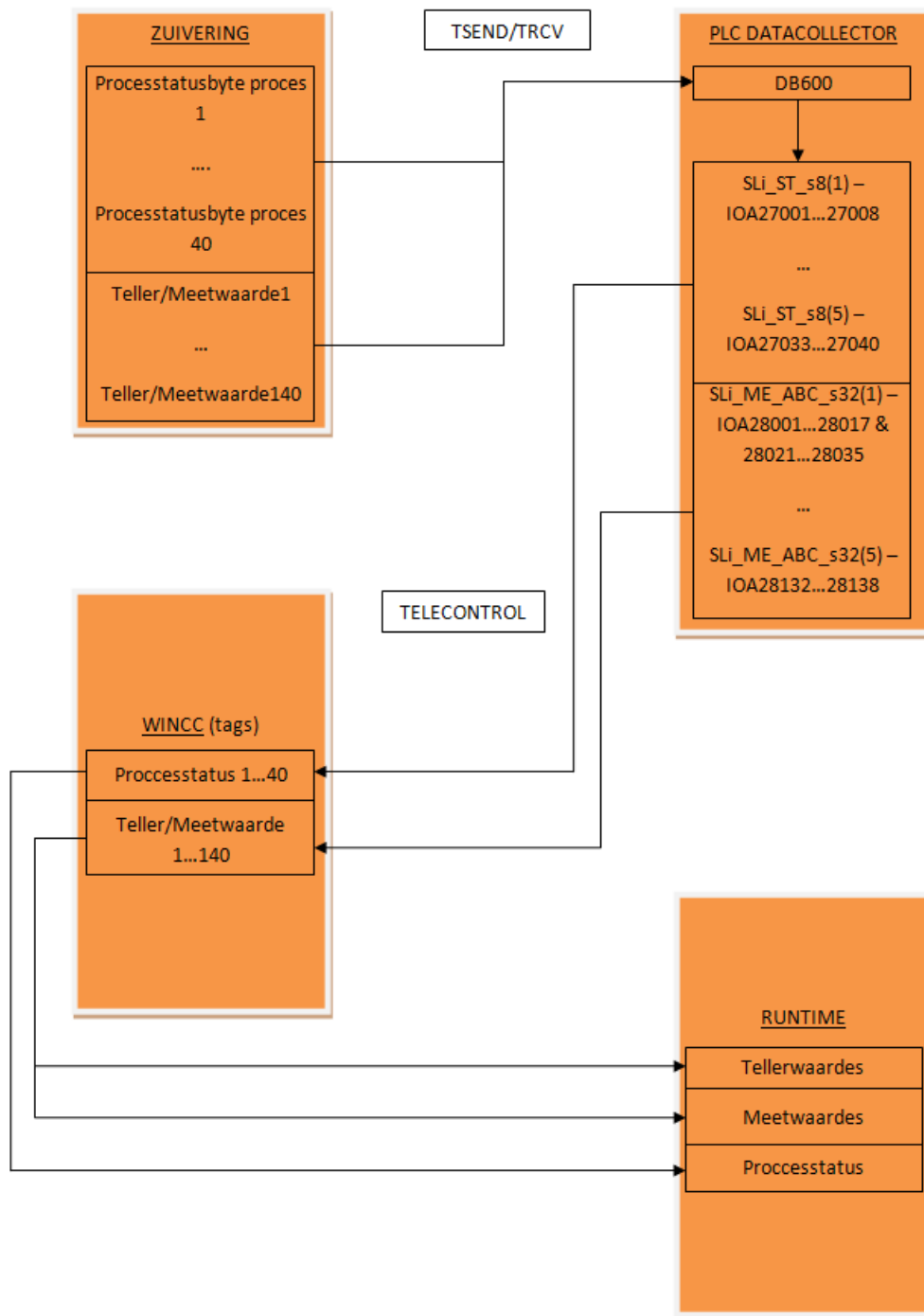
Tabel 3: Tabel bepalingen data

Naam	Echte namen	Grootte	Block type	IOA first Tonto	IOA last Tonto	Aantal blokken nodig	DB-Nummers
Alarm01...Alarm32	Alarm_31_00	32 bits	SLi_BO.s8 (type 08)	25001	25240	30	25001 - 25030
Event01...Event32	Event_31_00	32 bits	SLi_BO.s8 (type 08)	26001	26240	30	26001 - 26030
Counter1	TellerAantalVermalingen	1 real	SLi_ME.ABC.s32 (type 0A)	101	24001	240	30001 - 30240
Counter2	BedrijfsurenRotor	1 real	SLi_ME.ABC.s32 (type 0A)	102	24002		
Counter3	BedrijfsurenTafel	1 real	SLi_ME.ABC.s32 (type 0A)	103	24003		
Counter4	BedrijfsurenFlipper	1 real	SLi_ME.ABC.s32 (type 0A)	104	24004		
Counter5	BedrijfsurenRaam	1 real	SLi_ME.ABC.s32 (type 0A)	105	24005		
Counter6	BedrijfsurenWaterpomp	1 real	SLi_ME.ABC.s32 (type 0A)	106	24006		
Counter7	BedrijfsurenAfzuiging	1 real	SLi_ME.ABC.s32 (type 0A)	107	24007		
Counter8	BedrijfsurenDesinfectiepomp	1 real	SLi_ME.ABC.s32 (type 0A)	108	24008		
Counter9	TellerDesinfectie	1 real	SLi_ME.ABC.s32 (type 0A)	109	24009		
Counter10	TellerMRSA	1 real	SLi_ME.ABC.s32 (type 0A)	110	24010		
Counter11	TellerKlepA	1 real	SLi_ME.ABC.s32 (type 0A)	111	24011		
Counter12	TellerKlepB	1 real	SLi_ME.ABC.s32 (type 0A)	112	24012		
Counter13	TellerKlepC	1 real	SLi_ME.ABC.s32 (type 0A)	113	24013		
Counter14	TellerKlepD	1 real	SLi_ME.ABC.s32 (type 0A)	114	24014		
Counter15	TellerKlepE	1 real	SLi_ME.ABC.s32 (type 0A)	115	24015		
Counter16	TellerKlepF	1 real	SLi_ME.ABC.s32 (type 0A)	116	24016		
Counter17	TellerKlepG	1 real	SLi_ME.ABC.s32 (type 0A)	117	24017		
Counter18	TellerKlepH	1 real	SLi_ME.ABC.s32 (type 0A)	118	24018		
Counter19	TellerKlepJ	1 real	SLi_ME.ABC.s32 (type 0A)	119	24019		
Counter20	BedrijfsurenPompBalg	1 real	SLi_ME.ABC.s32 (type 0A)	120	24020		
Spare Counter1	Counter_21	1 real	SLi_ME.ABC.s32 (type 0A)	121	24021		
Spare Counter2	Counter_22	1 real	SLi_ME.ABC.s32 (type 0A)	122	24022		
Spare Counter3	Counter_23	1 real	SLi_ME.ABC.s32 (type 0A)	123	24023		
Spare Counter4	Counter_24	1 real	SLi_ME.ABC.s32 (type 0A)	124	24024		
Spare Counter5	Counter_25	1 real	SLi_ME.ABC.s32 (type 0A)	125	24025		

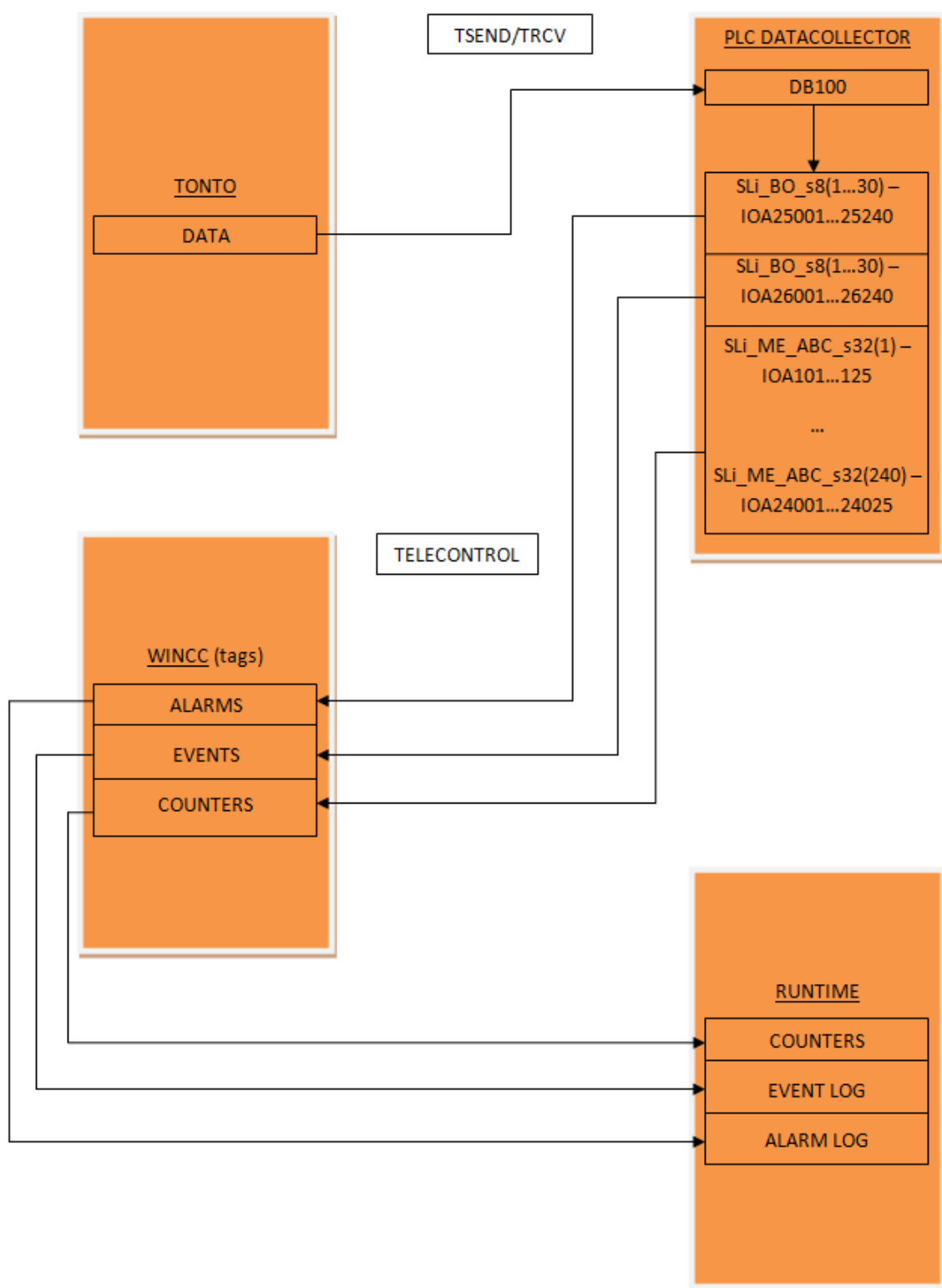
In de tabel is in de eerste kolom de naam te zien van de data. De tweede kolom geeft de namen weer zoals deze in de Tonto zijn aangegeven. Hierdoor is duidelijk waar alle counters bijhoren. De derde kolom geeft het datatype aan zoals deze is bepaald door de engineers van Siemens. De kolom met Block type geeft aan welk type van TeleControl gebruikt wordt. In dit voorbeeld is SLi_BO.s8 (type 08) een aanroep van maximaal 8 telegrammen waarin per telegram 32 losse bits worden verstuurd. Er zijn 32 alarmen per Tonto, die worden verstuurd met een variabele van 32 losse bits. Door dit TeleControl type worden er met één aanroep van zo'n TeleControl SLi_BO.s8 blok voor acht Tonto's tegelijk alle alarmen verstuurd naar WinCC. Dit betekent dat er in totaal, zoals aangegeven in de kolom "Aantal blokken nodig", 30 van deze aanroepen nodig zijn. Dit is de uitkomst van een eenvoudige rekensom; $\frac{240 \text{ Tonto's}}{8 \text{ Tonto's per aanroep}}$.

Dit ontwerp, zoals te zien in Tabel 3, is voor een klein deel gemaakt in de Inception fase, aangezien een andere engineer moest weten welke Block type er gebruikt moest worden met programmeren. Uiteindelijk heeft hij hier niks mee gedaan, aangezien alles van TeleControl verder door mij is geprogrammeerd in de Construction fase.

Naast de tabellen die gemaakt moesten worden, is er nog een schema gemaakt. Hiervoor geldt dezelfde reden als het ontwerp waarom de schema's in de Inception fase zijn gemaakt. De engineers van Siemens hadden deze schema's nodig om in het ontwerp document te voegen. Het schema gaf aan hoe de data van apparaat naar apparaat werd gestuurd en via welke techniek dit gebeurde. Er komt data vanaf de RWZI naar WinCC en er komt data vanaf de Tonto naar WinCC. In Figuur 5.2 is bijvoorbeeld te zien welke data er door een Tonto naar WinCC gestuurd wordt. In de schema's in Figuur 5.5 en Figuur 5.6 is te zien hoe deze data wordt verstuurd.



Figuur 5.5: Schema communicatie data vanaf RWZI naar WinCC



Figuur 5.6: Schema communicatie data vanaf Tonto naar WinCC

Er is te zien in beide schema's dat er in eerste instantie gebruik wordt gemaakt van TSEND/TRCV tussen de RWZI/Tonto PLC en de Datacollector. Dit is bepaald door de engineers van Siemens, evenals de DB-nummers waar de data in eerste instantie in terecht komt. Daarna komt de software van TeleControl aan het werk. De data wordt onderverdeeld over de aangemaakte TeleControl aanroepen van het bepaalde type.

Voorbeeld: In Figuur 5.5 is te zien dat de Processtatusbytes van de RWZI bytes zijn. Deze worden zoals eerder uitgelegd dus verstuurd via het standaard TeleControl SLi_ST_s8 blok. Met dit blok kunnen 8 van deze bytes worden verstuurd. Er zijn dus 5 van deze blokaanroepen nodig om alle 40 bytes te kunnen verzenden.

De IOA-nummers die zijn aangegeven zijn reeds bepaald in de tabellen van het ontwerp. Door middel van TeleControl zal de data via de IOAs worden verstuurd naar de WinCC Server. Hier worden de waarden in de desbetreffende tags gezet, waarna deze te zien zullen zijn in de Runtime omgeving.

Bij het schema in Figuur 5.6 gebeurt er ongeveer hetzelfde. Echter wordt er hier andere data verstuurd aangezien dit de data is dat uit de Tonto PLC komt.

6 Elaboration fase

6.1 Aanpassingen Vision Document

Aan het einde van de Inception fase is het Vision Document gemaakt. Deze moest daarna nog goedgekeurd worden door de opdrachtgever. Versie 1.0 van het Vision Document was nog niet helemaal naar wens van de opdrachtgever. Hij vond dat de opdrachtschrijving duidelijker moest maken wat mijn deel van het project zou worden en dat er een klein stukje geschreven moest worden over de Tonto's. Door deze wijzigingen in de opdrachtschrijving is het document verder aangepast wat betreft de planning. De iteratieplanning heeft er een iteratie bij gekregen voor het aanpassen van het Vision Document. Verder is er in de iteratieplanning het een en ander veranderd, doordat er al iteraties in de Inception fase waren gedaan. Zoals beschreven in hoofdstuk 5 is namelijk al een deel van het ontwerp in die fase gemaakt. Dit behoorde in mijn opzicht in eerste instantie tot de analyse. Dat is de reden dat de iteratie was ingepand voor de Elaboration fase. Deze iteratie is vervangen door de iteratie voor het opstellen van het Analyse Document. Deze iteratie was vergeten in de eerste versie van het Vision Document.

6.2 Analyse Pharmafilter

Verder is er in deze fase een analyse uitgevoerd over Pharmafilter en de Tonto's. Dit proces en het resultaat hiervan staat beschreven in hoofdstuk 3. Deze resultaten zijn ook opgenomen in het Analyse Document. Er was gepland om ook te verdiepen in de eisen van de opdrachtgever, maar op dat moment werd duidelijk dat er niet speciale eisen waren, maar er een document werd bijgehouden waarin stond wat er gemaakt moest worden. Verder is er alvast een deel van de Construction fase gedaan, aangezien er gevraagd was om alvast een aantal delen van de software te maken. Dit heeft ervoor gezorgd dat de Construction fase en de Elaboration fase door elkaar heen liepen. Ten slotte is er een planning gemaakt voor de Construction fase, waarna deze is opgenomen in het Analyse Document. Het proces hierachter is beschreven in hoofdstuk 4.

6.3 Onjuist gebruik van fases RUP

Wat er in deze fase nog mist is een analyse voor het systeem dat ontwikkeld diende te worden. Dit is echter gedaan in de Inception fase. De reden hiervoor is dat de planning van het project van Siemens en mijn planning niet synchroon liepen. Het staat meerdere keren beschreven in dit verslag dat dit ervoor heeft gezorgd dat de fases van RUP niet helemaal zijn uitgevoerd zoals het zou moeten zijn. Hierdoor was het voor mij erg lastig om te bepalen wat er in elke fase moest gebeuren. Hierdoor is de planning zoals gemaakt is in het Vision Document en Analyse Document niet helemaal gehandhaafd. Het resultaat hiervan is dat ervoor gekozen is om wat minder te focussen op de fases van RUP. De werkzaamheden zoals deze beschreven zijn in hoofdstuk 5 zijn door deze redenen ook gedeeltelijk in deze fase gedaan. Echter voor het leesgemak is ervoor gekozen om dit volledig in het hoofdstuk van de Inception fase te beschrijven.

7 Construction fase

De construction fase was eigenlijk bedoeld voor het maken van het ontwerp, die zou helpen bij het programmeren. Echter was er in de Inception fase al iets gedaan aan het ontwerp. Dit deel zou in eerste instantie nodig zijn voor een engineer van Siemens. In de Construction fase is duidelijk geworden dat met de informatie die was verkregen nog niks was gedaan. Bij nader inzien is dit ook vanzelfsprekend geweest, aangezien het een ontwerp betreft over TeleControl. De engineer was al begonnen aan het programmeren van TeleControl, maar dit zou door mij worden overgenomen. Die overname gebeurde in deze fase.

7.1 Aanmaken datatypes TeleControl

Door het ontwerp, waarvan in hoofdstuk 5 in Tabel 3 een deel is weergegeven, is er volledig bekend geworden welke datatypes van TeleControl er aangemaakt moesten worden. Deze datatypes moesten in de PLC nog aangemaakt worden. Dit is gedaan met behulp van "PLC data types". Hieronder in Figuur 7.1 is een voorbeeld te zien hoe het datatype SLi_BO.s8 (type B#16#08) is aangemaakt. Er is te zien hoe de opbouw van het datatype in elkaar zit. Er worden dus vier bytes verstuurd met dit datatype. Daarnaast wordt er een timestamp meegestuurd. Deze opbouw is te herleiden uit het type (B#16#08) van het datatype. Hoe dit werkt is terug te lezen in paragraaf 5.2.

IEC-BO-Type-08		
	Name	Data type
1	Pattern_1	Byte
2	Pattern_2	Byte
3	Pattern_3	Byte
4	Pattern_4	Byte
5	TimeStamp	Date_And_Time

Figuur 7.1: SLi_BO.s8 (type 08)

7.2 Timestamp

Vóór deze fase was er meeg geholpen aan het maken van de software. Dit is zo besloten zodat er niet een overname plaats zou vinden zonder dat er kennis was van de code zoals het tot dan toe was geprogrammeerd. Er is onder andere meeg geholpen met het tussendoor testen van de software en met problemen die hiermee naar boven kwamen. Een voorbeeld hiervan is het probleem dat er geen telegramen verstuurd werden via de gemaakte TeleControl verbinding. Er bleek een verkeerde timestamp meegegeven te worden met de telegramen. Dit is naar boven gekomen door te kijken naar de code. Elk telegram heeft zijn eigen timestamp, zoals te zien is in Figuur 7.2. Echter wanneer er niks in de timestamp wordt meegegeven, zal er geen telegram verstuurd worden. Nadat dit in de code verwerkt was, werden de telegramen wel verstuurd.

■	▼ Data[1]	"IEC-ME-Type-10"
■	Data	Real
■	TimeStamp	Date_And_Time

Figuur 7.2: Timestamp

7.3 WinCC verbinding

Aan het begin van de Construction fase moest er een WinCC project gemaakt worden. Hierin moest er een TeleControl verbinding opgezet worden. Dit was echter voor een deel al gedaan vóór deze fase. De rest van het WinCC project is in deze fase overgedragen aan de opdrachtgever, die het project verder heeft afgemaakt. Hierdoor kon er volledig gefocust worden op het programmeren van de PLC code.

7.4 STL of SCL

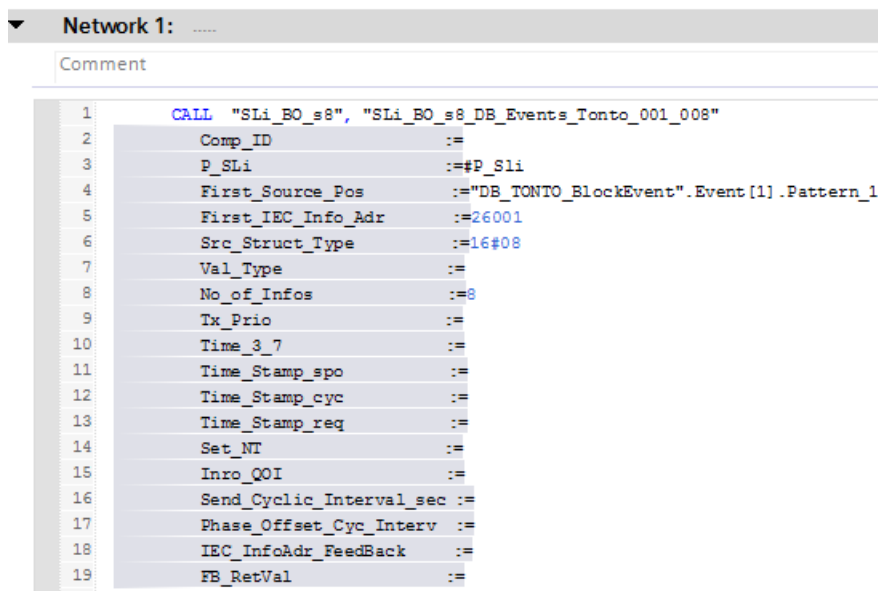
Een deel van deze code was al gemaakt. Echter was dit niet gebaseerd op mijn ontwerp. Zoals in Tabel 3 te zien is, zijn er voor bijvoorbeeld de counters 240 blokken nodig. Dit betekent dat er 240 keer geprogrammeerd moet worden dat er een TeleControl blok aangeroepen wordt. Hier is geen rekening mee gehouden door de engineer die tot die tijd de code had gemaakt voor TeleControl. Het Function Block (FB) dat de engineer had gemaakt was in de taal Statement List (STL). Als er 240 keer een FB-aanroep geprogrammeerd moet worden, betekent dit dat er 240 netwerken gemaakt moeten worden in het FB. Dit is erg veel knip en plak werk. En dat zijn dan alleen nog de netwerken voor de counters van de Tonto's. Dit probleem is opgelost door gebruik te maken van de programmeertaal Structured Control Language (SCL) in combinatie met Excel. SCL is namelijk regels pseudocode onder elkaar typen. In Figuur 7.3 is één van de oproepen in SCL te zien en in Figuur 7.4 is dezelfde oproep in STL te zien.

```

261 IF #ShiftValue = 1 THEN
262     "SLi_BO_s8_DB_Events_Tonto_001_008"(P_SLi := #P_SLi,
263                                         First_Source_Pos := "DB_TONTO_BlockEvent".Event[1].Pattern_1,
264                                         First_IEC_Info_Adr := 26001,
265                                         Src_Struct_Type := 16#08,
266                                         No_of_Infos := 8);
267 END_IF;

```

Figuur 7.3: Eén aanroep geprogrammeerd in SCL



Figuur 7.4: Eén aanroep geprogrammeerd in STL

Zoals te zien in de twee figuren is bij het gebruik van SCL weinig code nodig voor één aanroep. Daarnaast kan alle code onder elkaar wordt geplakt. Door het gebruik van Excel kan de volledige code automatisch worden gegenereerd. In Figuur 7.5 is te zien hoe drie aanroepen onder elkaar in Excel zijn gegenereerd.

236	IF #Shift_Value = 1 THEN	IF #Shift_Value =	1		THEN
237	SLI_BO_s8_DB_Events_Tonto_001_008(P_SLi:=#P_SLi,	SLI_BO_s8_DB_Events_Tonto_	1 001	8 008	(P_SLi:=#P_SLi,
238	First_Source_Pos:=DB_TONTO_BlockEvent".Event[1].Pattern_1,	First_Source_Pos:=DB_TONTO_BlockEvent".Event[]Pattern_1,
239	First_IEC_Info_Adr:=26001,	First_IEC_Info_Adr:=26			,
240	Src_Struct_Type:=16#08,	Src_Struct_Type:=16#08,			
241	No_of_Infos:=8);	No_of_Infos:=8);			
242	END_IF;	END_IF;			
243					
244	IF #Shift_Value = 2 THEN	IF #Shift_Value =	2		THEN
245	SLI_BO_s8_DB_Events_Tonto_009_016(P_SLi:=#P_SLi,	SLI_BO_s8_DB_Events_Tonto_	9 009	16 016	(P_SLi:=#P_SLi,
246	First_Source_Pos:=DB_TONTO_BlockEvent".Event[9].Pattern_1,	First_Source_Pos:=DB_TONTO_BlockEvent".Event[]Pattern_1,
247	First_IEC_Info_Adr:=26009,	First_IEC_Info_Adr:=26			,
248	Src_Struct_Type:=16#08,				
249	No_of_Infos:=8);				
250	END_IF;	END_IF;			
251					
252	IF #Shift_Value = 4 THEN	IF #Shift_Value =	4		THEN
253	SLI_BO_s8_DB_Events_Tonto_017_024(P_SLi:=#P_SLi,	SLI_BO_s8_DB_Events_Tonto_	17 017	24 024	(P_SLi:=#P_SLi,
254	First_Source_Pos:=DB_TONTO_BlockEvent".Event[17].Pattern_1,	First_Source_Pos:=DB_TONTO_BlockEvent".Event[]Pattern_1,
255	First_IEC_Info_Adr:=26017,	First_IEC_Info_Adr:=26			,
256	Src_Struct_Type:=16#08,				
257	No_of_Infos:=8);				
258	END_IF;	END_IF;			

Figuur 7.5: SCL code gegenereerd vanuit Excel

Het enige wat er nog moest gebeuren is de benodigde cellen kopiëren en plakken onder elkaar. Zo kon eenvoudig alle code worden gegenereerd. Door een simpele copy-paste van Excel naar het TIA Portal project waren de aanroepen in een veel kortere tijd gemaakt in SCL dan met STL had gekund.

7.5 WinCC tags

Voor het WinCC project heeft de opdrachtgever aan mij gevraagd om nog een klein deel te maken. De tags die ingevoerd moesten worden in het WinCC Tag Management waren er redelijk veel. De tags die via het TCP protocol over moesten komen konden automatisch uit de PLC geladen worden, echter de tags die via TeleControl overkwamen moesten handmatig worden ingevoerd. Ook hiervoor is Excel gebruikt. Door het exporteren van de lege taglijst naar een Excel-bestand werd er alvast een opmaak gemaakt. Het enige wat er toen nog moest gebeuren is het kopiëren en plakken van de tagnamen uit het ontwerp naar het Excel-bestand. De tags waar er meerdere van gemaakt moesten worden, hadden allemaal op het einde doorlopende nummers staan, de suffix (bijvoorbeeld: 1_TotaalAantalDesinfectie_1, ...2, ...3, enzovoorts), hierdoor konden deze tags in Excel direct worden gegenereerd door de cellen naar beneden door te trekken.

7.6 Aanmaken DB's

Voor elke aanroep van een TeleControl Block type moet er een Instance Data Block (DB) aangemaakt worden. Helaas kunnen die niet simpel aangemaakt worden. Er is gezocht naar manieren om bijvoorbeeld deze DB's aan te maken via SCL code of door .txt-bestanden te maken waar de DB's in aangemaakt konden worden. De versie van TIA Portal (V13 SP1 Update 8) die gebruikt is tijdens dit project had niet de mogelijkheid om op een snelle manier meerdere Instance DB's aan te maken. De eerder genoemde optie om de DB's aan te maken via een SCL code was wel mogelijk, alleen kon daar óf het DB-nummer opgegeven worden óf de naam van de DB, maar niet allebei. Er is uiteindelijk voor gekozen om alsnog de DB's op deze manier te genereren, met de juiste DB nummers. De DB nummers waren namelijk al vastgesteld in het ontwerp. Daarnaast zou TIA Portal alle DB nummers automatisch selecteren, waardoor het voor kon komen dat er meerdere dezelfde DB nummers voor zouden komen. Wanneer alle DB's aangemaakt waren moesten alle namen veranderd worden. Hiervoor is er gekozen om dit te doen in de Portal View van TIA Portal. Een andere optie had kunnen zijn om alle DB's in Project View te hernoemen, maar dan staan niet alle DB's netjes onder elkaar gesorteerd. De Portal View is hierin overzichtelijker. In de Portal View kan er namelijk onder "PLC programming" een lijst getoond worden met alle blokken, waaronder alle DB's. Het veranderen van alle namen was verder een kwestie van copy-paste en telkens een aantal nummers in de naam veranderen.

7.7 ShiftValue

Zoals in Figuur 7.3 al te zien is, wordt er gebruik gemaakt van een “#Shift_Value” als if-statement om een aanroep van een TeleControl Block daadwerkelijk te doen. Deze ShiftValue wordt gecontroleerd op een bepaalde waarde. De ShiftValue is in principe een statische waarde. In Figuur 7.6 is te zien hoe een ROR (Rotate Right) commando de actieve bit in de variabele opshift. Stel dat waarde in binaire code 00000001 is, wordt dit na het commando 00000010. De decimale waarde verandert dus van 1 naar 2. Hierna verandert deze weer naar 4, 8, 16, enzovoorts.

```
1 // To prevent a deadlock
2 IF #ShiftValue = 0 THEN
3     #ShiftValue := 128; // next will be 1...
4 END_IF;
5
6 #ShiftValue := ROR(IN := #ShiftValue, N := 1); // Make a trigger so that not all blocks are called every cycle
```

Figuur 7.6: ROR Commando

De reden voor deze ShiftValue is dat de aanroepen hierdoor niet allemaal tegelijk worden uitgevoerd. Dit was in eerste instantie wel het geval, waardoor de cyclustijd van de PLC opliep naar een tijd van meer dan 200 ms. Normaal gesproken zou dit een cyclustijd van rond de 30 ms zijn. Nu is een cyclustijd van 200 ms wellicht niet schrikbarend, maar het kon veel minder. Als het sneller kan, waarom zou de code dan niet zo gemaakt worden dat het ook daadwerkelijk sneller is. Naar mijn mening maakt dat de code mooier en sneller. Met deze instelling is ervoor gekozen om deze ShiftValue te maken. Zoals in Figuur 7.3 en Figuur 7.5 is te zien wordt er per aanroep een andere ShiftValue als IF-statement gebruikt. Er is keuze uit 1, 2, 4, 8, 16, 32, 64 en 128. Deze waardes worden verdeeld over alle aanroepen. Hierdoor worden per cyclus weer andere aanroepen uitgevoerd, maar niet meer allemaal tegelijk.

Verder is er in Figuur 7.6 te zien dat er een stuk code is geschreven om een deadlock te voorkomen. Bovenin de FB is een initialisatiewaarde ingesteld voor de ShiftValue variabele. Echter kan het voorkomen dat de PLC na een korte reset wellicht de waarde terugzet op 0. De kans dat dit gebeurt is waarschijnlijk zeer klein, maar er zijn slechts drie regels code nodig om het systeem robuust te maken. De keuze om dit toch te programmeren was voor mij hierdoor snel gemaakt. Het stuk code zorgt ervoor dat, wanneer de waarde van de variabele per ongeluk op 0 terechtkomt, de waarde automatisch op 128 wordt gezet. Hierdoor wordt de waarde de eerstvolgende cyclus weer door het ROR commando op 1 gezet.

7.8 Verbetering

Zoals de code uiteindelijk is opgeleverd, werkt het zoals het zou moeten doen. Echter zijn er nog bepaalde stukken in de code waarvan persoonlijk wordt gedacht dat deze nog mooier of sneller hadden gekund. Een voorbeeld hiervan is alle aanroepen van TeleControl die door mij zijn geprogrammeerd. Op dit moment wordt de ShiftValue gebruikt zoals in de vorige paragraaf is uitgelegd. Echter is hierbij simpelweg een byte genomen waarvan het bitje wordt verschoven. Dit betekent dat er acht groepen ontstaan. Er is gebleken dat dit naar behoren werkt. Het zou alleen ook zo gemaakt kunnen worden dat niet alle bits van de byte gebruikt worden. Dit betekent dat er minder dan acht groepen ontstaan. Resultaat hiervan is dat een aanroep sneller achter elkaar wordt gebruikt. Het versturen van telegrammen zal hierdoor versnellen. Verder zou het wellicht handig kunnen zijn om het aantal aanroepen variabel te maken. Als er bijvoorbeeld maar 150 Tonto's in een ziekenhuis staan, worden met dit systeem alsnog alle aanroepen doorlopen. Alle aanroepen gelden in totaal voor 240 Tonto's. Dit betekent dat er meerdere aanroepen constant voor niks gebruikt worden. Als dit variabel gemaakt wordt, kan er bij bijvoorbeeld 150 Tonto's gezorgd worden dat alleen de aanroepen gebruikt worden die nodig zijn voor die 150 Tonto's. Ook dit zal het proces versnellen.

8 Transition fase

De Transition fase is de laatste fase van RUP en dus ook de laatste fase die is doorlopen tijdens de afstudeerperiode. In deze fase was het de bedoeling dat er een testplan/rapport gemaakt werd en de overdracht van de software plaats zou vinden. In dit geval werd er in eerste instantie bedoeld de overdracht van Siemens aan Pharmafilter, maar dit is uiteindelijk de overdracht van de afstudeerder aan Siemens geworden.

8.1 Testplan/rapport

De fase begon met het maken van het testplan. Het was niet heel lastig om dit voor elkaar te krijgen. Wat de software uiteindelijk moest kunnen kon zo worden overgenomen uit het ontwerp. Daarnaast hadden de engineers van Siemens een eigen testplan gemaakt voor de acceptatietest van het gehele systeem. Hieronder viel ook het deel dat door mij gemaakt is. Uit dit testplan kon daardoor ook informatie gehaald worden. Het testplan is opgesteld op de manier hoe dit aangeleerd is op school. Het moest reproduceerbaar zijn en er moest duidelijk zijn wat er diende te gebeuren. Om erachter te komen of dit het geval was, is het testplan door mij doorlopen alsof het allemaal nieuw voor me was. Het testplan bestond uiteindelijk uit een lijstje met eisen/testgevallen die in een tabel zijn opgesteld, zie Tabel 4.

Tabel 4: Testplan

Test #	Details	Verwachte uitvoer	Daadwerkelijke uitvoer	OK / NOK	Opmerkingen
1	Trek de netwerkkabel uit de WinCC server.	Komend alarm op WinCC wordt getoond: Communicatiestoring WinCC <>Datacollector.			
2	Verander in de Tonto een aantal alarmen en/of statussen.	De verandering wordt niet op WinCC getoond.			
3	Verander in de RWZI een aantal alarmen en/of statussen.	De verandering wordt niet op WinCC getoond.			
4	Verander in de RWZI een teller- en of meetwaarde.	De verandering wordt niet op WinCC getoond.			
5	Sluit de netwerkkabel weer aan op de WinCC server.	Gaand alarm op WinCC wordt getoond: Communicatiestoring WinCC <>Datacollector.			
6	Verander in de Tonto een aantal alarmen en/of statussen.	Alarmen en/of statussen dienen met de juistedatum- en tijdstempel in het alarmarchief (Alarm Logging) worden getoond.			
7	Verander in de RWZI een aantal alarmen en/of statussen.	Alarmen en/of statussen dienen met de juistedatum- en tijdstempel in het alarmarchief (Alarm Logging) worden getoond.			
8	Verander in de RWZI een teller- en of meetwaarde.	Tellerstand en/of meetwaarde dient met de juistewaarde en datum- en tijdstempel in het archief (Tag Logging) worden getoond.			

Dit testplan is doorlopen door mijn opdrachtgever samen met Pharmafilter en mij. Alle testgevallen waren OK uit de test gekomen. Hierdoor verschilt het testrapport niet veel van het testplan op de ingevulde vakken van de tabel na. Het testrapport is uiteindelijk het bewijs dat de software werkt zoals het hoort te werken.

8.2 In bedrijf stellen

Naast het testplan/rapport was het ook de bedoeling dat er meegeholpen zou worden aan de overdracht van het systeem. Het in bedrijf zetten van het systeem bij het Reinier de Graaf ziekenhuis. Dit is helaas wegens de tijd niet meer doorgegaan. Hierdoor is de overdracht veranderd naar het overdragen van de software aan Siemens. De uitkomst hiervan is dat de overdracht een stuk sneller is gegaan dan van tevoren is geschat. De Transition fase is hierdoor ongeveer een week eerder afgelopen dan verwacht. De tijd die hierdoor vrij kwam is besteed aan het afstudeerverslag.

9 Extra werkzaamheden

Naast de werkzaamheden zoals beschreven in de voorgaande hoofdstukken, zijn er nog andere opdrachten uitgevoerd. In de vorige hoofdstukken zijn de werkzaamheden beschreven die direct te maken hadden met mijn deel van het project. Daarnaast is er echter ook meegewerkt aan andere delen van het project. Dit heeft ervoor gezorgd dat de tijd van de afstudeerperiode volledig is gevuld met opdrachten. In dit hoofdstuk zullen een aantal van deze werkzaamheden nog worden doorgenomen.

9.1 WinCC

Aan het begin van de afstudeerperiode, in de Inception fase, is er aan mij gevraagd om een deel van de WinCC omgeving te maken. Er moest onder andere uitgezocht worden welke datatypes er gebruikt moesten worden. Over het kiezen van de datatypes is in hoofdstuk 5 al het een en ander beschreven. Dat gaat echter enkel over de kant van de PLC. Daarnaast moest er uitgezocht worden wat dan de datatypes moesten zijn aan de kant van WinCC. Aan het begin was dit nog onduidelijk aangezien er nog niet bekend was welke datatypes er gebruikt zouden worden aan de kant van de PLC. Het uitzoeken voor WinCC heeft dus daarna plaatsgevonden. In Tabel 2 in hoofdstuk 5 is te zien welke TeleControl blokken er gebruikt zijn in de PLC. Bij al deze blokken passen meerdere datatypes die verstuurd konden worden. Degenen die uiteindelijk zijn gebruikt zijn hieronder te vinden in Tabel 5. Dit is bepaald door de engineers van Siemens in samenwerking met mij en dit staat in hun ontwerp. Echter hebben wij dit bepaald nadat er door mij was uitgezocht welke types er gebruikt konden worden in combinatie met TeleControl.

Tabel 5: Datatypes gebruikt in PLC

Data	Datatype
Alarmen vanuit Tonto	32 bits
Events vanuit Tonto	32 bits
Counters vanuit Tonto	1 real
Counters vanuit RWZI	1 real
Processatus vanuit RWZI	8 bits
Meetwaardes vanuit RWZI	1 real

Hierna kon er aan de WinCC kant worden bepaald welke datatypes er daar ingesteld moesten worden. Er was keuze uit de volgende types:

- Binary Tag
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag 8-bit character set
- Text tag 16-bit character set
- Raw Data Type
- Date/Time

Voor elke soort data dat verstuurd en ontvangen diende te worden was een eigen reden om een bepaald datatype van de lijst hierboven te gebruiken. Om dit aan te geven is Tabel 5 uitgebreid. De resultaten hiervan zijn te vinden in Tabel 6 hieronder.

Tabel 6: Datatypes gebruikt in WinCC met reden

Data	Datatype PLC	Datatype WinCC	Reden
Alarmeren vanuit Tonto	32 bits	Unsigned 32-bit value	De alarmeren worden uit een rij van 32 bits gehaald. Voorbeeld: stel de laatste vier bits zijn eerst 0001, dan betekent dit dat alarm01 afgaat. Als dit nu verandert naar 0101, dan betekent dit dat alarm01 en alarm03 afgaan. Alle bits staan dus voor afzonderlijke alarmeren.
Events vanuit Tonto	32 bits	Unsigned 32-bit value	De events worden uit een rij van 32 bits gehaald. Voorbeeld: stel de laatste vier bits zijn eerst 0001, dan betekent dit dat event01 actief is. Als dit nu verandert naar 0101, dan betekent dit dat event01 en event03 actief zijn. Alle bits staan dus voor afzonderlijke events.
Counters vanuit Tonto	1 real	Floating-point number 32-bit IEEE 754	De counters zijn van een bepaalde waarde die constant hoger wordt. Aangezien er een real wordt verstuurd vanuit de PLC, moet hier een 32-bit floating-point nummer worden ontvangen.
Counters vanuit RWZI	1 real	Floating-point number 32-bit IEEE 754	De counters zijn van een bepaalde waarde die constant hoger wordt. Aangezien er een real wordt verstuurd vanuit de PLC, moet hier een 32-bit floating-point nummer worden ontvangen.
Processus vanuit RWZI	8 bits	Unsigned 8-bit value	De status van de RWZI wordt op dezelfde manier overgebracht als de alarmeren en events van de Tonto's. Echter gaat het hier om slechts 8 bits.
Meetwaarden vanuit RWZI	1 real	Floating-point number 32-bit IEEE 754	De meetwaarden zijn van een bepaalde waarde die constant verandert. Aangezien er een real wordt verstuurd vanuit de PLC, moet hier een 32-bit floating-point nummer worden ontvangen.

9.2 PLC code

Naast WinCC is er ook een deel van de PLC code gemaakt dat niet direct te maken had met mijn deel van het project. Een aantal voorbeelden hiervan zijn reeds beschreven in hoofdstuk 7.

Er is ook een stuk code geschreven om het totaal aantal Tonto's te kunnen verdelen over een aantal statussen. Er kunnen in totaal 240 Tonto's in een systeem voorkomen. Hiervan zijn er slechts een aantal in gebruik. Ook kunnen er Tonto's in storing staan of een andere soort fout hebben. Er moest dus gecontroleerd worden wat de status was van elke Tonto. Hiervoor is het stuk code geschreven zoals te zien in Figuur 9.1.

```
REPEAT // Update different counters
  IF #NrAvailable[#TontoChecks] THEN
    #NumberOfTonto := #NumberOfTonto + 1;
  END_IF;

  // Check the number of Tonto's OK
  IF "DB_LocationStatusTonto".LocatieStatuswoord[#TontoChecks].%X0 = 1 THEN
    "DB_LocationStatusTonto".AantalTontosOK := "DB_LocationStatusTonto".AantalTontosOK + 1;
  END_IF;

  // Check the number of Tonto's Storing
  IF "DB_LocationStatusTonto".LocatieStatuswoord[#TontoChecks].%X1 = 1 AND #TontoChecks < 231 THEN
    "DB_LocationStatusTonto".AantalTontosStoring := "DB_LocationStatusTonto".AantalTontosStoring + 1;
  END_IF;

  // Check the number of Tonto's Spare
  IF "DB_LocationStatusTonto".LocatieStatuswoord[#TontoChecks].%X2 = 1 THEN
    IF NOT("DB_LocationStatusTonto".LocatieStatuswoord[#TontoChecks].%X1) THEN
      "DB_LocationStatusTonto".AantalTontosSpare := "DB_LocationStatusTonto".AantalTontosSpare + 1;
    ELSE
      "DB_LocationStatusTonto".AantalTontosSpareStoring := "DB_LocationStatusTonto".AantalTontosSpareStoring + 1;
    END_IF;
  END_IF;

  // Check the number of Tonto's not parametrized
  IF "DB_LocationStatusTonto".LocatieStatuswoord[#TontoChecks].%X15 = 1 THEN
    "DB_LocationStatusTonto".AantalLocatiesNietGeparametreerd := "DB_LocationStatusTonto".AantalLocatiesNietGeparametreerd + 1;
  END_IF;

  #HospitalTotal := #HospitalTotal + "DB_Crush".CountLocation[#TontoChecks]; // Count all crushes

  #TontoChecks := #TontoChecks + 1; // Check the IP / Locations 1...240
UNTIL #TontoChecks = 241 END_REPEAT;
```

Figuur 9.1: Code

10 Evaluatie

In dit hoofdstuk zal er een evaluatie te vinden zijn over de afstudeerperiode. Er zal worden gekeken of de manier van plannen de juiste is geweest tijdens het maken van het Vision Document. Daarnaast zal er ook voor alle andere tussenproducten een evaluatie beschreven worden.

10.1 Vision Document

Het Vision Document is gemaakt aan het begin van de afstudeerperiode. Hierin is een planning gemaakt voor de rest van de periode. Er is onder andere voor gekozen om gebruik te maken van RUP. RUP kent in totaal vier fases. Hieronder wordt geëvalueerd hoe de fases van RUP zijn verlopen, ten opzichte van de planning, tijdens de afstudeerperiode.

Inception fase De Inception fase is de eerste fase van RUP. In deze fase wordt er een plan gemaakt voor het project wat uitgevoerd zal worden. Voor deze fase waren er drie weken gepland. Uiteindelijk is gebleken dat dit geen correcte planning was. Door het binnenvallen tijdens een lopend project heeft er aan het begin een kennisoverdacht plaatsgevonden. Er is volledig uitgelegd wat de stand van zaken was voor het project op dat moment en wat er van mij werd verwacht. Echter door het niet synchroon lopen van mijn planning en dat van Siemens, zijn er iteraties uitgevoerd in fases waar dit niet had moeten gebeuren. Zoals in Hoofdstuk 5 en 6 is te lezen zijn deze fases hierdoor door elkaar heen gelopen. Toch is er zoveel mogelijk voor gezorgd dat de mijlpalen van elke fase zijn bereikt. Uiteindelijk heeft dit aan het einde van de Inception fase geresulteerd in een Vision Document die is goedgekeurd door de bedrijfsbegeleider, die tevens mijn opdrachtgever was. Helaas is er, zoals ook beschreven in hoofdstuk 5, een periode van ziekte geweest tijdens de Inception fase. Dit heeft er ook voor gezorgd dat de planning niet volledig gehanteerd kon worden. Deze periode van ziekte heeft ongeveer drie weken aangehouden. Hierdoor zijn er drie weken van mijn afstudeerperiode verloren gegaan. Er is geprobeerd om dit verlies op te vangen door de rest van de periode extra inspanning te leveren. Dit heeft voor mij de afstudeerperiode nog lastiger gemaakt.

Elaboration fase In het Vision Document is een planning opgenomen waarin was aangegeven dat de Elaboration fase drie weken zou duren. In deze fase was het de bedoeling dat er veel geanalyseerd zou worden. Doordat de planning van het project voor Siemens anders liep, is deze fase niet helemaal gelopen zoals in het Vision Document is aangegeven. Er moest vervroegd worden ontworpen en geprogrammeerd, waardoor het analyseren af en toe is verschoven. Hierdoor is de Elaboration fase voor een deel gemengd met de Inception fase, zoals hierboven benoemd, maar ook met de Construction fase. Er is tijdens de Elaboration fase al een ontwerp gemaakt en er is al het een en ander geprogrammeerd. Doordat er twee fases door elkaar heen zijn gelopen heeft deze fase vijf weken geduurd in plaats van de geplande drie weken. Dit is echter niet erg aangezien er al taken van de Construction fase zijn opgepakt in deze periode. Tijdens de Elaboration fase is er ondanks dat, wel een Analyse Document gemaakt, waarin een nieuwe planning is opgenomen. Hiermee zouden de verschuivingen worden opgevangen.

Construction fase Deze fase is vroeger begonnen dan in de planning van het Vision Document is vastgesteld. Echter in het Analyse Document is er een nieuwe planning opgenomen. Hierin is voor de Construction fase zeven weken gepland, vanaf het einde van de Elaboration fase, in plaats van de negen weken die er eerst gepland waren. Doordat er taken van de Construction fase al in de Elaboration fase zijn uitgevoerd, kon dit zo worden toegepast.

Transition fase In de Transition fase is het Testplan doorlopen op de software die is geprogrammeerd. Verder was het de bedoeling dat in deze twee weken de overdracht zou plaatsvinden van de software van Siemens aan Pharmafilter. Uiteindelijk is dit waarschijnlijk de enige fase die volledig is uitgevoerd zoals van te voren verwacht werd. Er is namelijk een testplan/testrapport gemaakt en de overdacht heeft in deze fase plaatsgevonden.

Opdrachtschrijving Tijdens het maken van de opdrachtschrijving voor in het Vision Document is er veelvuldig contact geweest van de opdrachtgever. Dit was niet erg lastig aangezien de opdrachtgever aan de andere kant van het bureau zat. Deze manier van werken was erg fijn, aangezien er zo constant contact gelegd kon worden met de opdrachtgever. Zoals in paragraaf 5.1 al is beschreven is de opdrachtschrijving meerdere keren aangepast. In het Vision Document is bijvoorbeeld te lezen dat er een SIMATIC Process Historian geprogrammeerd moet worden. Dit is uiteindelijk veranderd, aangezien de Process Historian door Pharmafilter zelf zou worden geprogrammeerd. Ten slotte is er in het Vision Document geen risico analyse opgesteld. Er werd aan het begin verondersteld dat dit niet nodig zou zijn. Bij nader inzien was het wel verstandig geweest om een risico analyse te maken, aangezien de planning van mij niet gelijk liep met de planning van Siemens en er een periode van ziekte is geweest tijdens de afstudeerperiode.

10.2 Analyse document

Om kennis op te doen over het bedrijf waar voor gewerkt werd en over het project zelf is er een analyse gedaan. De manier waarop deze analyse is uitgevoerd is nuttig geweest voor mijn afstudeerperiode. Door de analyse is de kennis verkregen die van tevoren nodig was om het project uit te kunnen voeren. Er hadden eventueel nog wat extra contactmomenten plaats kunnen vinden met de opdrachtgever en/of Pharmafilter. Echter was het document dat gebruikt is voor de analyse voldoende om genoeg informatie te verkrijgen en aan de slag te kunnen. Het Analyse Document is een document dat niet is gekeurd door de opdrachtgever. Naar mijn idee was dit niet nodig, aangezien het een document is die voor eigen gebruik is gemaakt. Het verzamelen van informatie en dit verkort opnemen in een document heeft een leerzaam effect gehad, waarmee het doel van het document ook is bereikt. Na het maken van het Analyse Document was er duidelijkheid over het project waaraan en het bedrijf waarvoor er gewerkt werd. Daarnaast was het Analyse Document een mooie afsluiting van de Elaboration fase van RUP. Hierdoor is er in het document gelijk een planning opgenomen voor de Construction fase die erna zou komen. Voor mij was het maken van het Analyse Document een handig hulpmiddel tijdens de afstudeerperiode.

10.3 Ontwerp

Hoe het maken van een ontwerp is verlopen had voor mij liever op een andere manier gegaan. Er was niks aan te doen dat het project al een maand bezig was toen de afstudeerperiode begon bij Siemens. Hierdoor was er dus al een deel van het ontwerp gemaakt. Gelukkig was er voor het deel van het project dat voor mij bestemd was nog niks gemaakt. Toch was het gevoel er dat er een achterstand was. Er moest direct aan het begin van de periode een ontwerp op tafel komen. Dit was erg lastig door het gebrek aan kennis van het project. Door deze overlapping van plannings is er een deel van mijn ontwerp al gemaakt voordat de Construction fase begon. Dit is ook te lezen in hoofdstuk 5. Daarnaast kon het ontwerp dat is gemaakt niet direct afgeleid worden vanuit de eisen, door de manier van werken tussen Siemens en Pharmafilter. Er waren geen duidelijke eisen op papier gezet. Voor mijn deel van het project was dat niet erg, aangezien Siemens hier de vrije hand in had gekregen. Siemens mocht bepalen hoe TeleControl geïmplementeerd zou worden. Wij hebben ervoor gezorgd dat dit naar onze mening zo efficiënt mogelijk zou gebeuren. Hiermee wordt bedoeld dat er onder andere niet te veel geheugen op zou gaan aan TeleControl. Hierbij waren wij echter wel weer gebonden aan de mogelijkheden van TeleControl. Hieruit is een ontwerp gekomen in de vorm van een Excel bestand. Daarnaast is er een schema gemaakt waarin te zien is hoe de data via TeleControl tussen de PLC en de WinCC omgeving gecommuniceerd wordt. Zowel de Excel tabellen als het schema zijn na controle van de opdrachtgever opgenomen in het ontwerp document van Siemens.

10.4 PLC code

De code is naar mijn mening gemaakt zoals dit werd verwacht. Het ontwerp van Siemens is volledig gehanteerd en er is tussendoor getest, waardoor er steeds gevalideerd kon worden of de software nog voldeed aan de eisen van Siemens en Pharmafilter. In paragraaf 7.8 is te lezen wat er naar mijn mening nog eventueel beter had gekund in de code.

10.5 Testplan

Het testplan/testrapport is een klein tussenproduct geweest. Het heeft mij persoonlijk geholpen met het testen van de software dat was gemaakt. Hierdoor kon er nagegaan worden of de software voldeed aan de eisen zoals deze uit het ontwerp te halen waren.

10.6 Extra werkzaamheden

De extra werkzaamheden die zijn uitgevoerd tijdens mijn afstudeerperiode hebben ervoor gezorgd dat er tijd is opgevuld dat anders leeg zou zijn geweest. Hierdoor is er constant werk geweest wat gedaan kon worden. Als er de gehele periode lang alleen maar gefocust zou worden op de opdracht zoals deze van tevoren beschreven is, waren daar waarschijnlijk geen 17 weken mee gevuld. Door dit soort opdrachten te doen is er meer kennis opgedaan over het programmeren en meewerken aan een project. Het maakte de afstudeerperiode ook leuker, door de afwisseling dat het veroorzaakte.

Referenties

- [Batelaan et al., 2012] Batelaan, M. V., Van den Berg, E. A., Koetse, E., Wortel, N. C., Rimmelzwaan, J., and Vellinga, S. (2012). *Evaluatierapport pharmafilter : full scale demonstratie in het Reinier de Graaf Gasthuis, Delft*. Amersfoort : STOWA.
- [Van der Hoek et al., 2013] Van der Hoek, J. P., Van Alphen, J., Kaas, R., and Van der Oost, R. (2013). Geneesmiddelen in de watercyclus. <https://www.ntvg.nl/artikelen/geneesmiddelen-de-watercyclus>. [Online; accessed 9-February-2016].

Begrippenlijst

DB Data Block. 24

FB Function Block. 23, 25

IOA Information Object Address. 13, 20

RWZI rioolwaterzuiveringsinstallatie. 6, 7, 14, 16, 18, 20, 27, 28

SCL Structured Control Language. 23, 24

STL Statement List. 23, 24

tag is een naam die wordt gegeven aan een bepaalde variabele. 24

telegram is een pakketje waarin bepaalde data wordt verstuurd via het TeleControl protocol. 13, 15–17, 22, 25

timestamp is de tijd waarop iets plaatsvindt, zoals een verandering van waarde of een verzending of ontvangst van gegevens. 13, 16, 22