

UNIT4 SOFTWARE B.V.

Afstudeerscriptie

Fabric Web Applicatie - Bijlagen

Raymond Meerburg

9/23/2012

Dit is een verzameling van de bijlagen bij de Fabric Web Applicatie afstudeerscriptie.

Afstudeerplan

Informatie afstudeerder en gastbedrijf *(structuur niet wijzigen)*

Afstudeerblok: 2012-1.2 (start uiterlijk 7 mei 2012)

Startdatum uitvoering afstudeeropdracht: 23 April 2012

Inleverdatum afstudeerdossier volgens jaarrooster: 24 september 2012

Studentnummer: 08056757

Achternaam: dhr Meerburg

Voorletters: RJ

Roepnaam: Raymond

Adres: IJsvogel 8

Postcode: 3362 KC

Woonplaats: Sliedrecht

Telefoonnummer: 0184422416

Mobiel nummer: 0611444701

Privé emailadres: rmeerburg@gmail.com

Opleiding: Technische Informatica

Locatie: Delft

Variant: voltijd

Naam studieloopbaanbegeleider: dhr. A.G.P. Pronk

Naam begeleidend examiner: John Visser

Naam tweede examiner: Henk van den Bosch

Naam bedrijf: UNIT4 Software B.V.

Afdeling bedrijf: Product Ontwikkeling

Bezoekadres bedrijf: Stationspark 1000

Postcode bezoekadres: 3364 DA

Postbusnummer: 102

Postcode postbusnummer: 3360 AC

Plaats: Sliedrecht

Telefoon bedrijf: 0184 444444

Telefax bedrijf: 0184 414819

Internetsite bedrijf: <http://www.unit4.nl/>

Achternaam opdrachtgever: dhr Klumpers

Voorletters opdrachtgever: K

Titulatuur opdrachtgever:

Functie opdrachtgever: Product Development Manager

Doorkiesnummer opdrachtgever:

Email opdrachtgever: kuno.klumpers@unit4.com

Achternaam bedrijfsmentor: dhr Nijsen

Voorletters bedrijfsmentor: Y

Titulatuur bedrijfsmentor:

Functie bedrijfsmentor: Software Architect

Doorkiesnummer bedrijfsmentor:

Email bedrijfsmentor: yuri.nijsen@unit4.com

Doorkiesnummer afstudeerder:

Functie afstudeerder (deeltijd/duaal):

TITEL AFSTUDEEROPDRACHT:

Bewijs dat de nieuwe business logic paraat is om te dienen als framework voor web applicaties.

BEDRIJF

UNIT4 is een internationaal opererend for-profit bedrijf welke zich bezig houdt met het ontwikkelen, verkopen, implementeren en ondersteunen van oplossingen waarmee organisaties gemakkelijk aan hun veranderende bedrijfsbehoeften kunnen voldoen.

Dit houdt in dat het bedrijf software ontwikkelt om bedrijfsprocessen te automatiseren. Hierbij kan gedacht worden aan ERP, HRM, finance, ICT etc.

UNIT4 N.V. is sinds 1998 genoteerd aan de Nederlandse aandelenbeurs, heeft meer dan 4.200 medewerkers in dienst en realiseerde in 2011 een omzet van € 454,7 miljoen.

PROBLEEMSTELLING

De business laag van UNIT4 is opgedeeld in vijf lagen. Dit systeem wordt Fabric genoemd. Fabric is een uitgebreid framework welke door derden gebruikt wordt om applicaties te schrijven die services van Unit4 gebruiken. Via Fabric kunnen gegevens zoals sales, logistics, CRM, financial etc. benaderd worden. Fabric maakt hiervan business objects aan die gebruikt kunnen worden in applicaties. Error handling, authenticatie, verificatie etc. wordt allemaal door Fabric geregeld.

Er is steeds meer vraag naar web applicaties. Het is wenselijk om de business logic zo in te richten dat er zowel desktop applicaties als web applicaties op kunnen draaien. Het is hierbij belangrijk dat beide applicaties kunnen berusten op dezelfde implementatie en dat er geen gebruikt wordt gemaakt van web of desktop specifieke functionaliteit.

Partners van UNIT4 willen graag Fabric gebruiken om services via het web aan te bieden. Unit4 weet niet of Fabric daar op dit moment gereed voor is. Zij willen bewijs hebben dat Fabric er klaar voor is.

Unit4 wil ook zelf Fabric gebruiken in web applicaties. De inrichting van de servers hiervoor moet gemaakt worden.

DOELSTELLING VAN DE AFSTUDEEROPDRACHT

Het doel van de opdracht is enerzijds het uitbrengen van een rapport over de paraatheid van Fabric m.b.t. het dienen als framework voor web applicaties. Anderzijds moeten die delen die nog niet in orde zijn, aangepast worden. Naast het onderzoeken van Fabric en het bouwen van de app moet ook onderzocht worden hoe de servers het beste ingericht kunnen worden m.b.t. performance en availability.

RESULTAAT

Bij succesvolle afronding van het project weet Unit4 zeker dat Fabric voldoende paraat voor de ontwikkeling van webapps die erop berusten. Dit betekent dat UNIT4 Fabric als ontwikkelplatform voor web apps kan aanbieden aan partners.

UIT TE VOEREN WERKZAAMHEDEN, INCLUSIEF EEN GLOBALE FASERING, MIJLPALLEN EN BIJBEHORENDE ACTIVITEITEN

Het bewijzen van en onderzoeken naar de paraatheid van Fabric zal gebeuren door een proof-of-concept web applicatie te bouwen welke enkele financiële bedrijfsprocessen kan beheren. De aard van de web app wordt een CRM systeem.

Enkele aspecten die hierbij belangrijk zijn:

- Configureerbaar en flexibel.
- Goede performance op de server.
- Caching van data bij de client en server om performance te verbeteren.
- Om kunnen gaan met foute input.

De ontwikkelmethode voor het hele project is SCRUM, er worden sprints van ongeveer twee weken aangehouden. Tijdens elke fase van de ontwikkeling worden de relevante UML modellen gemaakt. Software wordt ontwikkeld in C# met ASP .NET MVC 3 als framework.

Een globale fasering van het traject is als volgt:

- Week 1: Plan van aanpak schrijven, oriënteren op Fabric. Details opdracht verduidelijken. Project plannen.
- Week 2: Opstellen requirements web app. Ontwerp maken. Functionele ontwerpen maken, eisen bepalen. Implementatie plannen.
- Week 3-13: Implementatie web applicatie tegelijk met het aanpassen van Fabric.
- Week 13-15: client- en serverside caching implementeren.
- Week 15-16: Ontwikkelaars tests, gebruikerstests, laatste bugs fixen.
- Week 17-18: schrijven rapport, afmaken afstudeerdossier.

Het kerndoel van de opdracht is niet het bouwen van de web applicatie, maar het onderzoeken van de paraatheid van Fabric. Door de web applicatie te schrijven ontdek ik welke problemen ontwikkelaars in de toekomst kunnen verwacht en op deze punten kan Fabric aangepast worden. Dit betekent absoluut niet dat de ontwikkeling van de web applicatie triviaal is. Het doel is om zo veel mogelijk functionaliteit uit Fabric naar voren te brengen in de web applicatie.

OP TE LEVEREN (TUSSEN)PRODUCTEN

Als eerst zal een plan van aanpak opgeleverd worden. Daarna wordt bekeken welke technische en/of functionele ontwerpen er gemaakt en opgeleverd moeten worden.

De complete opdracht levert enerzijds een rapport op over de paraatheid van Fabric. Dit bestaat uit een beschrijving van welke aspecten er veranderd zijn, en of er nog meer aanpassingen nodig zijn.

Er kan ook een aangepaste versie van Fabric opgeleverd worden.

TE DEMONSTREREN COMPETENTIES EN WIJZE WAAROP

Hoofdcompetenties:

Bijlage I Afstudeerplan

- **A5 Opstellen van systeemeisen:** Het analyseren van de behoeften en het opstellen van de systeemeisen voor de webapp is een belangrijke stap in de ontwikkeling.
- **C8 Ontwerpen van een technisch informatie systeem:** Het ontwerp van de webapp wordt door mij gedaan. Hierbij ga ik het ontwerp zo goed mogelijk in UML uitdrukken.
- **C10 Ontwerpen van een systeemarchitectuur:** Nauw verwant aan C8, de verschillende componenten van het systeem, en de samenhangt van het ontworpen systeem in het grotere geheel wordt in UML uitgedrukt. Daarnaast worden er standaarden en inrichtingen gekozen voor het systeem.
- **D16 Realiseren van software:** Het is belangrijk dat de broncode niet alleen werkt, maar ook van goede kwaliteit is. Het moet zo makkelijk mogelijk zijn voor iemand anders om er mee verder te werken.
- **H5 Zelfstandig werken:** De opdracht moet zo veel mogelijk zelfstandig worden uitgevoerd.
- **D17 Testen van software systemen:** De software moet op een goede manier getest worden, de tests moeten relevant en dekkend zijn. Het resultaat van de tests moet op een zinvolle manier terug gekoppeld worden.

Randcompetenties:

- **C11 Ontwerpen van een user interface:** Er moet een ontwerp voor een user interface gemaakt worden, hoewel niet erg belangrijk kwa uiterlijk moet deze wel goed werken.

Competentie verantwoording

Vóór het afstuderen zijn een aantal competenties (ook beroepstaken genomen) ingevuld. Tijdens het afstudeertraject moeten deze competenties worden bewezen. Alle competenties hebben een ander onderwerp en andere toets criteria. In dit hoofdstuk geef ik aan welke competenties ik heb ingevuld en op welke manier ik heb aangetoond dat ik deze bezit.

Hoofdcompetenties:

- **A4 Kiezen van een ontwikkelstrategie en ontwikkelmethodiek:** De eisen van het eindproduct zijn vanaf het begin nog niet compleet duidelijk, er moet een ontwikkelstrategie worden gekozen die hiermee goed om kan gaan. Daarnaast moet er een adequate ontwikkelmethodiek worden gekozen.

In het plan van aanpak heb ik verschillende ontwikkelstrategieën met elkaar vergeleken en de sterke en zwakke kanten met elkaar afgewogen. Ik heb als eerste bepaald dat een incrementele strategie goed bij het project past. Die keuze heb ik verder vernauwd naar Scrum. Dit omwille de uitstekende flexibiliteit, maar zonder de overbodige complexiteit van andere ontwikkelmethoden.

Naast het kiezen van de ontwikkelstrategie heb ik ook de ontwikkelmethode vooraf vastgelegd. Ik een algemene werkwijze gedefinieerd en de invulling van de werkzaamheden per sprint gepland. Gedurende het project heb ik me goed gehouden aan de opgestelde richtlijnen.

- **A5 Opstellen van systeemeisen:** Het analyseren van de behoeften en het opstellen van de systeemeisen voor de applicatie is een belangrijke stap in de ontwikkeling.

Bij de aanvang van het project zijn de eisen nog verre van duidelijk. Gedurende het project zullen de eisen steeds worden aangevuld. De eisen komen voort uit de wensen van de product owner en input van mijzelf. Ik heb vooraf een schatting gemaakt van de eisen waaraan het product zal moeten voldoen. Daarnaast heb ik aan het begin van elke sprint de eisen geanalyseerd en omgezet in een user story. Ik heb waar nodig use cases en andere functionele diagrammen gemaakt bij de eisen. Het uiteindelijke product voldoet aan alle *must-have* eisen, en de meeste *should-have* eisen.

- **C8 Ontwerpen van een technisch informatie systeem:** Het ontwerp van de applicatie wordt door mij gedaan. Hierbij ga ik het ontwerp in UML uitdrukken.

De technische ontwerpen van het systeem worden in UML uitgedrukt. De UML diagrammen zijn tot stand gekomen door eerst een analyse van het huidige systeem, de eisen en functionele diagrammen uit te voeren. Deze analyse moet inzicht geven in de technische eisen van het systeem. Deze eisen worden in een voorgesteld ontwerp gerealiseerd. Tijdens de ontwikkeling is er soms afgeweken van het ontwerp. De technische diagrammen zijn daarop aangepast zodat deze relevant blijven.

Een ontwikkelaar kan het systeem begrijpen door naar de technische UML diagrammen te kijken. Deze geven het huidige ontwerp van het systeem goed weer. Bij de diagrammen is ook commentaar geleverd.

- **C10 Ontwerpen van een systeemarchitectuur:** Nauw verwant aan C8, de verschillende componenten van het systeem, en de samenhangt van het ontworpen systeem in het grotere geheel wordt in UML uitgedrukt. Daarnaast worden er standaarden gekozen voor het systeem.

Het project gaat om het bouwen van een nieuw systeem op een nieuw framework. Het complete ontwerp wordt door mij gemaakt. Hierbij is het belangrijk dat er altijd vooruit gedacht wordt aan nieuwe componenten of veranderingen in de eisen. Het systeem is ontworpen met flexibiliteit in het achterhoofd. Er is zo veel mogelijk gebruik gemaakt van abstracties en indirecties om de uitbreiding, aanpassing en onderhoud van het systeem te vergemakkelijken.

- **D16 Realiseren van software:** Het is belangrijk dat de broncode niet alleen werkt, maar ook van goede kwaliteit is. Het realiseren van het systeem moet op systematische wijze gebeuren. Het moet daarnaast zo gemakkelijk mogelijk zijn voor iemand anders om er mee verder te werken. De code moet van documentatie zijn voorzien.

Het realiseren van de software is dankzij de goede technische diagrammen meestal vlot verlopen. Er is in dit project gebruik gemaakt van meerdere programmeertalen (C#, JavaScript), opmaaktalen (HTML, CSS) en querytalen (SQL). Alle code functioneert naar behoren en is ook conform de richtlijnen binnen het bedrijf. De sterke kanten van C# en het .NET framework zijn waar mogelijk toegepast. De broncode is voorzien van commentaar in een formeel formaat evenals informele toelichtingen. Er is gelet op de organisatie van de broncode door gebruik te maken van regions en in de code zelf hebben variabelen zinnige namen gekregen en zijn statements logisch opgedeeld. De source code is door middel van static analysis onderzocht en verbeterd. De verschillende componenten van het systeem werken goed samen en het eindproduct functioneert volledig.

- **D17 Testen van software systemen:** De software moet op een goede manier getest worden, de tests moeten relevant en dekkend zijn. Het resultaat van de tests moet op een zinvolle manier terug gekoppeld worden.

Voor de klassen waar nodig en mogelijk zijn er unit tests gemaakt. Deze unit tests zijn tijdens het ontwikkelingstraject van onschatbare waarde geweest. Zij hebben de correctheid van het product bewezen wanneer zij slagen, of geven een waarschuwing af wanneer een verandering heeft gezorgd voor regressie. Hierdoor kan er telkens met goed vertrouwen aanpassingen aan afgeronde systemen worden uitgevoerd. Naast unit tests is er tegen het einde gebruik gemaakt van integratie tests. Deze tests hebben de samenwerking van alle componenten in het project met succes getest. Naast deze twee functionele testsoorten zijn er ook performance tests uitgevoerd. Hiervan zijn rapporten gemaakt die hebben geholpen bij het vinden van performance problemen. Aan de aanvang van het project is het test plan ingevuld, hierin staat hoe het testen precies zal geschieden.

- **G1 Praktische aspecten hanteren in projecten:** De opdrachtomschrijving moet duidelijk zijn geformuleerd. Juist omdat er onzekerheid bestaat over wat er precies geïmplementeerd moet worden is het belangrijk een risicoanalyse uit te voeren.

De opdracht omschrijving geeft het probleem duidelijk en ondubbelzinnig weer. De aanleiding, probleemstelling, doelstelling, op te leveren producten en opdrachtomschrijving komen hierin aan bod. In het plan van aanpak staat volgens welke manier er gewerkt wordt en welke fasering er wordt toegepast. Er wordt al een voorlopige indeling van de werkzaamheden gemaakt door middel van een planning. Verder zijn het plan

van aanpak zijn de risico's van het project in kaart gebracht en geanalyseerd. Aan elk risico is de kans en de schade geschat. Hierdoor zijn de belangrijkste risico's duidelijker naar voren gekomen waardoor het duidelijk werd aan welke risico's de meeste aandacht besteed moest worden.

- **H5 Zelfstandig werken:** De opdracht moet zo veel mogelijk zelfstandig worden uitgevoerd. Alle te gebruiken methodieken moeten zelf gekozen worden en ook de opdracht moet nog deels worden ingevuld door mij.

De opdracht is voor het overgrote deel zelfstandig uitgevoerd. Er is zelf een planning gemaakt en alle gebruikte technieken en methoden zijn door mij na afweging van voor en nadelen uitgekozen. Het is tijdens het project belangrijk geweest om zelf veel inbreng te geven in het systeem. Ik ben vrij geweest om features voor te stellen voor het product en te implementeren.

Er heeft alleen samenwerking plaatsgevonden met de begeleider of collega's wanneer ik onvoldoende kennis of informatie had over het systeem om een juist ontwerp te maken. Of wanneer er bugs optraden in Fabric die ik niet zelf kon repareren.

- **H7 Professioneel werken: Methodisch werken:** Het spreekt voor zich dat de in A4 opgestelde ontwikkelstrategie en methodiek goed gevolgd moet worden.

De opgestelde sprint planning is met uitzondering van één sprint altijd aangehouden. Er is een sprint uitgesteld omdat hierin een concept verslag opgeleverd moest worden. Binnen elke sprint zijn vaak de meeste user stories direct geïmplementeerd. In sommige gevallen zijn er user stories doorgeschoven naar volgende sprints. Juist omdat het niet altijd zeker is hoe lang een component duurt om te realiseren is er gewerkt met Scrum. Het is daarbij geen probleem als er een user story doorgeschoven moet worden.

Er is een vrij gedetailleerde werkwijze opgesteld aan het begin van het project, deze is gedurende de uitvoering met succes aangehouden.

Randcompetenties:

- **C11 Ontwerpen van een user interface:** Er moet een ontwerp voor een user interface gemaakt worden, hoewel niet erg belangrijk kwa uiterlijk moet deze wel goed werken.

Niet alleen de backend maar ook de interface van het product moet door mij worden ontworpen en geïmplementeerd. Hierbij is het belangrijk dat alle schermen duidelijk zijn en de gegevens op een logische manier zijn geordend. De gebruiker wordt waar mogelijk tijdens gebruik geassisteerd door het systeem.

Bespreking concept	Tussentijds assessment	Eerste beoordeling
---------------------------	-------------------------------	---------------------------

Formulier bespreking concept afstudeerdossier

Student: Raymond Meerburg

Studentnummer: 08056757

Datum: 20 juli 2012

Tijdens de bespreking is het volgende geconstateerd:		ja	nee
a	<i>Het voortgangsverslag is ontvangen</i>	x	
b	<i>Het afstudeerdossier is digitaal beschikbaar</i>		x
c	<i>Het afstudeerdossier is opgebouwd conform de richtlijnen</i>		x
d	<i>Het goedgekeurde afstudeerplan is aanwezig</i>	x	
e	<i>Het plan van aanpak is aanwezig</i>	x	
f	<i>Reeds geleverd commentaar is aanwezig</i>		
g	<i>Het afstudeerdossier geeft voldoende inzicht in de stand van zaken</i>	x	
h	<i>De afstudeeropdracht is tot nu toe naar behoren uitgevoerd</i>	x	

Verbeterpunten:

Verslag op diverse punten aanpassen.

Opmerkingen:

Raymond ligt op schema.

Naam begeleidend examiner: John Visser

Datum: 31- 07-2012

Dit formulier wordt door de begeleidend examiner digitaal ingevuld en per email naar de student verstuurd met een cc naar de coördinator van ICT & Media @ Work (A.M.Schipper@hhs.nl). Het formulier dient door de student te worden opgenomen in het afstudeerdossier.

Bespreking concept	Tussentijds assessment	Eerste beoordeling
--------------------	------------------------	--------------------

Formulier tussentijds assessment

Student: Raymond Meerburg

Studentnummer: 08056757

Datum: 29 augustus 2012

eerste/~~tweede~~ TTA:

Tijdens het tussentijds assessment is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	x	
b	Het afstudeerdossier is digitaal beschikbaar	x	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen	x	
d	Het goedgekeurde afstudeerplan is aanwezig	x	
e	Het plan van aanpak is aanwezig	x	
f	Reeds geleverd commentaar is aanwezig	x	
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken	x	
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	x	

Aanpak	O	T	V	G
Passend			x	
Theoretisch verantwoord			x	
Samenhang uitvoering beroepstaken			x	

Beroepstaken op afgesproken niveau uitgevoerd?		O	T	V	G
1	A5 Opstellen van systeemeisen			x	
2	C8 Ontwerpen van een technisch informatie systeem			x	
3	C10 Ontwerpen van een systeemarchitectuur			x	
4	D16 Realiseren van software			x	
5	H5 Zelfstandig werken			x	

Bijlage III.2

6	D17 Testen van software systemen			x	
7					
8					

Producten	O	T	V	G
<i>Tussenproducten</i>			x	
<i>Eindproducten</i>			x	

Effectief communiceren	O	T	V	G
<i>Binnen afstudeerbedrijf</i>			x	
<i>Afstudeerdossier</i>			x	

Reflectie	O	T	V	G
<i>Inzicht in eigen functioneren</i>			x	
<i>Inzicht in eigen leerproces</i>			x	

Toelichting per beoordelingscriterium

Aanpak
Gekozen is voor een geschikte ontwikkelmethode (SCRUM), en deze is voldoende uitgevoerd. De keuze is van relevante argumenten voorzien.

Beroepstaken op afgesproken niveau uitgevoerd?
Alle taken zijn op een voldoende niveau (HBO) uitgevoerd.

Producten
Het verslag is niet af, er moet nog veel worden gedaan (gedeelten van Sprints, gedeelten van de Evaluatie), maar de structuur en de inhoud van het huidige concept zijn voldoende en geven vertrouwen in het vervolg.

Bijlage III.2

Effectief communiceren
Raymond heeft bij het bedrijf een baan aangeboden gekregen en heeft deze geaccepteerd.

Reflectie
OK

Advies

x	Inleveren (bindend advies)
	Verlengen (vrijblijvend advies)
	Stoppen (vrijblijvend advies)

Besluit student

Aankruisen welke beslissing de student heeft genomen (alleen na vrijblijvend advies)

x	Afstudeerdossier wordt op afgesproken datum ingeleverd datum: 24 september 2012
	Afstudeerperiode wordt verlengd Inleverdatum:
	Student stopt met afstudeeropdracht

Naam begeleidend examinerator: John Visser

Naam tweede examinerator: Henk van den Bosch

Datum: 29 augustus 2012

Dit formulier wordt door de tweede examinerator digitaal ingevuld, waarna de begeleidend examinerator het per email verstuurt naar de student met een cc naar de coördinator van ICT & Media @ Work (A.M.Schipper@hhs.nl). Het formulier dient door de student te worden opgenomen in het afstudeerdossier.

Requirements analyse

Een uiteenzetting van de eisen van de applicatie

Raymond Meerburg

5/7/2012

In dit document staan de belangrijkste eisen waaraan de Fabric Web Applicatie moet voldoen. Ook de eisen van het Fabric bevindingen rapport staan hier in. De risico analyse is ook in dit document opgenomen.

1 INHOUD

2	Inleiding.....	2
3	Projectgrenzen	3
4	Randvoorwaarden	4
5	Eisen	5
5.1	Functionele eisen.....	5
5.2	Niet functionele eisen.....	6
6	Use cases	8
6.1	Authenticatie	8
6.2	Lijstweergave business objects.....	9
6.3	Aanmaken business objects	11
6.4	Opvragen business objects	12
6.5	Wijzigen business objects	12
6.6	Verwijderen business objects	13
6.7	Debiteur	13
6.8	Algemene features	14
6.9	Aanmaken business objects	16
7	Bevindingen rapport	17
8	Risicomanagement.....	18
8.1	Risicoanalyse.....	18

2 INLEIDING

In dit document zijn de eisen en specificaties vastgelegd van de te ontwikkelen web applicatie. De eisen worden opgedeeld in functionele en niet functionele eisen en krijgen een prioriteit toegekend. Na het bepalen van de eisen worden er use cases opgesteld die één taak beschrijven, de use cases geven het gebruik van het systeem weer.

Een risicoanalyse zal de belangrijkste risico's van dit project in kaart brengen en prioriteren. Elk risico heeft een tegenmaatregel, de maatregel kan ingezet worden om de kans van het risico te verminderen of de schade te beperken wanneer het risico optreedt.

Naast het ontwikkelen van de applicatie dient er ook een bevindingenrapport te worden opgesteld. De inrichting hiervan wordt beschreven aan het einde van dit document. Tot slot volgt ook een beknopte risico analyse.

3 PROJECTGRENZEN

De grenzen geven de afbakening van het project weer. Hiermee wordt duidelijk welke onderdelen er wél en niet horen bij de uitvoering van dit project.

- De volgende zaken vallen binnen de scope van dit project:
 - Web applicatie:
 - De implementatie van de applicatie.
 - Documentatie
 - Volledige documentatie van alle opgeleverde source code in XML formaat in de source files.
 - Toelichting bij opmerkelijke code.
 - Tests
 - Opstellen van tests die de correcte werking van de applicatie bewaken.
 - Beschrijving van tests waar nodig.
 - Bevindingen rapport:
 - Een bevindingen vanuit het oogpunt van een gebruiker van Fabric waarin naar voren komen de problemen, onhandigheden en fouten in Fabric.
 - Een mogelijke oorzaak en voorgestelde oplossing op de beschreven problemen en fouten.
 - Suggesties die het gebruik van Fabric vriendelijker kunnen maken voor ontwikkelaars.
- De volgende zaken van buiten de scope van dit project:
 - De release en/of distributie van het concept.
 - Het toepassen van de in het bevindingenrapport voorgestelde oplossingen.
 - Het volledig onderzoeken van alle Fabric functionaliteit.
 - Verder onderhoud aan de applicatie.

4 RANDVOORWAARDEN

Randvoorwaarden stellen eisen aan het project die vanuit het project niet kunnen worden beïnvloed. De voorwaarden geven vaak de verwachtingen van de opdrachtgever weer.

- Performance:
 - De applicatie reageert snel, zelfs wanneer er grote datasets worden gebruikt.
- Portability:
 - De interface van de web applicatie zal getest worden in Opera en Mozilla Firefox. Internet Explorer wordt niet expliciet ondersteund.
 - De applicatie richt zich op versie 4 van het .NET framework en versie 4 van het ASP .NET MVC framework.
- Overdraagbaarheid:
 - Er wordt zo veel mogelijk gehouden aan de code standaarden van UNIT4.
- Beveiliging:
 - De gebruikte authenticatie methode berust op betrouwbare en bewezen technieken, dit onderdeel wordt niet opnieuw geïmplementeerd.

5 EISEN

Voor het prioriteren van de items word gebruik gemaakt van de MoSCoW methode¹. Deze methode kent een prioriteit toe in de vorm van categorieën. Elk item komt in één van de volgende vier categorieën:

- **Must have.** Deze functie MOET in het product zitten. Dit is een vereiste functie, het product is geen succes tot alle Must have's erin zitten.
- **Should have.** Hoge prioriteit, dit is een belangrijke vereiste, maar kan op andere manier ingevuld worden indient noodzakelijk.
- **Could have.** Deze vereist is gewenst, maar niet noodzakelijk. Deze items worden ingevuld als de tijd het toe laat.
- **Won't have.** Er is overeenstemming dat dit item niet in het eindproduct zal zitten, wellicht kunnen deze eisen later in een volgend project worden toegevoegd.

Tijdens het project zullen over het algemeen de 'Must have' eisen als eerste worden aangepakt. Gevolgd door de 'Should have' eisen. Het is vaak wel tactisch zijn om alvast te denken aan eisen met een lagere prioriteit ook al komen deze nog niet aan bod. Vaak is het werk besparend als het system op een zodanige manier wordt ontwerpen dat er al rekening gehouden wordt met de minder belangrijke eisen die later aan bod komen. Op die manier hoeft het ontwerp van het systeem niet in een later stadium herzien te worden.

5.1 FUNCTIONELE EISEN

De functionele eisen geven de gewenste functionaliteit van het systeem weer. Hieruit kan bepaald werken *wat* het systeem moet kunnen.

De eisen van de applicatie gaan onder andere over de verschillende schermen. Veel schermen delen een type functionaliteit zoals het aanmaken, ophalen, wijzigen en verwijderen van objecten die bij de schermen horen. Alle eisen hebben een nummer, hiermee kunnen de eisen geïdentificeerd worden. De volgende functionele eisen zijn opgesteld:

TABEL 1 OPGAVE VAN FUNCTIONELE EISEN

Functionele eis	Prioriteit
Algemeen:	
Wisselen van scherm	Must
Produceren van log bestanden	Should
Wisselen van weergavetaal	Could
Authenticatie:	
Authenticatie van gebruikers	Must
Optie om ingelogd te blijven	Must
Lijstweergave van business objects:	
Lijst business objects kunnen opvragen	Must
Filteren van objecten in de lijst	Must
Waarden verspreiden over meerdere pagina's	Should
Sorteren van waarden in de lijst	Could
Maximale lijstgrootte aangeven	Could
Indicatie voortgang ophalen/zoeken	Could
Optioneel dynamisch bijladen van resultaten in plaats van paging gebruiken	Could

¹ <http://www.coleyconsulting.co.uk/moscow.htm>

Debiteuren:		
Debiteur opvragen		Must
Nieuwe debiteur aanmaken		Must
Debiteur wijzigen		Must
Debiteur verwijderen		Must
Orders opvragen bij een debiteur		Should
Statistieken weergeven bij een debiteur		Could
Facturen opvragen bij een klant		Could
Crediteuren:		
Crediteur opvragen		Must
Nieuwe crediteur aanmaken		Must
Crediteur wijzigen		Must
Crediteur verwijderen		Must
Orders:		
Order opvragen		Must
Nieuwe order aanmaken		Must
Order wijzigen		Must
Order verwijderen		Must
Opdrachtregels binnen een order wijzigen		Should
Opdrachtregels binnen een order aanmaken		Should
Opdrachtregels binnen een order verwijderen		Should
Product:		
Product opvragen		Must
Product verwijderen		Must
Product wijzigen		Must
Nieuw product aanmaken		Must
Factuuroverzicht:		
Weergave lijst per klant		Must
Ophalen factuur		Must
Samenvatting totaal/open facturen		Could
Account:		
Uitloggen		Must
Administratie selecteren		Must
Accountgegevens		Must
Licentiegegevens		Could
Home pagina		
Weergave van een stuk tekst		Should
Aanmaken en wijzigen business objects:		
Validatie van invoerwaarden		Must
Terugkoppeling van fouten bij validatie		Must
Autocomplete op velden waar mogelijk		Should
Onbruikbaar maken van velden die niet gewijzigd kunnen worden		Should

5.2 NIET FUNCTIONELE EISEN

De niet functionele eisen geven het gedrag van het systeem weer. Hieruit kan bepaald worden *hoe* het systeem aan bepaalde functionele eisen moet voldoen. Meestal gaat het om restricties op de invulling van een functionele eis.

In dit project zijn er zowel eisen aan de externe werking van het eindproduct als aan de interne realisatie ervan. De code van de applicatie moet kunnen dienen als voorbeeld en referentiemateriaal. Het is daarbij belangrijk dat de implementatie duidelijk is en ook de sterke kanten van Fabric goed weergeeft. De niet functionele eisen zijn daarom opgedeeld in applicatie en implementatie eisen:

TABEL 2 OPGAVE VAN NIET FUNCTIONELE EISEN

User story	Prioriteit
Applicatie:	
Voor zover mogelijk een consistentie in layout van de schermen	Must
Binnen 2 seconden reageren voor relatief eenvoudige operaties	Should
Vanaf elke pagina kan er uitgelogd worden	Should
Vanaf elke pagina kan de taal gewisseld worden	Should
Vanaf elke pagina kan de account pagina bezocht worden	Should
Gebruiksvriendelijk	Should
Duidelijke foutmeldingen	Should
Code:	
Eenvoudig toe voegen van nieuwe controllers	Must
Voorzien van documentatie	Must
Uitbreidbaar	Should
Flexibel	Should

6 USE CASES

In dit hoofdstuk zijn de belangrijkste use cases uitgewerkt op basis van de functionele eisen uit Tabel 1. In het vorige hoofdstuk staan alle eisen apart genoemd, sommige eisen komen sterk met elkaar overeen en kunnen in één generiek scenario worden weergegeven. Het gaat hier om de use cases voor het aanmaken, wijzigen, opvragen en verwijderen van business objecten. Er zijn een aantal uitzonderingen, hiervoor worden aparte use cases gemaakt.

Een use case bestaat uit een UML diagram waarin de use case schematisch wordt weergegeven en een daarbij horende beschrijving. De use cases geven weer hoe een gebruiker met het systeem om kan gaan; dus hoe er gebruik gemaakt wordt van de functionele eisen. Use cases zijn specifiek aan een bepaalde context, zo zijn er *standaardscenario's*, maar ook *foutscenario's*. De beschrijving bij een use case geeft meer inzicht in de situatie en geeft uitleg. Hierin staat de naam van de case, een samenvatting, actoren, pre- en postcondities, mogelijke uitzonderingen of speciale requirements en een stappenplan.

Alle use cases nemen aan dat er verbinding is met internet.

6.1 AUTHENTICATIE

Scenario 1.1	Authenticatie met geldige inlog gegevens
Prioriteit	Must have
Samenvatting	Kees logt succesvol in met zijn logingegevens
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre conditie	Kees is in bezit van geldige login gegevens
Stappen	1. Kees bezoekt de web applicatie 2. De hoofdpagina wordt getoond en er is nog niet ingelogd 3. Kees gaat naar de login pagina 4. Kees vult naam en wachtwoord in en kiest 'inloggen' 5. De applicatie controleert de gegevens van Kees en geeft aan dat deze correct zijn door kees terug te sturen naar de home page.
Alternatief	4a. Kees kiest ervoor om ingelogd te blijven
Post conditie	Kees is ingelogd en kan nu de beveiligde schermen bezoeken De home pagina wordt getoond Er wordt een authenticatie ticket op de machine van Kees geplaatst in de vorm van een tijdelijke cookie. 4a: Er is een permanente authenticatie ticket geplaatst op de machine van Kees

Scenario 1.2	Authenticatie met ongeldige inlog gegevens
Prioriteit	Must have
Samenvatting	Kees probeert in te loggen met ongeldige gegevens en wordt niet toegelaten.
Type	Ongeldig scenario
Actoren	Kees (eindgebruiker)
Pre condities	
Stappen	1. Kees bezoekt de web applicatie 2. De hoofdpagina wordt getoond en er is nog niet ingelogd 3. Kees gaat naar de login pagina 4. Kees vult naam en wachtwoord in en kiest 'inloggen' 5. De applicatie controleert de gegevens en toont de inlog pagina opnieuw. Op de pagina wordt aangegeven dat de login poging is mislukt.
Alternatief	

Post conditie	De inlog pagina wordt opnieuw aan Kees getoond en geeft aan waarom de login poging is mislukt
Scenario 1.3	Automatisch ingelogd blijven
Prioriteit	Must have
Samenvatting	Kees is direct ingelogd wanneer hij de pagina bezoekt na eerder te hebben aangegeven ingelogd te willen blijven.
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Er bestaat een authenticatie ticket op de machine van Kees
Stappen	1. Kees bezoekt de web applicatie 2. De applicatie controleert de login cookie van Kees 3. De pagina geeft al gelijk aan dat kees is ingelogd
Alternatief	
Post conditie	Kees is ingelogd en kan nu de beveiligde schermen bezoeken

6.2 LIJSTWEERGAVE BUSINESS OBJECTS

Scenario 2.1	Een lijst van business objects weergeven (index pagina)
Prioriteit	Must have
Samenvatting	Kees vraagt een lijst op van de business objects die in het systeem staan
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees bezit de rechten om een lijst business objects op te vragen Kees heeft een administratie gekozen
Stappen	1. Kees kiest het type object waar hij een lijst van wil zien door op de desbetreffende kop te klikken 2. De applicatie haalt de eerste pagina van de lijst van objecten op 3. De applicatie bouwt de pagina op en toont deze 4. Kees krijgt de pagina te zien
Alternatief	
Post conditie	De lijstweergave wordt

Scenario 2.2	Data wordt over meerdere pagina's verspreid
Prioriteit	Should have
Samenvatting	Kees krijgt een beperkte lijst data te zien wanneer de lijst anders erg groot zou zijn, om de volgende reeks uit de lijst te bekijken moet hij naar de volgende pagina navigeren
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees bezit de rechten om een lijst business objects op te vragen
Stappen	1. Kees voert een opdracht die een dataset oplevert welke groter is dan de standaard pagina grootte 2. De applicatie ontvangt het verzoek en bekijkt welke pagina er opgevraagd is 3. De juiste reeks data wordt opgehaald en in een pagina gezet 4. De pagina wordt aan Kees getoond
Alternatief	1a Kees voert zelf in hoeveel data er per pagina weergegeven moet worden 2 Als er geen specifieke pagina wordt opgevraagd wordt de eerste pagina aangenomen
Post conditie	De juiste pagina met data is aan Kees getoond

Scenario 2.3	Filteren van business objects
Prioriteit	Must have
Samenvatting	Kees filtert de lijst business objects door criteria in te voeren, alleen de objecten die voldoen aan de criteria worden getoond
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees bezit de rechten om een lijst business objects op te vragen
Stappen	1. Kees voert criteria in in de daarvoor bestemde velden en kiest 'zoeken' 2. De applicatie ontvangt de ingevoerde criteria en selecteert alléén de business objecten die voldoen aan de criteria 3. De lijst geselecteerde objecten wordt in de pagina gezet en aan Kees getoond
Alternatief	2a De ingevoerde criteria is niet geldig, de zoekpagina wordt getoond waarop de fouten zijn toegelicht
Post conditie	Kees krijgt de gefilterde lijst objecten te zien

Scenario 2.4	Sorteren van waarden in de lijst
Prioriteit	Could have
Samenvatting	Kees voert in op welke kolom hij de lijst wil sorteren
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees bezit de rechten om een lijst business objects op te vragen
Stappen	1. Kees voert in op welke kolom hij wil sorteren door op de kolom header te klikken 2. De applicatie verzamelt de gegevens en sorteert deze 3. De gesorteerde lijst wordt aan Kees getoond 4. Kees klikt nogmaals op dezelfde kolomheader 5. De applicatie verzamelt de gegevens en sorteert deze dit keer andersom 6. Kees geeft aan dat hij naar de volgende pagina wil navigeren 7. De applicatie onthoudt de sortering keuze en past deze toe voordat de tweede pagina wordt getoond
Alternatief	4a Kees klikt op een andere kolomheader, sorteren vindt nu plaats op de standaard manier, niet omgekeerd
Post conditie	De gesorteerde lijst objecten wordt aan Kees getoond.

Scenario 2.5	De voortgang van de zoekopdracht of selectie wordt aan de gebruiker getoond
Prioriteit	Could have
Samenvatting	Kees voert een zoekopdracht uit, terwijl deze bezig is wordt de voortgang van de operatie getoond
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees bezit de rechten om een lijst business objects op te vragen
Stappen	1. Kees voert een zoekopdracht uit met criteria 2. De applicatie verwerkt de zoekopdracht en houdt Kees op de hoogte van de voortgang door hierover informatie te sturen naar de browser 3. De browser toont een nep indicatie van de voortgang 4. Wanneer de operatie klaar is wordt het resultaat aan Kees getoond
Alternatief	
Post conditie	Kees is gedurende de zoekopdracht op de hoogte gehouden van de voortgang en het resultaat is getoond

Scenario 2.6	Dynamisch bijladen van objecten in een lijst
Prioriteit	Could have

Samenvatting	Kees vraagt een lijstweergave van objecten aan en geeft aan dat hij wil dat er meer bijgeladen wordt
Type	Ongeldig scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees bezit de rechten om een lijst business objects op te vragen
Stappen	1. Kees geeft aan dat hij wil dat er dynamisch resultaten worden bijgeladen 2. Kees voert een zoekopdracht uit 3. De applicatie verwerkt de zoekopdracht en toont het resultaat aan Kees 4. Wanneer Kees door de lijst scrollt wordt er automatisch een verzoek om de volgende pagina naar de applicatie verstuurd 5. De applicatie ziet dat dit een speciaal verzoek is en verpakt het resultaat van de operatie op een andere manier 6. De browser plakkt de volgende pagina naadloos aan de huidige pagina
Alternatief	5a Wanneer de laatste pagina is getoond, wordt dit aan Kees duidelijk gemaakt
Post conditie	De lijst is uitgebreid door de tweede pagina eraan vast te plakken

6.3 AANMAKEN BUSINESS OBJECTS

Scenario 3.1	Aanmaken van een nieuw business object
Prioriteit	Must have
Samenvatting	Kees maakt een nieuw business object aan
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om een object aan te maken
Stappen	1. Kees bezoekt de overzicht pagina (lijst opgave) van het desbetreffende business object 2. Kees kiest "nieuw" 3. De aanmaak pagina wordt getoond 4. Kees vult het formulier in en kiest "aanmaken" 5. De applicatie verwerkt het aanmaken van het object 6. De details pagina van het nieuwe object wordt getoond
Alternatief	
Post conditie	Het nieuwe object is aangemaakt en bevindt zich in de database Kees is geïnformeerd over het slagen van de operatie

Scenario 3.2	Aanmaken van een nieuw business object met ongeldige waarden
Prioriteit	Must have
Samenvatting	Kees vult bij het aanmaken van een nieuw business object een of meer ongeldige waarden in.
Type	Ongeldig scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om een object aan te maken
Stappen	1. Kees bezoekt de overzicht pagina (lijst opgave) van het desbetreffende business object 2. Kees kiest "nieuw" 3. De aanmaak pagina wordt getoond 4. Kees vult het formulier onjuist in en kiest "aanmaken" 5. De applicatie verwerkt het aanmaken van het object 6. De aanmaak pagina wordt opnieuw getoond waarbij de verkeerd ingevulde velden worden toegelicht
Alternatief	
Post conditie	De aanmaak pagina die aan Kees wordt getoond is voorzien van toelichtingen bij de ongeldige waarden Kees is geïnformeerd over het niet slagen van de operatie

6.4 OPVRAGEN BUSINESS OBJECTS

Scenario 4.1	Opvragen van de details van een business object
Prioriteit	Must have
Samenvatting	Kees vraagt de details pagina van een bestaand business object op
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om een object te bekijken
Stappen	1. Kees bezoekt de overzicht pagina (lijst opgave) van het desbetreffende business object 2. Kees selecteert een item uit de lijst 3. De applicatie zoekt het gekozen item op en bouwt de pagina op 4. Het gekozen item wordt getoond op een nieuwe pagina
Alternatief	2a Kees voert het id van het business object direct in de URL in
Post conditie	De details pagina van het gekozen business object wordt aan Kees getoond

Scenario 4.2	Opvragen van de details van een niet bestaand business object
Prioriteit	Must have
Samenvatting	Kees vraagt de details pagina op van een business object wat niet bestaat
Type	Ongeldig scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om een object te bekijken
Stappen	1. Kees voert het id van een business object in de URL in 2. De applicatie probeert het gespecificeerde object op te halen 3. Het object wordt niet gevonden 4. Kees krijgt een pagina te zien met een foutmelding
Alternatief	
Post conditie	Kees is geïnformeerd over het niet slagen van de operatie

6.5 WIJZIGEN BUSINESS OBJECTS

Scenario 5.1	Wijzigen van een of meer velden van een business object
Prioriteit	Must have
Samenvatting	Kees vraagt een business object op en wijzigt velden
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Het opgevraagde business object bestaat Kees is ingelogd en bezit de rechten om een object te wijzigen
Stappen	1. Kees vraagt het business object op 2. Kees wijzigt een of meer velden van het business object en kiest voor 'opslaan' 3. De applicatie controleert of alle ingevulde velden valide zijn 4. De applicatie past de wijzigingen toe en slaat het object op 5. De details pagina van het business object wordt aan kees getoond
Alternatief	
Post conditie	De ingevoerde wijzigingen van het business object zijn opgeslagen Kees is geïnformeerd over het slagen van de operatie

Scenario 5.2	Ongeldige wijzigingen uitvoeren op een business object
Prioriteit	Must have
Samenvatting	Kees geeft een of meer velden van het business object een andere waarde welke niet geldig is
Type	Ongeldig scenario
Actoren	Kees (eindgebruiker)
Pre condities	Het opgevraagde business object bestaat Kees is ingelogd en bezit de rechten om een object te wijzigen
Stappen	1. Kees vraagt het business object op 2. Kees wijzigt een of meer velden van het business object en kiest voor 'opslaan' 3. De applicatie controleert of alle ingevulde velden valide zijn 4. De applicatie ontdekt dat er ongeldige waarden zijn opgegeven 5. De wijzigingspagina wordt aan Kees getoond met validatie annotaties bij ongeldig ingevoerde waarden
Alternatief	
Post conditie	De wijzigingspagina is voorzien van toelichtingen bij de ongeldige waarden Kees is geïnformeerd over het niet slagen van de operatie

6.6 VERWIJDEREN BUSINESS OBJECTS

Scenario 6.1	Verwijderen van een business object
Prioriteit	Must have
Samenvatting	Kees verwijdert een business object
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om een object te verwijderen Het te verwijderen business object bestaat Kees heeft de details pagina van het business object opgevraagd (case 4.1)
Stappen	1. Kees kiest voor 'verwijderen' 2. Een bevestiging pagina wordt getoond 3. Kees kiest opnieuw voor 'verwijderen' 4. De applicatie zoekt het te verwijderen object op 5. De applicatie verwijdert het object 6. De overzicht (index) pagina wordt getoond
Alternatief	1a Kees bezoek direct de bevestigingspagina door de url zelf op te geven 3a Kees kiest voor 'niet verwijderen', de detail pagina wordt getoond
Post conditie	Het business object is verwijderd uit de database Kees is geïnformeerd over het slagen van de operatie

6.7 DEBITEUR

Scenario 7.1	Verwijderen van een debiteur
Prioriteit	Must have
Samenvatting	Kees probeert een debiteur te verwijderen
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om een debiteur te verwijderen Kees heeft de details pagina van de debiteur opgevraagd (case 4.1)
Stappen	1. Kees kiest voor 'verwijderen'

	- De applicatie verwerkt het verzoek en geeft een bevestiging pagina terug waarop is aangegeven dat alle orders die bij de debiteur horen ook verwijderd zullen worden. - Kees bevestigt de operatie - De applicatie zoekt alle orders op die bij de opgegeven debiteur horen - De applicatie verwijdert eerste al deze orders en daarna de debiteur zelf - De index pagina wordt aan Kees getoond
Alternatief	2a Kees kiest om de verwijder operatie af te breken, de detail pagina wordt hierop getoond
Post conditie	De debiteur en alle bijbehorende orders zijn verwijderd uit de database Kees is geïnformeerd over het slagen van de operatie

Scenario 7.2	Opvragen van orders bij een debiteur
Prioriteit	Could have
Samenvatting	Kees vraagt alle orders op die een bepaalde debiteur heeft geplaatst
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om debiteuren te bekijken en een lijst orders op te vragen
Stappen	- Kees kiest voor “orders bij deze debiteur” - De applicatie voert een zoekopdracht uit met het ingegeven debiteur nummer - De applicatie bouwt de pagina op en toont deze aan Kees
Alternatief	
Post conditie	De lijst met orders die horen bij de gekozen debiteur is aan Kees getoond

Scenario 7.3	Opvragen van statistieken bij debiteur
Prioriteit	Could have
Samenvatting	Kees vraagt de verkoopstatistieken op die bij een debiteur horen
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om debiteuren te bekijken en statistieken van een debiteur te bekijken Kees bevindt zich op de detail pagina
Stappen	- Kees kiest voor ‘statistieken’ - De applicatie zoekt de statistieken op bij het ingegeven debiteur id - De statistieken pagina wordt opgebouwd en getoond
Alternatief	1a Kees voert zelf de url in voor de statistieken pagina
Post conditie	De statistiekenpagina is aan Kees getoond

De use case “opvragen facturen” verloopt op dezelfde manier als de bovenstaande use case

6.8 ALGEMENE FEATURES

Scenario 8.1	Wisselen van weergavetaal
Prioriteit	Could have
Samenvatting	Kees verandert de taal van de applicatie
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	
Stappen	- Kees kiest een andere taal door te klikken op de taalselectie in het scherm - De applicatie wijzigt de weergave taal voor Kees
Alternatief	

Post conditie	De taal van de applicatie is gewijzigd naar de door Kees geselecteerde taal
----------------------	---

Scenario 8.2	Opvragen van een pagina zonder te zijn ingelogd
Prioriteit	Must have
Samenvatting	Kees probeert een pagina op te vragen zonder ingelogd te zijn
Type	Ongeldig scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees heeft login gegevens
Stappen	1. Kees kiest een beschermde pagina 2. De applicatie controleert welke rechten er zijn vereist om de opgevraagde pagina te bekijken 3. De applicatie controleert de identiteit van de gebruiker 4. De applicatie constateert dat Kees niet is ingelogd 5. De login pagina wordt getoond in plaats van de opgevraagde pagina 6. Kees logt in met zijn gegevens 7. De applicatie controleert de gegevens (case 1.1) van Kees 8. De origineel opgevraagde pagina wordt getoond
Alternatief	8a De ingevoerde gegevens zijn onjuist, de login pagina wordt opnieuw getoond
Post conditie	Kees is ingelogd in het systeem en de opgevraagde pagina wordt getoond

Scenario 8.3	Opvragen van een beschermde pagina
Prioriteit	Must have
Samenvatting	Kees vraagt een pagina op waar rechten voor nodig zijn
Type	Ongeldig scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd Kees bezit niet de rechten om de pagina te bekijken
Stappen	1. Kees kiest een pagina waartoe hij geen recht heeft 2. De applicatie controleert welke rechten er vereist zijn om de opgevraagde pagina te bekijken 3. De applicatie controleert de rechten van Kees 4. De applicatie constateert dat Kees niet de benodigde rechten bezit en toont een fout pagina
Alternatief	
Post conditie	De foutpagina is aan Kees getoond

Scenario 8.4	Produceren van log bestanden
Prioriteit	Should have
Samenvatting	Jan kan details van de door de applicatie uitgevoerde operaties terugvinden in log bestanden
Type	Ongeldig scenario
Actoren	Jan (systeembeheerder)
Pre condities	Het loggen is ingeschakeld
Stappen	1. Een gebruiker voert een operatie uit 2. De applicatie verwerkt de operatie 3. De applicatie maakt een log regel aan waarin enkele details van de operatie worden vastgelegd
Alternatief	
Post conditie	Informatie over de uitgevoerde operatie is terug te vinden in het log bestand

NB: Het aanmaken opvragen, wijzigen en verwijderen van opdrachtregels verloopt op dezelfde manier als het werken met andere business objecten, alleen vanuit het order scherm.

6.9 AANMAKEN BUSINESS OBJECTS

Scenario 9.1	Autocomplete bij invoervelden
Prioriteit	Should have
Samenvatting	Kees wordt bij het invullen van velden geholpen door het system d.m.v. autocomplete
Type	Standaard scenario
Actoren	Kees (eindgebruiker)
Pre condities	Kees is ingelogd en bezit de rechten om business objects aan te maken
Stappen	1. Kees selecteert een invoerveld 2. Er wordt een bericht naar de applicatie gestuurd waarin wordt gevraagd om autocomplete suggesties voor een bepaald veld 3. De applicatie vraagt de mogelijke waarden op voor het opgegeven veld 4. De applicatie selecteert alléén de waarden die lijken op de op dit moment ingevoerde waarde 5. De applicatie stuurt de geselecteerde waarden terug en toont deze in een autocomplete dropdown menu 6. Kees selecteert één van de gesuggereerde waarden
Alternatief	5a Kees voert nog meer tekens in waardoor het aantal suggesties verder verkleind wordt 6a Kees vult zelf de volledige tekst in en de suggesties verdwijnen
Post conditie	Kees heeft een waarde ingevoerd in het veld

7 BEVINDINGEN RAPPORT

Het bevindingen rapport moet inzicht geven in de kwaliteit van Fabric. Hiermee wordt bedoeld dat de problemen en mankementen in kaart worden gebracht. Waar mogelijk wordt er een oplossing voor het probleem voorgesteld.

De op- en aanmerkingen op Fabric zijn in vier categorieën onderverdeeld:

- **Functionele mankementen:** Hetgeen wat niet, of niet altijd goed werkt.
- **Niet-functionele mankementen:** Onderdelen die wel werken, maar niet intuïtief zijn, of onhandig.
- **Suggesties:** Onderdelen of functionaliteit die lijkt te missen.
- **Algemeen:** Opmerkingen die niet in de vorige drie categorieën passen en vaak niet aantoonbaar verkeerd zijn, maar waar verder onderzoek geboden is.

Elke opmerking heeft een prioriteit en een status. De prioriteit geeft aan hoe de belangrijkheid van het probleem wordt geschat, de prioriteit loopt van 1 (laag) tot 10 (kritiek). Sommige problemen zijn ten tijde van de oplevering van dit rapport al verholpen. Dit zal aan de status af te lezen zijn. De mogelijke oplossingen staan in volgorde van preferentie, waarbij de eerste oplossing het meest geprefereerd wordt.

8 RISICOMANAGEMENT

8.1 RISICOANALYSE

Een risicoanalyse geeft weer welke risico's er bestaan met een schatting van de kans dat het risico optreedt en een schatting van de impact. Aan de hand hiervan kan worden bepaald welke maatregelen er genomen moeten worden om het risico te beperken. Als de risicoanalyse slecht uitvalt, moet het project aangepast worden, of kan het zelfs helemaal niet doorgaan. Aan elk risico wordt een kans en een impact parameter gekoppeld, zie Tabel 3. Beiden uit een schaal van [1-10], deze worden met elkaar vermenigvuldigd en dit getal stelt het risicofactor voor.

TABEL 3 PRIORITERING RISICO'S

Risicofactor	Prioriteit
1-15	Zeer laag
16-30	Laag
31-45	Normaal
46-60	Hoger dan normaal
61-75	Hoog
76-100	Kritiek

Er zijn een aantal risico's geïdentificeerd, in Tabel 4 staat per risico een toelichting aangegeven.

TABEL 4 RISICOFACTOREN

Afwezigheid	
Omschrijving	De opdracht loopt vertraging op doordat de afstudeerder ziek is of om een andere reden langdurig afwezig is en daardoor niet kan werken.
Prioritering	Kans: 1, Impact: 5, Risicofactor: 5 Zeer laag
Toelichting	Ik ben al jaren niet meer ziek geweest, mocht er ziekte optreden is dit waarschijnlijk van korte duur.
Preventie	n.v.t.
Repressie	Bij langdurige afwezigheid kan er verlenging van de afstudeeropdracht worden aangevraagd.

Onjuiste sprint planning	
Omschrijving	De sprint indeling is niet goed waardoor er minder controle is over het project
Prioritering	Kans: 7, Impact: 5, Risicofactor: 35 Normaal
Toelichting	Het indelen van sprints zal moeilijk worden. Er is geen historische data waarmee te bepalen is hoeveel werk ik per sprint kan doen. Daarnaast is ook de complexiteit van de user stories niet volledig van te voren te schatten.
Preventie	Ruimte overlaten tijdens de planning voor het uitlopen van sprints, onderzoek doen naar de waarschijnlijke duur van een user story.
Repressie	Indeling aanpassen, user stories doorschuiven naar volgende sprints, maybe-haves laten vallen. Sprintduur verlengen.

Onervarenheid	
Omschrijving	Er wordt gebruik gemaakt van een aantal nieuwe technieken waar ik nog niet mee

	bekend ben, deze onervarenheid kan voor vertraging zorgen.
Prioritering	Kans: 10, Impact: 4, Risicofactor: 40 Normaal
Toelichting	De Scrum methode en de Fabric, CSLA en ASP.NET MVC frameworks zijn bij mij nog niet volledig bekend.
Preventie	Onderzoek doen naar alle onbekende technieken en opzoeken wat de 'best practices' zijn.
Repressie	Extra tijd steken in leren nieuwe techniek, zo nodig planning opschuiven.

Onduidelijke scope

Omschrijving	De scope van het project is niet goed vastgesteld.
Prioritering	Kans: 7, Impact: 7, Risicofactor: 49 Hoger dan normaal
Toelichting	Eenzijds bestaat het project uit het 'bewijzen' van het Fabric framework, anderzijds moet er wel een volledig werkende web applicatie opgeleverd worden. Er moet hiertussen een goede balans worden gevonden.
Preventie	Vroegtijdig met de opdrachtgever afspreken hoe de opdracht wordt ingedeeld.
Repressie	Opdracht herdefiniëren, disciplines beter afbakenen.

Coördinatieproblemen

Omschrijving	Het coördineren van alle werkzaamheden verloopt slecht.
Prioritering	Kans: 4, Impact: 7, Risicofactor: 28 Laag
Toelichting	Het project bestaat uit veel verschillende werkzaamheden. Deze moeten allemaal bijgehouden worden, als er taken slecht verlopen of achterlopen, kunnen andere taken ook verstoord worden.
Preventie	Taken zo goed mogelijk S.M.A.R.T. opstellen, proberen taken zo veel mogelijk onafhankelijk van elkaar uitvoerbaar te maken.
Repressie	Planning aanpassen, taken die slecht verlopen laten controleren door opdrachtgever.

De onervarenheid en de onduidelijke scope vormen de grootste risico's. De onervarenheid kan niet worden voorkomen, er wordt met dit risico rekening gehouden door veel informatie in te winnen voordat er met een nieuwe techniek wordt gewerkt. Dit wordt gerealiseerd door bij aanvang van het project te verdiepen in *design patterns* en *best practices* die van toepassing zijn op het ontwerpen van een MVC web applicatie.

Het tweede belangrijke risico wordt bestreden door zo veel mogelijk eisen van te voren op te stellen en duidelijk af te bakenen. De scope van alle eisen en functionaliteit van het programma moet duidelijk zijn voordat de functionaliteit wordt ontworpen en geïmplementeerd.

UNIT4 SOFTWARE B.V.

Bevindingenrapport Fabric

In opdracht van UNIT4 Software B.V.

Raymond Meerburg

9/22/2012

Fabric is vanuit verschillende perspectieven onderzocht. In dit rapport wordt een advies aangeboden en worden suggesties gegeven die de kwaliteit van Fabric kunnen verbeteren.

1 INHOUD

1	Inhoud	1
2	Inleiding.....	2
2.1	Achtergrond.....	2
2.2	Probleemstelling.....	2
2.3	Doelstelling.....	2
2.4	Leeswijzer	2
3	Functionele mankementen	4
4	Niet-functionele mankementen	7
5	Mogelijke verbeteringen	9
6	Mogelijk verder onderzoek nodig	12
7	Aanbevelingen en conclusies	14

2 INLEIDING

2.1 ACHTERGROND

Dit rapport is geschreven in opdracht van UNIT4 Software B.V.

“Fabric” is de naam van het nieuwe applicatie framework wat door UNIT4 wordt ontwikkeld als opvolger van de verouderde Universal Business Connector (UBC). Dit framework wordt aan partners van UNIT4 aangeboden die daarmee zelf applicaties kunnen bouwen die samenwerken met het hoofdproduct van UNIT4: Multivers.

Fabric is al anderhalf jaar in ontwikkeling, voor het einde van dit jaar moet de eerste versie beschikbaar worden gesteld. UNIT4 wil graag een kwaliteits- en paraatheidscontrole uitvoeren op Fabric voordat het wordt vrijgegeven.

Dit rapport is onderdeel van een grotere opdracht waarin een concept applicatie is gebouwd op Fabric. Het doel hiervan is om een presenteerbaar concept te bouwen die de mogelijkheden ten toon stelt en partners geïnteresseerd krijgt in ontwikkeling met het nieuwe framework. Daarnaast moet de broncode van de applicatie kunnen dienen als voorbeeld en referentie materiaal. De insteek van de opdracht is geweest om Fabric te benaderen vanuit de ogen van een partner. Alle problemen die zich hebben voorgedaan zijn vastgelegd en komen in dit rapport naar voren. Er wordt waar mogelijk advies gegeven over een mogelijke oplossing. Met dit rapport kan UNIT4 de kwaliteit van Fabric verbeteren voordat het wordt vrijgegeven aan partners.

2.2 PROBLEEMSTELLING

De kwaliteit van de verschillende onderdelen van Fabric wordt bewaakt door ruim duizend unit tests. Toch zegt dit niet genoeg over de algemene kwaliteit van het geheel. Er is nu niet voldoende zekerheid en vertrouwen dat Fabric (bijna) klaar is om gebruikt te worden.

Het is belangrijk om zelf eerst het framework uit te testen door het te gebruiken voor een daadwerkelijke applicatie. Alle problemen die zich tijdens deze ontwikkeling voordoen worden gedocumenteerd in dit verslag.

2.3 DOELSTELLING

De doelstelling van het rapport is om ontwikkelaars die aan Fabric werken in staat te stellen om de kwaliteit van het framework te verhogen door de aangegeven problemen aan te pakken. Ook krijgen zij inzicht in hoe een gebruiker van Fabric mogelijk met het systeem zal omgaan zo kunnen zij zich beter inleven in de verwachtingen van gebruikers.

2.4 LEESWIJZER

Dit rapport is bedoeld voor ontwikkelaars die de genoemde problemen nader gaan bekijken. Ook andere geïnteresseerden kunnen dit rapport gebruiken om inzicht te krijgen in de huidige stand van zaken binnen Fabric.

De op- en aanmerkingen op Fabric zijn in vier categorieën onderverdeeld:

- **Functionele mankementen:** Hetgeen wat niet, of niet altijd goed werkt.
- **Niet-functionele mankementen:** Onderdelen die wel werken, maar niet intuïtief zijn, of onhandig.
- **Suggesties:** Onderdelen of functionaliteit die lijkt te missen.
- **Algemeen:** Opmerkingen die niet in de vorige drie categorieën passen en vaak niet aantoonbaar verkeerd zijn, maar waar verder onderzoek geboden is.

Elke opmerking heeft een prioriteit en een status. De prioriteit geeft aan hoe de belangrijkheid van het probleem wordt geschat, de prioriteit loopt van 1 (laag) tot 10 (kritiek). Sommige problemen zijn ten tijde van de oplevering van dit rapport al verholpen. Dit zal aan de status af te lezen zijn. De mogelijke oplossingen staan in volgorde van preferentie, waarbij de eerste oplossing het meest geprefereerd wordt.

3 FUNCTIONELE MANKEMENTEN

In dit hoofdstuk worden de functionele mankementen in kaart gebracht. Het gaat hier om gedrag en resultaten van Fabric die aantoonbaar fout zijn. Dit kan een verkeerde output zijn, een interne foutmelding of andere problemen die zich voordoen zonder dat de ontwikkelaar die het gebruikt schuld heeft. Onder de functionele mankementen vallen ook tekortkomingen van Fabric.

Naam	1.1 Cache invalidation
Prioriteit	8
Status	Open
Beschrijving	De waarden in een Name-Value lists worden eenmaal uit de database geladen en blijven daarna in een cache zitten. Wanneer de onderliggende database wijzigt (een andere administratie wordt geselecteerd) blijven de waarden in de NVL caches ongemoeid.
(Mogelijke) oorzaak	De NVL's controleren zelf niet om de huidige administratie gelijk is aan de administratie van de gecacheerde waarden. Ook worden de NVL's niet op een andere manier op de hoogte gebracht van een administratie wijziging.
Mogelijke oplossing	• Laat de NVL's door middel van een event weten dat de cache niet meer geldig en dat ze opnieuw de database moeten uitlezen. • Laat de NVL's bij elke aanvraag controleren of de administratie is gewijzigd. • Maak het mogelijk om alle NVL caches op een centrale plek te invalideren. (moeilijk gemaakt door CRTP)

Naam	1.2 NVL laden
Prioriteit	4
Status	Open
Beschrijving	Als een NVL voor de eerste keer opgevraagd wordt kan het antwoord lang op zich laten wachten indien er een grote database achter staat.
(Mogelijke) oorzaak	De NVL's maken gebruik van lazy loading, dit levert een grote vertraging op wanneer er zeer veel waarden uit de database in de NVL geladen moeten worden.
Mogelijke oplossing	• Maak het mogelijk om NVL's te <i>preloaden</i>, de NVL wordt dan door een achtergrondproces geladen aan het begin van de applicatie. • Wijzig de Fetch functie zodat deze lambda's kan accepteren, geef de lambda door aan de DAL en geeft het resultaat gelijk terug aan de caller. Laadt daarna de volledige lijst in een achtergrondproces.

Naam	1.3 Context gevoelige objecten niet altijd juist geïmplementeerd
Prioriteit	7
Status	Open
Beschrijving	De inhoud van een NVL is contextgevoelig, deze is voor iedere database anders. De NVL's geven dezelfde waarden terug ook al is de administratie gewijzigd.
(Mogelijke) oorzaak	Op dit moment is van iedere NVL maar één instantie, deze bevat de gegevens van de eerste database waar gegevens uit zijn geladen.
Mogelijke oplossing	• Bij de aanroep van een NVL moet gekeken worden welke database de aanvrager open heeft staan.

Naam	1.4 openAdministration functie implementeert niet het verwachte gedrag.
Prioriteit	3
Status	Open
Beschrijving	• De openAdministration functie geeft altijd 'true' terug, ook als de administratie niet

	geopend kan worden.
(Mogelijke) oorzaak	De functie controleert niet of de administratie geldig is en geopend kan worden, er wordt alleen een variabele gezet.
Mogelijke oplossing	- Controleer of de administratie in de database geopend kan worden en geef het antwoord hierop terug aan de caller. - Controleer in ieder geval of de naam van de administratie geldig is.

Naam	1.5 Last modified veld van een Product
Prioriteit	2
Status	Open
Beschrijving	Een product moet onthouden wanneer deze voor het laatst is gewijzigd. Dit werkt niet.
(Mogelijke) oorzaak	n.v.t.
Mogelijke oplossing	Controleer waar er data aan het veld wordt toegewezen en waarom dit niet goed verloopt.

Naam	1.6 Lange wachttijd bij eerste operatie op een Order
Prioriteit	4
Status	Open
Beschrijving	Het uitvoeren van operaties met de Order klasse duurt de eerste keer erg lang mits er een grote database is aangesloten.
(Mogelijke) oorzaak	Het controleren van alle business rules van een Order heeft tot gevolg dat de CustomerNVL geladen moet worden. Dit duurt een lange tijd.
Mogelijke oplossing	- Indien de CustomerNVL alleen geladen hoeft te worden om te controleren of het debiteurnummer bij een Order juist is, controleer dan eerst of het nummer is gewijzigd, zo niet hoeft de controle niet worden uitgevoerd. - Zorg ervoor dat de CustomerNVL al is geladen voordat er operaties met een order worden uitgevoerd.

Naam	1.7 Csla controller controleert business rules op een onverwacht moment
Prioriteit	7
Status	Open
Beschrijving	Omdat Fabric gebouwd is op Csla spreekt het voor zich om de Csla controller te gebruiken als base class voor de controllers binnen het project. Het wijzigen van business objects via deze controller gaat soms verkeerd.
(Mogelijke) oorzaak	De Csla controller controleert de business rules van een business object zodra deze wordt gewijzigd. Binnen Fabric is er ten minste één rule die voor problemen zorgt wanneer deze op dit moment wordt gecontroleerd.
Mogelijke oplossing	- Pas de business rule aan en zorg ervoor dat rules in de toekomst alleen de staat van een object controleren, en geen side effects hebben. - Implementeer een Fabric controller die de business rules niet zelf controleert.

Naam	1.8 Onjuiste validatie op Business number veld bij een Supplier
Prioriteit	1
Status	Open
Beschrijving	De validatie op het veld 'business number' van de Supplier klasse is onjuist.
(Mogelijke) oorzaak	n.v.t.
Mogelijke oplossing	Verbeter de desbetreffende rule.

Naam	1.9 Prijsberekening van een Order is onjuist
Prioriteit	8
Status	Opgelost
Beschrijving	De prijzen die berekend worden door de Order klasse zijn onjuist
(Mogelijke) oorzaak	In de prijsberekening methode staat een fout m.b.t. het berekenen van korting bij een prijs.
Oplossing	Pas de methode aan zodat deze fout niet meer voorkomt.

Naam	1.10 SalesInvoiceInfo kan niet worden aangemaakt
Prioriteit	8
Status	Opgelost
Beschrijving	Er kan geen instantie van het SalesInvoiceInfo object worden aangemaakt.
(Mogelijke) oorzaak	De factory die het object moet maken bezit niet de juiste overloads en kan het object daardoor niet aanmaken.
Oplossing	Voeg de benodigde overloads toe aan de SalesInvoiceInfoFactory.

Naam	1.11 Inloggen met sessie
Prioriteit	7
Status	Open
Beschrijving	Er kan nog niet direct worden ingelogd met een sessie id, dit heeft tot gevolg dat alle context (zoals de gekozen administratie) bij ieder verzoek opnieuw ingesteld moet worden.
(Mogelijke) oorzaak	n.v.t.
Mogelijke oplossing	Implementeer een alternatieve login functie waarbij een sessie id wordt opgegeven in plaats van een naam en wachtwoord.

Naam	1.12 Command om CustomerId te genereren bevat fouten
Prioriteit	4
Status	Open
Beschrijving	Wanneer er geprobeerd wordt om een id te genereren en 'autoNumberCustomerId' is false wordt er een niet relevante exception gegeven.
(Mogelijke) oorzaak	Onbekend
Mogelijke oplossing	Controleer of alle benodigdheden in orde zijn vóórdat de rule wordt uitgevoerd.

Naam	1.13 Wissen van velden onmogelijk
Prioriteit	8
Status	Reeds opgelost
Beschrijving	Het leegmaken van een veld van een business object waar daarvoor een waarde heeft ingestaan is onmogelijk.
(Mogelijke) oorzaak	Wanneer Fabric ziet dat de nieuwe waarde van een veld leeg is, wordt deze niet toegewezen.
Mogelijke oplossing	Controleer de functies die verantwoordelijk zijn voor het toewijzen van waarden aan velden.

4 NIET-FUNCTIONELE MANKEMENTEN

Een niet-functioneel mankement is een constructie die wel naar behoren werkt, maar op een voor de gebruiker onhandige manier. Dit kan bijvoorbeeld zijn dat er bepaalde dingen vaak herhaal moeten worden, of veel gebruikte operaties die onnodig complex lijken te zijn. Het resultaat is in dit geval wel juist, maar de manier waarop dit resultaat bereikt wordt is moeilijk.

Naam	2.1 Het tonen van Customer en Suppliers kan niet op een abstracte manier worden gedaan.
Prioriteit	3
Status	Open
Beschrijving	Een Customer en Supplier delen veel functionaliteit. Zeker bij een simpel overzicht scherm is het wenselijk om het tonen van hun gegevens op een generieke manier af te handelen. Dat wordt nu erg moeilijk gemaakt. Het zou ideaal zijn om een scherm te maken die de informatie in een Organization kan weergeven. De debiteur en crediteur schermen kunnen deze functionaliteit beiden herbruiken.
(Mogelijke) oorzaak	Beide klassen leiden af van een OrganizationBase baseclass, maar omdat er gebruik wordt gemaakt van CRTP is het onmogelijk om een reference naar een OrganizationBase te maken die compatibel is met zowel een Customer als een Supplier. Dit betekent dat er code gedupliceerd moet worden of er moet een tussenlaagje gebouwd worden.
Mogelijke oplossing	• Oplossing: Onderzoek het voordeel van CRTP en weeg af of dit de extra complexiteit waard is. • Omweg: Voeg een expliciete cast toe van Customer en Supplier naar een Organization • Waarschuw dat een dergelijke aanpak niet zal werken.

Naam	2.2 Selectie van orders zonder boekjaar te specificeren
Prioriteit	2
Status	Open
Beschrijving	Bij het selecteren van OrderInfoList kan er een boekjaar worden meegegeven als criterium. Wanneer dit boekjaar niet wordt gespecificeerd gaat Fabric uit van het huidige boekjaar. Het verwachte gedrag zou zijn dat alle orders, ongeacht boekjaar, worden teruggegeven.
(Mogelijke) oorzaak	n.v.t. (Dit is het ontworpen gedrag.)
Mogelijke oplossing	• Onderzoek welk gedrag wenselijk is en pas de functie zo nodig aan. • Plaats een opmerking bij de functie die dit gedrag aangeeft.

Naam	2.3 Inconsistentie in type van CustomerGroupNVL
Prioriteit	3
Status	Open
Beschrijving	De CustomerGroupNVL heeft een integer als key in plaats van een string, zoals vrijwel alle andere NVL's.
(Mogelijke) oorzaak	n.v.t.
Mogelijke oplossing	• Onderzoek of het mogelijk is om hier ook een string te gebruiken.

Naam	2.4 Inefficiëntie bij invalidatie van NVL's
-------------	---

Prioriteit	4
Status	Open
Beschrijving	Wanneer de inhoud van een NVL moet veranderen wordt de gehele NVL verwijderd en opnieuw uitgelezen uit de database.
(Mogelijke) oorzaak	n.v.t.
Mogelijke oplossing	Onderzoek of het mogelijk is om het ontwerp van NVL's te herzien zodat het toevoegen of verwijderen van waarden wordt ondersteund.

Naam	2.5 Onjuist geïmplementeerde business rule
Prioriteit	6
Status	Open
Beschrijving	De business rule die hoort bij het 'OrderSurchargeVatCode' veld van een order controleert of de code bestaat in de VatCodeNVL in plaats van de OrderSurchargeVatCodeNVL.
(Mogelijke) oorzaak	n.v.t.
Mogelijke oplossing	Controleer of dit de juiste implementatie van die business rule is.

Naam	2.6 Inconsistentie in namespace van AccountManager
Prioriteit	3
Status	Open
Beschrijving	De AccountManager klasse bevindt zich niet in de verwachte namespace.
(Mogelijke) oorzaak	n.v.t.
Mogelijke oplossing	Wijzig de namespace van de AccountManager klasse

Naam	2. Inconsistentie in naamgeving bij CustomerInfoList
Prioriteit	3
Status	Opgelost
Beschrijving	Alle business objects kennen een 'Fetch' methode waarmee het object wordt opgehaald. Bij de CustomerInfoList klasse heeft deze functie een andere naam: 'GetCustomerInfoList'.
(Mogelijke) oorzaak	n.v.t.
Mogelijke oplossing	Pas de naamgeving aan zodat deze in lijn is met de rest van Fabric.

5 MOGELIJKE VERBETERINGEN

Naast het oplossen van mankementen kan de kwaliteit van Fabric ook worden verbeterd door een aantal functionele wijzigingen door te voeren. In dit hoofdstuk staan verscheidene aanpassingen die het gebruik van Fabric makkelijker kunnen maken. De beschrijving van een mogelijke verbetering volgt een iets anders formaat dan in de vorige twee hoofdstukken is gebruikt.

Naam	3.1 Sorteren van data in Fabric
Prioriteit	2
Status	Open
Beschrijving	De lijsten met data die aan Fabric worden opgevraagd zijn niet gesorteerd.
Toelichting	Het sorteren van deze lijst kan gemakkelijk zelf worden toegepast maar wellicht kan dit op een efficiëntere manier worden gedaan door de data door de database te laten sorteren.
Mogelijke implementatie	Geef Fetch functies additionele parameters die aangeven op veld gesorteerd moet worden (via een statically typed reference) en welke richting er op gesorteerd moet worden.

Naam	3.2 Paging van data
Prioriteit	8
Status	Reeds geïmplementeerd
Beschrijving	Sta gebruikers van Fabric toe om op te geven welke 'pagina' met data er teruggegeven moet worden.
Toelichting	Een web applicatie moet veel vaker data ophalen dan een desktop applicatie, het is inefficiënt om meer data op te halen dan nodig is, zoals op dit moment gebeurt.
Mogelijke implementatie	Voeg parameters toe aan de Fetch functies waarin staat welke pagina er opgevraagd moet worden en hoe groot deze pagina is.

Naam	3.3 Opnemen van een CustomerId veld in een OrderInfo
Prioriteit	6
Status	Reeds geïmplementeerd
Beschrijving	Een CustomerInfo geeft niet voldoende informatie over een order. Het id van de eigenaar (een Customer) ontbreekt.
Toelichting	
Mogelijke implementatie	Voeg het CustomerId veld ook toe aan de OrderInfo

Naam	3.4 Genereren van id's voor business objects
Prioriteit	5
Status	Open
Beschrijving	Fabric biedt voor sommige business objects al de mogelijkheid om het id automatisch te laten genereren (commands). De gebruiker heeft erg weinig inbreng in hoe deze id's worden gegenereerd.
Toelichting	Voor sommige gebruikers kan het van belang zijn dat de id's van objecten voldoen aan een bepaalde logica, de huidige commands staan bijna geen aanpassingen in de logica toe
Mogelijke implementatie	- Wijzig de commands zodat er meer inbreng mogelijk is in het genereer proces. - Sta ontwikkelaars toe om zelf een command te ontwikkelen die door Fabric gebruikt wordt om de id's te genereren.

Naam	3.5 Onverwachte conversie bij OrderInfoList
Prioriteit	6
Status	Open
Beschrijving	De OrderInfoList kent een property Open die alleen de 'open' orders bevat, deze open orders staan in een IEnumerable<OrderInfo> in plaats van een OrderInfoList
Toelichting	Doordat het resultaat in een ander formaat wordt terug gegeven is een constructie zoals "orders = orders.Open" niet mogelijk
Mogelijke implementatie	• Verander het type van de 'Open' property in een OrderInfoList.

Naam	3.6 Informatie over aantal resultaten
Prioriteit	7
Status	Reeds geïmplementeerd
Beschrijving	Wanneer er lijsten met business objects geladen worden is er geen informatie beschikbaar over de hoeveelheid objects de lijst bevat.
Toelichting	Het is voor gebruikers interessant om te weten hoeveel resultaten de ingegeven zoekopdracht heeft opgeleverd.
Mogelijke implementatie	• Voer naast een "Select *" ook een "Select Count(*)" uit • Tel het aantal elementen in de resultaatset (zeer inefficiënt)

Naam	3.7 Exists rule mist op productId binnen een OrderLine
Prioriteit	7
Status	Open
Beschrijving	Bij het aanmaken van een OrderLine wordt er niet gecontroleerd of het ingegeven productId wel bestaat.
Toelichting	Deze controle is nodig om de integriteit van de data te beschermen.
Mogelijke implementatie	• Voeg een business rule toe die dit veld controleert met behulp van de ProductNVL

Naam	3.8 Toevoegen van meer criteria bij het zoeken naar Products
Prioriteit	5
Status	Open
Beschrijving	Het zoeken naar een Product kan alleen op het id van het product.
Toelichting	Het is wenselijk dat er ook gezocht kan worden in de beschrijving, of binnen een prijs bereik o.i.d.
Mogelijke implementatie	• Voeg een ProductList klasse toe die een reeks Fetch methoden heeft voor de belangrijkste velden.

Naam	3.9 Mapping van NVL's
Prioriteit	3
Status	Open
Beschrijving	Als alle NVL's op een centrale plaats bereikbaar zijn kunnen er op een eenvoudige manier operaties op alle NVL's worden uitgevoerd.
Toelichting	Zeker omdat er problemen zijn met het invalideren van de caches van NVL's is het wenselijk om alle NVL's in de applicatie tegelijk te kunnen invalideren.
Mogelijke implementatie	• Pas de base class van de NVL's aan zodat deze alle instanties toevoegt in een statische lijst, of registreert bij een andere klasse die de instanties verzamelt.

Naam	3.10 Overzichtelijker maken van Fetch operaties
-------------	---

Prioriteit	5
Status	Open
Beschrijving	Bij klassen met veel velden waarop gefilterd kan worden kunnen de Fetch operaties veel overloads tellen, of zijn er veel default arguments.
Toelichting	Functionies met veel default arguments worden afgeraden omdat deze instabieler zijn dan functionies zonder default arguments.
Mogelijke implementatie	• Maak gebruik van het parameter object pattern om de verschillende zoekparameters samen te voegen. • Voeg voor elke samenstelling van zoekparameters een andere overload toe. Maar behoud in dit geval de functie met default arguments.

6 MOGELIJK VERDER ONDERZOEK NODIG

In dit hoofdstuk komen nog een aantal opmerkelijkheden naar voren waarvan niet direct gezegd kan worden of het een fout betreft, maar waar verder onderzoek geboden is.

Naam	4.1 Onjuiste instelling m.b.t. mobile
Prioriteit	3
Status	Open
Beschrijving	De FabricIdentity.Roles lijst is een MobileList implementatie, dit voldoet niet aan de verwachting. Ook de BusinessBase is een 'MobileBase'
Toelichting	Mogelijk is er ergens een fout in de detectie van het platform, waardoor er Mobile implementaties worden gebruikt.

Naam	4.2 Null returns bij uitlezen OrderInfoList
Prioriteit	8
Status	Open
Beschrijving	Items in een OrderInfoList wijzen soms naar null in plaats van een OrderInfo object.
Toelichting	Er is geen reproductie voor deze fout bekend.

Naam	4.3 Controle op ApproverId
Prioriteit	5
Status	Open
Beschrijving	Aan het ApproverId veld van een Order kunnen ongeldige waarden worden toegekend.
Toelichting	Er zijn geen business rules die de validiteit van het Approverid controleren.

Naam	4.4 Voor veel objecten geen gedeelde base class
Prioriteit	7
Status	Open
Beschrijving	Door het gebruik van CRTP is het in veel gevallen moeilijk om functionaliteit generiek te implementeren.
Toelichting	CRTP zorgt ervoor dat twee klassen die een base class lijken te delen dit toch niet doen. Er bestaan dan veel verschillende 'paden' van base class naar sub class die beginnen bij de eerste klasse die CRTP gebruikt. Om toch abstractie toe te kunnen passen moet er zo ver 'omhoog' gegaan worden in de hiërarchie dat de gewenste functionaliteit vaak niet bestaat in die 'hoge' klassen.

Naam	4.5 OrderLineAmount: Verplicht invullen van velden die onnodig lijken
Prioriteit	2
Status	Open
Beschrijving	Wanneer de 'AutoCalc' aan wordt gezet moeten er toch bedragen worden ingevuld in de prijs velden.
Toelichting	Naar verwachting zou AutoCalc betekenen dat de prijzen automatisch worden ingevuld. Toch blijven de prijs velden verplicht tijdens het aanmaken van het object.

Naam	4.6 Lege entries in NVL's
Prioriteit	3
Status	Open

Beschrijving	Veel NVL's bevatten één lege waarde, dit kan voor problemen zorgen.
Toelichting	Veel functies tonen alle waarden in een NVL, er moet in dit geval overal gecontroleerd worden of een entry geen null is zodat deze kan worden overgeslagen.

Naam	4.7 Fetch geeft geen null maar een exception als er geen match is gevonden
Prioriteit	5
Status	Open
Beschrijving	Wanneer een Fetch operatie geen resultaat heeft wordt een exception gegeven.
Toelichting	Een exception zou gereserveerd moeten zijn voor exceptionele omstandigheden. Zeker in een web applicatie kunnen gebruikers zomaar objecten opvragen met id's die niet bestaan.

Naam	4.8 Demo database is niet compleet
Prioriteit	6
Status	Open
Beschrijving	In de demo database moeten zo veel mogelijk verschillende 'situaties' staan zodat er getest kan worden zonder dat de ontwikkelaar zelf dingen aan de database moet toevoegen.
Toelichting	De NVL's salesVatScenarioNVL en purchaseVatScenarioNVL zijn leeg. Daarnaast zou er een tweede demo database moeten bestaan met een zeer groot aantal van de meest gebruikte business objects, zodat performance testing makkelijker toegepast kan worden.

Naam	4.9
Prioriteit	5
Status	Opgelost
Beschrijving	De authenticatie in Fabric verloopt standaard via Windows user name.
Toelichting	Windows authenticatie is van toepassing op desktop applicaties, in geval van een web applicatie moet de authenticatie automatisch op iets anders worden gezet.

7 AANBEVELINGEN EN CONCLUSIES

In dit rapport zijn de belangrijkste problemen met Fabric in kaart gebracht. Tijdens het onderzoek zijn er ook andere, veelal kleine, problemen gevonden die al direct opgelost konden worden. De problemen die nu nog open staan zijn grotendeels relatief kleine problemen die zich voordoen in bepaalde situaties. Deze zijn waarschijnlijk ook snel op te lossen. Er zijn echter ook een klein aantal vervelende problemen waarvoor een oplossing nog niet in zicht is. Dit is voornamelijk het globale cachen van NVL's en het inloggen door middel van een sessie. Wanneer deze twee problemen zijn opgelost en ook de belangrijkste kleine bugs zijn verholpen kan een eerste versie uitgebracht worden.

Vooralsnog is de functionaliteit van het gehele framework beperkt, tot nu toe zijn alleen de belangrijkste business objects geïmplementeerd. Hiermee kan uitstekend een simpele overzicht applicatie worden gebouwd, maar er moet nog veel aan Fabric worden toegevoegd voordat het complexere processen kan ondersteunen.