

Framework voor .NET batchjob ontwikkeling

Scriptie

```
applicatie host: VM3E000104V01.office01.internalcorp.net
parameters:
antalFout : <Nothing>
antalGoed : <Nothing>
antalVerwerkt : <Nothing>
SN : <Nothing>
laandwerkBWWW : J
laandwerkLoonsuppletie : J
laandwerkPeriode : 201102
laandwerkSuppletie : J
laandwerkValent : J
laandwerkWachtgeld : J
periode : <Nothing>
periodeJaar : <Nothing>
periodeMaand : <Nothing>
unMoment : 20110509074336
-----
parameter/input controle
-----
start batch verwerking
-----
selecterenClientenRechten
-----
-5-2011 7:43:59 [Information] BSN: 123, aanvraag: 2820 => Niet verwerkt: Geen lopende uitkering.
-5-2011 7:44:01 [Information] BSN: 8189730, aanvraag: 71604 => Niet verwerkt: Geen lopende uitkering.
-5-2011 7:44:01 [Information] BSN: 10151540, aanvraag: 48631 => Niet verwerkt: Geen lopende uitkering.
```

Auteur Pascal Widdershoven
Plaats Heerlen
Datum 7 juni 2011

Gegevens student	Pascal Widdershoven
Studentnummer	2110509
Profiel	Software Engineering
Stageperiode	07-02-11 – 17-06-11
Gegevens bedrijf	Algemene Pensioen Groep Oude Lindestraat 70 6411 EJ Heerlen
Bedrijfsbegeleider	Jos Philippen
Gegevens docentbegeleid(st)er	Erik van der Schriek

Getekend voor gezien door de bedrijfsbegeleider:

Datum:

De bedrijfsbegeleider,
Jos Philippen

Voorwoord

Deze scriptie is geschreven in het kader van mijn afstudeerstage bij APG te Heerlen. Dit document markeert het einde van mijn opleiding Informatica aan de Fontys Hogeschool te Eindhoven. Het voornaamste doel van dit document is het uiteenzetten en verklaren van de door mij gehanteerde werkwijze en de ervaringen die ik heb opgedaan tijdens deze stage.

De stage heeft plaatsgevonden in de periode februari tot en met juni 2011. In deze periode heb ik gewerkt aan de opdracht "Kwaliteitsverbetering batchjobs". Ik heb het erg naar mijn zin gehad bij APG en ik denk dat ik met recht kan zeggen dat de oplossing die uit dit project naar voren gekomen is de kwaliteit van met name foutafhandeling en logging van batchjobs in de toekomst significant zal verbeteren.

Ik wil bij deze graag mijn bedrijfsbegeleiders Jos Philippen en Jeroen van den Boorn bedanken voor hun begeleiding en betrokkenheid bij het project. Verder wil ik de collega's van de afdeling AOC bedanken voor de gezelligheid en de prettige werksfeer.

Ten slotte gaat mijn dank uit naar docentbegeleider Erik van der Schriek voor de begeleiding op proces vlak en de gegeven adviezen en feedback.

Pascal Widdershoven

Inhoudsopgave

Inhoudsopgave	3
Samenvatting	4
Summary	5
Verklarende woordenlijst	6
1. Inleiding	8
2. Het bedrijf	9
3. De opdracht	11
3.1. Situatie en probleemstelling	11
3.2. Doelstellingen	11
3.2.1. Hoofddoelstelling	11
3.2.2. Subdoelstellingen	12
3.2.3. Het bereiken van de doelstellingen	12
3.3. De opdrachtgever en betrokken partijen	12
3.4. Projectorganisatie	13
3.5. Eindproducten	13
3.6. Onderzoek	14
3.6.1. Onderzoeksvragen en resultaten	14
4. Werkzaamheden	17
4.1. Definitiestudie en onderzoek	17
4.2. Standaard voor batchjob ontwikkeling	17
4.3. Functioneel ontwerp	17
4.3.1. Batch framework	17
4.3.2. Batchjob B21	17
4.4. Technisch ontwerp	18
4.4.1. Batch framework	18
4.4.2. Batchjob B21	20
4.5. Ontwikkeling	20
4.5.1. Batch framework	20
4.5.2. Batchjob B21	21
4.5.3. Visual Studio template	21
4.6. Testen	21
4.6.1. Batch framework	21
4.6.2. Batchjob B21	21
4.7. Voorbereiding tot invoering / overdracht	22
4.8. Relatie tot het Tien Stappen Plan	22
4.8.1. Oriëntatiefase	23
4.8.2. Ontwikkelings- en oplossingsfase	23
4.8.3. Invoeringsfase	23
4.8.4. Conclusie	23
5. Uitdagingen en resultaten	24
5.1. Batch framework	24
5.1.1. Resultaat	24
5.1.2. Uitdagingen	24
5.2. Batchjob B21	26
5.2.1. Resultaat	26
5.2.2. Uitdagingen	26
6. Conclusie en aanbevelingen	27
Persoonlijke evaluatie stage	28
Literatuurlijst en bronvermelding	29
Internetpagina's	29
Bijlagenoverzicht	29

Samenvatting

Deze afstudeerstage is uitgevoerd bij APG te Heerlen. APG is specialist op het gebied van collectieve pensioenen in de publieke sector. APG heeft een intern IT bedrijf genaamd CIS dat zich onder meer bezig houdt met applicatie beheer en ontwikkeling.

De stage stond in het kader van de kwaliteitsverbetering van *logging* van *batchjobs*. Batchjobs zijn computerprogramma's die verzamelingen gegevens verwerken zonder menselijke interactie.

De gegevens worden vaak tijdens kantooruren verzameld door input vanuit verschillende applicaties en op een later tijdstip, bijvoorbeeld buiten kantooruren, verwerkt. De enige vorm van communicatie tussen mens en batchjob zijn de zogenaamde log files. Hierin wordt informatie vastgelegd over de uitvoering van een batchjob, wat er precies verwerkt is en indien van toepassing wat er fout ging.

Over de jaren zijn er binnen APG een groot aantal batchjobs ontwikkeld door verschillende teams. Er waren voorheen geen richtlijnen voor batchjob ontwikkeling waardoor er een wildgroei aan batchjobs ontstaan is die allemaal op een eigen wijze loggen en foutafhandeling verzorgen. Hierdoor is het erg lastig om in het geval van productie incidenten snel oorzaken van fouten te traceren en op te lossen.

Om deze problemen aan te pakken is ervoor gekozen om een standaard voor *.NET* batchjob ontwikkeling op te stellen. Deze standaard richt zich met name op regels en richtlijnen omtrent foutafhandeling en logging.

Om de standaard te ondersteunen en om de ontwikkeltijd van batchjobs te verkorten is er een *.NET framework* ontwikkeld dat allerlei generieke functionaliteit die alle of veel batchjobs delen te implementeren. Het framework is zo generiek mogelijk opgezet zodat het binnen de gehele organisatie ingezet zou kunnen worden.

Het framework biedt onder andere functionaliteiten voor logging en foutafhandeling en pakt op die manier de huidige problematiek aan. Het framework kan nu zorg dragen voor een gestructureerde logging output die qua formaat voor elke batchjob gelijk is.

Als test case voor het framework is een bestaande batchjob van de klant Loyalis Maatwerk Administraties herbouwd op basis van het framework. De testcase heeft de vele voordelen van het framework aangetoond. Zo levert de logging bruikbare, duidelijke foutmeldingen op en is net gestructureerd, is de omvang van de code met 33% afgenomen en de performance bij een realistische testrun met 47,6% toegenomen.

Summary

This internship is performed at APG, located in the city of Heerlen. APG is specialized in group pensions in the public sector. APG has an internal IT company named CIS which responsibilities include application management and development.

This internship focused on improving quality of *logging* by *batchjobs*. Batchjobs are computer programs which process large amounts of information without any human interaction. Information processed by batchjobs is usually collected during office hours by gathering input from various computer programs and is processed at a later point in time, usually outside office hours. The only form of communication between batchjobs and humans is through log files. Log files contain diagnostic information about batchjob execution, processed information and error reports if applicable.

Over the years a lot of batchjobs have been developed at APG by various development teams. There used to be no guidelines for batchjob development, causing a proliferation amongst batchjobs. This is particularly problematic for logging and error handling since various logging formats used by batchjobs make it very hard to track down error causes in the event of a production incident.

To address these issues a standard for batchjob development has been developed. This standard focuses on rules and guidelines concerning error handling and logging.

To support the standard and to speed up batchjob development a *.NET framework* is developed containing lots of features found in most if not all batchjobs. The framework is designed to be generic to be able to be used throughout the entire company.

The framework includes features like logging and error handling and addresses problems outlined above. Since the framework handles all logging in batchjobs, it can provide a decent, structured logging output and enforce extended error reports.

An existing batchjob from Loyalis Maatwerk Administraties, a customer of APG, was rebuild on top of the new batch framework and was used as a test case. The rebuild clearly showed major advantages in using the framework. The logging output contains useful, extended error reports and is well structured. The number of lines of code was reduced by 33% and performance increased by 47,6% during a realistic test run.

Verklarende woordenlijst

Begrip	Verklaring
Batchjob	Een computerprogramma dat een verzameling gegevens verwerkt zonder menselijke interventie. De gegevens worden vaak tijdens kantooruren verzameld door input vanuit verschillende applicaties en op een later tijdstip, bijvoorbeeld buiten kantooruren, verwerkt.
Logging	Het registreren van informatieve meldingen over de uitvoering van een computerprogramma. Denk hierbij bijvoorbeeld aan informatie over een fout die optreedt tijdens de uitvoering van het programma.
Technische fout in batchverwerking	Een fout met een technische oorzaak; bijvoorbeeld een bug in de code of een server systeem dat niet reageert. De fout wordt niet veroorzaakt door de eindgebruiker.
Functionele fout in batchverwerking	Een fout die wordt veroorzaakt door een gebruiker zelf, bijvoorbeeld doordat informatie die verwerkt moet worden incompleet of ongeldig is.
.NET	Een applicatieplatform van Microsoft.
VB.NET	Programmeertaal.
LMA	Loyalis Maatwerk Administraties, een klant van APG.
AOC	Applicatie Onderhoud Cluster, afdeling binnen APG / CIS waar de stage is uitgevoerd.
Framework	Een verzameling van programma code die een set aan generieke functionaliteiten biedt waar applicaties gebruik van kunnen maken.
Unit test	Tests die zich richten op het aantonen van correcte werking van (delen van) software. Deze tests worden op enig moment opgesteld en kunnen in het vervolg volledig geautomatiseerd uitgevoerd worden.
Visual Studio	Ontwikkelomgeving van Microsoft voor onder andere VB.NET.
Boilerplate	Code die vaak voorkomt in verschillende applicaties.
Stored procedure	Code in een database om gegevens op te halen of te manipuleren.
Visual Studio solution	Verzameling van Visual Studio projecten.
Typed Dataset	Onderdeel van het .NET framework dat een eenvoudige manier biedt om met informatie uit databases te werken.
Gegenereerde domein objecten	Gegenereerde domein objecten zijn VB.NET classes die door een tool gegenereerd worden op basis van Typed Datasets.
Logging provider	Een stuk software dat logging verwerkt, bijvoorbeeld naar de harde schijf of een database wegschrijft.

Mock objecten	Mock objecten zijn stukken code die het gedrag van een onderdeel uit het system nabootsen, “mocken”. Deze objecten worden gebruikt om onafhankelijk van randsystemen unit tests uit te kunnen voeren.
PIO	Proces Inrichting en Ondersteuning. PIO is een overkoepelende afdeling binnen APG voor alle ontwikkelafdelingen waar standaarden en generieke componenten worden ondergebracht.
Bootstrappen	Bootstrappen is het gereedmaken en instellen van alle benodigde configuratie en omgevingsinstellingen van een applicatie. Denk bijvoorbeeld aan het initialiseren van database verbindingen, logging etc.

1. Inleiding

APG is onderdeel van de APG Groep en specialist op het gebied van collectieve pensioenen in de publieke sector. Concern Information Systems is het interne IT bedrijf van APG dat zich onder meer bezig houdt met software beheer en ontwikkeling.

De software infrastructuur van Loyalis, een klant van APG, maakt veelvuldig gebruik van *batchjobs*. Batchjobs zijn computerprogramma's die een verzameling gegevens verwerken zonder menselijke interactie. Deze gegevens worden vaak tijdens kantooruren verzameld door input vanuit verschillende applicaties en op een later tijdstip, bijvoorbeeld buiten kantooruren, verwerkt.

Deze batchjobs zijn in de loop der jaren ontwikkeld door verschillende ontwikkelaars en projectgroepen en missen uniformiteit op het gebied van *logging*, foutafhandeling en applicatie ontwerp.

Deze situatie is nadelig voor alle betrokken partijen (de klant, de ontwikkelaars en de productiebeheerders, zie paragraaf 3.3). De wens bestaat dan ook om deze situatie aan te pakken door het opstellen van regels en richtlijnen voor *.NET* batchjob ontwikkeling.

Er zal onderzoek gedaan worden naar de wensen en eisen van betrokken partijen en naar mogelijkheden om de standaard op technisch vlak te ondersteunen. Ten slotte zal er een oplossing bedacht en ontwikkeld worden.

In hoofdstuk 2 is meer informatie te vinden over het bedrijf, in hoofdstuk 3 zijn de details van de opdracht beschreven, in hoofdstukken 4 en 5 zijn de werkzaamheden, uitdagingen en resultaten beschreven en in hoofdstuk 6 kunnen de conclusies teruggevonden worden.

2. Het bedrijf

APG is onderdeel van de APG Groep N.V. De onderneming heeft vestigingen in Heerlen, Amsterdam, New York en Hongkong, beheert ongeveer € 200 miljard pensioenvermogen en bedient zo'n 4 miljoen deelnemers. Hierbij zijn wereldwijd ongeveer 4 000 medewerkers betrokken.

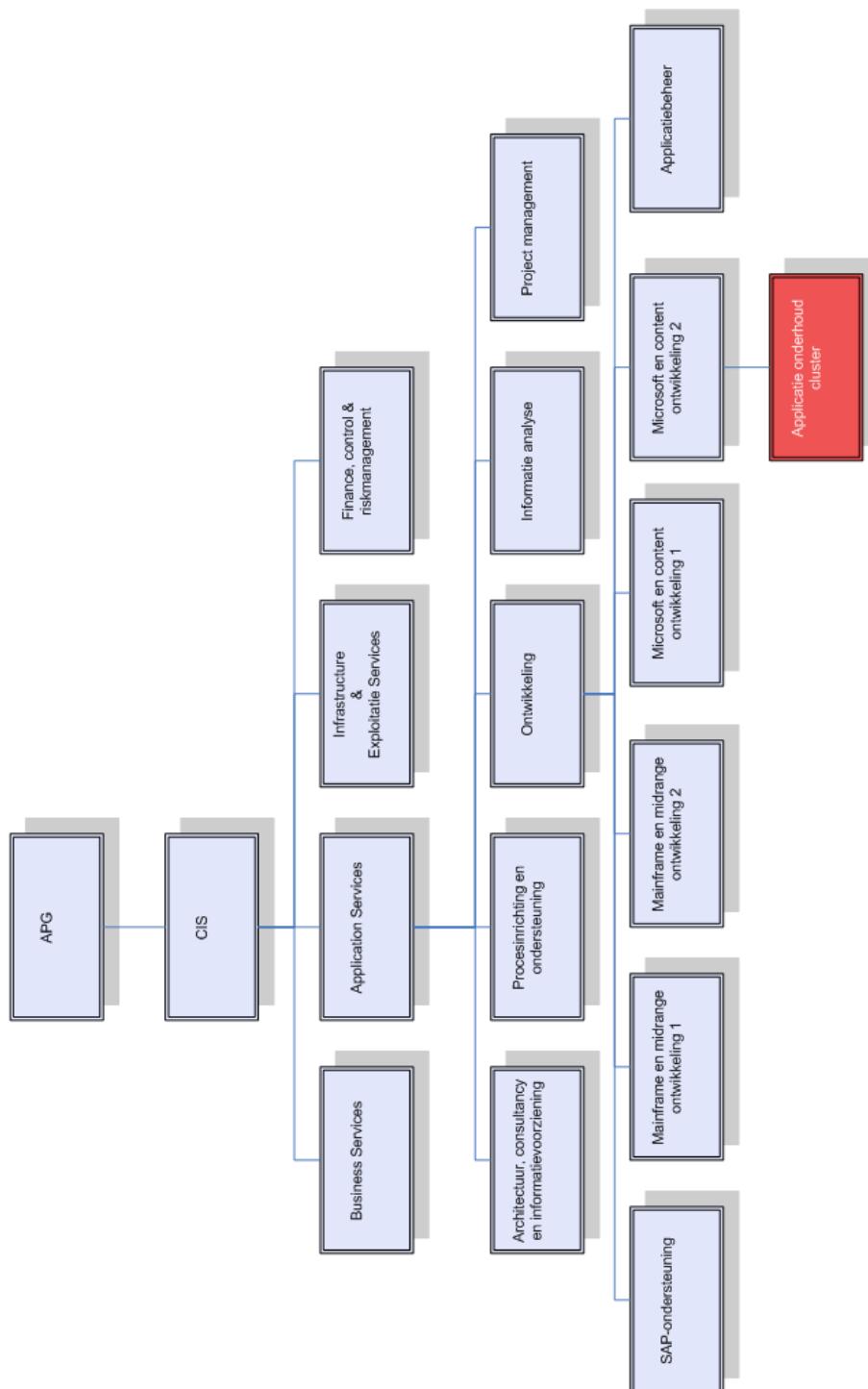
APG Groep N.V. heeft drie dochters: APG, Cordares en Loyalis.



APG is de uitvoerende tak van de organisatie. APG is specialist op het gebied van collectieve pensioenen in de publieke sector. APG verzorgt de uitvoering van het pensioen voor circa 2,8 miljoen Nederlanders. Dat betekent dat APG de administratie voert, de communicatie verzorgt en het pensioenvermogen van die sector beheert.

APG is ontstaan in 2008. In dit jaar heeft ABP de uitvoering van de pensioenregeling ondergebracht in de zelfstandige onderneming APG. Deze splitsing was nodig omdat het ABP via Cordares en Loyalis verzekeringen aanbood en aanbiedt en zo concurreerde met andere verzekeringsmaatschappijen. Deze maatschappijen beschouwden dat als oneerlijke concurrentie omdat het ABP een groot aantal 'gedwongen' klanten heeft. Nu wordt het verplichte pensioen door een andere organisatie uitgevoerd dan de vrijwillige verzekeringen.

De stage is uitgevoerd op de afdeling Concern Information Systems, binnen het team Applicatie Onderhoud Cluster (AOC) dat onderdeel is van de afdeling Microsoft & Content Ontwikkeling. Deze afdeling is verantwoordelijk voor het onderhoud van applicaties. Hierbij hoort ook aanbouw van nieuwe features aan systemen. In figuur 2-1 is een vereenvoudigde organisatiestructuur te zien, waarbij de afdeling AOC gemarkeerd is.



Figuur 2-1 Organisatiestructuur (vereenvoudigd)

3. De opdracht

3.1. Situatie en probleemstelling

De software infrastructuur van Loyalis Maatwerk Administraties (LMA), een klant van APG, maakt veelvuldig gebruik van batchjobs. Er zijn binnen LMA momenteel een kleine dertig batchjobs in gebruik. Deze batchjobs zijn in de loop der jaren ontwikkeld door verschillende ontwikkelaars en projectgroepen en missen uniformiteit op het gebied van logging, foutafhandeling en applicatie ontwerp. Dit levert een aantal problemen op:

- Wanneer er een fout optreedt tijdens de uitvoering van een batchjob is het vaak lastig te traceren waar in de applicatie deze precies optreedt. Veel batchjobs vangen fouten af maar loggen hier geen of weinig informatie over. Elke batchjob is zelf verantwoordelijk voor het implementeren van logging functionaliteit, wat tot gevolg heeft dat verschillende batchjobs verschillende hoeveelheden informatie loggen in verschillende formaten.

Het gevolg hiervan is dat foutmeldingen niet efficiënt afgehandeld kunnen worden. Omdat foutmelding berichten vaak technisch van aard zijn worden deze in veel gevallen direct doorgestuurd naar ontwikkelaars. Door ontbrekende informatie en het gebrek aan uniformiteit zijn ontwikkelaars te veel tijd kwijt aan het traceren van fouten wat een langere oplostijd en dus kosten met zich meebrengt. Een te lange oplostijd van incidentafhandeling is nadelig voor de klant omdat deze afhankelijk is van de resultaten van batchjobs. Als het lang duurt voor een incident afgehandeld is kunnen medewerkers van de klant vaak niet vooruit met hun werk.

Ook blijkt vaak na enig onderzoek dat de fouten een functionele oorzaak hebben, bijvoorbeeld verkeerd ingevulde invoer informatie. Idealiter zouden deze foutmeldingen in het functionele logging verslag terecht komen, zodat de foutmeldingen direct door een functioneel beheerder afgehandeld kunnen worden, zonder eerst langs een ontwikkelaar te moeten. Batchjobs moeten dus zo veel mogelijk zelf onderscheid kunnen maken tussen fouten met een technische- en fouten met een functionele oorzaak.

- De fragmentatie op het gebied van batchjob ontwikkeling komt de beheerbaarheid niet ten goede. Wanneer er een bug opgelost zou worden in een stuk algemene code, bijvoorbeeld logging code, dan moeten alle batchjobs bekeken worden of deze dezelfde code bevatten en als dit het geval is de oplossing handmatig doorvoeren. In de praktijk gebeurt dit natuurlijk zelfden of nooit, waardoor de fragmentatie alleen maar groter wordt. Een *framework* zou hierbij uitkomst bieden omdat bugs of wijzigingen in generieke functionaliteit dan maar op één plek aangepast hoeven te worden.
- Batchjobs maken gebruik van gegevens uit verschillende soorten databases. Deze databases generen elk hun eigen unieke foutmeldingen, waardoor er dus verschillende foutmeldingen bestaan voor dezelfde fouten. Degene die een foutmelding onderzoekt moet dus een terdege kennis van de gebruikte databases hebben om de foutmelding überhaupt te kunnen interpreteren.

3.2. Doelstellingen

3.2.1. Hoofddoelstelling

Het verkorten van de oplostijd van incidentafhandeling bij batchjobs door het gelijk trekken van logging output en foutafhandeling van batchjobs.

3.2.2. Subdoelstellingen

1. Het automatisch kunnen classificeren van foutmeldingen op basis van een technische of functionele oorzaak.
2. Het verkorten van de ontwikkeltijd van batchjobs door zoveel mogelijk functionaliteit die voor elke batchjob gelijk is in een *framework* te verwerken.
3. Dit framework zo generiek mogelijk opzetten zodat het breed inzetbaar is binnen de organisatie.
4. Fragmentatie in batchjobs in de toekomst beperken/voorkomen.

3.2.3. Het bereiken van de doelstellingen

Om bovenstaande doelstellingen te bereiken zal er een standaard opgezet worden voor batchjob ontwikkeling. Hierin zullen een aantal richtlijnen worden opgenomen die fragmentatie in de toekomst moeten voorkomen. Tevens zal er een vast formaat voor logging output worden vastgelegd.

Om deze standaard te ondersteunen en om de overige doelstellingen te behalen zal er een batch *framework* ontwikkeld worden dat voorziet in logging, foutafhandeling en alle overige functionaliteiten die batchjobs delen. Het framework zal zoveel mogelijk automatisch moeten kunnen bepalen of een fout een technische- of functionele oorzaak heeft om zo het afhandelen van fouten zo efficiënt mogelijk te kunnen laten verlopen.

Om subdoelstelling 3 te kunnen behalen zal er veel aandacht besteedt worden aan het ontwerp van het framework om het zo flexibel mogelijk te maken. Er zal gewaakt moeten worden voor een goede balans tussen functionaliteit, eenvoud en flexibiliteit.

Als het gewenste resultaat behaald wordt zullen batchjob ontwikkelaars in de toekomst enkel nog aandacht hoeven te besteden aan de specifieke functionaliteit van een batchjob. Dit zal de ontwikkeltijd en beheerbaarheid van batchjobs sterk ten goede komen. De huidige problematiek zal dan tot het verleden behoren.

3.3. De opdrachtgever en betrokken partijen

De opdrachtgever van dit project is Jos Philippen. Jos is de projectleider van AOC, de afdeling waar deze stage is uitgevoerd. Jos heeft dit project opgestart na het signaleren van de huidige problematiek door LMA en afdeling productiebeheer.

Er zijn bij dit project verschillende partijen betrokken en zij hebben allen baat bij een succesvolle afloop van het project.

Hieronder volgt een overzicht van de bij dit project betrokken partijen:



3.4. Projectorganisatie

In de dagelijkse uitvoering van het project is Jeroen van den Boorn degene die de meeste sturing geeft aan het project. Jos Philippen overziet het project van de zijlijn en richt zich met name op de begeleiding van het proces. Dit komt naar voren in de periodieke voortgangsgesprekken.

De opdracht en doelstellingen zijn helder, de meeste sturing vindt dan ook op technisch vlak plaats. De sturing is er met name op gericht om de oplossing zo goed mogelijk op de organisatie aan te laten sluiten. Het project kent een aantal harde technische requirements, maar binnen dit kader krijgt de projectnemer veel vrijheid bij het uitwerken van de oplossing.

3.5. Eindproducten

De eindproducten bestaan uit de volgende onderdelen:

- Een functionele standaard voor .NET batchjob ontwikkeling waarin richtlijnen en best practices beschreven worden.
- Een .NET *framework* dat ontwikkelaars helpt de standaard na te leven. Gebruik van dit framework helpt niet alleen bij het naleven van de standaard maar neemt ontwikkelaars ook veel werk uit handen doordat het veel generieke functionaliteit bevat die niet meer in elke batchjob zelf geïmplementeerd hoeft te worden.
Dit framework zal in VB.NET ontwikkeld worden.
- Een gebruikershandleiding voor het framework. Hierin zal beschreven worden hoe het framework in elkaar zit, hoe het werkt en hoe het optimaal ingezet kan worden.
- Documentatie van het framework. Dit document wordt gegenereerd op basis van het commentaar in de code. Alle onderdelen en functies van het framework worden hierin beschreven en toegelicht.
- Een *Visual Studio* template waarmee met een aantal klikken een nieuw batchjob project aangemaakt kan worden. Deze template maakt de benodigde bestanden voor een batchjob aan en legt een koppeling naar het batch framework.
- Een op basis van het framework geschreven en geteste implementatie van batchjob "LRS_B21".
LRS_B21 is één van de batchjobs binnen het applicatielandschap van LMA en is verantwoordelijk voor het bepalen van cliënten die in aanmerking komen voor het ontvangen van uitkering gerelateerde formulieren.
Deze specifieke batchjob is door de projectleiding gekozen omdat dit voor de klant een belangrijke batchjob is waaraan met name op gebied van logging veel te wensen over is. Tevens is deze batchjob technisch sterk verouderd (geen object oriented opzet, slechte scheiding tussen verschillende software lagen etc.) en omvat deze een redelijke hoeveelheid functionaliteit die past binnen de tijdsspanne van dit project.

Door deze batchjob te herbouwen op basis van het batch framework kan er al vroegtijdig inzicht verkregen worden in hoe het batch framework in de praktijk werkt. Deze informatie kan vervolgens gebruikt worden om het framework te optimaliseren.

Deze batchjob zal net als het framework in VB.NET ontwikkeld worden.

Aan het eind van de stage worden deze producten overgedragen aan collega's van de afdeling MCO/AOC, die het geheel in onderhoud zullen nemen. Bij een volgende release van de batchjobs zullen het framework en batchjob B21 in productie worden genomen.

3.6. Onderzoek

Om een goed beeld te krijgen van de probleemstelling, de achtergrond hiervan en eventuele oplossingen is er uitgebreid onderzoek gedaan naar een aantal aspecten.

Zo is allereerst in kaart gebracht wat de wensen van de verschillende betrokken partijen waren en wat de aanleiding was om dit project te starten. Daarna is onderzocht welke mogelijke oplossingen er voor deze problemen waren.

Hieruit zijn de in de volgende paragraaf beschreven onderzoeksvragen gedestilleerd.

3.6.1. Onderzoeksvragen en resultaten

Welke problemen zijn er met de huidige batchjobs?

- a. Welke gevolgen heeft dit en waarom is dit een probleem?

Voor de probleemstelling en gevolgen zie paragraaf 3.1.

De huidige situatie is problematisch omdat een langere oplostijd van incident afhandeling tot gevolg heeft dat medewerkers van de klant niet vooruit kunnen met hun werk. De resultaten van batchjobs moeten door medewerkers van de klant verwerkt worden. Wanneer een batch bij de uitvoering onverwacht is afgebroken moet eerst de fout opgespoord worden en vervolgens kan de batch opnieuw gedraaid worden. Daarna kunnen de medewerkers pas weer aan de slag.

Daarnaast zijn er interne SLA's waar aan voldaan moet worden. Doordat incidenten door de beschreven problemen nu vaak een te lange oplostijd hebben wordt het behalen van de afspraken uit deze SLA's bemoeilijkt.

Wat kan er gedaan worden om de huidige problematiek op te lossen?

Het voornaamste probleem is dat er nu geen duidelijke standaard of richtlijnen aanwezig zijn voor foutafhandeling en logging in batchjobs. Er zijn globale applicatie ontwikkelrichtlijnen, maar deze zijn niet specifiek genoeg om de huidige problemen tegen te gaan.

Om de huidige problematiek in de toekomst te voorkomen is het belangrijk om duidelijke, specifieke richtlijnen voor batchjob ontwikkeling op te stellen. Om te voorkomen dat verschillende interpretaties van de richtlijnen alsnog leiden tot technisch zeer verschillende batchjobs en ook om het voor ontwikkelaars eenvoudig te maken om de richtlijnen na te leven kan er een framework ontwikkeld worden dat het implementeren van de standaard stroomlijnt.

Welke partijen hebben welke belangen bij dit project?

Er zijn een aantal belanghebbende partijen in dit project. Er zijn geen tegenstrijdige belangen, maar elke partij heeft wel eigen belangen.

- *LMA*, de klant
Voor de klant is het zeer wenselijk dat het project slaagt omdat zij direct hinder ervaren van de huidige problematiek, bijvoorbeeld doordat foutverslagen onduidelijke of geen foutomschrijvingen bevatten. Als de doelstellingen van dit project behaald worden zullen productie incidenten sneller afgehandeld kunnen worden waardoor de impact van deze incidenten verkleind wordt.
- Afdeling productiebeheer
De afdeling productiebeheer is ook gebaat bij beter gestructureerde logging output en duidelijke foutomschrijvingen omdat dit het makkelijker maakt om te vast te stellen wie de foutmelding het beste kan oplossen.

- Ontwikkelaars
Ontwikkelaars hebben veel belang bij het slagen van dit project. Het wordt voor ontwikkelaars makkelijker om problemen te traceren omdat foutmeldingen meer relevante informatie zullen bevatten en omdat alle batchjobs dezelfde code structuur zullen hebben. Er hoeft dus niet eerst meer uitgezocht hoe deze batchjob nu weer in elkaar steekt.
Verder zal het ontwikkelen van batchjobs een stuk aangenamer worden doordat er minder saai en herhalend werk uitgevoerd hoeft te worden. Het framework zal namelijk de generieke functionaliteit die batchjobs delen implementeren. Een ontwikkelaar hoeft zich daarom enkel te richten op het stuk functionaliteit die iedere batchjob uniek maakt.

Welke informatie moeten foutmeldingen bevatten in het geval van:

- a. Functionele foutmeldingen in batch logging?
Het is met name belangrijk dat er in dit geval vermeld wordt bij welke situatie de fout optreedt. Bijvoorbeeld bij het behandelen van welke aanvraag, welke cliënt hierbij hoort etc. Daarnaast is het belangrijk dat vermeld wordt wat er precies fout gaat. De meeste functionele foutmeldingen zullen voortkomen uit het testen van condities in de code, de ontwikkelaar heeft dus de gelegenheid om tijdens de ontwikkeling te zorgen dat er een heldere, to the point geformuleerde foutmelding getoond wordt.
- b. Technische foutmeldingen in batch logging?
Ook hierbij is het belangrijk dat er vermeldt wordt om situatie het gaat. Met behulp van deze informatie kan de foutmelding eenvoudig gereproduceerd worden door de ontwikkelaar. Tevens is het belangrijk dat in het technisch log verslag vastgelegd wordt waar de fout precies vandaan komt, bijvoorbeeld door het loggen van de *stacktrace*. Een *stacktrace* wordt altijd door .NET aangeleverd als er een fout optreedt en bevat uitgebreide informatie over waar de fout precies opgetreden is.

Hoe kunnen fouten uit verschillende database producten generiek afgehandeld worden?

Het is wenselijk dat foutmeldingen van verschillende database producten automatisch vertaald kunnen worden, ook als deze foutmeldingen niet bekend zijn in het batch framework. Uit het onderzoek is gebleken dat dit echter niet mogelijk is. De foutmeldingen uit de verschillende database producten die onderzocht zijn delen te weinig gemeenschappelijke informatie om automatisch af te kunnen handelen.

Foutmeldingen van Oracle en MS SQL zijn in beide gevallen genummerd (zie bron B3) en onderverdeeld in verschillende categorieën/groepen. Helaas is echter gebleken dat de groepen die gehanteerd worden totaal niet aansluiten op wat in het project verwacht wordt van foutmeldingen; namelijk het kunnen vaststellen van een functionele dan wel technische oorzaak van foutmeldingen.

MS SQL server maakt bijvoorbeeld onderscheid tussen vier typen foutmeldingen. Twee van deze typen kunnen automatisch als technisch bestempeld worden, maar de foutmeldingen uit de twee andere typen kunnen allerlei oorzaken hebben.

Oracle maakt ook onderscheid tussen verschillende typen foutmeldingen, maar ook hier valt niet uit op te maken of deze foutmeldingen een technische of functionele oorzaak hebben.

Om toch foutmeldingen te kunnen vertalen is er gekozen voor een semiautomatische aanpak. Binnen het batch framework zijn een aantal generieke foutmeldingen gedefinieerd die vaak voorkomen. Hierbij is aangegeven of deze foutmeldingen een technische of functionele oorzaak hebben. Aan deze generieke foutmeldingen zijn vervolgens foutcodes uit verschillende database producten gekoppeld. Op het moment dat er een foutmelding optreedt bij een database operatie kan het framework kijken of deze foutmeldingen bekend is en als dat het geval is kan de foutmeldingen vertaald worden naar een generieke foutmelding. Op basis van het type van de generieke foutmelding kan vervolgens bepaald worden in welke log verslag (technisch of functioneel) de foutmelding vermeldt moet worden.

Het voordeel hiervan is dat de kans dat een foutmelding foutief geclassificeerd wordt zeer klein is, het nadeel is dat indien een foutmelding niet bekend is bij het framework deze ook niet vertaald kan worden. Er is voor gekozen om in dit geval de foutmelding als een technische fout te behandelen omdat foutmeldingen vaak technisch ingewikkeld omschreven zijn en een ontwikkelaar hier beter mee om kan gaan dan een functioneel beheerder.

4. Werkzaamheden

Het project is opgedeeld in verschillende fasen volgens de waterval methode. Deze fasen zullen in onderstaande paragrafen beschreven worden.

4.1. Definitiestudie en onderzoek

Deze fase stond in het kader van het in kaart brengen van de wensen en eisen van de bij dit project betrokken partijen. Veel informatie is verkregen via Jeroen, de inhoudelijke en technische bedrijfsbegeleider. Jeroen is erg goed op de hoogte van de problemen die spelen, de betrokken partijen en de systemen waar de problemen betrekking op hebben.

Tevens hebben er gesprekken plaatsgevonden met alle betrokken partijen, om van hen zelf te horen waar voor hen de grootste knelpunten zitten en wat voor hen belangrijke eisen aan het eindproduct zijn. Uit de gesprekken met de betrokken partijen kwamen de knelpunten die Jeroen al genoemd had duidelijk naar voren. Daarnaast had met name LMA, de eindgebruiker, nog een aantal kleine aanvullende wensen.

Uiteraard is ook de huidige situatie uitgebreid bestudeerd. De opzet van de huidige systemen, de broncode van de huidige batchjobs, gebruikte databases etc. In deze periode heeft er ook veel overleg met Jeroen plaatsgevonden over de huidige systemen; soms voor het vragen van verduidelijking over systeemonderdelen, of over waarom een onderdeel op de gebruikte manier in elkaar zit. Door het verzamelen van deze informatie was het mogelijk een goede oplossing voor de problemen te bedenken die goed paste binnen het applicatie landschap en de werkwijzen en gebruiken van de organisatie.

4.2. Standaard voor batchjob ontwikkeling

Op basis van de in de definitiestudie vergaarde informatie is in deze fase de standaard voor batchjob ontwikkeling opgesteld. De standaard omvat regels en richtlijnen die zich hoofdzakelijk richten op de huidige knelpunten. Ook bevat het document verwijzingen naar onderdelen van de algemene .NET ontwikkelrichtlijnen van APG die van toepassing zijn op batchjobs.

Daarnaast wordt in dit document een standaard structuur voor batchjobs beschreven en zijn er werkinstructies opgenomen voor werken met het bijbehorende batch framework.

4.3. Functioneel ontwerp

4.3.1. Batch framework

Op basis van de in de definitiestudie vergaarde informatie is er in deze fase een functioneel ontwerp opgesteld voor het batch framework. In dit document staat beschreven welke onderdelen er in het batch framework worden opgenomen en waar ze voor dienen. Tevens zijn er een aantal use cases opgesteld.

4.3.2. Batchjob B21

Er was al een functioneel ontwerp van B21 aanwezig, aangezien dit onderdeel geen nieuwbouw maar herbouw betrof.

Om het systeem goed te kunnen doorgronden zijn er op basis van het functioneel ontwerp Activity Diagrams, Use Cases en een conceptueel klassendiagram gemaakt. Het maken van deze documenten verschaftte veel inzicht in het systeem en de randsystemen waar deze applicatie afhankelijk van is.

4.4. Technisch ontwerp

4.4.1. Batch framework

Om een optimaal technisch ontwerp te kunnen realiseren is er onderzoek gedaan naar de mogelijkheden van het .NET framework. Tevens zijn de interne ontwikkel richtlijnen doorgenomen en is er overlegd met de bedrijfsbegeleider om inzicht te krijgen in welke technieken en ontwerpstandaarden gangbaar zijn binnen APG.

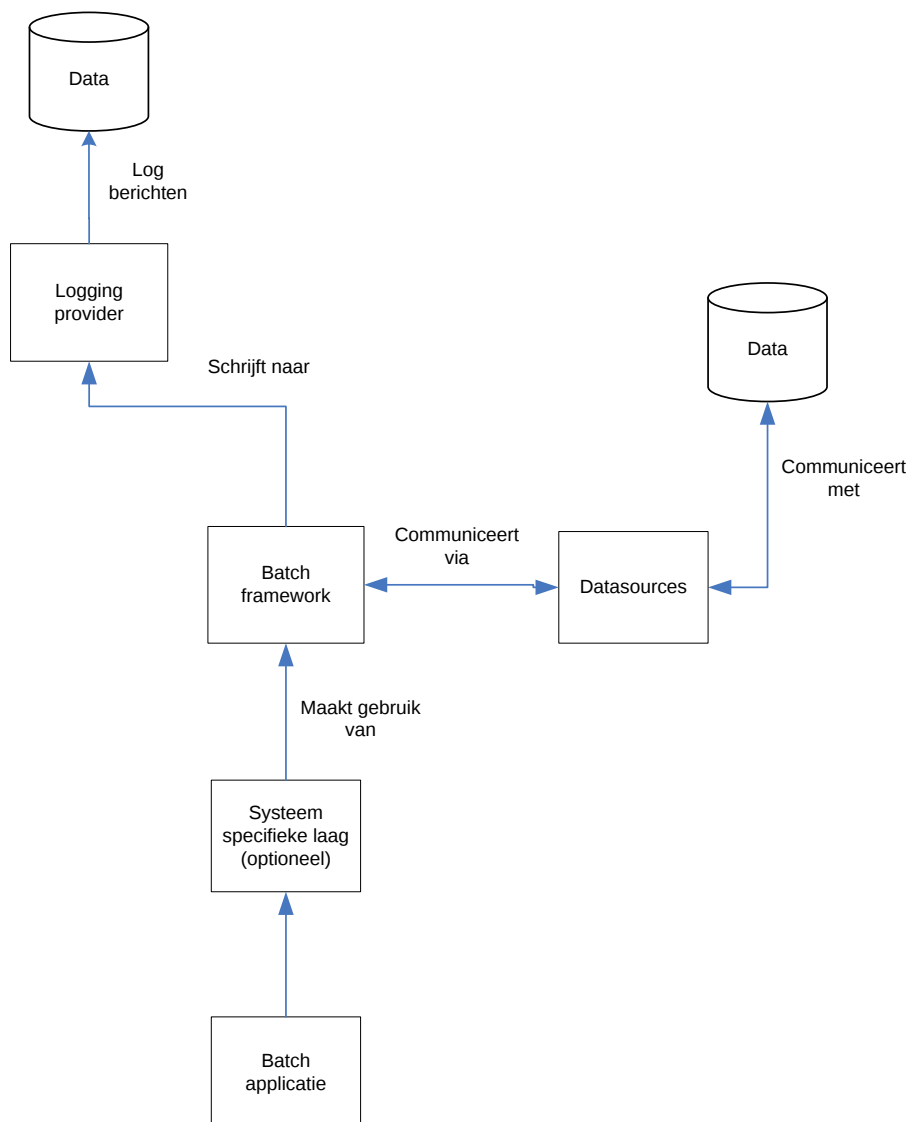
Op basis van deze informatie en het functioneel ontwerp is vervolgens een technisch ontwerp gemaakt. Dit technisch ontwerp bestaat uit de volgende onderdelen:

- Architectuur beschrijving
 - o Hierin is beschreven uit welke componenten het framework bestaat, met welke externe componenten het batch framework communiceert en hoe een batchapplicatie gebruikt maakt van het batch framework.
- Domein model

Na de afronding van het initiële technisch ontwerp is besloten het ontwerp te wijzigen zodat het framework nog breder inzetbaar werd. De voornaamste wijziging was dat de afhankelijkheid van externe systemen volledig geëlimineerd is door via interfaces te communiceren met externe componenten. Het framework moet immers zo breed mogelijk bruikbaar zijn dus het is niet wenselijk dat het framework afhankelijk is van het opslaan van gegevens in bijvoorbeeld een Oracle database.

De interfaces voor communicatie naar de externe componenten kunnen daarom buiten het framework geïmplementeerd worden in de batchjobs zelf bijvoorbeeld, of in een laag die tussen het framework en de batchjobs geplaatst kan worden.

Schematisch ziet het framework er als volgt uit:



Te zien is dat een batch applicatie, eventueel via een systeem specifieke laag, gebruikt maakt van het framework. De systeem specifieke laag is een laag over het framework heen waarin functionaliteiten die niet in alle batchjobs voorkomen, maar enkel in batchjobs die bij bepaalde systemen horen, te implementeren.

Tevens is te zien dat het framework via een zogeheten *logging provider* logging wegschrijft naar een datastore. De datastore zal in de meeste gevallen het bestandssysteem van een server zijn, maar door de flexibele architectuur van het framework kan dit elke vorm van opslag zijn. Het framework wordt geleverd met één logging provider op basis van het Microsoft Enterprise Library framework (zie hoofdstuk 5.1.2.2), maar logging providers kunnen ook vanuit een batch zelf of vanuit de systeem specifieke laag worden geleverd.

Het framework communiceert op een vergelijkbare manier via Datasources met externe datastores voor data opslag. In de meeste gevallen zal dit een database betreffen, maar een datasource kan ook het bestandssysteem of een webservice vertegenwoordigen.

4.4.2. Batchjob B21

Het technisch ontwerp van B21 bestaat uit enkel een domeinmodel. De oude implementatie was niet object oriented dus in dit opzicht verschilt de nieuwe implementatie wezenlijk. B21 is technisch verder niet complex. Een batch applicatie doet niets anders dan een berg informatie ophalen, verwerken en resultaten opslaan. Er zijn een aantal domein objecten die gebruikt worden om informatie uit de database te vertegenwoordigen. Er zijn verder geen complexe relaties. Het ontwerp is dan ook erg eenvoudig gebleven.

4.5. Ontwikkeling

4.5.1. Batch framework

De ontwikkeling fase is gestart met de ontwikkeling van het batch framework. Het ontwikkelingsproces van het framework is een iteratief proces geworden. Als eerste is het systeem ontwikkeld zoals het ontworpen was. Daarna is batchjob B21 geïmplementeerd op basis van het framework. Tijdens het ontwikkelen van B21 bleek dat er nog een aantal functionaliteiten waren die ook generiek geïmplementeerd konden worden en dat een aantal functionaliteiten beter iets gewijzigd konden worden. Er zijn daarom tijdens de ontwikkeling van B21 nog een aantal wijzigingen en toevoegingen gedaan aan het framework. De wijzigingen waren kleinschalig, aan de architectuur is niets gewijzigd. Het was interessant om te zien hoe tijdens het ontwikkelen van B21 nieuwe inzichten verkregen werden waarmee het framework verbeterd kon worden.

Het batch framework bestaat grofweg uit drie onderdelen: het implementeren van generieke functionaliteiten, logging en foutafhandeling.

4.5.1.1. Generieke functionaliteiten

Generieke functionaliteiten zijn die functionaliteiten die in elke batchjob gebruikt worden. Voorbeelden zijn stuurparameters, herstart informatie (informatie over de voortgang van een batchjob zodat deze de uitvoering kan hervatten op het moment dat de uitvoering tussentijds gestopt wordt), maar ook *boilerplate* code voor het bijvoorbeeld het kunnen starten, *bootstrappen* en stoppen van een batchjob.

In de huidige situatie wordt code voor al deze functionaliteiten telkens gekopieerd naar elke batchjob. Vaak worden er dan weer kleine aanpassingen gemaakt aan deze functionaliteiten.

Door het implementeren van deze functionaliteiten in een framework wordt er voorkomen dat elke batchjob deze code moet bevatten en dat er tientallen verschillende implementaties bestaan van deze functionaliteiten.

4.5.1.2. Logging

Zoals beschreven in hoofdstuk 3.1 is één van de grootste problemen in de huidige situatie het ontbreken van uniformiteit in logging van batchjobs. In de standaard zijn daarom richtlijnen hierover opgenomen. Om nog één stap verder te gaan is er in het framework een logging module opgenomen. Door het gebruik van deze module kan automatisch afgedwongen worden dat er in één uniform formaat gelogd wordt. Daarnaast maakt deze module het ontwikkelaars een stuk eenvoudiger om degelijke logging te implementeren in batchjobs. Op deze manier wordt het voor ontwikkelaars ook aantrekkelijk om het framework te gebruiken.

4.5.1.3. Foutafhandeling

Één van de wensen van de opdrachtgever was dat het framework zelf in staat zou zijn fouten tijdens de uitvoering van een batchjob te classificeren in “technische oorzaak” en “functionele (niet-technische) oorzaak”. Dit bleek helaas niet volledig uitvoerbaar, zie hoofdstuk 3.6.1 onderzoeksvraag “Hoe kunnen fouten uit verschillende database producten generiek afgehandeld worden?”.

4.5.2. Batchjob B21

De ontwikkeling van B21 stelde op technisch vlak weinig voor, maar op functioneel vlak des te meer. De applicatie bestaat bijna volledig uit functionele afhandeling van allerlei gegevens. Dit was heel anders programmeren dan ik tot nu toe gewend was. Doordat het systeem een stuk grootschaliger en complexer was dan waar ik tot nu toe aan gewerkt had was ik veel meer aangewezen op het functioneel ontwerp dan ooit te voren. Het functioneel ontwerp vormde echt de leidraad bij de ontwikkeling.

Door het werken aan een systeem van deze omvang en met een dergelijke hoeveelheid complexe materie werd het belang van een goed functioneel ontwerp en systeemdokumentatie goed duidelijk.

4.5.3. Visual Studio template

Om het framework eenvoudig inzetbaar te maken is er een Visual Studio template opgezet dat een project aanmaakt voor een nieuwe batchjob die gebruikt maakt van het framework. Hierdoor kan binnen enkele minuten een nieuwe batchjob aangemaakt worden die volledig aan de standaard voldoet.

4.6. Testen

4.6.1. Batch framework

Het framework is technisch getest door middel van *unit tests*. Doordat het framework flexibel opgezet is met losse componenten was het systeem goed te unit testen.

Om unit tests uit te kunnen voeren zonder afhankelijk te zijn van externe systemen zoals databases is het soms nodig om gebruik te maken van zogeheten *mock objecten*. Mock objecten zijn stukken code die het gedrag van een onderdeel uit het system mocken, nabootsen. Door bij tests gebruik te maken van deze mock objecten kunnen de tests volledig onafhankelijk van externe databases worden uitgevoerd. Met name in situaties waar tests automatisch uitgevoerd worden op bijvoorbeeld een server is dit zeer wenselijk. De omgeving waar de server in staat hoeft dan namelijk niet exact dezelfde omgeving te zijn als de ontwikkelomgeving van de ontwikkelaar zelf. Ook wanneer er in teams gewerkt wordt is dit zeer wenselijk, omdat elke ontwikkelaar een eigen ontwikkelomgeving heeft die af kan wijken van de andere ontwikkelomgevingen.

Door de flexibele opzet van het framework was het geen enkel probleem om bij het testen 'echte' objecten uit te wisselen voor mock objecten. Het uitvoeren van de tests werd daardoor zeer eenvoudig. Ook waren de unit tests ideaal om als regressietest te gebruiken. Doordat het uitvoeren van sommige taken binnen het framework via verschillende componenten verloopt, is het mogelijk dat een wijziging in component A problemen oplevert in component B. Dit is van te voren soms lastig te voorzien. Door het uitvoeren van unit tests na het doorvoeren van wijzigingen kunnen dit soort fouten eenvoudig gesignaleerd worden.

4.6.2. Batchjob B21

Een batchjob is inherent een stuk lastiger te unit testen. Dit komt doordat batchjobs gewoonlijk bestaan uit input, verwerking en output. De verwerking bestaat uit het testen van allerlei condities en het uitvoeren van acties op basis van deze condities. Het is erg lastig en bewerkelijk om hier goede unit tests voor te schrijven. In deze gevallen worden batchjobs dan ook vaak getest door middel van functioneel testen.

Functioneel testen wilt zeggen dat er een aantal testgevallen gemaakt worden die realistische input simuleren, dat deze testgevallen aan het systeem gevoed worden en dat er vervolgens getest wordt of de output voldoet aan hetgeen er verwacht wordt.

Het opstellen van deze testgevallen vereist een grote hoeveelheid materiekennis. De testgevallen zijn dan ook in samenwerking met een collega met uitgebreide materiekennis opgesteld. Het is namelijk

cruciaal dat de testgevallen realistische gegevens bevatten en dat de testgevallen een zo groot mogelijk deel van het systeem dekken.

In het geval van batchjob B21 was er naast de functionele tests nog een tweede manier om de juiste werking van de software aan te tonen. B21 is namelijk een herschrijving van de bestaande B21 batchjob. Het gedrag van de nieuwe implementatie kon dus vergeleken worden met het gedrag van de oude implementatie. Omdat niet uit te sluiten is dat de oude implementatie volledig bug vrij is, is deze methode echter alleen ter ondersteuning gebruikt. De functionele tests zijn leidend voor het aantonen van de kwaliteit en juiste werking van de software.

4.7. Voorbereiding tot invoering / overdracht

Om de kennis en resultaten die tijdens de uitvoering van dit project zijn opgedaan te behouden worden aan het eind van de stageperiode kennis en producten overgedragen aan het bedrijf.

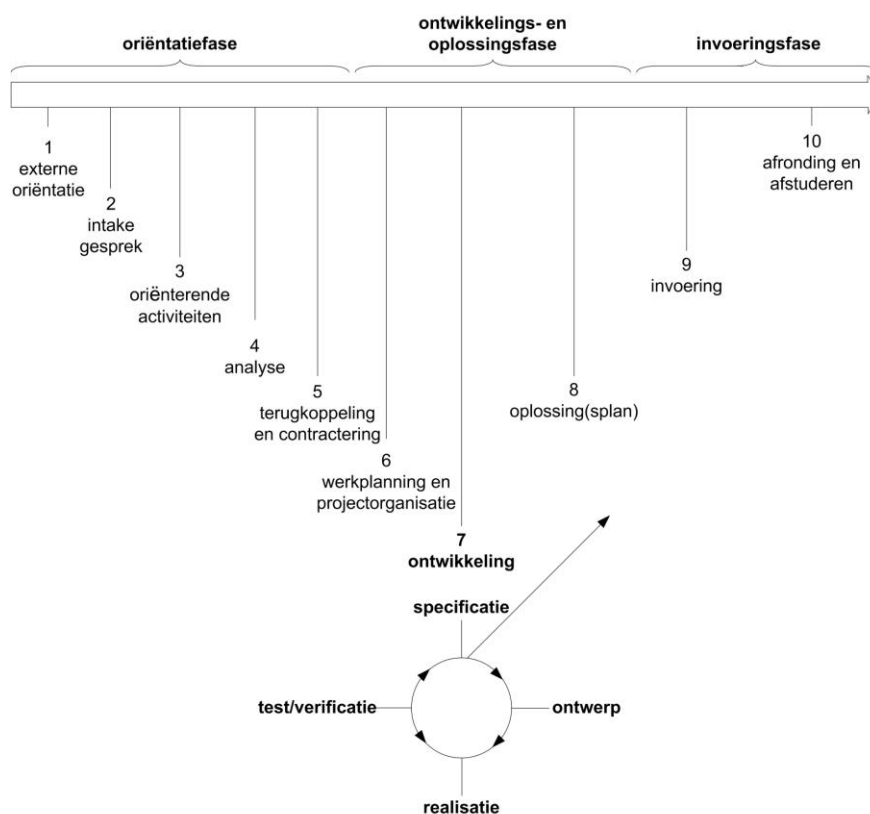
De kennis zal worden overgedragen in de vorm van de standaard voor .NET batchjob ontwikkeling en de ontwerpen, handleiding en documentatie van het batch framework.

De eindproducten zullen worden overgedragen aan collega's van de afdeling AOC. Vanuit hier zullen deze producten op een later tijdstip worden overgedragen aan de afdeling PIO (Proces Inrichting en Ondersteuning). PIO is een overkoepelende afdeling voor alle ontwikkelafdelingen waar standaarden en generieke componenten worden ondergebracht. Vanuit hier kan de oplossing dan ook uitgerold worden naar de afdelingen die hier interesse in hebben.

4.8. Relatie tot het Tien Stappen Plan

Het TSP (Tien Stappen Plan) is goed over de tijdens deze stage gevolgde waterval methode heen te leggen. Alle tien stappen uit het TSP komen in de hier voor genoemde fases terug.

Tien Stappen Plan¹⁾ Afstudeerstage voor (Technisch) Informatici



¹⁾ vrij naar "Competent Afstuderen en Stagelopen", Kempen en De Keizer, Noordhoff 2006

4.8.1. Oriëntatiefase

De oriëntatiefase komt naar voren in de hier voor beschreven definitiestudie en onderzoek fase. Alle werkzaamheden in deze fase werken naar het opstellen van het PID toe. Belangrijk is dat aan het eind van deze fase voor alle betrokken partijen duidelijk is wat de drijfveer voor het project is, wat het doel van het project is en welke wensen en eisen er bestaan vanuit de betrokken partijen.

4.8.2. Ontwikkelings- en oplossingsfase

Het ontwikkelingsdeel komt naar voren in de fases “Functioneel ontwerp”, “Technisch ontwerp”, “Ontwikkeling” en “Testen”.

Het oplossingsplan zal formeel worden gepresenteerd tijdens het tweede bedrijfsbezoek, maar zal informeel al voor die tijd worden opgeleverd.

4.8.3. Invoeringsfase

De invoeringsfase omvat in dit project enkel de overdracht van kennis, producten en documentatie, zoals beschreven in de fase “Vorbereiding tot invoering / overdracht”. De afdeling AOC zal zorg dragen voor de invoering binnen de organisatie omdat dit geen onderdeel van de opdracht is.

4.8.4. Conclusie

Het TSP sluit nauw aan bij de in het verleden vaker gebruikte waterval methode. Het TSP beschrijft in feite in een groter detail de stappen van het waterval model. Het was prettig om het TSP als leidraad te kunnen gebruiken bij het uitvoeren van de stage.

5. Uitdagingen en resultaten

5.1. Batch framework

5.1.1. Resultaat

De ontwikkeling van het batch framework is goed verlopen. Hieronder volgen een aantal uitdagingen en beschrijvingen van hoe deze overwonnen zijn.

5.1.2. Uitdagingen

5.1.2.1. Design van het framework

Omdat één van de doelstellingen is om het framework binnen verschillende systemen in te kunnen zetten moet het framework eenduidig opgezet worden. Het framework moet generiek zijn, maar wel genoeg functionaliteit bevatten om bruikbaar te zijn. Om deze doelstelling te bereiken is er voor een gelaagde structuur gekozen waarbij het framework niet afhankelijk is van het gebruik van bepaalde externe componenten. Ook bestaat door deze architectuur de mogelijkheid om functionaliteit uit het framework in een van de lagere lagen uit te breiden of zelfs te overschrijven.

5.1.2.2. Microsoft Enterprise Library

Hoewel het framework gebruik maakt van een aantal externe componenten, dwingt het framework niet het gebruik van specifieke externe componenten af. Door gebruik te maken van een lagenstructuur kan het framework bijvoorbeeld met elke database software en elk logging systeem werken.

Binnen APG wordt regelmatig van de Microsoft Enterprise Library (zie bron B2) gebruikt gemaakt. Dit is een framework van Microsoft dat generieke implementaties biedt voor verscheidene veel voorkomende software componenten zoals caching, security en logging. Vanuit AOC bestond dan ook de wens om bij het framework een logging laag te leveren die gebruikt maakt van de Enterprise Library.

De uitdaging zat in het feit dat Enterprise Library normaliter geconfigureerd wordt door middel van een configuratie bestand waar instellingen voor onder andere de logging in geplaatst moeten worden. Dit zou betekenen dat elke batch applicatie een eigen configuratie bestand zou moeten hebben met daarin steeds weer dezelfde instellingen, met een aantal kleine wijzigingen zoals bijvoorbeeld de bestandsnaam van het logbestand. Één van de doelstellingen van het framework is juist om dit soort boilerplate code in batchjobs tegen te gaan.

Een andere requirement van logging in batchjobs is dat de naam van log bestanden de datum en tijd van uitvoeren bevatten. Ook dit bleek met de standaard werking van Enterprise Library niet mogelijk.

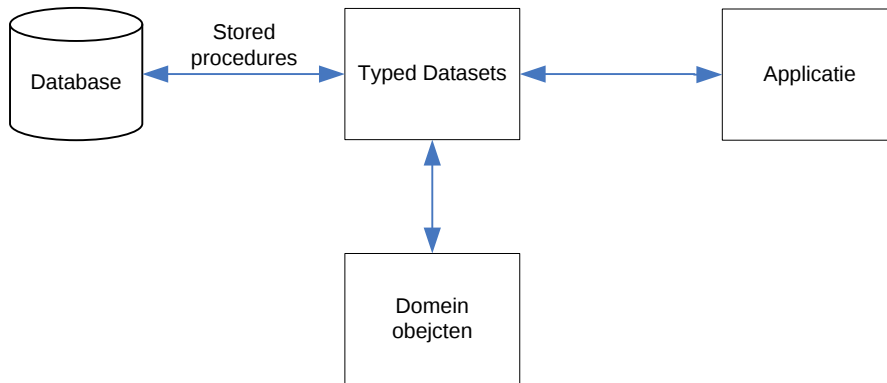
Naar enig onderzoek is gebleken dat, hoewel ongedocumenteerd, het mogelijk is om Enterprise Library Logging in plaats van met een configuratie bestand te configureren vanuit code (zie bron B1). Dit biedt volledige flexibiliteit om tijdens het uitvoeren van de software de bestandsnaam van logging bestanden samen te stellen en ook om tijdens het uitvoeren additionele log bestanden aan te maken. Dit is exact hetgeen dat wenselijk was voor het framework en dit bleek dan ook perfect te werken.

5.1.2.3. Typed datasets in combinatie met Oracle Database 10g

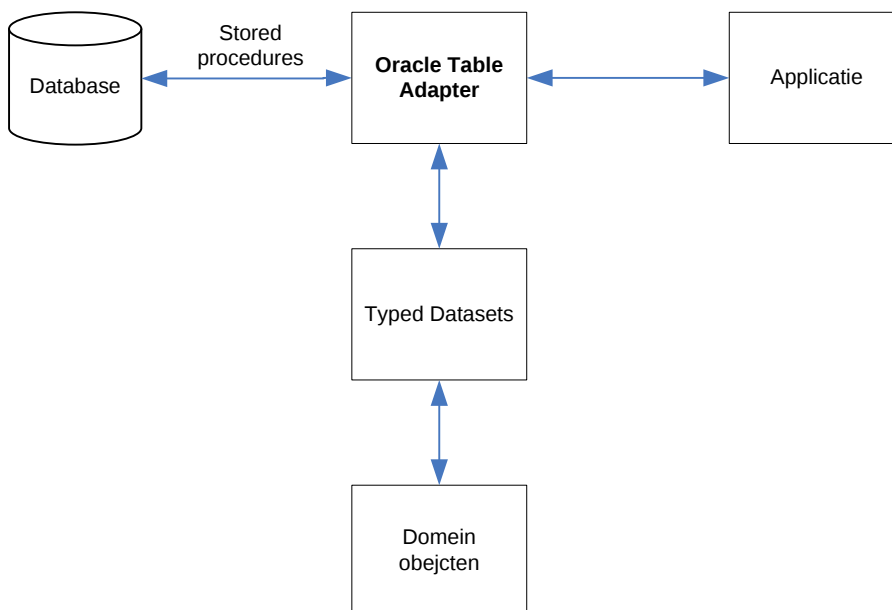
De ontwikkelrichtlijnen binnen APG schrijven voor dat database queries niet in applicaties zelf opgenomen moeten worden, maar dat deze in *stored procedures* in de database thuis horen. Daarnaast wordt voorgeschreven dat er zoveel mogelijk met *Typed Datasets* en *gegenereerde domein objecten* gewerkt moet worden. Dit, in combinatie met stored procedures, werkt out of the box met MS SQL databases, maar niet met Oracle Database in combinatie met de Oracle 10g client, hetgeen in de productie omgeving gebruikt wordt.

Om toch aan deze richtlijnen te kunnen voldoen en omdat het gebruik hiervan ook erg prettig werkt is er besloten om een stuk software te ontwikkelen dat het mogelijk maakt om Typed Datasets in combinatie met Oracle stored procedures te kunnen gebruiken. Schematisch ziet dat er als volgt uit:

Typed datasets in combinatie met MS SQL stored procedures (ingebouwde functionaliteit in .NET)



Typed datasets in combinatie met Oracle stored procedures



Het bovenste diagram toont de werking van Typed Datasets in combinatie met MS SQL, zoals ingebouwd in het .NET framework.

Het onderste schema toont de werking van Typed Datasets in combinatie met de ontwikkelde Oracle Table Adapter. Zoals te zien is neemt de Oracle Table Adapter een centrale plek in. De applicatie communiceert met de Oracle Table Adapter. De Oracle Table Adapter communiceert op basis van de Typed Datasets en domein objecten via stored procedures met de database.

5.2. Batchjob B21

5.2.1. Resultaat

De ontwikkeling van B21 op basis van het batch framework is goed verlopen. Het was erg prettig om te werken op basis van het framework, het neemt erg veel werk uit handen. Dit is ook terug te zien in het aantal regels code (LOC) van het nieuwe B21 project ten opzichte van het oude project: 2177 regels ten opzichte van 3276. Dit is een besparing van ruim 33%. Een kleinere code base is voordelig in een aantal opzichten:

- Makkelijker te onderhouden doordat er minder 'ruis' in de code is, enkel core functionaliteit.
- Een kleinere code base is, mits goed geschreven, beter te doorgronden voor personen die de code niet zelf geschreven hebben.

Ook op het gebied van performance heeft de nieuwe implementatie een behoorlijk voordeel geboekt. In een testopstelling waarbij 19978 aanvragen verwerkt werden deed de oude implementatie daar 31,5 minuten over. De nieuwe implementatie deed er 16,5 minuten over; een performance winst van 47,6%.

5.2.2. Uitdagingen

5.2.2.1. Functioneel ontwerp doorgronden

Aangezien de ontwikkeling van B21 geen nieuwbouw maar herbouw was, lag er al een functioneel ontwerp. Het functioneel ontwerp zat redelijk goed in elkaar, maar was niet op alle punten compleet. De maker van het ontwerp ging er duidelijk vanuit dat de lezer een uitstekende materie kennis heeft, wat het voor een leek nogal lastig te doorgronden maakt. Deze materie is namelijk zeer omvangrijk en complex. Door een aantal gesprekken met de bedrijfsbegeleider is alles echter helder geworden.

6. Conclusie en aanbevelingen

Nu de stageperiode ten einde loopt zijn alle geplande werkzaamheden succesvol voltooid. Mijn werk wordt overgedragen aan de collega's van de afdeling AOC. Het framework en de standaard voor .NET batchjob ontwikkeling zullen in eerste instantie binnen AOC blijven en zullen later worden ondergebracht bij de afdeling PIO zodat ook andere afdelingen kunnen profiteren van de voordelen van de standaard en het batch framework.

Batchjob B21 zal bij een volgende release in productie worden genomen. De klant zal echter eerst nog acceptatie tests uitvoeren. Wanneer deze release plaats gaat vinden is vooralsnog niet bekend.

Uit de ontwikkeling van batchjob B21 is gebleken dat het gebruik van het framework veel voordelen heeft. De nieuwe B21 implementatie omvat 33% minder regels code dan de oude implementatie, hetgeen de ontwikkeltijd aanzienlijk verkort. Ook het opsporen van fouten zal in de toekomst gemakkelijker gaan omdat de omvang van de code kleiner is en omdat de batchjob enkel de business logica bevat.

Ook voor de klant levert het framework een prettige verbetering op. De batchjobs die in de toekomst op basis van het framework ontwikkeld gaan worden of herschreven worden om van het framework gebruik te maken, zullen allen een gestructureerde logging output opleveren. Dit maakt het eenvoudig om de logging output te verwerken met behulp van bijvoorbeeld Excel.

Voor de afdeling productiebeheer zal het eenvoudiger worden om het oplossen van productie incidenten door te spelen naar de juiste afdeling of persoon omdat de technische logging van het framework uitgebreider is dan de huidige technische logging.

Graag zou ik APG willen aanbevelen om de opgeleverde oplossing in de toekomst breed toe te passen binnen de organisatie. Hoe breder de oplossing toegepast wordt hoe groter het voordeel wordt. Het is wellicht nuttig om ook andere afdelingen te betrekken bij de verdere ontwikkeling van de standaard en het framework om de oplossing zo nauw mogelijk aan te kunnen laten sluiten bij de werkwijzen van de andere afdelingen.

Persoonlijke evaluatie stage

Voor aanvang van de stage had ik de volgende doelstellingen:



Tijdens mijn stage hoop ik ervaring op te doen in het werken binnen een grote organisatie. APG is een erg grote organisatie en bij het uitvoeren van deze opdracht krijg ik daar veelvuldig mee te maken.

Ook op communicatief gebied zal ik tijdens deze stage veel kunnen leren. Ik zal in contact treden met een klant, collega's binnen een team en verschillende afdelingen van de organisatie.

Op technisch gebied zal ik te maken krijgen met technieken waar ik nog weinig ervaring mee heb. Deze technieken worden in enterprise omgevingen veel gebruikt, dus het is erg belangrijk en leerzaam om hier ervaring mee op te doen.

Alle drie deze punten zijn in de stage uitgebreid aan bod gekomen. Ik heb gezien hoe het er in een groot bedrijf aan toe gaat. Zo vond ik de flexibiliteit in het indelen van de werktijden erg prettig, maar de bureaucratie kan wel erg vervelend zijn. Zo kan het bijvoorbeeld maanden duren voor een nieuwe versie van een software component gebruikt mag worden, omdat daar eerst toestemming voor moet komen en uitgebreid getest moet worden.

Ook op communicatie gebied heb ik tijdens deze stage veel geleerd. Ik heb veel contact gehad met collega's en heb ook meerdere malen met de klant gesproken.

Op technisch gebied heb ik onder andere gewerkt met VB.NET, Oracle database, MS SQL server, Team Foundation Server, Typed Datasets, LINQ, etc. Veel van deze technieken/producten waren nieuw voor mij, het was dan ook erg leerzaam om hier mee te kunnen werken.

Binnen APG heerst een professionele sfeer. De collega's zijn allen erg kundig en met name van Jeroen heb ik veel kunnen leren.

Ik heb geprobeerd om tijdens deze stage zo geordend mogelijk te werken. Vanaf dag één heb ik plannings gemaakt, to-do lijsten bijgehouden, gereflecteerd op mijn functioneren en me bijzonder hard ingezet om het best mogelijke resultaat neer te zetten.

In het verleden is het geordend en gestructureerd werken wel eens een struikelblok voor mij geweest, maar in deze stage kan ik met recht zeggen dat dat dit keer niet het geval is geweest. Met name het opschrijven van taken zodra ze me ter oren kwamen, of zodra ik ze bedacht is erg nuttig geweest. Het hielp me bij het overzicht houden op de grote hoeveelheid aan werkzaamheden die voltooid moesten worden en het was elke keer weer een genot om een item van mijn lijstje te kunnen strepen.

Tijdens het uitvoeren van de stage zijn uiteraard veel zaken naar voren gekomen die op school behandeld zijn. Het was leuk om te zien dat veel werkzaamheden die op school weinig nut leken te hebben nu ineens zeer nuttig bleken te zijn. Zaken als use cases maken, sequence diagrammen opstellen etc. hebben in grotere projecten veel meer waarde dan in de duidelijk afgebakende projecten op school. Met name doordat ik het ontwerp van batchjob B21 moest doornemen was goed te merken hoe belangrijk deze documenten zijn. Ze bieden een duidelijke kijk op de belangrijkste aspecten van een systeem en zijn door het gebruik van vaste formaten eenvoudig te begrijpen.

Ik heb het gevoel dat ik me door de opleiding goed staande heb kunnen houden binnen de professionele sfeer bij APG. Ik heb nooit het gevoel gehad dat ik tekort schoot in mijn kennis. Wel merk ik soms dat ik nog ervaring mis, maar dat zal voor niemand een verrassing zijn. Hetgeen je in ervaring mist kun je compenseren met frisse, nieuwe ideeën en het is leuk om te zien dat deze input gewaardeerd wordt.

Literatuurlijst en bronvermelding

Internetpagina's

ID	Pagina
B1	http://forums.asp.net/p/994264/1949754.aspx
B2	http://msdn.microsoft.com/en-us/library/ff647292.aspx
B3	http://download.oracle.com/docs/cd/B19306_01/server.102/b14219.pdf

Bijlagenoverzicht

ID	Titel
I	PID
II	Richtlijnen en handleiding .NET batchjob ontwikkeling