



## **Project Inrichting Andonlight Systeem**



**Afstudeerscriptie**

**EPCOR B.V.**

**Versie: 1.16**

**Datum: 03-07-2012**

**Student: Eric IJntema**

**Studentnummer: 1139126**

**Opleiding: Informatica deeltijd aan de HU**

## Managementsamenvatting

Gebleken is dat het huidige Andonlight systeem niet voldoet aan de eisen van de *key-users*. Op basis van dit feit is de volgende hoofdvraag opgesteld:

“Hoe kan bij Epcor B.V. door het vernieuwen van het Andonlight systeem de gebruiksvriendelijkheid verbeterd worden voor de *key-users* aan de hand van hun voorkeuren en welke data moet het systeem opslaan zodat deze geanalyseerd kan worden, zodat op de bevindingen die hieruit voortkomen geanticipeerd kan worden?”

Voor het verder in kaart brengen van de problemen met het huidige systeem en de eisen en benodigde informatie voor een nieuw systeem, zijn de volgende analyse producten opgeleverd:

- Systeemmodel
- Business Process Model
- Nulmeting
- Boomschema's
- Eisen

Met de producten uit de analyse en de hoofdvraag als uitgangspunt zijn voor het ontwerp van een nieuw systeem de volgende producten opgeleverd:

- Use Cases
- Database Model
- Klassendiagrammen
- Sequence Diagrammen
- Software Architectuur

Tot slot is er stilgestaan bij de implementatie van het nieuwe systeem.

Zoals uit de hoofdvraag blijkt zal het nieuwe systeem data gaan verzamelen over het gebruik van de Andonlights. Het nieuwe systeem zal data over de Andonlight problemen en oplossingen verzamelen met behulp van boomschema's. Dit gebeurt doordat de gebruiker een bepaald pad in een boomschema selecteert, met aan het einde van het pad eventueel een opmerking.

Het nieuwe systeem wordt een transactie gebaseerd systeem. Dit garandeert dat de opgeslagen data altijd consistent blijft.

Er is gekozen voor een Systeem dat niet geïntegreerd wordt in het Quantum Controle ERP pakket<sup>1</sup>. De voornaamste redenen hiervoor zijn: de integratie kost veel tijd en geld en er moet samengewerkt worden met een bedrijf in America, wat de nodige risico's met zich meebrengt. Het enige voordeel van het integreren van de *clients* in Quantum Controle<sup>2</sup> is dat de medewerkers die het systeem gebruiken één applicatie minder hebben openstaan op hun *desktop*.

De data zal wel opgeslagen worden in de database van Quantum Controle. De voornaamste reden hiervoor is dat de tabellen voor het nieuwe Andonlight systeem refereren naar tabellen in de Quantum database. Als het Andonlight systeem een eigen database met verwijzingen naar tabellen in een andere database zou hebben, dan zijn de verwijzingen niet referentieel integer. Dit zorgt ervoor dat de Andonlight data inconsistent kan

---

<sup>1</sup> ERP staat voor Enterprise Resource Planning. Een ERP pakket wordt gebruikt ter ondersteuning van alle processen binnen een bedrijf. Een ERP pakket bestaat vaak uit verschillende modules, die gebruikt kunnen worden.

<sup>2</sup> ERP pakket dat door Epcor B.V. gebruikt wordt.

worden. Ook is het moeilijker om Andonlight informatie op te vragen als de data verdeeld is over twee databases. Als de Andonlight tabellen in de Quantum Controle database geplaatst worden doen deze problemen zich niet voor.

# Inhoudsopgave

<b>MANAGEMENTSAMENVATTING</b>	<b>2</b>
<b>INHOUDSOPGAVE</b>	<b>4</b>
<b>INLEIDING</b>	<b>5</b>
<b>1 PROBLEEMSTELLING</b>	<b>6</b>
<b>2 DOELSTELLING</b>	<b>7</b>
<b>3 ANALYSE</b>	<b>8</b>
3.1 SYSTEEMMODEL	8
3.2 BUSINESS PROCES MODEL	12
3.3 NULMETING	16
3.3.1 Data van oude Andonlight database	16
3.3.2 Nulmeting data van testserver	17
3.4 BOOMSHEMA'S	17
3.5 EISEN	20
3.5.1 Functionele eisen	20
3.5.2 Niet functionele eisen	20
<b>4 ONTWERPRESULTATEN</b>	<b>21</b>
4.1 USE CASES	21
4.1.1 Use Case specificatie UC001 'Hulp vragen bij ondersteunende afdeling'	22
4.1.2 Use Case specificatie UC003 'Oplossing invoeren'	26
4.1.3 Use Case specificatie UC004 'Wisselen view dashboard'	28
4.2 DATABASE MODEL	30
4.3 KLASSENDIAGRAMMEN	30
4.3.1 Use Case specificatie UC001 'Hulp vragen bij ondersteunende afdeling'	30
4.3.2 Use Case specificatie UC003 'Oplossing invoeren'	31
4.3.3 Use Case specificatie UC004 'Wisselen view dashboard'	32
4.4 SEQUENCE DIAGRAMMEN	32
4.4.1 Use Case specificatie UC001 'Hulp vragen bij ondersteunende afdeling'	33
4.4.2 Use Case specificatie UC003 'Oplossing invoeren'	36
4.4.3 Use Case specificatie UC004 'Wisselen view dashboard'	37
4.5 SOFTWARE ARCHITECTUUR	37
4.5.1 Logical View	38
4.5.2 Implementation View	39
4.5.3 Deployment View	39
<b>5 IMPLEMENTATIE</b>	<b>41</b>
5.1 SYSTEEMTEST	41
<b>6 ONDERBOUWING</b>	<b>42</b>
6.1 DATABASE EN APPLICATIE KEUZES	42
6.2 DESIGN PATTERN	43
6.3 SYSTEEM GEBASEERD OP TRANSACTIES	44
6.4 BOOMSHEMA'S	44
<b>7 WERKWIJZE</b>	<b>45</b>

<b>8</b>	<b>CONCLUSIE EN AANBEVELINGEN</b>	<b>46</b>
<b>9</b>	<b>EVALUATIE PROCESGANG</b>	<b>47</b>
	<b>BRONVERMELDING</b>	<b>48</b>
	<b>BIJLAGE A: PLAN VAN AANPAK</b>	<b>49</b>
	<b>BIJLAGE B: EVALUATIE EIGEN FUNCTIONEREN</b>	<b>68</b>
	<b>BIJLAGE C: RESULTATEN NULMETING</b>	<b>69</b>

## Inleiding

Epcor B.V. is 100% dochter van Air France/KLM. Er zijn ongeveer 100 mensen werkzaam.

Het primaire proces van Epcor B.V. bestaat uit reparatie en revisie aan pneumatisch aangedreven componenten en Auxiliary Power Units (APU's) van vliegtuigen. Bij Epcor B.V. werken verschillende monteurs in het primaire proces. Een voorbeeld van het proces waar een component doorheen gaat is:

*Write Up & Incoming Test, Disassembly, Cleaning, Modification1, Modification2, Parts Inspection and Write Up, Repair, Part Management, Assembly en Final Test.*

Dit proces vindt plaats op de Shop afdeling.

Verder zijn er voor het primaire proces ondersteunende processen. De ondersteunende processen worden door de volgende afdelingen uitgevoerd: QA (Quality Assurance), MP (Material Planning), Engineering, Logistiek, Magazijn, HRM, Finance/IT, Sales en CAR (Customer Account Representatives) en Management.

Epcor B.V. maakt gebruik van het Lean Six Sigma concept. Dit concept is ontwikkeld bij Toyota Motor Company. Het concept is in grote lijnen gebaseerd op het wegnemen van zeven vormen van verspilling: transport, voorraad, beweging, wachten, *overprocessing*, overproductie en defecten. Lean Six Sigma focust op wat de klant wil. Toegevoegde waarde wordt bepaald door de wensen van de klant. Alles wat hier niet aan bijdraagt is verspilling. De interne processen zijn met deze filosofie ingericht.

Binnen Epcor B.V. wordt gebruikt gemaakt van een Andonlight systeem. Dit systeem draagt bij aan het terugdringen van de verspillingvormen beweging en wachten. Met het huidige systeem kunnen monteurs om hulp vragen bij collega monteurs, de *supervisor* (leidinggevende van de monteurs) en de ondersteunende afdelingen: MP, Magazijn, CAR en Engineering. Hierover meer in de probleemstelling.

De uitvoerder van de projectopdracht is werkzaam op de afdeling Finance/IT die zich naast *finance* bezig houdt met de ICT inrichting van Epcor B.V. De afdeling Finance/IT wordt ondersteund door een externe ICT-dienstverlener.

# 1 Probleemstelling

Om de probleemstelling duidelijk te maken zal eerst de werking van het huidige Andonlight systeem worden toegelicht.

Met het huidige Andonlight systeem kunnen monteurs met behulp van een *client* programma op hun workstation om hulp vragen. Met de *client* kan de monteur de fysieke Andonlight bedienen. De fysieke Andonlight is een licht op een paal die bij de werkplek van de monteur staat. Het licht kan op groen, oranje en rood gezet worden. Als de monteur de Andonlight op groen heeft gezet, is in de Shop afdeling voor iedereen zichtbaar dat de monteur aan het werk is en dat er geen problemen zijn. Als de monteur de Andonlight op oranje zet, wordt er om hulp gevraagd bij de collega monteurs en de *supervisor*. Kan bij de oranje Andonlight het probleem niet verholpen worden, dan zet de monteur de Andonlight op rood. Behalve dat de lamp van de fysieke Andonlight op rood gaat wordt er ook een bericht van de *client* van de monteur doorgegeven aan de *dashboardserver*. De *dashboardserver* geeft op grote schermen op de ondersteunende afdelingen weer dat de monteur een probleem heeft. Vervolgens kan een medewerker van de ondersteunende afdeling naar de Shop afdeling lopen, om de monteur te helpen. Nadat het probleem behandeld is, wordt de Andonlight weer op groen gezet.

Vanuit de Shop en ondersteunende afdelingen komen verschillende meldingen van medewerkers dat het huidige systeem niet optimaal functioneert. Het systeem is niet altijd betrouwbaar. Er moet onderzocht worden hoe de gebruiksvriendelijkheid van het systeem verbeterd kan worden, voor de *key-users*<sup>3</sup>.

Ook houdt het huidige systeem geen bruikbare data vast. Hierdoor is er niets bekend over het gebruik van het huidige systeem. Er is bijvoorbeeld niet bekend of bepaalde problemen veel voorkomen of dat bepaalde problemen veel tijd kosten of informatie over het gebruik per monteur of per afdeling.

Met behulp van het bovengenoemde is de volgende hoofdvraag opgesteld:

“Hoe kan bij Epcor B.V. door het vernieuwen van het Andonlight systeem de gebruiksvriendelijkheid<sup>4</sup> verbeterd worden voor de *key-users* aan de hand van hun voorkeuren en welke data moet het systeem opslaan zodat deze geanalyseerd kan worden, zodat op de bevindingen die hieruit voortkomen geanticipeerd kan worden?”

---

<sup>3</sup> Onder de *key-users* vallen de medewerkers van het primaire proces en de medewerkers van de ondersteunende processen, die direct gebruik maken van het systeem, maar ook de medewerkers die van de opgeslagen gegevens gebruik gaan maken.

<sup>4</sup> Alles waardoor de gebruiker van het systeem zijn werk beter kan uitvoeren, waarbij rekening wordt gehouden met de principes van Lean Six Sigma

## 2 Doelstelling

De doelstelling bestaat uit de volgende twee punten:

- De mogelijkheid creëren om de data op te slaan en terug te zien, zodat historische data beschikbaar wordt en hierop geanticipeerd kan worden.
- Verhogen van de gebruiksvriendelijkheid voor *key-users* aan de hand van hun voorkeuren.

In de analysefase zijn de volgende producten conform het Plan van Aanpak<sup>5</sup> opgesteld:

- Systeemmodel
- Business Process Model
- Nulmeting
- Boomschema's
- Eisen

Het huidige systeem is in kaart gebracht met behulp van een systeemmodel. De voor het huidige systeem relevante processen zijn in kaart gebracht met behulp van een Business Process Model. Vergelijkingsmateriaal is vastgelegd in de vorm van een nulmeting. De informatie die nodig is over de Andonlight problemen en oplossingen is vastgelegd in boomschema's. Tot slot zijn de eisen van het nieuwe systeem vastgelegd. De analyse is te vinden in hoofdstuk 3 'Analyse'.

Met behulp van de informatie uit de analysefase is de doelstelling uitgewerkt in de vorm van een ontwerp voor een nieuw Andonlight systeem.

Voor het ontwerp zijn in hoofdstuk 4 'Ontwerpresultaten' de volgende producten conform het Plan van Aanpak opgesteld:

- Use Case model
- Use Case specificaties
- Database model
- Klassendiagrammen
- Sequence diagrammen
- Software Architectuur

In de Use Cases wordt stilgestaan bij de functionaliteiten van het nieuwe systeem. In het database model wordt een ontwerp van de tabellen voor de benodigde data weergegeven. In de klassendiagrammen wordt de samenhang tussen de verschillende klassen per Use Case in beeld gebracht. Ook worden in de klassendiagrammen de methodes en variabelen van de klassen weergegeven. In de *sequence* diagrammen worden de methode aanroepen tussen de klassen per Use Case weergegeven. Met behulp van RUP 4+1 view model wordt gekeken naar de software architectuur.

Tot slot wordt er stilgestaan bij de implementatie van het nieuwe systeem in hoofdstuk 5 'Implementatie'.

---

<sup>5</sup> Het Plan van Aanpak is te vinden in Bijlage C: Plan van Aanpak

### 3 Analyse

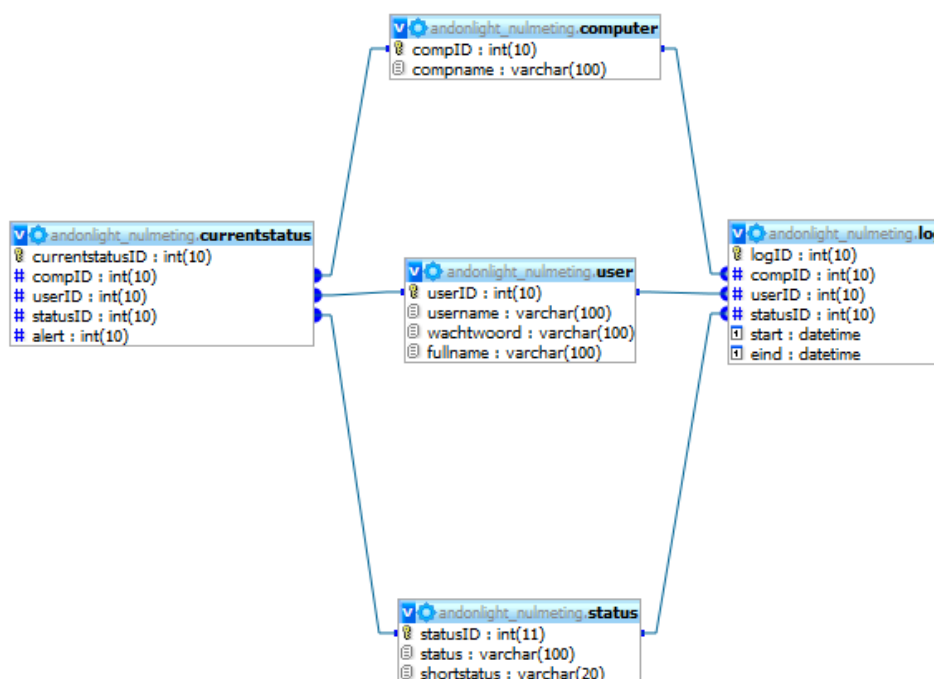
In de analyse fase zoals deze beschreven staat in het Plan van Aanpak zijn de volgende producten opgeleverd:

- Systeemmodel
- Business Proces Model
- Nulmeting
- Boomschema's
- Eisen

In het systeemmodel is de werking van het huidige systeem in kaart gebracht. In het BPM zijn de processen die betrekking hebben op het huidige Systeem in kaart gebracht. Tijdens de nulmeting is data vastgehouden over het gebruik van het huidige systeem. Dit kan dienen als vergelijkingsmateriaal voor het nieuwe systeem, nadat deze in gebruik is genomen. In de boomschema's is per ondersteunende afdeling vastgelegd welke informatie opgeslagen moet worden over Andonlight problemen (de redenen waarom de Andonlight op rood wordt gezet) en de bijbehorende oplossingen. De eisen zijn gegroepeerd in functionele- en niet-functionele eisen. Hierover meer in de paragraaf 'Eisen'.

#### 3.1 Systeemmodel

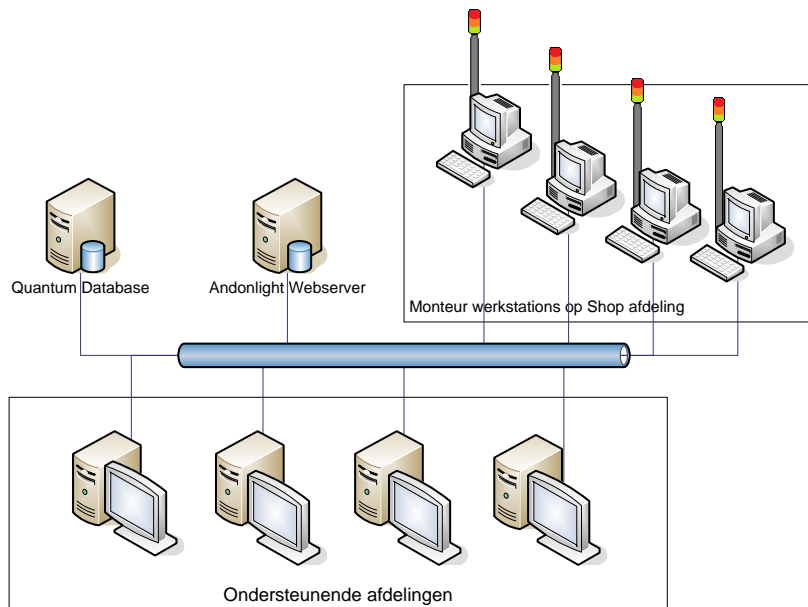
Het systeemmodel is gemaakt door de code en de database van het huidige systeem te bestuderen en door de ontwikkelaar van het systeem interviewen. In deze paragraaf wordt gekeken naar de werking en de gebreken van het huidige systeem.



Figuur 3.1 Database structuur



In figuur 3.1 is de database structuur van het huidige systeem te zien. In het verleden sloeg het systeem permanente *logdata* op, maar deze functionaliteit is later verwijderd. De *log* tabel wordt op dit moment niet door het systeem gebruikt. In de *currentstatus* tabel wordt de huidige status van alle monteurs tijdelijk vastgehouden. De *currentstatus* tabel heeft refererende sleutels naar de *user*, computer en status tabel. In de computer tabel staan de werkstations van de monteurs. In de *user* tabel staan de monteursgegevens en in de status tabel staan de mogelijk statussen. De mogelijke statussen bestaan uit de ondersteunende afdelingen als de Andonlight op rood wordt gezet en *supervisor* als de Andonlight op oranje wordt gezet.



**Figuur 3.2 Infrastructuur**

De infrastructuur is te zien in figuur 3.2. Op het workstation van de monteur draait een *client*. De *client* heeft een ODBC koppeling naar een MSSQL database. De Andonlight webserver host een dashboard pagina waarop de gegevens van de monteurs te zien is. Computers op de ondersteunende afdelingen geven de dashboard pagina op grote schermen weer. De Andonlight webserver haalt periodiek met een ODBC koppeling statusgegevens (die de *clients* in de MSSQL database hebben opgeslagen) op. Met behulp van deze gegevens haalt de webserver met een ODBC koppeling het 'workordernummer' en de taak uit de Quantum Controle database op. In de *workorder* staan de werkinstructies die de monteur heeft gekregen, voor een bepaald component. Een taak is onderdeel van een werkinstructie, bijvoorbeeld *cleaning*.



**Figuur 3.3 Monteur client**

In figuur 3.3 is de *client* voor de monteurs te zien. Het is een *stand-alone* applicatie geschreven in Visual Basic. VB is niet object georiënteerd. Alle code staat door elkaar. Soms worden er functies gebruikt. De verbindingsgegevens voor de database staan in de code.

De *client* logt de computer ID (op basis van de Windows computernaam), de gebruiker (op basis van de Windows gebruiker) en de status van de Andonlight in de MSSQL database.

Als het werkstation van de monteur niet goed wordt afgesloten of crasht, blijft de fysieke Andonlight aan staan.



Figuur 3.4 Server dashboardscherm

In figuur 3.4 is het dashboardscherm van de *server* te zien. Op het dashboardscherm zijn verschillende subafdeling van de Shop afdeling te zien, bijvoorbeeld 'Test'. Verder zijn de werkplekken van de monteurs te zien. Deze worden weergegeven als rechthoeken. Per werkplek wordt van de monteur die aan het werk is status, naam, workordernummer en taak weergegeven. De status wordt aangegeven door de kleur van de rechthoek.

De gebruikte programmeertaal voor de *server* is: Active Server Pages. ASP (niet .NET) is niet object georiënteerd. Er wordt geen gebruik gemaakt van klassen of functies. HTML (HyperText Markup Language) en ASP code staat door elkaar heen. De gebruikersnamen en wachtwoorden voor de database verbindingen staan in de code.

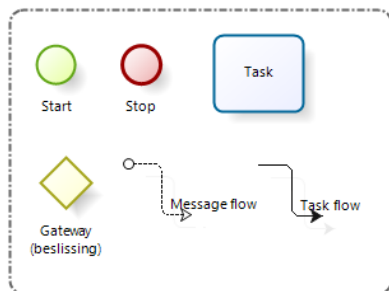
De locatie van de verschillende werkstations op de plattegrond is gebonden aan de computer ID. De *server* vergelijkt de door de *client* gelogde computer en gebruiker met twee tabellen: computer en *user*. De code van de *server* gaat er bij iedere computer in de computer tabel vanuit dat deze een vaste locatie heeft. De computer wordt op die locatie weergegeven op de plattegrond. Dit betekent dat als er een workstation bij komt of wordt verplaatst dat deze niet op de verkeerde locatie wordt weergegeven. Als er een nieuwe monteur bij komt staat zijn of haar naam niet in de *user* tabel, dus bij een rode Andonlight wordt de naam van deze monteur niet op het dashboard weergegeven.

De huidige webserver laat een maximum van 5 TCP verbindingen toe. Dit komt omdat deze eigenlijk alleen voor testdoeleinden bedoeld is. Het komt regelmatig voor dat het dashboardscherm bij een ondersteunende afdeling niet weergegeven wordt door de *webserver*, omdat medewerkers de pagina ook op hun eigen workstation open hebben staan, waardoor met maximum aantal verbindingen overschreden wordt.

Als het werkstation van de monteur crasht terwijl de Andonlight op rood staat blijft de *server* op het dashboard weergegeven dat de medewerker van die werkplek een probleem heeft, totdat het werkstation opnieuw wordt opgestart en de status van de Andonlight wordt gewijzigd.

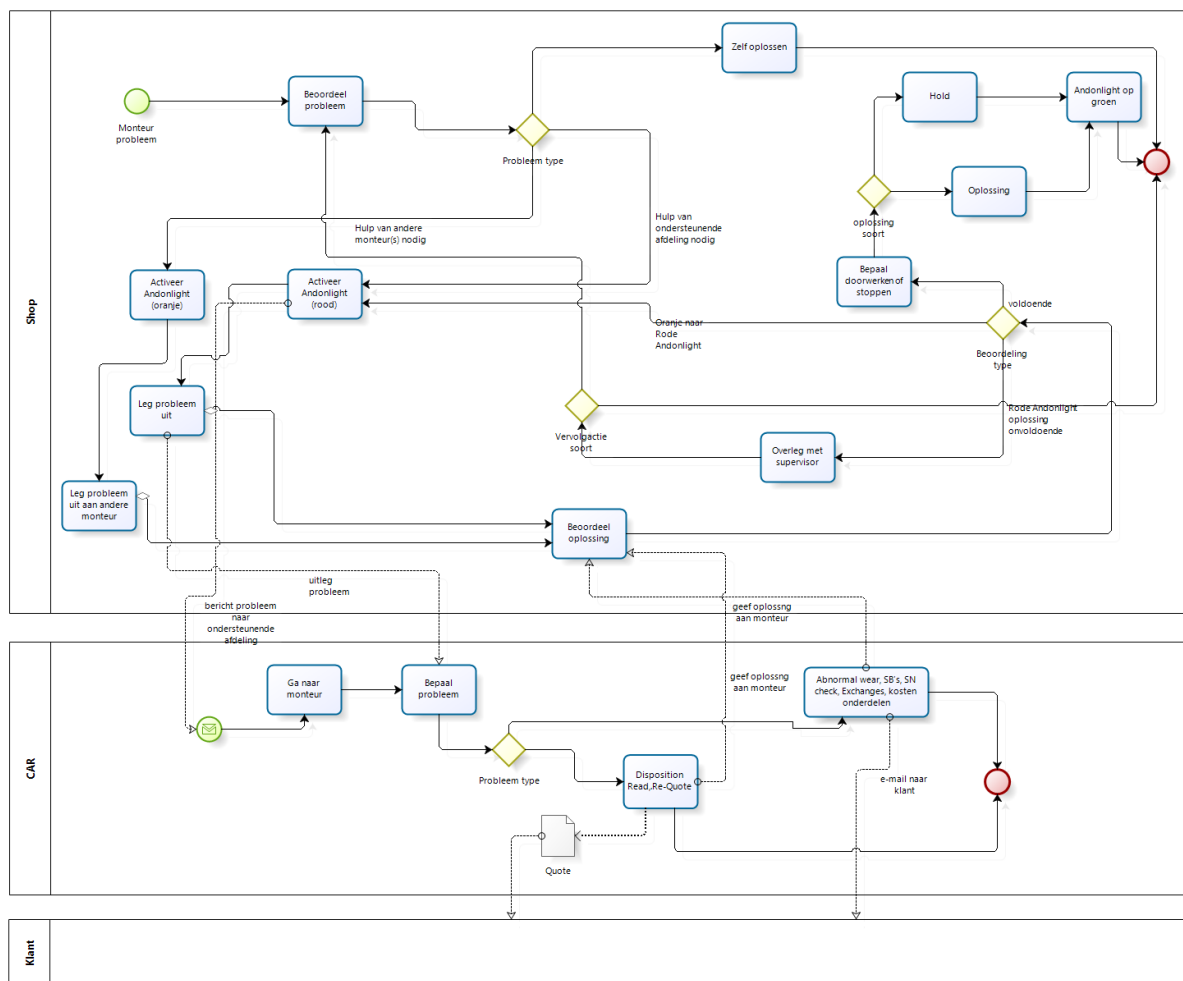
## 3.2 Business Proces Model

Om de huidige bedrijfsprocessen die betrekking hebben op het Andonlight systeem in kaart te brengen wordt gebruik gemaakt van een zogenaamd Business Process Model. Voor een BPM gelden de regels van de Business Process Modeling Notation. In figuur 3.5 worden de belangrijkste componenten weergegeven.



**Figuur 3.5 BPM legenda**

De BPM's zijn gemaakt door op iedere ondersteunende afdeling een dag mee te lopen en op de Shop afdeling twee dagen. De processen die betrekking hebben op het huidige Systeem zijn in kaart gebracht omdat deze als uitgangspunt genomen moeten worden voor het vaststellen van de functionaliteiten van het nieuwe systeem. De processen zijn ontwikkeld volgens het Lean Six Sigma concept en kunnen tijdens dit project niet veranderd worden.



**Figuur 3.6 BPM Shop en CAR afdeling**

In figuur 3.6 is de BPM van de CAR afdeling te zien. Deze afdeling verzorgt de communicatie met de klant. Het kan voorkomen dat een monteur pas verder kan werken als de klant akkoord gaat met een bepaalde handeling. Voorbeelden zijn: als er bepaalde onderdelen gebruikt moeten worden die extra kosten met zich mee brengen, *abnormal wear* (abnormale slijtage), als een serie nummer gecontroleerd moeten worden, als er een offerte opgesteld moet worden, als een component (hiermee wordt bedoeld een volledig onderdeel, bijvoorbeeld een klep of een APU) omgewisseld kan worden voor een gelijkwaardige component of modificaties aan een component.

Over het algemeen verloopt de communicatie met de klant via e-mail. Nadat de CAR afdeling heeft besloten wat er moet gebeuren, wordt aan de monteur gemeld of het component in de kast geplaatst moet worden of dat hij/zij kan doorwerken. (Als de monteur de component in de kast moet zetten krijgt hij/zij van zijn/haar supervisor een andere component om aan te werken.)









structuur van de database te zien. In de *currentstatus* tabellen wordt tijdelijk informatie opgeslagen die nodig is voor het functioneren van het systeem. Voor de nulmeting zit hier geen nuttige informatie in. In de *user* tabel staan de gegevens van de gebruiker. In dit geval is dit de monteur. In de computer tabel staan de gegevens van de werkstations van de monteurs. In de *status* tabel staan de gegevens van de verschillende afdelingen of *supervisor* waaraan assistentie gevraagd kan worden door de monteur. Als een Andonlight op oranje of rood en vervolgens weer op groen wordt gezet, werd dit gelogd in de *log* tabel. In deze tabel wordt met behulp van refererende sleutels vastgelegd welke monteur de Andonlight op rood heeft gezet, welk werkstation de monteur gebruikt, van welke afdeling assistentie nodig is en de begin- en eindtijd.

Deze data kan gebruikt worden om het huidige en het nieuwe systeem te vergelijken. De volgende gegevens kunnen bijvoorbeeld vergeleken worden: aantal Andonlights per afdeling/*supervisor*, per monteur, per tijdseenheid en combinaties hiervan. Naast aantallen Andonlights die zijn aangezet kunnen uit de database ook allerlei gegevens gehaald worden over de periode waarover Andonlights aan hebben gestaan.

### 3.3.2 Nulmeting data van testserver

Door ASP code toe te voegen aan dashboard pagina is er over de maand mei van 2012 data naar een bestand gelogd. De volgende gegevens zijn gelogd als een Andonlight op oranje of rood is gezet:

- Monteur gegevens
- Begin- en eindtijd
- De *workorder*
- De specifieke taak die de monteur aan het uitvoeren is

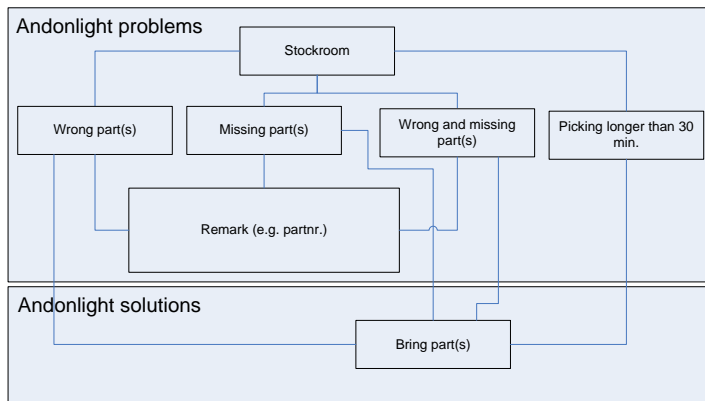
Deze data kan op dezelfde manier gebruikt worden om te vergelijken als de data uit de oude Andonlight database, alleen is er nu per Andonlight die is aangezet ook bekend welke *workorder* de monteur had en welke taak werd uitgevoerd toen de Andonlight werd aangezet.

## 3.4 Boomschema's

In de (omgekeerde) boomschema's is per ondersteunende afdeling vastgelegd welke informatie opgeslagen moet worden over Andonlight problemen (de redenen waarom de Andonlight op rood wordt gezet) en de bijbehorende oplossingen. De boomschema's zijn hiërarchisch opgebouwd, met bovenaan de afdeling en hieronder de probleem categorieën. Iedere categorie kan weer verder uitgesplitst zijn in subcategorieën. De structuur van de oplossing is hetzelfde.

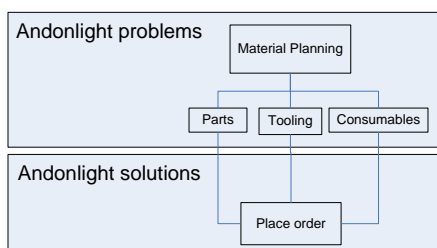
De oplossingsbomen zijn gekoppeld aan hoofdprobleemcategorieën. Stel namelijk dat de oplossingsbomen aan meeste uitgesplitste categorieën in de probleembomen gekoppeld zouden worden, dan zou per extra categorie op dit niveau een extra oplossingsboom gemaakt moeten worden. Dit loopt al vrij snel uit de hand.

De data is verkregen door gesprekken te voeren met medewerkers van de ondersteunende afdelingen.



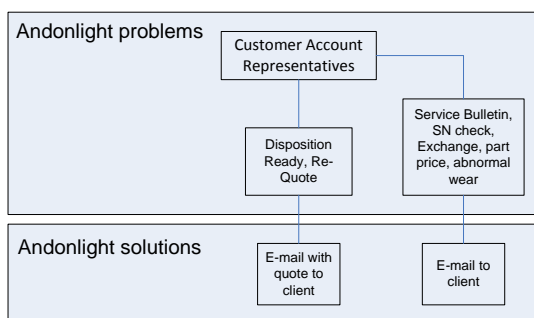
**Figuur 3.10 Boomschema Magazijn**

In figuur 3.10 is het boomschema voor de Magazijn (*stockroom*) afdeling te zien. Alle problemen hebben te maken met onderdelen. Er kan een fout zijn gemaakt bij het picken, waarbij een onderdeel vergeten en/of verkeerd is. Als het picken langer duurt dan 30 minuten is dit ook een reden om de Andonlight op rood te zetten voor de Magazijn afdeling. De oplossing is altijd dat een medewerker van de Magazijn afdeling één of meerdere onderdelen komt brengen.



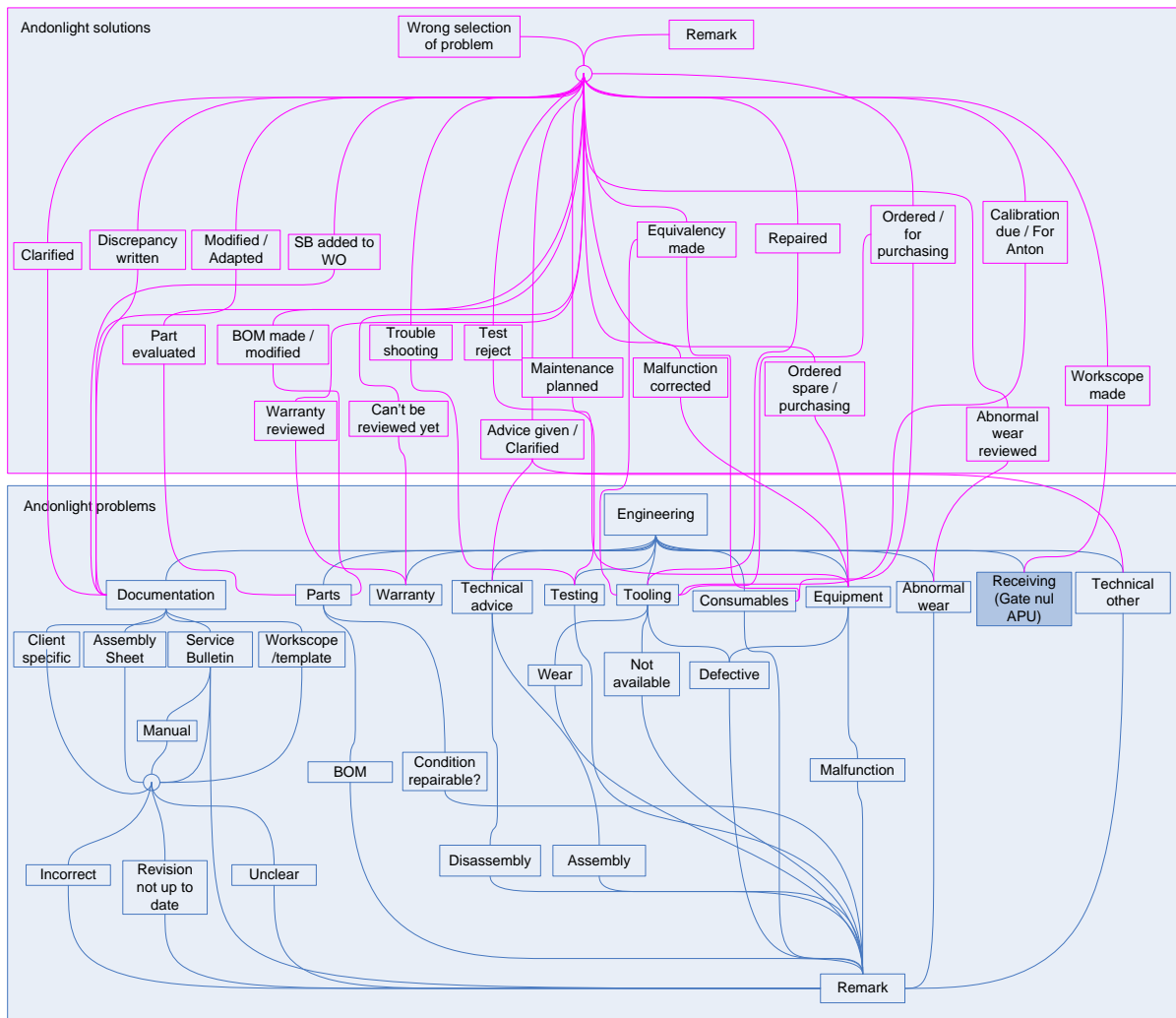
**Figuur 3.11 Boomschema Materiaal Planning**

In figuur 3.11 is het boomschema voor de Materiaal Planningsafdeling te zien. Bij de Materiaal Planningsafdeling kan het voorkomen dat er een onderdeel, stuk gereedschap of verbruiksartikel besteld moet worden. Ik kan gebeuren door de Andonlight voor deze afdelingen op rood te zetten. Als oplossing kan er een bestelling gedaan worden.



**Figuur 3.12 Boomschema Customer Account Representatives**

In figuur 3.12 is het boomschema voor de CAR afdeling te zien. Het kan voorkomen dat met de klant overlegd moet worden voordat de monteur verder kan met de werkzaamheden. In dit geval zet de monteur de Andonlight op rood voor de CAR afdeling. De oplossing is altijd dat de CAR afdeling een e-mail naar de klant zendt. De CAR afdeling beoordeelt ook of de monteur verder mag werken of dat het component waar aan gewerkt wordt in de kast gezet moet worden.



**Figuur 3.13 Boomschema Engineering**

In figuur 3.13 is het boomschema voor de Engineering afdeling te zien. (Voor de leesbaarheid hebben de verbindingslijnen voor de oplossingen een andere kleur gekregen.) De monteurs raadplegen de Engineering afdeling als er technisch advies nodig is.

Om alle problemen en oplossingen voor deze afdeling uit te leggen is in sommige gevallen veel vaktechnische kennis nodig. Daarom zullen alleen de hoofdprobleemcategorieën toegelicht worden. Belangrijk is dat de juiste categorieën en de juiste hiërarchie is gekozen. *Documentation*: alle problemen die betrekking hebben op documenten. *Parts*: problemen die betrekking hebben op de staat van een component. *Warranty*: problemen die betrekking hebben op garantie van een component. *Technical advice*: problemen bij het in elkaar zetten of uit elkaar halen van een component. *Test*: problemen die ontstaan bij het testen van een component. *Tooling*: problemen met gereedschap. *Consumables*: problemen met verbruiksartikelen. *Equipment*: problemen met apparatuur. *Abnormal wear*: abnormale slijtage van een component. *Receiving*: het in ontvangst nemen van een APU. *Technical other*: overige problemen.

## 3.5 Eisen

Door gesprekken met de *key-users* zijn de functionele- en niet-functionele eisen van het nieuwe systeem vastgesteld. Bij iedere ondersteunende afdeling en de Shop afdeling is een dag meegelopen. Ook is een 'brainstorm sessie' gehouden met de *key-users*. De tweedelige *brainstorm* sessie bestond uit: een presentatie over het project inrichting Andonlight systeem, het in kaart brengen van de processen die betrekking hebben op het Andonlight systeem en de eisen van het nieuwe Andonlight systeem.

Onder functionele eisen wordt verstaan de functionaliteiten die het nieuwe systeem moet bezitten. Deze worden uitgewerkt in Use Cases. De niet-functionele eisen zijn alle eisen die geen functionele eis zijn.

### 3.5.1 Functionele eisen

De functionele eisen bestaan uit de functionaliteiten van het huidige systeem en de volgende extra functionaliteiten:

- De mogelijkheid om de data op te slaan en terug te zien, zodat historische data beschikbaar wordt en hierop geanticipeerd kan worden.
- De probleemdefinitie van de monteur moet op het dashboardscherm weergegeven worden, zodat deze bij de ondersteunende afdeling bekend is
- Op het dashboardscherm moet weergegeven worden hoe lang de Andonlight al op rood staat
- Het dashboardscherm moet beschikken over een tabel weergave en een plattegrond weergave, waarin de monteurs met een probleem te zien zijn

### 3.5.2 Niet-functionele eisen

De volgende niet-functionele eisen zijn vastgesteld:

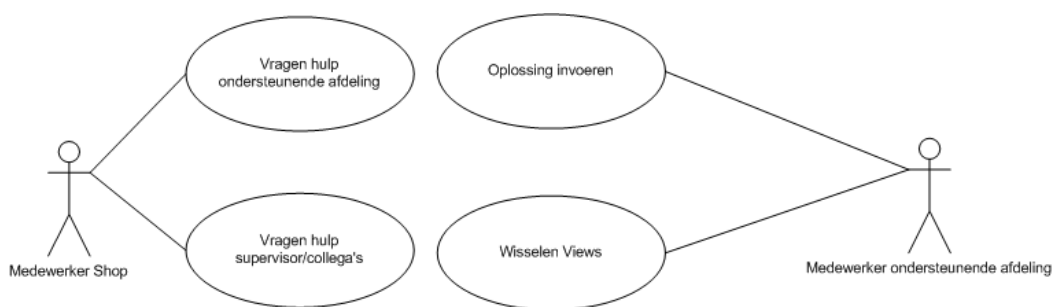
- Gui gebaseerd
- Boomschema's aanpasbaar
- Schaalbaar
- Betrouwbaar
- Onderhoudsarm
- Het GUI van het systeem dient in het Engels te zijn

## 4 Ontwerpresultaten

In de ontwerpfase van het project inrichting Andonlight systeem zijn een aantal producten opgeleverd. In paragraaf 4.1 'Use Cases' zijn de functionaliteiten van het nieuwe systeem uitgewerkt. In paragraaf 4.2 'Database model' wordt het ontwerp van de database besproken. In paragraaf 4.3 'Klasse diagrammen' wordt de samenhang tussen de klassen die per Use Case nodig zijn besproken. In paragraaf 4.4 'Sequence diagrammen' wordt per Use Case besproken welke functie aanroepen nodig zijn om de Use Case mogelijk te maken. (Per klasse is ook aangegeven bij welke deel van de 'Model View Presenter' *design pattern* deze behoort.) In paragraaf 4.5 'Software Architectuur' wordt met behulp van de 4+1 RUP view model gekeken naar de software architectuur.

### 4.1 Use Cases

Met behulp van Use Cases worden de interacties met het systeem beschreven.



Figuur 4.1 Use Case model

In figuur 4.1 zijn de Use Cases te zien die zijn vastgesteld bij het bepalen van de functionele eisen van het nieuwe Systeem.

Tabel 4.1 Omschrijving Use Cases

Code	Use Case Naam	Omschrijving	Gewicht	Prioritering
UC001	Vragen hulp ondersteunende afdeling	In de <i>client</i> voert de medewerker van de Shop afdeling het probleem in. Vervolgens wordt de fysieke Andonlight op rood gezet. Het probleem wordt weergegeven op het dashboardscherm bij de ondersteunende afdelingen. De ondersteunende medewerker komt naar de medewerker van de Shop afdeling toe en behandelt het probleem. Hierna kan de Andonlight weer op groen gezet worden. Nadat dit gebeurd is worden de probleem gegevens en begintijd, eindtijd, Shop medewerkernr. en workordernr. naar de database geschreven.	3	M
UC002	Vragen hulp supervisor/collega	Deze use case is hetzelfde als UC001 alleen worden er geen probleem gegevens opgeslagen in de database. Inplaatst van rood wordt de Andonlight.	3	M

Code	Use Case Naam	Omschrijving	Gewicht	Prioritering
		op oranje gezet. Daarom zal deze Use Case niet apart gespecificeerd worden.		
UC003	Oplossing invoeren	De medewerker van de ondersteunende afdeling zoekt het opgeloste probleem op in Systeem en voert hiervoor de oplossing in. De oplossing wordt aan het probleem gekoppeld.	3	M
UC004	Wisselen view dashboard	De medewerker van de ondersteunende afdeling klikt op knop op pagina dashboard, om de view te wisselen. Hierdoor wordt de view van de dashboard pagina aangepast.	1	C

In tabel 4.1 is de omschrijving van de Use Cases te zien. Ook hebben de Use Cases een gewicht gekregen variërend 1 tot en met 3 op basis van de complexiteit en hebben ze een prioriteit gekregen. 'Must Have': deze Use Case is onmisbaar voor de bruikbaarheid van het Systeem. 'Could Have': de Use Case heeft een duidelijke toegevoegde waarde, maar zonder is er nog steeds een bruikbaar Systeem.

**Tabel 4.2 Actoren**

Code	Actor	Omschrijving
A001	Medewerker ondersteunende afdeling	Medewerker op de ondersteunende afdeling. De ondersteunende afdelingen zijn: Sales & Customer Services, Materiaal planning, Engineering en Magazijn
A002	Medewerker Shop	Monteur in het primaire proces op de Shop afdeling

In tabel 4.2 is een omschrijving van de actoren die handelingen op het systeem uitvoeren te zien.

In de volgende paragrafen zullen de Use Cases in detail worden besproken in de vorm van Use Case specificaties.

#### 4.1.1 Use Case specificatie UC001 'Hulp vragen bij ondersteunende afdeling'

**Tabel 4.3 Kenmerken**

Kenmerk	Omschrijving
<i>Aanleiding (Trigger)</i>	Shop medewerker heeft een probleem die hij/zij niet kan oplossen en ook de supervisor en collega medewerkers kunnen het niet oplossen.
<i>Actors</i>	Medewerker Shop
<i>Wijze van uitvoering</i>	interactief
<i>Samenhang met andere Use Cases</i>	Nadat deze use case is uitgevoerd die de use case 'Oplossing invoeren' uitgevoerd te worden, door de medewerker van de ondersteunende afdeling die het probleem heeft opgelost. Deze use case wordt meestal uitgevoerd na de use case 'Hulp vragen bij supervisor/collega's'.
<i>Frequentie van uitvoering</i>	Tussen de 10 en de 60 x per werkdag

In tabel 4.3 worden de kenmerken van deze Use Case specificatie weergegeven.

Tabel 4.4 Basisscenario — UC001 Hulp vragen bij ondersteunende afdeling

Actor	Medewerker Shop
Preconditie	Andonlight <i>client</i> voor Shop medewerker is gestart.
Scenario beschrijving	<ol style="list-style-type: none"> <li>1. De Medewerker klikt op het <i>dropdown menu</i>, om de afdeling waarvan hulp nodig is te selecteren.</li> <li>2. De <i>client</i> vraagt om de lijst met afdelingen aan de <i>server</i>. (De <i>server</i> heeft bij het opstarten deze gegevens geladen uit een .xml bestand.)</li> <li>3. De <i>server</i> stuurt bij ontvangst van het bericht de lijst.</li> <li>4. De medewerker selecteert de afdeling.</li> <li>5. De <i>client</i> stuurt de afdelingselectie naar de <i>server</i>.</li> <li>6. De <i>client</i> geeft een nieuw <i>drop-down menu</i> weer, waarin het hoofdprobleemtype selecteert kan worden.</li> <li>7. De <i>server</i> onthoudt bij ontvangst de afdelingselectie bij deze gebruiker.</li> <li>8. De medewerker klikt op het <i>drop-down menu</i> voor het probleemtype.</li> <li>9. De <i>client</i> vraagt om de lijst van probleemtypen ( eventueel behorende bij het eerder geselecteerde probleemtype).</li> <li>10. De <i>server</i> stuurt bij ontvangst een lijst met probleemtypen.</li> <li>11. De medewerker selecteert in het <i>drop-down menu</i> het probleemtype.</li> <li>12. De <i>client</i> stuurt de selectie naar de <i>server</i>.</li> <li>13. De <i>client</i> geeft een nieuw <i>drop-down menu</i> weer, waarin het probleemtype selecteert kan worden.</li> <li>14. De <i>server</i> onthoudt bij ontvangst de selectie bij deze gebruiker.</li> <li>15. Stappen 8 t/m 14 herhalen zich tot het probleem (door selecties van probleemtypen uit een omgekeerd boomschema) helemaal definieert is. In sommige gevallen kan bij het einde van de probleemdefinitie een opmerking ingevoerd worden, bijvoorbeeld een onderdeelnummer naar de magazijn afdeling. Dit is afhankelijk van het soort probleem. Voor de gebruiker is het duidelijk dat het probleem volledig is gedefinieerd als er geen nieuw <i>drop-down menu</i> verschijnt na de laatste selectie of als er een opmerking venster wordt getoond.</li> <li>16. De medewerker klikt op de knop om de Andonlight op rood te zetten.</li> <li>17. De <i>client</i> zet de fysieke Andonlight op rood. (Periodiek zal de <i>client</i> nu een bericht naar de <i>server</i> sturen dat de Andonlight nog steeds op rood staat en de <i>server</i> bevestigt de ontvangst met een bericht terug. Dit is nodig om te zorgen dat <i>server</i> in <i>sync</i> is met de <i>client</i>.)</li> <li>18. De <i>client</i> stuurt een bericht aan de <i>server</i> dat de Andonlight op rood is gezet. Als er aan het einde een opmerking is ingevoerd, wordt deze ook meegestuurd.</li> <li>19. De <i>server</i> bewaart alle eerder ontvangen gegevens in het geheugen. Ook wordt de starttijd in het geheugen bewaard.</li> <li>20. De <i>server</i> geeft op het dashboardscherm weer dat de betreffende Shop medewerker hulp nodig heeft van de aangegeven afdeling. Ook wordt de probleemdefinitie op het dashboard weergegeven.</li> <li>21. De medewerker klikt op de knop om de Andonlight weer op groen te zetten, (Nadat het probleem is opgelost.)</li> <li>22. De <i>client</i> zet de fysieke Andonlight op groen.</li> <li>23. De <i>client</i> stuurt een bericht aan de <i>server</i> dat de Andonlight op groen is gezet.</li> <li>24. De <i>server</i> schrijft de eerder ontvangen gegevens (die nog steeds in het geheugen staan) en de eindtijd naar de database. (Eén volledige transactie in nu afgerond.)</li> <li>25. De <i>server</i> geeft op het dashboard het probleem niet meer weer.</li> </ol> <p>*Opmerking 1: doordat de <i>server</i> de boomschema's bij het opstarten uit een .xml bestand laad, wordt voldaan aan de eis van aanpasbare boomschema. Het .xml kan namelijk eenvoudig aangepast worden.</p> <p>*Opmerking 2: zoals in stap 17. is besproken wordt er altijd voor gezorgd dat de <i>server</i> in <i>sync</i> is met de <i>client</i>. Dit draagt bij aan de eis van een betrouwbaar systeem. Zoals in stap 24. te lezen is wordt de data pas naar de database beschreven al een volledige transactie is uitgevoerd. Dit draagt ook bij aan de betrouwbaarheid.</p>
Postconditie	De Andonlight staat op groen.

In tabel 4.4 wordt stapsgewijs weergegeven hoe deze Use Case wordt uitgevoerd.

Figuur 4.2 Schermontwerp van scherm 'Andonlight client shop'

In Figuur 4.2 is het schermontwerp te zien van de *client* deel van het systeem voor de monteurs. In de *client* wordt het probleem gedefinieerd, kan als nodig een opmerking toegevoerd worden en kan de Andonlight bediend worden. De *client* draait op het werkstation van de monteur op de Shop afdeling.

Mechanic	Department	Workorder	Task
Bert de Vries	PL	12341234	cleaning
Bart Heuvel	ADIS	43214321	disassembly
Alex de Graaf	OEM	88884444	assembly

Timer	Problem definition	Remark
00:06	documentation → manual → unclear	omschr. van onduidelijkheid
00:13	parts	partomschrijving of nr.
00:23	missing part(s)	partnr(s)

Figuur 4.3 Schermontwerp van scherm 'Andonlight dashboard'

In Figuur 4.3 is het dashboardscherm te zien. Dit scherm wordt door het *server* deel van het systeem weergegeven op grote beeldschermen op de ondersteunende afdelingen. Per monteur wordt de voor- en



achternaam, subafdeling waar de monteur zich bevindt, workordernummer en taak weergegeven. Van de Andonlight wordt de tijd dat deze aan staat, de probleemdefinitie en de eventuele opmerking weergegeven. De ondersteunende afdeling waarvan de monteur hulp nodig heeft wordt weergegeven door de kleur van de regel. Hierdoor kan van grote afstand door medewerkers van de ondersteunende afdeling gezien worden of een monteur hulp nodig heeft.

**Tabel 4.5 Legenda bij de veldbeschrijvingstabellen**

Label	Naam van het veld zoals dit op het scherm zichtbaar is
Verplicht	Ja, nee of n.v.t. (als een veld een output veld is, is verplicht niet van toepassing)
Status	I(nput) veld, O(utput) veld of M(uteerbaar) veld. Input, d.w.z. door de gebruiker in te vullen. Output, d.w.z. door het systeem gevuld en niet door de gebruiker te wijzigen. Muteerbaar, d.w.z. initieel door het systeem gevuld en door de gebruiker te wijzigen. Bij de waarde M moet de kolom Validatie minimaal gevuld zijn met de initiële waarde.
Formaat / lengte	Type veld (pulldown, select, radio, numeric, text, date etc.) en lengte: zowel de getoonde lengte als de door het systeem geaccepteerde maximale lengte. Bijv. een straatnaam veld kan 30 posities op het scherm weergeven maar 50 posities accepteren. Het betreffende invulvak gaat dan 'scrollen.' Het eerste getal geeft de input lengte aan. Als er één getal wordt weergegeven betekent dit dat input lengte en door het systeem geaccepteerde lengte samenvallen.
Validatie	Alle validaties / business rules die op het betreffende veld van toepassing zijn (waardebereik, formaat, verhouding tot andere velden, wijzigbaarheid, tabvolgordes etc.). Ook sorteringsspecificaties kunnen hier een plaats krijgen, evenals initiële vulling en default waarden.
Toelichting	Verhelderende informatie die niet in een van de andere kolommen past.

In tabel 4.5 is een legenda te zien waarin de kolommen van Tabel 6.6 (en alle andere veldbeschrijvingstabellen) wordt uitgelegd.

**Tabel 4.6 Beschrijving van velden**

Label	Verplicht	Status	Formaat / lengte	Validatie	Toelichting
Department/supervisor	J	I	Text(20,30)	Dropdown menu	
Problem type	J	I	Text(20,30)	Dropdown menu	
Remark	J	I	Text(100,300)	Alle tekens die in tekst gebruikt kunnen worden.	

In tabel 4.6 worden de velden van het scherm 'Andonlight client shop' besproken.

**Tabel 4.7 Foutscenario 1 — Vragen hulp ondersteunende afdeling transacties niet volledig afgerond**

Actor	Hetzelfde als bij UC001 Hulp vragen ondersteunende afdeling.
Preconditie	Tot en met stap 16 van UC001 Hulp vragen ondersteunende afdeling.
Scenario beschrijving	<ol style="list-style-type: none"> <li>1. De <i>server</i> ontvangt geen statusbericht meer van de client, nadat de Andonlight op rood is gezet.</li> <li>2. De <i>client</i> ontvangt geen bevestigingsbericht meer van de server en zal na een vastgestelde periode aan de gebruiker melden dat de verbinding weggevallen is.</li> <li>3. De <i>server</i> zet de Andonlight na een vastgestelde periode op groen en schrijft de transactiegegevens naar de database.</li> </ol>
Postconditie	Hetzelfde als bij Basisscenario — UC001 Hulp vragen bij ondersteunende afdeling.

In tabel 4.7 wordt weergegeven hoe het systeem handelt als nadat de Andonlight op rood is gezet de verbinding verbroken wordt. In deze situatie is de transactie niet volledig. Bij deze transactie wordt daarom in de *Problem* tabel de waarde van *incomplete* op *true* gezet.

#### 4.1.2 Use Case specificatie UC003 'Oplossing invoeren'

Tabel 4.8 Kenmerken

Kenmerk	Omschrijving
<i>Aanleiding (Trigger)</i>	Rode Andonlight is opgelost en medewerker van de ondersteunende afdeling is weer terug bij zijn/haar werkplek. Nu dient de oplossing ingevoerd te worden.
<i>Actors</i>	Medewerker ondersteunende afdeling.
<i>Wijze van uitvoering</i>	Interactief.
<i>Samenhang met andere Use Cases</i>	Wordt uitgevoerd nadat een rode Andonlight is opgelost.
<i>Frequentie van uitvoering</i>	10-60 x per werkdag.

Tabel 4.9 Basisscenario — Oplossing invoeren

Actor	Medewerker ondersteunende afdeling.
Preconditie	Andonlight client voor ondersteunende afdeling is gestart.
Scenario beschrijving	<ol style="list-style-type: none"> <li>De <i>client</i> toont een lijst met Andonlight problemen waar nog geen oplossing aan gekoppeld is.</li> <li>De medewerker selecteert het juiste Andonlight probleem, door op één van de regels te klikken.</li> <li>De <i>client</i> stuurt de probleemselectie naar de <i>server</i>.</li> <li>De <i>server</i> onthoudt bij ontvangst de selectie.</li> <li>De <i>client</i> geeft een <i>drop-down menu</i> weer waarin het oplossingstype geselecteerd kan worden.</li> <li>De medewerker klikt op het <i>drop-down menu</i> voor het oplossingstype.</li> <li>De <i>client</i> vraagt om de lijst van oplossingstype (eventueel behorende bij het eerder geselecteerde oplossingstype).</li> <li>De <i>server</i> stuurt bij ontvangst een lijst met oplossingstypen.</li> <li>De medewerker selecteert in het <i>drop-down menu</i> het oplossingstype.</li> <li>De <i>client</i> stuurt de selectie naar de <i>server</i>.</li> <li>De <i>client</i> geeft een nieuw <i>drop-down menu</i> weer, waarin het oplossingstype selecteert kan worden.</li> <li>De <i>server</i> onthoudt bij ontvangst de selectie bij deze gebruiker.</li> <li>De <i>client</i> herhaalt de stappen 6 t/m 12 tot alle oplossingstypen geselecteerd zijn. Afhankelijk van het soort oplossing is er aan het einde de mogelijkheid om een opmerking in een tekstvenster in te voeren. Het definiëren van de oplossing gaat op vergelijkbare manier als bij de probleemdefinitie in UC001.</li> <li>De medewerker klikt op de knop om de oplossing op te slaan.</li> <li>De <i>client</i> stuurt een bericht aan de <i>server</i> op de gedefinieerde oplossing op te slaan.</li> <li>De <i>server</i> schrijft bij ontvangst de oplossingsgegevens en naar database. Deze gegevens worden gekoppeld aan het opgeloste probleem.</li> </ol>
Postconditie	Systeem toont een lijst met Andonlight problemen waar nog geen oplossing voor bijgevoerd is.

Client support departments

Support department	Mechanic	Problem definition	Shop department	Workordernr.	Starttime	Endtime
Engineering	Dirk v/d Heuvel	Documentation→Assembly Sheet	PL	12341234	01-01-12 09:34	01-01-12 09:58

Main solution type  
Main solution type  
Sub solution type  
subtype1  
Sub solution type  
subtype n

Remark  
Text here

Save solution

Figuur 4.4 Schermontwerp van scherm ‘Client support department’

In figuur 4.4 *client* deel van het systeem te zien die medewerkers van de ondersteunende afdeling kunnen gebruiken om een oplossingsdefinitie toe te voeren aan het probleem die de medewerker eerder heeft opgelost.

Tabel 4.10 Beschrijving van velden

Label	Verplicht	Status	Formaat / lengte	Validatie	Toelichting
Support department	n.v .t	O	Text(4,4)	-	
Mechanic	n.v .t	O	Text(15,15)	-	
Problem definition	n.v .t	O	Text(50,50)	-	
Shop department	n.v .t	O	Text(4,4)	-	
Workordernr.	n.v .t	O	Text(10,10)	-	
Starttime	n.v .t	O	Text(14,14)	-	
Endtime	n.v .t	O	Text(14,14)	-	
Main solution type	J	I	Text(15,15)	Dropdown menu	
Sub solution type	J	I	Text(15,15)	Dropdown menu	
Remark	J	I	Text(300,300)	Minimaal 10 tekens en maximaal 300.	

In tabel 4.10 worden de velden van het scherm 'Client support department' besproken.

#### 4.1.3 Use Case specificatie UC004 'Wisselen view dashboard'

De reden voor deze Use Case is dat bij overleg met de *key-users* is gebleken dat sommigen een plattegrond prettig vinden, maar andere liever een grote tabel met gegevens hebben. Beide *views* hebben hun eigen voor- en nadelen. De plattegrond geeft in een oogopslag weer waar de monteur met een probleem zich bevindt. De subafdeling waar de monteur zich bevindt wordt in de grote tabel view ook weergegeven door de afkorting van de subafdeling, maar dit is alleen te lezen als een gebruiker vrij dicht bij het scherm is. Het voordeel van een grote tabel is dat informatie groter op het scherm weergegeven kan worden. Daarom wordt aan de gebruikers van het dashboardscherm de mogelijkheid verschaft om van *view* te wisselen. Bij de plattegrond *view* wordt de ondersteunende afdeling waar hulp van nodig is aangegeven door een kleur.

Tabel 4.11 Kenmerken

Kenmerk	Omschrijving
<i>Aanleiding ("Trigger")</i>	De medewerker van de ondersteunende afdeling wil de 'view' van het dashboard veranderen.
<i>Actors</i>	Medewerker ondersteunende afdeling
<i>Wijze van uitvoering</i>	Interactief
<i>Samenhang met andere Use Cases</i>	-
<i>Frequentie van uitvoering</i>	1 x per maand

Tabel 4.12 Basisscenario — Wisselen view dashboard

Actor	Medewerker ondersteunende afdeling
Preconditie	Dashboard is gestart
Scenario beschrijving	<ol style="list-style-type: none"><li>1. De medewerker ondersteunende afdeling klikt op de knop om de weergave van het dashboard aan te passen.</li><li>2. Het dashboard stuurt een bericht naar de <i>server</i> om de weergave aan te passen.</li><li>3. De <i>server</i> ververscht het scherm vanaf nu met de aangepaste weergave. (Deze instelling wordt onthouden doordat de <i>server</i> een cookie plaatst bij het dashboard.)</li></ol>
Postconditie	'View' van scherm is aangepast

Andonlight dashboard

Change view

Test

mechanic workorder task timer prob. type1 → subtype n remark

C73, C74, C76, C77, CMD

PH, OLE, OAD, OST

PL

EOM

APAS

APIN

ADIS

Location unknown

Figuur 4.5 Schermontwerp van scherm ‘Andonlight dashboard (alternatieve view)’

In figuur 4.5 is alternatieve *view* van dashboardscherm te zien. Het verschil tussen de normale *view* en deze *view* is dat de subafdeling waar de monteur zich bevindt op een schematische plattegrond wordt weergegeven, inplaats van in een kolom.

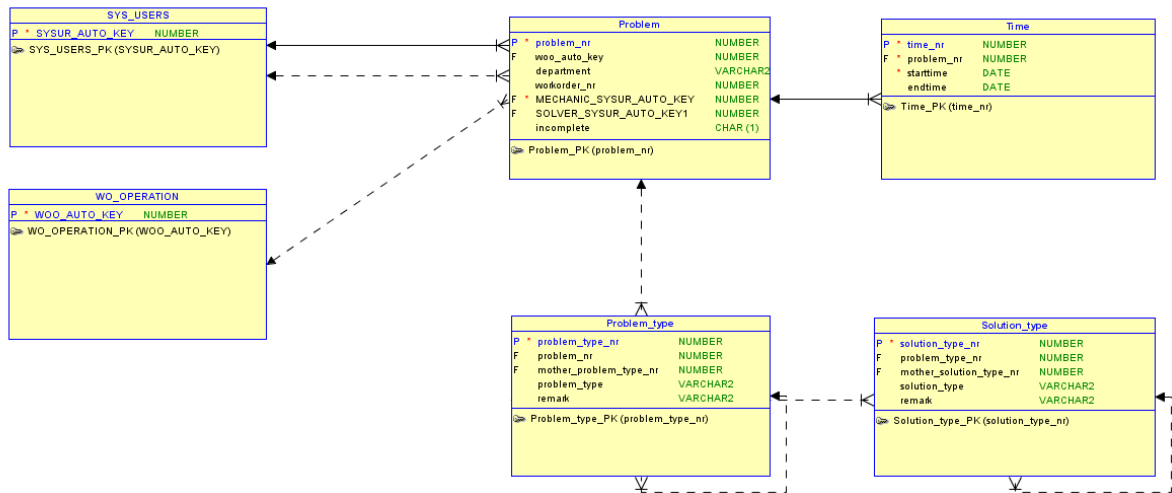
De plattegrond draagt bij aan de eis van schaalbaarheid en onderhoudbaarheid van het nieuwe systeem, omdat bij nieuwe medewerkers, een extra werkstation of een werkstation die verplaatst is het systeem niet aangepast hoeft te worden, zoals bij het huidige systeem wel het geval is. De locatie en medewerkergegevens zullen namelijk uit de Quantum Controle database gehaald worden in plaats van vergeleken met een vaste *user*- en computer tabel.

Tabel 4.13 Beschrijving van velden

Label	Verplicht	Status	Formaat / lengte	Validatie	Toelichting
Mechanic	n.v .t	O	Text(15,15)		
Department	n.v .t	O	Text(4,4)		
Workorder	n.v .t	O	Text(10,10)		
Task	n.v .t	O	Text(15,15)		
Timer	n.v .t	O	Text(5,5)		
Problem definition	n.v .t	O	Text(40,40)		
Remark	n.v .t	O	Text(300,300)		

In tabel 4.13 worden de velden van 'Andonlight dashboard' scherm besproken.

## 4.2 Database model



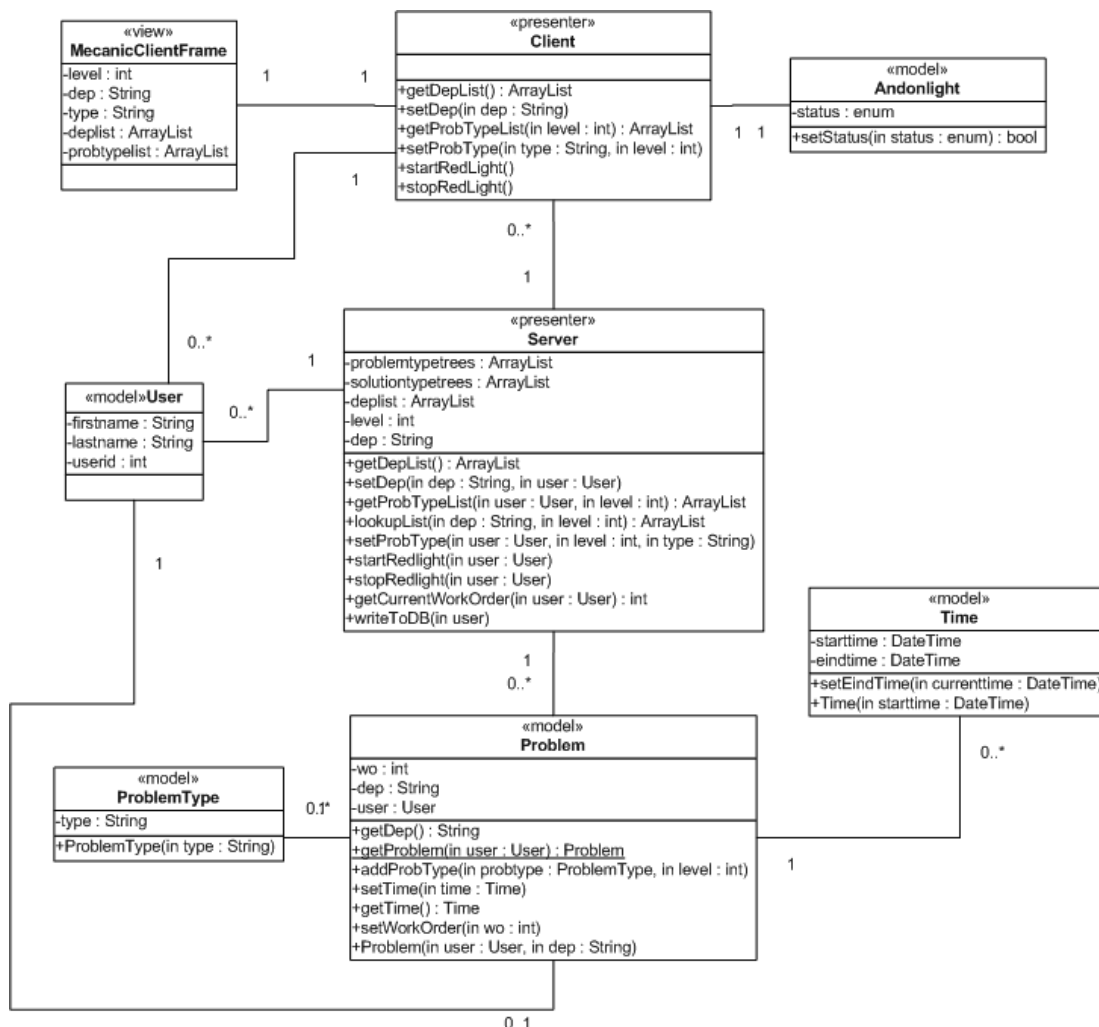
Figuur 4.6 Database model

De vier rechter tabellen in Figuur 4.6 worden toegevoegd aan de bestaande Oracle Quantum Controle database. De *Problem* tabel heeft drie verwijzingen naar tabellen van met Quantum Controle ERP pakket. Twee naar de *SYS\_USERS* tabel, één voor de monteur gebruiker en één voor de gebruiker van de ondersteunende afdeling. Verder is er een koppeling naar de *WO\_OPERATION* tabel. Dit is de hoofdtabel voor de *workorders*. Hierdoor kan een *workorder* aan een probleem gekoppeld worden. Voor de definitie van een probleem en een bijbehorende oplossing bestaan de *Problem\_type* en *Solution\_type* tabellen. Deze tabellen hebben ook een verwijzing naar zichzelf, hierdoor wordt het mogelijk om een selectie uit een omgekeerd boomschema hierin op te slaan. *Problem\_type* en *Solution\_type* corresponderen van de *nodes* in de boomschema's.

## 4.3 Klassendiagrammen

In de klassendiagrammen per Use Case wordt de samenhang tussen de verschillende klassen die nodig zijn weergegeven. Ook worden de methoden en variabelen van de klassen weergegeven. Details over de werking van de methoden wordt bij de *sequence* diagrammen uitgelegd.

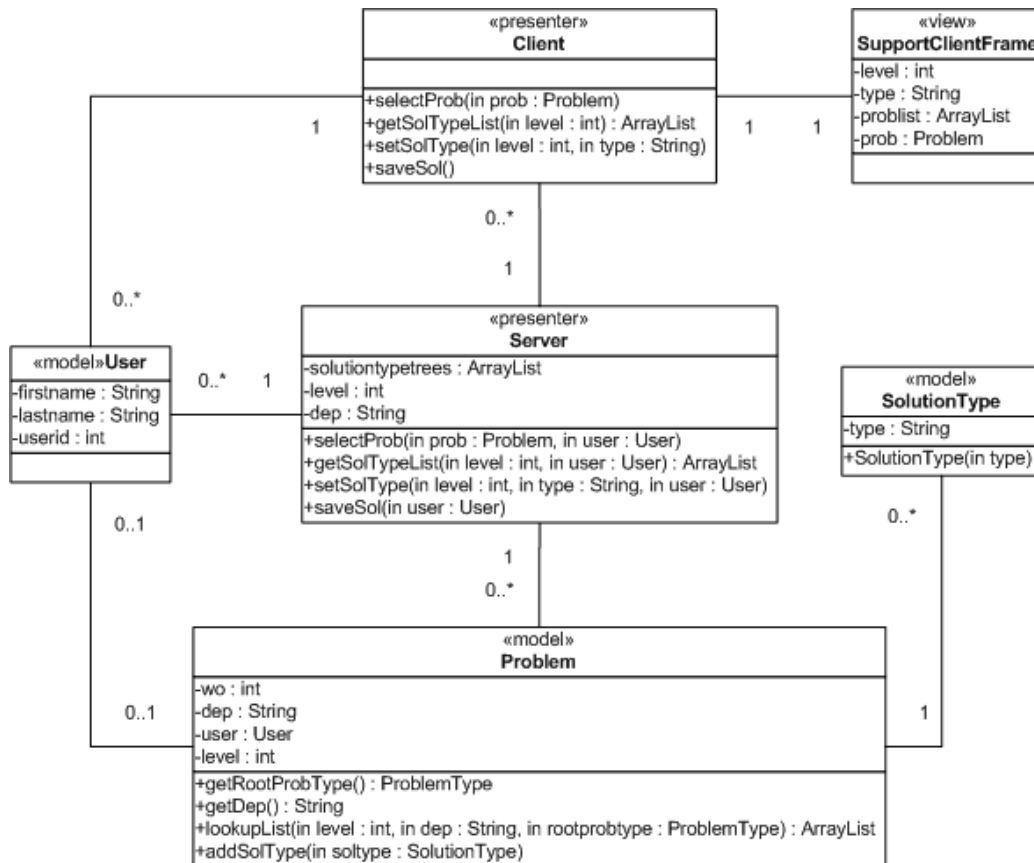
### 4.3.1 Use Case specificatie UC001 'Hulp vragen bij ondersteunende afdeling'



Figuur 4.7 Klassendiagram bij UC001 'Hulp vragen bij ondersteunende afdeling'

In figuur 4.7 is het klassendiagram bij UC001 'Hulp vragen bij ondersteunende afdeling' te zien. De niet voor de hand liggende dingen zullen besproken worden. *Level* is het niveau in het boomschema. *Type* is het probleem type. '*Dep*' betekent de ondersteunende afdeling. Probleem is afgekort met '*prob*'. In '*problemtypetrees*' worden de probleemboomschema's opgeslagen.

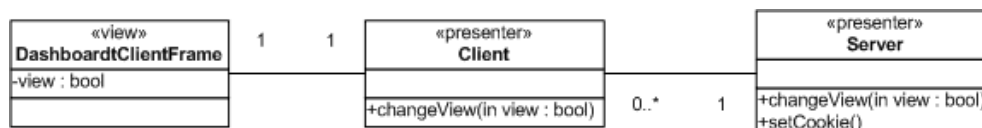
#### 4.3.2 Use Case specificatie UC003 'Oplossing invoeren'



Figuur 4.8 klassendiagram bij UC003 'Oplossing invoeren'

In figuur 4.8 is het klassendiagram te zien bij UC003 'Oplossing invoeren' te zien. *Solution* is bij lange methode namen afgekort naar 'sol'. Met behulp van de methode *lookupList* wordt de lijst met oplossingstypen opgezocht.

#### 4.3.3 Use Case specificatie UC004 'Wisselen view dashboard'



Figuur 4.9 Klassendiagram bij UC004 'Wisselen view dashboard'

In figuur 4.9 is het klassendiagram te zien bij UC004 'Wisselen view dashboard'. De methode *changeView* wordt gebruikt om de *view* van de dashboardserver aan te passen. Met behulp van *setCookie* wordt de geselecteerde *view* onthouden door het plaatsen van een *cookie*.

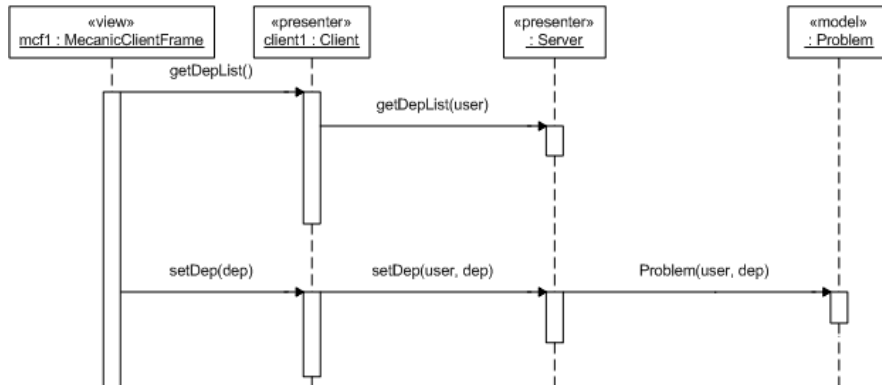
## 4.4 Sequence diagrammen

Een *sequence* diagram geeft weer welke functie aanroepen bij welke klassen nodig zijn om bepaalde handelingen in een Use Case uit te voeren. Belangrijk is dat de geschikte klasse(n) worden gekozen, om een bepaalde handeling uit te voeren. Ook moeten de klassen de juiste informatie ontvangen, in de vorm van parameters, om de handeling uit te kunnen voeren.

In de paragrafen die volgen zullen de *sequence* diagrammen bij de eerder besproken Use Cases besproken worden.

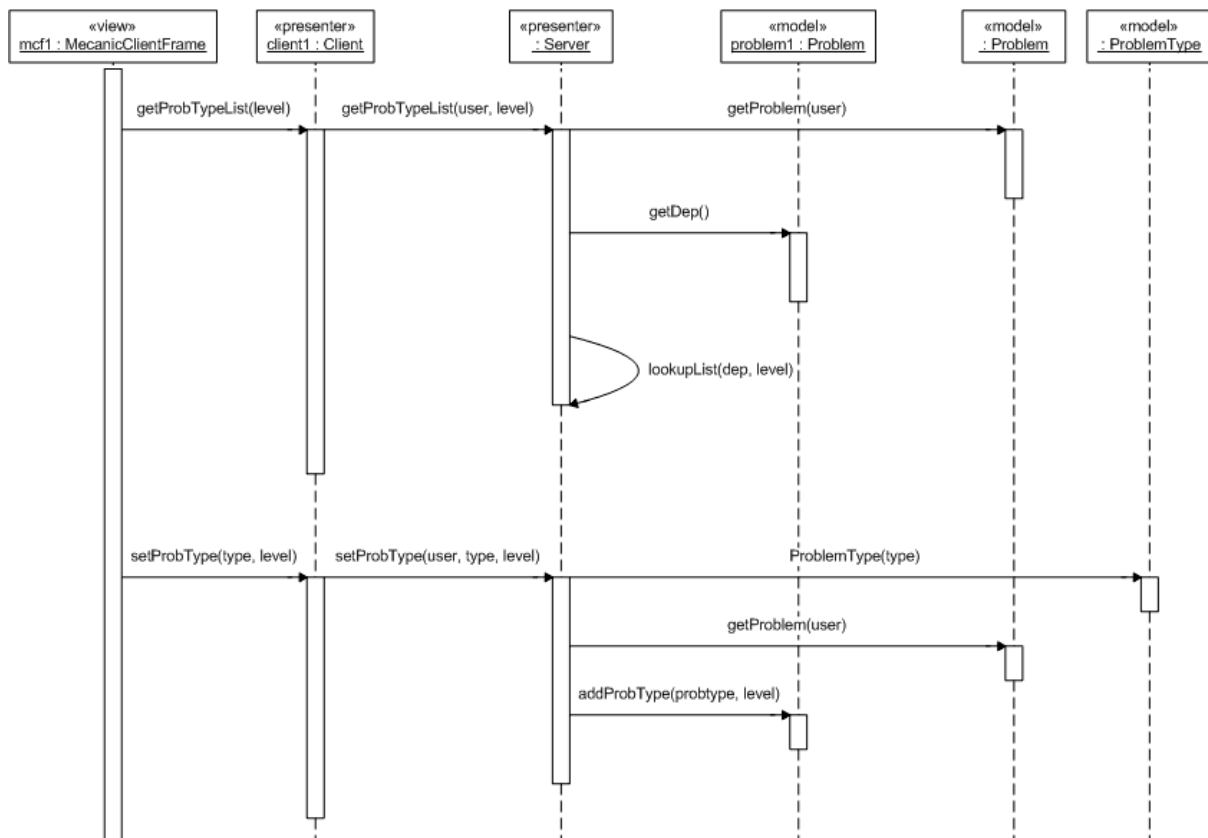


#### 4.4.1 Use Case specificatie UC001 'Hulp vragen bij ondersteunende afdeling'



Figuur 4.10 Selecteer afdeling

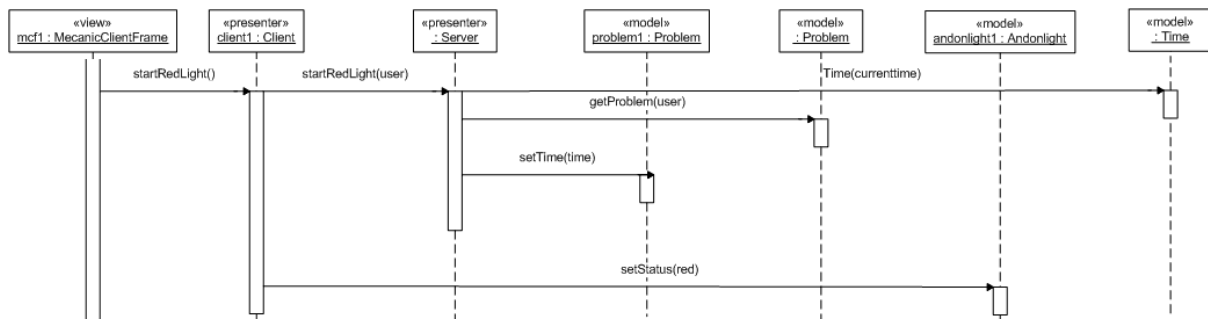
In figuur 4.10 is te zien wat er gebeurt als de monteur in de *client* de afdeling selecteert. Als de monteur op het *drop-down menu* klikt vraagt de *client presenter* de lijst op bij de *server*. Vervolgens kan de *client* de lijst weergeven. Als de monteur in de *client* een afdeling selecteert wordt de selectie van de *client presenter* naar de *server presenter* gestuurd. De *server presenter* maakt een nieuwe instantie van een *Problem* object aan en slaat in dit object de monteur gebruiker en de afdelingsselectie op.



**Figuur 4.11** Selecteer probleem

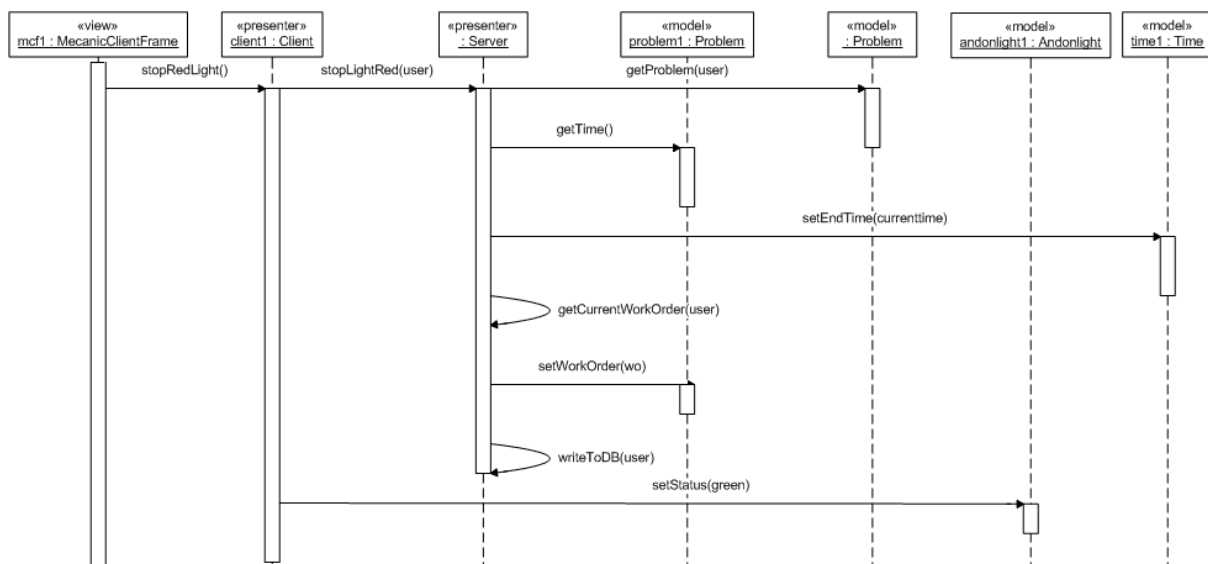
In figuur 4.11 is te zien hoe een probleem type in een *drop-down menu* geselecteerd kan worden. Bij het klikken op het *drop-down menu* wordt de lijst opgehaald met behulp van een aanroep van de GUI klasse *MecanicClientFrame* naar de methode *getProbList* van de *client presenter* met als parameter *level*. *Level* geeft aan op welk niveau in het boomschema de GUI klasse zit. De *client presenter* doet vervolgens dezelfde aanroep naar de *server presenter*, met een extra parameter: *user*. De *client presenter* heeft bij het opstarten bepaald om welke gebruiker (in dit geval een monteur) het gaat. De *server presenter* zoekt aan de hand van de gebruiker de eerder aangemaakte *Problem* klasse op. Met behulp van de *Problem* klasse van de gebruiker kan de eerder geselecteerde afdeling opgehaald worden door de *server presenter*. De *server presenter* roept vervolgens de methode *lookupList* aan met parameters afdeling en niveau. Met de afdeling kan het juiste boomschema geselecteerd worden en met het niveau kan de lijst met *nodes* in de boom die op een bepaald niveau zitten bepaald worden.

De GUI klasse slaat vervolgens de selectie op door de methode *setProbType* met parameters *type* en *niveau* aan te roepen. De *client presenter* doet dezelfde aanroep naar de *server presenter* met de extra parameter: *user*. De *server presenter* maakt vervolgens een nieuwe instantie van de klasse *ProblemType* en geeft de eerder gemaakte selectie in de vorm van de *type* parameter mee aan de *constructor* van de *ProblemType* klasse. Vervolgens wordt met behulp van de gebruiker de *Problem* klasse opgezocht door de *server presenter* en tot slot kan de *server presenter* de nieuw aangemaakte *ProblemType* klasse instantie opslaan en de *Problem* klasse. Ook wordt hierbij de *level* parameter meegegeven, zodat de selectie op het juiste niveau opgeslagen wordt.



**Figuur 4.12** Andonlight van groen of oranje naar rood

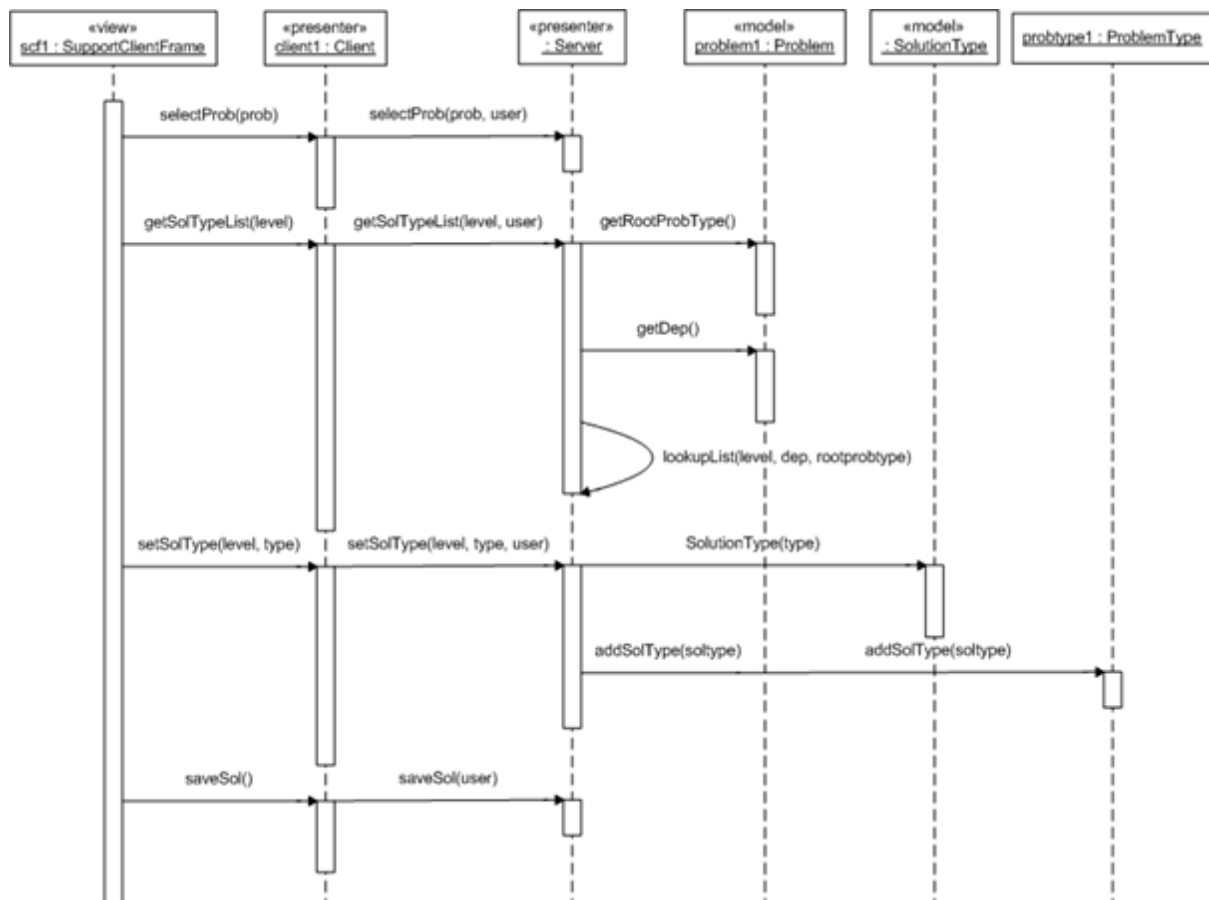
Nadat het probleem gedefinieerd is kan de Andonlight op rood gezet worden. In figuur 4.12 is te zien dat als de Andonlight op rood wordt gezet, dat de GUI klasse een aanroep naar de methode *startRedLight* van de *client presenter* doet. De *client presenter* doet dezelfde aanroep naar de *server presenter* met als extra parameter de gebruiker. De *server presenter* maakt vervolgens een instantie van de *Time* klasse aan met als starttijd de huidige tijd. De eerder aangemaakte *Problem* klasse van de gebruiker wordt opgezocht door de *server presenter*. De aangemaakte *Time* klasse wordt opgeslagen in de *Problem* klasse. Hierna zet de *client presenter* de fysiek Andonlight aan met behulp van de methode *setStatus* van de *Andonlight* klasse.



**Figuur 4.13** Andonlight van rood naar groen

Nadat het probleem is behandeld kan de Andonlight weer op groen gezet worden. In figuur 4.13 is te zien dat de GUI klasse dan de methode *stopRedLight* van de *client presenter* aanroept. De *client presenter* doet dezelfde aanroep naar de *server presenter* met de gebruiker als parameter. De *server presenter* zijn de *Problem* klasse op aan de hand van de gebruiker. Vervolgen vraagt de *server presenter* de *Time* klasse van de *Problem* klasse op. De huidige tijd wordt als eind tijd opgeslagen in de *Time* klasse. Vervolgens zoekt de *server presenter* het workordernummer van de *workorder* waar de monteur op dat moment mee bezig is aan de hand van de gebruiker. Hierna wordt het workordernummer opgeslagen in de *Problem* klasse van de gebruiker. Vervolgens gebruikt de *server presenter* de methode *writeToDB* om alle gegevens die verzameld zijn weg te schrijven naar de database. Het wegschrijven naar de database zal met Hibernate gebeuren. Hierover meer in paragraaf 5 'Software Architectuur'. Tot slot zet de *client presenter* de Andonlight op groen met behulp van de methode *setStatus* van de *Andonlight* klasse.

#### 4.4.2 Use Case specificatie UC003 'Oplossing invoeren'



Figuur 4.14 Koppelen oplossing

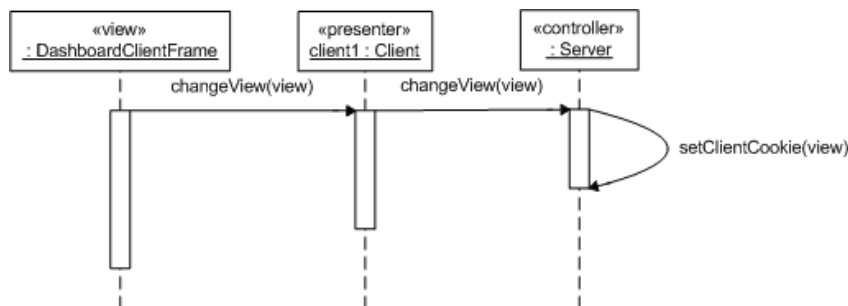
De ondersteunende medewerker selecteert het probleem dat eerder is behandeld, waar een oplossing aan gekoppeld moet worden. In figuur 4.14 is te zien dat de GUI klasse het geselecteerde probleem doorstuurt met behulp van een methode aanroep naar *selectProb* in de *client presenter*, met het probleem als parameter. De *client presenter* doet dezelfde aanroep maar dan naar de *server presenter*, met de gebruiker als extra parameter. (De *server presenter* onthoudt de selectie bij de van de support medewerker.)

Als op het *drop-down menu* voor het oplossingstype wordt geklikt, wordt de lijst van oplossingen van het juiste niveau in het boomschema opgehaald. Dit gebeurt met een aanroep naar de *getSolTypeList* methode van de GUI klasse naar de *client presenter* klasse. De meegestuurde parameter is het niveau in het boomschema. Dezelfde aanroep wordt gedaan van de *client presenter* naar de *server presenter*, met als extra parameter de gebruiker. Vervolgens wordt er een aanroep van de *server presenter* naar de *getRootProbType* methode in de eerder geselecteerde *Problem* klasse gedaan. Deze methode stuurt als antwoord het probleem type dat als eerste geselecteerd is bij het definiëren van het probleem. Dit is dus het minst gedetailleerde probleem type. Hier wordt later de oplossing aan gekoppeld. Vervolgens haalt de *server presenter* de ondersteunende afdeling van het probleem op met een aanroep naar de *getDep* methode in de *Problem* klasse. Met behulp van het niveau, de afdeling en het probleem type als parameters kan de *server presenter* met een aanroep naar *lookupList* de lijst van oplossingen uit het juiste niveau van de juiste boomschema halen.

Hierna wordt het geselecteerde oplossingstype opgeslagen. Dit gebeurt vanuit de GUI klasse door de aanroep naar de *setSolType* methode in de *client presenter*. De parameter zijn het niveau en het oplossingstype. De *client presenter* doet vervolgens dezelfde aanroep naar de *server presenter*, met als extra parameter de gebruiker. Vervolgens maakt de *server presenter* een nieuwe instantie van een *SolutionType* klasse met als parameter voor de *constructor* het oplossingstype. De nieuw aangemaakte *SolutionType* klasse wordt tot slot in de *ProbleemType* klasse opgeslagen door een aanroep van de *server presenter* naar de *addSolType* methode in de eerder opgevraagde *ProblemType* klasse.

Om alle verzamelde gegevens op te slaan in de database wordt door de GUI klasse een aanroep gedaan naar de *saveSol* methode in de *client presenter*. De *client presenter* doet dezelfde aanroep in de *server presenter*, met als extra parameter de gebruiker. Dit wordt opgevoerd door Hibernate.

#### 4.4.3 Use Case specificatie UC004 'Wisselen view dashboard'

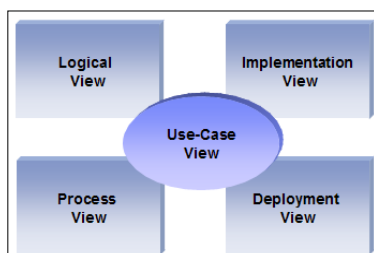


Figuur 4.15 Wisselen 'view'

In figuur 4.15 is te zien wat er gebeurt als de *view* van het dashboard wordt veranderd. De *DashboardClientFrame* klasse aan de browser kant doet een aanroep naar de *changeView* methode in de *Client* klasse, met als parameter de gewenste *view*. De *Client* klasse doet vervolgens dezelfde aanroep naar de *Server* klasse. Deze klasse bevindt zich op de *webserver*. De *Server* klasse verandert in deze methode de *view* en onthoudt de geselecteerde *view* door een *cookie* bij de *client* te plaatsen. Dit gebeurt door een aanroep naar de *setClientCookie* methode, die als parameter de geselecteerde *view* heeft. (Zowel de *client* als de *server* kant kunnen in Java klassen geschreven worden, omdat gebruik gemaakt wordt van GWT, hierover meer in het volgende paragraaf.)

## 4.5 Software Architectuur

In deze paragraaf zal ingegaan worden op de software architectuur van het nieuwe systeem.

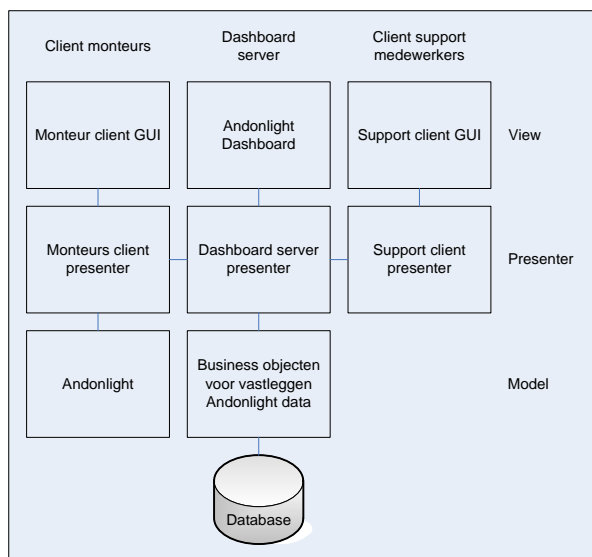


Figuur 4.16 4+1 RUP view model

Hiervoor wordt gebruik gemaakt van het RUP 4+1 view model. Het RUP 4+1 view model is te zien in figuur 4.16. Use Cases zijn eerder behandeld. Bij de Proces View (communicatie van processen) wordt niet apart stilgestaan, omdat dit al behandeld is in de vorm van de *sequence* diagrammen. Ook worden de processen die draaien al in de Deployment View weergegeven.

In de Logical View wordt stilgestaan bij de conceptuele structuur van het systeem. In de Implementation View wordt stilgestaan bij de technische invulling van de conceptuele structuur. In de Deployment View wordt stilgestaan bij de hardware eisen van het systeem, rekening houdend de informatie uit andere *views* en de relevante niet-functionele eisen.

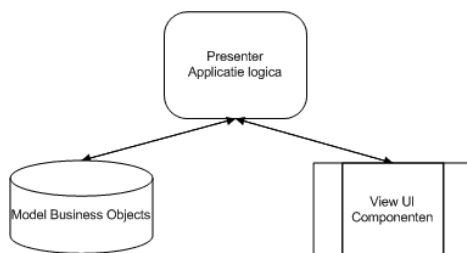
#### 4.5.1 Logical View



Figuur 4.17 Logical view

In figuur 4.17 is de Logical view te zien. Er wordt onderscheidt gemaakt tussen de *client* voor de monteurs, de *client* voor de *support* medewerkers en de *dashboardserver*.

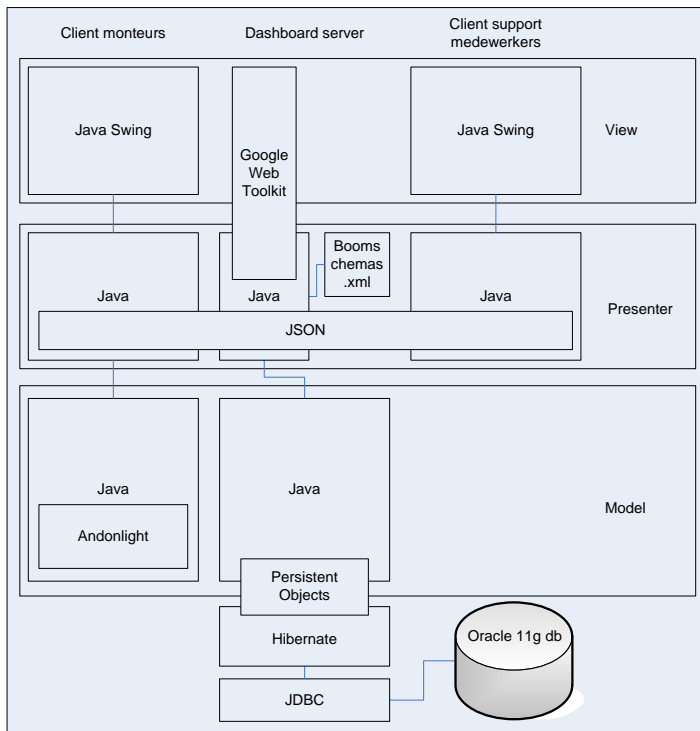
De *clients* en de *server* zijn met behulp van het Model View Presenter *design pattern* onderverdeeld in een *model*, een *view* en een *presenter* component.



Figuur 4.18 Model View Presenter design pattern

In figuur 4.18 is de structuur van MVP te zien. De *model* verschaft met behulp van een interface aan de *presenter* toegang tot de data. De *view* verzorgt de GUI. *Events* in de *view* worden met behulp van parameters in methode aanroepen doorgegeven aan de *presenter*. Als er bij de aanroep gegevens opgeslagen moeten worden gebruikt de *presenter* de *model* interface. Als de *view* gegevens moet weergeven gaat dit ook met een methode aanroep naar de *presenter*. De *presenter* gebruikt de doorgegeven parameters om de opgevraagde gegevens samen te stellen en terug te sturen.

#### 4.5.2 Implementation View



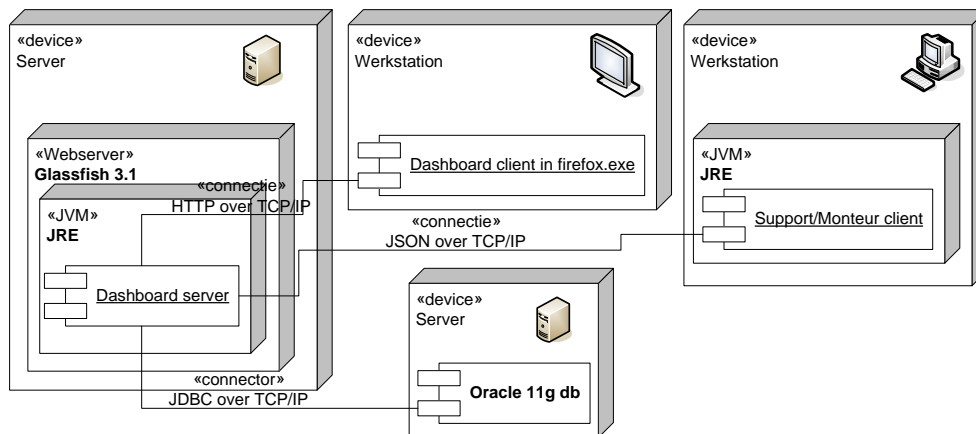
Figuur 4.19 Implementation view

In figuur 4.19 is de Implementation View te zien. De *clients* voor de monteurs en support medewerkers zullen voor de *view* gebruik maken van de Swing GUI klasse. De *dashboardserver* zal in de *view* gebruik maken van Google Web Toolkit, voor het weergeven naar de dashboard webpagina. Het voordeel van GWT is dat deze gebruikmaakt van een *crosscompiler* die Java klassen compileert naar Javascript, hierdoor kan het hele systeem in Java geschreven worden en is het integreren van AJAX (Asynchronous JavaScript and XML) technieken zeer eenvoudig. Ook verschaft GWT een GUI builder.

De *presenter* zal voor de communicatie tussen de *clients* en de *server* gebruik maken van JavaScript Object Notation. JSON wordt gebruikt voor de uitwisseling van datastructuren, met behulp van JSON-expressies.

In de *model* wordt gebruik gemaakt van Hibernate om Java data objecten naar database tabellen te mappen.

#### 4.5.3 Deployment View



**Figuur 4.20 Deployment View**

De niet-functionele eisen die betrekking hebben op de Deployment View zijn:

- Schaalbaar
- Betrouwbaar

In figuur 4.20 is de samenhang te zien tussen de *executables* die nodig zijn voor het nieuwe systeem.

De monteur- en support medewerkers *clients* zullen gaan draaien op de werkstations van de medewerkers. Op ieder werkstation is Java al geïnstalleerd. De *clients* zullen gaan draaien in de Java Runtime Environment. Naar verwachting zullen de *clients* weinig RAM en processor tijd opeisen. De huidige werkstations voldoen aan de eisen van de *clients*.

De tabellen voor het Andonlight systeem zullen in de bestaande Oracle 11g database geplaatst worden van het Quantum ERP pakket. De database *server* heeft hiervoor voldoende capaciteit.

De dashboard pagina wordt weergegeven op grote beeldschermen op de ondersteunende afdeling, in een webbrowser, die zal gaan draaien op een werkstation die alleen voor dit doet gebruikt wordt. Dit zal heel weinig capaciteit van deze machines opeisen.

De *dashboardserver* zal gaan draaien in de JRE van een Glassfish *webserver*. Glassfish is eenvoudig te installeren en te configureren. Configuratie gaat met behulp van een *web interface*. Glassfish is geschikt voor verschillende platformen waaronder Windows. De huidige *dashboardserver* draait op een werkstation. Rekening houdend met schaalbaarheid en betrouwbaarheid wordt aangeraden om de Glassfish *server* voor het nieuwe systeem op hardware te installeren die speciaal bedoeld is om als *webserver* te gebruiken. Eventueel kan gekozen worden voor een *server* met veel capaciteit die virtueel opgedeeld is. De gereserveerde capaciteit voor de *dashboardserver* kan dan eenvoudig uitgebreid worden, als dit nodig is. Naar verwachting zal de *webserver* met het huidige aantal medewerkers niet veel capaciteit nodig hebben. Tijdens dit project tot nu toe is een maximum van 5 gelijktijdige rode Andonlights waargenomen.

Tot slot is de huidige netwerkverbinding Gigabit Ethernet. Gigabit Ethernet verschaft ruimvoldoende bandbreedte voor het nieuwe systeem.



## 5 Implementatie

Tijdens en nadat het systeem ontwikkeld is, zal deze in een testomgeving getest worden. Als de eerste versie van het systeem af is zal conform het Plan van Aanpak een systeem test uitgevoerd worden. Hierover meer in de volgende paragraaf. Nadat de systeemtest is uitgevoerd en het systeem eventueel verbeterd is, zal het systeem uitgerold worden. Als het nieuwe systeem is uitgerold kan gekeken worden of de boomschema's aangepast of aangevuld moeten worden.

### 5.1 Systeemtest

Conform de productplanning van het Plan van Aanpak zal nadat het systeem in een testomgeving draait een systeem test uitgevoerd worden.

De invoer van het systeem bestaat voornamelijk uit *drop-down menu's* en tekst invoervensers. Hierbij kan de gebruiker relatief weinig fout doen. Er moet alleen gelet worden op een minimum en maximum aantal tekens in de tekst invoervensers. Daarom is gekozen voor een systeemtest. Bij een systeemtest wordt namelijk het resultaat van een bepaalde actie getest. Omdat het project relatief klein is en omdat het systeem door één programmeur wordt gemaakt is er voor gekozen om geen *white box* tests uit te voeren. (In een *white box* test wordt naar de interne werking van een systeem gekeken.)

In een systeemtest wordt het systeem als een *black box* gezien. Dit houdt in dat de interne werking van het systeem onbekend is. Er wordt per Use Case een Test Case opgesteld. In de Test Cases worden testgevallen geformuleerd. Deze hebben de vorm ALS x DAN y. Een voorbeeld voor de monteur *client* is: ALS in de *client* de afdeling geselecteerd wordt, DAN verschijnt er een nieuw *drop-down menu* voor het selecteren van het eerste probleemtype. De testgevallen worden vervolgens vertaald naar testacties met een actie en een verwacht resultaat en werkelijk resultaat. Als het verwachte- en het werkelijk resultaat afwijken, moet het systeem gecorrigeerd worden.

## 6 Onderbouwing

In dit hoofdstuk wordt stilgestaan bij de gemaakte keuzes.

### 6.1 Database en Applicatie keuzes

Tabel 6.1 Applicatie- en Database beslissingen

Database					
Tabellen in Quantum database			Eigen database		
voordelen	nadelen	weging	voordelen	nadelen	weging
Eenvoudig te queriën		3	Zonder externe partij op te zetten		1
Referentieel integer		3		Queries moeten op meerdere databases uitgevoerd worden	-3
	Er moet hiervoor een <i>schema</i> gemaakt worden door Quantum	-1		Referentiele integriteit zeer moeilijk te implementeren	-3
		5			-5
Clients					
Integreren in Quantum			Standalone Windows applicatie		
voordelen	nadelen	weging	voordelen	nadelen	weging
Alles in één		1	Tijd		3
Onderhoud door Quantum		5	Relatief lage kosten		3
	Tijd	-5	Geen communicatie problemen		3
	Kosten	-5	Alleen gelimiteerd door mogelijkheden van Java programmeertaal		2
	Communicatie (moet in samenwerking met Quantum geprogrammeerd worden)	-5		Monteur heeft twee applicaties open staan op Desktop	-1
	Gelimiteerd door mogelijkheden binnen Quantum	-1		Onderhoud door Epcor	-5
		-10			5

In Tabel 6.1 is te zien hoe voor de database en de *clients* een keuze is gemaakt. De voor- en nadelen zijn gewogen met behulp van punten van 1 tot 5. Positief bij een voordeel en negatief bij een nadeel.

Er moeten hoe dan ook gegevens uit de Quantum Controle database gebruikt worden. Als de tabellen voor het Andonlight systeem in een eigen database worden gezet zullen verwijzingen naar tabellen in de Quantum Controle database niet referentieel integer zijn. De Andonlight *Problem* tabel heeft bijvoorbeeld refererende sleutels die verwijzen naar de Quantum *SYS\_USERS* tabel. Als er een verwijzing is naar een bepaalde gebruiker dan blijft de refererende sleutel bestaan in de Andonlight *Problem* tabel ongeacht of de sleutel waar naartoe verwezen wordt nog bestaat. De bovengenoemde situatie is niet op een nette manier op te lossen, zolang er twee databases zijn. Volgens de leverancier Component Controle<sup>7</sup> worden echter gegevens nooit verwijderd. Toch wordt dit als een nadeel gezien, omdat hier nooit vanuit gegaan mag worden. Als de gegevens in twee databases staan zullen er ook twee verbindingen gemaakt moeten worden, als er een query uitgevoerd wordt. Niet alle tools voor het presenteren van de informatie kunnen dit. Op basis van de argumentatie hierboven

<sup>7</sup> Leverancier van het ERP pakket dat door Epcor B.V. gebruikt wordt: Quantum Control

wordt geadviseerd om de Andonlight tabellen in de bestaande Quantum Controle database te integreren. Hiervoor kan Component Controle een *schema* maken. Met behulp van deze *schema* kunnen de Andonlight tabellen toegevoegd worden.

Voor de *clients* is er de mogelijkheid om deze in het Quantum Controle ERP pakket te integreren, of om hiervoor een *standalone* Windows applicatie te ontwikkelen. Een webapplicatie is niet overwogen, omdat om dit mogelijk te maken *custom plugins* geschreven zouden moeten worden, zodat de webapplicatie toegang krijgt tot de Windows gebruikersnaam, computernaam en fysieke Andonlight *driver*, om de Andonlight te kunnen bedienen. De voordelen van het integreren van de *clients* in Quantum Controle wegen niet op tegen de nadelen. Geadviseerd wordt om de *clients* (voor monteurs en *support* medewerkers) als *standalone* Windows applicaties te ontwikkelen. Bij de ontwikkeling van het systeem zal een Application Programming Interface geschreven worden voor de *driver* die de fysieke Andonlight bedient. Ook zullen de *clients* met behulp van een API toegang hebben tot de *dashboardserver*. Hierdoor is het voor Component Controle heel eenvoudig om op een later tijdstip als nog de *clients* in Quantum te integreren. Hiervoor hoeft dan alleen code aan het Quantum ERP pakket toegevoerd te worden. De *clients* in Quantum maken dan gebruik van de API's.

## 6.2 Design pattern

Het nieuwe systeem moet gegevens vastleggen in een database. Ook hebben de *clients* een GUI waarmee de Andonlight bediend kan worden en waarmee gegeven aan de *server* doorgegeven kunnen worden en gegevens van de *server* opgevraagd kunnen worden. De *server* geeft de Andonlight statussen grafisch weer op de dashboardschermen en verwerkt aanroepen van de *clients*.

Het is daarom logisch om een *design pattern* te kiezen dat minimaal onderscheid maakt tussen een data onderdeel, een gebruikersinteractie onderdeel en een onderdeel waarin de logica wordt ondergebracht.

Er is gekozen voor het MVP (Model View Presenter) *design pattern*, te zien in figuur 4.18.

Het voordeel van MVP in vergelijking met andere *design patterns* zoals MVC (Model View Controller) is dat de *view* alleen met de *presenter* communiceert en niet ook met de *model*. Hierdoor is er een minder sterke koppeling tussen de *view* en de rest van het systeem. Het voordeel hiervan is dat de *view* makkelijker vervangen kan worden door een andere *view*. Voor Epcor B.V. is dit een voordeel omdat eventueel de *view* op een later tijdstip eenvoudig in het Quantum ERP geïntegreerd kan worden.

Een tweede voordeel is dat MVP goed samengaat met GWT (Google Web Toolkit) die gebruikt gaat worden voor een deel van de *view* en Hibernate die gebruikt gaat worden voor de *model*.

Als MVP in de meest strikte vorm wordt geïmplementeerd geeft de *view* gebruikers events aan de *presenter* door met een aanroep zonder parameters en zonder dat er een variabele wordt terug gestuurd. Vervolgens haalt de *presenter* de data om de *event* af te handelen op met behulp een *interface* die de *view* verschaft. Nadat de data is verwerkt wijzigt de *presenter* de *view* met behulp van de *view interface*.

Er is niet voor deze strikte vorm gekozen, omdat het voordeel van een toename in *loose coupling* niet opweegt tegen de toename in extra complexiteit, door meer methode aanroepen.

## 6.3 Systeem gebaseerd op transacties

Vanaf het moment dat een Andonlight op rood wordt gezet, tot het moment dat deze weer op groen wordt gezet, wordt data verzameld door het systeem. Vervolgens wordt deze data vastgelegd in de database in de vorm van een *Problem*. Omdat het systeem *clients* en een *server* heeft is het mogelijk dat de verbinding verbroken wordt voordat de Andonlight weer op groen wordt gezet. Als hier geen rekening mee gehouden wordt kan er incomplete data in de database opgeslagen worden.

Als de periode tussen een rode en een groene Andonlight als een transactie wordt gezien wordt de data alleen in de database opgeslagen als een transactie volledig is afgerond. Dit garandeert dat de data in de database compleet is.

Als de verbinding verbroken wordt voordat de transactie is afgerond kan er een beslissing genomen wat er met de incomplete data wordt gedaan. Deze kan weggegooid worden of toch opgeslagen worden in de database, maar dan met een opmerking erbij dat de transactie niet helemaal uitgevoerd is. Epcor B.V. heeft voor het laatste gekozen, omdat hierdoor geen enkele data verloren gaat. Het is geen probleem voor Epcor B.V. dat sommige kolommen in de database hierdoor niet altijd gevuld worden.

## 6.4 Boomschema's

Problemen en oplossingen kunnen geclassificeerd worden in bepaalde groepen aan de hand van kenmerken die ze bezitten. Het nadeel hiervan is echter dat er altijd problemen en oplossingen zullen zijn die buiten de groepen vallen. Deze komen dan in een groep 'overige', maar deze geeft weinig nuttige informatie.

Een andere aanpak is dat medewerkers met behulp van een tekst omschrijven wat het probleem of de oplossing is. Hierdoor kunnen alle problemen en oplossingen omschreven worden. Het nadeel hiervan is uiteraard dat problemen en oplossingen op veel verschillende manieren omschreven zullen worden. Analyse van de data wordt hierdoor zo goed als onmogelijk.

Er is daarom voor gekozen om een combinatie van beide te maken. Met ondersteunende afdelingen zijn boomschema's gemaakt met groepen waar een hoog percentage van de problemen en oplossingen niet onder de groep overige vallen. Mocht een bepaald probleem of oplossing toch onder 'overige' vallen dan is het mogelijk om deze in een opmerking te omschrijven. Ook is het bij problemen en oplossingen die niet onder de groep 'overige' vallen in sommige gevallen mogelijk om een opmerking toe te voegen, waardoor het probleem of de oplossing nog verder en detail gedefinieerd kan worden. Ook kan met behulp van een opmerking extra informatie toegevoegd worden, bijvoorbeeld een onderdeelnummer.

## 7 Werkwijze

Er is gewerkt met het Plan van Aanpak als uitgangspunt.

Om het project in het algemeen structuur te geven is gebruik gemaakt naar PRINCE2. Deze methode is onder andere gebruikt om de risico's, belanghebbenden, planning en de scope te bepalen. In de Opstart-, Initiatie- en Analysefase is gebruik gemaakt van deze methode.

Voor het ontwerp en de ontwikkeling van het Systeem is gekozen voor de OpenUP. OpenUP is een iteratief softwareontwikkelingsproces. Deze methode biedt uitgebreide mogelijkheden voor het uitvoeren en beheersen van het softwareontwikkelingsproces. Het is geschikt voor kleine projecten. Ook heeft de uitvoerder van het project ervaring met deze methode. In de ontwikkelingsfase waar het ontwerp ook deel van uitmaakt is OpenUP gebruikt. De ontwikkelingsfase is onderverdeeld in de Elaboration-, Construction- en de Transition fase. In de Elaboration fase en een deel van de Construction fase zijn de ontwerp producten opgeleverd. In het resterende deel van de Construction fase zal het systeem ontwikkeld worden. In de Transition fase zal het systeem getest worden en vervolgens zal de uitrol plaatsvinden. Ook zal er in deze fase een handleiding bij het systeem gemaakt worden.

In de Analyse fase is meegelopen op alle relevante ondersteunende afdelingen en de Shop afdeling. Dit heeft bijgedragen aan het in kaart brengen van de relevante processen en de werking van het huidige systeem. De boomschema's zijn gemaakt in samenwerking met medewerkers van de ondersteunende afdelingen.

De ontwikkelaar van het huidige systeem is geïnterviewd. Dit heeft bijgedragen aan het in kaart brengen van het huidige systeem.

De uitvoerder van het project heeft een Lean Six Sigma training gekregen van een expert binnen Epcor B.V.

Er is een tweedelige *brainstorm* sessie gehouden met de *key-users* waarin de functionele- en niet-functionele eisen voor het nieuwe systeem zijn vastgesteld.

Aan het einde van het operationeel management team overleg is uitleg gegeven over wat het project inhoudt en zijn door het management beslissingen genomen over de extra functionaliteiten van het systeem.

Er is periodiek gesproken met de opdrachtgever van het project, een medewerker waaraan een begeleidende taak is gedelegeerd, de Shop manager (hoofd van de Shop afdeling) en met de studie begeleider over de voortgang en de aanpak.

Er zijn *conference calls* gehouden met Component Controle. De *conference calls* zijn gehouden om beslissingen te kunnen nemen over de applicatie- en databasekeuze.

## 8 Conclusie en Aanbevelingen

Met behulp van de producten uit de analyse- en ontwerp fase is antwoord gegeven op de hoofdvraag:

“Hoe kan bij Epcor B.V. door het vernieuwen van het Andonlight systeem de gebruiksvriendelijkheid verbeterd worden voor de *key-users* aan de hand van hun voorkeuren en welke data moet het systeem opslaan zodat deze geanalyseerd kan worden, zodat op de bevindingen die hieruit voortkomen geanticipeerd kan worden?”

Producten analyse fase:

- Systeemmodel
- Business Process Model
- Nulmeting
- Boomschema's
- Eisen

Producten ontwerp fase:

- Use Cases
- Database Model
- Klassendiagrammen
- Sequence Diagrammen
- Software Architectuur

Ook is er stilgestaan bij de implementatie van het nieuwe systeem.

De producten uit de analyse- en ontwerpfase en de informatie over de implementatie dienen als een uitstekende basis voor de ontwikkeling en implementatie van het nieuwe systeem.

De opdrachtgever stelde voor om het onderhoud van het nieuwe systeem uit te laten voeren door de externe ICT-dienstverlener van Epcor B.V. Ik beveel dit ook aan.

Verder kan er een extra functionaliteit aan de monteur- en support *client* toegevoegd worden, waardoor een overzicht van alle probleem- en oplossingsboomschema's weergegeven kan worden. Dit helpt de medewerker met de juiste keuze maken in de boom.

Tot slot zou er een extra functionaliteit aan het systeem toegevoegd kunnen waarmee met behulp van een GUI de structuur van de boomschema's aangepast kan worden. De toegevoegde waarde hiervan is afhankelijk van hoe vaak er in de praktijk aanpassingen gedaan worden.

## 9 Evaluatie procesgang

Er is relatief veel tijd gestoken in het opstellen van een kwalitatief goed Plan van Aanpak. Dit is achteraf een goede keus geweest, omdat er vrij weinig tijd was tot de *deadline*. Er kon daarom weinig uitlooptijd ingepland worden. De voortgang kon continue gecontroleerd worden met behulp van de productplanning en rekening gehouden met de belangrijkste risico's en de geformuleerde *scope*.

De analysefase heeft redelijk wat tijd in beslag genomen, omdat hiervoor met veel verschillende medewerkers gesproken is. Deze medewerkers hebben uiteraard ook allemaal hun eigen agenda. De opdrachtgever heeft mij van begin af aan hierop geattendeerd. Hierdoor zijn in een vroeg stadium al afspraken met medewerkers gemaakt.

Er was geanticipeerd dat tijdens de *brainstorm* sessie de *key-users* vooral over de processen zouden gaan discussiëren en niet over de eisen van het nieuwe systeem, omdat er toen wat meningsverschillen waren over de processen. Hier is rekening mee gehouden door tijdens de sessie een duidelijke agenda met punten te presenteren. Per punt was een bepaald tijd gereserveerd.

Sommige *key-users* hadden het idee dat bepaalde extra functionaliteiten van het nieuw systeem mogelijk de processen negatief zouden beïnvloeden. Over de functionaliteiten zijn toen beslissingen genomen door het management. Er was hiervoor geen vergadering ingepland, omdat het dan waarschijnlijk te lang zou duren om iedereen bij elkaar te krijgen. Het is besproken aan het einde van het Operationeel Management Team overleg, omdat dan toch iedereen al bij elkaar zit. Hierdoor is er weinig tijd verloren gegaan.

Aan het begin van de ontwerpfase dreigde de applicatie en database keuzes veel tijd te gaan kosten. Daarom is een beslissing genomen door de voor- en nadelen van de verschillende mogelijkheden te wegen.

## Bronvermelding

(sd). Opgehaald van RUP op Maat: <http://www.rupopmaat.nl/>

Bentley, J. L. (2000). *Programming Pearls*. Addison-Wesley.

*BPMN 2.0*. (sd). Opgehaald van Object Management Group: <http://www.omg.org/spec/BPMN/2.0/PDF>

Hedeman, B., van Heemst, G., & Fredriksz, H. (2009). *Projectmanagement op basis van PRINCE2*. van Haren Publishing B.V.

Korpershoek, I., & Groenendijk, B. (sd). *Databases en Access 2003*.

Moergestel, L. J. (2007). *Computersystemen en embedded systemen*. Sdu uitgevers bv.

Nathans, H. (2010). *Adviseren als tweede beroep*. Kluwer.

*OpenUP*. (sd). Opgehaald van <http://epf.eclipse.org/wikis/openup/>

Skansholm, J. (sd). *Java from the Beginning*. Addison-Wesley.

van der Lans, R. (sd). *Het SQL Leerboek, 6e druk*. Acedemic Service.

Warmer, J., & Kleppe, A. (sd). *Praktisch UML 4de editie*. Pearson Education.



## **Bijlage A: Plan van Aanpak**

### **Plan van Aanpak**

**EPCOR B.V.**

**Versie: 1.3**

**Datum: 13-04-2012**

**Student: Eric Yntema**

**Studentnummer: 1139126**

## Versiebeheer

Versie	Datum	Omschrijving	Opmerking
1.0	18-03-2012	Initiële versie	
1.1	30-03-2012	Revisie	
1.2	03-04-2012	Revisie	Aanpassingen in projectprobleemstelling, projectorganisatie, business case en definitielijst
1.3	13-04-2012	Revisie	Aanpassingen aan scope en producten na gesprek met docentbegeleider

## Distributie

Naam	Rol	v0.1	v0.2	v0.3	v0.4	v1.0	v1.1		

# INHOUD

INHOUD	51
--------	----

INLEIDING	52
-----------	----

## 1 PROJECTDEFINITIE 53

1.1.	AANLEIDING	<b>FOUT! BLADWIJZER NIET GEDEFINIEERD.</b>
1.2.	PROJECTPROBLEEMSTELLINGEN	53
1.3.	PROJECTDOELSTELLINGEN	54
1.4.	BUSINESS CASE	54
1.5.	CONTEXT	54
1.6.	SCOPE	55
1.7.	RANDVOORWAARDEN	55
1.7.1.	RANDVOORWAARDEN KWALITEIT	55
1.7.2.	RANDVOORWAARDEN TIJD	55
1.8.	PROJECTRISICO'S	57

## 2 PROJECTAANPAK 58

2.1.	HOOFDLIJNEN AANPAK	58
2.2.	FASERING EN BESLUITMOMENTEN	58
2.2.1.	DE OPSTARTFASE	58
2.2.2.	INITIATIEFASE	59
2.2.3.	ANALYSEFASE	59
2.2.4.	ONTWIKKELINGSFASE	59
2.2.5.	AFRONDFASE	61
2.3.	PROJECTORGANISATIE	61
2.3.1.	ORGANISATIESCHEMA	62
2.3.2.	ROLLEN EN ROLVERDELING	62
2.3.3.	OVERLEGVORMEN EN FREQUENTIES	63
2.4.	PROJECTPLANNINGEN	63
2.4.1.	UITGANGSPUNTEN	63
2.4.2.	PRODUCTPLANNING	63

## 3. PROJECTBEHEERSING 65

3.1.	BEHEERSING VAN DE SCOPE	65
3.2.	BEHEERSING VAN TIJD EN GELD	65
3.3.	BEHEERSING VAN KWALITEIT	65

DEFINITIELIJST	65
----------------	----

BIJLAGEN	66
----------	----

## INLEIDING

Dit document vormt het Plan van Aanpak voor het Project inrichting Andonlight Systeem. Hierin wordt de opdracht, de planning en Project beheersing vastgelegd.

### Referentiedocumenten

Dit Plan van Aanpak is gebaseerd op de volgende documenten:

- Productplanningv1.3.xls.

# 1. PROJECTDEFINITIE

In dit hoofdstuk wordt duidelijk wat het Project inhoudt.

## 1.1. Aanleiding

Om de aanleiding tot de opdracht uit te kunnen leggen moeten eerst een aantal dingen over EPCOR B.V. uitgelegd worden.

Het primaire proces van EPCOR B.V. bestaat uit reparatie en revisie aan pneumatisch aangedreven componenten en Auxiliary Power Units (APU) van vliegtuigen. Binnen EPCOR B.V. werken verschillende monteurs in het primaire proces. Een voorbeeld van het proces waar een Unit doorheen gaat is:

*Write Up & Incoming Test, Disassembly, Cleaning, Modification1, Modification2, Parts Inspection and Write Up, Repair, Part Management, Assembly, Final Test.*

Binnen EPCOR B.V. wordt gebruik gemaakt van een Andonlight Systeem. Dit is een 'stoplicht' die de monteur in een programma op zijn/haar computer kan gebruiken, bij zijn/haar werkplek. Als de monteur begint met werk wordt de Andonlight aangezet met een groen licht. Zolang de monteur kan doorwerken blijft de Andonlight op groen. Als de monteur niet verder kan gaan moet de Andonlight gebruikt te worden. De Andonlight kan op oranje of rood gezet worden. De monteur moet de Andonlight eerst op oranje zetten. Dat betekent dat de monteur om hulp gaat vragen bij andere monteurs. Als het probleem hierdoor opgelost kan worden mag de Andonlight weer op groen. Kan het probleem hierdoor niet opgelost worden dat moet de monteur de Andonlight op rood te zetten. Hierdoor kan de monteur hulp vragen van een ondersteunende afdeling. Op de betreffende ondersteunende afdeling kan dit gezien worden op een groot scherm. Ook wordt er een waarschuwingsgeluid gegeven. Een medewerker van de ondersteunende afdeling loopt naar de monteur toe. Als het probleem is behandeld is moet de monteur de Andonlight weer op groen zetten. Het kan soms voorkomen dat de Andonlight direct van groen op rood wordt gezet. Een voorbeeld is als er onderdelen besteld moeten worden, (andere monteurs kunnen de monteur hier niet bij helpen).

Dit is hoe de Andonlights op dit moment bij EPCOR B.V. werken. Het Andonlight Systeem wordt nu alleen gebruikt als communicatiemiddel. Er worden geen gegevens opgeslagen. EPCOR B.V. wil graag het gebruik en de effectiviteit van het Andonlight Systeem kunnen meten en analyseren. Ook wil EPCOR B.V. dat de gebruiksvriendelijkheid van het Andonlight Systeem wordt verbeterd voor 'key-users'. Onder de 'key-users' vallen de medewerkers van het primaire proces en de medewerkers van de ondersteunende processen die direct gebruik maken van het Systeem, maar ook de medewerkers die van de opgeslagen gegevens gebruik gaan maken. Uit de bovengenoemde wensen is de opdracht ontstaan.

## 1.2. Projectprobleemstellingen

Hoe kan bij EPCOR B.V. door het vernieuwen van het Andonlight Systeem de gebruiksvriendelijkheid verbeterd worden voor de key-users aan de hand van hun voorkeuren en hoe kan het Systeem data opslaan zodat deze geanalyseerd kan worden, zodat op de bevindingen die hieruit voortkomen geanticipeerd kan worden?

### 1.3. Projectdoelstellingen

EPCOR B.V. wil het Andonlight Systeem met de onderstaande punten verbeteren:

- De mogelijkheid om de data op te slaan en terug te zien, zodat historische data beschikbaar wordt en hierop geanticipeerd kan worden.
- Verhogen van de gebruiksvriendelijkheid voor key-users aan de hand van hun voorkeuren.

### 1.4. Business Case

De Business Case verschaft informatie over het bestaansrecht van het Project.

Redenen voor het Project

EPCOR B.V. ontvangt haar winst uit het primaire proces. Ieder onderdeel bij EPCOR B.V. waar handelingen op worden verricht, heeft een bepaalde 'Turn around Time'. Dit is de tijd tussen het ontvangen van het onderdeel tot het terugzenden hiervan naar de klant. Momenten waarbij het proces stil komt te staan, omdat op de klant gewacht moet worden, mogen uiteraard van de TaT afgehaald worden.

Verkorting van de TaT betekent voor EPCOR B.V. een betere service aan de klant, minder kosten aan bijvoorbeeld uurloon van de monteurs en een betere positie ten opzichte van de concurrent.

Iedere keer als een monteur een probleem heeft die hij/zij niet kan oplossen loopt de TaT door. Het is dus heel erg van belang dat de ondersteunende afdelingen snel reageren en het probleem op de juiste manier oplossen.

Een Andonlight Systeem dat goed is afgestemd op de gebruikers kan hieraan bijdragen. Ook kunnen bepaalde structurele problemen waarvoor een Andonlight aangezet wordt vastgesteld worden en kan er ook een structurele oplossing gezocht worden.

Er zullen geen voorspellingen gedaan worden over hoeveel een vernieuwd Systeem de TaT omlaag zal brengen. Hiervoor is bij de aanvang van het Project onvoldoende data beschikbaar.

Wel zal er een nulmeting worden gedaan.

### 1.5. Context

EPCOR B.V. is 100% dochter van Air France/KLM. Er zijn ongeveer 100 mensen werkzaam. De processen binnen EPCOR B.V. worden door moderne IT-systemen aangestuurd en na

bewerking worden de componenten en APU's getest en luchtwaardig verklaard. EPCOR B.V. werkt onder strikte kwaliteitsnormen van de internationale luchtvaartautoriteiten.

Ik ben als stagiair werkzaam op de afdeling Finance/IT die zich naast Finance bezig met de IT inrichting van EPCOR B.V. De afdeling Finance/IT wordt ondersteund door een externe IT-dienstverlener.

Het primaire proces van EPCOR B.V. is zoals eerder vermeld reparatie, revisie en testen van pneumatisch aangedreven componenten en APU's (Auxiliary Power Unit). Dit proces vindt plaats in de 'Shop' afdeling.

Verder zijn er voor het primaire proces ondersteunende processen. De ondesteunende processen worden door de volgende afdelingen uitgevoerd: QA (Quality Assurance), MP (Material Planning), Logistiek, Magazijn, HRM, Finance/IT, Sales en CAR (Customer Account Representatives) en Management.

Binnen EPCOR B.V. wordt gebruik gemaakt van het een ERP pakket genaamd Quantum Controle. Dit is een ERP pakket specifiek bedoeld voor de luchtvaart.

## 1.6. Scope

De scope wordt gedefinieerd als het geheel van op te leveren producten inclusief subproducten en de eisen waaraan deze moeten voldoen en de activiteiten die uitgevoerd moeten worden om de producten te behalen.

Binnen de scope van dit Project valt:

Alles wat in de planning en randvoorwaarden is opgenomen

Buiten de scope van dit Project valt alles wat niet is genoemd, waaronder:

- Medewerkers trainen/motiveren het Systeem op de juiste manier te gebruiken;
- Ontwikkelen van software die geïntegreerd is in het Quantum Controle ERP pakket.

Een vorige stagiair heeft het gebruik van het Andonlight Systeem al onderzocht.

Ik zal haar bevindingen waar mogelijk gebruiken bij dit Project.

## 1.7. Randvoorwaarden

### 1.7.1. Randvoorwaarden Kwaliteit

Er zal een nulmeting uitgevoerd worden. Dit kan naderhand als vergelijkingsmateriaal gebruikt worden.

Op verschillende momenten zal met verschillende belanghebbenden de voortgang van het Project besproken worden. In de planning kunt u hierover meer vinden.

### 1.7.2. Randvoorwaarden Tijd

Het adviserende deel van het Project moet uiterlijk 3 juli 2012 afgerond zijn. Ik zal 3 dagen per week gemiddeld 8 uur per dag om locatie bij EPCOR B.V. aan het Project werken en één dag van 8 uur thuis.



## 1.8. Projectrisico's

Risico	Omschrijving	Oorzaak	Kans	Impact	Beïnvloed doelstelling
Tijdnood	De deadline van het Project staat vast.	Slechte planning of onverwacht problemen of onduidelijk/onvoldoen de afgebakende scope. Vastgestelde deadline	gemiddeld	Onvolledig, of kwalitatief slecht eindproduct , of vertraging	ja
Communicatieproblemen	Onvoldoende of kwalitatief slechte communicatie met belanghebbenden	Slechte planning. Belanghebbende heeft weinig tijd of ziet belang van Project niet in.	gemiddeld	Onvolledig, of kwalitatief slecht eindproduct , of vertraging	ja
Ontwikkeluitdagingen	Programmeer- of modelleeruitdagingen	Onvoldoende kennis en ervaring. Verkeerde keus van ontwikkeltaal, ontwikkel omgeving of ontwikkelmethode	gemiddeld	kwalitatief slecht eindproduct , of vertraging, of er moeten concessies gedaan worden	ja
Eisen van externen	De opdrachtgever wil graag dat de data wordt opgeslagen in de huidige database van het Quantum Controle ERP pakket.	De leverancier van het ERP pakket heeft bepaalde eisen aan de database	hoog	Er moeten concessies gedaan worden	ja

Hierboven is een overzicht van de risico's die ik heb vastgesteld en beoordeeld. Om de risico's te reduceren of zelfs te vermijden heb ik een aantal maatregelen getroffen:

1. Om te beginnen wordt het Project afgebakend door de geformuleerde doelstelling, de opgestelde planning per product en activiteit en de opgestelde scope.
2. Voor de communicatie met de belanghebbenden is een planning gemaakt en alle gesprekken worden voorbereid met vragen en er worden aantekeningen gemaakt.
3. Voor de ontwikkeluitdagingen heb ik gekozen voor OpenUP als softwareontwikkelingsproces om de Systeemontwikkeling zoveel mogelijk te beheersen.
4. Tot slot is er contact gelegd met de leverancier van het ERP pakket, zodat de mogelijkheden in kaart gebracht kunnen worden. Het resultaat hiervan wordt vervolgens vastgelegd in de scope en planning.

## 2. PROJECTAANPAK

In antwoord op de in hoofdstuk één omschreven Projectdefinitie treft u in dit hoofdstuk een voorstel voor de aanpak, fasering en planning aan.

### 2.1. Hoofdlijnen aanpak

Voordat met het ontwerpen begonnen kan worden, moeten de voor het Project relevante bedrijfsprocessen in kaart worden gebracht, geanalyseerd en wellicht aangepast. Ook zijn er aardig wat afdelingen en mensen betrokken bij dit Project. Daarom is gekozen voor PRINCE2 als Projectmanagementmethode. De PRINCE2 staat los van de inhoud van het Project. De PRINCE2 methode zal het Project in het algemeen structuur geven.

Voor het ontwerp van de database en het Systeem is gekozen voor de OpenUP. OpenUP is een iteratief softwareontwikkelingsproces. Deze methode biedt uitgebreide mogelijkheden voor het uitvoeren en beheersen van het softwareontwikkelingsproces en onderscheidt de fasen: Inception, Elaboration, Construction en Transition.

Voor het elimineren van verspilling in het productieproces wordt binnen EPCOR B.V. gebruik gemaakt van Lean Six Sigma. Er zijn binnen EPCOR B.V. verschillende mensen die Lean Six Sigma expert zijn. In dit Project zal Lean Six Sigma ook gebruikt worden.

### 2.2. Fasering en besluitmomenten

In deze paragraaf treft u een overzicht van de fasen en producten aan.

Ik onderscheid de volgende fasen:

- Opstartfase
- Initiatiefase
- Analysefase
- Ontwikkelingsfase
  - Elaboration
  - Construction
  - Transition
- Afrondfase

#### 2.2.1. De Opstartfase

In deze fase wordt het afstudeervoorstel opgesteld. Hiermee wordt bepaald of het Project gestart kan worden.

### **2.2.2. Initiatiefase**

In deze fase wordt het Plan van Aanpak opgesteld. Hierin wordt meer in detail dan het afstudeervoorstel beschreven wat het Project inhoudt en hoe het Project uitgevoerd en beheerst gaat worden.

### **2.2.3. Analysefase**

In deze fase worden de voor het Project relevante processen en het huidige Systeem in kaart gebracht. Ook zal er een nulmeting uitgevoerd worden.

De processen zullen in kaart gebracht worden door het opstellen van een Business Process Model. Het BPM bestaat uit business objecten en business processen. De business processen worden gedefinieerd met behulp van de Business Process Model and Notation, een standaard voor proces modellering. Een BPM met behulp van BPMN is visueel en is goed te begrijpen door alle belanghebbenden.

Er zal een nulmeting uitgevoerd worden, zodat er vergelijkingsmateriaal is. Dit zal gebeuren door tijdelijk een server in de richtten die het dashboard van het Systeem door de tijd heen zal vastleggen in video bestanden, waar vervolgens data uitgehaald kan worden.

Aan het einde van deze fase zal een Business Process advies rapport opgesteld worden.

Hierin komen de verbeterpunten van de processen en welke hiervan in het Systeem geïmplementeerd kunnen worden.

### **2.2.4. Ontwikkelingsfase**

Nadat de relevante onderdelen van de onderneming geanalyseerd zijn, kan gestart worden met de ontwikkelingsfase.

Hiervoor zal OpenUP gebruikt worden. Dit is een ontwikkelingsproces gebaseerd op RUP.

Met iteraties wordt incrementeel een eindproduct opgebouwd. In iedere iteraties kan op bepaalde onderdelen gefocust worden. Als bijvoorbeeld alleen een ontwerp gemaakt wordt kunnen de implementatiedelen van de iteraties weggelaten worden.

De reden dat voor OpenUP gekozen is dat het is toegespitst op kleine Projecten. Het geeft structuur en de resultaten zijn makkelijk overdraagbaar. Als bijvoorbeeld de implementatie in een ander Project uitgevoerd zou worden, dan kunnen hiervoor extra iteraties gemaakt worden worden.

Voor het ontwerp zal één iteratie doorlopen worden. Dit heeft te maken met de grootte van de Project opdracht en de hoeveelheid personen die het Project uitvoeren (één persoon).

OpenUP onderscheidt een aantal fasen. De fasen die doorlopen gaan worden, worden hieronder besproken.

#### **2.2.4.1. Elaboration**

In deze fase wordt gestart met het Systeemontwerp. Bij het opstellen van een reeks producten zal gaandeweg steeds duidelijker wordt hoe het Systeem eruit gaat zien. In een Datamodel zal weergegeven worden welke data door het Systeem opgeslagen en gebruikt gaat worden.

In Use Cases zal voor de belanghebbende duidelijk worden welke actoren welke handelingen met het Systeem kunnen uitvoeren om bepaalde doelen te bereiken.

In het Database Model wordt het Datamodel verder uitgewerkt.

In het Software Architectuur Document worden met verschillende 'Views' verschillende aspecten van de architectuur weergegeven: Use Case, Logisch, Implementatie, Deployment.

#### **2.2.4.2. Construction**

In deze fase wordt het ontwerp op klasse en functie niveau verder uitgewerkt. De producten zijn klasse- en sequence diagrammen. Vervolgens wordt alles bij elkaar gezet in een Functioneel Ontwerp. Hierna begint het echte ontwikkelen van de Client en Server. Hiervoor moet een webserver opgezet worden. Vervolgens zal de code worden geschreven. Hierbij wordt onderscheid gemaakt zitten drie code. De 'Data-Layer' bevat de klassen die toegang tot de data verschaffen. De 'Domain-Layer' bevat de klassen die uitvoeren wat door de 'User-Interface' klassen gevraagd wordt. De 'UserInterface-Layer' bevat de klassen die de communicatie met de gebruiker mogelijk maken.

#### **2.2.4.3. Transition**

Als de code van het eindproduct gereed is wordt een Systeemtest uitgevoerd. Een Systeemtest is een test van het Systeem waarbij gekeken wordt of het voldoet aan de eisen.

Als het Systeem voldoet kan de uitrol van Client en Server uitgevoerd worden en er wordt een gebruikershandleiding opgesteld.

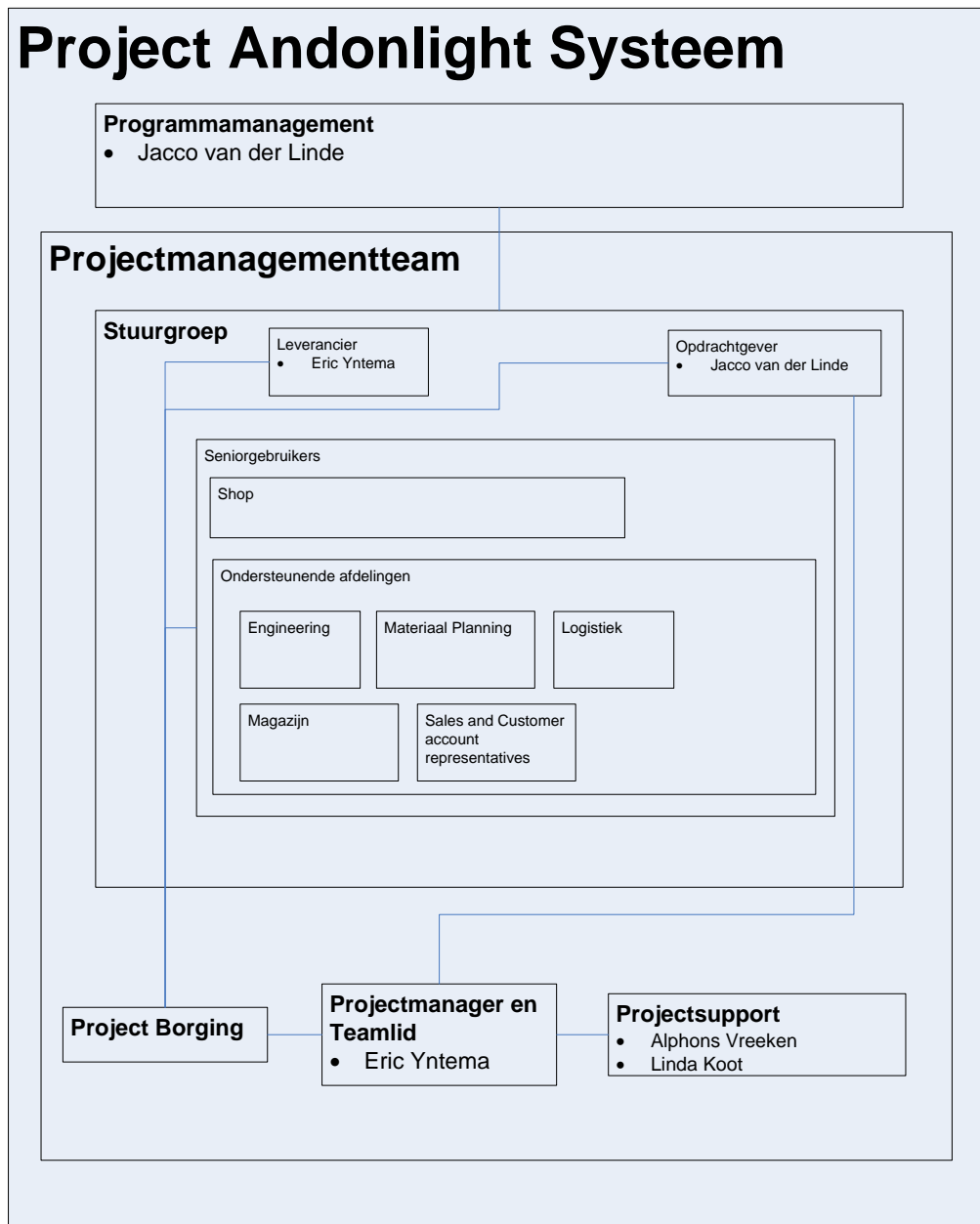
#### **2.2.5. Afrondfase**

In deze fase wordt het Project geëvalueerd en eventueel aangepast en wordt de afstudeerscriptie opgesteld en ingeleverd.

### **2.3. Projectorganisatie**

In deze paragraaf wordt de Projectorganisatie in kaart gebracht die van belang is voor het Project.

### 2.3.1. Organisatieschema



### 2.3.2. Rollen en rolverdeling

Voor een succesvol Project is het volgens PRINCE2 essentieel dat de volgende partijen betrokken zijn bij de aansturing van het Project: gebruikers, het bedrijf en de leverancier. Hierdoor worden de belangen van de verschillende partijen geborgd. De belangrijkste vertegenwoordigers van deze partijen vormen samen de stuurgroep. Binnen de stuurgroep wordt onderscheid gemaakt tussen de volgende rollen: opdrachtgever, seniorgebruikers en de leverancier.

Jacco van der Linde (Finance Director) heeft dit Project goedgekeurd bij EPCOR B.V. Hij is ook de opdrachtgever binnen het Projectmanagementteam.

De seniorgebruikers zijn de monteurs in het primaire proces in de Shop. Verder zijn er een aantal afdelingen die het primaire proces ondersteunen: Engineering, Materiaal Planning, Logistiek, Magazijn en Sales and Customer account representatives.

Ik heb binnen de stuurgroep de rol van leverancier, verder heb ik de dagelijkse leiding over het Project en zal ik ook het Project uitvoeren.

Tot slot zijn er nog een aantal andere belangrijke personen. Alphons Vreeken is de software ontwikkelaar van het huidige Andonlight Systeem. Linda Koot is expert op het gebied van Lean Six Sigma.

### 2.3.3. Overlegvormen en frequenties

Om de voortgang te bewaken en om bekend te worden met de bedrijfsprocessen en het Systeem zijn er verschillende momenten ingepland waarop ik met medewerkers in gesprek ga:

Persoon/afdeling	frequentie	overlegsoort	doelstelling
Jacco van der Linde	2 wekelijks	bespreking	bespreken lopende zaken en voortgang
Rainish Lalai	wekelijks	bespreking	bespreken lopende zaken en voortgang
Yolanda de Jong (Shop manager)	2 wekelijks	bespreking	bespreken lopende zaken en voortgang
Monteurs	incidenteel	meelopen	bekend worden met processen en gebruik Andonlight Systeem
Ondersteunende afdelingen	incidenteel	meelopen	bekend worden met processen en gebruik Andonlight Systeem
Linda Koot	eenmalig	uitleg	uitleg Lean Six Sigma
Stuurgroep	incidenteel	bespreking	voortgang

## 2.4. Projectplanningen

In deze paragraaf treft u de planningen aan, gerelateerd aan de hierboven beschreven producten.

### 2.4.1. Uitgangspunten

Ik besteed 32 uur per week aan dit Project. Waarvan ik 3 keer 8 uur op locatie bij EPCOR B.V. ben. De resterende 8 uur werk ik thuis.

### 2.4.2. Productplanning

De planning wordt in een zogenaamd balkenschema bijgehouden. Hierin wordt op de dag nauwkeurig bijgehouden welke activiteiten worden uitgevoerd en aan welke producten gewerkt gaat worden.

In de planning zal zoveel als mogelijk uitlooptijd worden opgenomen.

In de bijlage kunt u het balkenschema terug vinden.



### **3. PROJECTBEHEERSING**

Dit hoofdstuk beschrijft op hoofdlijnen de opzet en invulling van de beheersing van het Project.

#### **3.1. Beheersing van de Scope**

Er kunnen altijd onverwachtse veranderingen in de Scope ontstaan. Als er veranderingen zijn in de Scope dienen alle belanghebbenden hiervan tijdig op de hoogte gesteld worden. Ook moet de planning eventueel aangepast en beoordeeld worden.

#### **3.2. Beheersing van Tijd en Geld**

Binnen de beschikbare tijd moet rekening gehouden worden met voldoende uitloop. Er is geen budget voor dit Project. Mochten er extra kosten ontstaan (bijvoorbeeld door de aanschaf van maatwerk software), dan gaat dit in overleg met de opdrachtgever.

#### **3.3. Beheersing van Kwaliteit**

In de planning wordt per product bijgehouden of het product geaccepteerd is door de opdrachtgever. Mocht het product niet direct geaccepteerd worden dan dient deze aangepast te worden tot deze wel geaccepteerd wordt.

Behalve het opleveren van de producten volgens de planning wordt aan het einde van het Project in de evaluatie gekeken in hoeverre de opdrachtgever tevreden is met het resultaat.

### **DEFINITIELIJST**

#### **Systeem**

Het Andonlight Systeem

#### **Processen**

De voor het Systeem relevante processen

#### **Project**

Het Project inrichten Andonlight Systeem

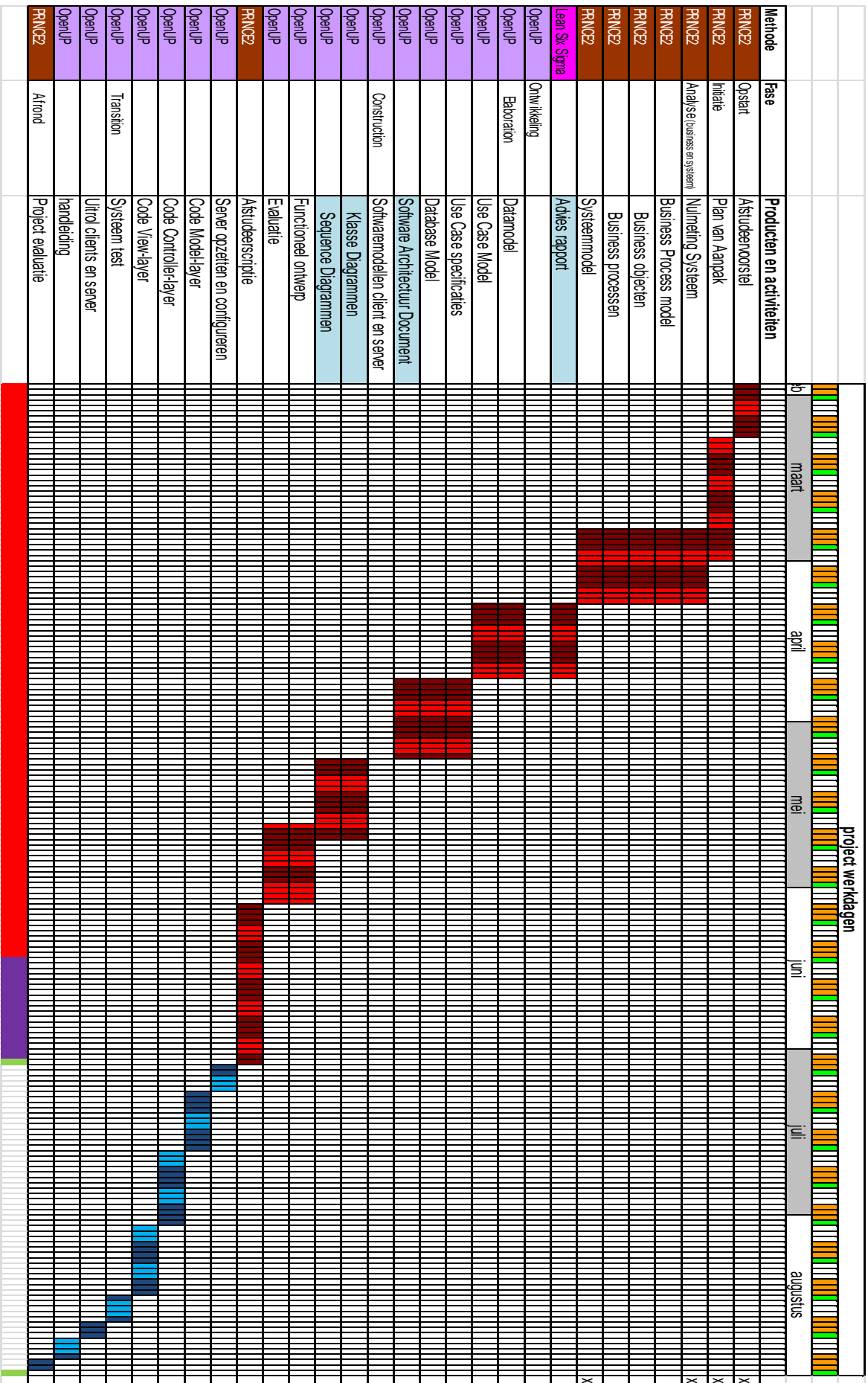
#### **Key-users**

Onder de 'key-users' vallen de medewerkers van het primaire proces en de medewerkers van de ondersteunende processen, die direct gebruik maken van het Systeem, maar ook de medewerkers die van de opgeslagen gegevens gebruik gaan maken.

### **Gebruiksvriendelijkheid**

Alles waardoor de gebruiker van het Systeem zijn werk beter kan uitvoeren, waarbij rekening wordt gehouden met de principes van Lean Six Sigma

## **BIJLAGEN**



## **Bijlage B: evaluatie eigen functioneren**

Ik ben achteraf blij dat ik voor dit project heb gekozen. Het is een duidelijk afgebakend project dat voldoende uitdaging biedt. Ik heb met veel plezier aan de opdracht gewerkt tot nu toe. Het leukste onderdeel moet nog komen, namelijk: het ontwikkelen en implementeren van het nieuwe systeem.

Ik heb veel geleerd, onder anderen hoe een project goed georganiseerd moet worden en hoe de juiste oplossing gevonden kan worden voor een bepaald software ontwikkelingsprobleem. Ook heb ik vrij zelfstandig moeten werken, omdat bij Epcor B.V. geen interne ICT afdeling bestaat, waar vragen gesteld kunnen worden.

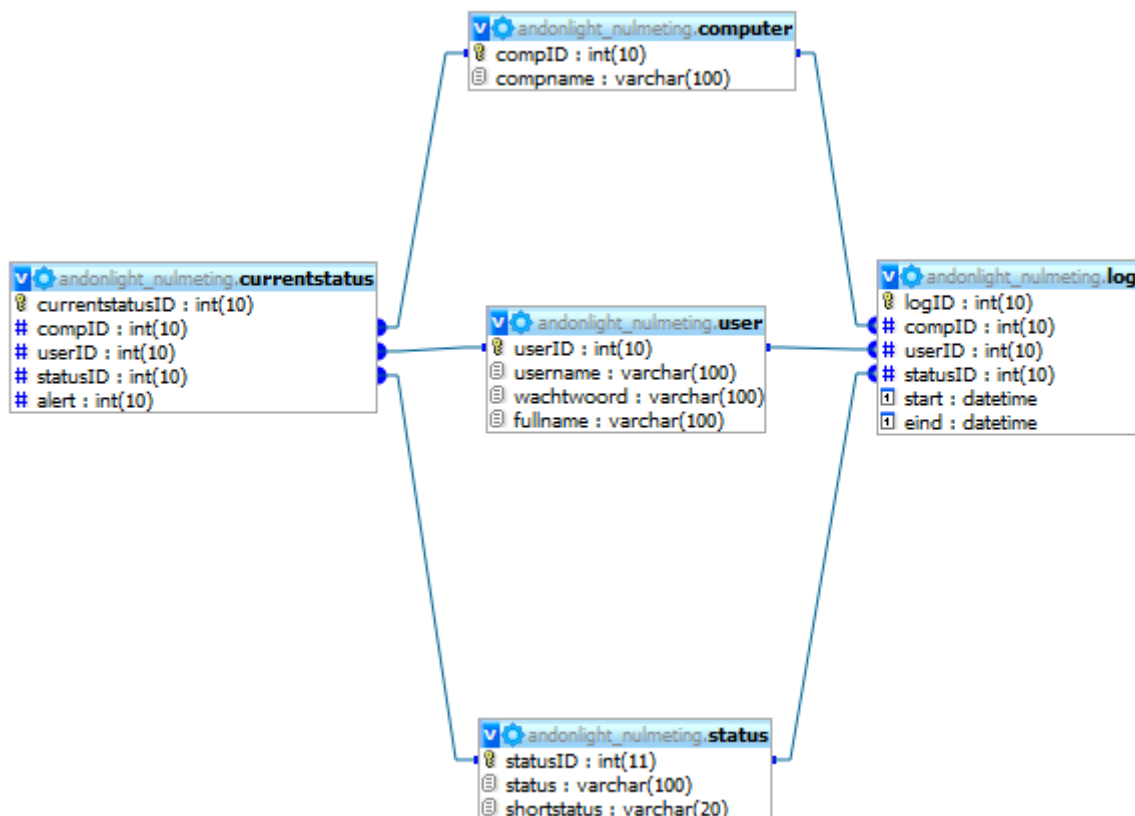
## Bijlage C: resultaten nulmeting

### Nulmeting data

#### Oude Andonlight database

De oude Andonlight database blijkt veel nuttige informatie te bevatten. Er is iedere werkdag gelogd beginnende op 2010-12-01 en eindigend op 2011-07-11. In de logtabel staan bijna 10.000 records. De database structuur is weergegeven in figuur 0.1 en bestaat uit de volgende tabellen:

- Currentstatus
- Computer
- User
- Status
- Log



Figuur 0.1 database structuur

#### Kwaliteit van de data

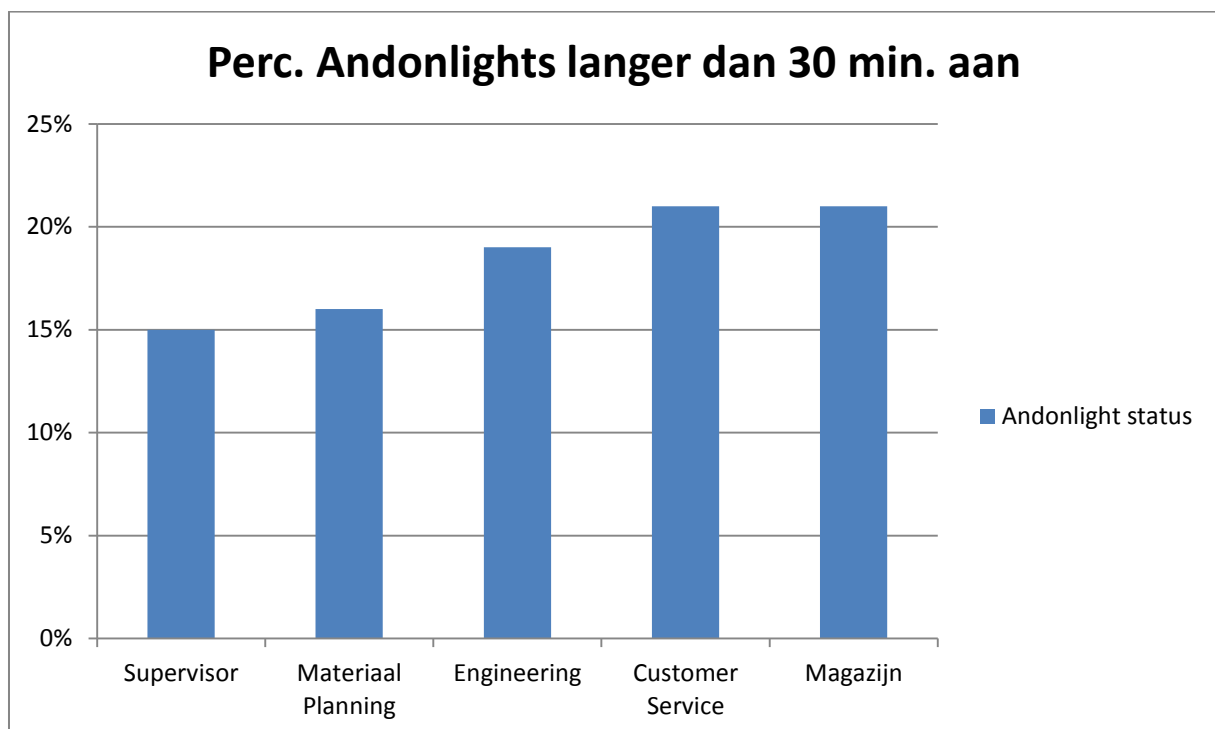
Er zijn een aantal records in de logtabel die alleen een begintijd hebben. Mijn ervaring is dat dit gebeurt als de Andonlight status niet wordt gewijzigd en de computer van de monteur wordt afgesloten. Hierdoor krijgt de Andonlight client niet de mogelijkheid om de eindtijd te loggen. De records zijn willekeurig verdeeld door de tijd heen. Records zonder eindtijd zijn niet bruikbaar.

Verder zijn er drie records waarvan de logtijden niet kloppen, die hieronder worden weergegeven. Er is langer gelogd dan één normale werkdag. Ik vermoed daarom dat er iets mis is gegaan, waardoor de Andonlight niet uitgezet is. Deze records zijn ook niet bruikbaar.

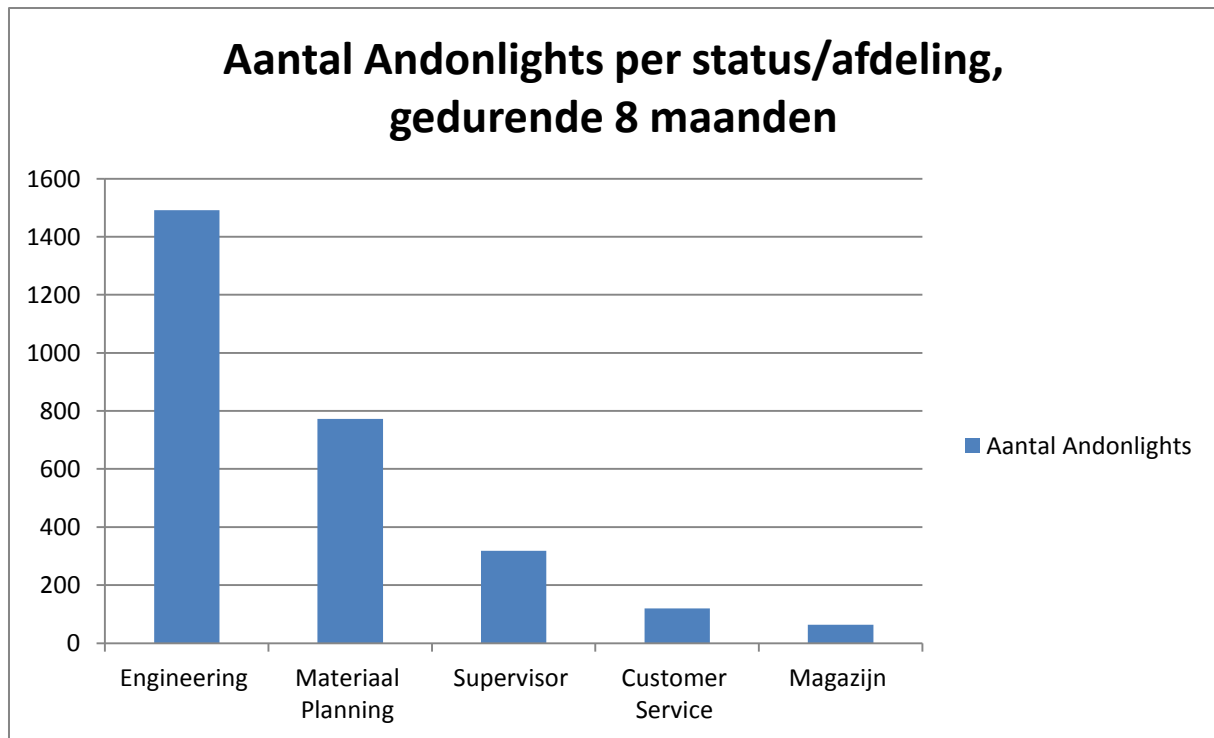
start	eind	tijd	status
9-2-2011 10:59	10-2-2011 6:28	19:28	Supervisor
16-2-2011 5:38	17-2-2011 7:53	26:15	Engineering
31-5-2011 12:51	31-5-2011 12:07	11:16	Engineering

## Informatie uit de data

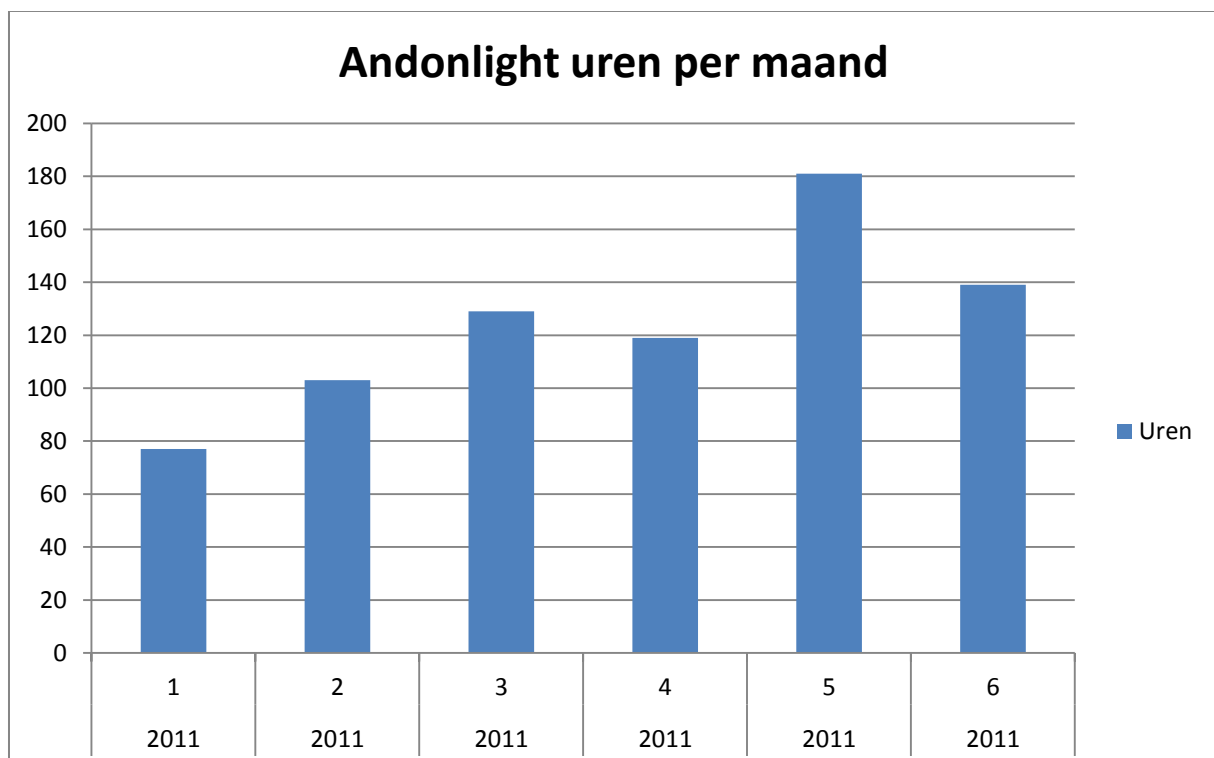
In figuren 0.2 tot en met 0.5 worden voorbeelden gegeven van informatie die uit de data te halen is.



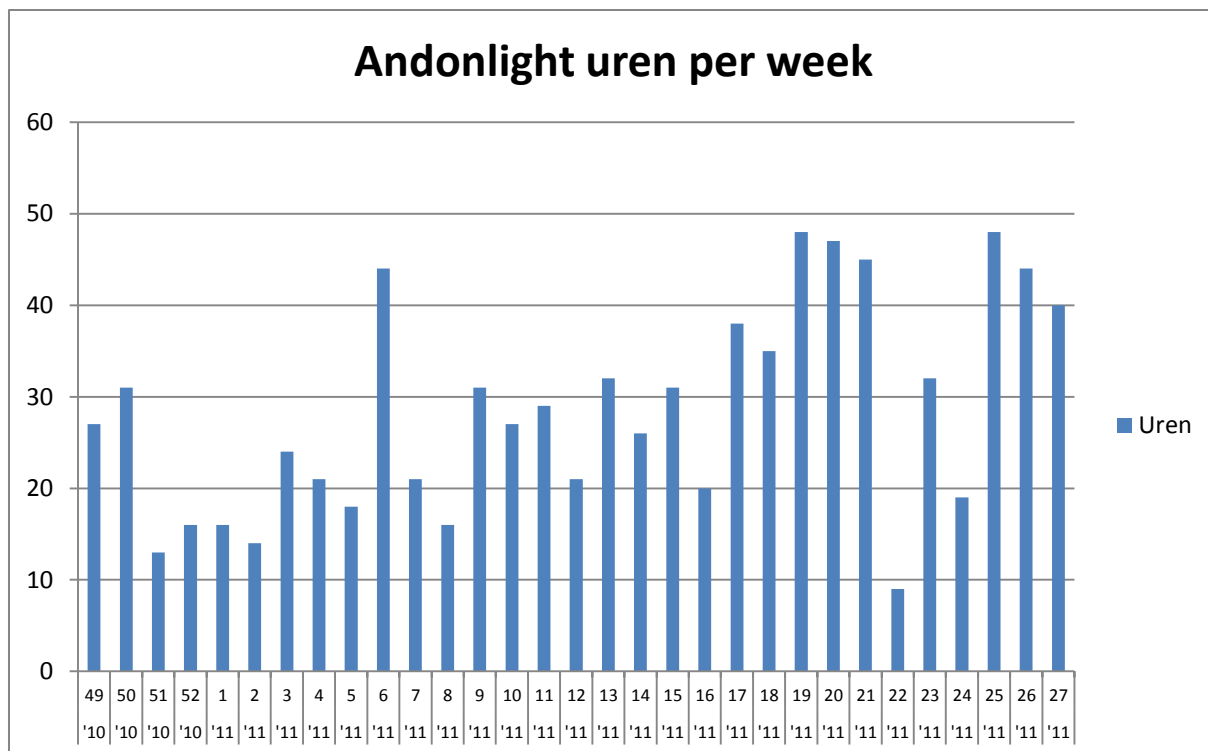
Figuur 0.2 percentage Andonlights die langer dan 30 minuten op rood staan per afdeling



Figuur 0.3 Aantal Andonlight per status/afdeling



Figuur 0.4 Andonlight user per maand



**Figuur 0.5 Andonlight uren per week**

Opmerking: de eerste en laatste maand en week respectievelijk in figuur 0.5 zijn niet meegenomen in de laatste twee grafieken, omdat deze niet helemaal gevuld zijn.