



# ProjectenMonitor

Een software ontwikkeltraject voor de provincie Noord-Brabant

|                           |                                       |
|---------------------------|---------------------------------------|
| <b><i>Student</i></b>     | <b><i>Ruud Aarts 2011985</i></b>      |
| <b><i>Organisatie</i></b> | <b><i>provincie Noord-Brabant</i></b> |
| <b><i>Begeleider</i></b>  | <b><i>Onno Vollinga</i></b>           |
| <b><i>Docent</i></b>      | <b><i>Frans Meulenbroeks</i></b>      |

***Augustus 2012***

## **Dankwoord**

Het is eind 2004 en ik zit aan tafel tegenover Marion Miezenbeek. Zij is coach van mijn outplacementtraject. 'Wat wil je nu echt doen?', vraagt ze me op een vriendelijke, maar toch wel enigszins doordringende wijze. Met deze vraag sluit ze een hoofdstuk af van jaren in verschillende functies die ik geen van allen ambieerde.

Een klein half jaar nadien start ik als onbetaalde stagiair bij de provincie Noord-Brabant. Als eind dat jaar blijkt dat er geen werk voor me is binnen ict-beheer krijg ik een aanstelling bij de documentaire informatie voorziening. Ondertussen ben ik datzelfde jaar gestart met de opleiding Informatica aan Fontys Hogeschool in Eindhoven. Ik kan nog geen letter programmeren en niets, maar dan ook niets kan me nog weerhouden.

'Het zijn er twee!', was de heugelijke uitroep van de gynaecoloog als hij tegen de kerstdagen de sonde over de buik van mijn vrouw beweegt. Zeven maanden later ben ik vader van vier kinderen. Ik heb een baan en een gezin, wat kan ik me nog meer wensen? Een opleiding natuurlijk! In het tweede collegejaar ga ik gebukt onder twee modules en het resultaat laat zich raden. Ik besluit een tandje terug te schakelen.

Het is nu 04 augustus 2012 en werk nu al bijna vier jaar aan de afdeling technisch beheer bij de provincie en leg de laatste hand aan deze scriptie. Ik wil de provincie en mijn collega's bedanken, omdat ze me de laatste zeven jaar alle ruimte hebben gegeven om deze opleiding te kunnen voltooien. Echter gaat mijn grootste dank uit naar mijn vrouw die me altijd door dik en dun is blijven steunen.

Ruud Aarts

## Inhoudsopgave

|                                                            |           |
|------------------------------------------------------------|-----------|
| <b><i>Inhoudsopgave</i></b>                                | <b>2</b>  |
| <b><i>Samenvatting</i></b>                                 | <b>5</b>  |
| <b><i>Summary</i></b>                                      | <b>6</b>  |
| <b><i>1 Inleiding</i></b>                                  | <b>7</b>  |
| <b><i>1.1 Inleiding</i></b>                                | <b>8</b>  |
| <b><i>2 De organisatie provincie Noord-Brabant</i></b>     | <b>9</b>  |
| <b><i>2.1 Een overzicht van de organisatie</i></b>         | <b>10</b> |
| 2.1.1 Positionering binnen de Overheid                     | 10        |
| <b><i>2.2 Politiek en Bestuur</i></b>                      | <b>10</b> |
| 2.2.1 Commissaris van de Koningin                          | 10        |
| 2.2.2 Provinciale Staten                                   | 10        |
| 2.2.3 Gedeputeerde Staten                                  | 10        |
| 2.2.4 Organisatiestructuur                                 | 10        |
| 2.2.5 Bureaus                                              | 11        |
| 2.2.6 Organogram                                           | 11        |
| 2.2.7 Het Informatie Kern Team (IKT)                       | 12        |
| <b><i>2.3 Projecten en Procedure</i></b>                   | <b>13</b> |
| <b><i>3 Probleemstelling, conclusies en opdracht</i></b>   | <b>14</b> |
| <b><i>3.1 Het Informatie Kern Team wil uitdragen</i></b>   | <b>15</b> |
| 3.1.1 Probleemstelling                                     | 15        |
| 3.1.2 Doelstelling                                         | 16        |
| <b><i>3.2 Opdrachtverfijning</i></b>                       | <b>16</b> |
| <b><i>3.3 Onderzoek</i></b>                                | <b>17</b> |
| 3.3.1 Vraagstelling                                        | 17        |
| 3.3.2 Deelvraag                                            | 17        |
| 3.3.3 Conclusies                                           | 17        |
| 3.3.4 Aanbevelingen                                        | 18        |
| <b><i>3.4 De opdracht voor uitvoering</i></b>              | <b>19</b> |
| 3.4.1 Doelstelling                                         | 19        |
| 3.4.2 Scope                                                | 19        |
| <b><i>4 Het Projectplan voor de 'ProjectenMonitor'</i></b> | <b>21</b> |

|                                                                                   |           |
|-----------------------------------------------------------------------------------|-----------|
| <b>4.1 Fasering</b>                                                               | <b>22</b> |
| <b>4.2 De initiatiefase</b>                                                       | <b>23</b> |
| 4.2.1 Werkzaamheden                                                               | 23        |
| 4.2.2 Projectmedewerkers                                                          | 24        |
| 4.2.3 Communicatie                                                                | 24        |
| <b>4.3 Ontwerpfase</b>                                                            | <b>25</b> |
| 4.3.1 Functioneel ontwerp                                                         | 25        |
| 4.3.2 Grafische Interface                                                         | 25        |
| 4.3.3 Technisch Ontwerp                                                           | 26        |
| <b>4.4 De Implementatiefase</b>                                                   | <b>27</b> |
| <b>4.5 Testfase van het Systeem</b>                                               | <b>28</b> |
| <b>4.6 Oplevering en in Productie name van het systeem</b>                        | <b>29</b> |
| <b>5 Projectuitvoering</b>                                                        | <b>30</b> |
| <b>5.1 Opstartfase, formatie Plan van Aanpak</b>                                  | <b>31</b> |
| 5.1.1 Plan van aanpak                                                             | 31        |
| 5.1.2 Samenstellen Projectfasering                                                | 31        |
| 5.1.3 DSDM                                                                        | 32        |
| 5.1.4 Tijdsbesteding en achtervang                                                | 32        |
| <b>5.2 Ontwerpfase</b>                                                            | <b>33</b> |
| 5.2.1 Functioneel Ontwerp                                                         | 33        |
| 5.2.2 Gegevensverzameling over het IKT-proces                                     | 33        |
| 5.2.3 Grafische Gebruikers Interface                                              | 34        |
| 5.2.4 MoSCoW                                                                      | 36        |
| 5.2.5 Technisch Ontwerp                                                           | 37        |
| 5.2.6 Ontwerp Database                                                            | 37        |
| 5.2.7 Ontwerp van server en client                                                | 40        |
| <b>5.3 Implementatiefase</b>                                                      | <b>43</b> |
| 5.3.1 Inleiding                                                                   | 43        |
| 5.3.2 Implementatie van de database                                               | 43        |
| 5.3.3 Bouw van server- en clientapplicaties                                       | 44        |
| <b>5.4 Stand van Zaken</b>                                                        | <b>47</b> |
| <b>5.5 Wat moet er nog gebeuren?</b>                                              | <b>47</b> |
| <b>6 Reflectie</b>                                                                | <b>48</b> |
| <b>6.1 Reflectie op mijn rol als projectleider en de provinciale organisatie.</b> | <b>49</b> |
| <b>7 Verklarende Woordenlijst</b>                                                 | <b>50</b> |

|          |                                                          |           |
|----------|----------------------------------------------------------|-----------|
| <b>8</b> | <b><i>Literatuurlijst en/of bronvermelding</i></b>       | <b>56</b> |
| 8.1      | Literatuur en/of bronvermelding                          | 57        |
| <b>9</b> | <b><i>Bijlagen</i></b>                                   | <b>58</b> |
| 9.1      | Bouw van een model Grafische Interface in de ontwerpfase | 59        |
| 9.2      | Plan van Aanpak                                          | 61        |

## Samenvatting

In januari 2012 is er een start gemaakt met een software ontwikkeltraject binnen de provincie Noord-Brabant in opdracht van het Informatie Kern Team (IKT). Het IKT is autonoom gepositioneerd binnen de organisatie. Deze autonomie heeft het team nodig in de rol die het bekleedt. Het IKT onderhoudt een procedure om ict-gerelateerde projecten in een aantal stappen te keuren voorafgaand aan de eigenlijke projectuitvoering. De IKT-procedure en de voortgang ervan was altijd onzichtbaar voor de buitenwereld. Er werd dan ook gezocht naar een oplossing om de zichtbaarheid te vergroten. Het Informatie Kern Team stelt voor om een bestaande applicatie, genaamd kredietcrisimonitor, in te richten met de eisen en wensen die van toepassing zijn op de IKT-procedure.

**Vraagstelling** Het IKT heeft gevraagd om te onderzoeken op welke manier de IKT-procedure kan worden gepresenteerd. Kan er gebruik worden gemaakt van bestaande software of wordt er gestart met een nieuw ontwikkeltraject.

**Onderzoek** Al snel kwamen er vanuit het onderzoek naar de achtergrond en kwaliteit van de bestaande applicatie grote risico's naar voren. Een daarvan was het ontbreken van documentatie. Waardoor er geen inschatting kan worden gemaakt over de mogelijkheden en kwaliteit van de applicatie. Dit zou tot ernstige projectvertraging kunnen leiden.

**Advies** Het uiteindelijke advies daarop was om geen gebruik te maken van de bestaande software. De opdrachtgever heeft dit advies overgenomen en vanuit hier is een projectopdracht samengesteld.

**Opdracht** De opdracht houdt in dat er een systeem wordt ontwikkeld waar gegevens aangaande de IKT-procedure kunnen worden verwerkt, opgeslagen door IKT-medewerkers en gepresenteerd aan belanghebbenden en de rest van de organisatie. De scope van deze opdracht gaat niet verder dan het ontwikkelen van de invoermodule en de database. De ontwikkeling van de presentatiemodule ligt daar buiten. Met deze opdracht ben ik aan de slag gegaan. In een plan van aanpak is de projectfasering uiteengezet. Na afronding van deze initiatiefase is het systeemontwerp opgemaakt in een functioneel en een technisch ontwerp. Op dit moment loopt de eerste iteratie van de implementatiefase van het project. Deze verloopt zeker niet zonder slag of stoot. In de tweede iteratie van de implementatie treedt ook de testfase in. Er wordt dan gewerkt aan fouten en verfijning van de applicaties. De verwachting is dat de opdracht eind oktober wordt afrondt.

## Summary

In January 2012 a start was made with a software engineering project within the province of Noord-Brabant assigned by the Informatie Kern Team (IKT). The IKT is autonomous positioned within the organization. This autonomy is needed in the role that it occupies. The IKT maintains a procedure for approving ict-related projects. The procedure approves several stages before the start of the actual project implementation. The IKT-procedure and its progress was always invisible to the outside world. Therefore the team sought a solution to increase the visibility of its procedure. The Informatie Kern Team proposes to an existing application, called kredietcrisimonitor. It could be a possible solution to reuse this software with a layer of requirements suited for the IKT-procedure.

**Question** The IKT has asked to investigate how the IKT-procedure can be presented. Is it possible to reuse the existing software or is it better to launch a new development trajectory?

**Research** In an early state investigating the background and quality of existing application major risks came up. One of them was the lack of documentation. Causing no assessment could be made on the possibilities and quality of the application. This would lead to serious project delay.

**Advisory** The overall advisory was to leave the proposal of reusing the existing software. The team adapted the advice and from this point on the final project assignment was put together.

**Assignment** Develop a system where data from the IKT-procedure can be processed, stored and presented by IKT-employees to the projects stakeholders and organization.

The scope of this assignment was reduced to develop a client, server and database. The presentation module is excluded. The project started with formatting a plan of approach where all the phases are described. After completion of this initial phase the system design is drawn up in a functional and technical design. At this time step one of the implementation phase of the project is heading to its end. Discovering that plans not always turned out as ment, decisions were made. In the second step of the software implementation the testing phase takes off. The system will be tested on errors and fine tuning will take place. The expectation is to finish off this assignment at the end of October this year.



## **1 Inleiding**



## **1.1 Inleiding**

Binnen de provincie Noord-Brabant heb ik van het IKT (Informatie Kern Team) de opdracht gekregen voor een software ontwikkeling traject. In dit afstudeerverslag beschrijf ik het verloop van dit traject. Daarmee start ik met een globaal overzicht van de provinciale organisatie en een beschrijving van de plaats en rol van het Informatie Kern Team binnen deze organisatie.

De opdrachtgever heeft voorafgaand aan de opdracht een aantal vragen gesteld over mogelijke ontwikkeltrajecten. De conclusies en aanbevelingen kunt u lezen in de paragrafen vraagstelling, conclusies en aanbevelingen.

Op de aanbevelingen is een besluit genomen wat uiteindelijk heeft geleid tot de definitieve projectopdracht om een software ontwikkeltraject te starten voor een nieuw systeem.

Hierna volgt een beschrijving over het projectproces. Hierin zet ik uiteen hoe en met welke methodieken ik dit project ga uit te voeren.

In de uitvoering beschrijf daarna ik het verloop van het project. Hoe is het projectproces in de uitvoering verlopen? Wat is er anders gelopen en welke besluiten daarmee zijn gevallen.

Het project is nog niet afgerond op het moment van schrijven van dit verslag. Op dit moment wordt er volop gewerkt aan de implementatie van het systeem. Ik beschrijf na het uitvoerende gedeelte de stand van zaken en welke werkzaamheden het project nog voor de boeg heeft.



## **2 De organisatie provincie Noord-Brabant**

## **2.1 Een overzicht van de organisatie**

### **2.1.1 Positionering binnen de Overheid**

De provincie vormt de schakel tussen de gemeenten en het Rijk en wordt daarom ook wel het middenbestuur genoemd. Ze voert, kort samengevat, de taken uit die voor gemeenten te groot zijn en voor het Rijk te klein.

De provinciale kerntaken richten zich op ruimte, economie, milieu, mobiliteit en cultuur.

## **2.2 Politiek en Bestuur**

### **2.2.1 Commissaris van de Koningin**

De commissaris van de koningin is voorzitter van Provinciale Staten en van het college van Gedeputeerde Staten. In tegenstelling tot de andere provinciebestuurders is de commissaris niet gekozen door de inwoners, maar benoemd door de Kroon (Koningin en ministers).

### **2.2.2 Provinciale Staten**

Provinciale Staten (PS) vormen het collectief van volksvertegenwoordigers binnen de provincie. Noord-Brabant kent als grote provincie (meer dan 2.500.000 inwoners) 55 Statenleden. Deze worden iedere vier jaar rechtstreeks gekozen door de stemgerechtigde inwoners.

### **2.2.3 Gedeputeerde Staten**

Het college van Gedeputeerde Staten (GS) vormt het dagelijks bestuur van de provincie. De leden, gedeputeerden, worden voor een periode van vier jaar gekozen door Provinciale Staten.

### **2.2.4 Organisatiestructuur**

De bijna 1500 medewerkers van de provincie zijn verdeeld over 5 directies. Van deze vijf directies zijn er vier die zich ieder richten op een bepaald beleidsveld. Het gaat om de directies Economie & Mobiliteit (E&M), Ecologie (ECL),

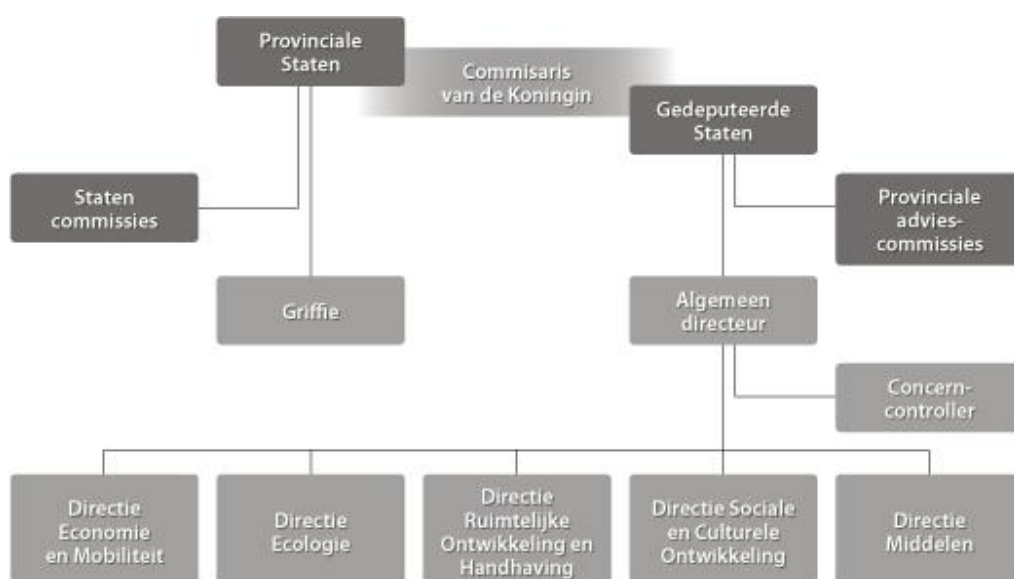
Ruimtelijke Ontwikkeling & Handhaving (ROH) en Sociale & Culturele Ontwikkeling (SCO). De vijfde directie is Middelen (MID) en richt zich op ondersteunende taken als personeelszaken, financiën, ICT en communicatie. Iedere directie wordt geleid door een directeur. Samen zitten de 5 directeuren in de Directieraad, waarin ook de algemeen directeur en de secretaris zitting hebben.

### 2.2.5 Bureaus

Binnen individuele directies is er een verdere onderverdeling in bureaus. Een bureau houdt zich bezig met een specifiek aspect binnen het beleidsveld van de betreffende directie. Ieder bureau heeft, met andere woorden, zijn eigen specialisatie. Zo kent de directie Ecologie onder meer de bureaus Natuur & Landschap en Bodem.

### 2.2.6 Organogram

Hieronder is de voorgaande uiteenzetting van de provinciale organisatie nogmaals schematisch weergegeven. Hier wordt ook de scheiding tussen Gedeputeerde en Provinciale Staten duidelijk.



### **2.2.7 Het Informatie Kern Team (IKT)**

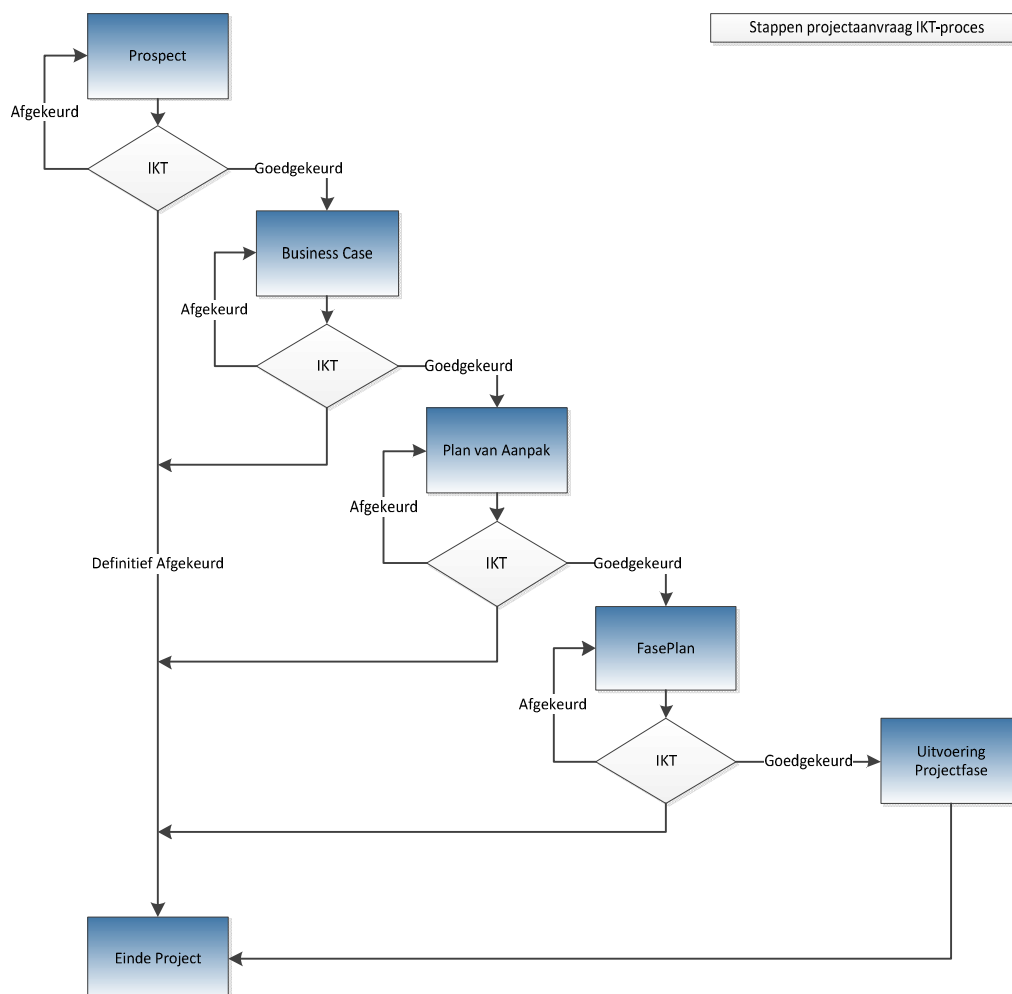
Deze ontwikkelopdracht is afkomstig van het Informatie Kern Team. Het team is niet weergegeven in het organogram, maar kan worden gepositioneerd boven de vijf directies met besluitvoering op directieniveau. Jaarlijks starten binnen de provincie Noord-Brabant vanuit de verschillende directies tientallen ict-gerelateerde projecten. Opdrachtgeverschap en verantwoording over deze projecten ligt bij de directies zelf. Coördinatie en controle van deze projecten binnen de Informatie-Kolom (I-Kolom\*) komt samen in Informatie Voorziening (IV) met Onno Vollinga als programmamanager. De programmamanager heeft het overzicht in de voortgang van projecten in uitvoering en is voorzitter van het Informatie Kern Team.

Het doel van het Informatie Kern Team is ook om te bepalen en/of te onderzoeken of projecten grond hebben om te starten. Er moet bijvoorbeeld een Business Case worden ingediend waar een haalbaarheidsstudie in wordt verwoord. Het team is met die reden samengesteld uit verschillende expertises waar onder andere het Bureauhoofd ICT-Beheer (bureau ITD; Informatie, Techniek en Data) en een lid van het ICT Architectuur Team. In de beoordeling maken ze gebruik van het IKT-proces. Alle projecten doorlopen dit proces voor ze mogen starten met de eigenlijke projectuitvoering. Hiermee wordt de onafhankelijke positionering in het organogram duidelijk, want niet alle projecten krijgen groen licht van het Informatie Kern Team.

## Het Informatie Kern Team

### 2.3 Projecten en Procedure

In de vorige paragraaf is duidelijk geworden dat alle ict-gerelateerde projecten pas tot uitvoering over mogen gaan na goedkeuring middels het IKT-procedure. Dit proces is onderverdeeld in een aantal stappen. De eerste stap is het indienen van een project prospect en in de laatste stap worden één of meerdere faseplannen voorgelegd. Over iedere stap neemt het Informatie Kern Team een besluit. Besluiten zijn niet zwart-wit, maar in het ongunstigste geval besluit het Informatie Kern Team dat een project -of een deel van het project (als het uit meerdere fases bestaat) niet tot uitvoering komt. De stappen die de procedure doorloopt zijn hier weergegeven.





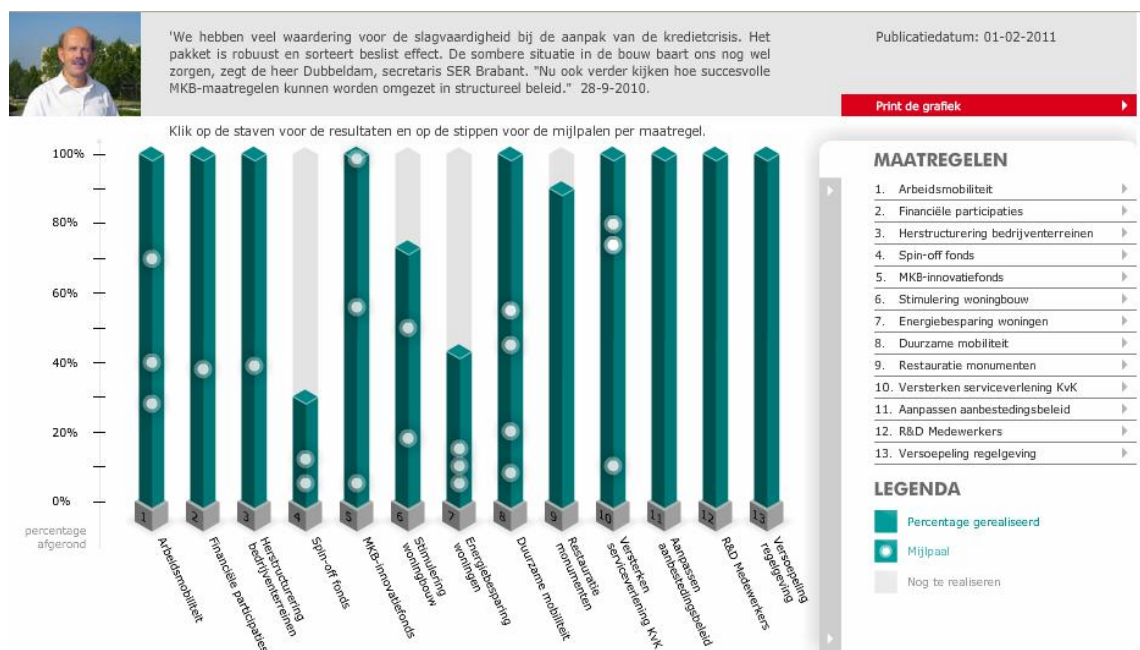
### **3 Probleemstelling, conclusies en opdracht**

## 3.1 Het Informatie Kern Team wil uitdragen

### 3.1.1 Probleemstelling

Het Informatie Kern Team is zich ervan bewust dat er een informatiebehoefte bestaat met betrekking tot de stand van zaken van projecten die de IKT-procedure doorlopen. Dit komt voort uit het constant moeten beantwoorden van vragen van belanghebbenden van projecten over de status van projecten binnen de procedure. Daarnaast wil het Informatie Kern Team zich profileren naar de organisatie toe. Het team is van mening dat er een manier moet zijn om de procedure algemeen inzichtelijk te krijgen voor de organisatie. Zij willen dit samenpakken in een Projectenmonitor.

Er is door het team reeds nagedacht over oplossingen. Daar is onder andere een project genaamd kredietcrisismonitor\* uit voren gekomen. Dit is een website waarin plannen en resultaten worden getoond om de kredietcrisis in de provincie Noord-Brabant te lijf te gaan. Vanuit het team is de suggestie gegeven om deze applicatie aan te passen naar de eisen en wensen voor de Projectenmonitor.



Afbeelding [www.kredietcrisismonitor.nl](http://www.kredietcrisismonitor.nl)



### **3.1.2 Doelstelling**

Uiteindelijk wil het Informatie Kern Team haar procedure op een of andere manier presenteren. Ze hebben al wel een duidelijk beeld wat ze willen laten zien. Dit is namelijk de voortgang van projecten binnen de IKT-procedure. Maar het team weet nog niet hoe ze dit concreet kunnen maken. De uiteindelijke doelstelling van de opdracht is om belanghebbenden inzicht te geven over de stand van zaken van projecten binnen de IKT-procedure.

## **3.2 Opdrachtverfijning**

Het Informatie Kern Team is op zoek naar een middel om de IKT procedure algemeen inzichtelijk te maken voor belanghebbenden van projecten.

Om te komen tot een advies voor de ontwikkeling van een dergelijk communicatiesysteem zijn de volgende stappen doorlopen.

- Kennisvergaring over de IKT procedure.
- Korte inventarisatie wat de opdrachtgever verwacht (eisen,wensen).
- Onderzoeken of het bestaande programma een optie is?
- Opsommen van conclusies en een aanbeveling maken.

Als het voorstel is goedgekeurd kan de projectopdracht worden geformuleerd en het plan van aanpak verder worden uitgewerkt.

### **3.3 Onderzoek**

#### **3.3.1 Vraagstelling**

Vanuit het Informatie Kern Team is de volgende vraag gesteld.

- Op welke wijze kan het beste invulling worden gegeven aan het presenteren van de IKT-procedure?

#### **3.3.2 Deelvraag**

- Is het hergebruik van het bestaande systeem (kredietcrisismonitor) toereikend?

#### **3.3.3 Conclusies**

- Het Informatie Kern Team heeft een systeem nodig waarin het gegevens kan invoeren, opslaan en presenteren.
- Algemeen kan men stellen dat hergebruik van een bestaand systeem als een goedkope oplossing inzetbaar is. Dit heeft wel de keerzijde dat een uitgebreid pakket van eisen en wensen niet van toepassing kan zijn. Na een korte inventarisatie blijkt dat de opdrachtgever eisen heeft die niet in dit systeem zijn opgenomen. Herschrijven van de bestaande software is dan ook onvermijdelijk.
- Het bestaande systeem, kredietcrisismonitor, blijkt niet door de provincie zelf geïmplementeerd. De broncode is wel in het bezit, maar het ontbreekt aan documentatie. Navraag bij de ontwikkelaar over eventuele documentatie van het systeem levert niets op. Er is geen documentatie voorhanden. Dit brengt een groot nadeel met zich mee. Op basis van systeemdokumentatie kan er namelijk al vrij snel een beoordeling plaatsvinden of de software kan worden aangepast aan de behoefte van het Informatie Kern Team. Daarnaast geeft deze documentatie al enig zicht op de kwaliteit van de programmeerde code. De beperkte kennis over dit systeem en het ontbreken van enige documentatie maakt dat een project op basis van software hergebruik onnodig veel risico's met zich meebrengt. Ook kunnen er dan geen harde afspraken worden gemaakt met betrekking tot eisen en wensen van de opdrachtgever. De reden

hiervoor is dat het vooraf niet duidelijk is en moeilijk kan worden ingeschat of deze eisen en wensen in te passen zijn.

- De ontwikkeling van een nieuw systeem brengt waarschijnlijk meer kosten met zich mee dan het hergebruik van software. Een groot voordeel van een blanco start is dat eisen en wensen makkelijker te realiseren zijn.

### **3.3.4 Aanbevelingen**

1. Hergebruik van de bestaande software is sterk af te raden gezien het ontbreken van systeemdokumentatie en parate kennis over het systeem.
2. Ontwikkel een nieuw systeem dat gegevens kan opslaan en publiceren.

### **3.4 De opdracht voor uitvoering**

Naar aanleiding van de conclusies en aanbevelingen gaat Onno Vollinga, lid van het IKT en opdrachtgever van dit project, akkoord met de aanbeveling om een nieuw systeem te ontwikkelen. Het Informatie Kern Team wil wel vasthouden aan de presentatie van gegevens zoals weergegeven door de website kredietcrisismonitor.

#### **3.4.1 Doelstelling**

Ontwikkel een systeem waarin:

- de gegevens van de IKT-procedure in worden opgeslagen en gepresenteerd;
- medewerkers van het Informatie Kern Team gegevens kunnen invoeren, opvragen en wijzigen;
- de provinciale organisatie en belanghebbenden de vorderingen van projecten binnen de IKT-procedure kunnen herleiden.

#### **3.4.2 Scope**

Met de opdrachtgever heb ik afspraken gemaakt over de werkzaamheden die binnen dit deel van het project vallen.

Het systeem zal bestaan uit

- een invoermodule;
- persistentie van gegevens\*;
- een presentatiemodule.

Het ontwerpen en ontwikkelen van de presentatiemodule ligt buiten de scope van mijn werkzaamheden. Ik zal hier slechts een informerende en/of coördinerende rol in hebben.

Voor het overige deel van het project ben ik verantwoordelijk. De taken en werkzaamheden die hierbij horen zijn:

- projectleiding;
- schrijven van project- en systeemdocumentatie;
- communicatie;
- systeemontwerp;

- systeemimplementatie;
- oplevering en ondersteuning.

Gezien de omvang van de opdracht is met de opdrachtgever afgesproken dat de projectgroep alleen uit mijzelf bestaat. Daarnaast ben ik wel vrij om medewerkers binnen de provinciale organisatie aan te spreken voor ondersteuning bij werkzaamheden.



## **4 Het Projectplan voor de 'ProjectenMonitor'**

### 4.1 Fasering

Het project wordt opgedeeld in fasen. Iedere fase is onderscheidend in werkzaamheden en documentering. De fasen volgen elkaar op en staan of vallen met de afronding van de voorgaande fase. Dit project zal de volgende fasen doorlopen.



Voor het systeemontwerp worden de volgende technieken gebruikt.

- MoSCoW\*
- UML\* (Unified Modeling Language)
  - o Use Cases
  - o Class Diagram
  - o Sequence Diagram
- ERD\* (Entity Relation Diagram)

Voor het testen van het systeem zal TMap\* de basis vormen.

## **4.2 De initiatiefase**

### **4.2.1 Werkzaamheden**

In deze fase worden een aantal werkzaamheden gedaan om het project op te starten namelijk.

- kennismaking Informatie Kern Team;
- verkennend gesprek met de opdrachtgever over de projectaanpak;
- schrijven plan van aanpak.

Het verkennend gesprek met de opdrachtgever over het project is erop gericht om elkaars verwachtingen over het project duidelijk te stellen. Verder worden de volgende onderwerpen besproken:

- projectfasering;
- projectplanning;
- bronnen binnen de organisatie;
- werkwijze/afhankelijkheden organisatie.

In het plan van aanpak wordt onder andere de fasering concreet uiteengezet. De onderwerpen die hierin worden behandeld zijn:

- scope van de opdracht;
- randvoorwaarden aangaande de opdracht;
- beheersing project;
  - o Risico-inventarisatie
  - o Kwaliteitsbeheersing
  - o Ontwikkelmethodiek
  - o Projectfasen en beschrijving
  - o Projectdocumenten
  - o Organisatie, taken en verantwoordelijkheden
  - o Kosten
  - o Planning
  - o Communicatie.



#### **4.2.2 Projectmedewerkers**

Eerder is reeds genoemd dat het project slechts plaats zal bieden aan één projectmedewerker. Dat betekent niet dat dit project verstoken zal zijn van resources. Er is veel kennis binnen de provincie op het gebied van software ontwikkeling. In de verschillende stadia van het project zal dan ook gebruik worden gemaakt van deze kennis op het gebied van systeemontwerp, implementatie en testen.

#### **4.2.3 Communicatie**

Communicatie gedurende het project vindt op de volgende manieren plaats.

- Driewekelijks een terugkoppeling met de opdrachtgever met betrekking tot de stand van zaken over de projectvoortgang. Deze gesprekken worden vooraf over de gehele projectduur ingepland.
- Eénmaal per vier weken zal er een voortgangsrapportage worden gemaakt.
- De projectdocumentatie wordt weggeschreven in een systeemmap op het provinciale netwerk en is toegankelijk voor projectmedewerkers, opdrachtgever en leden van het Informatie Kern Team.
- Verslaglegging van gesprekken zal worden beperkt worden tot onderwerpen en actiepunten (beknopte verslaglegging).

## **4.3 Ontwerpfase**

### **4.3.1 Functioneel ontwerp**

De ontwerpfase wordt opgesplitst in twee delen. In het eerste gedeelte wordt intensief met de opdrachtgever en leden van het Informatie Kern Team samengewerkt om tot een Functioneel Ontwerp te komen. In deze fase worden een aantal bijeenkomsten/workshops belegd die moeten leiden tot de invulling van een aantal facetten van het systeem. Die facetten zijn:

- functionaliteiten van het systeem;
- eisen en wensen;
- grafische gebruikers interface (gui)\*;
- data dictionary\*.

De functionaliteiten waar het systeem aan moet voldoen worden uitgewerkt in Use Cases\*.

De eisen en wensen worden verpakt in de MoSCoW methodiek. Dit om gradatie van belangrijkheid mee te geven aan de individuele eisen en wensen. Niet alles zal haalbaar zijn binnen de tijd of techniek.

### **4.3.2 Grafische Interface**

De ontwikkeling van de grafische interface zal in stappen gebeuren. Deze stappen zijn:

- één op één met de opdrachtgever wordt een aantal ideeën aangaande de layout geschetst;
- uitwerken van een ontwerp en presenteren aan leden van het Informatie Kern Team;
- in een gezamenlijke sessie bepalen opdrachtgever en IKT leden welke aanpassingen of toevoegingen nodig zijn;
- tijdens de systeemimplementatie volgt nog een iteratie om te bepalen of de grafische interface voldoet aan de verwachting en of er nog wijzigingen moeten plaatsvinden;
- een testteam test de grafische interface;
- wijzigingen en het oplossen van fouten kunnen in de tweede iteratie van de implementatiefase plaatsvinden.

Het samenstellen van de data dictionary gebeurt op basis van het vergaren en bestuderen van de documentatie over het IKT-proces. Daarna zal er nog een gesprek plaatsvinden met een lid van het team.

De data dictionary wordt net als alle andere onderwerpen gebundeld in het Functioneel Ontwerp document.

### **4.3.3 Technisch Ontwerp**

Het tweede deel van de ontwerpfase start als het eerste deel is goedgekeurd door de opdrachtgever. Hier wordt het functioneel ontwerp omgezet naar een technisch ontwerp.

Om tot een technisch ontwerp te komen worden een aantal technieken/modellen gebruikt.

- De data dictionary zal dienen als basis voor het Entiteit Relatie Diagram voor de database.
- Er wordt bepaald uit welke componenten het systeem bestaat en waar ze zich in dat systeem bevinden.
- De applicatiecomponenten worden ieder weer uitgewerkt in een klassendiagram\*.
- Complexe functionaliteiten worden uitgetekend in sequentiediagrammen\*.

Het is niet de bedoeling om solistisch het technisch ontwerp in te kleuren. Om de kwaliteit van de ontwerpen te borgen worden ze voorgelegd en besproken met een aantal collega's werkzaam in het Architectuur Team en de afdeling InformatieSystemen en Producten (ISP). Daarnaast zal het technisch ontwerp ook worden gepresenteerd aan de opdrachtgever.

Het ontwerp wordt gebundeld in een Technisch Ontwerpdocument.

## **4.4 De Implementatiefase**

In de implementatiefase wordt het ontwerp daadwerkelijk omgezet naar code. Hoe goed een ontwerp ook is, er kan niet zondermeer vanuit worden gegaan dat het één op één kan worden omgezet in code. In het document Technisch Ontwerp is een hoofdstuk Wijzigingen ingelast. Hier zullen tijdens de implementatie alle afwijkingen op het ontwerp worden gedocumenteerd. Ook worden de wijzigingen doorgevoerd in de ontwerptekeningen. Met als doel de documentatie te laten aansluiten met het uiteindelijke systeem.

De implementatie vindt plaats in twee iteraties.

Eerste iteratie:

- werkend systeem met een client, server en database
- grafische gebruikers interface

Aan het eind van de eerste iteratie is er een werkend systeem. De grafische gebruikers interface (gui) zal dan nogmaals worden voorgelegd en besproken met de opdrachtgever en leden van het Informatie Kern Team.

De tweede iteratie:

- starten met de testfase
- foutherstel
- wijzigingen in de gui doorvoeren

Aan het einde van de tweede iteratie is het systeem klaar voor gebruik.

Het hele ontwikkeltraject vindt plaats in een ontwikkelomgeving. De provincie hanteert een OTAP-straat\* (Ontwikkel, Test, Acceptatie en Productie) voor applicaties en systemen.

## **4.5 Testfase van het Systeem**

Het testen van het systeem zal door verschillende personen met een verschillende achtergrond in verscheidene stadia (ontwikkel/test/acceptatie) van het systeem plaatsvinden. Het testen zal bij aanvang van de implementatie al starten door de ontwikkelaar.

De ontwikkelaar test zijn klassen en de juiste werking van interactie tussen klassen (integratietesten). In een later stadium voert hij systeem- en interne\* en externe\* performance testen uit.

De provincie heeft de expertise in huis voor het testen van systemen en applicaties op basis van TMap. Deze fase wordt bewust uit handen gegeven aan een intern testteam, om te voorkomen dat bepaalde fouten niet worden onderkend (de bakker die zijn eigen brood keurt). Het testteam voert de testen uit in de Test en Acceptatie omgevingen op de onderstaande onderdelen. Deze testen starten na de oplevering van de eerste iteratie van de implementatiefase.

- In de testomgeving wordt gekeken of alle functionaliteiten werken en/of er geen technische gebreken zijn.
- De acceptatie-omgeving wordt gebruikt om samen met toekomstige gebruikers te beoordelen of alle functionaliteiten, eisen en wensen zoals benoemd in het Functioneel Ontwerp ook daadwerkelijk zijn geïntegreerd in het systeem.

Het testen van het systeem in de test-omgeving loopt gelijk met de tweede iteratie van de implementatiefase. De acceptatiefase verloopt een stuk korter. De verwachting is dat hier geen grote gebreken meer aan het licht komen.

#### **4.6 Oplevering en in Productiename van het systeem**

Op het moment dat de opdrachtgever akkoord gaat met het systeem in de acceptatieomgeving, dan is het gereed voor productie.

Bij oplevering wordt het systeem overgedragen aan de opdrachtgever/eigenaar. Daarbij wordt ook alle documentatie die gedurende het project is opgebouwd overhandigd. Evenals beheer- en gebruikersdocumentatie over het systeem.

Het beheer van applicaties en systemen is bij de provincie Noord-Brabant gescheiden in applicatie, functioneel en technisch beheer. Voor dit systeem wordt dat als volgt belegd:

- |                      |                                            |
|----------------------|--------------------------------------------|
| • Functioneel beheer | Binnen het Informatie Kern Team.           |
| • Technisch beheer   | Afdeling Informatie Techniek en Data.      |
| • Applicatiebeheer   | Afdeling Informatie Systemen en Producten. |

Om te komen tot een uiteindelijke in productiename van het systeem is er ook een rol voor de afdeling technisch beheer (ITD) weggelegd. Deze rol is tot nu toe niet benoemd. Vanaf de implementatiefase is er al contact met technisch beheer. Zij beheren de OTAP-omgeving, verzorgen backup, onderhoud op servers en prepareren desktop applicaties. Voorafgaande aan iedere projectfase –vanaf implementatie tot aan de opleverfase- zal er een inschatting moeten worden gemaakt welke resources nodig zijn binnen technisch beheer. Daarbij moet ook rekening worden gehouden met aanvragen van werkzaamheden en tijd tot effectuatie.



## **5 Projectuitvoering**

## **5.1 Opstartfase, formatie Plan van Aanpak**

### **5.1.1 Plan van aanpak**

De eerste fase heeft als doel om het plan van aanpak op te leveren. Met de opdrachtgever is gesproken om een aantal onderwerpen helder te krijgen. De uitkomsten zijn bepalend voor mijn verdere projectplan. De onderstaande punten zijn bepalend.

- De provincie werkt volgens projectmethodiek Prince2\*.
- In ontwikkeltrajecten wordt DSDM\* toegepast.
- Vrij in keuze voor een programmeertaal mits deze aansluit op de ict infrastructuur.
- Planning einde project augustus.
- Goedkeuring van opdrachtgever noodzakelijk voor:
  - o Plan van Aanpak;
  - o Functioneel Ontwerp;
  - o Grafische Interface;
  - o In productiename systeem.

### **5.1.2 Samenstellen Projectfasering**

Prince2\* is een basis waarop projecten binnen de provinciale organisatie worden uitgevoerd. De vraag is in hoeverre dit project moet voldoen aan alle elementen uit Prince2. Deze vraag wordt neergelegd bij de sr. Demand Manager. Hieruit volgt de conclusie dat mijn projectfasering sluitend is voor de een goed projectverloop. Dit mede gezien de schaalgrootte van het project en die is relatief klein in vergelijking met andere projecten binnen de organisatie. De projectfasering blijft dan ook zoals hieronder weergegeven.

- Initiatiefase
- Ontwerpfase
  - o Functioneel Ontwerpfase
  - o Technische Ontwerpfase
- Implementatiefase
- Testfase
- Opleverfase



### **5.1.3 DSDM**

Een ander aspect is het ontwikkelen van een systeem op basis van Dynamic Systems Development Method (DSDM\*). Navraag bij de afdeling Informatie Producten en Systemen (ISP) leert dat deze methodiek als zodanig niet meer wordt toegepast bij de provincie.

Met de opdrachtgever kom ik overeen dat de ontwikkeling van de grafische interface in een aantal iteraties zal plaatsvinden. Ontwerp en invulling hiervan start met de aanvang van de ontwerpfase.

### **5.1.4 Tijdsbesteding en achtervang**

De voortgang van het project borg ik in besteedbare tijd. Daarom maak ik de volgende afspraken met mijn leidinggevende en de opdrachtgever.

- twee werkdagen per week projectwerk mits de huidige werkzaamheden dat toelaten;
- projecttijd wordt flexibel geagendeerd.

Verder zal een lid van het Management Team van de I-Kolom als plaatsvervanger optreden bij ziekte of uitval van de opdrachtgever.

## **5.2 Ontwerpfase**

### **5.2.1 Functioneel Ontwerp**

Deze fase is erop gericht zoveel mogelijk informatie van opdrachtgever en toekomstige gebruikers te verkrijgen. Als basis dienen de volgende punten:

- functionaliteiten;
- eisen en wensen;
- grafische gebruikers interface (gui);
- data.

### **5.2.2 Gegevensverzameling over het IKT-proces**

Om alle gegevens te verzamelen die nodig zijn voor het systeem volg ik de volgende stappen.

- verzamelen van documenten omtrent het IKT-proces;
- individuele/informele gesprekken over het IKT-proces met leden van het Informatie Kern Team;
- gezamenlijke bijeenkomst om de functionaliteiten, eisen en wensen te bepalen;
- één op één gesprek met een lid van het Informatie Kern Team over systeemdta;
- gezamenlijke bijeenkomst met de opdrachtgever en een lid van het Informatie Kern Team over het functioneel ontwerp.

De bijeenkomsten en gesprekken zijn voorbereid. Op basis van de deskresearch en verzamelde informatie zijn alle gesprekken voorbereid. Wat inhoudt dat een aantal functionaliteiten, eisen en wensen al zijn ingevuld. Dit doe ik ook met de data dictionary. De rol van de opdrachtgever is hierin controlerend en aanvullend.

Uiteindelijk komen alle gegevens samen in een functioneel ontwerp document. Daarin zijn ook Use Cases per functionaliteit in uitgewerkt. Dit ontwerp is besproken met de opdrachtgever en beoordeeld door een systeemanalist van de afdeling Informatie Systemen en Producten. Met de afronding van het functioneel ontwerp start het technisch ontwerp van het systeem.

### 5.2.3 Grafische Gebruikers Interface

Als de gegevensverzameling bijna is afgerond, dan wordt er gestart met het ontwerpen met de grafische gebruikersinterface. De onderstaande stappen zijn doorlopen om hier invulling aan te geven.

- In een sessie met de opdrachtgever is schetsmatig (mock-ups\*) naar een eerste opzet toegewerkt. Er is gekeken naar:
  - o menustructuur;
  - o inhoud van de menu's;
  - o vlakverdeling;
  - o plaats en groeperen van de data;
  - o overzichtelijkheid.
- Op basis van de uitkomsten van de eerste sessie met leden van het Informatie Kern Team is er met behulp van Netbeans\* een werkende model-gui gebouwd en ter beoordeling aangeboden aan gebruikers en opdrachtgever. In de bijlage wordt beschreven hoe dit model tot stand is gekomen.
- In een gezamenlijke sessie met leden van het Informatie Kern Team is dit model nogmaals doorlopen op de punten die in de eerste sessie zijn bepaald.

Het aanbieden van een werkbaar model wierp direct zijn vruchten af. Voor de gebruikers was het tastbaar. Hierdoor werd in de gezamenlijke sessie al snel duidelijk wat de overige wensen aangaande de grafische interface waren. Er komen ook een aantal technische wensen ter sprake. Afhankelijk van de relevantie en of haalbaarheid is hier invulling aan gegeven.

De gebruikers krijgen nog een laatste model ter beoordeling. Hierin is de gewijzigde menustructuur, vlakverdeling en datapresentatie te zien. Aan het einde van de eerste iteratie van de implementatie volgt een laatste beoordeling. Die is bedoeld om eventueel nog kleine wijzigingen aan te brengen.

Hieronder is een afbeelding van het model van de grafische interface te zien. Er is een bijlage opgenomen waarin wordt uitgelegd hoe dit model tot stand is gekomen.

The screenshot displays the 'ProjectenMonitor' application window. The title bar includes the application name and standard window controls. Below the title bar is a menu bar with 'Bestand', 'Bewerken', and 'Info'. A toolbar contains a 'ProjectGegevens Ophalen' button. The main interface is divided into a left sidebar and a central content area. The sidebar lists two projects: '01 Bravissimo' (selected) and '02 Intranet 2.0'. The central area has tabs for 'ProjectGegevens proj.nr.' and 'Prospect'. The 'ProjectGegevens proj.nr.' tab is active, showing a form with the following fields: 'Projectnaam' (Bravissimo), 'Directie' (EenM), 'Afdeling', 'Opdrachtgever', and 'Projectleider'. To the right of these fields are radio buttons for 'Modus' (Alleen Lezen, selected; Bewerken). Below the form is a large 'Algemene ProjectInfo' text area. At the bottom left, there are checkboxes for 'Selecteer voor publicatie' (checked), 'Project beëindigd', and a date field for 'Datum beëindigd'. On the right side, a 'Status' dropdown menu shows 'Prospect' and 'Business Case'. Below it are 'Bekijk Fase' and 'Nieuwe Fase' buttons. At the very bottom, a footer bar contains 'ProjectGegevens', 'Opslaan', and 'Annuleren' buttons.

Afbeelding van het gui-model

#### **5.2.4 MoSCoW**

MoSCoW is een ezelsbruggetje voor Must, Should, Could, Won't. De eisen en wensen van de opdrachtgever zijn uiteindelijk ook nog eens gerangschikt met behulp van dit model. Dit is toegepast om helder te krijgen wat de basiswaarden binnen het systeem moeten zijn. Wat mag als er nog tijd over is en wat willen we nog, maar valt buiten dit project. Voor dit systeem geldt bijvoorbeeld dat er zeker één gebruikersaccount moet zijn. Als het kan wil de opdrachtgever een Acces Control List (ACL)\*, dus meerdere gebruikers met daaraan gekoppeld rechten op het systeem.

### 5.2.5 Technisch Ontwerp

Het technisch systeemontwerp wordt in stappen/delen opgebouwd. Niet alles zal tot in detail worden uitgewerkt. Dit geldt met name voor de invulling/uitwerking van de server en client componenten.

Eerst worden de componenten van het systeem bepaald. De componenten in dit systeem bestaan uit:

- client applicaties;
- server applicatie;
- database.

### 5.2.6 Ontwerp Database

Binnen het functioneel ontwerp is een data dictionary opgemaakt. Inhoudelijk zijn hier waar nodig de eisen en wensen en de data uit de gegevensverzameling in verwerkt. Er is al een onderverdeling gemaakt in data die gezamenlijk in één tabel kan komen.

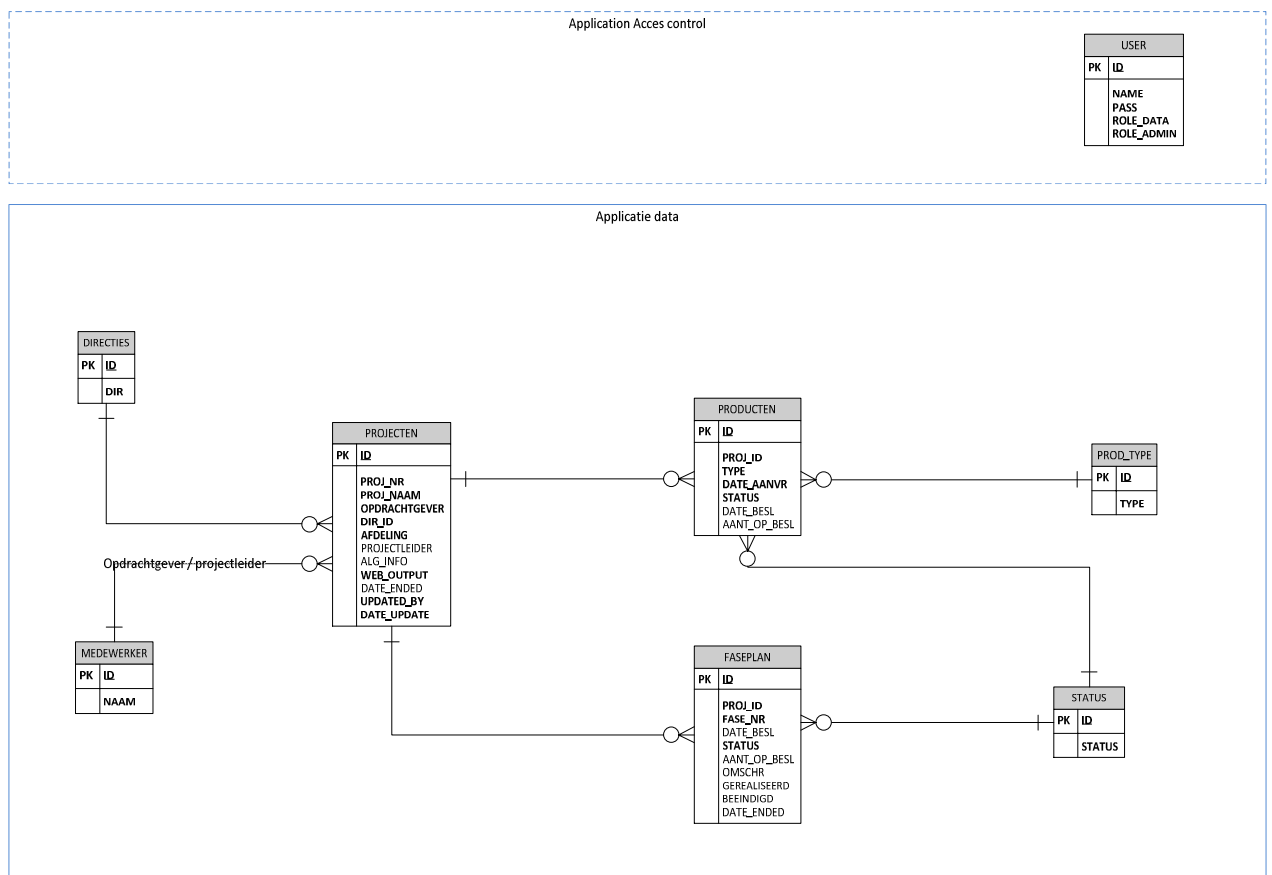
Het ontwerp van de relationele database wordt weergegeven in een Entiteit Relatie Diagram (ERD). Om tot een opzet te komen worden de volgende stappen genomen.

- bepalen entiteiten;
- data per entiteit bepalen;
- samenhang van de entiteiten onderling bepalen.

Na deze slag is het eerste datamodel geverifieerd door een database ontwikkelaar. Dit zal hierna nog een paar keer gebeuren om te voorkomen dat er fouten insluipen.

De volgende stappen zijn genomen om tot een datamodel te komen:

- sleutelwaarden\* bepalen;
- unieke waarden bepalen;
- mogelijkheid tot hulptabellen i.v.m. wederkerigheid van data;
- de samenhang tussen entiteiten vereist vreemde sleutelwaarden;
- bepalen gegevenstypen van de data (bijvoorbeeld: number, char, date);
- voorkomen van data redundantie\*;
- constraints\* en triggers\* bepalen.



ERD projectenmonitor

Sommige zaken in het ontwerp van het datamodel waren logischer dan anderen. De eisen en wensen moesten worden vertaald naar het systeem. De vraag komt waar je deze eisen en wensen inpast. Bijvoorbeeld: een voorwaarde is dat er per provinciale directie maximaal 12 projecten worden gepubliceerd in de

presentatiemodule. Deze voorwaarde kan zowel op server- als databaseniveau worden ondervangen. Beveiliging is binnen dit systeem geen onderwerp. Daarom is gekozen dit op applicatieniveau te regelen (server applicatie).

Hulptabellen zijn bepaald op basis van wederkerende data. Denk aan het invullen van gegevens als directienamen of de status van een project. Dit soort data wordt in de clientapplicatie aangeboden in keuzelijsten. Door de lijsten onderdeel te maken van de database zijn ze eenvoudig te wijzigen.

Het bepalen van de unieke sleutelwaarden is eenvoudig opgelost door iedere tabel te voorzien van een kolom met een unieke nummering. Een voordeel is dat deze data niet kunnen worden gewijzigd van buiten de database. Bijvoorbeeld: een update op een unieke sleutelwaarde betekent dat andere tabellen ook een update krijgen omdat deze waarde er als vreemde sleutel in voorkomt. Dat is nu niet nodig.



### 5.2.7 Ontwerp van server en client

De invulling voor het ontwerp van de server en client software hebben een heel andere benadering dan die van de database. In het database ontwerp ligt vast hoe data zich tot elkaar verhoudt. Het ontwerp van een applicatie geeft de werking van de applicatie weer. In het componentendiagram zijn de componenten al bepaald. De applicaties worden van grof naar fijn ontworpen. Ze worden uiteindelijk weergegeven in een klassendiagram. De stappen om daar toe te komen:

- bepalen welke klassen op de rand van de applicatie communiceren met de andere systeemcomponenten;
- interface klassen\* bepalen;
- bepalen vanuit de eisen en wensen van de opdrachtgever hoe mogelijk het programmaverloop kan zijn. Welke klassen horen daarbij?;
- bepalen hulp- en ondersteunende klassen;
- attributen\* en methoden\* binnen klassen bepalen;
- klassen plaatsen in een packagestructuur\*.

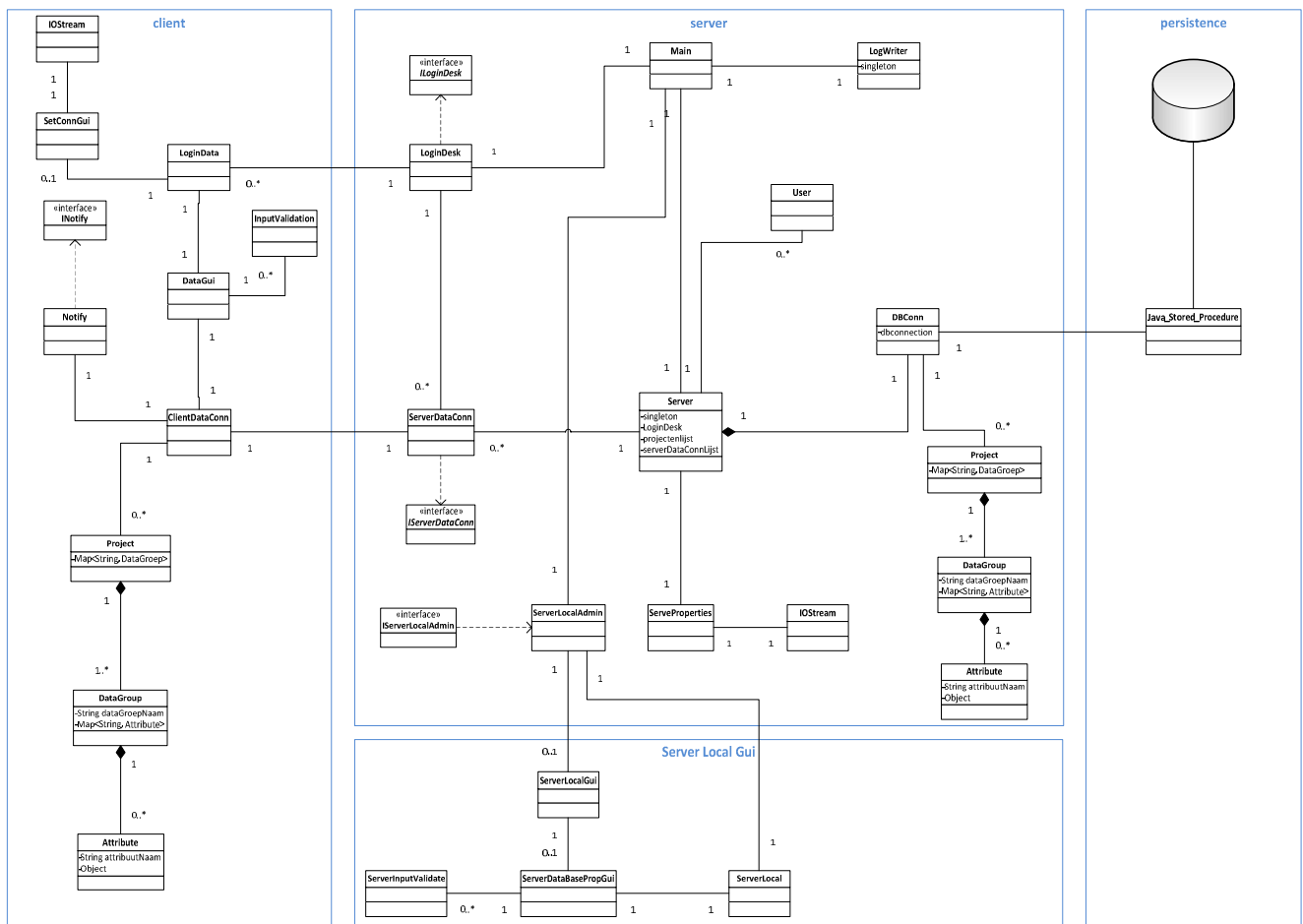
Naast deze stappen worden er ook sequentiediagrammen gemaakt. Dit om complexe programmaverlopen snel inzichtelijk te krijgen. Voor het inloggen door een gebruiker is bijvoorbeeld zo'n dergelijk diagram samengesteld.

De klassenformatie van de clientapplicatie krijgt min of meer een lineaire vorm. Anders dan de serverapplicatie die in zijn ontwerp met een centrale klasse een stervorm heeft. De server heeft dan ook te maken met meerdere externe componenten zoals verschillende clients en database.

De clients en server praten met elkaar door middel van Java\* RMI\* (Remote Method Invocation). Deze methodiek is eenvoudig te implementeren, maar vereist wel specifieke implementatie op de klassen die ermee te maken krijgen. Dat is de reden dat alle klassen die middels RMI met elkaar communiceren in kaart worden gebracht. Van elke klasse waar RMI deel uitmaakt wordt een interface klasse ontworpen.

Bij het intekenen van de klassen wordt ook een scheiding aangebracht tussen domein en grafische interface. In een diagram is dat abstract mooi uit te tekenen. In de praktijk betekent dit dat je het ontwerp zo maakt dat de grafische interface onafhankelijk is van de onderliggende applicatie. Er wordt dus buiten presentatie

en het collecteren van gegevens zo min mogelijk in de grafische interface geprogrammeerd.



klassendiagram projectenmonitor

links de gebruikersclient, midden boven de server, midden onder de beheerclient en rechts de database.

In het basisonwerp worden, zover als mogelijk, de functionaliteiten, eisen en wensen vanuit het functioneel ontwerp opgenomen. Daarnaast heeft het systeem nog een aantal ondersteunende klassen nodig. Dit zijn bijvoorbeeld klassen voor datavalidatie, opslaan en lezen van bestanden of die logging vanuit het systeem verwerken.

In de klassen van het basissysteem en de interfaceklassen worden attributen en methoden aangegeven. Methoden worden ook zichtbaar weergegeven in sequentiediagrammen. Maar niet alle klassen worden nu al voorzien van alle attributen en methoden. Dat zal gebeuren in de implementatiefase. Bij invulling van het ontwerp is gelet op:

- Hoe wordt data getransporteerd vanaf de client naar de database en omgekeerd?
- Welke gegevensuitwisseling vindt er plaats tussen de remote klassen?
- Moeten gegevens tussentijds worden opgeslagen op het systeem?

Door niet alle attributen en methoden nu al vast te leggen ontstaat er een mate van vrijheid in de implementatiefase. Echter wordt deze vrijheid wel beperkt doordat de structuur en dus ook interactie van de klassen al vastligt. Daarnaast zal in de implementatiefase het coderen volgens een code conventie\* plaatsvinden. Daarin is onder andere belegd dat alle methoden binnen klassen moet worden gedocumenteerd.

Als het ontwerp gereed is, dan worden de klassen gerangschikt in packages\*. Dit gebeurt op basis van functionaliteit binnen de applicaties. Als voorbeeld staan alle grafische klassen van een applicatie samen in een package net als alle klassen die dienen voor persistentie van gegevens.

Net als bij het ontwerp van het datamodel worden de ontwerpen getoetst door een collega.

Het geheel van ontwerpen wordt daarna gebundeld in een Technisch Ontwerp document. Hierin staan de ontwerpmodellen, interfacebeschrijvingen en staan onderwerpen beschreven zoals applicatie schaalbaarheid en veiligheid.

## **5.3 Implementatiefase**

### **5.3.1 Inleiding**

In deze fase wordt gestart met het bouwen van het ontworpen systeem. De volgende elementen spelen hierin een rol:

- database;
- server applicatie;
- server grafische interface;
- client applicatie;
- client grafische interface.

### **5.3.2 Implementatie van de database**

Deze implementatie bestaat uit het samenstellen van scripts. In deze scripts worden de volgende elementen van de database weergegeven:

- tabellen met daarin;
  - o Sleutelwaarden;
  - o Unieke waarden.
- constraints op tabellen;
- sequences;
- triggers;
- stored procedures;

De scripts worden vervolgens geladen in een Oracle database ontwikkelomgeving.

Het plan was om alle Stored Procedures in Java te schrijven. Oracle biedt namelijk deze mogelijkheid. Met deze keuze was het mogelijk om Java als programmeertaal over de hele linie te handhaven. Echter is de kennis om deze implementatie door te voeren niet aanwezig. De tijd die tijdens de implementatie wordt gespendeerd om dit werkend te krijgen heeft zijn weerslag op de voortgang van het project. De opdrachtgever is daar niet tevreden over. Om de projectvoortgang niet verder op te houden wordt de implementatie van de stored procedures in samenwerking met de afdeling ISP afgerond. De consequentie is dat de stored procedures in PL/SQL in plaats van Java worden geprogrammeerd.

### 5.3.3 Bouw van server- en clientapplicaties

Voor het inrichten/bouw van de applicaties is gekozen voor de programmeertaal Java. Java is een object georiënteerde taal en is niet afhankelijk van een computer besturingssysteem.

De documenten uit de ontwerpfase dienen als basis voor de bouw van de applicaties. Dit zijn de volgende documenten:

- functioneel ontwerpdocument;
- technisch ontwerpdocument;
- klassendiagrammen;
- sequentiediagrammen;
- datamodel (ERD).

Aan het technisch ontwerp document is een extra hoofdstuk toegevoegd. Hierin komen alle wijzigingen te staan die tijdens de implementatie worden doorgevoerd. Behalve het functioneel ontwerp zijn de overige documenten dynamisch van aard. Wijzigingen ten opzichte van het technisch ontwerp worden tijdens de implementatie ook doorgevoerd in de ontwerpdocumentatie.

Om de codering overzichtelijk te houden wordt een code conventie toegepast. In de conventie staan afspraken over zaken als:

- layout;
- inspringen van regels;
- gebruik van spaties;
- gebruik van lege regels;
- gebruik van hoofd- en kleine letters;
- begrijpelijke naamgeving aan attributen en methoden.

Buiten de bovenstaande regels is ook afgesproken dat alle methoden een beschrijving krijgen. In deze beschrijving staat wat de methode doet, wat zijn de input variabelen en zijn output (pre/post conditie).

Het programmeren zelf gebeurt in de ontwikkelapplicatie Netbeans. Dit programma is ook uitgerust met functionaliteiten om een grafische interface te bouwen. Verder wordt er gebruik gemaakt van een online versiebeheerprogramma.

De eigenlijke bouw van de applicaties vindt gefaseerd plaats. De stappen die worden genomen zijn de volgende:

#### Eerste iteratie

1. Eerst zijn alle packages met daarin de klassen en interfaces aangemaakt.
2. De Remote (RMI) structuur is geïmplementeerd.
3. De databasetoegang wordt geregeld.
4. Lokale persistentie en logging.
5. De functionaliteiten vanuit het Functioneel ontwerp.
6. Hulpklassen voor bijvoorbeeld datavalidatie.
7. Grafische interfaces worden gebouwd en gekoppeld aan de applicatie.

#### Tweede iteratie

8. Verfijnen, solide maken van de applicaties.
9. Debuggen en herstellen van fouten.
10. Doorvoeren van wijzigingen op de grafische interface van de client.

De eerste zeven stappen maken onderdeel uit van de eerste iteratie van de implementatiefase. De volgorde is gekozen om eerst een werkende basis op te zetten. Vanaf punt 3 kunnen alle systeemcomponenten met elkaar communiceren. Die communicatie wordt ondersteund met gegevens die lokaal worden opgeslagen. Denk dan aan credentials om aan te loggen op een database server.

Het opzetten van lokale persistentie in punt 4 was in het ontwerp niet beoogd. Dit kwam pas aan het licht tijdens de implementatie. Database credentials, servernamen en poorten zijn dynamisch en kunnen niet ingecapseld worden in de code. Als gevolg hiervan zijn er een aantal maatregelen op de applicaties genomen:

- implementatie van klassen die bestanden kunnen lezen en schrijven;
- aanpassingen in de opstartroutine van de applicaties;
- ontwikkelen van een grafische beheerinterface voor de server;
- een extra grafische interface voor de client.

Het gevolg van deze extra implementatie was een uitloop in de planning. In samenspraak met de opdrachtgever en huidige leidinggevende is voor deze implementatie tijdelijk extra tijd ingeruimd.

Nu de basis van de applicaties is gelegd volgt de implementatie van de functionaliteiten. De implementatie van een functionaliteit gaat door de gehele code heen. Bijvoorbeeld wordt voor het ophalen van projectgegevens uit de database gestart met coderen in de client. De client vraagt deze gegevens aan de server die op zijn beurt deze gegevens aan de database opvraagt.

Als de essentie van de functionaliteiten is geïmplementeerd volgt de volgende stap. In deze stap worden eisen vanuit het functioneel ontwerp geïmplementeerd die niet direct met de functionaliteiten te maken hebben. Dat is bijvoorbeeld gegevensvalidatie om de werking en continuïteit van het systeem te garanderen. Daarnaast ligt er een eis dat een gebruiker tot maximaal één uur na zijn laatste actie op het systeem ingelogd is op het systeem.

De laatste stap is de bouw van de grafische module van de client. De grafische beheermodule van de server is reeds meegenomen onder punt 4 Lokale Persistentie. Deze module wordt in zijn geheel opnieuw gebouwd ondanks het bestaan van het gui-model dat reeds in de ontwerpfase is uitgewerkt. Dit is niet zonder reden. Het model is vervuild, heeft een snelle, slordige codering en is niet geschikt om te gebruiken. In principe zijn hier dezelfde argumenten van toepassing als voor het hergebruik van bestaande software. Het model staat dan ook echt model.

### **5.4 Stand van Zaken**

Zoals hiervoor al werd geschreven is de implementatie van het systeem in volle gang en is het project nog niet ten einde ten tijde van dit stuk.

- De database draait in de ontwikkelomgeving. De opdrachtgever kan hiermee gaan uitkijken naar een derde partij die invulling geeft aan de ontwikkeling en bouw van de presentatiemodule.
- De server applicatie is bijna klaar en er is reeds gestart met het inrichten van de client applicatie.
- Het testteam binnen de provincie heeft de opdracht aanvaard en zijn op basis van de ontwerpdocumentatie bezig een testplan op te zetten.

### **5.5 Wat moet er nog gebeuren?**

Eenvoudig verwoord zijn dat alle werkzaamheden vanaf de start van de tweede iteratie van de implementatiefase. In een korte opsomming hieronder weergegeven zijn dat:

- goedkeuring krijgen over de grafische interface;
- testen van het systeem;
- fouterstel van het systeem;
- schrijven documentatie voor de verschillende beheergroepen en de gebruikersgroep;
- opleveren van het systeem.

Er ligt dus nog behoorlijk wat werk voor de boeg. Zoals het er nu uit ziet wordt het systeem eind oktober opgeleverd.





## **6 Reflectie**

### ***6.1 Reflectie op mijn rol als projectleider en de provinciale organisatie.***

In mijn opdrachtbeschrijving voor Fontys heb ik een aantal persoonlijke ontwikkelpunten opgesomd. Eén daarvan was de rol als projectleider. Een rol die ik niet ambieer, maar wel moest invullen om deze opdracht te vervullen. Het valt me eerlijk gezegd niet tegen deze rol uit te voeren, maar het uiteenzetten en bijhouden van werkplanningen is absoluut niet mijn sterkste kant. Zeker als je nooit zelfstandig aan een project bent gestart is het maken van een planning je begeven op glad ijs.

Het projectproces, de fasen die ik doorloop en de werkzaamheden die ik daarin uitvoer zijn grotendeels vertrouwd. Zeker als het aankomt op technisch ontwerp en implementatie van het systeem. Het samenstellen van documentatie als projectplannen en rapporten is voor mij een lastige bezigheid. Daar bedoel ik mee dat ik de plannen en rapporten niet uit mijn mouw schud. Ik denk dat dit meer een constatering moet zijn dan een punt voor verbetering.

De omgang met mensen heeft me nooit in de weg gezeten. Dat merk ik ook nu weer. Ik maak makkelijk nieuwe contacten en de interactie met de opdrachtgever, leden van het Informatie Kern Team en collega's die ik bij dit project heb betrokken lopen soepel.

Binnen de provinciale organisatie kan ik terecht met al mijn vragen aangaande de opdracht. Het valt me reuze mee als ik merk hoe behulpzaam collega's zijn. Met die reden kan ik de provincie Noord-Brabant als opdrachtgever voor een afstudeerstage van harte aanbevelen.



## **7 Verklarende Woordenlijst**

**Access Control List**

Access Control List (ACL) voor computersystemen is een specificatie van welke rollen en rechten gebruikers hebben binnen een dergelijk systeem.

**Command Line**

Grafisch venster op een computerbeeldscherm waar door middel van het typen van tekst opdrachten (commands) aan de computer worden gegeven. De meeste command line toepassingen/applicaties zien we nu nog terug in Linux besturingssystemen.

**Data Dictionary**

Een lijst gebruikt in software ontwikkeling met daarin informatie over gegevens/data zoals onder andere een beschrijving, datatype en welke onderlinge relatie deze data met elkaar heeft.

**Database Constraints**

Gedefinieerde voorwaarden/restricties in een database.

**Database Redundantie**

Het meerdere keren opslaan van dezelfde data in dezelfde database.

**Database Sleutelwaarden**

Een primaire sleutelwaarde (primary key) in een relationele database is een unieke waarde per record in een tabel waarmee het record wordt geïdentificeerd. Dezelfde sleutelwaarde kan ook voorkomen, als koppeling, in een andere tabel en wordt dan omschreven als vreemde sleutel (foreign key).

**Database Trigger**

Een geautomatiseerde actie in de database. De voorwaarden om de actie te starten zijn vooraf gedefinieerd.

**DSDM**

Dynamic Systems Development Method, of kortweg DSDM, is een Rapid application development methode, die voornamelijk wordt gebruikt bij projecten voor ontwikkeling van gecomputeriseerde informatiesystemen.

**ERD**

Entity Relationship Diagram is een manier om op een abstracte wijze het ontwerp c.q. de samenhang van een database weer te geven.

**Gegevenspersistentie**

Onder persistentie van gegevens in de ict wordt verstaan het opslaan van data op een manier dat deze data later weer kan worden opgevraagd. Bijvoorbeeld het opslaan van gegevens in een database of bestand.

**Graphical User Interface**

Afgekort GUI en vertaald als Grafische Gebruikers Interface is het zichtbare deel van een computersysteem of applicatie waar de gebruiker mee werkt. Een grafische interface maakt gebruik van visuele elementen in plaats van tekstopdrachten zoals op een command line.

**I-Kolom**

De Informatie-Kolom is een verzameling bureaus/afdelingen binnen de directie Middelen van de provincie Noord-Brabant die als gemeenschappelijke deler digitale informatie beheren.

**Java**

Een object georiënteerde programmeertaal.

**Java RMI**

Staat voor Remote Method Invocation en is een implementatie binnen de programmeertaal java om onafhankelijke java applicaties op hetzelfde of verschillende computersystemen met elkaar te laten communiceren.

**Interface klasse**

Een onderdeel van een objectgeoriënteerde applicatie. Een onafhankelijk stuk code dat, onder bepaalde voorwaarden, kan worden geïmplementeerd door andere klassen.

**Klassendiagram**

Maakt deel uit van UML. Een schematische weergave van een software applicatie voor object georiënteerde programmeertalen.

**Kredietcrisimonitor**

[www.kredietcrisimonitor.nl](http://www.kredietcrisimonitor.nl) is een website met een overzicht van een pakket maatregelen dat in 2010 is gelanceerd vanuit de provincie Noord-Brabant om de crisis in Brabant te bestrijden.

**Mock-up**

Is een techniek waarbij een schets of een serie schetsen van vensters van een software applicatie worden gebruikt in de ontwikkeling van een grafische gebruikers interface.

**MoSCoW**

Een hulpmiddel dat wordt gebruikt binnen de systeemontwikkeling om de eisen en wensen van een opdrachtgever te prioriteren. MoSCoW is een ezelsbruggetje en staat voor:

- Must           Moet in het eindresultaat terugkomen, zonder deze eis is het product niet bruikbaar.
- Should        Zeer gewenst, maar zonder is het product wel bruikbaar.
- Could         Mag alleen aan bod komen als er tijd genoeg is.
- Won't         Zal in dit project niet aan bod komen maar kan in de toekomst, bij een vervolgproject, interessant zijn.

**Netbeans**

Een software ontwikkelprogramma van het bedrijf Oracle.

**OTAP-straat**

Staat voor Ontwikkel, Test, Acceptatie en Productie-omgeving. Dit is een gangbare opstelling voor een ict infrastructuur waarbij de kwaliteit van de aangeboden software aan de gebruiker geborgd wordt door het doorlopen van de gehele OTAP-straat.

**Performance testen Externe**

Het testen van een software applicatie in verhouding tot andere software componenten zoals besturingssysteem, netwerk en database.

**Performance testen Interne**

Ook wel software profiling genoemd is het testen van de prestatie van de applicatie zelf. In de meeste software ontwikkelapplicaties is dit soort tooling standaard ingebed.

**Prince2**

Voluit PRojects IN Controlled Environments is een gestructureerde methode voor projectmanagement.

**Rapid application development**

(RAD) is een softwareontwikkelmethode, ook wel methodische aanpak voor het ontwikkelen van software in projectvorm. RAD is een concept dat organisaties in staat stelt softwaresystemen sneller en van betere kwaliteit te (laten) ontwikkelen.

**Sequentiediagram**

Maakt deel uit van UML. Een schematische weergave om aan te geven hoe een functionaliteit van een applicatie de software doorloopt.

**Software Attribuut**

Een waarde binnen de programmeercode.

**Software Code Conventie**

Reeks afspraken met betrekking tot de notatie en/of layout van software codering met als bedoeling standaardisatie en vereenvoudiging van de leesbaarheid.

**Software Methode**

Een stukje code dat kan worden aangeroepen met opgave van variabelen om een uitkomst te genereren. Dat kan bijvoorbeeld zijn een optelsom of andere bewerking. Het voordeel hiervan is dat deze code dan slechts éénmaal voor hoeft te komen binnen de gehele code.

**Software Packagestructuur**

Groeperen van klassen binnen een software applicatie die een onderlinge samenhang hebben of overeenkomstige eigenschappen bezitten.

**TMap**

Is een methodiek om software te testen ontwikkeld in Nederland door het bedrijf Sogeti.

**UML**

Unified Modeling Language, afgekort UML, is een modelmatige taal om objectgeoriënteerde analyses en ontwerpen voor een informatiesysteem te kunnen maken. UML is ontworpen door Grady Booch, James Rumbaugh en Ivar Jacobson in de jaren negentig en het is sinds 1997 een standaard. Kenmerkend is dat de UML-modellen een grafische weergave zijn van bepaalde aspecten van het informatiesysteem.

**Use Cases**

Maakt deel uit van UML. Een Use Case beschrijft op welke mogelijke wijze een gebruiker van een software applicatie een functionaliteit van die applicatie doorloopt.





## **8 Literatuurlijst en/of bronvermelding**

## **8.1 Literatuur en/of bronvermelding**

- Core Java Volume II Advanced Futures  
Cay S. Horstmann, Gary Cornell  
Eight edition 2008  
Sun Microsystems Press
- Praktisch UML 3de editie  
Jos Warmer en Anneke Kleppe  
Derde druk 2006  
Pearson Education Benelux
- Databases en SQL  
Ton de Rooij  
Derde druk 2005  
Academic Service
- User Interface Design, A Software Engineering Perspective  
Soren Lauesen  
First printed in 2005  
Pearson Addison Wesley



## **9 Bijlagen**

## **9.1 *Bouw van een model Grafische Interface in de ontwerpfase***

Om de gebruiker zo snel mogelijk iets tastbaars te geven werd de keus gemaakt om een werkend model te ontwerpen. De start van het ontwerp gebeurt na de gegevensverzameling.

Basisgegevens voor ontwerp van de model gui:

- functionaliteiten;
- eisen en wensen;
- data dictionary;
- use cases;
- use cases vertaald naar menustructuur (boomstructuur).

In de eerste stap is er schetsmatig op papier te werk gegaan. Hier is vooral gekeken naar de opzet van de menu's en de vlakverdeling.

Met de bovenstaande input wordt een start gemaakt met de bouw. De model gui wordt gemaakt met behulp van het programma Netbeans. Deze ontwikkel applicatie bevat ook een uitgebreide module voor het ontwikkelen van een grafische interface.

- De aanroep van de functionaliteiten zijn verwerkt in een menubalk bovenin de applicatie. Gebruikers zijn hier aan gewend!
- Bepaald was dat gegevensoverzichten in een smalle kolom aan de linkerzijde in het scherm geplaatst.
- De metadata wordt dan rechts gepresenteerd.

De vlakverdeling staat. De volgende stap is het groeperen van data. Eerst is er onderscheid gemaakt tussen permanente gegevens en procesgegevens van de IKT-procedure. Onder permanente gegevens vallen bijvoorbeeld projectnummer en projectnaam. Gegevens per processtap zijn ingedeeld als procesgegevens. De permanente gegevens zijn blijvend in beeld. De procesgegevens zijn in beeld afhankelijk van welke processtap de gebruiker van het project wil inzien.

Dan is de verdeling van de attributen in de velden. De schikking moet logisch zijn voor de gebruiker. Bijvoorbeeld adresgegevens waarbij de indeling naam, adres, woonplaats gebruikelijk is.

Om het voor de gebruiker levend te maken is er ook data aan het model toegevoegd.

Als laatste stap wordt ervoor gezorgd dat het model ook echt werkt. Alle functionaliteiten en knoppen werken. De gebruiker krijgt dan een reëel beeld van wat hij kan verwachten.

The screenshot displays the 'ProjectenMonitor' application window. The title bar includes the application name and standard window controls. The menu bar contains 'Bestand', 'Bewerken', and 'Info'. A toolbar at the top left features a 'ProjectGegevens Ophalen' button. The main interface is divided into a left sidebar and a central content area. The sidebar lists two project entries: '01 Bravissimo' (selected) and '02 Intranet 2.0'. The central area is titled 'ProjectGegevens proj.nr.' and 'Prospect'. It contains several input fields: 'Projectnaam' (filled with 'Bravissimo'), 'Directie' (a dropdown menu showing 'EenM'), 'Afdeling', 'Opdrachtgever', and 'Projectleider'. Below these is a large text area labeled 'Algemene ProjectInfo'. At the bottom left of the central area, there are checkboxes for 'Selecteer voor publicatie' (checked), 'Project beëindigd', and a date field for 'Datum beëindigd'. On the right side, there is a 'Modus' section with radio buttons for 'Alleen Lezen' (selected) and 'Bewerken'. Below this is a 'Status' section with a dropdown menu showing 'Prospect' and 'Business Case'. At the bottom right of the central area are two buttons: 'Bekijk Fase' and 'Nieuwe Fase'. The bottom status bar includes the text 'ProjectGegevens' and two buttons: 'Opslaan' and 'Annuleren'.

Afbeelding van de model gui

## **9.2 *Plan van Aanpak***