

2013

Automatisering Artikelbeheer

Scriptie





Strukton
Worksphere



Strukton
Worksphere

Titelpagina

AFSTUDEERVERSLAG VOOR FONTYS HOGESCHOOL ICT

Afstudeerder	
Naam	Sjors van Osch
Studentnummer	2152631
Opleiding	HBO-ICT Software Engineering

Afstudeerperiode	
Van	19-11-2012
Tot	25-03-2013
Aantal werkdagen	82

Afstudeerbedrijf	
Naam	Strukton Worksphere
Afdeling	SSC Gegevensbeheer
Adres	Science Park Eindhoven 5206
Postcode	5692 EG
Plaats	Son
Bedrijfsbegeleider	Peter Blokk

Verslag	
Titel	Automatisering Artikelbeheer
Ondertitel	Scriptie
Datum uitgifte	28-03-2013

Getekend voor gezien door bedrijfsbegeleid(st)er:

Datum:

28-3-2013

De bedrijfsbegeleider,

Voorwoord

Deze scriptie heb ik geschreven als beschrijving en onderbouwing van mijn afstudeerproject. Voor het uitvoeren van dit project ben ik werkzaam geweest als stagiaire op de afdeling Gegevensbeheer van Strukton Worksphere. Tijdens dit afstudeerproject heb ik mij bezig gehouden met het automatiseren van het verwerken van Artikel- en Kortingsgegevensbestanden.

Hierbij zou ik graag mijn docentbegeleider Huub van Thienen willen bedanken voor zijn uitstekende inhoudelijke en procedurele begeleiding.

Tevens wil ik graag Peter Blokk, mijn bedrijfsbegeleider bij Strukton Worksphere, bedanken voor het mogelijk maken en begeleiden van dit project.

Verder spreek ik hierbij mijn dank uit naar de volgende medewerkers van Strukton Worksphere voor hun hulp en ondersteuning:

- Ton van den Boomen
- Twan van der Kruijs
- Giel Peeters
- Mark Ooijen
- Patrick de Man



Strukton
Worksphere



Inhoudsopgave

Samenvatting	4
Summary	5
Verklarende Woordenlijst	6
1. Inleiding	9
2. Het bedrijf	10
3. De Opdracht	13
4. Bestanden	23
5. Softwaresysteem	25
6. Nieuwe Situatie	26
7. Performance	28
8. Flexibiliteit	31
9. Bestandscommunicatie	43
10. Databasecommunicatie	44
11. Rapporten	53
12. Logboek	53
13. Configuratie	54
14. Text Encoding	55
15. Testen	58
16. Deployment	61
17. Bedrijfsimpact	62
18. Conclusies en aanbevelingen	63
Evaluatie	64
Literatuurlijst	65
Bijlagen	67



Samenvatting

Strukton Worksphere is een installatiebedrijf dat zich onder andere bezig houdt met de installatie van beveiligingssystemen, liften, verlichting, verwarming, etc..

Alle artikelen die Strukton Worksphere gebruikt voor het uitvoeren van haar taken als installatiebedrijf, zijn opgenomen in haar eigen softwaresystemen. Het invoeren van deze gegevens gebeurt op basis van bestanden. Momenteel is het verwerken van deze gegevens grotendeels handmatig werk dat erg tijdrovend is voor de medewerkers van de afdeling Gegevensbeheer.

Mijn opdracht is dan ook het automatiseren van het verwerken van de artikel- en kortingsbestanden die worden opgenomen in het softwaresysteem van Strukton Worksphere. Aan het einde van dit project moeten de gegevensbeheerders minder tijd kwijt zijn met het verwerken van de gegevens en deze makkelijk kunnen controleren.

De opdracht is uitgevoerd op basis van een variant op het V-Model en Agile. Hierbij zijn eerst de grote lijnen van het project uitgezet. Deze zijn later per component op detailniveau uitgewerkt. Er is voor deze manier van werken gekozen vanwege het feit dat een aantal aspecten van de opdracht bij aanvang nog onduidelijk waren. Deze moesten in een later stadium worden geanalyseerd. Tijdens het uitvoeren van de opdracht was er tevens veel contact mogelijk met de eindgebruikers. Hierdoor konden aanpassingen makkelijk en snel worden doorgevoerd. In tegenstelling tot een echt Agile ontwikkelmethode worden niet alle tussenstappen los gedocumenteerd en getest.

Omdat er grote hoeveelheden gegevens moeten worden verwerkt door het nieuwe softwaresysteem is er tijdens dit project gezocht naar de optimale balans tussen functionaliteit, flexibiliteit, performance en memory management.

Uiteindelijk is er een nieuw softwaresysteem gerealiseerd voor het verwerken van de artikel- en kortingsgegevens. Dit nieuwe systeem zorgt niet alleen voor een geautomatiseerde verwerking van deze gegevens maar genereert ook rapporten met informatie over de geïmporteerde gegevens voor verschillende afdelingen.

Daarnaast zorgt het nieuwe systeem voor een standaardisering van de Text Encoding. Hierdoor kunnen er geen fouten meer voorkomen met betrekking tot speciale karakters en zijn omschrijvingen van artikelen duidelijker leesbaar.

Vanwege de grote hoeveelheid gegevens die verwerkt worden en de mogelijke mutaties hierop is het belangrijk dat het nieuwe systeem na in productie name gecontroleerd en getest blijft worden. Vooral omdat alle prijzen waarmee Strukton Worksphere rekent voor zowel inkoop als verkoop gebaseerd zijn op de gegevens die door het nieuwe softwaresysteem worden verwerkt.

Summary

The project described in this document was done at Strukton Worksphere. As an installation company, Strukton Worksphere's activities include but are not limited to outfitting buildings with security systems, elevators, lighting and heating.

All products used by Strukton Worksphere are defined in their own software system. These records need to be updated regularly, which at the moment, is a largely manual task. This means it is very time consuming for the data management staff workers.

The assignment of this project was automating the procedure of processing the product and discount files being used by Strukton Worksphere. At the end of this project the data management staff workers should be able to spend less time processing the product and discount data. The processed data should also be easily verifiable.

A mixture between the V-Model and Agile was used during this project. At first the major aspects of the project were designed. At a later stage the details were filled in at a component level. This development model was chosen because some details of the project were not clear at the start and had to be defined by trial and error. During this project there also was the ability to communicate directly and frequently with end users. Because of this the project details could easily be altered if needed. Unlike a real Agile development method, not all processes were individually documented and tested.

As large amounts of data need to be processed there was searched for the perfect balance between functionality, flexibility, performance and memory management for the new software system.

The result of this project is a new automated software system designed to process product and discount data. Besides processing the data it also generates multiple reports to verify different aspects of the processed data for various departments.

As the new software system also standardizes the Text Encoding through the entire workflow there will no longer be any errors because of special characters in the data and making the product descriptions more readable for users.

Because of the large amounts of data and mutations on this data it is vital that the system is to be extensively tested and debugged after this project. Especially since all prices used for calculation by Strukton Worksphere are processed by the new software system.



Verklarende Woordenlijst

ADO.NET Entity Framework (EF)

Het ADO.NET Entity Framework is een open source framework van Microsoft waarmee databases tabellen worden vertaald naar objecten. Hierdoor is de communicatie tussen een applicatie en de database makkelijker en hoeft de programmeur geen kennis te hebben van SQL. Het Entity Framework is een vorm van Object Relational Mapping (ORM).

Application Programming Interface (API)

Een Application Programming Interface protocol is bedoeld als interface waarmee softwarecomponenten met elkaar kunnen communiceren.

Cores

Cores zijn het aantal feitelijke processors op een CPU. Een CPU kan meerdere Cores hebben. Een Quad-Core processor is bijvoorbeeld één CPU met vier Cores.

Een computer kan op zijn beurt weer meerdere CPU's hebben. Een computer met bijvoorbeeld twee Quad-Core CPU's heeft dus 8 Cores.

CPU

Staat voor Central Processing Unit. Dit is de processor van een computer.

CSV-bestand

Een CSV-bastand is een databestand waarin alle velden zijn gescheiden door middel van komma's (Comma Separated Values).

eCatalog

Dit is de digitale artikelcatalogus van Strukton Worksphere. Hierin zijn alle artikelen van Strukton Worksphere opgenomen. eCatalog is onderdeel van Focus maar heeft zijn eigen Microsoft SQL database.

eCustomer

eCustomer is net als eCatalog onderdeel van focus. Hieruit wordt informatie gehaald over verkoopkortingen.

Enterprise Service Bus (ESB)

Dit is een architectuur waarmee de communicatie tussen de afnemers van diensten (services) en aanbieders hiervan, vereenvoudigd wordt.

Europees Technisch Informatie Model (ETIM)

ETIM is een stichting die zorgt voor het bevorderen van een uniforme, elektronische communicatie binnen de installatiesector.

Focus

Focus is een softwaresysteem dat specifiek voor en door Strukton Worksphere gemaakt is. Dit systeem zorgt voor de koppeling tussen meerdere systemen en geeft hiervoor een gebruiksvriendelijke front end. Een van deze systemen is SAP. Met Focus kunnen onder andere inkooporders en werkbonden worden gemaakt. Focus staat voor Front Office Customer Service en maakt gebruik van Microsoft SQL databases.

ICC-bestand

ICC is een standaard kortingsbestand formaat dat is gedefinieerd door ETIM en gebruikt wordt in de installatiebranche.

InstallCalc

InstallCalc is een geautomatiseerd calculatiesysteem dat gebruikt wordt door Strukton Worksphere. InstallCalc maakt gebruik van een Microsoft SQL database. InstallCalc wordt onder andere gebruikt voor het maken van productcalculaties en het opstellen van offertes.

Mod-bestanden

Dit is een collectie bestanden ter aanvulling van PAB2 bestanden. Deze bestanden bevatten extra informatie over handelsartikelen.

MoSCoW

Dit is een methode voor het definiëren van de mate van belangrijkheid van de verschillende doelstellingen/requirements.

MoSCoW staat voor:

- Must have
- Should have
- Could have
- Won't Have (but would like)

Multithreading

Multithreading is het gebruiken van meerdere threads binnen één proces(application). Door het gebruik van meerdere threads kunnen taken tegelijk worden uitgevoerd.

Object Linking and Embedding, Database (OLEDB)

Dit is een API gemaakt door Microsoft. Hiermee kunnen data worden opgehaald uit relationele databases maar ook uit niet-relationele databases zoals object databases, spreadsheets en bestanden die niet perse SQL implementeren.

PAB2-bestanden

Dit is een collectie bestanden waarin artikelgegevens staan beschreven. PAB2 is een standaard formaat binnen de installatiebranche en is gedefinieerd door ETIM.

Singleton

Een singleton is een klasse waarvan maar één instantie kan bestaan.

**Static**

Van een static klasse of methode hoeft geen instantie te bestaan om er gebruik van te maken.

Text editor

Een text editor is een applicatie waarmee tekstbestanden kunnen worden aangepast. Een voorbeeld hiervan is Notepad.

Thread

Iedere applicatie die wordt opgestart wordt ook wel een proces genoemd. Een proces kan bestaan uit een of meerdere threads die een deel van de programmacode kunnen uitvoeren.

XML-bestand

XML staat voor Extensible Markup Language. Dit is een standaard syntaxis van formele opmaaktalen waarmee men gestructureerde gegevens kan weergeven in de vorm van platte tekst. Deze tekst is leesbaar door zowel mens als machine.

1. Inleiding

Meer dan één miljoen. Zoveel verschillende artikelen kunnen door Strukton Worksphere worden geleverd.

Strukton Worksphere is een installatiebedrijf en is onderdeel van Strukton. Alle artikelen, met hun bijbehorende kortingen en toeleveranciers, die door Strukton Worksphere geleverd kunnen worden, moeten in de verschillende softwaresystemen van Strukton Worksphere worden opgenomen. Dit is de verantwoordelijkheid van de afdeling gegevensbeheer. Het verwerken van deze gegevens kost momenteel erg veel tijd en de gegevens kunnen niet worden gecontroleerd op correctheid.

Om dit probleem op te lossen is er bedacht om het huidige systeem van verwerken van artikel- en kortingsgegevens te vervangen door een geautomatiseerd systeem.

Het nieuwe softwaresysteem dat uit dit project ontstaat zou ervoor moeten zorgen dat de artikelen- en kortingsgegevens meer up-to-date en correct zijn.

Voor dit project is eerst een globale schets gemaakt van hoe het nieuwe systeem eruit moet komen te zien. Deze is later meer gedetailleerd uitgewerkt per component.

Een globale overzicht van de opdracht en de uitvoering hiervan is te lezen in de hoofdstukken 2 t/m 6 en 18.

Meer technische informatie over het eindresultaat van de opdracht is te vinden in de hoofdstukken 7 t/m 17.

Hoofdstuk 2: Informatie over het bedrijf waar dit project heeft plaatsgevonden.
Hoofdstuk 3: Details van de opdracht.
Hoofdstuk 4: Inzage in de bestanden waarmee tijdens dit project gewerkt wordt.
Hoofdstuk 5: Uitleg van de vormgeving van het nieuwe softwaresysteem.
Hoofdstuk 6: Beschrijving van het nieuwe systeem.
Hoofdstuk 7: De keuzes met betrekking tot performance.
Hoofdstuk 8: De keuzes met betrekking tot flexibiliteit.
Hoofdstuk 9: Communicatie met de verschillende bestanden.
Hoofdstuk 10: Communicatie met de verschillende databases.
Hoofdstuk 11: Het genereren van rapporten.
Hoofdstuk 12: Het genereren van logboeken.
Hoofdstuk 13: Configuratie van het nieuwe systeem.
Hoofdstuk 14: Problemen met betrekking tot Text Encoding.
Hoofdstuk 15: Het testen van het nieuwe systeem en de resultaten hiervan. .
Hoofdstuk 16: Het uitrollen van het nieuwe systeem in de productieomgeving.
Hoofdstuk 17: Beschrijving van de gevolgen van dit project voor het bedrijf.
Hoofdstuk 18: De uiteindelijke conclusies en aanbevelingen aan de hand van dit project.



2. Het bedrijf

2.1.Strukton

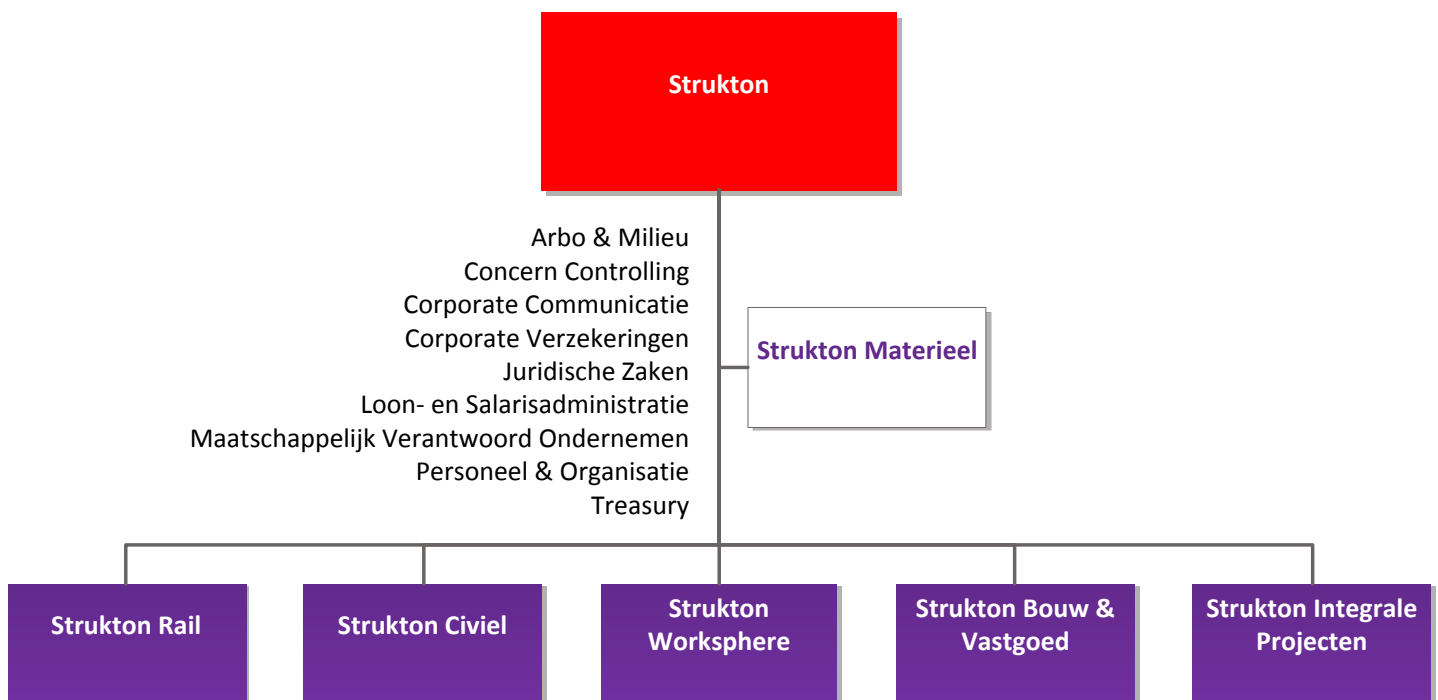
Strukton is opgericht in 1921 als onderdeel van de Nederlandse Spoorwegen. Het is één van de vijf grootste bouwbedrijven van Nederland en één van de grootste rail-infraspecialisten van Europa. Sinds 2010 is Strukton niet meer in handen van de Nederlandse Spoorwegen, maar onderdeel van Oranjewoud N.V..

Strukton Worksphere is één van vijf divisies binnen Strukton.

Andere divisies zijn:

- Strukton Bouw
- Strukton Civiel
- Strukton Integrale Projecten
- Strukton Rail

Alle divisies samen tellen ruim 6.100 medewerkers. Bij Strukton Worksphere werken ongeveer 1600 mensen verdeeld over 13 vestigingen.



Figuur 2-1: Organogram Strukton

2.2.Strukton Worksphere

Coen Nolte senior startte in 1923 het elektrotechnisch installatiebedrijf Nolte. Stork nam Nolte over in 1989. Na enkele fusies van Stork Nolte met andere divisies binnen Stork krijgt het in 2001 de naam Stork Worksphere. In 2006 werd Stork Worksphere overgenomen door Strukton en draagt sindsdien de naam Strukton Worksphere.

Strukton Worksphere creëert een optimale omgeving waarin mensen prettig kunnen verblijven door te zorgen voor betrouwbare, duurzame en efficiënte technische voorzieningen. Denk hierbij aan beveiligingssystemen, liften, verlichting, verwarming, etc..

Strukton Worksphere neemt opdrachtgevers de zorg voor het gebouw uit handen door verantwoordelijkheid te nemen voor het ontwerp, de realisatie en het beheer en onderhoud. Hierdoor kunnen klanten zich concentreren op het primaire proces.

Bron: <http://www.struktonworksphere.nl>

De organisatie van Strukton Worksphere is opgebouwd uit een directie, staffuncties en vestigingen. Iedere vestiging heeft dezelfde structuur en is verantwoordelijk voor een bepaalde regio. De staffuncties zijn ter ondersteuning van alle vestigingen. In bijlagen II en III zijn deze structuren van de volledige organisatie en haar vestigingen afgebeeld in een organogram.

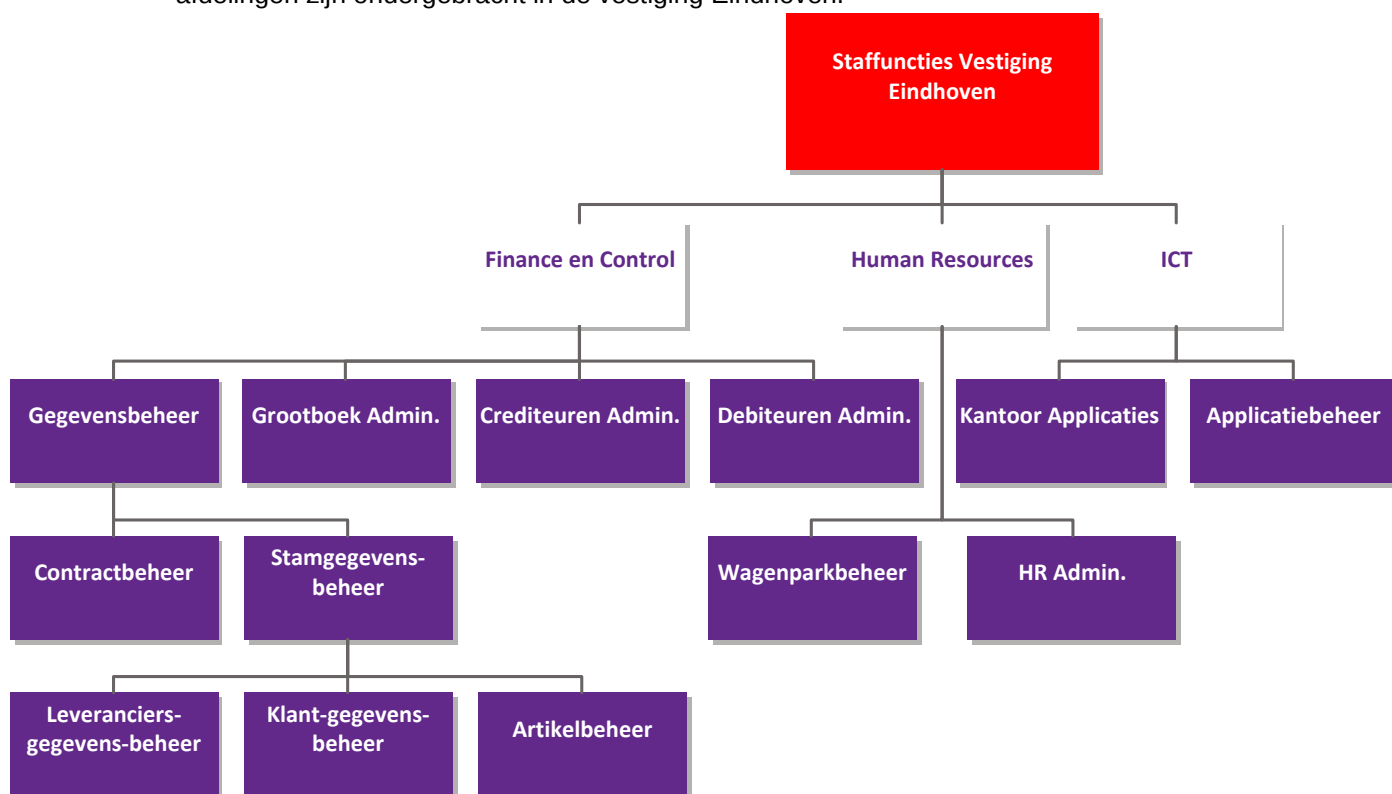
2.3. Betrokken afdelingen

Bij dit project zijn de afdelingen Gegevensbeheer en ICT Applicatiebeheer betrokken.

De afdeling Gegevensbeheer is verantwoordelijk voor het beheren van contract- en artikelgegevens binnen Strukton Worksphere. Het eindproduct van dit project is bedoeld voor werknemers van deze afdeling. Omdat de functionele eisen van dit project in nauwe samenwerking met deze afdeling worden geformuleerd, wordt het project ook uitgevoerd op deze afdeling. Hierdoor is het mogelijk om snel feedback te krijgen op producten en, indien nodig, aanpassingen te doen aan de richting van het project.

De afdeling ICT Applicatiebeheer ontwikkelt en onderhoudt ICT applicaties voor diverse afdelingen binnen Strukton Worksphere. Het betreft hier voornamelijk het eigen softwaresysteem Focus. Deze afdeling geeft, indien nodig, technische ondersteuning tijdens het ontwikkelen en testen van het eindproduct.

De afdelingen Gegevensbeheer en ICT Applicatiebeheer zijn allebei staffuncties van Strukton Worksphere en vallen organisatorisch gezien onder het Hoofdkantoor. Beide afdelingen zijn ondergebracht in de vestiging Eindhoven.



Figuur 2-2: Organogram Staffuncties Vestiging Eindhoven

3. De Opdracht

3.1. Projectomgeving

Om de opdracht goed te kunnen begrijpen is het belangrijk eerst te weten welke binnen Strukton Worksphere aanwezige systemen een rol spelen bij dit project.

Dit project heeft invloed op drie verschillende systemen binnen Strukton Worksphere. Zowel direct als indirect. Deze systemen zijn:

- Focus
- InstallCalc
- SAP

Focus

Focus is een softwaresysteem dat specifiek voor en door Strukton Worksphere is gemaakt. Dit systeem zorgt voor de koppeling tussen meerdere systemen en geeft hiervoor een gebruiksvriendelijke interface. Met Focus kunnen onder andere inkooporders en werkbonden worden gemaakt.

Tot Focus behoort het eCatalog systeem waarin producten, handelsartikelen, toeleveranciers, prijzen en inkoopkortingen zijn opgenomen. Tevens bevat Focus het eCustomer systeem waarin klanten en verkoopkortingen zijn opgenomen.

InstallCalc

InstallCalc is een geautomatiseerd calculatiesysteem dat gebruikt wordt door Strukton Worksphere. InstallCalc maakt gebruik van een Microsoft SQL database. InstallCalc wordt onder andere gebruikt voor het maken van productcalculaties en het opstellen van offertes.

Binnen Strukton Worksphere zijn meerdere InstallCalc systemen, namelijk voor:

- Beheer
- Productie
- Installatiedesk
- Kovo
- Hessenpoort

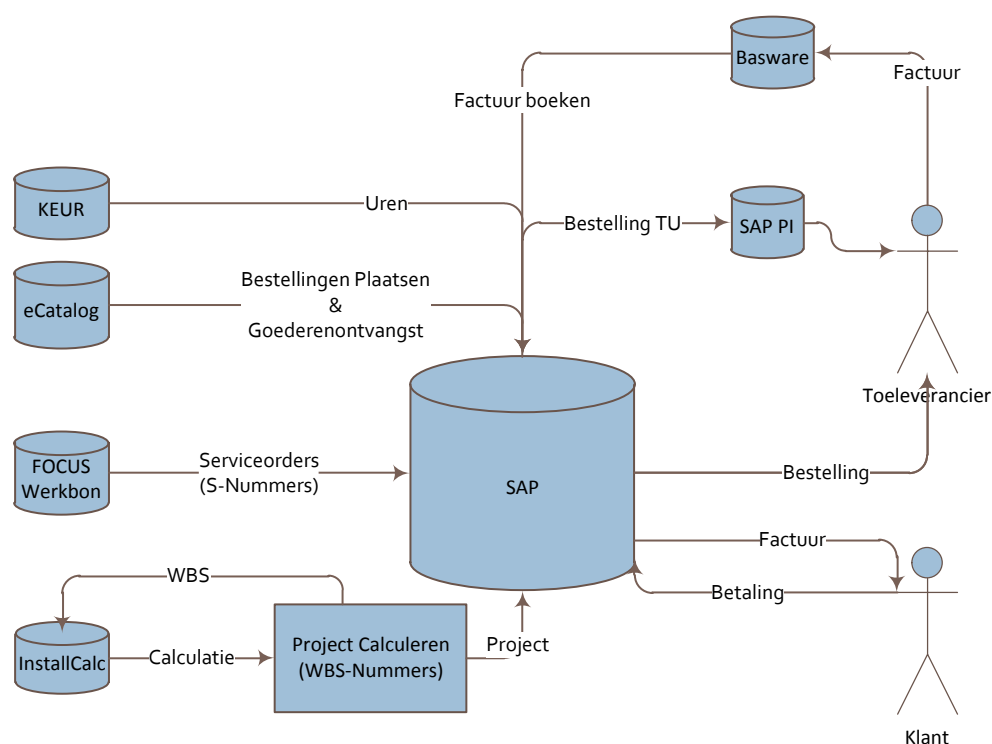
Deze verschillende InstallCalc systemen bevatten niet dezelfde gegevens maar alleen de gegevens die van belang zijn voor de desbetreffende afdelingen. Alle gegevens worden eerst in het Beheer InstallCalc systeem gezet en vanuit daar in de overige systemen.

Dit project heeft direct invloed op het Beheer InstallCalc systeem en indirect op de overige systemen.

SAP

SAP is een geautomatiseerd ERP systeem. Bijna alle softwaresystemen binnen Strukton Worksphere communiceren met SAP.

Bij Strukton Worksphere wordt SAP onder andere gebruikt voor het calculeren en factureren van orders van klanten, het opstellen en verzenden van orders naar toeleveranciers, het bijhouden van de urenregistratie, het koppelen van uren en materialen aan projecten, het aanmaken van serviceorders en het opstellen van rapporten.



Figuur 3-1: Softwareomgeving Strukton Worksphere

De gegevens die SAP gebruikt voor diverse taken komen onder andere uit eCatalog en InstallCalc. In Figuur 3-1 ziet u een globaal overzicht van de relevante softwareomgeving voor dit project binnen Strukton Worksphere. Hierop is te zien dat gegevens uit eCatalog en InstallCalc gebruikt worden voor het opstellen van facturen voor de klanten en bestellingen naar de toeleveranciers.

Dit project heeft dus indirect invloed op SAP en alle processen die hieraan gekoppeld zijn.

3.2.Beginsituatie

Strukton Worksphere krijgt artikelgegevens aangeleverd van verschillende toeleveranciers via een tussenbedrijf genaamd 2BA. Dit tussenbedrijf zorgt dat Strukton Worksphere alle data aangeleverd krijgt in het PAB2 formaat. Ook krijg Strukton Worksphere van een aantal toeleveranciers rechtstreeks artikelbestanden aangeleverd. Deze bestanden zijn niet altijd in het PAB2 formaat. Strukton Worksphere ontvangt tevens rechtstreeks van de toeleveranciers ICC-kortingsbestanden en van sommige toeleveranciers Mod-bestanden. Mod-bestanden geven extra informatie over de producten. Dit zijn dus aanvullingen op de PAB2-bestanden.

In de huidige situatie worden artikel- en kortingsbestanden verwerkt en vergeleken door middel van macro's en tabellen in Microsoft Access. Vanwege de beperkingen van Microsoft Access kunnen niet alle gegevens tegelijk worden verwerkt. Daarom worden de gegevens eerst opgedeeld per toeleverancier voordat deze worden verwerkt.

Tevens is het nodig om meerdere bewerkingen op de aangeleverde bestanden uit te voeren voordat deze met import tools in de verschillende databases kunnen worden gezet. Deze bewerkingen zijn macro's in Microsoft Access afgewisseld met handmatige acties zoals bijvoorbeeld het tussendoor comprimeren van de Access database om te zorgen dat het bestand niet te groot wordt.

Na het uitvoeren van alle bewerkingen met Access zijn er nieuwe bestanden gegenereerd. Deze tekst- en CSV-bestanden (Comma Separated Values) kunnen door de importtools van InstallCalc en eCatalog worden ingelezen naar de desbetreffende databases.

De gegevens van de belangrijkste toeleveranciers worden één keer per week verwerkt. Van de overige toeleveranciers worden de gegevens maar één keer per maand verwerkt.

Omdat artikelgegevens en kortingsgegevens nooit tegelijk worden aangeleverd en aparte verwerkingsmethoden hebben, zijn dit twee verschillende workflows.

Artikelgegevens verwerken

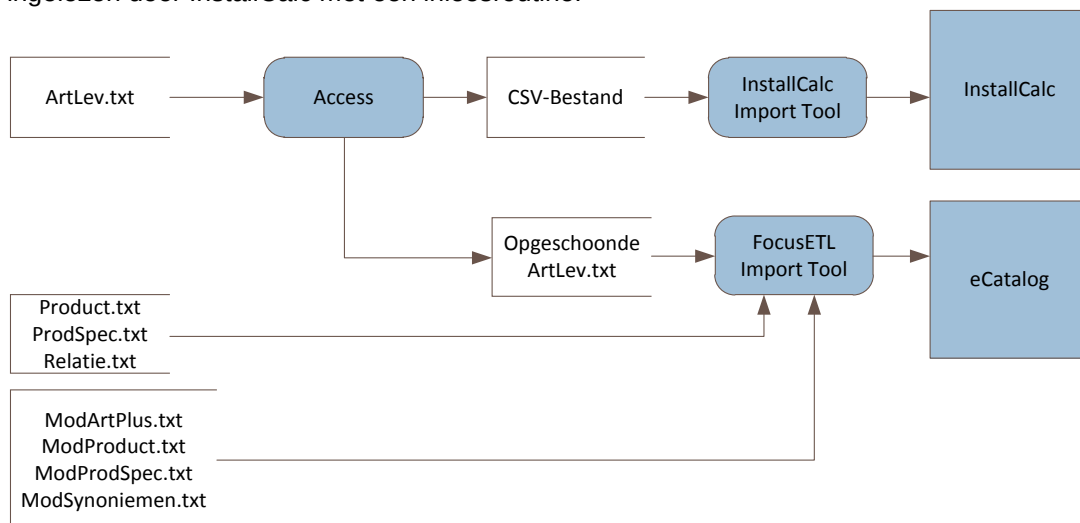
Bij het verwerken van de artikelgegevens gaat het om de PAB2-bestanden bestaande uit:

- ArtLev.txt
- Product.txt
- ProdSpec.txt
- Relatie.txt

En de Mod-bestanden:

- ModArtPlus.txt
- ModProduct.txt
- ModProdSpec.txt
- ModSynoniemen.txt

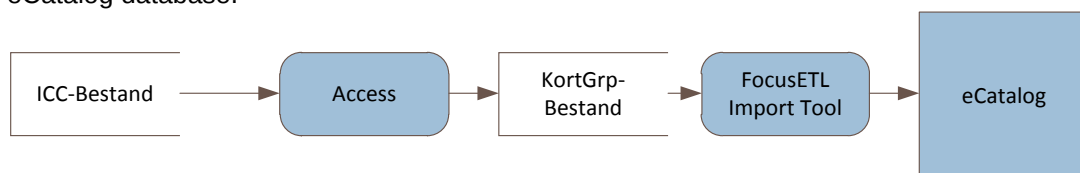
Al deze bestanden worden opgedeeld per toeleverancier. Alleen het ArtLev.txt bestand wordt verwerkt door Access. Deze verricht hier verschillende mutaties op en maakt een nieuw ArtLev.txt bestand dat kan worden geïmporteerd door FocusETL in eCatalog. Tevens maakt Access een CSV-bestand dat kan worden ingelezen door InstallCalc met een inleesroutine.



Figuur 3-2: Workflow verwerken artikelgegevens

Kortingsgegevens verwerken

De kortingsgegevens staan in ICC-bestanden. Deze worden per toeleverancier aangeleverd en hoeven dus niet meer te worden opgedeeld. Het ICC-bestand wordt door Access omgezet naar een nieuw format genaamd KortGrp en opgeslagen in een bestand. Dit nieuwe bestand kan worden ingelezen door FocusETL naar de eCatalog database.



Figuur 3-3: Workflow verwerken kortingsgegevens

3.3.Nadelen van het huidige systeem

Vanwege de grote hoeveelheid gegevens die verwerkt moeten worden en de hoeveelheid verschillende handmatige handelingen die hierbij verricht moeten worden, zijn medewerkers van de afdeling gegevensbeheer nu zo'n 6 uur per dag bezig met het verwerken hiervan. Hierdoor houden deze werknemers weinig tijd over voor het uitvoeren van hun andere werkzaamheden of het controleren van de gegevens die ze verwerken.

Tevens is er geen controle op de gegevens die worden geïmporteerd. Men weet bijvoorbeeld niet of er grote wijzigingen zijn in de inkoopkortingen waarmee ook met

de verkoopkorting rekening gehouden moet worden. Of prijswijzigingen die zo groot zijn dat ze niet kunnen kloppen.

Omdat het verwerken van de gegevens zo veel tijd kost en het dus niet mogelijk is de gegevens van alle toeleveranciers wekelijks te verwerken, zijn veel gegevens niet up-to-date.



3.4. Doel van het Project

Het doel van dit project is het automatiseren van het verwerken van de artikel- en kortingsgegevens op een dusdanige manier dat de maximale belasting van deze taak niet meer dan één uur per dag is voor een medewerker van de afdeling Gegevensbeheer.

Onderzoeksvraag:

Hoe kunnen de artikel- en kortingsgegevens worden verwerkt door middel van automatisering?

3.5. Opdrachtoomschrijving

De projectopdracht is het maken van een applicatie waarmee artikel- en kortingsgegevens efficiënt en controleerbaar verwerkt kunnen worden. Deze applicatie moet gegevens in PAB2-, MOD-, ICC-bestanden kunnen muteren tot bruikbare gegevens voor InstallCalc en eCatalog (onderdeel van Focus). Tevens moet er gekeken worden naar de manier van inlezen van deze gegevens.

Voor verschillende afdelingen worden rapporten gegenereerd over de geïmporteerde gegevens. Bijvoorbeeld over prijsveranderingen van handelsartikelen. Hierdoor kunnen gegevens makkelijk gecontroleerd worden en kan er indien nodig actie op ondernomen worden.

Tevens moeten standaardkortingen van handelsartikelen kunnen worden overgezet van InstallCalc naar eCatalog.

Als laatste is er nog de wens om makkelijk gegevens uit verschillende bestanden en databases te kunnen vergelijken en hiervan rapporten te kunnen genereren.

Hieronder valt het vergelijken tussen:

- Database Focus met database InstallCalc
- PAB2-bestand met InstallCalc- en eCatalog database
- ICC-bestand met de Focus database

Omdat dit deel van de opdracht als extra wordt beschouwd en de meeste tijdswinst voor het bedrijf in het verwerken van de gegevens zit, zal dit alleen worden gemaakt indien er tijd over is tijdens het project.

Strukton Worksphere is voor deze opdracht op zoek naar een oplossing die zowel flexibel als toekomstgericht is. Het moet dus mogelijk zijn voor medewerkers van de afdeling Gegevensbeheer of ICT Applicatiebeheer om aanpassingen te doen aan het nieuwe softwaresysteem na afsluiting van dit project. Denk hierbij aan het toevoegen van nieuwe functionaliteit of het aanpassen van Business Rules of databronnen.

Hieruit kunnen we extra onderzoeksvragen afleiden:

- Heeft het toevoegen van een Rule Engine aan het nieuwe softwaresysteem additionele waarde?
- Zo ja, welke Rule Engine past het beste bij dit nieuwe softwaresysteem?

3.6. Eisen en wensen

Voor het verduidelijken van de opdracht zijn een aantal eisen en wensen gedefinieerd op basis van de opdrachtoomschrijving. Om binnen deze eisen en wensen de prioriteiten te bepalen is gebruik gemaakt van de MoSCoW methode. De prioriteit per eisen en wensen is bepaald in samenspraak met de eindgebruikers en belanghebbenden binnen Strukton Worksphere.

Tabel 3-1: Doelstellingen MoSCoW

Requirements	Prioriteit
Een eindgebruiker kan data uit PAB2-artikelbestanden naar de databases van eCatalog en InstallCalc importeren.	Must Have
Een eindgebruiker kan data uit Mod-bestanden toevoegen aan de databases van eCatalog.	Must Have
Een eindgebruiker kan data uit ICC-kortingsbestanden importeren in de database van eCatalog.	Should Have
Een eindgebruiker moet inkoopkortingen in eCatalog met InstallCalc kunnen vergelijken.	Could Have
Een eindgebruiker moet artikelen in InstallCalc, eCatalog en een PAB2-bestand met elkaar kunnen vergelijken.	Could Have
Een eindgebruiker moet verkoopkortingen met inkoopkortingen kunnen vergelijken.	Could Have
Een eindgebruiker kan rapporten genereren met gewijzigde artikelen.	Could Have
Een eindgebruiker kan rapporten genereren met artikelen waarvan een prijswijziging heeft plaatsgevonden.	Must Have
Een eindgebruiker kan rapporten genereren met artikelen waarvan de stamgegevens zijn veranderd.	Must Have
Een eindgebruiker kan rapporten genereren met kortingsgroepen die vervallen of veranderd zijn.	Should Have
Een eindgebruiker kan rapporten genereren met artikelen van kortingsgroepen die vervallen of veranderd zijn.	Should Have
Een eindgebruiker kan rapporten genereren voor de afdeling inkoop met kortingsgroepen waarvan de kortingen of condities vervallen of veranderd zijn. Hiervoor moeten Business Rules voor	Should Have



kunnen worden gedefinieerd. Bijvoorbeeld alleen de artikelen waarbij het prijsverschil hoger is dan een bepaald percentage.	
Een eindgebruiker kan rapporten genereren met artikelen waarvan de gegevens in de verschillende databases en bestanden niet overeen komen.	Could Have
Een eindgebruiker kan rapporten genereren met dossiers waarvan de inkoop- of verkoopkorting niet klopt.	Could Have
Een eindgebruiker moet rapporten kunnen genereren van verschillen in verkoopkorting tussen Focus en InstallCalc.	Could Have
De Business Rules die de applicaties gebruiken moeten gewijzigd kunnen worden.	Must Have
De standaardkortingen in de InstallCalc database moeten naar de eCatalog database geschreven kunnen worden.	Should Have
De belasting voor een medewerker van de afdeling gegevensbeheer voor het verwerken van artikel- en kortingsgegevens mag maximaal één uur per dag zijn.	Must Have
Grote mutaties moeten 's avonds zonder menselijke tussenkomst kunnen worden uitgevoerd om de belasting op de databases te minimaliseren.	Must Have
Business Rules moeten binnen gemiddeld 5 minuten aan te passen zijn door medewerkers van de afdeling Gegevensbeheer of Applicatiebeheer.	Should Have

Voor Strukton Workspere is het voornaamste doel van dit project dat men nieuwe artikelgegevens efficiënter kan verwerken. Dit kost momenteel namelijk erg veel tijd en inzet van medewerkers. Tevens is er hierdoor niet de mogelijkheid om alle data in het systeem, met de gewenste frequentie, up-to-date te houden. Hierdoor kan het voorkomen dat artikelen gefactureerd worden aan klanten voor prijzen die niet meer haalbaar zijn. Dit kost Strukton Workspere dus geld.

Om deze reden is er voor gekozen om een hogere prioriteit te geven aan het verwerken van nieuwe gegevens, dan het vergelijken van data. Het vergelijken van data gebeurt momenteel voornamelijk voor het opsporen van fouten indien er bij de afdeling gegevensbeheer een klacht binnenkomt. Tevens worden hiermee wijzigingen in receptartikelen gecontroleerd. In de nieuwe situatie wil Strukton

Worksphere echter een actievere houding innemen bij het preventief controleren van gegevens zodat fouten tot een minimum beperkt blijven.

3.7.Probleemgebieden

Bij deze opdracht zijn een aantal probleemgebieden al op voorhand te bepalen. Deze probleemgebieden komen voornamelijk door de grote hoeveelheid aan gegevens die verwerkt moeten worden.

De probleemgebieden zijn:

- Performance
- Memory Management
- Flexibiliteit
- Functionaliteit

Performance

Er moet worden gekeken hoe het nieuwe softwaresysteem gemaakt kan worden zodat de belasting voor de eindgebruiker binnen de gestelde eisen valt en de gegevens in een nacht verwerkt kunnen worden.

Memory Management

Omdat er veel gegevens moeten worden verwerkt en de hoeveelheid intern geheugen van de computer (ook wel RAM) beperkt is, moet er bewust worden omgegaan met deze resource.

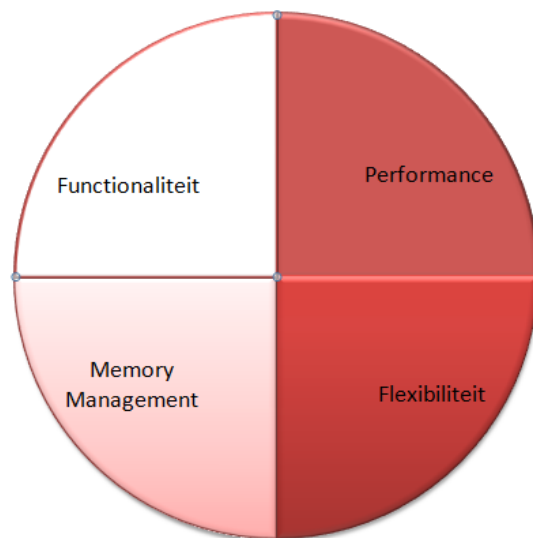
Er moet dus gekeken worden hoeveel gegevens in weinig geheugen kunnen worden geladen en op welk moment dit zou moeten.

Flexibiliteit

Omdat er in de toekomst wijzigingen kunnen zijn in de Business Rules (regels die de bedrijfstoestand definiëren) of omdat er extra functionaliteit gewenst is bij het softwaresysteem moet er gekeken worden naar toevoegen van flexibiliteit in het nieuwe software systeem.

Functionaliteit

Er moet gekeken worden of alle gewenste functionaliteit in het nieuwe softwaresysteem wordt opgenomen zonder dat er te veel moet worden ingeleverd op de andere probleemgebieden.



Figuur 3-4: Spanningsveld

Spanningsveld

Deze probleemgebieden geven een onderling spanningsveld. Als op één van deze probleemgebieden verbetering wordt aangebracht zal dit meestal ten koste gaan van



de rest. Er moet dus gezocht worden naar een evenwicht tussen deze probleemgebieden.

3.8. Ontwikkelmethoden

Er zijn verschillende methoden voor het ontwikkelen van software zoals het V-Model, DSDM en Agile. Voor dit project is een keuze gemaakt uit deze methoden op basis van de volgende aspecten:

- Nauwe samenwerking met de eindgebruikers.
- Uitgebreide documentatie noodzakelijk.
- Een aantal aspecten moeten proefondervindelijk worden vastgesteld. Hierdoor is grote flexibiliteit gewenst.

Tijdens dit project is er gebruik gemaakt van een variant op het V-Model en Agile. Hierbij zijn eerst de grote lijnen van het project uitgezet. Nadat de richting van het project duidelijk was is het ontwerp gedetailleerder gemaakt door de wensen per component te specificeren. In tegenstelling tot een echt Agile ontwikkelmethode worden niet alle tussenstappen los gedocumenteerd en getest.

3.9. Ontwikkelomgeving

Door Strukton Worksphere zijn geen eisen geformuleerd voor de taal waarin het software systeem, dat tijdens dit project wordt gemaakt, moet worden geschreven. Aangezien de afdeling ICT Applicatiebeheer al hun software in C# .Net schrijft ligt deze taal voor de hand.

Binnen Strukton Worksphere draaien alle systemen op Microsoft Windows en zijn alle databases Microsoft SQL Servers.

De ontwikkeling van het softwaresysteem zal gebeuren met het programma "Microsoft Visual Studio 2012". Hiernaast wordt gebruik gemaakt van de databasetools "Microsoft SQL Server Management Studio" en "SQL Server Profiler" en de decompiler JetBrains dotPeek.

3.10. Producten van het Project

Tijdens dit project zullen de volgende producten worden gemaakt:

- Plan van Aanpak
- Functioneel Ontwerp
- Technisch Ontwerp
- Gebruikershandleiding
- Source Code van het softwaresysteem
- Installer met het softwaresysteem

4. Bestanden

Zoals aangegeven in de opdrachtomschrijving moeten er meerdere bestanden verwerkt worden om alle artikel- en kortingsgegevens op te nemen in de systemen van Strukton Worksphere.

Hieronder volgt een beschrijving van de verschillende bestanden die gebruikt worden.

PAB2

PAB2 is een collectie met bestanden die wordt aangeleverd door het tussenbedrijf 2BA. Deze bestanden bevatten artikelgegevens van alle toeleveranciers van Strukton Worksphere. PAB2 bestanden zijn tekstbestanden waarin de kolommen TAB-gescheiden zijn.

De PAB2 collectie bestaat uit de volgende bestanden.

ArtLev.txt

Dit bestand bevat informatie over handelsartikelen.

Product.txt

Hierin staat informatie over producten.

ProdSpec.txt

In dit bestand staan extra specificaties van de producten.

Relatie.txt

Het Relatie.txt bestand bevat informatie over de verschillende toeleveranciers.

Mod

Mod bestanden zijn aanvullingen op de PAB2 bestanden. Deze bestanden bevatten extra informatie over de handelsartikelen en producten. Alleen de toeleverancier Technische Unie levert deze bestanden aan.

Er bestaan de volgende Mod-bestanden.

ModArtPlus.txt

Hierin staat aanvullende informatie over handelsartikelen.

ModProduct.txt

Dit bestand bevat aanvullende informatie over de producten.

ModProdSpec.txt

In dit bestand staan aanvullende specificaties van de producten.

ModSynoniemen.txt

Hierin staan synoniemen voor productgroepen.



Strukton
Worksphere

ICC

ICC-bestanden worden rechtstreeks door de toeleveranciers van Strukton Worksphere aangeleverd.

De eerste regel van een ICC-bestand is het Beginrecord. Dit record bevat informatie over de toeleveranciers van het bestand. De overige records in het bestand zijn Conditierecords. Deze bevatten inkoopkortingen van handelsartikelen.

Alle regels in het ICC-bestand hebben kolommen die een vaste lengte hebben.

5. Softwaresysteem

Omdat de artikel- en kortingsbestanden door verschillende bedrijven worden aangeleverd komen deze bestanden bijna nooit tegelijk bij Strukton Worksphere aan. Hierdoor worden deze bestanden ook in twee aparte workflows verwerkt.

Aangezien er twee verschillende workflows zijn, is er in het nieuwe softwaresysteem voor gekozen om deze workflows als twee losse applicaties te maken. Deze losse applicaties worden wel samen geïnstalleerd.

Daarnaast is er een losse applicatie bedacht voor het vergelijken van gegevens in de bestanden en de verschillende databases. Deze laatste applicatie zal tijdens de scope van dit project niet worden gemaakt, maar is wel beschreven in het functioneel ontwerp.

De nieuwe applicaties zullen worden:

- Artikelverwerker
- Kortingsverwerker
- Vergelijker

Voor de Artikelverwerker en Kortingsverwerker is er gekozen voor het maken van Console Applicaties. Dit zijn applicaties die draaien in een soort "DOS"-scherm.

Er is voor Console Applicaties gekozen, omdat bij deze programma's het verwerken van de gegevens het belangrijkste is en mooiheid van de gebruikers interface (UI) van ondergeschikt belang is. Dit betekent overigens niet dat het niet hebben van een uitgebreide UI ten koste moet gaan van de functionaliteit of bedienbaarheid van de applicaties.

Voor de Vergelijker applicatie is er wel gekozen voor het maken van een applicatie met een uitgebreide UI. Het bekijken van grote hoeveelheden gegevens, om te vergelijken, is voor de gebruiker duidelijker met een Windows Forms applicatie. Deze Vergelijker applicatie wordt tijdens dit project niet gemaakt. Hiervoor is te weinig tijd tijdens dit project en deze is van ondergeschikt belang aan de overige applicaties.

6. Nieuwe Situatie

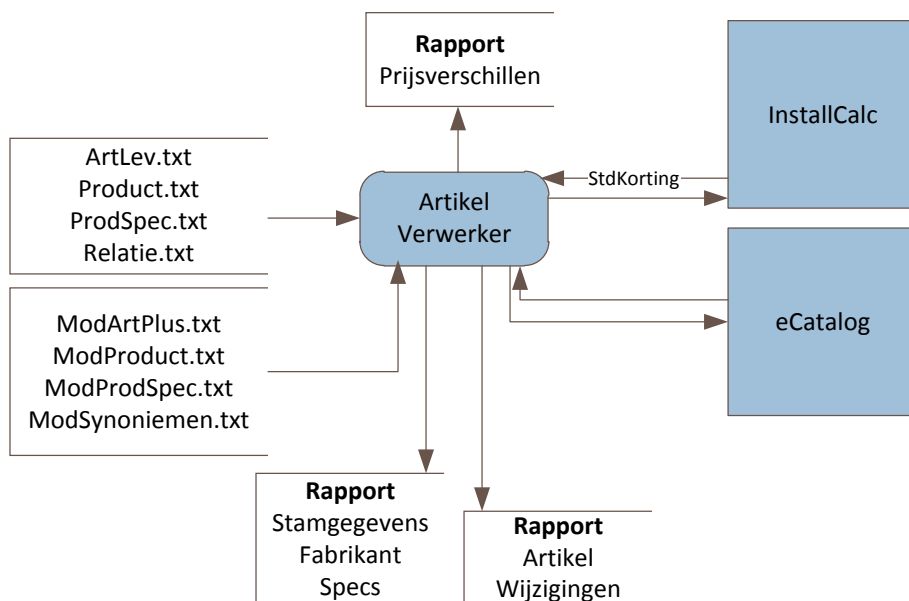
In dit hoofdstuk wordt de nieuwe manier van handelen van zowel de eindgebruiker als van het nieuwe softwaresysteem uitgelegd. In de hoofdstukken hierna zullen de verschillende aspecten worden beschreven waar tijdens het maken van dit nieuwe systeem rekening mee is gehouden.

6.1. Artikelverwerker

Voor het verwerken van de artikelbestanden moeten deze PAB2- en Mod-bestanden in de ophaalmap gezet worden door de eindgebruiker. Hierna kan de Artikelverwerker applicatie worden opgestart.

De applicatie vraagt aan de gebruiker of, en voor welke toeleveranciers, de standaardkortingen in de InstallCalc database naar de eCatalog database geschreven moeten worden. Indien de gebruiker hier relatiecodes van de toeleveranciers opgeeft worden hiervan de standaardkortingen overgezet. De gebruiker kan er ook voor kiezen om deze stap over te slaan.

Na het overzetten van de standaardkortingen wordt het ArtLev.txt bestand ingeladen. Op de gegevens in dit bestand worden de Business Rules toegepast voor InstallCalc met behulp van de Rules Engine. Van de gegevens wordt nu een nieuw ArtLev.txt bestand gemaakt speciaal voor het inlezen naar InstallCalc. InstallCalc wordt opgestart door de Artikelverwerker applicatie en krijgt de opdracht het nieuw aangemaakte ArtLev.txt bestand te importeren.



Figuur 6-1: Artikelverwerker Nieuwe Situatie

Terwijl het ArtLev.txt bestand door InstallCalc wordt geïmporteerd gaat de Artikelverwerker verder met het toepassen van de Business Rules op het originele ArtLev.txt bestand. Hierna importeert de Artikelverwerker de gegevens van dit bestand zelf naar de database van eCatalog.

Indien er tijdens het importeren een handelsartikel in het ArtLev.txt bestand wordt gevonden waarbij de kortingsgroep onbekend is, zal deze niet worden geïmporteerd en opgeslagen in de cache van de applicatie. Zodra beide databases de gegevens in het ArtLev.txt bestand hebben geïmporteerd zal de gebruiker gevraagd worden kortingspercentages in te voeren voor alle onbekende kortingsgroepen. Hierna worden alle handelsartikelen in de cache geïmporteerd naar InstallCalc en eCatalog.

Nu is het ArtLev.txt bestand volledig geïmporteerd en gaat de applicatie verder met het importeren van de overige PAB2-bestanden, gevolgd door de Mod-bestanden. Indien er tijdens het importeren van één van deze bestanden een record wordt gevonden die aan de criteria voldoet om aan een rapport te moeten worden toegevoegd zal dit gebeuren.

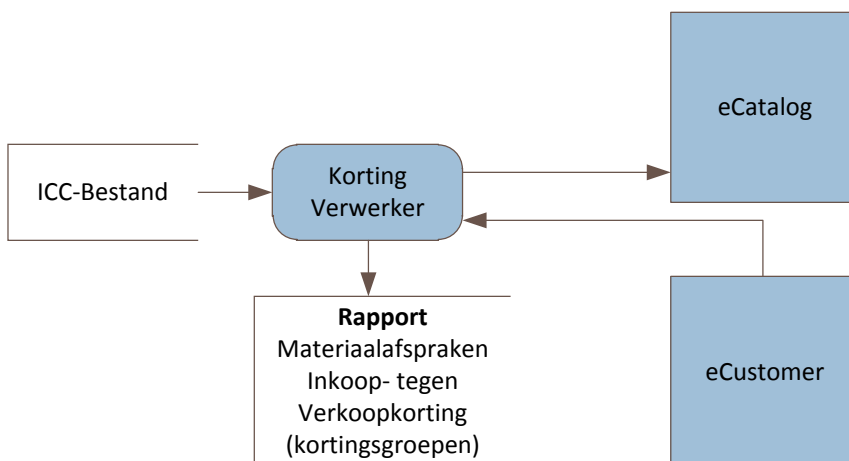
Aan het einde van de applicatie laat deze zien hoe lang er is gedaan over het importeren van de verschillende bestanden. De gebruiker kan nu de applicatie afsluiten met een druk op een toets.

In VII vindt u een volledig overzicht van de Artikelverwerker workflow in de vorm van een Activity Diagram.

6.2.Kortingsverwerker

Voor het verwerken van het ICC-kortingsbestand moet deze ook in een ophaalfolder worden gezet zoals ingesteld in het configuratiebestand.

Zodra de gebruiker de kortingverwerkingsapplicatie opstart zal deze het ICC-bestand inladen en deze omzetten naar het formaat dat ingelezen kan worden in de eCatalog database. Hierna worden de gegevens in eCatalog geïmporteerd en wordt er een rapport gegenereerd over de kortingsgroepen waarvan de kortingen zijn veranderd.



Figuur 6-2: Kortingsverwerker Nieuwe Situatie

In VIII vindt u een volledig overzicht van de Kortingsverwerker workflow in de vorm van een Activity Diagram.

Aan het einde van beide applicaties heeft de gebruiker informatie over het importeren in de vorm van verschillende rapporten en de logboeken.



7. Performance

Omdat er met de nieuwe software veel gegevens verwerkt moeten worden is performance een belangrijk aspect. Een tijdswinst van een halve seconde op het muteren van één record lijkt misschien niet veel, maar als een applicatie meer dan een miljoen records moet verwerken schiet dit aardig op.

Voor dit project is op verschillende punten gekeken naar performance, zoals bij het inlezen van bestanden en de technieken voor het schrijven naar de verschillende databases. Die performanceaspecten zullen in dit hoofdstuk niet beschreven worden. Raadpleeg hiervoor de betreffende hoofdstukken.

In dit hoofdstuk wordt er gekeken bij welke taken tijdens de workflows van het systeem performance verbeteringen kunnen worden aangebracht. Hierbij wordt niet alleen rekening gehouden met de tijd die de computer bezig is met het uitvoeren van een taak, maar ook met het minimaliseren van de tijd dat de computer en de gebruiker op elkaar moeten wachten.

De taken die veruit het langste duren zijn die waarbij gegevens worden geïmporteerd naar de verschillende databases. Omdat dit meerdere uren kan duren is het belangrijk dat dit automatisch kan en dat de gebruiker hierbij dus niet steeds hoeft op te letten en gegevens hoeft in te voeren.

7.1. Onbekende Kortingsgroep

Bij het importeren van de ArtLev.txt bestanden kan het zijn dat de kortingsgroep die in het bestand aan een handelsartikel hangt niet bekend is in de databases. Hiervoor moet de gebruiker dus handmatig gegevens invoeren.

Aangezien de applicatie de gegevens in het ArtLev.txt bestand één voor één verwerkt zou de gebruiker steeds tussendoor gegevens moeten invoeren. Hierbij kan de applicatie niet zelfstandig werken, moet de gebruiker bij het importeren bij de computer blijven en is er steeds een wachttijd tussen het van de computer vragen van het percentage van de kortingsgroep en het invoeren van de gebruiker. Dit is dus geen ideale situatie.

Om dit probleem op te lossen is er voor gekozen om gebruik te maken van een cache. Handelsartikelen waarvan de kortingsgroep niet bekend is worden niet meteen toegevoegd aan de databases van InstallCalc en eCatalog. Deze handelsartikelen worden tijdelijk opgeslagen in een cache terwijl de overige handelsartikelen verwerkt worden. Aan het einde van de applicatie wordt achter elkaar aan de gebruiker gevraagd de kortingspercentages van de onbekende kortingsgroepen in te vullen. Hierna worden alle handelsartikelen in de cache in de databases geïmporteerd.

7.2.Multitasking

Omdat het importeren van de gegevens zo lang kan duren, zou het een stuk sneller worden als verschillende taken tegelijk zouden kunnen worden uitgevoerd.

Iedere applicatie kan taken uitvoeren door middel van “threads”. Deze threads kunnen maar één taak tegelijk uitvoeren. Door het gebruiken van meerdere threads binnen een applicatie kunnen verschillende taken tegelijk worden uitgevoerd. Dit gebruik van meerdere threads heet multithreading.

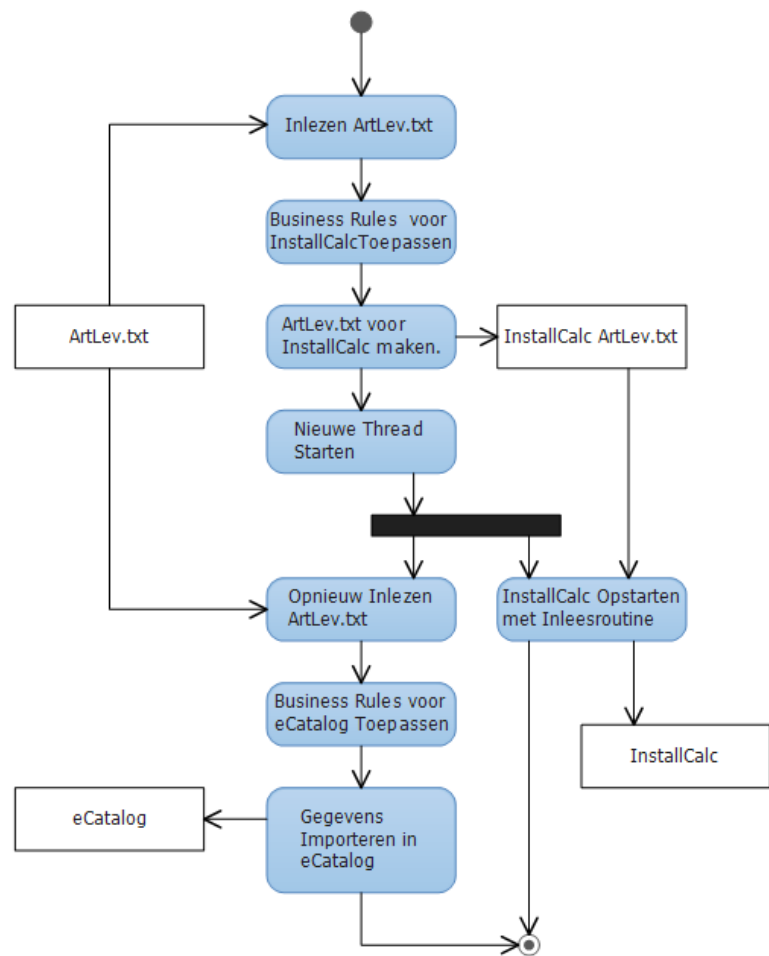
Bij het importeren van artikelgegevens is gekozen voor het tegelijk importeren naar de databases van InstallCalc en eCatalog. Hiervoor worden dus twee threads gebruikt. Doordat deze twee importetaken tegelijk uitgevoerd worden is er een enorme tijdswinst aangezien beide taken meerdere uren kunnen duren.

Bij multithreading kunnen er wel problemen ontstaan met betrekking tot concurrency. Deze problemen kunnen bijvoorbeeld ontstaan doordat verschillende threads bewerkingen willen uitvoeren op dezelfde gegevens.

Als de acties van de verschillende threads niet gesynchroniseerd worden is er kans op datacorruptie doordat oude gegevens worden uitgelezen of doordat gegevens worden overschreven.

Tevens kan het niet synchroniseren van threads leiden tot een deadlock of resource starvation. Bij een deadlock wachten beide threads totdat de andere klaar is met een bewerking, en staan dus oneindig op elkaar te wachten. In het geval van resource starvation krijgt een thread nooit toegang tot een resource en kan dus nooit een taak afmaken. In deze ernstige gevallen hangt of crasht de applicatie.

Problemen met concurrency kunnen worden opgelost door middel van synchronisatie. Hierbij wordt ervoor gezorgd dat meerdere concurrently-executing



Figuur 7-1: Artikelverwerker Multithreading



threads nooit tegelijk een resource kunnen benaderen. Als één thread een resource in gebruik heeft moeten de andere threads wachten totdat deze weer is vrijgegeven.

Voor het verwerken van de artikelgegevens is er geen sprake van synchronisatie, omdat beide threads binnen de applicatie geen resources hoeven delen. Op het moment dat er begonnen wordt met het importeren naar de databases van InstallCalc en eCatalog hebben deze allebei hun eigen gegevens.

Het logboek moet wel gesynchroniseerd worden, omdat alle threads gebruik maken van het zelfde logbestand (zelfde resource).

Zie Figuur 7-1: Artikelverwerker Multithreading voor een simplistische weergave van de multithreading in de artikelverwerker.

8. Flexibiliteit

Het is een eis van Strukton Worksphere dat het in de toekomst mogelijk is aanpassingen te doen aan het nieuwe softwaresysteem zodat kan worden ingespeeld op bijvoorbeeld veranderingen in Business Rules of als er meer functionaliteit van het systeem gewenst is.

Voor het creëren van deze flexibiliteit is er gekeken naar twee aspecten, namelijk het modulair maken van het softwaresysteem en het toepassen van een Rule Engine om Business Rules eenvoudig aan te passen en overzichtelijk te houden.

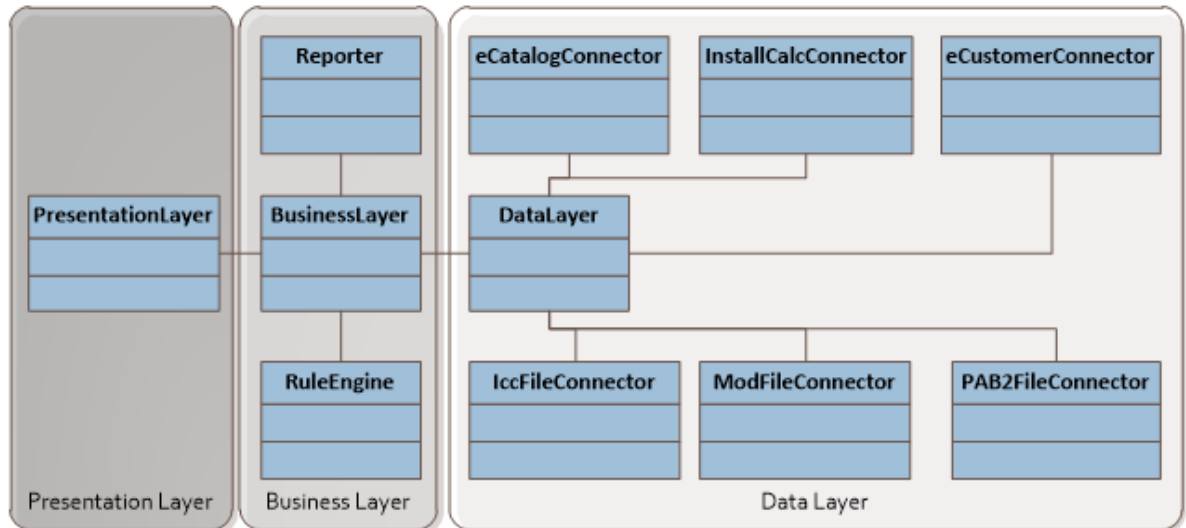
8.1.Modulariteit

Modulariteit binnen een softwaresysteem is het opdelen van de software in kleinere stukjes (modules). Deze modules hebben hun eigen taken en verantwoordelijkheden binnen de software.

Het toepassen van modulariteit aan een software applicatie heeft de volgende voordelen:

- Het is makkelijk om functionaliteit te veranderen omdat alle functionaliteit gescheiden is in modules.
- Er kunnen makkelijk modules worden toegevoegd om zo de functionaliteit van de applicatie te vergroten.
- Door het toepassen van modulariteit kunnen makkelijk delen van een applicatie worden getest door modules afzonderlijk te testen.
- Modules van dit softwaresysteem kunnen worden gebruikt in andere softwaresystemen. Hierdoor hoeft er voor andere systemen die dezelfde functionaliteit nodig hebben niet opnieuw hetzelfde worden geprogrammeerd.

In het volgende packagediagram is de modulariteit van het nieuwe softwaresysteem in kaart gebracht.



Figuur 8-1: Packagediagram Nieuwe Systeem

Software kan op verschillende manieren worden opgebouwd zonder dat de eindgebruiker hier iets van merkt. Voor dit project is er gekozen voor een scheiding tussen de gegevens die de applicatie gebruikt, de interface die de eindgebruiker ziet en de communicatie tussen deze twee. Deze manier van scheiden heet het Three Layer Model.

8.1.1. Three Layer Model

Bij het Three Layer Model is een applicatie in de volgende lagen opgedeeld.

- Presentation Layer
- Business Layer
- Data Layer

8.1.1.1. Presentation Layer

De Presentation Layer is het deel van de software die voor de eindgebruiker te zien zal zijn. Deze bestaat uit de Graphical User Interface (GUI) en alles wat deze nodig heeft voor het communiceren met de Business Layer.

Omdat de applicaties van dit softwaresysteem gemaakt zijn als Console Applicaties bevat deze laag een beperkte GUI. Door toch te kiezen voor het toepassen van het Three Layer Model is het in de toekomst makkelijk om een uitgebreidere GUI toe te voegen.

De Presentation Layer bevat zelf geen functionaliteit, hiervoor is de Business Layer verantwoordelijk.

8.1.1.2. Business Layer

De Business Layer is de laag die tussen de Presentation Layer en de Data Layer in zit. Deze zorgt er voor dat de Presentation Layer informatie (data) kan opvragen uit en wegschrijven naar alle bronnen die gekoppeld zijn aan de Data Layer. De Presentation Layer kan dus nooit rechtstreeks met de Data Layer communiceren.

Tevens zorgt de Business Layer dat er eventuele berekeningen of bewerkingen op de data worden uitgevoerd. Dit is dus de laag waarin het grootste deel van de intelligentie van een applicatie zit.

Voor dit softwaresysteem beschikt de Business Layer, naast een Data Layer, over een Reporter om rapporten te maken en een Rule Engine om Business Rules toe te passen.

8.1.1.3. Data Layer

De Data Layer is verantwoordelijk voor alle data in de applicatie. Deze zorgt dus dat er data voor de applicatie beschikbaar is om mee te werken en schrijft nieuwe of aangepaste informatie naar de juiste informatiebron. Informatiebronnen kunnen o.a. een database of een bestand zijn.



8.2. Rule Engine

Business Rules

Business Rules definiëren of beperken een aspect van een bedrijf.
Een algemeen voorbeeld van een Business Rule is:

“Als een artikel in kortingsgroep 15 zit mag de netto prijs niet hoger zijn dan €100.”

Deze Business Rules kunnen na verloop van tijd veranderen, hetgeen invloed kan hebben op de softwaresystemen binnen een bedrijf.

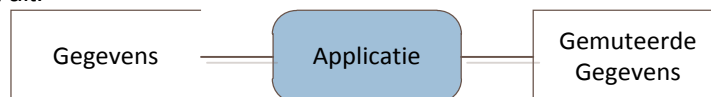
De Business Rule die voor dit project van belang zijn, zijn aangeleverd door Strukton Workspace.

Er zijn twee manieren om Business Rules te verwerken in een softwaresysteem. Namelijk “hard coded” of met behulp van een Rule Engine.

Hard Coded

Als Business Rules “hard coded” in een applicatie zitten, zijn deze in de applicatie geprogrammeerd. Om deze aan te passen zal dus de source code van de applicatie moeten worden gewijzigd en zal de applicatie opnieuw moeten worden gecompileerd.

Als een applicatie hard coded Business Rules toepast op een set gegevens ziet dit er als volgt uit.



Figuur 8-2: Hard Coded

De applicatie laadt de gegevens en muteert deze op basis van de hard coded Business Rules die her en der verspreid door de applicatie staan.

Voordelen

Hard coded Business Rules zijn snel te implementeren, omdat hiervoor geen volledige Rule Engine hoeft te worden geschreven of te worden gebruikt. Tevens is uitvoeren van hard coded Business Rules sneller dan met behulp van een Rule Engine. De performance is dus hoger.

Nadelen

Als Business Rules veranderen is het veel meer werk om deze aan te passen in de applicatie. Tevens zullen deze wijzigingen gedaan moeten worden door iemand die een behoorlijke kennis heeft van de programmeertaal van de applicatie (in dit geval C#). Ook is de kans dat er nieuwe bugs in het systeem worden toegevoegd door deze aanpassingen groter omdat niet altijd duidelijk is wat de invloed is van de wijzigingen op de applicatie.

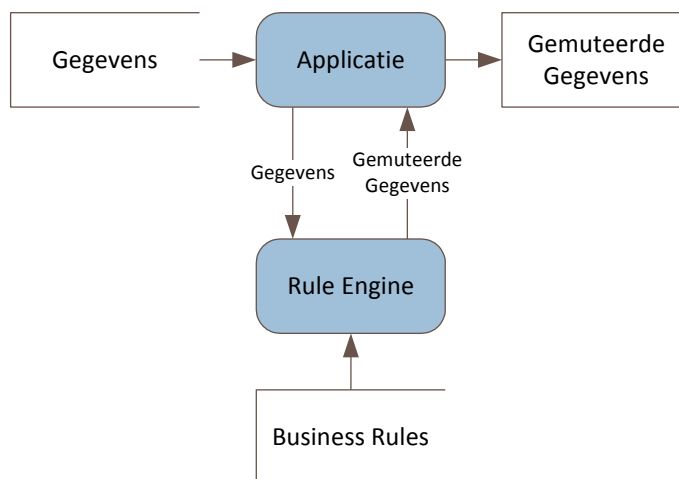
De Business Rules zijn bij een hard coded implementatie ook niet afgezonderd van de rest van de applicatie. Deze staan tussen de andere code. Hierdoor is er minder sprake van modulariteit en zijn de Business Rules niet overzichtelijk.



Rule Engine

Een Rule Engine zorgt dat Business Rules worden vertaald naar code waarmee data gemuteerd kunnen worden. Met behulp van zo'n Rule Engine kunnen deze Business Rules makkelijk worden aangepast zonder aanpassingen aan de code van de applicatie te hoeven doen. Hierdoor kunnen deze aanpassingen door een eindgebruiker zelf gedaan worden. Er is dus geen ontwikkelaar meer nodig na oplevering van de applicatie.

Een Rule Engine werkt als volgt:
De applicatie heeft een set gegevens.
Deze gegevens worden aan de Rule Engine gegeven. De Rule Engine muteert alle gegevens op basis van de gedefinieerde Business Rules. Deze gemuteerde gegevens worden teruggestuurd naar de applicatie.



Voordelen

Figuur 8-3: Rule Engine

Flexibiliteit:

Bij het gebruik van een Rule Engine hoeft de applicatie niet opnieuw gecompileerd en deployed te worden als een Business Rule is aangepast.

Eigendomsrecht:

Als Business Rules “hard coded” zijn, zijn ze eigendom van de ontwikkelaar. Bij een Rule Engine kunnen de eindgebruikers of beheerders zelf hun eigen Business Rules beheren.

Transparantie:

Als er een fout in een Business Rule staat hoeft een ontwikkelaar niet de hele code door te spitten, met een debugger, op zoek naar de fout. Een eindgebruiker kan zelf met de Rule Engine naar de fout zoeken.

Kracht:

Het is veel makkelijker om ingewikkelde Business Rules met elkaar te combineren met behulp van een Rule engine, dan met “if-then-else” code, in een applicatie.

Nadelen

De initiële implementatie van een Rule Engine kost meer tijd dan het “hard coden” van de Business Rules. Tevens zal de performance van de applicatie iets lager zijn omdat alle gegevens door de Rule Engine moeten worden gehaald.

Om te beslissen of het toepassen van een Rule Engine bij dit project rendabel is zijn de voor en nadelen hiervan gewogen.

Tabel 8-1: Weging Hard Coded vs Rule Engine

Methode	Flexibiliteit	Modulariteit	Implementatie snelheid	Performance
Hard Coded	- -	-	+	+
Rule Engine	+ +	+	-	-

Uit deze weging blijkt dat de voordelen van het gebruik van een Rule Engine groter zijn dan de nadelen. Het grote voordeel van de Rule Engine met betrekking tot de flexibiliteit om wijzigingen in Business Rules in de toekomst toe te kunnen passen in het softwaresysteem weegt zwaarder dan het snel kunnen programmeren van de applicaties tijdens de ontwikkelfase. Daarom is er voor dit project gekozen om gebruik te maken van een Rule Engine.

8.2.1. Verschillende Rule Engines

Nu bepaald is dat er gebruik gemaakt gaat worden van een Rule Engine moet er nog gekeken worden naar welke specifieke Rule Engine er voor dit project gebruikt gaat worden. Omdat er veel verschillende Rule Engines zijn, is er eerst een lijst met wensen opgesteld voor de Rule Engine die we gaan gebruiken.

Deze wensen zijn:

- Makkelijk en snel te implementeren.
- Moet goed gedocumenteerd zijn.
- Moet een actief project zijn (krijgt updates).
- Moet commercieel gebruikt kunnen worden met behoud van de gesloten licentie van de applicatie.
- Business Rules moeten aan te passen zijn met weinig tot geen training van eindgebruikers.
- Open-source is een pre.

Op basis van deze wensen zijn de volgende Rule Engines vergeleken:

- Windows Workflow Rules Engine
- Drools .NET
- inRule
- Rules Engine



Windows Workflow Foundation (WWF) Rule Engine

Deze Rule Engine van Microsoft maakt deel uit van het .NET Framework. Hierdoor kan deze net als de rest van het .NET Framework gratis gebruikt worden voor zowel commerciële als persoonlijke doeleinden.

De Windows Workflow Foundation Rule Engine heeft zijn eigen UI klassen waardoor een applicatie voor het aanpassen van Business Rules snel en makkelijk te maken is.

De Business Rules zijn bij de Rule Engine gedefinieerd in C#. Omdat de syntax voor deze Business Rules redelijk makkelijk is en de gebruiker helpt met behulp van autocomplete is het wijzigen hiervan te doen na beperkte training van de gebruiker en een aantal voorbeelden in de handleiding.

Microsoft BizTalk Server

BizTalk is een Enterprise Service Bus met een Rule Engine als subfunctie. Hierdoor is deze Rule Engine minder uitgebreid dan die van specifieke Rule Engine applicaties. BizTalk draait apart op een server en niet als onderdeel van een applicatie.

Het veranderen van Business Rules in BizTalk is te moeilijk om de eindgebruikers te laten uitvoeren. De BizTalk Rule Engine is meer bedoeld als een centraal punt voor alle Business Rules die beheerd worden door IT'ers dan als een Rule Engine voor eindgebruikers. Er bestaat wel veel documentatie over de implementatie van een BizTalk.

Ook als alleen de Rule Engine van BizTalk gebruikt wordt moet een volledige Biztalk Server licentie aangeschaft worden. Prijs: \$10,138.

Drools .NET

Drools .NET is een .NET poort van de bekende Drools Rule Engine voor JAVA. In tegenstelling tot zijn JAVA tegenhanger is dit een klein project met weinig documentatie dat sinds 2010 niet meer is geüpdate.

Drools .NET is wel gratis, open-source en heeft de volgende licenties:

- Apache Software License
- BSD License
- MIT License

Met deze licenties mag Drools .NET gratis gebruikt worden in commerciële closed source applicaties.

inRule

Dit is een Rule Engine speciaal gemaakt voor .NET. InRule is makkelijk te implementeren en te bedienen voor eindgebruikers zonder de mogelijkheid te verliezen tot complexe Business Rules. Tevens heeft inRule een uitgebreide documentatie met implementatie voorbeelden voor ontwikkelaars.

inRule heeft verschillende licenties die moeten worden aangeschaft. Voor dit project zouden de volgende onderdelen van inRule moeten worden aangeschaft:

irServer

Hierop draait de Rule Engine. De licenties hiervoor zijn gebaseerd op het aantal gebruikers.

irAuthor

Voor het aanpassen van Business Rules. De licenties hiervoor zijn gebaseerd op het aantal gebruikers.

irSDK

integratie van de Rule Engine in applicaties. De licenties hiervoor zijn gebaseerd op het aantal CPU's of cores die gebruikt worden.

Doordat voor al deze componenten losse licenties moeten worden aangeschaft is dit een erg dure oplossing.



8.2.2. Vergelijking

Hieronder vindt u een weging van de voor- en nadelen van de verschillende Rule Engines.

Tabel 8-2: Weging verschillende Rule Engines

Engine	Licentie	Open-source	Documentatie	Prijs	Implementatie snelheid	Gebruiks-vriendelijkheid
Windows Workflow Foundation Rules Engine	++	Ja	++	++	++	+
BizTalk	-	Nee	++	-	-	--
Drools .NET	++	Ja	--	++	-	-
inRule	--	Nee	++	--	+	++

Pakketten als BizTalk en inRule bieden veel meer functionaliteit dan de kleinere Rule Engines. Hierdoor zijn de kosten ook meteen hoger. Aangezien het merendeel van deze functionaliteit in dit project toch niet gebruikt wordt, is er niet voor deze Rule Engines gekozen.

WWF Rule Engine en Drools .NET zijn beide kleine lichtgewicht Rule Engines die uitermate geschikt zijn voor de workflows die in de applicaties van dit project gebruikt worden.

Voor dit project is ervoor gekozen om gebruik te maken van de Windows Workflow Foundation Rules Engine. Deze gratis en open-source Rule Engine is goed gedocumenteerd en makkelijk te implementeren en te onderhouden.

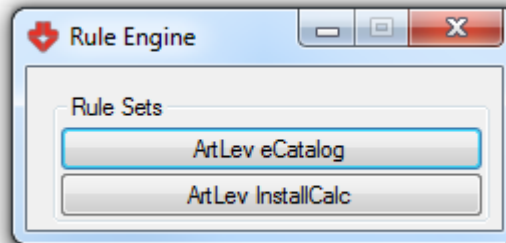
De Business Rules in deze Rule Engine worden opgeslagen in "RuleSets" in ".rules"-bestanden. De locatie van deze bestanden met benodigde extra informatie kan worden beschreven in het configuratiebestand.

Voor het aanpassen van de Business Rules door de eindgebruiker zal een losse applicatie worden geschreven op basis van de UI klassen hiervoor, in het .NET Framework.

8.2.3. Business Rules Aanpassen

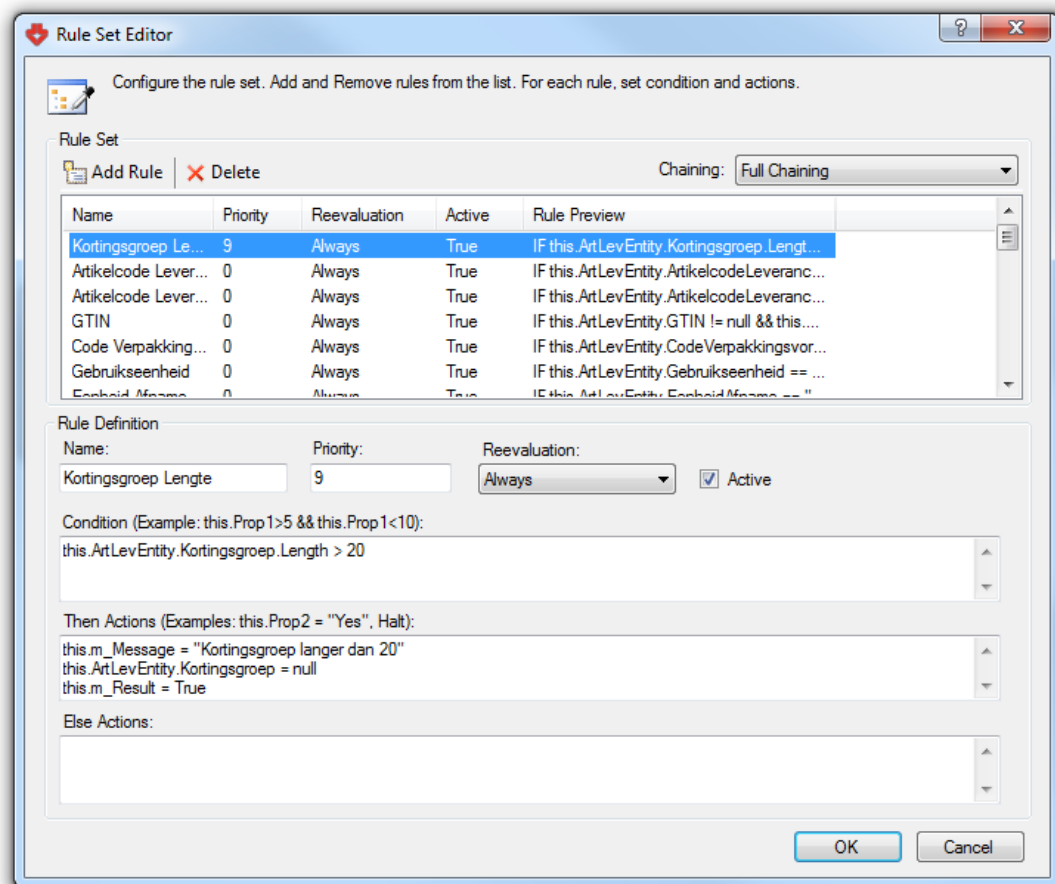
Zoals gezegd heeft de Windows Workflow Foundation Rule Engine heeft zijn eigen UI klassen waardoor het maken van een applicatie voor het aanpassen van Business Rules snel en makkelijk te maken is.

Voor dit project is er een kleine applicatie gemaakt met behulp van deze klassen. In het configuratiebestand worden de Rule Sets gedefinieerd. Deze bij elkaar horende sets van Business Rules worden door de applicatie dynamisch weergegeven in het menu. De gebruiker kan in dit menu selecteren welke Rule Set hij wil aanpassen.



Figuur 8-4: Menu Rule Sets

Nadat de gebruiker een keuze heeft gemaakt krijgt deze een lijst te zien met alle Business Rules in deze set. Hier kan de gebruiker nieuwe Business Rules aanmaken en bestaande wijzigen.



Figuur 8-5: Rule Set Editor

De Business Rules zijn gedefinieerd in C# "if-then-else" statements. Om te zorgen dat eindgebruikers maar minimale kennis van C# hoeven te hebben, worden voorbeelden van de meest gebruikte soorten Business Rules opgenomen in de handleiding.



9. Bestandscommunicatie

De gegevensbestanden die moeten worden uitgelezen bestaan uit twee soorten. Bestanden waarbij de gegevenskolommen een vaste lengte hebben of waar deze gescheiden zijn door een TAB. Om deze bestanden uit te kunnen lezen is gebruik gemaakt van serialisatie.

9.1. Serialisatie

Bij het inlezen van gegevens uit bestanden met behulp van serialisatie worden de regels van het bestand omgezet in door de programmeur voorgedefinieerde klassen. Uit deze klassen kan een programmeur makkelijk informatie opvragen van het record. Om de verschillende soorten bestanden te kunnen uitlezen zijn er twee soorten serializers gemaakt. Één voor het uitlezen van bestanden waarbij de gegevenskolommen een vaste lengte hebben en één voor bestanden waarbij de kolommen gescheiden zijn door TAB's. Na het maken van deze serializers is Serialisatie snel te implementeren voor de verschillende bestanden. Tevens zijn serializers snel in het inlezen en omzetten van de gegevens.

Omdat de verschillende bestanden erg veel informatie kunnen bevatten worden de records in deze bestanden één voor één geserialiseerd en verwerkt. Hierdoor kan het werkgeheugen van de computer niet vollopen en zijn er dus geen problemen met betrekking tot memory management.

9.2. Tijdelijke Bestanden

Voordat een applicatie een bestand gaat verwerken wordt hiervan eerst een tijdelijke kopie gemaakt. Hierdoor is de kans op fouten door handelingen van de gebruiker tijdens het verwerken kleiner.

Voor het importeren van een ArtLev.txt bestand wordt een tijdelijke ArtLev.txt gemaakt waarop de Business Rules al zijn toegepast.



10. Databasecommunicatie

10.1. Huidig Systeem

Bij de huidige manier van gegevens verwerken werd er gebruik gemaakt van twee applicaties om deze gegevens naar de verschillende databases te schrijven. Deze applicaties lezen niet de rechtstreeks aangeleverde bestanden van 2BA. Deze worden eerst voorbereid met behulp van Access.

Voor eCatalog werd gebruik gemaakt van de applicatie FocusETL. Dit is een importprogramma speciaal gemaakt voor het Focus systeem van Strukton Worksphere door een extern bedrijf.

Om gegevens naar de InstallCalc database te schrijven moest men InstallCalc zelf gebruiken. InstallCalc heeft verschillende inleesroutines waarmee de gegevens kunnen worden geïmporteerd.

Op de huidige manier van gegevensverwerking moeten dus twee afzonderlijke applicaties worden gedraaid om te importeren.

10.2. Nieuw Systeem

In het nieuwe systeem is meer functionaliteit gewenst waardoor het nodig is om, behalve met InstallCalc en eCatalog te kunnen communiceren, ook gegevens van eCustomer te kunnen opvragen. Dit is nodig voor het maken van rapporten waarin de inkoopkorting en de verkoopkorting met elkaar worden vergeleken. Daarnaast kan het verwerkingsproces een stuk sneller en met een verminderde belasting voor de medewerkers van de afdeling Gegevensbeheer door het importeren op te nemen in het nieuwe systeem.

In Tabel 10-1 staat een overzicht over wat de applicaties met de verschillende databases moeten kunnen doen.

Tabel 10-1: Database Communicatie

	Artikelverwerker	Kortingsverwerker
eCatalog	Lezen, Schrijven	Lezen, Schrijven
eCustomer	-	Lezen
InstallCalc	Lezen, Schrijven	-

De databases eCatalog en eCustomer zijn beide onderdeel van Strukton Worksphere's eigen Focus systeem. eCatalog bevat informatie over handelsartikelen, kortingen en toeleveranciers. In eCustomer zit informatie over de verkoopkortingen van deze handelsartikelen en de klanten die deze afnemen.

InstallCalc is een applicatie gemaakt door derden en beschikt over zijn eigen database.

De communicatie met zowel eCatalog als eCustomer gebeurt met behulp van het Entity Framework (zie paragraaf 10.3 Entity Framework). Ook het opvragen van gegevens uit de database van InstallCalc gebeurt met het Entity Framework.

Mapping

De structuur van de databases is anders dan die in de bestanden die worden ingelezen. Hierdoor moet er een “mapping” gemaakt worden over hoe de relaties liggen tussen de databasetabellen en de bestanden. Binnen Strukton Worksphere is er helaas weinig kennis aanwezig over deze mapping.

eCustomer

De mapping voor eCustomer was het makkelijkst. Hieruit hoefden maar een paar tabellen te worden opgevraagd en deze waren redelijk bekend bij de medewerkers van Strukton Worksphere. De gegevens in deze tabellen worden alleen gebruikt voor het maken van rapporten.

In bijlage IV staan deze tabellen waaruit gegevens worden opgevraagd.

eCatalog

Voor eCatalog was het maken van een mapping een stuk moeilijker en veel meer werk. Bij eCatalog moet naar veel tabellen geschreven worden en er moeten veel gegevens uit opgehaald worden.

In bijlage V vindt u een overzicht van alle tabellen die van belang zijn voor dit project. Dit is maar een klein deel van de tabellen van de eCatalog database.

Om de mapping van deze tabellen te achterhalen is gekeken naar de importtool die op dit moment gebruikt wordt; FocusETL. Van deze importtool was geen source code beschikbaar en er zat ook geen beschrijving bij over wat deze deed en hoe deze de gegevens wegschreef naar eCatalog.

Om toch te achterhalen hoe de mapping van FocusETL in elkaar steekt is deze tool “reverse engineered”. Dit houdt in dat de gecompileerde FocusETL applicatie die bij de gebruikers is geïnstalleerd, terug werd gebracht naar zijn source code. Hiervan is een testproject gemaakt in Visual Studio en uit de code kon de mapping worden herleid.

Op basis van deze mapping is een library gemaakt voor het nieuwe systeem die alle functionaliteit bevat voor de communicatie met de eCatalog database.

InstallCalc

Ook in de database van InstallCalc moeten veel gegevens worden weggeschreven en ook hier is de databasestructuur afwijkend van de bestandsstructuur.

Voor InstallCalc is er geen importtool die kan worden reverse engineered om te achterhalen wat de mapping is. InstallCalc maakt gebruik van inleesroutines waarvan de mapping niet gezien kan worden.



Voor het opvragen van de gegevens van InstallCalc die nodig waren, was het uitzoeken van de mapping niet zo'n probleem. Dit waren maar een beperkt aantal velden en deze waren makkelijk te vinden. Het opvragen van deze gegevens gebeurt dus ook weer met behulp van het Entity Framework. In bijlage V staat een overzicht van de tabellen die hierbij gebruikt worden.

Het wegschrijven van de gegevens naar InstallCalc is een ander verhaal. Net als bij eCatalog zijn dit veel tabellen waar gegevens naar moeten worden geschreven. Het achterhalen van de mapping hiervan is dus zeer tijdrovend.

Om toch op een makkelijke en snelle manier naar de InstallCalc database te kunnen schrijven is er gebruik gemaakt van de mogelijkheid om InstallCalc op te starten met een opstartroutine. Hierover wordt meer verteld in hoofdstuk 10.4.

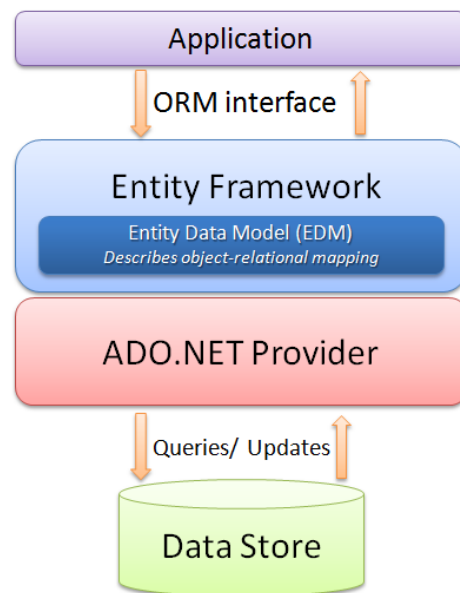
10.3. Entity Framework

Bij het door Microsoft gemaakte Entity Framework kunnen de gegevens in een database worden aangeroepen als objecten in de code van een applicatie. Hierdoor is het makkelijk om de datastructuur van de database aan te passen in de code. Tevens is het makkelijker voor de programmeur, omdat deze zelf geen SQL aanroepen naar de database hoeft te doen. De mogelijkheid is wel aanwezig om via het Entity Framework SQL aanroepen te doen indien dit gewenst is.

Het Entity Framework is een laag tussen de applicatie en de ADO.NET Provider. Dit Framework bevat een kopie van alle tabellen met bijbehorende relaties van de database in de vorm van klassen. De relatie tussen deze klassen en de feitelijke database is beschreven in het Entity Data Model.

De programmeur kan in de applicatie met het Entity Framework communiceren op dezelfde manier als met "normale" klassen.

Als er gegevens naar de database geschreven moeten worden doet de programmeur dit in het Entity Framework. Deze houdt alle veranderingen bij en schrijft deze naar de database zodra de programmeur de methode hiervoor aanroept. Het Entity Framework kan zelf niet rechtstreeks communiceren met de database. Hiervoor moet deze eerst SQL queries genereren op basis van de veranderingen. Deze queries geeft het Framework door aan de ADO.NET Provider welke deze uitvoert. Alle wijzingen in het Entity Framework zijn nu naar de database geschreven.



Figuur 10-1: EF Communicatie

10.3.1. Performance

Omdat er veel gegevens met behulp van het Entity Framework naar de verschillende databases moeten worden geschreven is het wel belangrijk dat dit ook een beetje snel gaat en niet te veel geheugen in beslag neemt.

Om te zorgen dat de performance van het Entity Framework optimaal is, zijn er verschillende manieren van het gebruik hiervan bekeken. In de voorbeelden op de volgende pagina's is gekeken naar de performance van het Entity Framework tijdens het importeren naar de eCatalog database.



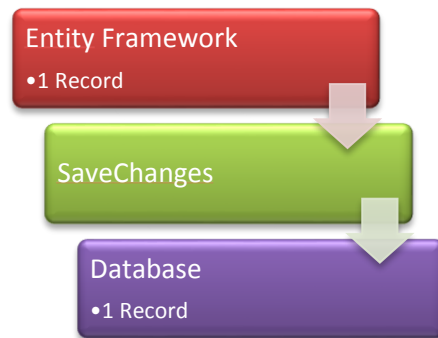
Meteen naar database schrijven

De meest eenvoudige manier om het Entity Framework te gebruiken is om na iedere wijziging deze meteen naar de database te schrijven.

In de Artikelverwerker applicatie ziet dit er als volgt uit.

De applicatie laadt een bestand en maakt van de gegevens één voor één objecten. Van deze objecten wordt één object naar het Entity Framework geschreven. Deze wijziging wordt meteen opgeslagen in de database. Hierna wordt hetzelfde gedaan met de overige gegevens.

Deze aanpak kan prima zijn voor een applicatie die weinig gegevens hoeft te verwerken maar heeft voor dit project twee nadelen.



Figuur 10-2: EF één voor één

Ten eerste kost het aanroepen van de methode om de wijzigingen naar de feitelijk database te schrijven (SaveChanges) de meeste tijd. Bij deze manier van opslaan wordt de methode aangeroepen voor ieder record dat naar de database moet worden geschreven.

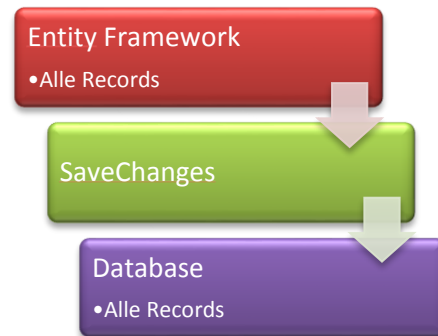
Ten tweede neemt het Entity Framework steeds meer intern geheugen in beslag bij iedere wijziging. Dit komt doordat het Entity Framework alle wijzigingen bij wil houden en hiervoor erg veel metadata bijhoudt. Indien er dus grote hoeveelheden gegevens moeten worden verwerkt zal het geheugen langzaam vollopen en de applicatie crashen.

Aan het einde naar database schrijven.

De tweede manier die gebruikt zou kunnen worden is het schrijven van gegevens naar de database als alle wijzigingen naar het Entity Framework zijn geschreven.

In de Artikelverwerker applicatie ziet dit er als volgt uit.

De applicatie laadt een bestand en maakt van de gegevens één voor één objecten. Deze objecten worden allemaal één voor één naar het Entity Framework geschreven. Als alle wijzigingen in het Entity Framework zijn opgenomen wordt de SaveChanges methode aangeroepen. Nu staan alle wijzigingen in de database.



Figuur 10-3: EF alles tegelijk

Deze wijziging wordt meteen opgeslagen in de database. Hierna wordt hetzelfde gedaan met de overige gegevens.

Nu is het snelheidsprobleem opgelost aangezien de SaveChanges maar één keer wordt aangeroepen.

Het geheugenprobleem is hiermee nog niet opgelost. Nog steeds worden alle wijzigingen naar het Entity Framework geschreven en deze neemt dus nog steeds veel geheugen in beslag bij het verwerken van grote hoeveelheden gegevens.



Batch Systeem

De oplossing is gevonden in het implementeren van een batch systeem. Voor dit batch systeem is een eigen SaveChanges methode gemaakt. Deze methode kan na iedere wijziging worden aangeroepen maar houdt zelf bij of de gegevens ook naar de database moeten worden geschreven. Dit wordt bepaald door drie factoren.

Ten eerste kan worden opgegeven na hoeveel records er feitelijk naar de database moet worden geschreven. Zodra het Entity Framework dus bijvoorbeeld 500 wijzigingen heeft, worden deze naar de database geschreven.

Daarnaast wordt bijgehouden of het record dat verwerkt wordt het laatste is van de gegevensset. Indien dit zo is moet deze sowieso naar de database worden geschreven ongeacht of de batchhoeveelheid is behaald.

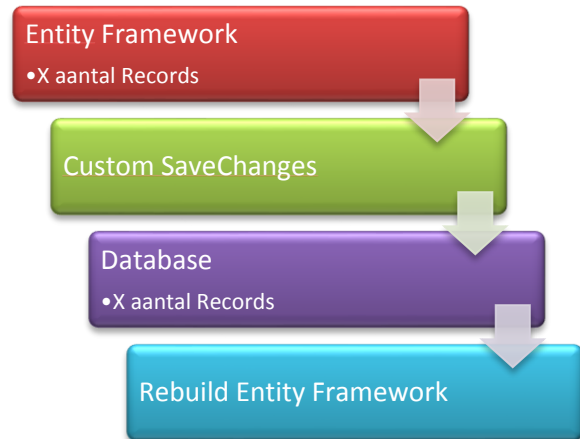
Als laatste kan de gebruiker zelf nog forceren dat er meteen naar de database wordt geschreven.

Het verwerken met Entity Framework is met dit batch systeem sneller dan bij het meteen schrijven naar de database en langzamer dan aan het einde schrijven naar de database. Het probleem met het vollopen van het intern geheugen is echter nog steeds niet opgelost.

Door het toepassen van dit batchsysteem ontstaat er de mogelijkheid voor een extra truckje waarmee het geheugenprobleem is opgelost. Dit is het "rebuilden" van het Entity Framework.

Iedere keer als de eigen SaveChanges methode wordt aangeroepen en deze feitelijk naar de database heeft geschreven wordt dit rebuilden gedaan. Hierbij wordt het Entity Framework helemaal weggegooid en opnieuw opgebouwd. Omdat alle wijzigingen al naar de database zijn geschreven hebben we geen dataverlies in het Entity Framework en door dit rebuilden is al het geheugen dat door deze wijzigingen bezet was, vrijgekomen.

Door het implementeren van dit batchsysteem zie de workflow er in de Artikelverwerker als volgt uit. De applicatie laadt een bestand en maakt van de gegevens één voor één objecten. Deze objecten worden allemaal één voor één naar het Entity Framework geschreven. Bij iedere wijziging van het Entity Framework wordt de eigen SaveChanges methode aangeroepen. Als er een X aantal wijzigingen zijn verricht in het Entity Framework schrijft de eigen SaveChanges methode de wijzigingen naar de database. Nu wordt het Entity Framework weggegooid en opnieuw opgebouwd en gaat de applicatie verder met de volgende records.



Figuur 10-4: EF Batch

Door het toepassen van dit batchsysteem is het gebruik van het Entity Framework redelijk snel en het intern geheugen zal niet vollopen.

Nu zijn alle performance en memory management problemen met betrekking tot het Entity Framework opgelost.

10.4. Opstartroutine

Zoals eerder gezegd worden gegevens in het nieuwe systeem geïmporteerd in de InstallCalc database door middel van een opstartroutine.

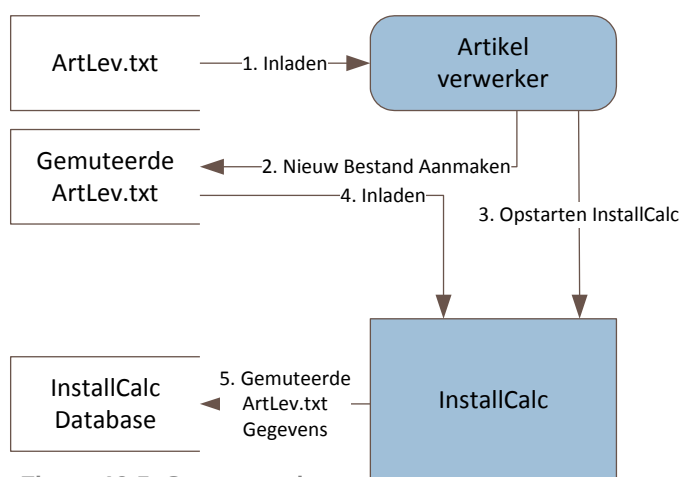
Tijdens het opstarten van InstallCalc kunnen parameters worden meegegeven. Met behulp van deze parameters kan worden aangegeven dat InstallCalc bepaalde bestanden in een bepaalde database moet zetten. Hiervoor hoeft de gebruiker niets te doen na het opstarten van InstallCalc.

Om deze opstartroutine te automatiseren is deze opgenomen in de Artikelverwerkingsapplicatie. De applicatie start dus zelf InstallCalc op en zegt met behulp van parameters dat deze bestanden moet importeren met behulp van InstallCalc's eigen inleesroutines. Omdat InstallCalc het inlezen zelf doet is binnen de Artikelverwerkingsapplicatie geen kennis nodig over de mapping van de InstallCalc database.

Het importeren naar de InstallCalc database gaat nu op de volgende manier.

De Artikelverwerker laadt een ArtLev.txt bestand. De gegevens in dit bestand worden gemuteerd op basis van de Business Rules en opgeslagen in een nieuw ArtLev.txt bestand speciaal voor InstallCalc. Nu start de Artikelverwerker InstallCalc op en zegt dat deze het nieuwe ArtLev.txt bestand moet importeren. InstallCalc

gaat nu zelfstandig het bestand importeren in de database. Als InstallCalc klaar is met het importeren laat deze dat weten aan de Artikelverwerker die nu verder kan met zijn workflow.



Figuur 10-5: Opstartroutine

Er is voor het inlezen met behulp van een opstartroutine gekozen omdat er tijdens dit project geen tijd is om de hele mapping van de InstallCalc database in kaart te brengen. Het gebruik van een opstartroutine is een snelle en bewezen manier om de gegevens op de juiste plaats in de database te zetten. Het nadeel van deze methode is wel dat alleen de gegevens die gedefinieerd zijn in de opstartroutine kunnen worden verwerkt. Tevens is er weinig feedback van de opstartroutine naar de artikelverwerker waardoor er problemen kunnen ontstaan als er iets fout gaat tijdens het importeren.

11. Rapporten

In het huidige systeem worden gegevens geïmporteerd naar de verschillende databases. Deze gegevens worden nooit gecontroleerd op correctheid, tenzij werknemers van andere afdelingen fouten ontdekken en deze melden. Omdat gegevens die worden geïmporteerd informatie bevatten over artikelen en hun prijzen en de verkoopprijzen ook op basis van deze gegevens worden bepaald, is het belangrijk om te weten of de gegevens die worden geïmporteerd ook correct zijn. Incorrecte gegevens kunnen zorgen dat artikelen te goedkoop worden verkocht en dat Strukton Worksphere hier dus verlies op maakt. Tevens kan het zo zijn dat artikelen te duur worden aangeboden en dat Strukton Worksphere hierdoor opdrachten verliest aan zijn concurrenten.

De rapporten die Strukton Worksphere gegenereerd wil hebben zijn niet alleen voor de afdeling Gegevensbeheer, maar ook voor bijvoorbeeld de afdeling Inkoop. Deze rapporten bevatten o.a. informatie over wijzigingen in producten, prijzen, kortingen, etc..

Het aanmaken van rapport records gebeurt “Event Driven” tijdens het importeren. Zodra er bijvoorbeeld een artikel is, waarbij een prijswijziging is geconstateerd en deze aan een rapport moet worden toegevoegd, zal deze een Event opgooien. Op basis van dit Event zal de reden dat dit artikel moet worden toegevoegd aan een rapport, samen met de relevante data van het artikel, als record worden toegevoegd aan een tijdelijk rapportbestand. Zodra de methode CreateReports wordt aangeroepen, zullen alle tijdelijke rapportbestanden naar de uiteindelijke rapport folder verplaatst worden.

Alle rapport bestandsnamen zijn uniek op basis van een toegevoegde timestamp.

Voor meer informatie over de rapporten die gegenereerd worden zie bijlage XIII.

12. Logboek

Hoewel het nieuwe systeem uitgebreid getest is, zijn er altijd onvoorziene omstandigheden die voor kunnen komen tijdens het draaien van een softwareapplicatie. Als een applicatie bijvoorbeeld toch een keer crasht, is het natuurlijk belangrijk om te weten wat de applicatie al had gedaan voordat deze crashte, waar deze mee bezig was toen de applicatie crashte en misschien wat een mogelijk oorzaak was van de crash. Alle verwerkte gegevens voor de crash zijn correct in de databases geïmporteerd. Door het verhelpen van de oorzaak van de crash en door de verwerking opnieuw te draaien worden alle gegevens correct geïmporteerd.

Om de gebruiker of applicatiebeheerder hierover te informeren worden er door de applicaties logboeken bijgehouden. Alle applicaties maken iedere keer als deze gedraaid worden een nieuw logboek aan. Deze logboeken worden allemaal opgeslagen met behulp van een timestamp.

De klasse die verantwoordelijk is voor de logboeken is een thread-safe singleton, waardoor deze door alle andere klassen kan worden aangeroepen zonder dat deze concurrency problemen ondervindt. Iedere keer als er iets aan het logboek moet worden toegevoegd wordt dit meteen naar een logboekbestand geschreven. Hierdoor is het logboekbestand up-to-date als de applicatie onverhoeds crasht.



13. Configuratie

Om de eindgebruikers meer flexibiliteit te geven voor hun configuratie van het nieuwe softwaresysteem is ervoor gezorgd dat deze een aantal instellingen van het systeem kunnen veranderen nadat dit is geïnstalleerd.

XML

De meeste instellingen kunnen de eindgebruikers aanpassen met behulp van een XML-configuratiebestand. In dit bestand staat beschreven waar de applicaties de gegevensbestanden vandaan moeten halen en de gegevens voor de opstartroutine van InstallCalc. Tevens zijn de Business Rules Sets gedefinieerd in het configuratiebestand. Dit zijn verzamelingen van Business Rules die toepasbaar zijn op dezelfde data.

Omdat het configuratiebestand een XML-bestand is, is deze makkelijk met behulp van een texteditor aan te passen door de eindgebruiker.

Voor meer informatie over de inhoud en opbouw van het XML-configuratiebestand zie bijlage XIV.

App.Config

Naast het XML-bestand is er per applicatie in het softwaresysteem een app.config bestand. Met dit bestand kunnen de databases worden ingesteld die de betreffende applicaties gebruiken.

Voor meer informatie over de inhoud en opbouw de app.config-bestanden die bijlage XV.

14. Text Encoding

14.1. Speciale Karakters

Bij het huidige systeem zijn er problemen met speciale karakters in bestanden die worden aangeleverd. Hierin kunnen bijvoorbeeld karakters zitten als '½', 'ï', 'µ', etc. die niet correct worden weggeschreven naar de databases.

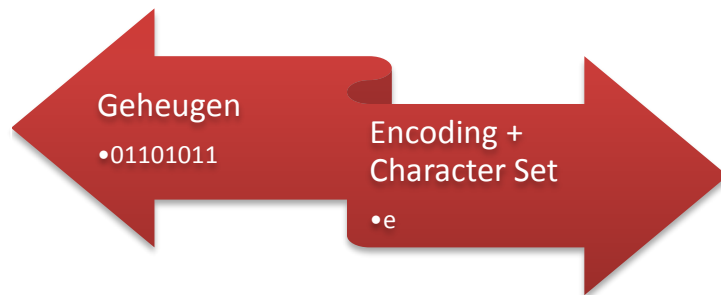
De oorzaak van dit probleem ligt bij de verschillen in Text Encoding tussen de bestanden, de programma's die de bestanden verwerken en de databases waar de gegevens ingezet worden.

14.2. Wat is Text Encoding?

Om te beginnen moet men begrijpen dat er eigenlijk geen tekst bestaat in een computer. Tekst is namelijk de combinatie van binaire data, Text Encoding en een Character Set. Samen vormen deze alle karakters binnen de computer.

Een Character Set vertelt ons welke karakters er beschikbaar zijn en Text Encoding geeft een manier om deze indexen efficiënt te representeren in binaire vorm. Text

Encoding zegt dus dat bijvoorbeeld bij de binaire data 01101011 index 107 hoort. In de betreffende Character Set hoort bij index karakter 'e'.



Figuur 14-1: Text Encoding

Als nou een bestand wordt geopend door twee verschillende applicaties die een verschillende Text Encoding gebruiken kan het zijn dat de binaire data 01101011 zich bij het ene programma vertaalt naar het karakter 'e' en bij het andere programma naar het karakter 'b'. Ook zou het kunnen dat er bij de binaire data helemaal geen karakter bekend is bij een andere Text Encoding.

Daarnaast kan het ook nog zo zijn dat binaire data waar één karakter bij hoort, in een andere encoding geïnterpreteerd kan worden als de codering van meerdere karakters. Dit kan onder andere problemen geven bij het schrijven van deze tekst naar een database waar de lengte van de tekst gelimiteerd is.

14.3. Waarom verschillende Text Encodings?

Vroeger was intern geheugen van computers zeer gelimiteerd. Daarom werd er gekozen voor het toewijzen van 8 bits aan karakters voor tekst. Met deze 8 bits kunnen 265 verschillende karakters worden gedefinieerd. In deze 8 bits zouden niet alleen Latijnse karakters moeten komen, maar ook karakters van bijvoorbeeld de Russische,

Japanse en Chinese taal. Om toch alle karakters in deze 8 bits op te kunnen slaan wordt gebruik gemaakt van Text Encoding en Character Sets.

Er zijn niet alleen verschillen in Character Sets en Text Encoding tussen verschillende talen. Zo zijn er heel veel verschillende Character Sets en Text Encodings die gebruik maken van de Latijnse karakters. Dit komt doordat veel grote computerfabrikanten zoals bijvoorbeeld IBM hun eigen Character Sets en Text Encodings hebben gedefinieerd.

14.4. Text Encoding binnen Strukton Worksphere

Zoals gezegd komen fouten in Text Encoding door het gebruik van verschillende Encodings op een tekst. Bij de workflow voor het verwerken van artikelgegevens worden de volgende Text Encodings gebruikt.



Figuur 14-2: Text Encoding huidige systeem

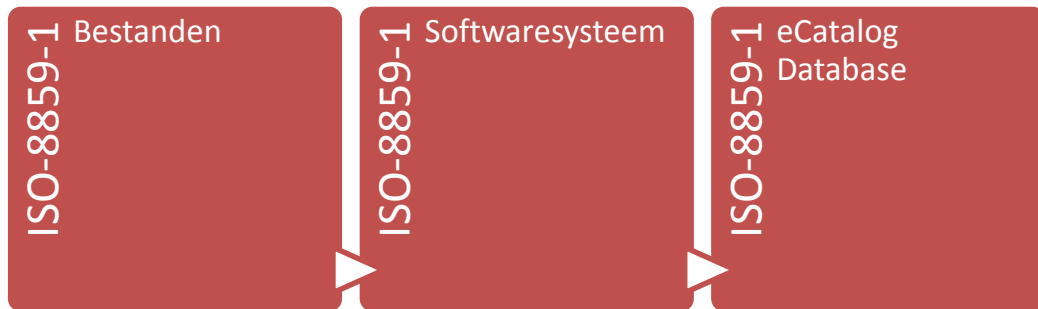
De bestanden die aangeleverd worden en de database waar deze in worden geschreven hebben dezelfde Text Encoding, namelijk ISO-8859-1. Het overzetten van gegevens uit de bestanden naar de database zou dus geen problemen op moeten leveren. Ware het niet dat de gegevens hiertussen een aantal keer in Text Encoding veranderen. Door deze tussentijdse veranderingen wordt de tekst onherstelbaar hervormd zodat de tekst die uiteindelijk in de database wordt gezet niet meer hetzelfde is als de tekst in de bestanden. De boosdoener is hier niet het omzetten van de ISO-8859-1 encoding naar de verschillende UTF encodings, deze vertaling wordt correct uitgevoerd. Het teruggaan naar ISO-8859-1 vanuit UTF is de oorzaak van de problemen.

Binnen het huidige systeem wordt dit probleem opgelost door zowel in Access als in FocusETL speciale karakters te verwijderen.

Dit heeft twee problemen. Je weet namelijk niet welke karakters goed gemuteerd worden tussen de verschillende Text Encodings. Hierdoor kan het altijd zo zijn dat er een karakter tussen zit dat niet verwijderd wordt zodat er toch fouten optreden. Daarnaast kan de tekst slecht leesbaar worden doordat er teveel karakters verwijderd zijn.

14.5. Oplossing

In het nieuwe systeem is ervoor gekozen om het probleem van speciale karakters op te lossen door de Text Encoding door de workflows heen te standaardiseren. Hierbij is de ISO-8859-1 Encoding, die door zowel de bestanden als door de databases gebruikt wordt, gehandhaafd in het nieuwe softwaresysteem dat de gegevens verwerkt.



Figuur 14-3: Text Encoding nieuwe systeem

Alle speciale karakters die in de bestanden kunnen staan worden nu correct overgezet naar de databases.

InstallCalc heeft dezelfde problemen met Text Encoding als eCatalog. InstallCalc heeft net als eCatalog een database met de ISO-8859-1



15. Testen

15.1. Belangrijke aspecten om te testen

Voor dit project is het belangrijk dat de juiste data zich op de juiste plaats bevinden. Daarom wordt er eerst getest of de gegevens die uit de bestanden komen correct vertaald worden naar C# objecten.

Als deze data correct zijn, zullen deze gemuteerd worden en naar eCatalog en InstallCalc geschreven worden. Het is zeer belangrijk dat de gegevens in deze twee applicaties correct zijn. Met name de gegevens met betrekking tot het bepalen van prijzen mogen geen fouten bevatten.

Als laatste moet er worden getest of de gegevens die naar de verschillende rapporten worden geschreven correct zijn en of deze überhaupt wel in het betreffende rapport gezet hadden mogen worden.

15.2. Manieren van testen

Voor het testen van de applicaties is gekozen voor het testen van verschillende aspecten op verschillende manieren.

15.2.1. Situatie Tests

Er worden een hoop aspecten getest met behulp van situatie tests. Dit zijn tests waarbij het softwaresysteem geconfigureerd wordt om een bepaald aspect te testen.

Op deze manier kunnen zaken getest worden als:

- Wat gebeurt er als een applicatie geen connectie kan maken met een database?
- Wat gebeurt er als een connectie met een database wegvalt tijdens het importeren?
- Wat gebeurt er al een bestand dat moet worden ingeladen ontbreekt?

15.2.2. Unit Tests

Door middel van Unit Tests worden de inleesroutines van de applicaties getest. Indien deze tests succesvol zijn, komen de gegevens die de applicaties hebben ingelezen overeen met de gegevens die in de bestanden staan.

Voor het controleren van de rapporten zal tevens gebruik worden gemaakt van Unit Tests. Hierbij zal getest worden of de juiste records aan de juiste rapporten worden toegevoegd.

15.2.3. Databases vergelijken

Om te controleren of de mutaties en het importeren van de gegevens op de juiste manier gebeurt, zullen er gegevens via de oude methode en met de nieuwe applicaties worden ingelezen naar testdatabases. Deze testdatabases zijn kopieën van de productie database. Daarna zullen deze gegevens worden vergeleken met behulp van “Schema Compare” in Visuale Studio. Hiermee kunnen twee databases naast elkaar worden gezet en worden alle verschillen weergegeven.

In eerste instantie zal het vergelijken tussen de oude en de nieuwe methode van gegevensverwerking gedaan worden met een set testgegevens. Dit is een kleine set gegevens waarop veel verschillende mutaties zouden moeten worden uitgevoerd. Als deze test succesvol is afgerond zijn de belangrijkste data aspecten al getest. Maar om er zeker van te zijn dat alle data goed verwerkt worden, wordt hierna een test gedaan met een volledige update met echte gegevens die door de afdeling gegevensbeheer in de productie database wordt gezet. Hierna worden dezelfde gegevens in een testdatabase gezet en worden deze twee databases met elkaar vergeleken.

15.3. Testresultaten

Tijdens het testen kwamen er veel kleine foutjes aan het licht. Zo waren er kleine verschillen in de gegevens die met het nieuwe softwaresysteem waren ingelezen en de gegevens die op de oude manier waren ingelezen. Vanwege de grote hoeveelheid gegevens en mogelijke mutaties hierop viel dit te verwachten. Deze fouten waren snel op te lossen. Het is wel zo dat dit soort fouten na dit project nog steeds aanwezig kunnen zijn, maar niet aan het licht komen, doordat de gegevens waarmee tijdens dit project getest is niet alle mogelijke mutaties op de bestanden hebben afgevangen. Dit is ook bijna onmogelijk door de grote hoeveelheid gegevens en mogelijke mutaties hierop.

Omdat voor dit project een bestaande importapplicatie (FocusETL) is gebruikt als uitgangspunt, werden fouten die daarin zitten overgenomen in het nieuwe softwaresysteem. Voor zover deze fouten aan het licht kwamen zijn deze verholpen. Ook waren er fouten gevonden in de bestaande databasestructuur. Er waren bijvoorbeeld kolommen in de database die niet groot genoeg waren om de informatie te bevatten die daarvoor bestemd was.

Tijdens het testen bleek ook dat niet alle omliggende informatiesystemen van Strukton Worksphere, die gebruik maken van de gegevens die worden ingelezen, overweg konden met de speciale karakters die nu worden doorgelaten. Naar alle waarschijnlijkheid kan dit worden opgelost met een aantal kleine wijzigingen in die omliggende systemen. Om dit te doen zullen deze systemen wel uitgebreid getest moeten worden. Omdat dit buiten de scope van dit project ligt, is er besloten om van het meenemen van de speciale karakters een instelling te maken in het XML-configuratiebestand. Hierin wordt deze instelling bij oplevering uitgezet. Indien in de toekomst blijkt dat alle systemen hiermee om kunnen gaan kan deze instelling altijd nog aangezet worden.

Een groot probleem tijdens het testen was de performance van het nieuwe software systeem. Deze viel in eerste instantie zwaar tegen. Het systeem was toen geïnstalleerd op de InstallCalc server en communiceerde met de Acceptatie eCatalog database. Het is nodig om het nieuwe softwaresysteem op de server van InstallCalc te installeren om automatisch naar InstallCalc te kunnen importeren.

Het verwerken van een ArtLev-bestand met nog geen 20.000 records duurde in deze configuratie meer dan 6 uur. Dit is onacceptabel aangezien dit niet het enige bestand is dat verwerkt wordt, en bovendien zijn de bestanden meestal vele malen groter dan dit bestand. Hierdoor zou een update van alle gegevens niet in één nacht kunnen gebeuren. De reden dat de verwerking zo lang duurde, heeft vooral te maken met de server waarop het softwaresysteem is geïnstalleerd. Deze server was niet snel genoeg om de verwerking binnen een redelijke tijd tot stand te brengen. Om dit probleem op te lossen is ervoor gekozen om een nieuwe virtuele server aan te maken waarop InstallCalc en het softwaresysteem geïnstalleerd werden. Deze server was snel genoeg om een ArtLev-bestand met ongeveer 40.000 records in ongeveer een uur te verwerken.

16. Deployment

De applicaties zullen deployed worden met behulp van een installer. Deze installer zal gemaakt worden met behulp van InstallShield LE voor Visual Studio. Met een installer is het mogelijk om de applicaties op een makkelijke manier op Windows systemen te installeren. Alle applicaties zijn samengevoegd in één installer. Tevens zorgt de installer voor snelkoppelingen naar de applicaties op het bureaublad en het startmenu.

De installer zal tevens een .NET Framework 4.0 web installer bevatten. Hiermee kan dit Framework, dat nodig is voor het draaien van de applicaties, geïnstalleerd worden op computers met een internetconnectie.

Bij de deployment zal een standaard configuratie bestand worden meegeleverd. Indien dit niet (meer) aan de eisen voldoet zal dit bestand na het voltooien van de installatie moeten worden aangepast door de gebruiker.



17. Bedrijfsimpact

De uiteindelijke bedrijfsimpact van het implementeren van een nieuw softwaresysteem is altijd pas te constateren nadat dit een tijdje in gebruik is geweest. Toch kan er al veel gezegd worden over veranderingen die teweeg zijn gebracht naar aanleiding van dit project.

Om te beginnen zullen de medewerkers van de afdeling Gegevensbeheer veel minder lang zelf feitelijk bezig zijn met het verwerken van de artikel- en kortingsgegevens. Hierdoor houden zij tijd over voor andere taken zoals het opsporen en oplossen van fouten die al in de databases aanwezig zijn, verkoopkortingsafspraken- en crediteuren beheer.

Omdat het verwerken van de gegevens zo veel tijd kostte, werden de belangrijkste toeleveranciers altijd wekelijks geüpdate en de overige maandelijks. Nu de belasting voor de medewerkers van de afdeling Gegevensbeheer verminderd is, kunnen alle toeleveranciers wekelijks worden geüpdate. Hierdoor zijn de gegevens actueler en de prijzen die berekend worden naar de klant in een betere verhouding tot de inkooprijzen. Hierdoor is de kans kleiner dat Strukton Worksphere offertes maakt die te goedkoop zijn, waardoor er geen winst gemaakt wordt. Aan de ander kant is de kans ook kleiner dat Strukton Worksphere offertes maakt die zo duur zijn dat de order naar de concurrent gaat. Door dit nieuwe softwaresysteem zijn deze risico's dus verminderd.

Door het nieuwe softwaresysteem is er controle op de gegevens die via de bestanden worden geïmporteerd. Zo kunnen verkoopkortingen nu direct worden afgestemd op de inkoopkortingen. Tevens kunnen fouten in de bestanden makkelijk worden opgespoord zoals bijvoorbeeld prijzen die ineens veel te hoog zijn geworden.

Omdat de Text Encoding nu door alle aspecten van het systeem gelijk blijft, kunnen hier geen fouten meer door ontstaan en hoeft hier dus niet meer op te worden gecontroleerd. Ook zijn de omschrijvingen van artikelen nu veel duidelijker omdat speciale karakters hier nu in opgenomen zijn.

Samengevat zijn de veranderingen binnen het bedrijf dus:

- Tijdswinst
- Actuelere gegevens
- Betere verkoopprijzen
- Rapporten over gegevens
- Verkoopkortingen sluiten aan op inkoopkortingen
- Duidelijkere gegevens door leestekens

18. Conclusies en aanbevelingen

Tijdens dit project zijn verschillende aspecten onderzocht om aan alle doelstellingen te voldoen. Zo is het nieuwe systeem modulair opgebouwd en is er gebruik gemaakt van een Rule Engine. Beide oplossingen zorgen voor grote flexibiliteit van het systeem waardoor toekomstige aanpassingen makkelijk door te voeren zijn.

Tevens zijn er onderzoeken gedaan naar de beste manieren om de gegevens vanuit de bestanden naar de verschillende databases te schrijven waarbij er een balans is tussen flexibiliteit, functionaliteit, performance en memory management.

Het standaardiseren van de Text Encoding in het nieuwe systeem zorgt voor minder fouten in de databases en betere leesbaarheid van onder andere omschrijvingen van artikelen.

Het resultaat van dit alles is een werkend geautomatiseerd verwerkingssysteem die de medewerkers van de afdeling Gegevensbeheer veel tijd scheelt. Het verwerken van de gegevens kan voor de applicatie meerdere uren duren, de belasting voor de medewerkers zelf is echter maximaal een uur geworden. Hierdoor kunnen zij de gegevens vaker bijwerken en zich meer bezig houden met hun overige taken. Tevens is het mogelijk alle ingelezen gegevens makkelijk te controleren door middel van rapporten.

Toch blijven er na dit project punten voor verbetering. Zo zou de opstartroutine van InstallCalc vervangen moeten worden door een directe communicatie met de database. Ook zou de kennis binnen Strukton Worksphere vergroot moeten worden voor wat betreft de relatie tussen de bestanden die zij verwerkt en de databases waar deze inkomen.

Bij Strukton Worksphere worden bestanden verwerkt met veel gegevens en veel mogelijke mutaties hierop. Hierdoor is het onmogelijk om binnen de tijd van dit project alles goed te testen. Het is daarom noodzakelijk het systeem te blijven testen en te controleren. Vooral omdat alle prijzen waarmee Strukton Worksphere rekent voor zowel inkoop als verkoop, gebaseerd zijn op de gegevens die door het nieuwe softwaresysteem worden verwerkt.



Evaluatie

Het doel van dit project was het realiseren van een geautomatiseerd softwaresysteem voor het verwerken van artikel- en kortingsbestanden.

Om dit doel te bereiken heb ik diverse aspecten onderzocht. Vooral op het gebied van performance, memory management en flexibiliteit. Deze onderzoeken waren nodig voor het maken van een applicatie die de grote hoeveelheid gegevens binnen een beperkte tijd kan verwerken.

Hiernaast heb ik onderzoek moeten doen naar de mapping van de bestanden naar de verschillende databases. Dit heeft veel tijd gekost omdat de kennis hierover niet volledig binnen Strukton Workspace beschikbaar was.

Tijdens dit project heb ik niet alleen een geautomatiseerde variant gemaakt van de bestaande verwerkingsmethode maar heb ik tevens gekeken of hier ook geen fouten in zaten. Deze fouten heb ik voor zover gevonden opgelost.

De grootste problemen tijdens de ontwikkeling van het nieuwe softwaresysteem betroffen de performance en het testen van de grote hoeveelheid gegevens.

De performance van mijn softwaresysteem was heel wisselend. Dit kwam doordat de servers waarop het softwaresysteem draaide niet snel genoeg waren. Dit is uiteindelijk opgelost door mijn softwaresysteem op een andere server te installeren. Het was makkelijker geweest als dit in een vroeger stadium was gebeurd zodat ik eerder een duidelijk beeld had van de werkelijke performance van mijn applicatie.

Voor het testen van mijn applicatie had ik in eerste instantie niet veel tijd uitgetrokken. Gelukkig liep ik vóór op schema zodat ik de applicatie wat uitgebreider kon testen. Dit was wel nodig aangezien er heel veel kleine foutjes in de gegevens van de database in de testomgeving zaten.

Uiteindelijk heeft dit project geleid tot een werkend geautomatiseerd softwaresysteem. Dit softwaresysteem werd voor afronding van het project al in de productie omgeving gebruikt.



Literatuurlijst

Boeken

Laan, G, **Softwareontwikkeling met UML en JAVA**, 1^e druk, Pearson Education Benelux, Amsterdam, 2007

Websites

CareerRide

Pagina over het V-Model.

<http://careerride.com/testing-v-model.aspx>

Christophe Geers' Blog

Pagina over de Windows Workflow Rule Engine.

<http://cgeers.com/2008/01/26/using-rules-outside-of-a-workflow/>

Code Project

Pagina over de Dynamic System Development Method.

<http://www.codeproject.com/Articles/5097/What-Is-DSDM>

Pagina over het Entity Framework.

<http://www.codeproject.com/Articles/363040/An-Introduction-to-Entity-Framework-for-Absolute-B>

Pagina over het gebruik van OleDb voor het importeren van tekstbestanden.

<http://www.codeproject.com/Articles/27802/Using-OleDb-to-Import-Text-Files-tab-CSV-custom>

DroolsDotNet

Website over de .Net implementatie van Drools.

<http://droolsdotnet.codehaus.org/>

ETIM Wiki Nederland

De Wikipedia website van ETIM. Beschrijft o.a. PAB2- en ICC-bestanden.

<http://wiki.etim.nl/wiki/Hoofdpagina>

InRule Technology

Officiële website van de fabrikant van de InRule Rule Engine.

<http://www.inrule.com/default.aspx>

ISTQB bexamcertification

Pagina over het Watervalmodel.

<http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/>

Pagina over het V-Model.

<http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/>



Joel on Software – The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)

Pagina over Text Encoding.

<http://www.joelonsoftware.com/articles/Unicode.html>

Microsoft Developers Network

Pagina over de Windows Workflow Foundation Rules Engine.

<http://msdn.microsoft.com/en-us/library/aa480193.aspx>

Pagina over het Entity Framework.

<http://msdn.microsoft.com/en-us/data/ef.aspx>

Pagina over Threads.

<http://msdn.microsoft.com/en-us/library/system.threading.thread.aspx>

Richard Banks – Agile and .NET

Pagina over de Windows Workflow Rule Engine.

<http://www.richard-banks.org/2007/08/how-to-use-windows-workflow-rules.html>

Select Business Solutions

Pagina over het Watervalmodel.

<http://www.selectbs.com/analysis-and-design/what-is-the-waterfall-model>

Wikipedia

Pagina over de Dynamic System Development Method.

http://en.wikipedia.org/wiki/Dynamic_systems_development_method

Pagina over het V-Model.

<http://en.wikipedia.org/wiki/V-Model>

Pagina over het Watervalmodel.

<http://nl.wikipedia.org/wiki/Watervalmethode>



Strukton
Worksphere

Bijlagen

I. Plan van Aanpak

Titelpagina

Document

Titel	Automatisering Artikelbeheer
Ondertitel	Plan van Aanpak
Datum aangemaakt	19-11-2012
Datum laatste wijziging	02-03-2013
Versie	7

Afstudeerder

Naam	Sjors van Osch
Studentnummer	2152631

Afstudeerbedrijf

Naam	Strukton Worksphere
Afdeling	SSC Gegevensbeheer
Adres	Science Park Eindhoven 5206
Postcode	5692 EG
Plaats	Son



Achtergronden

Strukton WorkspHERE

Strukton WorkspHERE creëert, samen met opdrachtgevers, een optimale omgeving waarin mensen prettig kunnen verblijven. Dit doet Strukton WorkspHERE door te zorgen voor betrouwbare, duurzame en efficiënte technische voorzieningen.

Strukton WorkspHERE neemt opdrachtgevers de zorg voor het gebouw uit handen, door verantwoordelijkheid te nemen voor het ontwerp, de realisatie en het beheer en onderhoud. Hierdoor kunnen klanten zich concentreren op het primaire proces en zijn zij verzekerd van een veilige, comfortabele en duurzame werk- en verblijfsomgeving.

Strukton WorkspHERE is onderdeel van Strukton, full service provider voor infrastructuur- en accommodatieoplossingen.

Betrokken afdelingen

Bij dit project zijn de afdelingen SSC Gegevensbeheer en de afdeling ICT Applicatiebeheer betrokken. Het eindproduct van dit project is bedoeld voor werknemers van de afdeling SSC Gegevensbeheer. Tevens worden er in samenwerking met deze afdeling de functionele eisen voor het project geformuleerd.

Ook de afdeling ICT Applicatiebeheer is betrokken bij dit project. Deze geeft technische ondersteuning tijdens het ontwikkelen van het eindproduct.

Eerdere Projecten

Dit project dient ter (gedeeltelijke-) vervanging van een applicatie die momenteel draait in MS Access. Deze applicatie gemaakt door een van de eindgebruikers van dit project (Ton van den Boomen).

Bij de huidige applicatie in MS Access liep men aan tegen het probleem dat men niet alle data in een keer kon verwerken, het proces ondoorzichtig en foutgevoelig was, en dat er nog steeds veel stappen handmatig verricht moeten worden. Dit maak de processen omtrent het beheren van artikelen inefficiënt en tijdrovend. Vanwege deze limitaties van Microsoft Access is er voor gekozen dit project te starten.

Voor een gedeelte van dit project is al eerder door een bedrijf een opzet van een Functioneel Ontwerp gemaakt.



Projectopdracht

De projectopdracht is het maken van een applicatie waarmee inkoopgegevens efficiënt en controleerbaar verwerkt kunnen worden. Deze applicatie kan PAB2 bestanden openen en deze omzetten in CSV- en TXT-bestanden welke gelezen kunnen worden door databases van InstallCalc en Ecatalog.

Ook moet er een oplossing worden gevonden voor het omzetten van ICC-bestanden naar CSV- en TXT-bestanden.

Van wijzigingen in zowel PAB2- als ICC-bestanden moeten rapporten kunnen worden gemaakt. Ook moet er de mogelijkheid zijn om een rapport te maken van uitlopende en vervallen artikelen.

Tevens is er de wens om makkelijk gegevens uit verschillende bestanden en databases te kunnen vergelijken en hiervan rapporten te kunnen genereren. Hieronder valt het vergelijken tussen:

- Database Focus met database InstallCalc
- PAB2-bestand met InstallCalc- en Ecatalog database
- ICC-bestand met de Focus database



Om een duidelijk overzicht te geven van de verschillende aspecten van de projectopdracht worden deze geformuleerd in doelstellingen.

Doelstellingen

Voor het verduidelijken van de opdracht zijn de doelstellingen gedefinieerd op basis van de opdrachtomschrijving. Om binnen deze doelstellingen de prioriteiten te bepalen is gebruik gemaakt van de MoSCoW methode.

Tabel 0-1: Doelstellingen MoSCoW

Requirements	Prioriteit
Een eindgebruiker kan data uit PAB2-artikelbestanden naar de databases van eCatalog en InstallCalc importeren.	Must Have
Een eindgebruiker kan data uit Mod-bestanden toevoegen aan de databases van eCatalog.	Must Have
Een eindgebruiker kan data uit ICC-kortingsbestanden importeren in de database van eCatalog.	Should Have
Een eindgebruiker moet inkoopkortingen in eCatalog met InstallCalc kunnen vergelijken.	Could Have
Een eindgebruiker moet artikelen in InstallCalc, eCatalog en een PAB2-bestand met elkaar kunnen vergelijken.	Could Have
Een eindgebruiker moet verkoopkortingen met inkoopkortingen kunnen vergelijken.	Could Have
Een eindgebruiker kan rapporten genereren met gewijzigde artikelen.	Could Have
Een eindgebruiker kan rapporten genereren met artikelen waarvan een prijswijziging heeft plaatsgevonden.	Must Have
Een eindgebruiker kan rapporten genereren met artikelen waarvan de stamgegevens zijn veranderd.	Must Have
Een eindgebruiker kan rapporten genereren met kortingsgroepen die vervallen of veranderd zijn.	Should Have
Een eindgebruiker kan rapporten genereren met artikelen van kortingsgroepen die vervallen of veranderd zijn.	Should Have
Een eindgebruiker kan rapporten genereren voor de afdeling inkoop	Should Have



met kortingsgroepen waarvan de kortingen of condities vervallen of veranderd zijn. Hiervoor moeten Business Rules kunnen worden gedefinieerd. Bijvoorbeeld alleen de artikelen waarbij het prijsverschil hoger is dan een bepaald percentage.	
Een eindgebruiker kan rapporten genereren met artikelen waarvan de gegevens in de verschillende databases en bestanden niet overeen komen.	Could Have
Een eindgebruiker kan rapporten genereren met dossiers waarvan de inkoop- of verkoopkorting niet klopt.	Could Have
Een eindgebruiker moet rapporten kunnen genereren van verschillen in verkoopkorting tussen Focus en InstallCalc.	Could Have
De Business Rules die de applicaties gebruiken moeten gewijzigd kunnen worden.	Must Have
De standaardkortingen in de InstallCalc database moeten naar de eCatalog database geschreven kunnen worden.	Should Have
De belasting voor een medewerker van de afdeling gegevensbeheer voor het verwerken van artikel- en kortingsgegevens mag maximaal één uur per dag zijn.	Must Have
Grote mutaties moeten 's avonds zonder menselijke tussenkomst kunnen worden uitgevoerd om de belasting op de databases te minimaliseren.	Must Have
Het nieuwe systeem moet flexibel zijn voor toekomstige verandering in de Business Rules en het aanpassen en toevoegen van functionaliteit.	Should Have

Voor Strukton Worksphere is het voornaamste doel van dit project dat men nieuwe artikelgegevens efficiënter kan verwerken. Dit kost momenteel namelijk erg veel tijd en inzet van FTE's. Tevens is er hierdoor niet de mogelijkheid om alle data in het systeem, met de gewenste frequentie, up-to-date te houden. Hierdoor kan het voorkomen dat artikelen gefactureerd worden aan klanten voor prijzen die niet meer haalbaar zijn. Dit kost Strukton Worksphere dus geld.



Om deze reden is er gekozen om het verwerken van nieuwe gegevens hoger te classificeren dan het vergelijken van data. Het vergelijken van data gebeurt momenteel voornamelijk voor het opsporen van fouten indien bij de afdeling gegevensbeheer een klacht binnenkomt. Tevens worden hiermee wijzigingen in receptartikelen gecontroleerd. In de nieuwe situatie wil Strukton Worksphere echter een actievere houding innemen bij het preventief controleren van gegevens zodat fouten tot een minimum beperkt blijven.

Projectactiviteiten

Vorbereiding

- Maken Plan van Aanpak.
- Opstellen Communicatieplan.

Onderzoek

Er moet worden onderzocht wat de beste manier is om de behandeling van artikelgegevens te automatiseren.

Tijdens dit project zal gekeken worden welke onderdelen er geautomatiseerd kunnen worden en hoe dit het beste gedaan kan worden.

Hierbij zal onder andere gekeken worden naar:

- Projectopdracht
- Software Ontwikkelmethoden
- Software Design Patterns
- Rule Engines
- Bestands-IO
- Database-IO
- Rapportgeneratie
- Bedrijfsimpact

Projectopdracht

Strukton Worksphere heeft aangegeven wat er verwacht wordt van dit project. Hierbij dacht men aan het automatiseren van in ieder geval één bestaande workflow met nog veel handmatig werk.

Tijdens dit project gaat er onderzocht worden of dit wel de beste oplossing voor het probleem is of dat deze ene workflow misschien beter zou kunnen worden opgenomen met meerdere workflows in één volledig geautomatiseerd proces.

Rule Engines

Artikelgegevens zijn onderhevig aan meerdere “Business Rules”. Een voorbeeld hiervan is:

“Als een artikel in kortingsgroep 15 zit mag de netto prijs niet hoger zijn dan €100.”

Tijdens dit project zullen de verschillende “Business Rules” in kaart gebracht worden. Hierna wordt onderzocht wat de beste manier is om deze “Business Rules” te vertalen naar code. Dit zou bijvoorbeeld kunnen door middel van “if statements” maar ook door het gebruik van een Rule Engine.

Er gaat onderzocht worden of er voor C# .Net Rule Engines zijn, welke hiervan het best bij dit project past en of het verstandig is om deze toe te passen.

Bestands-IO

Er wordt tijdens dit project gekeken naar de verschillende mogelijkheden voor het lezen van en schrijven naar bestanden. Tevens zal onderzocht worden of het interessant is om data in bestanden te serialiseren naar objecten.



Database-IO

Er zal gekeken worden naar verschillende methoden voor het communiceren met databases en welke hiervan het beste past bij dit project.

Voorbeelden van deze methoden zijn o.a.:

- ADO .Net
- Entity Framework

Rapportgeneratie

Er zal gekeken worden naar de mogelijkheden voor het genereren van rapporten. Kan hierbij de bestaande Excel-template worden gebruikt of zou hier iets anders voor moeten worden gemaakt? Wat zijn de mogelijkheden binnen C# voor het genereren van rapporten?

Bedrijfsimpact

Er zal worden onderzocht wat de bedrijfsimpact van dit project is.

Denk hierbij aan bijvoorbeeld:

- Wat op de manier van werken binnen Strukton Worksphere?
- Wat zijn de financiële consequenties?
- Indien er door dit project tijdswinst is bij het verwerken van artikelgegevens, wat gebeurt er met deze tijdswinst?

Documentatie

- Er zal een Functioneel ontwerp moeten worden opgesteld op basis van een bestaand Functioneel ontwerp. Dit zal aangepast worden aan de huidige eisen.
Voor dit Functioneel ontwerp worden o.a. de volgende onderdelen gemaakt:
 - Procesbeschrijving
 - Gebruikersomgeving
 - Werkomgeving
 - Activity Diagrams
 - Use Cases
- Er zal een Technische ontwerp worden opgesteld. Hierin worden alle aspecten met betrekking tot de beslissingen die zijn genomen tijdens het ontwerpproces beschreven.
Voor dit Technisch ontwerp worden o.a. de volgende onderdelen gemaakt:
 - Domain Model
 - Sequence Diagrams
- Er zal een handleiding worden gemaakt waarmee medewerkers van de afdeling Gegevensbeheer de applicatie kunnen bedienen.

Programmeren

- Er zal software geprogrammeerd worden die voldoet aan de in het Functioneel- en het Technisch ontwerp beschreven eisen.

Testen

- De software zal getest worden op basis van een test plan. Hierin worden de belangrijkste aspecten van de software beschreven en de manier waarop deze getest gaan worden.

Training

- De primaire eindgebruikers zullen training krijgen om te kunnen werken met de software.

Afsluiting

- Er zal een Scriptie worden geschreven met de belangrijkste bevindingen en aspecten van dit project.
- Op basis van de Scriptie zal er een presentatie gehouden worden op de Fontys Hogeschool te Eindhoven.



Projectgrenzen en randvoorwaarden

Projectgrenzen

- Er zal technische en functionele documentatie worden opgesteld over de eindapplicatie.
- Er zal een softwaresysteem worden opgeleverd die tenminste voldoet aan de vooraf opgesteld eisen.
- Er zal een handleiding worden geschreven voor de bediening van het softwaresysteem.
- De primaire gebruikers van het softwaresysteem zullen worden getraind in het werken met de applicatie.
- Er zal geen verdere technische of functionele support gegeven worden aan gebruikers na oplevering van de eindproducten.
- Er zal na goedkeuring van het functionele ontwerp door de hiervoor verantwoordelijke geen mogelijkheid zijn om extra eisen qua functionaliteit toe te voegen.

Randvoorwaarden

- Alle projectleden en belanghebbenden moeten voldoende beschikbaar zijn voor het project.
- De doelstellingen moeten binnen de vooraf vastgestelde tijd van 75 dagen gerealiseerd kunnen worden.
- Er moet overeenstemming zijn tussen de belanghebbenden en de projectleden over de eisen van het product.
- Er moet voldoende materieel zijn om het de producten van het project te ontwerpen en te realiseren (computer, software, etc.).
- Er moeten gegevensbestanden beschikbaar zijn waarmee de bewerkingen van het softwaresysteem worden uitgevoerd.
- Het moet mogelijk zijn informatie van de databases die betrekking hebben op dit project op te vragen.



Producten

- Plan van Aanpak
- Communicatieplan met begeleidingsdocent
- Functioneel ontwerp
- Technisch ontwerp
- Prototype Software Systeem (Source Code)
- Test plan
- Release Software Systeem
- Handleiding
- Scriptie
- Presentatie

Kwaliteit

Er zullen parameters gedefinieerd worden waarmee de kwaliteit van het eindproduct (softwaresysteem) word beoordeeld. Tevens zullen functionele documenten en applicaties beoordeeld worden door zowel Ton van den Boomen als Peter Blokx.

Alle Technische documenten zullen worden beoordeeld door Peter Blokx en één of meerdere experts op het betreffende gebied van de afdeling ICT.

Voor het uitvoeren van dit project zal gebruik gemaakt worden van tekentechnieken als UML met bijvoorbeeld Sequence Diagrams, Domain Models, Entity Relationship Diagrams etc..

Tevens zal er gebruik gemaakt worden van ten minste de volgende software:

- Microsoft Access
- Microsoft Word
- Microsoft Excel
- Microsoft Visio
- Microsoft Visual Studio
- Microsoft SQL Server Management Studio



Projectorganisatie

Projectgroep

Naam:	Sjors van Osch
Functie:	Projectleider
Adres:	Duinbeek 3
Postcode:	5653PL
Plaats:	Eindhoven
Telefoonnummer:	0627435457
Email:	s.w.vanosch@gmail.com
Beschikbaarheid:	40 uur per week gedurende 75 dagen
Bevoegdheden en Verantwoordelijkheden:	• Maken van duidelijke afspraken (waaronder de bedrijfsbezoeken) met docentbegeleider en bedrijfsbegeleider die vastgelegd worden in het plan van aanpak en communicatieplan. • Uitvoeren van de opdracht conform de afspraken met de bedrijfsbegeleider. • Het regelmatig informeren van de bedrijfsbegeleider en de docentbegeleider over de voortgang van de opdracht. • Planmatig werken aan het verslag en dit regelmatig bespreken met de docentbegeleider. • Initiatief nemen als er zich problemen voordoen met in eerste instantie de bedrijfsbegeleider en indien nodig de docentbegeleider en/of de stage/afstudeercoördinator. • Is eindverantwoordelijk voor alle beslissingen en producten die tijdens het project gemaakt worden.



Strukton
Worksphere

Externe Belanghebbenden / Ondersteuning

Naam:	Ton van den Boomen
Functie:	Projectadviseur, Eindgebruiker
Adres:	Science Park Eindhoven 5206
Postcode:	5692 EG
Plaats:	Son
Telefoonnummer:	040-2825577
Email:	ton.vandenboomen@strukton.com
Bevoegdheden en Verantwoordelijkheden:	• Is inhoudelijk deskundig en begeleidt de afstudeerder om een zo goed mogelijk resultaat te behalen; hij begeleidt de afstudeerder dus ook bij het opstellen van het plan van aanpak. • Helpt mee met het formuleren van doelstellingen. • Controleert of het Plan van Aanpak en het Functionele ontwerp overeen komen met zijn visie voor het eindproduct. • Test of het softwaresysteem voldoet aan de vooraf gestelde eisen.



Naam:	Peter Blokx
Functie:	Bedrijfsbegeleider
Adres:	Science Park Eindhoven 5206
Postcode:	5692 EG
Plaats:	Son
Telefoonnummer:	040-2825533
Email:	peter.blokx@strukton.com
Bevoegdheden:	
Bevoegdheden en Verantwoordelijkheden:	• Draagt zorg voor een adequate werkplek van voldoende kwaliteit, zodat de afstudeerder de opdracht adequaat kan uitvoeren. • Ontvangt de afstudeerder bij aanvang van het afstuderen en introduceert hem in de organisatie. • Zorgt dat de werkzaamheden uitgevoerd worden conform de doelstellingen van het afstuderen. • Ontvangt de docent bij het bedrijfsbezoek en bespreekt met hem en de afstudeerder het sociaal en vakinhoudelijk functioneren. • Overlegt met de afstudeerder en/of de docentbegeleider indien dat nodig is. • Tekent het afstudeerverslag voor gezien. • Adviseert de docentbegeleider t.a.v. de beoordeling van de werkzaamheden en het verslag. • Woont de afstudeervoordracht bij en stelt daarbij vragen over de werkzaamheden, het verslag en/of de presentatie.



Strukton
Worksphere

Naam:	Huub van Thienen
Functie:	Docentbegeleider
Adres:	Rachelsmolen 1
Postcode:	5612 MA
Plaats:	Eindhoven
Telefoonnummer:	08850 73547
Email:	huub.vanthienen@fontys.nl
Verantwoordelijkheden:	• Bezoekt tijdens de afstudeerperiode het afstudeerbedrijf en bespreekt de voortgang tot dan toe aan de hand van de ingeleverde documenten; • Begeleidt de afstudeerder procesmatig; • Begeleidt de afstudeerder bij het schrijven van het afstudeerverslag en bij de voorbereiding van de afstudeervoordracht; • Is verantwoordelijk –na overleg met de bedrijfsbegeleider- voor de beoordeling van de werkzaamheden en het afstudeerverslag; • Geeft feedback aan de stage/afstudeercoördinator over het bedrijf, de opdracht, de begeleiding etc. • Woont de afstudeervoordracht bij en stelt daarbij vragen over de werkzaamheden, het verslag en/of de presentatie; • Beoordeelt mede de eindpresentatie.

Voor verdere technische ondersteuning zijn er diverse medewerkers van de ICT ontwikkeling afdeling beschikbaar.

Tijdsverantwoording

Tijdsverantwoording wordt aan Strukton Worksphere gedaan door middel van een intern software systeem genaamd KEUR.

Aan Fontys zal de tijdsverantwoording gerapporteerd worden door middel van een tijdsverantwoordingsformulier.



Planning

Onderdeel	Afhankelijkheid	Datum
Communicatieplan opleveren	-	07-12-2012
Plan van Aanpak opleveren	-	07-12-2012
Onderzoeken nodig voor Functioneel Ontwerp afgerond	Goedkeuring Plan van Aanpak	21-12-2012
Functioneel Ontwerp opleveren	Onderzoeken nodig voor Functioneel Ontwerp afgerond	11-01-2013
Onderzoeken nodig voor Technisch Ontwerp afgerond	Goedkeuring Functioneel Ontwerp	25-01-2013
Technisch Ontwerp opleveren	Onderzoeken nodig voor Technisch Ontwerp afgerond	08-02-2013
Prototype Applicatie opleveren	Goedkeuring Technisch Ontwerp	08-03-2013
Release Applicatie + Source Code opleveren	Goedkeuring Prototype Applicatie	15-03-2013
Concept scriptie opleveren	Goedkeuring al het bovenstaande	18-03-2013
Handleiding opleveren	Goedkeuring Prototype Applicatie	20-03-2013
Training geven aan eindgebruikers	Goedkeuring Handleiding	Een dag tussen 20-03-2013 en 22-03-2013
Scriptie opleveren	Goedkeuring al het bovenstaande	25-03-2013
Presentatie geven	Goedkeuring al het bovenstaande	Een dag tussen 15-04-2013 en 17-04-2013



Kosten en Baten

Kosten

- Projectleden a de stagevergoeding per maand maal vier maanden.
- Tijd van belanghebbenden voor begeleiding van project a hun uurtarief.
- Kosten van gebruik apparatuur voor ontwerpen en ontwikkelen van het softwaresysteem (Laptop etc.).

Baten

- Tijdsbesparing van minimal 3 uur per dag voor een medewerker gegevensbeheer.
- Mogelijkheid tot hogere frequentie van het actualiseren van artikelgegevens met als gevolg minder prijsafwijkingen in facturen en minder belasting voor de crediteurenadministratie.
- Minder kans op verliezen als gevolg van gebruik van foutieve artikelprijzen bij het maken van offertes in InstallCalc.

Risico's

- Onvoldoende tijd om alle doelstellingen van het project te realiseren.
- Onvoldoende beschikbaarheid van belanghebbenden.
- Veranderingen in het PAB2 formaat of het vervangen hiervan door bijvoorbeeld PAB3.
- Veranderingen in het ICC formaat.
- Veranderingen in de structuur van de InstallCalc, Ecatalog of Focus database.



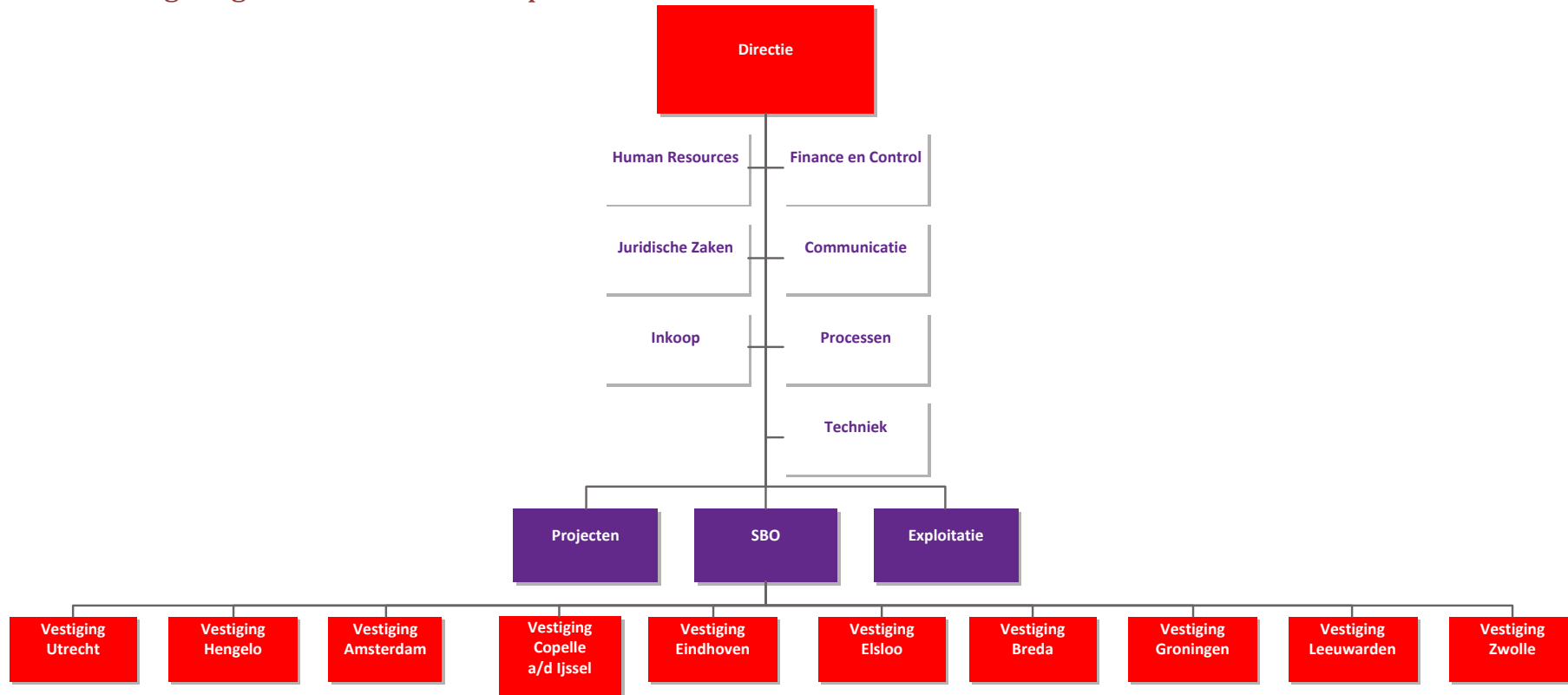
Communicatieplan

	Datum	e-mail/post/postvak
Inleveren PvA en communicatieplan i.o.m. begeleidend docent	09-12-2012	mail
Eerste bedrijfsbezoek	Uiterlijk 21-12-2012	--
Bespreken plan van aanpak en communicatieplan	Conform afspraak	mail
Inleveren afstudeerverslag concept	Uiterlijk 17-03-2013	mail
Bespreken afstudeerverslag concept	Uiterlijk 22-03-2013	mail
Tweede bedrijfsbezoek/tevens beoordelings-gesprek	i.o.m. begelei-dend docent en bedrijfsmentor	Volgens afspraak
Inleveren definitief afstudeerverslag zowel digitaal op CD als in viervoud op papier	25-03-2013	Stage-afstudeerbureau kamer 4.29
Afstudeervoordrachten april 2013	15-04-2013 t/m 17-04-2013	Bekendmaking via uitnodiging



Strukton
Worksphere

II. Organogram Strukton Worksphere

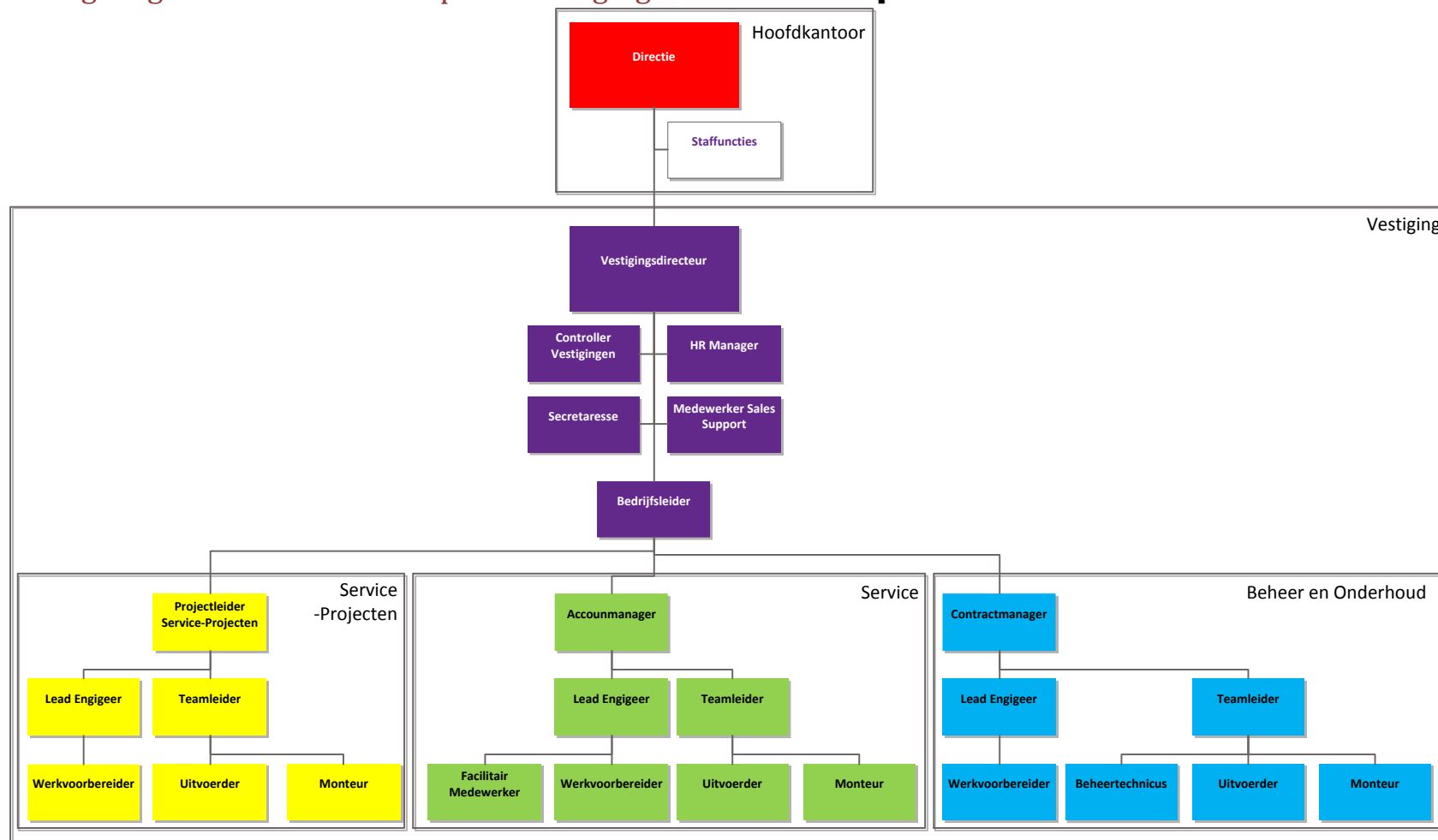




Strukton

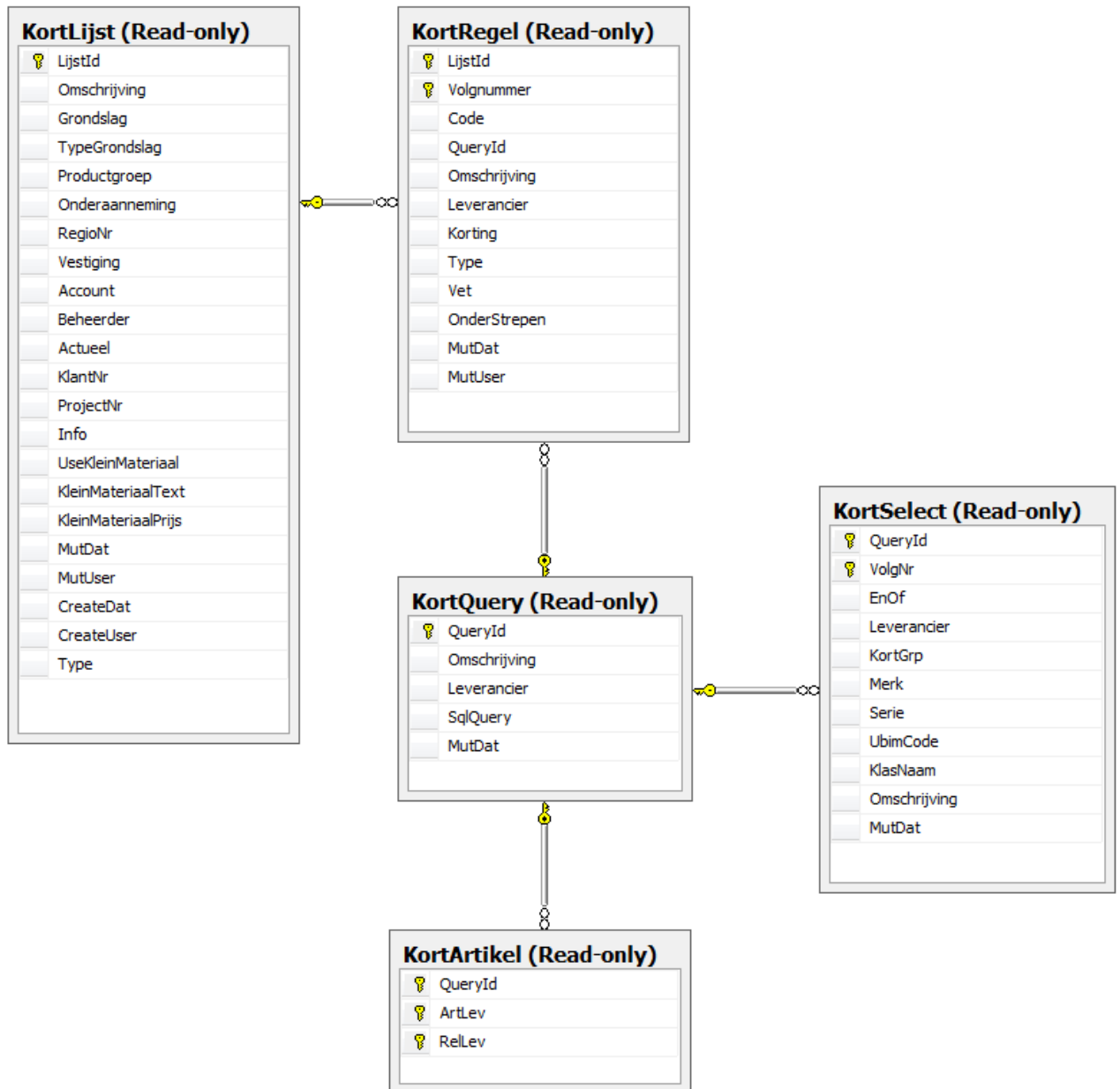
WorkspHERE

III. Organogram Strukton WorkspHERE Vestiging



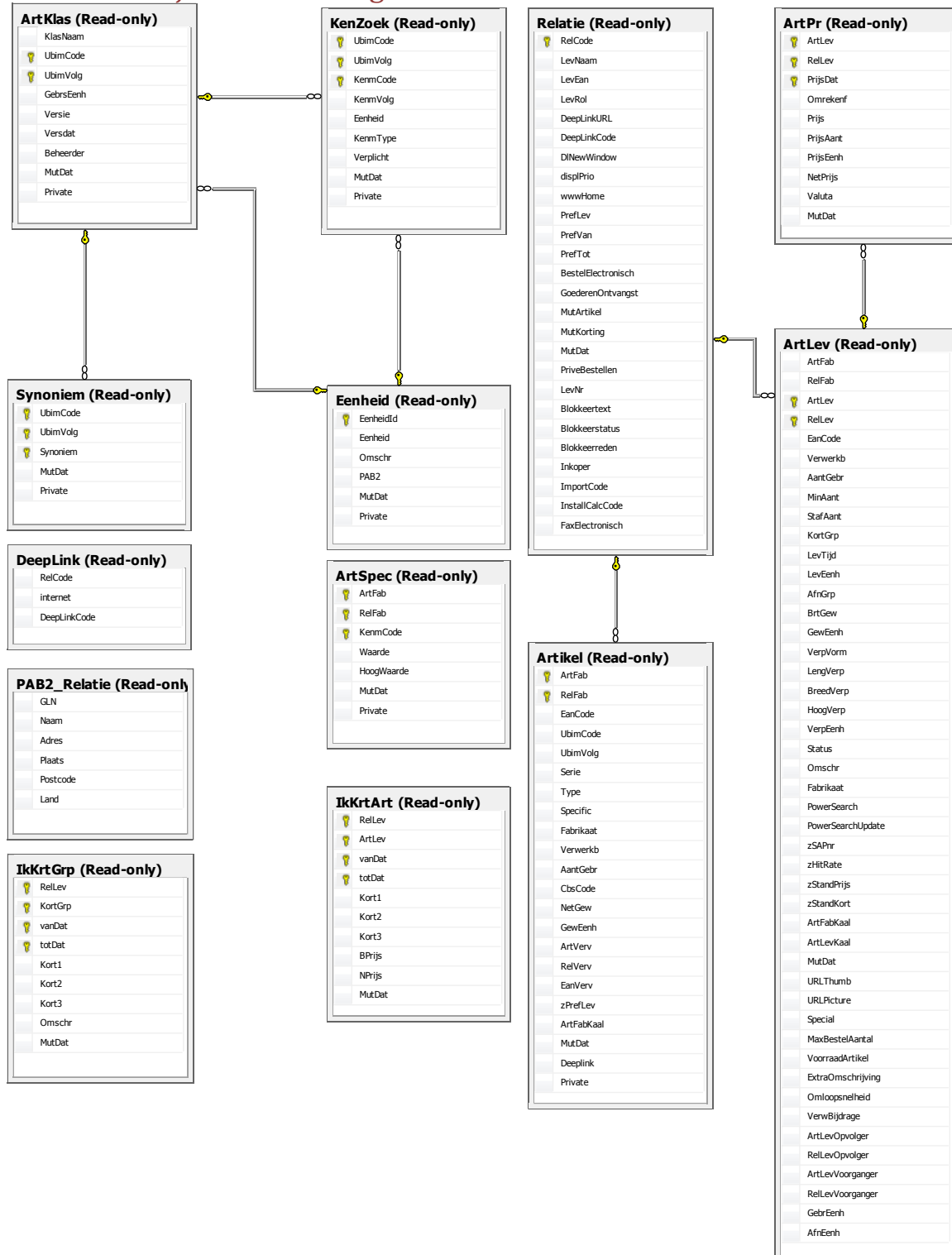


IV. Gedeeltelijk ERD eCustomer



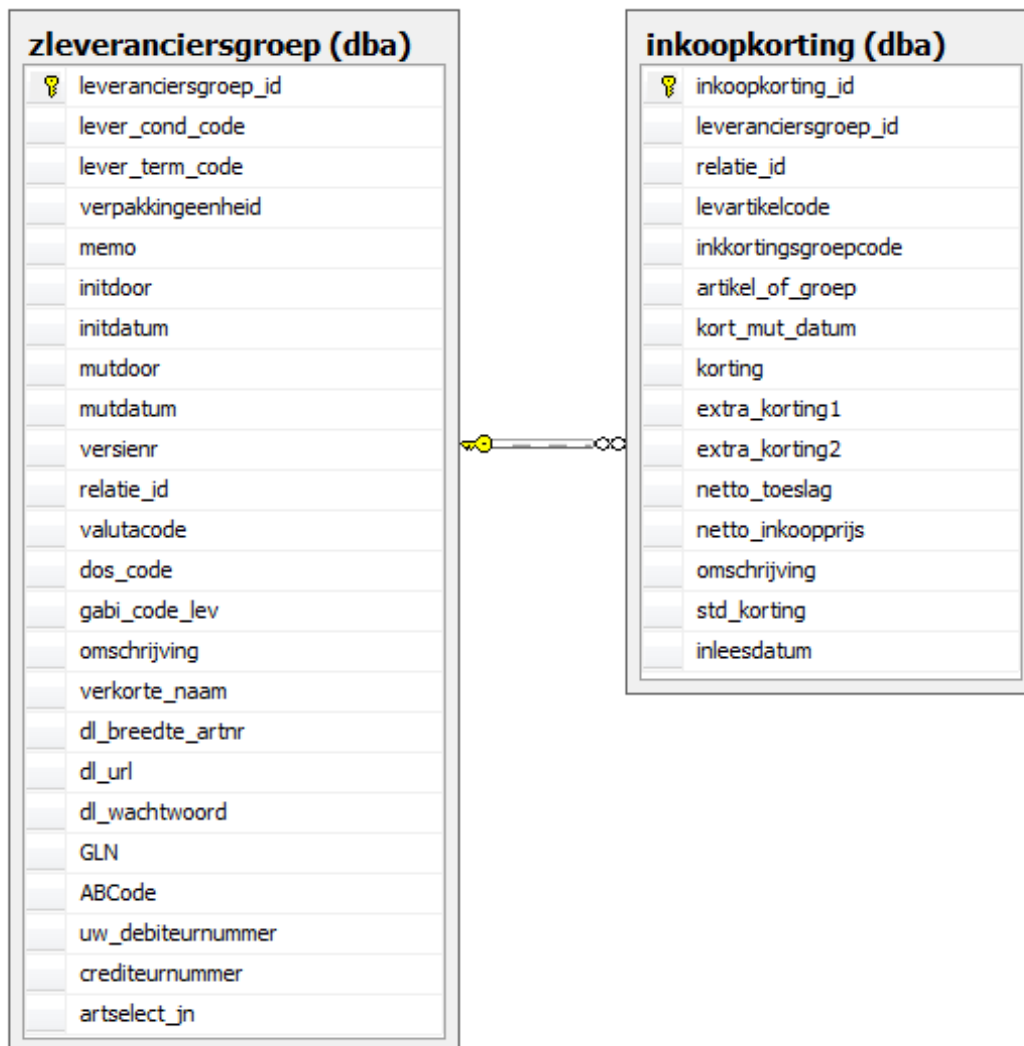


V. Gedeeltelijk ERD eCatalog





VI. Gedeeltelijk ERD InstallCalc



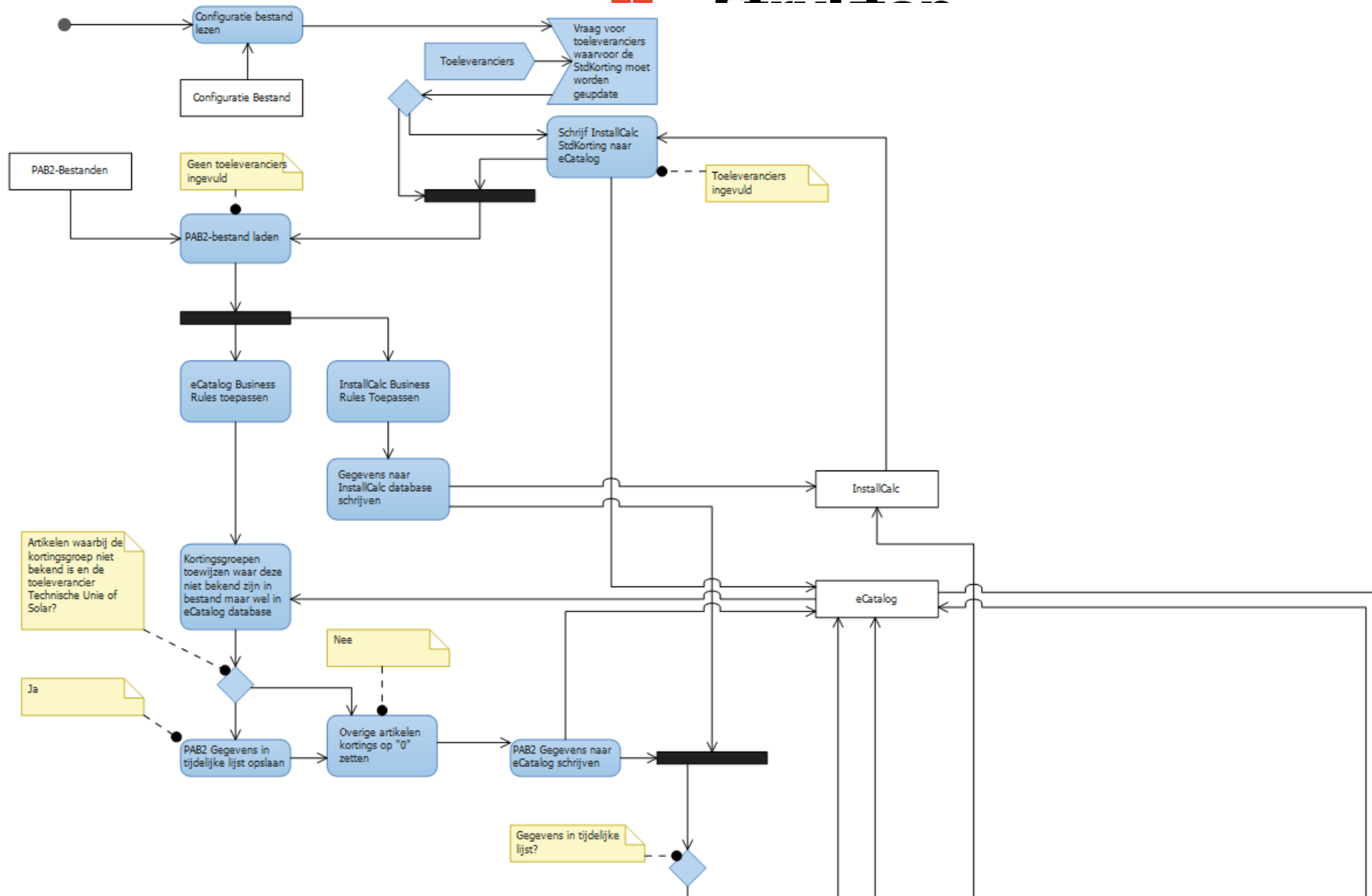


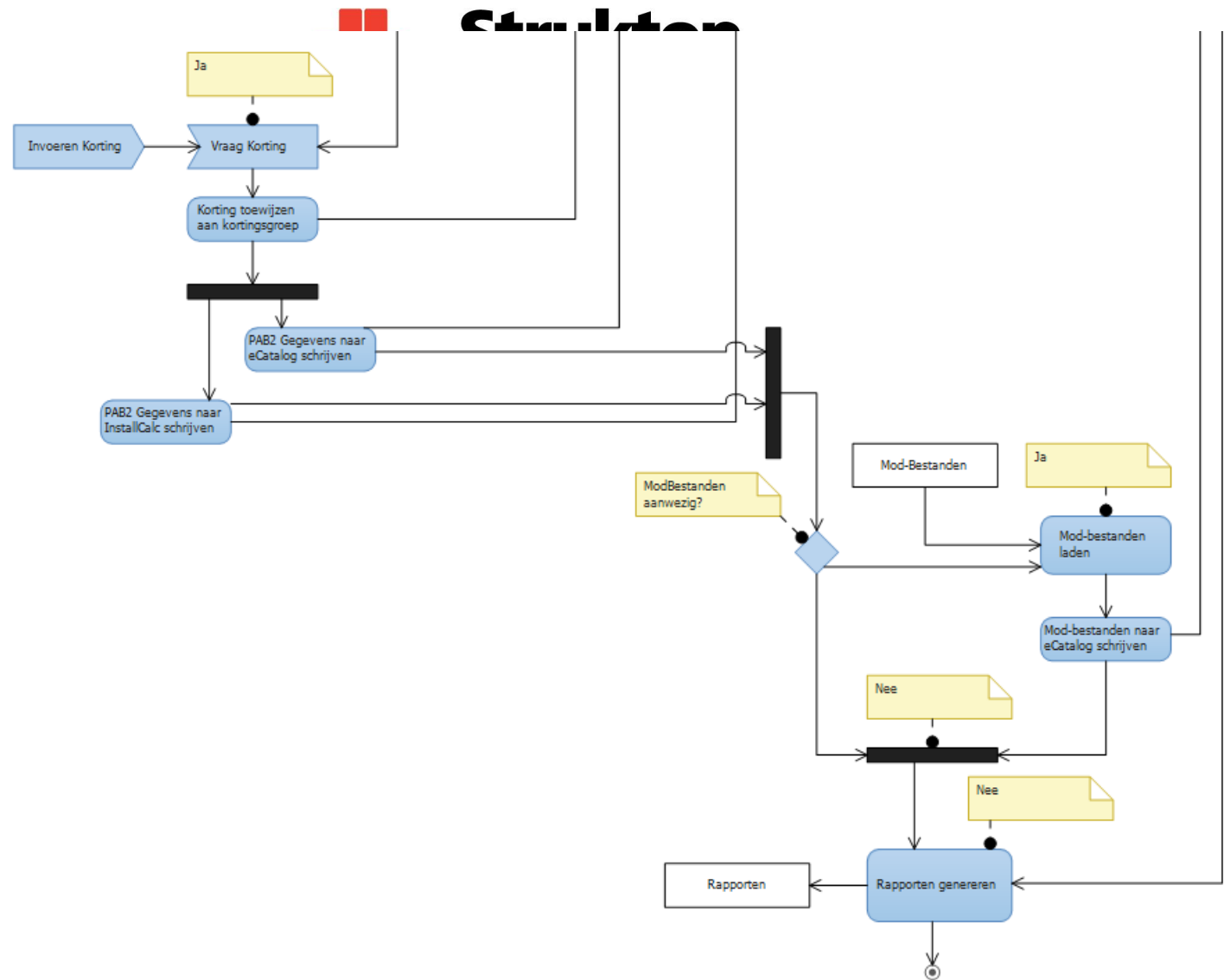
Strukton
Worksphere

VII. Activity Diagram Artikelbestanden Importeren

Totaaloverzicht









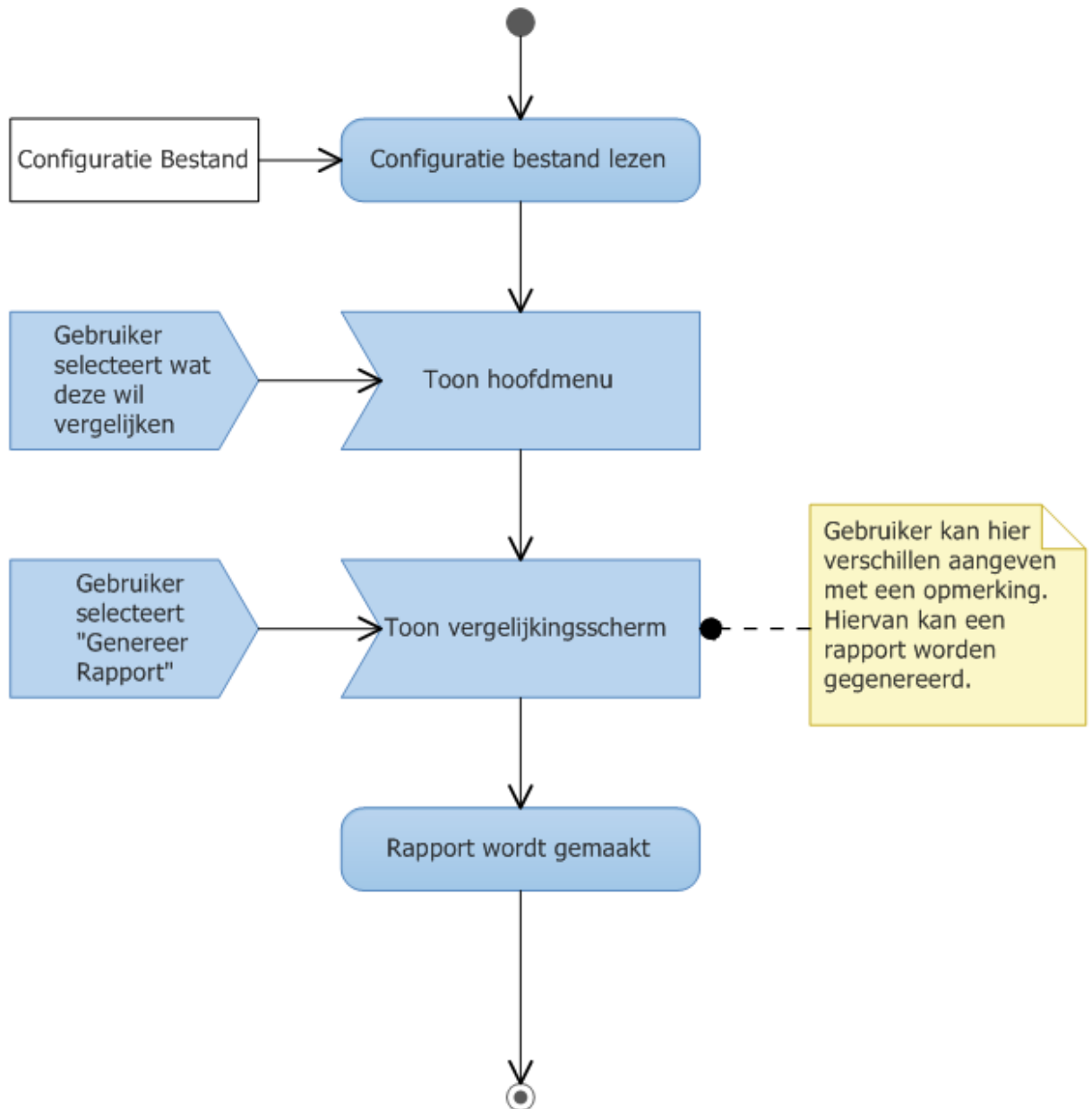
Strukton
Worksphere

VIII. Activity Diagram Kortingsbestanden Importeren

3



IX. Activity Diagram Data Vergelijken



X. PAB2-Bestanden

Waarden in PAB2-bestanden hebben vaste veldlengten.

Formaat	Type
N	Geheel getal (zonder voorloopnullen)
A	Alfanumeriek
Z	Geheel getal met voorloopnullen
D	Decimaal getal D 6.4 betekend max. 6 cijfers voor het decimaalsymbool maximaal 4 na het decimaalsymbool. Maximale waarde in dit geval: "999999.9999" De lengte van dit getal inclusief het decimaalsymbool is 11. In het bestand hoeft er niet uitgevuld te worden met nullen. Waarde "5.3" of "6" is ook toegestaan.

- Decimaalsymbool is "." (punt).
- Een duizend-scheidingsteken is niet toegestaan.

Relatie.txt

Veldnaam	Begin	Lengte	Fomaat	Vereist
GLN	1	13	N	Ja
Naam	14	35	A	Nee
Straat en huisnummer	49	35	A	Nee
Plaats	84	35	A	Nee
Postcode	119	9	A	Nee
Land	128	2	A	Nee

ProdSpec.txt

Veldnaam	Begin	Lengte	Fomaat	Vereist
Productcode fabrikant	1	20	A	Ja
GLN fabrikant	21	13	N	Ja
Code kenmerk	34	8	Z	Ja
Code waarde	42	8	Z	Nee
Technische specificatie	50	70	A	Nee
Numerieke waarde	120	20	D 15.3	Nee
Logische waarde	140	3	A	Nee
Kenmerkonderwaarde	143	20	D 15.3	Nee
Kenmerkbovenwaarde	163	20	D 15.3	Nee

**ArtLev.txt**

Veldnaam	Begin	Lengte	Fomaat	Vereist
Mutatiecode	1	1	N	Ja
Artikelcode	2	20	A	Ja
leverancier				
GLN leverancier	22	13	N	Ja
GTIN	35	14	N	Nee
Ingangsdatum	49	8	N	Nee
prijsneutrale gegevens				
Code	57	3	A	Ja
bestelbaarheid				
Code verwerkbaar	60	3	A	Ja
Statuscode	63	3	A	Nee
GTIN opvolger	66	14	N	Nee
Artikelcode	80	20	A	Nee
opvolger				
GTIN voorganger	100	14	N	Nee
Artikelcode	114	20	A	Nee
voorganger				
Aantal	134	16	D 12.3	Ja
gebruikseenheden				
Gebruikseenheid	150	3	A	Ja
GLN fabrikant	153	13	N	Nee
Productcode	166	20	A	Nee
fabrikant				
GTIN product	186	14	N	Nee
Artikelcode	200	20	A	Nee
fabrikant artikel				
GTIN fabrikant	220	14	N	Nee
artikel				
Afnamegroep	234	17	A	Nee
code				
CBS-nummer	251	17	A	Nee
Artikelomschrijving	268	70	A	Nee
Code	338	3	A	Nee
verpakkingsvorm				
Bruto gewicht	341	19	D 15.3	Nee
Eenheid gewicht	360	3	A	Nee
Hoogte verpakking	363	19	D 15.3	Nee
Lengte verpakking	382	19	D 15.3	Nee
Breedte	401	19	D 15.3	Nee
verpakking				
Eenheid afmeting	420	3	A	Nee
Eenheid afname	423	3	A	Nee
Minimum afname	426	16	D 12.3	Nee
Stapgrootte	442	16	D 12.3	Nee
afname				
Levertijd	458	15	N	Nee



Eenheid levertijd	473	6	A	Nee
Ingangsdatum prijsinformatie	479	8	N	Nee
Code BTW-tarief	487	1	A	Nee
BTW-percentage	488	18	D 13.4	Nee
Indicatie fabrikantprijs volgen	506	3	A	Nee
Bruto prijs bewerkingstoeslag	509	16	D 11.4	Nee
Kortingsgroep	525	35	A	Nee
Bruto prijs	560	16	D 11.4	Nee
Aantal prijsbasis	576	10	D 6.3	Nee
Prijseenheid	586	3	A	Nee
Netto prijs	589	16	D 11.4	Nee
Omrekenfactor prijs- naar besteleenheid	605	16	D 12.3	Nee
Valutacode	621	3	A	Nee

**Product.txt**

Veldnaam	Begin	Lengte	Fomaat	Vereist
Mutatiecode	1	1	N	Ja
Productcode fabrikant	2	20	A	Ja
GLN fabrikant	22	13	N	Ja
GTIN product	35	14	N	Nee
Ingangsdatum	49	8	N	Nee
Productomschrijving	57	70	A	Nee
Statuscode	127	3	A	Nee
GTIN product opvolger	130	14	N	Nee
Productcode opvolger	144	20	A	Nee
GTIN product voorganger	164	14	N	Nee
Productcode voorganger	178	20	A	Nee
Netto Gewicht	198	19	D 15.3	Ja
Eenheid gewicht	217	3	A	Ja
Fabrikaat	220	35	A	Nee
Productserie	255	35	A	Nee
Producttype	290	35	A	Nee
Code	325	4	Z	Nee
productgroep				
Volgnummer	329	3	Z	Nee
productklasse				
Versie normblad	332	2	Z	Nee
Status normblad	334	1	N	Nee
Deeplink	335	512	A	Nee



XI. Mod-Bestanden

Mod-Bestanden zijn TAB-gescheiden ("t").

Formaat	Type
N	Geheel getal (zonder voorlooppullen)
A	Alfanumeriek
Z	Geheel getal met voorlooppullen
D	Decimaal getal D 6.4 betekent max. 6 cijfers voor het decimaalsymbool maximaal 4 na het decimaalsymbool. Maximale waarde in dit geval: "999999.9999" De lengte van dit getal inclusief het decimaalsymbool is 11. In het bestand hoeft er niet uitgevuld te worden met nullen. Waarde "5.3" of "6" is ook toegestaan.

- Decimaalsymbool is "." (punt).
- Een duizend-scheidingsteken is niet toegestaan.

ModArtPlus.txt

Veldnaam	Fomaat	Vereist
ArtikelcodeLeverancier	A	Ja
GLNLeverancier	N	Ja
KM01_Special	N	Nee
KM02_OnwaarschijnlijkHoog		
KM03_Thumbnail	A	Nee
KM04_Afbeelding	A	Nee
KM05_Voorraadhoudend	A	Nee
KM06_KorteOmschrijving	A	Nee
KM07_Omloopsnelheid	A	Nee
KM08_StandaardVerkoopkortingpercentage	D	Nee
KM09_Gebruikseenheid	A	Nee

ModProdSpec.txt

Veldnaam	Fomaat	Vereist
ArtikelcodeLeverancier	A	Ja
Productcode fabrikant	A	Ja
EAN adrescode fabrikant	N	Nee
Code kenmerk	A	Ja
Code waarde	A	Nee
Technische specificatie	A	Nee
Numerieke waarde	D	Nee
Logische waarde	A	Nee
Kenmerkonderwaarde	N	Nee
Kenmerkbovenwaarde	N	Nee
Eenheid	A	Nee



ModProduct.txt

Veldnaam	Fomaat	Vereist
Mutatiecode	N	Ja
ArtikelcodeLeverancier	A	Ja
Productcode fabrikant	A	Nee
GLN fabrikant	N	Nee
GTIN product	N	Nee
Ingangsdatum product	N	Nee
Productomschrijving	A	Nee
Statuscode	A	Nee
GTIN product opvolger	N	Nee
Productcode opvolger	A	Nee
GTIN product voorganger	N	Nee
Productcode voorganger	A	Nee
Netto gewicht	D 15.3	Nee
Eenheid gewicht	A	Nee
Fabrikaat	A	Nee
Productserie	A	Nee
Producttype	A	Nee
Code productgroep	Z	Nee
Volgnummer productklasse	Z	Nee
Versie normblad	Z	Nee
Status normblad	N	Nee
Deeplink	A	Nee

ModSynoniemen.txt

Veldnaam	Fomaat	Vereist
Artikelklasse	A	Ja
Synoniem	A	Ja

XII. ICC-Bestanden

De eerste regel van een ICC bestand heet het beginrecord. In deze regel staat informatie over de toeleverancier.

De rest van de regels zijn Conditierecords. Hierin staan kortingen gedefinieerd.

Waarden in ICC-bestanden hebben vaste veldlengten.

Formaat	Type
N	Geheel getal (zonder voorloopnullen)
A	Alfanumeriek
Z	Geheel getal met voorloopnullen
JJJJMMDD	Datum als jaar maand dag

- Decimaalsymbool is "." (punt).
- Een duizend-scheidingsteken is niet toegestaan.

Beginrecord

Veldnaam	Begin	Lengte	Formaat	Vereist
Leveranciers-ID	1	13	A	Ja
n.v.t.	14	7	A	Nee
Klant-ID	21	20	A	Ja
Productiedatum	41	8	JJJJMMDD	Nee
Aantal regels	49	6	Z	Nee
Versienummer	55	5	A	Nee
Leveranciersnaam	60	35	A	Nee
GLN klant	95	13	A	Nee

Conditierecord

Veldnaam	Begin	Lengte	Formaat	Vereist
Kortingsgroep	1	20	A	Nee
Artikelnummer	21	20	A	Nee
Omschrijving	41	50	A	Nee
Korting 1	91	5	Z	Nee
Korting 2	96	5	Z	Nee
Korting 3	101	5	Z	Nee
Netto prijs	106	9	Z	Nee
Ingangsdatum	115	8	JJJJMMDD	Ja
Einddatum	123	8	JJJJMMDD	Nee



XIII. Rapporten

Artikelverwerker

Wijzigingen.csv
Mutatiecode
Leverancier Relatie Code
Artikelcode Leverancier
Kortingsgroep
Status
Omschrijving Oud
Omschrijving Nieuw
Gebruikseenheid Oud
Gebruikseenheid Nieuw
AantalGebruikseenheden Oud
AantalGebruikseenheden Nieuw
Omrekenfactor Oud
Omrekenfactor Nieuw
Reden voor Rapportage

Dit rapport is een lijst met artikelen waarin de kortingsgroep niet bekend is in de databases, de status "uitlopend" is, de mutatiecode "vervallen" is, de omrekenfactor, bestaande uit; Gebruikseenheid, gebruiksaantal, Prijsseenheid, aantalprijsbasis, omrekenfactor, gewijzigd is en als de omschrijving is gewijzigd.

StamgegevensFabrikantSpecs.csv
Leverancier Relatie Code
Artikelcode Leverancier
Omschrijving Oud
Omschrijving Nieuw
EAN Code Oud
EAN Code Nieuw
Artikelcode Fabrikant Oud
Artikelcode Fabrikant Nieuw
Datum
Reden voor Rapportage

Dit rapport is een lijst met artikelen waarin de fabrikantcode of de GTIN code is veranderd.

Prijsverschillen.csv
Leverancier Relatie Code
Artikelcode Leverancier
Omschrijving
Brutoprijs ph Oud
Brutoprijs ph Nieuw
InkoopKortingsGroep Oud
InkoopKortingsGroep Nieuw
Korting 1 Oud
Korting 1 Nieuw
Verschil Eurocenten
Verschil Netto Procentueel

Dit rapport is een lijst met artikelen waarbij een prijswijziging heeft plaatsgevonden.

Kortingsverwerker

MateriaalafsprakenInkoopVsVerkoop_KortGrpen.csv
Dossiernummer
Dossier omschrijving
Query nr
QueryOmschrijving
Leverancier
Inkoop kortingsgroep
Omschrijving
Inkoopkorting
Verkoopkorting
Opmerking
InstallCalc Ja/Nee
Account manager
Beheerder

In dit rapport staan dossiers van kortingsgroepen die vervallen of veranderd zijn.



XIV. XML-Configuratiebestand

```
<?xml version="1.0" encoding="iso-8859-1"?>
<Config>
  <InstallCalc>
    <Path></Path>
    <DataSourceName></DataSourceName>
    <Username></Username>
    <Password></Password>
  </InstallCalc>
  <PAB2>
    <FileDirectory></FileDirectory>
  </PAB2>
  <MOD>
    <FileDirectory></FileDirectory>
  </MOD>
  <ICC>
    <FileDirectory></FileDirectory>
  </ICC>
  <MutationSettings>
    <SpecialCharacters>
      <Remove></Remove>
      <Characters></Characters>
    </SpecialCharacters>
    <RemoveQuotes></RemoveQuotes>
    <RemoveDecimals> </RemoveDecimals>
    <RemoveKortGrpNotInFile></RemoveKortGrpNotInFile>
  </MutationSettings>
  <RuleSets>
    <RuleSet>
      <Name></Name>
      <Use></Use>
      <Filename></Filename>
      <ObjectType></ObjectType>
    </RuleSet>
    <RuleSet>
      <Name></Name>
      <Use></Use>
      <Filename></Filename>
      <ObjectType></ObjectType>
    </RuleSet>
  </RuleSets>
</Config>
```



InstallCalc

Path

Het pad naar de "xoffice.exe" van InstallCalc.

DataSourceName

De naam van de InstallCalc datasource die gebruikt moet worden.

Username

De gebruikersnaam die gebruikt moet worden.

Password

Het wachtwoord van de gebruiker die bij "Username" is ingevoerd.

PAB2

Filedirectory

De directory waar de PAB2-bestanden zijn opgeslagen.

MOD

Filedirectory

De directory waar de Mod-bestanden zijn opgeslagen.

ICC

Filedirectory

De directory waar de ICC-kortingsbestanden zijn opgeslagen.

MutationSetting

SpecialCharacters

Remove

Als deze op true staat worden alle karakters die onder Characters staan verwijderd van de omschrijvingen in ArtLev.txt.

Characters

Geeft aan welke karakters verwijderd moeten worden.

RemoveQuotes

Als deze op true staat worden single en double quotes van ArtLev.txt omschrijvingen verwijderd.

RemoveDecimals

Als deze op true staat worden decimalen die geen rond getal zijn op 1 gezet en worden eenheden op PCE (st) gezet.

RemoveKortGrpNotInFile

Als deze op true staat worden kortingsgroepen van de betreffende toeleveranciers waarvan het ICC-bestand wordt geïmporteerd, verwijderd uit de eCatalog database, als deze niet in het ICC-bestand staan.

RuleSets

RuleSet

Name

De naam van de RuleSet.

Use

Hier wordt aangegeven of de RuleSet gebruikt moet worden. Mogelijke waarden zijn "true" of "false".

Filename

De bestandsnaam van de RuleSet.

ObjectType

Het type object waar de RuleSet op van toepassing is.



XV. App.Config

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit
http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework,
Version=4.4.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
requirePermission="false" />
  </configSections>
  <connectionStrings>
    <add name="eCatalogDatabaseConnection"
connectionString="metadata=res://*/eCatalogDatabaseModel.csdl|res://*/eCatalogDatabase
Model.ssdl|res://*/eCatalogDatabaseModel.msl;provider=System.Data.SqlClient;provider
connection string=&quot;;data source=localhost\SJORSLOCALMSSQL;initial
catalog=eCatalog;integrated security=True;
MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.EntityClient" />
    <add name="InstallCalcDatabaseConnection"
connectionString="metadata=res://*/InstallcalcDatabaseModel.csdl|res://*/InstallcalcDa
tatabaseModel.ssdl|res://*/InstallcalcDatabaseModel.msl;provider=System.Data.SqlClient;p
rovider connection string=&quot;;data source=localhost\SJORSLOCALMSSQL;initial
catalog=InstallCalc;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.EntityClient" />
    <add name="eCustomerDatabaseConnection"
connectionString="metadata=res://*/eCustomerDatabaseModel.csdl|res://*/eCustomerDataba
seModel.ssdl|res://*/eCustomerDatabaseModel.msl;provider=System.Data.SqlClient;provide
r connection string=&quot;;data source=localhost\SJORSLOCALMSSQL;initial
catalog=eCustomer;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.EntityClient" />
  </connectionStrings>
  <entityFramework>
    <defaultConnectionFactory
type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">
      <parameters>
        <parameter value="v11.0" />
      </parameters>
    </defaultConnectionFactory>
  </entityFramework>
</configuration>
```