

# Werken met Open Source



**Kritische (succes)factoren  
bij Open Source projecten**



**Marteniek Bierman**

Bedrijfskundige Informatica  
Informatie & Communicatie Academie ICA  
Hogeschool van Arnhem en Nijmegen HAN

Scriptie  
ter afronding van de opleiding Bedrijfskundige Informatica  
aan de Informatie & Communicatie Academie van de  
Hogeschool van Arnhem en Nijmegen.

Deze scriptie is mede tot stand gekomen dankzij de inzet van het  
management en de ontwikkelaars van:



Topicus B.V.  
Brinkpoortstraat 11  
7411 HR Deventer  
0570 – 662662  
[www.topicus.nl](http://www.topicus.nl)



Infa B.V.  
Dorpsstraat 151-155  
2903 LA Capelle aan den IJssel  
010-2646666  
[www.infa.nl](http://www.infa.nl)

Begeleiders: Dhr. Jan Hugo Wijbenga (Mentor)  
Dhr. Jorg Janssen (Assessor)  
Dhr. Maarten Kok (Topicus)

Omslag: Ontwerp waarin "Cathedral" en "Bazaar" (zie hoofdstuk  
"Literatuurstudie") geflankeerd worden door Topicus en Linux  
Logo's.

Copyright © M.N. Bierman Zutphen

Alle informatie uit dit document mag worden hergebruikt in publicaties, presentaties, of in  
welke andere vorm dan ook, mits de herkomst wordt genoemd en de informatie in de  
nieuwe vorm op eenzelfde manier als deze voor iedereen beschikbaar wordt gesteld.

## Voorwoord

Aan het einde van mijn studie Bedrijfskundige Informatica aan de Informatie en Communicatie Academie van de Hogeschool Arnhem en Nijmegen (ICA/HAN), is het uitvoeren van een afstudeeropdracht de afronding van de totale opleiding. Het onderwerp van deze afstudeeropdracht moet in de lijn liggen van het vakgebied Bedrijfskundige Informatica. Ik heb hiervoor het onderwerp "Open Source" gekozen.

"Open Source" is de afgelopen jaren tot een wijd verbreid, maar ook enigszins ondoorzichtig begrip geworden. Sommige partijen beweren dat Open Source hetzelfde is als open standaarden<sup>1</sup>. Hierbij zou alleen de manier waarop Open Source software wordt aangeropen vrij toegankelijk zijn. Anderen verkondigen dat Open Source geheel vrij is om kosteloos te gebruiken, te verkopen of weg te geven<sup>2</sup>. Hierdoor zou elk onderdeel van de Open Source software dus zonder meer van iedereen zijn. Daartussen bevindt zich een rijke verzameling andere interpretaties van dit begrip. Het is voor degene die met Open Source gaat werken, daarom niet zo helder welke "soort" Open Source er nu eigenlijk bedoeld wordt, met alle risico's die daarbij horen.

Het helder krijgen van deze risico's en het ombuigen ervan tot succesfactoren, is één van de kernpunten van deze afstudeerscriptie. De opzet is om inzicht te krijgen welke zaken er van belang zijn bij het werken met Open Source.

Deze scriptie is gemaakt om een handvat te bieden aan ieder die in de praktijk met Open Source aan het werk gaat. Dat kan iemand zijn die via de hiërarchie in de organisatie te horen krijgt "Dat er met Open Source gewerkt moet gaan worden". Of iemand die een stuk software voor algemeen gebruik aan de rest van wereld wil aanbieden. Maar ook iemand die overweegt om een specifiek stuk Open Source software in een commercieel product op te nemen. Ieder zal in deze scriptie bruikbare informatie vinden.

Deze scriptie is tot stand gekomen door verkenning van bestaande literatuur en recent uitgevoerd wetenschappelijk onderzoek. Maar vooral het horen van praktijkervaringen en meningen van mensen die met Open Source software werken, leverde een schat aan informatie op. Het is mijn bedoeling om met deze scriptie bij te dragen aan inzicht in het fenomeen "Open Source" en aan de verbreiding van het (juiste) gebruik ervan.

Ik heb aandacht besteed aan de insteek vanuit een organisatorische functie, een projectleider of beleidsmaker. Pure technische zaken komen minder aan de orde. Regels programmacode zijn in deze scriptie niet te vinden. Daar zijn andere bronnen voor. Verwijzingen naar deze bronnen zijn echter wel weer opgenomen.

Tot slot, ik gebruik in deze scriptie de omschrijving "Open Source". Hoewel ik ook andere notaties ben tegengekomen blijf ik deze gebruiken. Het gebruik van CamelCaps (eerste letter van een lettergreep is een hoofdletter) doet me af en toe terugdenken aan de tijd dat ik zelf intensief met (bron) code te maken had.

Marteniek Bierman    Zutphen voorjaar 2007

---

<sup>1</sup> Trends in IT 2004|2005

<sup>2</sup> Understanding Open Source software development

# Inhoudsopgave

<b>1</b>	<b>MANAGEMENTSAMENVATTING.....</b>	<b>6</b>
<b>2</b>	<b>INLEIDING.....</b>	<b>7</b>
2.1	OPDRACHT.....	8
2.2	ONDERZOEKSVRAAG .....	8
2.3	SUBVRAGEN .....	9
2.4	WAT IS NU EIGENLIJK "OPEN SOURCE" .....	9
2.5	WAT ZIJN OPEN SOURCE PROJECTEN .....	10
2.6	VERDERE AFBAKENING VAN HET ONDERZOEK .....	10
2.7	AANPAK .....	11
<b>3</b>	<b>LITERATUURSTUDIE .....</b>	<b>12</b>
3.1	KLASSIEKERS .....	12
3.1.1	<i>The Cathedral &amp; the Bazaar</i> .....	12
3.1.2	<i>Understanding Open Source Software development</i> .....	13
3.1.3	<i>The unauthorized white papers</i> .....	13
3.1.4	<i>Trends in IT 2004  2005</i> .....	13
3.2	RECENTE PUBLIKATIES .....	14
3.2.1	<i>Understanding Open Source communities</i> .....	14
3.2.2	<i>Open Source Jaarboek 2006-2007</i> .....	15
3.2.3	<i>Trends in IT 2006  2007</i> .....	15
3.3	VOORLOPIGE CONCLUSIES.....	15
<b>4</b>	<b>PRAKTIJK .....</b>	<b>16</b>
4.1	DE MIDOFFICE VAN DE GEMEENTE ALMERE .....	16
4.1.1	<i>Openheid blijkt kritisch</i> .....	16
4.1.2	<i>Open Source zit in de lift, maar leveranciers blijven achter</i> .....	17
4.1.3	<i>Succesfactoren als faalfactoren</i> .....	17
4.2	WICKET.....	17
4.2.1	<i>Het voordeel van een goede paraplu</i> .....	18
4.2.2	<i>Het voordeel van een goede community</i> .....	18
4.2.3	<i>Meedoen met een goede community</i> .....	18
4.3	BUSINESS CASE COMPIERE .....	19
4.3.1	<i>Eerste verkenning</i> .....	19
4.3.2	<i>Nader onderzoek</i> .....	20
4.3.3	<i>Check de techniek</i> .....	20
4.3.4	<i>Check de community</i> .....	20
4.3.5	<i>Contact met de community</i> .....	21
4.4	VOORLOPIGE CONCLUSIES.....	22
<b>5</b>	<b>RESULTATEN.....</b>	<b>23</b>
5.1	FACTOREN, DE KRITISCHE SUCCESFACTOREN WORDEN ZICHTBAAR .....	23
5.1.1	<i>Algemene kennis over het systeem</i> .....	23
5.1.2	<i>Functionele kennis over het systeem</i> .....	24
5.1.3	<i>Technische kennis over het systeem</i> .....	24
5.1.4	<i>Juridische kennis over het systeem</i> .....	24
5.1.5	<i>Organisatorische kennis over het systeem</i> .....	25
5.1.6	<i>Kennis over beheer en onderhoud van het systeem</i> .....	25
5.1.7	<i>Kennis over juridisch en politiek beleid</i> .....	25
5.2	WIE HEBBEN MET KRITISCHE SUCCESFACTOREN TE MAKEN .....	25
5.2.1	<i>Aanbieders</i> .....	25
5.2.2	<i>Gebruikers</i> .....	26
5.2.3	<i>Combinaties</i> .....	26

5.3	THEMA'S, ORDENING VAN FACTOREN .....	26
5.3.1	<i>Beschikbaarheid</i> .....	27
5.3.2	<i>Toegankelijkheid</i> .....	27
5.3.3	<i>Continuïteit</i> .....	27
<b>6</b>	<b>METHODE.....</b>	<b>28</b>
6.1	ONTWIKKELING VAN DE METHODE.....	28
6.1.1	<i>Het platform</i> .....	28
6.1.2	<i>Publicaties</i> .....	28
6.1.3	<i>Community website</i> .....	29
6.1.4	<i>Commerciële website(s)</i> .....	29
6.1.5	<i>Code repository</i> .....	29
6.1.6	<i>Contacten</i> .....	29
6.2	HET SCHEMA .....	29
6.3	HET MODEL.....	31
6.4	GEBRUIK VAN DE METHODE .....	32
6.4.1	<i>Voorbeeld: de implementatie Open Source ERP systeem</i> .....	32
6.4.2	<i>Wanneer is een factor voldoende geborgd</i> .....	33
<b>7</b>	<b>CONCLUSIE EN AANBEVELINGEN.....</b>	<b>35</b>
7.1	TERUG NAAR DE ONDERZOEKSVRAGEN.....	35
7.2	BEANTWOORDING VAN DE HOOFDVRAAG .....	36
7.3	AANBEVELINGEN VOOR NADER ONDERZOEK.....	37
7.3.1	<i>Het borgen van de factoren</i> .....	37
7.3.2	<i>De mate van borging van de factoren</i> .....	37
<b>8</b>	<b>BIJLAGEN .....</b>	<b>38</b>
8.1	RECENTE ONTWIKKELINGEN .....	38
8.1.1	<i>Nationaal politiek beleid</i> .....	38
8.1.2	<i>Europese politiek, patenten en belangen</i> .....	38
8.1.3	<i>Dreigen met patenten en aantonen van "desbewustzijn"</i> .....	39
8.2	DE MEEST GEBRUIKTE OPEN SOURCE LICENTIES .....	40
8.3	VOETNOTEN EN NADERE INFORMATIE .....	41
8.4	GERAADPLEEGDE LITERATUUR .....	44
8.5	GEÏNTERVIEWDEN .....	45
8.6	BRONVERMELDING VAN DE AFBEELDINGEN .....	45
8.7	REVIEWERS.....	45

## 1 Managementsamenvatting

Open Source is in de branche van de software ontwikkeling een factor geworden om rekening mee te (gaan) houden. Maar *hoe* kan er dan rekening mee gehouden worden? Welke invloeden spelen een rol? Welke factoren zijn bepalend voor succes?

Om hierop antwoord te kunnen geven is er gekeken naar de geschiedenis van Open Source. Daarnaast zijn de resultaten van wetenschappelijk onderzoek bestudeerd. Tot slot leert de praktijk een aantal waardevolle lessen.

Hierdoor komen allereerst de kritische succesfactoren zelf naar voren. Dit zijn:

*Algemene kennis over het systeem;*  
*Functionele kennis over het systeem;*  
*Technische kennis over het systeem;*  
*Juridische kennis over het systeem;*  
*Organisatorische kennis over het systeem;*  
*Kennis over beheer en onderhoud van het systeem;*  
*Kennis over (Inter)Nationaal politiek beleid en wetgeving.*

Daarnaast komt naar voren dat Open Source gebruikt wordt door verschillende groepen mensen. Er zijn *aanbieders*, die de Open Source software maken en er zijn *gebruikers*, die de Open Source software gebruiken. Ook kunnen er combinaties van aanbieders en gebruikers ontstaan. Het blijkt dat zowel de aanbieders als de gebruikers met dezelfde kritische succesfactoren te maken hebben, alleen vanuit verschillende invalshoeken. Aanbieders moeten ervoor zorgen dat informatie beschikbaar is, zodat gebruikers deze informatie kunnen vinden en gebruiken.

Vervolgens komt naar voren dat er verschillende thema's binnen een Open Source project te herkennen zijn, waar kritische succesfactoren een rol spelen. Weliswaar moeten bij iedere fase van een Open Source project alle kritische succesfactoren beschouwd en eventueel geborgd worden, maar er is gedurende het project een verschuiving in de thema's te onderkennen. De genoemde thema's zijn:

*Beschikbaarheid;*  
*Toegankelijkheid;*  
*Continuïteit.*

Om *succesvol* met de kritische factoren om te kunnen gaan, is er een methode ontwikkeld. Deze methode bestaat uit:

- Bepaal het tijdstip waarin het project verkeert;
- Bepaal het thema wat op dat moment op het project van toepassing is;
- Bepaal de factor die op dit tijdstip kritisch is;
- Bepaal de eigen positie binnen het project;
- Bepaal via welk platform deze factor geborgd kan worden;
- Haal de informatie van het platform;
- Gebruik de informatie bij de borging van de betreffende factor;
- Beschouw de geborgde factor in verhouding met de andere factoren.

Tot slot is het van belang om te beseffen dat Open Source een dynamisch geheel is, dat per dag kan veranderen (en dat ook doet!). Open Source is *serious business* aan het worden. De informatie in deze scriptie, en de uitgewerkte methode kan bijdragen om er succesvol mee om te gaan.

## 2 Inleiding

*Binnen vier jaar neemt het gebruik van Open Source software toe met meer dan een factor drie. De verwachte totale omzet bedraagt dan 4,3 miljard euro.<sup>3</sup>*

De afgelopen jaren heeft het gebruik van Open Source een grote vlucht genomen. Wat in de jaren tachtig en negentig van de vorige eeuw is gestart uit interesse en gedreven hobbyïsme, is intussen uitgegroeid tot een serieuze, soms ook ongrijpbare, entiteit in de markt van de software ontwikkeling. Duizenden mensen over de hele wereld dragen bij aan het tot stand komen en verbeteren van software voor allerlei toepassingen. Evenzoveel organisaties, commercieel of anderszins, gebruiken Open Source bij het realiseren van hun doelstellingen.

Vijftien jaar geleden was er nog minimale berichtgeving over Open Source. Er werd hooguit gepubliceerd over meningsverschillen tussen docenten en studenten van universiteiten<sup>4</sup>. Tegenwoordig zijn er berichten over multinationale ondernemingen als IBM, Sun en Microsoft die (een deel van) hun software producten wel of niet als Open Source zullen gaan aanbieden. Alles wijst erop dat het gebruik van Open Source voorlopig zal toenemen, zowel in volume als in marktaandeel. Hoewel niet altijd even zichtbaar, is Open Source gewoon niet meer weg te denken uit de huidige maatschappij.

Dit betekent dat meer en meer mensen bij Open Source betrokken zullen raken. Steeds meer mensen zullen bijdragen aan de ontwikkeling van Open Source Software. Maar ook steeds meer mensen zullen die Open Source Software gebruiken voor de ondersteuning van hun dagelijks werk.

Het valt op dat praktische informatie, over hoe er *in de praktijk* met de mogelijkheden van Open Source kan worden omgegaan, weinig voorhanden is. Er bestaat een grote hoeveelheid populaire en wetenschappelijke publicaties. Deze gaan in op de historie van Open Source, hoe bepaalde organisaties ermee omgaan of op de abstracte aspecten ervan. Ook wordt er een ware marketing veldslag uitgevochten rondom de, al dan niet lagere, kosten die Open Source met zich mee zou brengen. Hoe dan ook, er is (nog) weinig voorhanden dat kort en bondig aangeeft waar je op moet letten als je met Open Source aan het werk wilt gaan.

In deze scriptie is juist die vraag cruciaal: *“Waar moet ik op letten om succesvol met Open Source te werken”*?

De hiernavolgende paragrafen beschrijven eerst de volledige opdracht. Daarna volgt een beschrijving van problemen die zich voordeden en welke oplossingen werden gevonden. Uit de opdracht wordt de onderzoeksvraag afgeleid. Aan de hand hiervan worden daarna de subvragen opgesteld. Vervolgens wordt er geformuleerd wat er onder “Open Source” verstaan wordt. Tot slot wordt de verdere afbakening van het onderzoek beschreven.

---

<sup>3</sup> Computable 8 juni 2007, artikel over onderzoek uitgevoerd door IDC, zie ook de bijlage

<sup>4</sup> Andrew Tannenbaum versus Linus Torvalds, zie ook de bijlage

## 2.1 Opdracht

De opdracht voor dit afstudeerproject is ontstaan uit het besef dat Open Source een steeds belangrijker plaats op de softwaremarkt aan het innemen is. Hierdoor is bij de opdrachtgever de behoefte ontstaan om verder naar de mogelijkheden te (laten) kijken. Deze opdrachtgever, Infa B.V. in Capelle aan den IJssel, wil onderzoeken of het van belang kan zijn om (onderdelen van) software die gevoerd wordt, als Open Source naar buiten te brengen. Er zijn ideeën over welke onderdelen het hier zou moeten gaan. Ook is er een beeld van de voordelen die verschillende partijen hiervan zouden kunnen hebben. Omdat deze ideeën en beelden voornamelijk in algemene zin onder woorden te brengen zijn, wordt er bepaald dat ook de opdracht een generiek karakter moet hebben. Het moet niet uitmaken op welk software onderdeel van Infa de resultaten worden toegepast, het moet allemaal algemeen geldend zijn. Zo ontstaat de opdracht om de kritische succesfactoren, die bij Open Source projecten een rol spelen, te onderzoeken. Opdrachtgever voor Infa B.V. in de persoon van Albert van den Broek, Manager van de bedrijfsunit Projects.

Halverwege het onderzoek vindt er echter een wisseling van de wacht plaats. Het resultaat is een nieuw dienstverband bij Topicus B.V. in Deventer. Topicus is, onder andere, gelieerd aan de Universiteit Twente en daarom zeer geïnteresseerd in onderzoek en innovatie. Hierdoor kan het opdrachtgeverschap zonder veel problemen door Topicus worden overgenomen. Wel wordt de opdracht zelf wat aangescherpt. Topicus heeft al ervaring met Open Source. Het wordt daarom van belang geacht om de opdracht minder algemeen en meer praktisch gericht te maken. Behalve onderzoek naar de kritische succesfactoren, moet ook bekeken worden hoe er het beste met deze factoren kan worden omgegaan. Opdrachtgever is Topicus B.V. in de persoon van Maarten Kok, directeur van de bedrijfsunit Finance

## 2.2 Onderzoeksvraag

Om helder de onderzoeksvraag voor het onderwerp neer te zetten is eerst nagedacht over het directe belang van degene die met Open Source (gaat) werken. Deze mensen moeten praktisch nut ondervinden van het resultaat. Dat dit een goede richtlijn is komt gedurende het onderzoek naar voren. Er komt namelijk een behoorlijke verzameling wetenschappelijke verhandelingen voorbij. Hiervan blijkt dat deze, behalve hun wetenschappelijke waarde, weinig nut voor praktische toepassing hebben. Uiteindelijk is de formulering van de onderzoeksvraag:

*Wat zijn kritische succesfactoren bij Open Source projecten en hoe kunnen deze geborgd worden?*

### 2.3 Subvragen

Voortgaand op de onderzoeksvraag zijn de volgende subvragen naar voren gekomen:

*Wat is Open Source?*

*Wat zijn Open Source projecten?*

*Zijn alle Open Source projecten gelijk van aard?*

*Bij welke fasen van een project spelen kritische succesfactoren een rol?*

*Wanneer is er sprake van een succesvol project?*

*Wanneer zijn succesfactoren kritisch?*

*Bestaat er onderscheid in de mate van het kritisch zijn van deze factoren?*

*Is het borgen van deze factoren te incorporeren in een project?*

*Is er een methode te ontwikkelen die het bepalen en borgen van deze factoren voor de projectleiding makkelijker kan maken?*

### 2.4 Wat is nu eigenlijk "Open Source"

De eerste subvraag is als essentieel voor het gehele onderzoek aangemerkt. Er is onderkend dat het onderzoek staat of valt met een duidelijke definitie van het begrip Open Source<sup>5</sup>.

Zoals al eerder gemeld, bestaan er nogal wat verschillende interpretaties van het begrip "Open Source". Het is belangrijk om hierin enige structuur te scheppen zodat duidelijk is waar het eigenlijk over gaat.

In algemene zin geeft de vertaling van de term "Open Source" een goed houvast. "Open" wordt in de Nederlandse vertaling geïnterpreteerd als *toegankelijk* of *transparant*, zoals een *open* deur, *open verslaglegging* of een *open brief*. "Source" is de term voor bron en wordt in deze context gehanteerd als de *broncode* van een stuk software. Het is het deel waaraan door programmeurs gewerkt wordt. Het is het deel dat in een specifieke taal als bijvoorbeeld C, Delphi, C# of Java geschreven is. De broncode is de container voor het idee achter de software (belangrijk voor kwesties over intellectueel eigendomsrecht). Uitgaande van deze vertaling is natuurlijk een grote hoeveelheid interpretaties te maken.

De meest smalle interpretatie die voorbij komt is "Open Source zijn open standaarden". Deze definitie zal in deze scriptie zeker niet worden aangehouden. Deze is namelijk onjuist. Open Source is veel meer dan een definitie van de interface in het publieke domein. Ook is deze definitie niet houdbaar tegen de hierboven genoemde vertaling. Open standaarden laten op geen enkele manier

---

<sup>5</sup> De begrippen "Open Source" en "Open Source Software" worden als identiek beschouwd. In deze scriptie wordt de benoeming "Open Source" aangehouden

iets weten over hoe zaken in de bron geregeld zijn en daar gaat het nou juist om. Om een metafoor te gebruiken: "Er komt wisselstroom met een spanning van 220 volt en een frequentie van 50 hertz uit het stopcontact". Dit is een open standaard, publieke informatie. Het zegt echter helemaal niets over hoe deze stroom bij de bron, in de centrale, gemaakt wordt.

De meest brede interpretatie is ook al even genoemd: "Open Source is geheel vrij om kosteloos te gebruiken, te verkopen of weg te geven". Dat is al meer in de buurt, maar dekt de lading (nog) niet helemaal. In de meeste gevallen waar met Open Source gewerkt wordt, speelt de licentie een grote rol. Dit betekent dat zaken niet geheel vrij zijn. Er is vaak juist wel wat vastgelegd over *de mate waarin(!)* de broncode vrij is om kosteloos te gebruiken, te verkopen of weg te geven. Duidelijk heeft de definitie van de OSI<sup>6</sup>, die dieper ingaat op de kwestie de gebruikte licentie hier invloed op gehad.

In deze scriptie zal dan ook de volgende definitie worden gehanteerd:

*Open Source is software waarvan de broncode, tegen de in de licentie gestelde voorwaarden, beschikbaar is gemaakt voor een bredere groep dan alleen de houder van het intellectuele eigendom.*

## 2.5 Wat zijn Open Source projecten

Ook de tweede subvraag is als belangrijk getypeerd. Zeker de afgelopen jaren worden er nogal wat initiatieven "Open Source projecten" genoemd. Zo worden ook teksten (in Blogs of in andere vormen) nog wel eens aangeduid als Open Source project. Dit omdat iedereen de betreffende tekst mag gebruiken en iedereen er immers wat aan kan en mag veranderen. In de geest van Open Source, zoals eerder beschreven, is dit dus niet onjuist, echter wel wat ruim genomen. Om zaken helder te houden worden in deze scriptie alleen projecten bedoeld die te maken hebben met het automatiseren van bepaalde processen. Projecten die dus een duidelijke ICT component in zich hebben. Daarnaast maken deze projecten gebruik van Open Source software, of leveren Open Source software op. Strikte definitie wordt dus:

*Een Open Source project is een automatiseringsproject waarin Open Source software gebruikt wordt, of wat Open Source software zal opleveren.*

## 2.6 Verdere afbakening van het onderzoek

Om het geheel verder af te bakenen is gesteld dat alleen serieuze projecten in het onderzoek meegenomen zullen worden. De mate van serieuze is bepaald aan de hand van een schatting van de hoeveelheid tijd die er aan een project besteed is. Ook de visie op de langere termijn (zo die bekend is) speelt een rol.

Daarnaast is er bepaald dat de verdienmodellen, die bij veel Open Source projecten wel degelijk een belangrijke rol spelen, niet uitputtend behandeld zullen worden. De verkenning van dit onderwerp geeft aan dat hier zeer interessante onderzoeksmogelijkheden liggen. Niet in de laatste plaats omdat er aangetoond kan worden dat er (ook) met Open Source geld te verdienen is. Echter,

---

<sup>6</sup> Open Source Initiative, zie bijlage

uitwerking hiervan vergt meer tijd dan beschikbaar is. Dit zou niet in een juiste verhouding staan met de andere onderwerpen die onderzocht worden.

Verder is besloten om kritische succesfactoren, die ook bij "gewone" automatiseringsprojecten een rol spelen, niet bij het onderzoek te betrekken. Vanzelfsprekend is bijvoorbeeld het werken met een planning bij Open Source projecten van belang. Ook communicatie met de opdrachtgever en de uitvoerders is een kritische succesfactor. Alleen zullen deze zaken dezelfde kritische rol vervullen in een automatiseringsproject waarbij geen Open Source gebruikt wordt. Als zodanig zijn ze dus niet typische Open Source factoren en worden dus buiten beschouwing gelaten.

Tot slot is hier de opmerking op zijn plaats, dat projecten die de eerste keer als Open Source project aangemerkt kunnen worden, dit niet noodzakelijk de tweede keer ook zijn. Zo is het gebruik van een Open Source component de eerste keer wel degelijk een Open Source project, en zijn de kritische succesfactoren van toepassing. Wordt dezelfde component echter hergebruikt, dan zijn (als het goed is) de meeste factoren intussen geborgd en is het geen volledig Open Source project meer. Voorbeelden hiervan zijn het eerste gebruik van Open Source databases en Content Management Systems (CMS).

## **2.7 Aanpak**

De hoeveelheid informatie over Open Source is nogal omvangrijk. Om de onderzoeksvragen te kunnen beantwoorden is daarom eerst een verkenning uitgevoerd door middel van literatuuronderzoek. Hierbij is van belang om te vermelden dat niet alleen "echte" boeken, maar ook publicaties (en andere vormen) via internet gebruikt zijn. Door middel van praktijkonderzoek, interviews en case studies, is de verkenning verder getoetst en uitgewerkt. De gevonden punten worden in het resultaat beschreven. Hierna wordt er gekeken welke samenhang er ontdekt kan worden. Uitwerking hiervan levert een methode op. Aan de hand van deze methode worden er vervolgens conclusies getrokken en worden de onderzoeksvragen beantwoordt.

Om aandacht te besteden aan het feit dat Open Source een actueel en dynamisch onderwerp is, wordt er tot slot ook aandacht besteed aan recente gebeurtenissen. Deze worden voor de ontwikkelingen rondom Open Source van belang geacht.

### 3 Literatuurstudie

De afgelopen jaren is er een behoorlijke hoeveelheid publicaties over het onderwerp Open Source verschenen. Type "Open Source" in Google en het resultaat is ruim 600 miljoen hits. De combinatie "Open Source" en "Books" geeft 132 miljoen hits. De combinatie "Open Source" en "Literature" levert 1.2 miljoen hits op. Nou leiden al die "hits" niet noodzakelijk tot unieke boeken maar het is dus zeker moeilijk om een keuze te maken.

Om evengoed uit de voeten te kunnen is er bepaald dat er een aantal boeken "van vroeger" en een aantal boeken "van nu" bestudeerd worden. De bedoeling hiervan is een beeld te krijgen van de ontwikkeling die Open Source heeft doorgemaakt, maar ook om te kijken of er in het recente verleden ook al kritische succesfactoren te herkennen waren.

De keuze van de boeken zelf is soms op eigen inzicht gemaakt, dit is bijvoorbeeld het geval bij het Open Source Jaarboek. Maar vaak is het advies gevolgd van anderen, die geacht worden er wel verstand van te hebben. Zo is *The Cathedral & the Bazaar* bestudeerd op aanraden van de heer Marco Wobben (consultant bij BCP-Software).

#### 3.1 Klassiekers

Voor de verkenning van de geschiedenis van Open Source is een drietal boeken en een trendrapportage uitgekozen. Deze verzameling geeft een mooi beeld van hoe Open Source er in het nabije verleden (vaak minder dan 10 jaar geleden) voor stond. Velen beschouwen intussen *The Cathedral & the Bazaar* als standaard werk voor Open Source. De boeken van Feller & Fitzgerald en Rosenberg blijken echter minstens zo waardevol. Ook de Trends in IT bevat belangrijke informatie, zeker in combinatie met de recente versie ervan. In de volgende paragrafen worden deze titels beschouwd en worden de belangrijkste punten naar voren gehaald.

##### 3.1.1 The Cathedral & the Bazaar

Eric S. Raymond schrijft in 1997 *The Cathedral & the Bazaar*. Dit boek is intussen een van de standaardwerken over Open Source geworden. Allereerst beschrijft Raymond de ervaringen met zijn eigen Open Source Project Fetchmail, een systeem<sup>7</sup> dat mail doorstuurt over een lokaal netwerk (LAN)<sup>8</sup>. Daarnaast schrijft hij over de voorgeschiedenis van Open Source en het ontstaan van Linux. Hier wordt duidelijk dat Linux niet "uit het niets groot is geworden" maar dat het heel duidelijk te plaatsen is tussen andere (Open Source) initiatieven uit die tijd. Wellicht is het boek vooral zo interessant omdat het paradigma Cathedral & Bazaar geïntroduceerd wordt. Dit paradigma slaat op de twee verschillende manieren waarop, volgens Raymond, software kan worden ontwikkeld.

Eenzijds kan de Cathedral stijl worden toegepast. Centraal georganiseerde onderdelen vallen naadloos in elkaar en vormen zo één geheel. Er ontstaat een monoliet die niet of slechts met grote moeite aan te passen is. Als de

<sup>7</sup> De begrippen "Systeem" en "Applicatie" zijn weliswaar niet identiek, toch wordt in deze scriptie de benoeming "Systeem" aangehouden, ook als "Applicatie" juister zou zijn.

<sup>8</sup> Zie de bijlage

bouwmeester wegvalt of als er een enkel onderdeel weggenomen wordt, stopt in het beste geval de bouw en in het slechtste geval stort alles in elkaar.

Aan de andere kant is er de Bazaar<sup>9</sup> stijl, een open ruimte waarin alle onderdelen zelf hun plaats vinden en zich onderling organiseren. Groot vertrouwen (en openheid!) is nodig om software op deze manier te realiseren. Maar neem één, twee of meerder onderdelen weg en het geheel gaat gewoon door. Binnen korte tijd zijn de lege plekken gevuld met onderdelen die kwalitatief alweer beter kunnen zijn dan hun voorgangers.

Een voorbeeld van de Cathedral stijl is een commercieel ERP system zoals SAP. Een voorbeeld van de Bazaar stijl zijn de Linux distributies.

### **3.1.2 Understanding Open Source Software development**

In dit boek geven Joseph Feller en Brian Fitzgerald een blik op de positie van Open Source aan het begin van deze eeuw (2002). Het is duidelijk een momentopname, maar daarom juist interessant. Er wordt goed duidelijk hoe Open Source zich vanaf het prille begin heeft ontwikkeld. De auteurs proberen een overzicht van de toenmalige stand van zaken te geven aan de hand van de steekwoorden "wat, hoe, wie, waar". Het geheel is wetenschappelijk goed onderbouwd en helpt om de fundamenteën van Open Source te begrijpen. Vooral wordt er duidelijk welke enorme groei Open Source in de afgelopen jaren heeft doorgemaakt. Zo is bijvoorbeeld de hoeveelheid bruikbare Open Source systemen geweldig toegenomen. Feller en Fitzgerald beschrijven in hun boek 21 producten, waaronder Linux, KDE, Perl en Emacs, en benoemen deze als "Major Open Source Projects". Tegenwoordig is een dergelijke lijst vele malen langer.

### **3.1.3 The unauthorized white papers**

Donald Rosenberg geeft in dit boek een mooie kijk in de keuken van de (Noord Amerikaanse) business rondom Open Source rond het jaar 2000. Hij beschrijft verschillende systemen en de organisaties die ze ontwikkelen. Alles gezien vanuit een bedrijfskundig gezichtspunt. Zeer leerzaam is de kijk op de concurrentie en interactie tussen een aantal grote ondernemingen als IBM, Sun, Red Hat en Microsoft. Open Source blijkt al langere tijd hét breekijzer om aan de positie van de concurrent te wrikken. Intussen zijn er aanzienlijk meer systemen en organisaties in het Open Source speelveld gekomen. De verhoudingen tussen grote spelers, zoals Rosenberg ze beschrijft, zijn echter weinig veranderd.

### **3.1.4 Trends in IT 2004| 2005**

In deze gids, die ieder jaar uitkomt, worden de belangrijkste trends en verwachtingen binnen de IT branche beschreven. Veelal gaat een beschrijving gepaard met het aanhalen van onderzoeken van anderen. Het onderwerp Open Source wordt ook beschreven. Het is duidelijk dat de samenstellers nog beperkte kennis van Open Source hebben, omdat het voornamelijk over (de toepassing van) Linux gaat. Zo ontstaat de indruk dat er geen andere Open Source systemen zouden zijn. Daarnaast worden er wat vergissingen in de feiten gemaakt. Open Source wordt gelijkgesteld met open standaarden wat toch echt niet correct is.

---

<sup>9</sup> Zie de bijlage

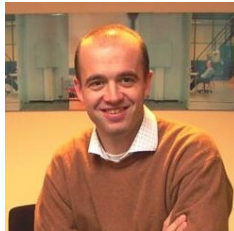
Tevens wordt Lindows<sup>10</sup> aangehaald als de door de rechter verboden naam voor Linux.

Interessant zijn de onderzoeken naar de bekendheid en de mate waarin Open Source wordt gebruikt. Hieruit blijkt dat vooral in de branches transport & vervoer en bij handelsorganisaties, Open Source al veel langer wordt toegepast.

### 3.2 Recente publikaties

Om overzicht en vergelijking tussen heden en verleden te creëren is ook een drietal recente publicaties bestudeerd. Om tevens de nadruk wat meer bij de huidige situatie in Nederland te leggen is er gekozen voor boeken van Nederlandse auteurs. Dit maakt het ook makkelijker om eventueel verdere achtergrond informatie te verkrijgen. Met name bij de doctoraal scriptie van R. van Wendel de Joode blijkt het doorpraten over de publicatie een schat aan concrete informatie op te leveren.

#### 3.2.1 Understanding Open Source communities



Voor het behalen van zijn doctorsgraad aan de TU Delft publiceert Ruben van Wendel de Joode in 2005, de bevindingen van zijn onderzoek naar de organisatie van de Open Source communities. Deze communities zijn vaak de drijvende kracht achter de instandhouding en de vernieuwing van Open Source producten. Hij ontdekt dat deze Open Source communities moeilijk te doorgronden zijn. De manier waarop ze zijn georganiseerd, en de drijfveer van de leden is een complex geheel. Communities blijken zelforganiserend te zijn. Het is dus niet mogelijk om uit te gaan van een algemeen model waar communities aan voldoen. Wel zijn er een aantal regels te herkennen waardoor continuïteit en de juiste (samen)werking met de communities geborgd is.

Er volgt een persoonlijk interview met de auteur bij de TU Delft. Hier wordt verder ingegaan op zaken die bij de ontwikkeling van Open Source een belangrijke rol spelen. Met name de (omgang met de) community komt uitgebreid aan de orde. Daarnaast worden ook verdienmodellen en kwesties rondom licentiëring besproken. De scheiding tussen gebruiker en aanbieder van Open Source wordt onderkend. Ook worden kritische succesfactoren voor beide groepen geformuleerd en beschouwd.

---

<sup>10</sup> Lindows is een software distributie met onder andere een Linux versie erin. Zie ook de bijlage

### 3.2.2 Open Source Jaarboek 2006-2007

Het Open Source Jaarboek geeft een breed beeld van de huidige positie van Open Source. Hierbij ligt de nadruk natuurlijk op Nederland, maar hier en daar worden er vanzelfsprekend ook internationale uitstapjes gemaakt. Het is een boek dat allereerst een goed overzicht geeft van welke systemen het meeste gebruikt worden. Daarnaast geeft het ook zeer waardevolle informatie over de Nederlandse politiek en gaat diep in op licentie- en auteursrecht kwesties. Aan het einde wordt er ook toekomstvisie geschetst en is er een uitgebreide lijst met contactinformatie.

Met name de informatie over de politiek en de licenties is gebruikt bij het formuleren van de kritische succesfactoren.

### 3.2.3 Trends in IT 2006| 2007

In de meest recente versie van dit boek valt op dat de tekst dezelfde is als de eerdere uitgave. Kennelijk is Open Source stabiel genoeg om ervan uit te gaan dat er sinds 2004|2005 niets veranderd is. Wat ook opvalt is dat de weergegeven onderzoeken een ander beeld laten zien dan vroeger. Zo is de toepassing van Open Source enorm gestegen. In het onderwijs en de overheid is het gebruik van Linux toegenomen. Ook de banken en verzekeraars beginnen Linux te ontdekken. Voor het eerst wordt er nu ook overzicht gegeven van het gebruik van Open Source in het algemeen (dus niet meer alleen Linux). Vooral in de zakelijke dienstverlening worden intussen veel Open Source systemen gebruikt. Daarnaast laten ook de maatschappelijk georiënteerde branches een toename zien.

### 3.3 Voorlopige conclusies

Respect in de omgang met de community blijkt, voor zowel de aanbieder als de gebruiker, leidend. De copyright en licentiekwesties zijn, juist in de Open Source wereld, transparant en werkbaar. Kennis van de strekking ervan is echter onontbeerlijk, om niet met verrassingen geconfronteerd te worden. Politiek kan zowel kans (voorrang voor Open Source en mogelijk innovatie subsidies) als bedreiging (toestaan van patenten op software) vormen. Ook op het gebied van nieuwe versies met al dan niet gewenste nieuwe functionaliteit, moeten keuzes worden gemaakt. De keus bestaat hier uit helemaal in eigen beheer nemen of volgen van de community. Beide brengen hun eigen voor- en nadelen met zich mee. De indeling in gebruiker en aanbieder, en de geformuleerde kritische succesfactoren, zijn meegenomen in de ontwikkelde methode<sup>11</sup>.

---

<sup>11</sup> Zie hoofdstuk 7 "Methode"

## 4 Praktijk

Behalve theoretische kennis uit publicaties, is ook kennis uit de praktijk gewenst. Hiervoor zijn drie recente Open Source projecten onderzocht. Deze projecten zijn gericht op drie zeer uiteenlopende domeinen. Deze domeinen zijn de gemeentelijke overheid, gebruikers van het World Wide Web en de bedrijfsunit Finance van Topicus B.V.

### 4.1 De Midoffice van de gemeente Almere



In april 2003 wordt een artikel gepubliceerd<sup>12</sup>, waarin beschreven wordt hoe de gemeente Almere, als één van de eersten aan de slag gaat met Open Source. Er wordt contact gezocht en in het interview met de betrokken beleidsadviseur, de heer J. Verwaard, komt naar voren welke zaken een rol spelen bij het kiezen voor Open Source. Ook verwachtingen en mogelijke bedreigingen komen aan de orde.

In Juni 2007 wordt er opnieuw een interview gehouden. Eerdere uitspraken en verwachtingen worden tegen het licht gehouden. Ook de voor deze scriptie geformuleerde onderzoeksvraag komt naar voren. De *kritische succesfactoren*, maar ook de tegenvorm *kritische faalfactoren* worden uitgediept.

Tot slot vindt er een interview plaats met de heer M. Krans, de (toenmalige) projectleider van leverancier EMaxx. Hierdoor kunnen beide kanten van het verhaal worden beschouwd.

#### 4.1.1 Openheid blijkt kritisch

Openheid blijkt een van de belangrijkste punten te zijn bij de ontwikkeling in Open Source voor een overheidsorganisatie. Er is binnen de overheid intussen een bewustwordingsproces op gang gekomen, dat het gebruik van Open Source wel degelijk stimuleert. De uitwerking ervan laat echter nog wel ruimte voor verbetering. Van belang is de openheid in het opstellen van specificaties en het afdwingen van openheid bij de leveranciers. De ervaring leert dat er nog teveel ruimte gelaten wordt voor interpretatie. Hierdoor worden interfaces niet altijd consistent geïmplementeerd. Ook komt het voor dat leveranciers zich maar gedeeltelijk houden aan de gewenste openheid van de geleverde code. Systemen worden op die manier weer (deels) gesloten met alle gevolgen van dien. De opzet bij Open Source projecten is juist om gebruik te maken van de veelheid aan

<sup>12</sup> Computable 4 april 2003, artikel over de Open Source midoffice van Almere, zie ook de bijlage

kennis en ervaring die er bij de verschillende overheidsorganisaties is. Als leveranciers die kennis vervolgens inkapselen, terwijl deze eigenlijk gedeeld had moeten worden, is het doel niet bereikt.

#### 4.1.2 Open Source zit in de lift, maar leveranciers blijven achter

Het blijkt dat Open Source intussen gezien wordt als een serieus alternatief. Ook op Europees niveau komt er steeds meer ruimte om Open Source toe te passen. Cruciaal is hierbij (alweer) de eenduidigheid van de specificaties. Met name daar waar het interoperabiliteit betreft, is dit nog vaak wat onder de maat. Belangrijk argument om Open Source te gebruiken, is de mogelijkheid om koppelverkoop door leveranciers te vermijden. Er is een goed besef dat er zuinig met gemeenschapsgeld moet worden omgesprongen. Doorgaand op het financiële aspect, blijkt dat de huidige leveranciers vaak (nog) geen verdienmodel voor Open Source hebben ontwikkeld. De ontwikkelde Open Source systemen worden nog erg veel gezien als eenmalige projecten waarvan de resultaten vervolgens "kwijt" zijn. Uurtje factuurtje is kennelijk het hoogst haalbare. De overheidsleveranciers doen ook geen moeite om communities op te zetten of te ondersteunen.

#### 4.1.3 Succesfactoren als faalfactoren

Belangrijke les uit de ontwikkeling van de midoffice van Almere is het feit dat (te) weinig aandacht voor succesfactoren kan leiden tot het falen van het project. De code van de midoffice is intussen Open Source geworden. De leverancier heeft de code onder de GPL<sup>13</sup> naar buiten gebracht<sup>14</sup>. Er is echter weinig support voor een community georganiseerd of op een andere manier iets gedaan om de rest van de wereld tot bijdragen te verleiden. Het effect hiervan is dat er geen verdere ontwikkeling op Open Source basis plaatsvindt.

## 4.2 Wicket



Op 18 augustus 2004 begint Jonathan Locke, Wicket<sup>15</sup> als Open Source project. Wicket is een framework voor webapplicaties, geschreven in Java. Wicket voorziet in de mogelijkheid om vanuit de HTML pagina's, aanroepen uit te voeren naar de functionaliteit van het framework. Vanzelfsprekend is deze functionaliteit uit te breiden. In september 2004 groeit de Wicket community met Nederlandse deelnemers en wordt het Project op SourceForge ondergebracht. In februari van het daaropvolgende jaar (2005) wordt er flink promotie gevoerd, onder andere op de JavaOne conferentie in San Francisco en op de JavaPolis conferentie in Antwerpen. Hierdoor staat Wicket vanaf die tijd definitief op de kaart.

<sup>13</sup> Zie bijlage

<sup>14</sup> <https://www.uitwisselplatform.nl/projects/midoffice/>

<sup>15</sup> Zie bijlage

#### 4.2.1 Het voordeel van een goede paraplu

In juni 2007 verhuist het Wicket project definitief naar Apache. Vanaf dat moment wordt Wicket gezien als een volwassen framework voor professioneel gebruik. De verhuizing van het project naar Apache heeft meerdere redenen:

- Apache beschikt over een betere infrastructuur dan SourceForge;
- Apache heeft een goede bereikbaarheid, spam krijgt er minder kans;
- Apache heeft een goede naamsbekendheid opgebouwd;
- Wicket heeft duidelijk meerwaarde voor de Apache webserver.
- Apache heeft een hechte ("Established") community;

Vooraf het laatste punt blijkt veel positieve meerwaarde mee te brengen.

#### 4.2.2 Het voordeel van een goede community

De Wicket Community is te vergelijken met een voetbalvereniging. Alle leden van de voetbalvereniging hebben een eigen leven, vaak totaal verschillend van elkaar. Toch zijn ze bij het voetballen een hechte groep, die elkaar vaak door en door kent. Als er dus een wedstrijd gespeeld moet worden (of als er supporters nodig zijn) dan kunnen ze van elkaar op aan en klaren met elkaar de klus. De samenhang én de inhoudelijke kennis van de Wicket community is zó goed dat er wordt overwogen om langere responstijden in te bouwen. Dit om ook anderen dan de harde kern van de community, de gelegenheid te geven om te reageren op vragen vanuit de gebruikers. Reactietijd is op dit moment ongeveer 5 minuten(!).

Er wordt wel gezegd dat een goed Open Source project prima kan bestaan met slechte code....én een goede community. Andersom betekent vaak het einde van het project.

#### 4.2.3 Meedoen met een goede community

De Wicket community kent geen expliciete leider<sup>16</sup>. Wel zijn er verschillende rollen verdeeld. Zo zijn er leden van de community met een heldere visie, terwijl anderen sterker zijn in een rol als administrator. Zes personen wordt beschouwd als het minimum aantal voor een goed functionerende community. Meer is natuurlijk prima, en een maximum wordt niet genoemd. Wie bij wil dragen aan de code van Wicket, kan mee doen met de community. Acceptatie vindt plaats door middel van een (interne) stemming over de geschiktheid van de kandidaat. Bij goedkeuring krijgt het nieuwe lid commit- (schrijf) rechten op de code. Het beoordelen van aanpassingen in de code vindt dus plaats op het niveau van de persoon en *niet* op het niveau van de aanpassing zelf. Individuele rechten en plichten worden geregeld via een persoonlijk contract met de Apache organisatie. Zowel de ontwikkelaars die aan de code werken, als de gebruikers die van zich laten horen, worden beschouwd als behorende tot de Wicket community.

---

<sup>16</sup> Dit in tegenstelling tot bijvoorbeeld Linux waar Linus Torvalds duidelijk het middelpunt is

### 4.3 Business Case Compiere



In het voorjaar van 2007 is er bij de bedrijfsunit Topicus Finance gekeken naar de automatiseringsbehoefte in de nabije toekomst. De unit bestaat op dat moment uit 5 personen. De verwachting is echter dat de unit binnen een jaar zal uitgroeien naar 12 personen. Daarna is nog verdere groei te verwachten. Om deze groei bij te kunnen houden, wordt er alvast gekeken naar de automatiseringsbehoefte die een dergelijke groei met zich mee zal brengen. Om vooruit te lopen op de gebeurtenissen wordt de implementatie van een ERP systeem overwogen. Tevens speelt de behoefte om ervaring op te doen met dergelijke systemen een rol.

#### 4.3.1 Eerste verkenning

Om tot meer inzicht van de mogelijkheden te komen, is een eerste verkenning uitgevoerd. Al snel worden de vier grootste randvoorwaarden duidelijk:

1. Het systeem moet goedkoop zijn, het liefst gratis;
2. Het systeem mag geen (grote) afhankelijkheden met zich meebrengen;
3. Het systeem moet de huidige manier van werken ondersteunen;
4. Het systeem moet projectmanagement ondersteunen.

De volgende actie is om te kijken wat er aan Open Source ERP voorhanden is. Zowel bij Google, Yahoo, als AltaVista levert de zoekstring "open source erp" direct een link naar de Compiere website<sup>17</sup> op. Ook via andere wegen, zoals via Zibb<sup>18</sup>, of via het Open Source jaarboek, komt Compiere steeds als eerste naar voren. Hoewel er ook andere Open Source ERP systemen bestaan (zoals OpenBravo, OpenTabs, of TinyERP), springt Compiere er iedere keer weer uit. Daarom wordt Compiere gekozen om nader te onderzoeken. Compiere wordt gevoerd door ComPiere inc. een bedrijf in Portland Oregon (VS). ComPiere inc. is een klein bedrijf, met een connectie met de Portland State University en met Partners over de hele wereld. In Nederland bestaat de partner vertegenwoordiging uit ActFact, H2-Consulting, en HintTech. Compiere staat op nummer 5 van de meest actieve communities op SourceForge. Compiere is terug te vinden op meerdere, onafhankelijke, Open Source sites en wordt daar beschouwd als een volwassen ERP oplossing voor het midden en kleinbedrijf. Qua volwassenheid staat Compiere op dezelfde hoogte als Linux. Het is dan ook niet vreemd dat er contacten zijn met Linus Thorvalds en andere Open Source professionals.

<sup>17</sup> <http://www.compiere.org/>

<sup>18</sup> <http://www.zibb.nl/>

### 4.3.2 Nader onderzoek

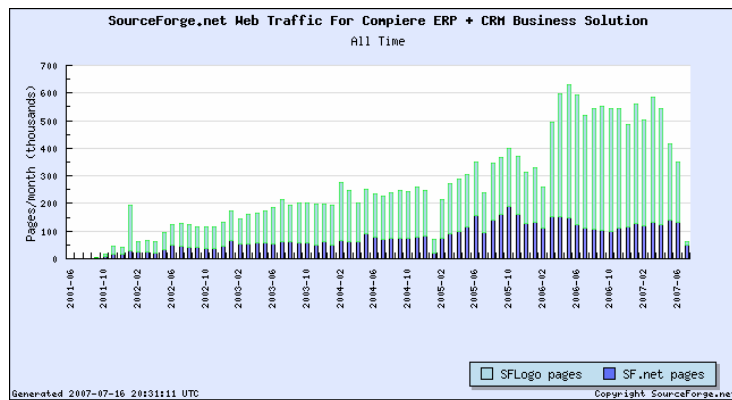
Compiere wordt geïnstalleerd om de functionaliteit te bekijken. Dit blijkt niet gemakkelijk te zijn. De installatie zelf verloopt weliswaar zonder problemen, ook de installatie van de benodigde Oracle database levert geen problemen op, maar het systeem is lastig in te richten. Dit omdat een handboek ontbreekt en het systeem geheel in het Engels is geïmplementeerd. Nu komt naar voren hoe goed het verdienmodel rondom Compiere is uitgedacht. Zowel het handboek als de Nederlandse vertaalbibliotheek blijken tegen betaling te verkrijgen. De kern van het systeem is voor iedereen gratis, is er (iets) meer gewenst, dan moet men daarvoor betalen. Na de installatie van de Nederlandse bibliotheek en met het handboek erbij verloopt de functionele verkenning voorspoedig. De functionaliteit die het systeem biedt is veelbelovend.

### 4.3.3 Check de techniek

De binnenkant van het systeem wordt bekeken. Er is intussen toegang tot de broncode gevraagd en gekregen. De broncode wordt gekopieerd en door een senior Java programmeur beoordeeld op techniek. Ook deze beoordeling valt positief uit. Het geheel ziet er netjes, overzichtelijk en vooral logisch uit. Ook staat er keurig in ieder bestand een verwijzing naar de licentie (GPL). De programmeur verklaart dat hij hier technisch wel op door zou kunnen gaan als dat nodig mocht zijn.

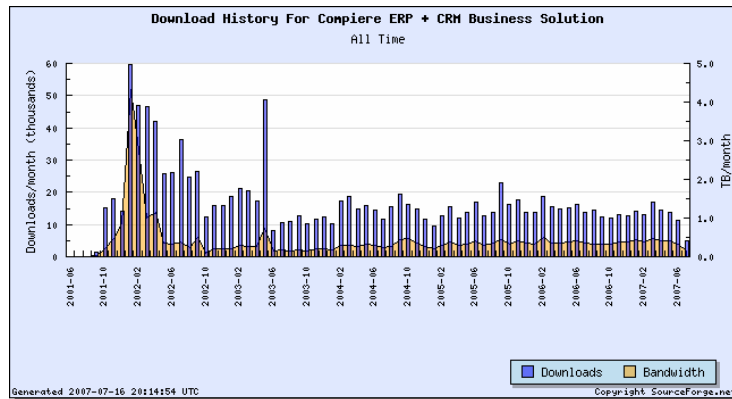
### 4.3.4 Check de community

De community website van Compiere<sup>19</sup> wordt bezocht. Op deze site is veel informatie te vinden over welke zaken er lopen, waar problemen zitten, hoe deze opgelost kunnen worden, etc. Door regelmatig de site te bekijken, is er ook een goed beeld te vormen van de community zelf. Al snel is duidelijk hoeveel personen er actief zijn (71) en wie de projectleider is (Jorg Janke). Ook zijn de hoeveelheid berichten die er op de fora staan (ruim 28000) en vele andere zaken interessant. De statistische informatie die SourceForge biedt, doet daar nog een schepje bovenop. Zo laat de algemene activiteit rondom de site een opgaande lijn zien. Kennelijk is er dus een gestaag toenemende interesse voor Compiere.

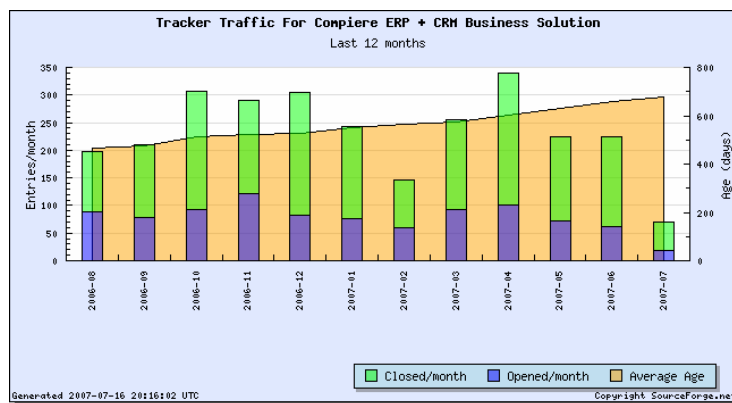


<sup>19</sup> [http://sourceforge.net/tracker/?group\\_id=29057](http://sourceforge.net/tracker/?group_id=29057)

Ook het aantal downloads van het systeem is stabiel (gemiddeld 15000 per maand).



Daarnaast worden ook de fouten die gemeld worden gestaag opgelost.



#### 4.3.5 Contact met de community

Tot slot wordt er contact opgenomen met de community. Tijdens dit contact wordt duidelijk dat de aangepaste code voorgelegd zal worden aan een team van Compiere ontwikkelaars. Deze zullen bepalen of de aangepaste code opgenomen wordt in het Open Source product.

Hiermee is tevens de laatste stap gezet van de verkenning van Compiere. Alles wijst erop dat Compiere een geschikte kandidaat is om op door te gaan. Als het project verder gaat zal het globale ontwerp de eerstvolgende stap zijn.

#### 4.4 Voorlopige conclusies

Het loont de moeite om de verkenning van een Open Source project stap voor stap uit te voeren. Voortschrijdend inzicht speelt namelijk een belangrijke rol.

Het is van belang om, behalve voor het product, ook voldoende aandacht te hebben voor het laten landen van het systeem in de organisatie.

Er is een verschil tussen de grote openheid van Open Source communities en gemeentelijke overheidsorganisaties. De laatste zijn behoorlijk gesloten. Het wordt dus moeilijk om voor overheden de support van een Open Source community te organiseren. Dit verklaart wellicht waarom Open Source bij de overheid nog niet zo van de grond gekomen is. Eigenlijk zou er een Community opgericht moeten worden die zich helemaal richt op de Open Source software die door de Nederlandse overheden gebruikt wordt.

Open Source stopt niet bij (de aflevering van) het product. Eigenlijk begint het dan pas goed. Vanaf dat moment komt de code beschikbaar voor de rest van de wereld en start het "Open Source proces"<sup>20</sup>.

Bij een project, dat tot doel heeft om het resultaat als Open Source aan te bieden, is het cruciaal om afspraken te maken over de mate van "OpenSource-hed"<sup>21</sup>. Ook is het van belang om helder vast te leggen welke partij de ondersteuning hiervan gaat verzorgen.

Communities gaan nogal verschillend om met hun interne organisatie. Feitelijk is hier geen eenduidigheid in te vinden.

Een goede community kan een goed werkend systeem maken met slechte code, door deze code steeds verder te verbeteren. Een slechte community met slechte code zal niet tot goede resultaten komen.

Referenties zijn aan het begin van een Open Source project niet zo belangrijk. Naarmate het project groeit en bekender wordt, zijn referenties wel van belang. Geïnteresseerden gaan namelijk in hun verkenningstraject juist op zoek naar deze referenties.

Op de community website is een schat aan informatie te vinden die helpt bij de beoordeling van een Open Source product en de community die erbij hoort.

Kritische succesfactoren laten zich lastig indelen aan de hand van de verschillende fasen van een Open Source project. Ze blijken ook invloed te hebben over de fasen heen. Het type project geeft ook maar weinig houvast. Zowel een project dat Open Source gebruikt, als een project dat Open Source oplevert, heeft te maken met dezelfde factoren. Deze factoren worden echter wel vanuit een verschillend perspectief benaderd. Verdere abstrahering is dus noodzakelijk om tot een valide indeling te kunnen komen.

---

<sup>20</sup> Zie bijlage

<sup>21</sup> Zie bijlage

## 5 Resultaten

Bestudering van het onderzochte materiaal leidt tot het inzicht dat er drie belangrijke domeinen te onderscheiden zijn: *factoren*, *personen*, en *thema's*. Met andere woorden: *welke* kritische succesfactoren kunnen er onderscheiden worden, *voor wie* zijn deze factoren belangrijk, en *welke ordening* is er hierin aan te brengen. Een belangrijke ondervinding is dat de gevonden factoren op verschillende momenten de aandacht hebben (of zouden moeten hebben!). Opvallend is daarbij dat het perspectief, bijvoorbeeld het tijdstip of de persoon, anders is, maar dat de betreffende succesfactor dezelfde blijkt te zijn. Daarom is het reëel om te stellen dat:

*De kritische succesfactoren moeten in iedere fase van een Open Source project (steeds) opnieuw beschouwd en geborgd worden.*

Er is hier dus duidelijk sprake van *iteratie*. In de volgende paragrafen worden de genoemde domeinen verder uitgewerkt en komen dus (eindelijk) de kritische succesfactoren en hun samenhang met de andere domeinen aan het licht.

### 5.1 Factoren, de kritische succesfactoren worden zichtbaar

Er is intussen voldoende onderzocht materiaal voorhanden om te kunnen zoeken naar de kritische succesfactoren zelf. De vraag is gesteld:

*Welke verschillende factoren zijn te onderscheiden, waarvan gesteld mag worden dat deze kritisch zijn voor succes.*

In onderstaande paragrafen wordt op een rij gezet welke kritische succesfactoren er naar voren komen. Er wordt beschreven welke eigenschappen deze factoren hebben en hoe ze het beste geborgd kunnen worden. Belangrijk is hierbij te begrijpen dat bij "systeem" of "onderhanden systeem" ruim geïnterpreteerd moet worden. Dus zowel een systeem dat als Open Source wordt aangeboden, als een gewenst systeem, maar ook een systeem dat nog gebouwd moet gaan worden of nog in aanbouw is.

#### 5.1.1 Algemene kennis over het systeem

Het is van belang om algemene kennis over Open Source te hebben. Historie en huidige stand van zaken worden belangrijk op het moment dat er met derden over Open Source gecommuniceerd wordt. Ook algemene kennis van het (onderhanden) systeem en mogelijke alternatieven is om dezelfde reden belangrijk. De kennis en ervaring met Open Source die er bij anderen bestaat, moet niet onderschat worden. Deze informatie komt beschikbaar als bekend is "waar het over gaat". Non-informatie, veroorzaakt door verholde onkunde, komt helaas vaak voor. De beste manier om deze factor als gebruiker te borgen is het lezen van publicaties (boeken, vakliteratuur, Internet, e.d.) en het voeren van gesprekken met mensen die bij Open Source projecten betrokken zijn. De aanbieder moet er voor zorgen dat algemene informatie over het systeem beschikbaar is op het Internet te vinden is met zoekmachines. Daarnaast biedt het toenemende aantal (overzicht)publicaties over Open Source ook goede mogelijkheden om het systeem onder de aandacht te brengen.

### 5.1.2 Functionele kennis over het systeem

Het is van belang om functionele kennis te hebben over het onderhanden systeem. Hierdoor kan er namelijk vergeleken worden. Vergelijkingen worden gemaakt met andere systemen, met punten uit de specificaties, of met een wensenlijst. Ook kan er bekeken worden of een systeem echt wel dat biedt wat nodig is. De aanbieder borgt deze factor door de specificaties van het systeem en een installeerbare versie aan te bieden. Daarnaast kunnen er handboeken en vertalingen beschikbaar worden gemaakt<sup>22</sup>. De gebruiker borgt de factor door de specificaties te bestuderen en het systeem te installeren en de mogelijkheden ervan te verkennen. Op die manier ontstaat er een goed beeld van de functionaliteit van het systeem. Handboeken en vertalingen kunnen dit proces versnellen.

### 5.1.3 Technische kennis over het systeem

Technische kennis van het systeem is nodig om de kwaliteit te kunnen bepalen. Hiervoor moet de broncode van het systeem toegankelijk zijn, zodat deze onderzocht kan worden. Om de inzichtelijkheid verder te vergroten moet de code voorzien zijn van commentaar. Daarnaast moet de opbouw modulair<sup>23</sup> zijn. Het geheel moet op een elegante manier zijn opgesteld; netjes overzichtelijk en logisch. Er kan een style guide<sup>24</sup> ter beschikking gesteld worden, zodat duidelijk wordt welke formattering er gebruikt wordt. De aanbieder borgt de factor door het bovenstaande door te voeren. De gebruiker moet al deze handgrepen gebruiken om inzicht in de code te verwerven. Het is zeer aan te raden om senior ontwikkelaars de code te laten beoordelen. Zij kunnen als geen ander bepalen of de code goed genoeg is om op verder te kunnen ontwikkelen.

### 5.1.4 Juridische kennis over het systeem

Juridische kennis over (licentiëring van) het systeem is van belang om te kunnen bepalen welke rechten en plichten er aan het systeem vastzitten. Daarnaast kan het heel goed zijn dat het systeem gebruik maakt van systemen van derden. Voor deze andere systemen kunnen ook weer andere licenties gelden. Om ervoor te zorgen dat er gewerkt kan worden, zonder inbreuk te maken op het copyright of de licentie van iemand anders, is het van belang goed op de hoogte te zijn de geldende licenties. Informatie hierover wordt door de aanbieder via de website van de community bekend gemaakt. Daarnaast horen (verwijzingen naar) de Open Source licenties in de bestanden van de gebruikte code te staan. De gebruiker moet echter meer weten dan alleen de plaats en het soort licentie dat van toepassing is. De gebruiker moet de inhoud en strekking van de licentie zelf goed begrijpen. De beste manier om dit te bereiken is de verschillende (Open Source) licenties te bestuderen en onderling te vergelijken<sup>25</sup>. Het is ook heel verhelderend om hierna ook een closed source<sup>26</sup> licentie te bekijken en de verschillen te onderkennen.

---

<sup>22</sup> Dit zijn tevens mogelijkheden voor alternatieve verdienmodellen

<sup>23</sup> Zie bijlage

<sup>24</sup> Zie bijlage

<sup>25</sup> Overzicht van de meest gebruikte Open Source Licenties is in de bijlage opgenomen

<sup>26</sup> Van closed source systemen is de code per definitie niet openbaar, daarom wordt het ook wel proprietary software genoemd. Meestal gaat het dan ook om commerciële producten. MSWord is een typisch closed source systeem

### 5.1.5 Organisatorische kennis over het systeem

Organisatorische kennis over de omgeving van het systeem is van belang om te kunnen inschatten hoe betrouwbaar de community van het systeem is. Dit speelt een rol bij de beslissing om de community te volgen of niet. Een gebruiker kan ook heel goed een eigen weg kiezen en het systeem zelf verder ontwikkelen. Bij het organiseren hiervan spelen de voorgaande factoren weer een rol. De leden van de community worden via het internet door de aanbieder bekend gemaakt. De activiteit van de community, zoals het aantal nieuwe versies van een systeem gedurende de tijd, of het aantal opgeloste bugs, wordt meestal op de Open Source projecten website (zoals SourceForge of Apache) aangegeven.

### 5.1.6 Kennis over beheer en onderhoud van het systeem

Kennis over beheer en onderhoud komt naar voren als de community gevolgd wordt. Het is dan belangrijk te weten hoe levensvatbaar het systeem is en hoe opvolgende versies aangeboden gaan worden. Ook de plaats waar het systeem geparkeerd wordt als er geen community voor is, moet bekend zijn. Daarnaast is het voor de gebruiker van belang om invloed te hebben op de ontwikkeling van het systeem. Ook voor de aanbieder is feedback vanuit de gebruikers noodzakelijk om fouten te achterhalen en nieuwe ontwikkelingen te kunnen bepalen. Daarom moet de aanbieder (of de supporter daarvan) er voor zorgen dat er faciliteiten als meldingsystemen, wensenlijsten, fora, mailinglists en dergelijke beschikbaar zijn.

### 5.1.7 Kennis over juridisch en politiek beleid

Hier speelt dat er voorkeursbeleid en mogelijk ook subsidies voor Open Source projecten georganiseerd worden. Om de mogelijke veranderingen in rechten en plichten te kunnen blijven inschatten, moeten zowel de aanbieder als gebruiker op de hoogte blijven van de juridisch en politieke ontwikkelingen op het gebied van Open Source. Vanzelfsprekend spelen hier wetgeving over copyrights en patenten een rol.

## 5.2 Wie hebben met kritische succesfactoren te maken

Opvallend is dat de kritische succesfactoren weliswaar voor iedereen gelden, maar dat het omgaan ermee voor ieder verschillend kan zijn. Het loont daarom de moeite om aandacht te besteden aan de vraag welke rol er precies binnen een Open Source project vervuld wordt. De vraag is gesteld:

*Welke ordening is er te maken van de verschillende groepen personen die direct betrokken zijn bij een Open Source project?*

In de volgende paragrafen worden de verschillende rollen verder uitgewerkt.

### 5.2.1 Aanbieders

Aanbieders zijn degenen die Open Source ontwikkelen en aanbieden. Aanbieders werken meestal vanuit eigen initiatief en hebben een persoonlijke motivatie, geld is minder een drijfveer. Deze persoonlijke motivatie blijkt zeer divers en zeker *niet* eenduidig. Er zijn ook commerciële organisaties die als aanbieder Open Source produceren. Vaak werken deze in opdracht, maar er zijn ook organisaties

die daadwerkelijk met Open Source ondernemen en daarin succesvol zijn. Deze organisaties zijn erin geslaagd een goed werkend verdienmodel op Open Source toe te passen.

Hoewel er ook solo aanbieders zijn, bestaan aanbieders in de meeste gevallen uit groepen. 6 personen wordt als het minimum beschouwd om als aanbieder goed te kunnen functioneren. Aanbieders zijn een belangrijk deel van de community.

### 5.2.2 Gebruikers

Gebruikers zijn degenen die Open Source inzetten voor persoonlijk gebruik of bij ondersteuning van een (hun) organisatie. Voorbeelden hiervan zijn het gebruik van Open Source datalagen als Hibernate<sup>27</sup> of het gebruik van Open Source kantoorpakketten als OpenOffice<sup>28</sup>. Gebruikers kunnen goed ingedeeld worden in twee groepen: Gebruikers die zich ook met de (verdere) ontwikkeling van het systeem bezighouden en gebruikers die dit niet doen. De eerste groep wordt overwegend beschouwd als onderdeel van de community.

### 5.2.3 Combinaties

Er ontstaan ook combinaties van gebruikers en aanbieders. Een organisatie kan eerst opdracht geven om een nieuw systeem te ontwikkelen. Vervolgens kan deze organisatie het ontwikkelde systeem als Open Source aanbieden. Ook door toepassing van de licentie ontstaan dergelijke combinaties. Een gebruiker kan gehouden worden om, als de code wordt aangepast, deze weer als aanbieder beschikbaar te stellen. Combinaties van aanbieders en gebruikers ontstaan ook. Een aanbieder kan een eigen Open Source systeem beschikbaar stellen. In een nieuwe versie wordt functionaliteit geïmplementeerd die (deels) door een ander Open Source systeem geleverd wordt. Het zal dus duidelijk zijn dat ieder zich goed bewust moet zijn van de rollen, en vaak van de bijbehorende (licentie) verplichtingen, die er vervuld worden.

## 5.3 Thema's, ordening van factoren

Tot slot wordt er gekeken of er ordening is aan te brengen in de gevonden factoren. Deze thematisering is van belang voor het afleiden van een methode. Daarom is de vraag gesteld:

*Hoe kunnen de gevonden kritische succesfactoren worden ingedeeld, zodat er een samenhangend geheel ontstaat?*

In de hiernavolgende paragrafen zijn de gevonden thema's benoemd en uitgewerkt.

---

<sup>27</sup> <http://www.hibernate.org/>

<sup>28</sup> <http://www.OpenOffice.org/>

### **5.3.1 Beschikbaarheid**

In de eerste plaats het thema *beschikbaarheid*. Nadruk ligt hier op het beschikbaar zijn van het systeem zelf, en informatie over het systeem. Het gaat hier om informatie over de functionaliteit, over *wat* het desbetreffende systeem kan.

### **5.3.2 Toegankelijkheid**

Daarnaast het thema *toegankelijkheid*. Hierbij ligt de nadruk op het inzicht in de binnenkant van het systeem. De benadering is hier technischer van aard. Het gaat hier om de technische en juridische kanten van het systeem, over *hoe* het desbetreffende systeem werkt.

### **5.3.3 Continuïteit**

Tot slot het thema *continuïteit*. Hierbij is inzicht in de mensen die invloed hebben op het systeem het belangrijkste. Er wordt dus de nadruk gelegd op menselijke aspecten. Het gaat hier vooral om de organisatorische omgeving, over *wie* er met het systeem te maken hebben en wie er invloed op hebben.

## 6 Methode

Na het uitwerken van de resultaten wordt er een algemeen geldende methode ontwikkeld. De bedoeling is dat deze methode voldoende houvast geeft om de onderzoeksvragen te kunnen beantwoorden. Tevens moet de methode handvaten aanbieden aan degenen die met Open Source aan het werk willen gaan. Nadat de methode is ontwikkeld, worden er een voorbeeld gegeven van het gebruik van de methode.

### 6.1 Ontwikkeling van de methode

Om de methode te realiseren is eerst het *platform*, het vierde domein, geïntroduceerd. Via het platform wordt informatie over de kritische succesfactoren ter beschikking gesteld. Vervolgens zijn alle resultaten schematisch geordend.

Hierna wordt er uit het schema het model gecreëerd. Dit model vormt de basis van de methode. Uitgaande van het model worden de stappen bepaald die achtereenvolgens gezet moeten worden om te komen tot (juiste) borging van de kritische succesfactoren.

#### 6.1.1 Het platform

Uit de resultaten zijn eerder de personen, de tijdstippen en de factoren gehaald die in een Open Source project een rol spelen. Nu worden uit de resultaten ook de platforms gehaald. Onder platform wordt verstaan het medium dat de informatie bevat. Met deze informatie kan de betreffende kritische succesfactor geborgd worden. Hierbij komt naar voren dat de platforms gebruikt worden door zowel de aanbieders als door de gebruikers. Het platform wordt dus vanuit twee *verschillende* kanten benaderd. Opvallend is dat het plaatsen en ophalen van informatie op het platform niet 1:1 overeenkomt met de indeling in aanbieders en gebruikers. Beide plaatsen informatie op de platforms en beide raadplegen deze informatie, al naar gelang de situatie. Dit onderstreept eens te meer dat het Open Source proces geen éénrichtingsverkeer is, maar altijd een *interactie* tussen meerdere partijen. Tevens blijkt hieruit de onmisbaarheid van de genoemde platforms. Als ze er niet zijn kan de interactie niet plaatsvinden. Daarmee is het creëren en onderhouden van de platforms dus een (meta)kritische succesfactor. De volgende vraag is gesteld:

*Waar kan informatie geplaatst of gehaald worden zodat de betreffende factor geborgd kan worden?*

In de volgende paragrafen wordt de gevonden platforms beschreven.

#### 6.1.2 Publicaties

Publicaties zijn belangrijk informatiedragers als het gaat om algemene kennis van Open Source. Er zijn veel boeken en digitale presentaties beschikbaar waar kennis en ervaring van anderen is vastgelegd. Veelal is de moeilijkheid om te kiezen welke publicaties er wel en welke niet gelezen moeten worden. Advies inwinnen bij anderen (zie contacten) kan hier zeer waardevol zijn.

### 6.1.3 Community website

Communities hebben vaak een eigen website. Hierop wordt allerlei informatie gepubliceerd over de lopende projecten van de community. In de meeste gevallen is de community website onderdeel van een website met meerdere Open Source projecten<sup>29</sup>.

### 6.1.4 Commerciële website(s)

Naast de community website bestaan er voor veel systemen ook commerciële websites. Dit zijn de sites waarop vaak aanvullende producten, zoals handboeken, vertalingen of uitbreidingen van het systeem te koop worden aangeboden. Deze sites zijn van organisaties die met Open Source ondernemen. Meestal verwijzen deze sites ook naar de community site.

### 6.1.5 Code repository

Via de community website kan er toegang gevraagd worden tot de code repository. Dit is het versiebeheer systeem waar de code van het betreffende project bewaard en onderhouden wordt (hier is dus de daadwerkelijke Open Source te vinden).

### 6.1.6 Contacten

Naarmate er meer met Open Source gewerkt wordt, zijn er dus ook steeds meer mensen te vinden die iets van Open Source afweten. Ook op politiek en juridisch gebied zijn er personen te vinden die goed op de hoogte zijn van specifieke onderwerpen van Open Source. Het loont de moeite om deze contacten te zoeken en te onderhouden.

## 6.2 Het schema

Op de bovenbeschreven manier ontstaat er een schema. Uit dit schema is af te lezen is *welke* kritische succesfactoren er op een *bepaald tijdstip* spelen. Ook is te zien voor *wie* en op *welke manier* deze factoren van belang zijn. Tot slot is er zichtbaar gemaakt *waar* de informatie te vinden is om de factor te borgen:

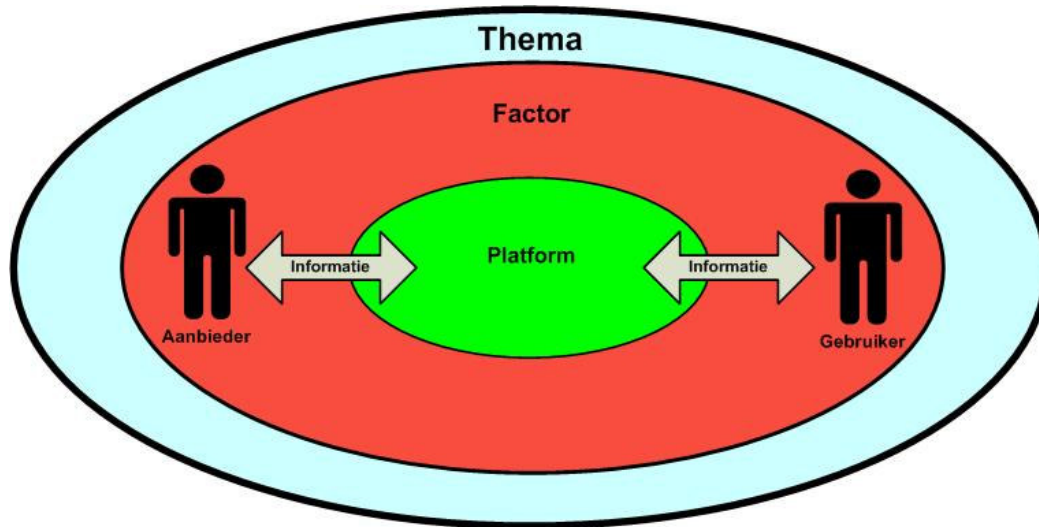
---

<sup>29</sup> Zoals: [www.sourceforge.net](http://www.sourceforge.net), <http://www.apache.org>, of <https://www.uitwisselplatform.nl/>

<b>Thema</b>	<b>Factor</b>	<b>Platform</b>	<b>Aanbieder</b>	<b>Gebruiker</b>
<b>Beschikbaarheid</b>	Algemene kennis over het systeem	Diverse publicaties & Contacten & Community Website	Algemene informatie in het publieke domein	Inzicht in de historie van Open Source
			Referenties naar projecten	Overzicht van beschikbare systemen
			Historie en laatste nieuws	
	Functionele kennis over het systeem		Specificaties	Overzicht van beschikbare functionaliteit
		Commerciële Website(s)	Installeerbaar systeem	Inzicht in bruikbaarheid van beschikbare functionaliteit
			Handboek	
	Vertalingen			
<b>Toegankelijkheid</b>	Technische kennis over het systeem	Code repository	Aanbieden van de broncode	Inzicht in kwaliteit van de code & Inzicht in kwaliteit van de eigen ontwikkelaars
			Commentaar in de code	
			Modulaire opbouw	
		Elegante code		
	Juridische kennis over het systeem	Code repository & Community Website	Style Guide	Inzicht in rechten en plichten
			Afhankelijkheid van andere systemen	
Licentie				
<b>Continuïteit</b>	Organisatorische kennis over het systeem	Code repository	Namen van de ontwikkelaars in de code	Inzicht in betrouwbaarheid van de community
		Community Website	Leden van de community	
			Activiteit van de community	
	Incrementele patches		Inzicht in levensvatbaarheid van het systeem	
	Weeshuis			
	Kennis over beheer en onderhoud van het systeem	Community Website	Meldingensysteem	Verbetering van het systeem
			Wensenlijst	Invloed op ontwikkeling
			Forum	
		Mailinglist		
	Kennis over (Inter)nationaal politiek beleid en wetgeving	Diverse publicaties & Contacten	Gebruikmaken van aanbestedingen en subsidies	Inspelen op voorkeursbeleid
			Inzicht in geldigheid licenties	Inzicht in verandering van verplichtingen

### 6.3 Het model

Uit het schema is een model afgeleid, dat als basis dient voor de methode. Dit model bestaat uit een aantal concentrische cirkels die de eerste drie domeinen (thema, factor, en platform) symboliseren. Daarnaast staan er in het model symbolen voor het vierde domein (de aanbieder en de gebruiker).



Het model geeft weer dat er, van buiten naar binnen, eerst gekeken moet worden naar het thema wat in een bepaalde fase van het project speelt. Vervolgens wordt er gekeken welke kritische succesfactor er op dit tijdstip een rol speelt. Daarna wordt er gekeken via welk platform de benodigde informatie neergezet, dan wel opgehaald kan worden, om de betreffende factor te borgen. Er is al eerder opgemerkt dat zowel de aanbieder als de gebruiker, op verschillende tijdstippen, informatie van het platform afhalen en er op zetten.

Dus achtereenvolgens:

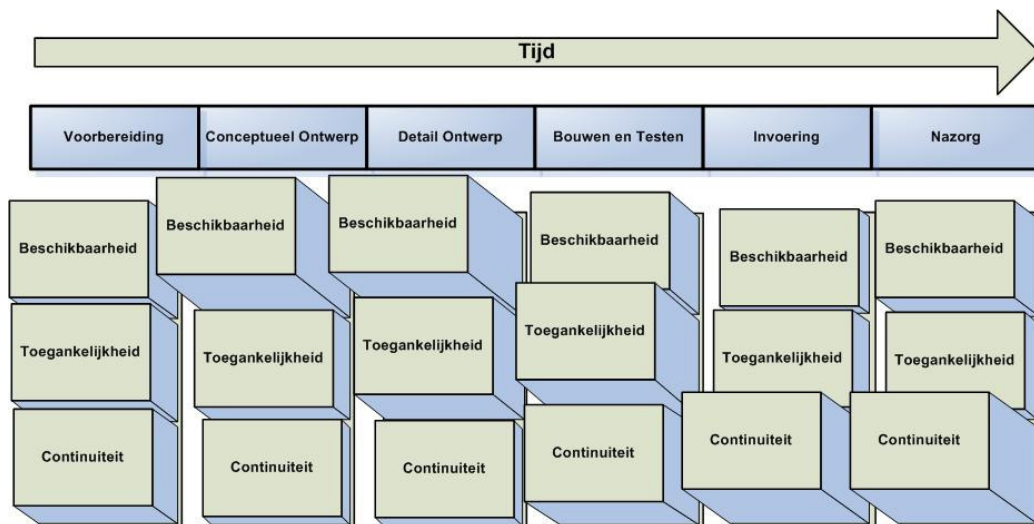
1. Bepaal het tijdstip waarin het project verkeert;
2. Bepaal het thema wat op dat moment op het project van toepassing is;
3. Bepaal de factor die op dit tijdstip kritisch is;
4. Bepaal de eigen positie binnen het project;
5. Bepaal via welk platform deze factor geborgd kan worden;
6. Haal de informatie van het platform;
7. Gebruik de informatie bij de borging van de betreffende factor;
8. Beschouw de geborgde factor in verhouding met de andere factoren.

## 6.4 Gebruik van de methode

Niet alle kritische succesfactoren zijn op hetzelfde moment van belang. Bij iedere fase van een Open Source project moet weliswaar opnieuw gekeken worden of alle succesfactoren nog wel goed geborgd zijn. Maar het zwaartepunt van de aandacht verschuift over de verschillende fasen van het Open Source project heen. Om dit inzichtelijk te maken is er een voorbeeld opgesteld. Het voorbeeld behandelt de implementatie van een Open Source ERP pakket. Hiervoor is het schema van het klassieke implementatietraject van een ERP pakket genomen. Aan dit schema zijn de verschillende thema's van de methode gerelateerd. Al naar gelang een thema meer of minder aandacht vergt, is het groter of kleiner weergegeven.

### 6.4.1 Voorbeeld: de implementatie Open Source ERP systeem

In het onderstaande schema is het verschuiven van de aandacht gedurende de verschillende fasen van het ERP implementatietraject aangegeven.

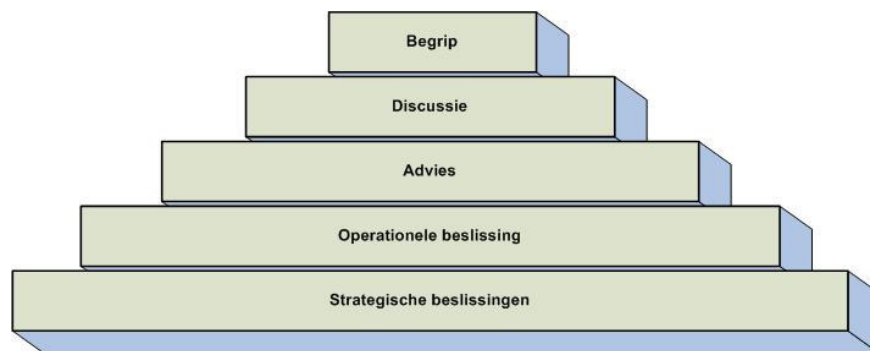


In de eerste fase van het ERP implementatietraject (voorbereiding) is het zwaartepunt gelijk verdeeld. Er wordt allerlei informatie verzameld om te kunnen komen tot een beslissing *welk* Open Source ERP pakket er gekozen gaat worden. Is die beslissing eenmaal genomen dan wordt er globaal gekeken of alle kritische succesfactoren in de toekomst geborgd kunnen worden. Vandaar dat nu alle thema's evenveel aandacht krijgen. In de fasen daarna (conceptueel- en detailontwerp) ligt het zwaartepunt van de aandacht bij *beschikbaarheid* omdat er gedetailleerde informatie moet zijn om te kunnen bepalen welke functionaliteit er al in het pakket zit, wat aangepast of eventueel nieuw gebouwd moet worden. Tijdens de daaropvolgende fase (bouwen en testen) krijgt het thema *toegankelijkheid* natuurlijk veel aandacht, er wordt namelijk met de code van het pakket zelf gewerkt. Er moet echter ook voldoende aan *continuïteit* gedaan worden. Dit omdat de community een rol gaat spelen. Welke aanpassingen worden door de community overgenomen? Welke aanpassingen kunnen eventueel

door de community zelf worden uitgevoerd? Hoe moet de community benaderd worden om dit soort dingen voor elkaar te krijgen? Tijdens de volgende fase (invoering) ligt het zwaartepunt duidelijk helemaal bij *continuïteit*. Er worden door testers en gebruikers fouten gevonden, maar er komen ook nieuwe eisen en wensen naar voren. Deze worden gecommuniceerd met de community, of met eigen ontwikkelaars. Tijdens de laatste fase (nazorg) is het zaak om, behalve *continuïteit*, weer voldoende aandacht te besteden aan *beschikbaarheid*. Opgedane ervaring, nieuwe functionaliteit, wellicht nieuwe referenties. Allemaal zaken die weer beschikbaar gemaakt moeten worden voor de rest van de wereld.

#### 6.4.2 Wanneer is een factor voldoende geborgd

Bij het toepassen van de methode komt de vraag naar voren wanneer een kritische succesfactor voldoende geborgd is. Met andere woorden, wanneer is er genoeg aandacht aan besteed om het kritische karakter ervan weg te nemen. Om de mate van borging van een factor te kunnen bepalen, kan er het beste gekeken worden naar het risico dat er mee genomen kan worden. Om hier vorm aan te geven is de *borgingspiramide* opgesteld.



De top van deze piramide wordt gevormd door het risico dat met de minste inspanning veilig genomen kan worden. Vervolgens zijn de lagere treden van de piramide activiteiten met steeds grotere risico's. Dus achtereenvolgens worden de vragen gesteld:

- Is er voldoende aandacht besteed zodat er voldoende begrip is over de factor?
- Is er voldoende aandacht besteed zodat er met anderen een zinnige discussie is te voeren over de factor?
- Is er voldoende aandacht besteed zodat er een goed advies over de factor uitgebracht kan worden?
- Is er voldoende aandacht besteed om over de factor een operationele beslissing te nemen?
- Is er voldoende aandacht besteed om over de factor een strategische beslissing te nemen?

Op het moment dat er een negatief antwoord gegeven wordt, is de mate van borging bepaald.

**Bijvoorbeeld:** Er is aandacht besteed aan het verkrijgen van technische kennis over een systeem (thema toegankelijkheid). De code is uit de code repository gekopieerd en een senior programmeur heeft er een oordeel over gegeven.

Mogelijke antwoorden kunnen zijn:

- Is er voldoende aandacht besteed zodat er voldoende begrip is over de factor?  
*Ja, er is bekend dat code netjes en overzichtelijk moet zijn opgebouwd.*
- Is er voldoende aandacht besteed zodat er met anderen een zinnige discussie is te voeren over de factor?  
*Ja, er kan aangegeven en onderbouwd worden dat de code modulair van opzet is, dat excepties correct worden afgevangen en dat er object oriëntatie is toegepast.*
- Is er voldoende aandacht besteed zodat er een goed advies over de factor uitgebracht kan worden?  
*Ja, er is vastgesteld dat deze code technisch de beste is van alle onderzochte systemen tot nu toe.*
- Is er voldoende aandacht besteed om over de factor een operationele beslissing te nemen?  
*Ja, de code is zo inzichtelijk dat het risico genomen kan worden om hier een pilot mee te beginnen.*
- Is er voldoende aandacht besteed om over de factor een strategische beslissing te nemen?  
*Nee, er is te weinig commentaar in de code gevonden, hierdoor is het risico te groot om er meteen de gehele kantoorautomatisering op te baseren.*

De factor *technische kennis over een systeem* is dus geborgd tot en met het niveau van operationele beslissingen.

## 7 Conclusie en aanbevelingen

Open Source is de afgelopen jaren tot een wijd verbreid, maar ook enigszins ondoorzichtig begrip geworden. De reden hiervan is dat Open Source niet gedragen wordt door één bedrijf of organisatie. Er zijn miljoenen mensen over de hele wereld actief met de productie en het gebruik van Open Source. Daardoor zijn er evenzoveel manieren om het te definiëren en ermee om te gaan. Dit houdt in dat Open Source moeilijk te analyseren is. Er blijken (nog) geen algemene regels te bestaan over hoe er met Open Source omgegaan moet worden. Het is echter wel mogelijk om een aantal kritische succesfactoren te onderkennen en handvaten te bieden hoe deze geborgd kunnen worden.

### 7.1 Terug naar de onderzoeksvragen

#### ***Wat is Open Source?***

Zoals in paragraaf 2.4 al naar voren is gekomen:

*Open Source is software waarvan de broncode, tegen de in de licentie gestelde voorwaarden, beschikbaar is gemaakt voor een bredere groep dan alleen de houder van het intellectuele eigendom.*

#### ***Wat zijn Open Source projecten?***

Zoals in paragraaf 2.5 al naar voren is gekomen:

*Een Open Source project is een automatiseringsproject waarin Open Source software gebruikt wordt, of wat Open Source software oplevert.*

#### ***Zijn alle Open Source projecten gelijk van aard?***

Nee, uit het onderzoek komt naar voren dat Open Source projecten vanuit een aanbieders- of een gebruikersrol uitgevoerd kunnen worden. Ook combinaties zijn mogelijk. Daarnaast kan de functionaliteit en de omvang van de Open Source systemen per project verschillen. Open Source projecten zijn dus niet gelijk van aard.

#### ***Bij welke fasen van een project spelen kritische succesfactoren een rol?***

Zoals uit de beschouwing van de onderzoeksresultaten blijkt, moeten de kritische succesfactoren in iedere fase van een Open Source project opnieuw beschouwd en geborgd worden.

#### ***Wanneer is er sprake van een succesvol project?***

Hier is al eerder onderzoek naar gedaan. Deze vraag leidde tot de volgende definitie van een succesvol project<sup>30</sup> :

“Bij de uitvoering van een ICT project is de kwaliteitsbeleving van de klant en het behaalde financiële rendement van de leverancier, maatgevend voor het succes van het project.”

Een project waarvan de klant en de leverancier blij zijn geworden is dus succesvol.

Het kan dus heel goed dat een (Open Source) project over tijd en budget heen gaat maar, omdat het wel goed bijdraagt aan de bedrijfsvoering van de klant, toch succesvol is.

---

<sup>30</sup> Zie bijlage

**Wanneer zijn succesfactoren kritisch?**

Succesfactoren zijn kritisch als er, bij het niet borgen ervan, serieus gevaar ontstaat voor het succes van het project. Dus als de gebruiker (klant) of de aanbieder (leverancier) er niet blij van dreigen te worden.

**Bestaat er onderscheid in de mate van het kritisch zijn van de succesfactoren?**

Nee, hoewel er wel onderscheid is in hoe een ongeborgde succesfactor (en de gevolgen daarvan) achteraf nog recht te zetten is.

**Is het borgen van deze factoren te incorporeren in een project?**

Ja, mits deze incorporatie integraal onderdeel is van de faseplanning

**Is er een methode te ontwikkelen die het bepalen en borgen van deze factoren voor de projectleiding makkelijker kan maken?**

Ja, deze methode is ontwikkeld als onderdeel van deze afstudeerscriptie

**7.2 Beantwoording van de hoofdvraag****Wat zijn kritische succesfactoren bij Open Source projecten en hoe kunnen deze geborgd worden ?**

Kritische succesfactoren bij Open Source projecten zijn die factoren die, mits geborgd, belangrijk bijdragen aan het succes van het Open Source project. Zijn ze echter niet geborgd, dan kunnen ze tot het mislukken van het project leiden. Er zijn zeven kritische succesfactoren te onderkennen te weten:

*Algemene kennis over het systeem;*

*Functionele kennis over het systeem;*

*Technische kennis over het systeem;*

*Juridische kennis over het systeem;*

*Organisatorische kennis over het systeem;*

*Kennis over beheer en onderhoud van het systeem;*

*Kennis over (Inter)Nationaal politiek beleid en wetgeving.*

Deze kunnen onderverdeeld worden in drie thema's te weten:

*Beschikbaarheid;*

*Toegankelijkheid;*

*Continuïteit.*

Deze factoren moeten in iedere fase van een project met Open Source opnieuw beschouwd en geborgd worden. Dit wordt gerealiseerd door de ontwikkelde methode iteratief toe te passen bij iedere (nieuwe) fase van een Open Source project.

### 7.3 Aanbevelingen voor nader onderzoek

#### 7.3.1 Het borgen van de factoren

Tijdens het onderzoek komt geen éénduidige manier naar voren, om het borgen van de factoren te realiseren. De in paragraaf 5.1 beschreven manieren van borgen, komen direct uit de praktijk. Het is interessant om hier nader onderzoek naar te doen. Opvallend is dat in vrijwel alle gevallen informatie gezocht, dan wel beschikbaar gemaakt moet worden. Op deze informatie worden belangrijke beslissingen gebaseerd. Wellicht is dit een aangrijpingspunt en kan er onderzocht worden of hier op een hoger niveau samenhang in te ontdekken is.

#### 7.3.2 De mate van borging van de factoren

Tijdens de ontwikkeling van het model komt de vraag naar voren wanneer een kritische succesfactor voldoende geborgd is. Het blijkt lastig te bepalen wanneer het kritische karakter ervan weg is genomen. Op deze vraag is (nog) geen kwantitatief antwoord gevonden. Hoewel in de vorige paragraaf al is aan gegeven dat het meestal gaat om het ophalen of plaatsen van informatie en het nemen van beslissingen op grond van die informatie, is er (nog) geen manier gevonden om dit te kwantificeren.

Om toch kwalitatief te kunnen bepalen of er voldoende aan borging is gedaan, is in paragraaf 6.4 de *borgingspiramide* geïntroduceerd.

Hoewel de *borgingspiramide* enige houvast biedt bij het bepalen van de mate van borging van een kritische succesfactor, blijft er toch onzekerheid bestaan. In die zin is het interessant om de mate van borging, en hoe die bereikt kan worden, verder te onderzoeken.

## 8 Bijlagen

### 8.1 Recente ontwikkelingen

In de paragrafen hierna wordt aangegeven waarom Open Source een zeer dynamisch karakter heeft. Er zijn iedere dag ontwikkelingen die de moeite waard zijn om te volgen. Een drietal van deze ontwikkelingen is beschreven.

#### 8.1.1 Nationaal politiek beleid

In november 2002 schreef de Tweede Kamer geschiedenis door unaniem de motie Vendrik te aanvaarden. Deze motie schoof een politiek fundament onder het belang van de Nederlandse overheden bij het gebruik van Open Source. Hoewel de uitvoering van de motie aan veel kanten nog te wensen overlaat, kennelijk hebben de regeringen Balkenende andere belangen, is het een duidelijk statement dat in Nederland Open Source voorrang hoort te krijgen:

*Tweede Kamer, Vergaderjaar 2002/2003, 28 600 XIII, nr.30*

*De Kamer, gehoord de beraadslaging, constaterende, dat software een cruciale rol speelt in de kennissamenleving; voorts constaterende, dat de aanbodzijde van de softwaremarkt op dit moment sterk geconcentreerd is en het veranderen van leverancier vaak hoge overstapkosten met zich brengt; van mening, dat dit de mededinging beperkt en de samenleving niet optimaal profiteert van de mogelijkheden die software biedt; verzoekt de regering zich maximaal in te zetten om hier verbetering in aan te brengen; verzoekt voorts de regering ervoor te zorgen dat in 2006 alle door de publieke sector gebruikte software aan de open standaarden voldoet; verzoekt voorts de regering actief de verspreiding en de ontwikkeling van software met een open broncode (Open Source Software) in de publieke sector te stimuleren en hiervoor concrete en ambitieuze doelstellingen te formuleren, en gaat over tot de orde van de dag.*

#### 8.1.2 Europese politiek, patenten en belangen

Tijdens de voorbereidingen voor deze opdracht werd op 24 september 2003 door de Europese Commissie het wetsvoorstel aangaande softwarepatenten aangenomen. In dit voorstel zou geregeld worden hoe uitvindingen in software gepatenteerd kon worden. De uitwerking van het voorstel strandde echter op 6 juli 2005 op het Europese Parlement dat, boos over het weinig transparante optreden van de Eurocommissarissen en de raad van Europa, het voorstel van tafel veegde. Rondom het voorstel is veel commotie geweest. Niet in de laatste plaats speelde het (on)democratische gehalte en de grote, vaak tegenstrijdige, belangen van veel organisaties een belangrijke rol. Hoe verdere Europese wetgeving tot stand zal komen, en of democratische principes of de belangen van een klein aantal multinationals daarin leidraad zijn, is zeker van belang. De uitwerking van deze wetgeving zal ongetwijfeld effect op software ontwikkeling hebben. Budgetten voor ontwikkeling zouden wel eens voor een belangrijk deel gespendeerd (moeten) worden aan juridische verankering van de gebruikte technische constructies in de software producten. Alleen al het managen van het risico op (onbewust gebruik van) gepatenteerde constructies, zal aanzienlijke inspanningen vergen. Daarom is het interessant om te kijken of het gebruik van Open Source dit risico kan beïnvloeden. Daarnaast is het inzichtelijk hebben van juridische aansprakelijkheid, voortkomend uit een mogelijk patent, natuurlijk een

kritische succesfactor van de eerste orde bij ieder software project, dus ook voor ieder Open Source project. In de volgende paragraaf wordt alvast een voorzet gegeven

Meer informatie over de Europese patenten en octrooien is te vinden op: <http://eupat.ffii.org/>

### **8.1.3 Dreigen met patenten en aantonen van "desbewustzijn"**

Op het gebied van recente ontwikkelingen is ook de samenwerking tussen Microsoft en Novell tekenend voor wat er zich afspeelt op het gebied van Open Source. Microsoft is lange tijd afwijzend geweest tegenover Open Source. De omarming van Novell's Suse Linux wordt enerzijds gezien als een erkenning van de onontkoombaarheid van Open Source door Microsoft. Anderzijds beweert Microsoft nu te kunnen aantonen dat Open Source systemen inbreuk maken op een behoorlijk aantal (235) van hun patenten. Hierdoor ontstaat een interessante situatie. Weliswaar lukt het Microsoft om onrust te creëren op de almaar groeiende markt voor Open Source producten. Maar er kan niet eeuwig bedreigd worden, als er niet ook passende juridische actie wordt ondernomen. In Nederland is het bijvoorbeeld wel degelijk strafbaar om iemand anders een dief te blijven noemen zonder daadwerkelijk aangifte te doen. Maar er speelt meer. Zoveel meer zelfs, dat een kritische succesfactor ermee geborgd kan worden.

Stel de implementatie van een strategisch stuk Open Source software wordt overwogen. De eerder beschreven dreiging van Microsoft is een kritische succesfactor (Thema: Toegankelijkheid, Factor: Juridische kennis over het systeem, Belang van de gebruiker: Inzicht in rechten en plichten). Het is dus van belang om te weten welke octrooien er van toepassing zijn op het beoogde stuk strategische software. Is die informatie eenmaal bekend, dan kunnen hierop beslissingen worden genomen. Microsoft zal deze informatie niet graag verstrekken, als de octrooien eenmaal bekend zijn is de dreiging die ervan uitgaat verdwenen. Nu schrijft de wet op dit punt voor:

Artikel 70 lid 3 ROW (Rijks Octrooi Wet)

*Schadevergoeding kan slechts worden gevorderd van hem, die de handelingen desbewust verricht. Men wordt in elk geval geacht desbewust te hebben gehandeld, indien de inbreuk is gepleegd na verloop van dertig dagen, nadat men bij deurwaardersexploit op de strijd tussen de handelingen en het octrooi is gewezen.*

Ook regelgeving vanuit de WTO (World Trade Organization), vervat in het TRIPS verdrag (Agreement on Trade Related Aspects of Intellectual Property Rights) doet hier uitspraak over:

Artikel 45 lid 1 TRIPS-verdrag

*The judicial authorities shall have the authority to order the infringer to pay the right holder damages adequate to compensate for the injury the right holder has suffered because of an infringement of that person's intellectual property right by an infringer who knowingly, or with reasonable grounds to know, engaged in infringing activity.*

Er kan dus slechts schadevergoeding worden gevorderd als iemand weet, desbewust is, of in redelijkheid desbewust had kunnen zijn, dat er inbreuk

gemaakt wordt op een octrooi van iemand anders. Het gaat er dus om desbewust te worden van eventuele octrooiën. Er kan dus heel goed een brief opgesteld worden aan de juridische afdeling van Microsoft. In deze brief staat het verzoek om (als het even kan via een aangetekend stuk of deurwaardersexploit) aan te geven welke van de 235 octrooiën er precies van toepassing zouden zijn. Indien Microsoft hier niet binnen een redelijke termijn op antwoord, kan er in redelijkheid ook niet gesteld worden dat iemand desbewust zou kunnen zijn van een mogelijke schending van een Microsoft octrooi. Gegeven de uitspraken van Microsoft zelf, is er dan immers alles aan gedaan om desbewust te worden.

Met andere woorden, dreigen met patenten is één ding, gehouden worden aan de fatsoenlijke uitvoering ervan is een tweede.

## 8.2 De meest gebruikte Open Source licenties

Licentie	Beschrijving
General Public License (GPL)	Dit is de meest gebruikte licentie. 70% van de Open Source projecten wordt uitgebracht onder deze licentie. Belangrijkste eigenschap is dat de licentie vastlegt dat modificaties in de oorspronkelijke code, van de software die verder verspreid wordt (!), zelf ook onder de GPL openbaar gemaakt moeten worden.  <a href="http://www.opensource.org/licenses/gpl-license.php">http://www.opensource.org/licenses/gpl-license.php</a>
Berkely Software Distribution (BSD)	Bij de BSD is het niet verplicht om aangepaste code openbaar te maken bij verspreiding. De BSD legt wel vast dat er naar copyright verwezen moet worden en dat er een (niet) aansprakelijkheidsclausule opgenomen moet worden.  <a href="http://www.opensource.org/licenses/bsd-license.php">http://www.opensource.org/licenses/bsd-license.php</a>
Mozilla Public License	De MPL legt onder andere vast dat er mogelijkheid is om een commerciële (niet Open Source) versie van de software uitgebracht mag worden. Mozilla Firefox valt onder de MPL.  <a href="http://www.opensource.org/licenses/mozilla1.1.php">http://www.opensource.org/licenses/mozilla1.1.php</a>

Een uitstekend overzicht van de meeste Open Source licenties, inclusief de letterlijke tekst ervan, is te vinden op:

<http://www.opensource.org/licenses/alphabetical>

### 8.3 Voetnoten en nadere informatie

#### **Voetnoot 3 - Hoofdstuk 2,:**

Worldwide Open Source Software Business Models 2007–2011 Forecast:

A Preliminary View

<http://www.idc.com/getdoc.jsp?containerId=206681>

Onderzoek uitgevoerd door: Matthew Lawton, Robert Notarfonzo mei 2007

IDC Framingham, USA

<http://www.idc.com>

#### **Voetnoot 4 - Hoofdstuk 2:**

Samenhang tussen Minix, ontwikkeld voor onderwijsdoeleinden door prof. A.S. Tannenbaum (<http://www.cs.vu.nl/~ast/>) en Linux, gestart door Linus Torvalds.

Meer Info:

[http://www.softpanorama.org/People/Torvalds/Finland\\_period/minix\\_as\\_a\\_precursor\\_of\\_linux.shtml](http://www.softpanorama.org/People/Torvalds/Finland_period/minix_as_a_precursor_of_linux.shtml)

#### **Voetnoot 6 - Hoofdstuk 2, paragraaf 2.4:**

OSI, Open Source Initiative is als non profit organisatie in 1997 opgericht door Eric S. Raymond en Bruce Perens. OSI hanteert de OSD, de Open Source Definition:

1. De licentie mag niemand verbieden de software gratis weg te geven of te verkopen.
2. Het programma moet de broncode bevatten en moet verspreiding toestaan, zowel als broncode als in gecompileerde vorm.
3. De licentie moet aanpassingen toestaan, en ook van het originele programma afgeleide werken.
4. De licentie mag verspreiding van aangepaste broncode alleen verbieden als de licentie wel de verspreiding van zogenaamde patchbestanden bij de originele broncode toestaat, met het doel het programma aan te passen bij het bouwen ervan.
5. De licentie mag niet discrimineren tegen welke persoon of groep personen dan ook.
6. De licentie mag niemand verhinderen het programma te gebruiken in een bepaald toepassingsgebied.
7. De rechten verbonden aan het programma moeten gelden voor iedereen aan wie het programma gedistribueerd wordt.
8. De rechten die bij het programma horen, mogen niet afhankelijk zijn van het feit dat het programma deel uitmaakt van een bepaalde software distributie.
9. De licentie mag geen beperkingen stellen aan andere software die is verspreid samen met het betreffende programma.
10. Geen van de bepalingen van de licentie mag slaan op bepaalde technologie of interface stijl.

#### **Voetnoot 8 - Hoofdstuk 3, paragraaf 3.1.1:**

Fetchmail zorgt voor de (re)routing van mail die ontvangen wordt over het Internet en vervolgens via een lokaal netwerk naar de computer gestuurd moet worden waar de desbetreffende geadresseerde op dat moment (!) aan het werk is. De kneep zit 'm bij dit systeem in de heradressering die moet plaatsvinden.

**Voetnoot 9 - Hoofdstuk 3, paragraaf 3.1.1:**

Een bazaar is een oosterse markt. Veelal gevormd op een marktplein of een overdekte ruimte. In deze ruimte nemen de handelaren een plaatsje in van waaruit zij hun waren aan de man brengen. Omdat de plaatsjes onderling worden vastgesteld weten de meeste handelaren waar hun collega's te vinden zijn. Stel dus dat één handelaar in groente wegvalt (bijvoorbeeld door een familieaangelegenheid) dan zullen zijn collega's een klant gemakkelijk naar een andere groentehandelaar kunnen verwijzen. Zij kunnen in ieder geval de richting aangeven waarin gezocht moet worden. Een bazaar heeft dus een sterk zelfregelend vermogen en is daardoor goed te vergelijken met een Open Source Community.

**Voetnoot 10 - Hoofdstuk 4, paragraaf 3.1.4:**

Michael Robertson richtte in 2002 het bedrijf Lindows op, het product Lindows is een distributie(!) van Linux die erg op Windows van Microsoft lijkt. Bovendien konden de meeste Windows systemen er op draaien. Het was dan ook inderdaad de bedoeling om een goed alternatief voor Windows te bieden. De rechter besloot in 2004 dat de naam Lindows te veel leek op Windows en verbood deze naamgeving. Sindsdien is deze linux distributie op de markt onder de naam Linspire.

**Voetnoot 12 - Hoofdstuk 4, paragraaf 4.1:**

Artikel van Rik Sanders in de Computable van 4 april 2003  
Hierin wordt beschreven hoe Almere een Open Source Midoffice laat ontwikkelen door Emaxx uit Enschede:

<http://www.computable.nl/artikel.jsp?archive=1&id=1391108>

**Voetnoot 13 - Hoofdstuk 4, paragraaf 4.1.3:**

GPL, Gnu Public License

Één van de bekendste licenties waaronder Open Source uitgebracht wordt.  
Een dergelijke licentie waarborgt de rechten (en plichten) van de gebruiker.

**Voetnoot 15 - Hoofdstuk 4, paragraaf 4.2:**

Meer info over het ontstaan van Wicket bij SourceForge:

<http://wicket.sourceforge.net/Introduction.html>

Meer informatie over de huidige stand van zaken bij Apache:

<http://incubator.apache.org/wicket/planet-wicket.html>

**Voetnoot 20 - Hoofdstuk 4, paragraaf 4.4:**

Het "Open Source proces" wordt hier gezien als de (iteratieve) interactie tussen de gebruikers van de software en de community ervan. De gebruikers melden de gevonden fouten in de functionaliteit van de software en maken hun wensen kenbaar. De leden van de community kiezen hieruit de onderdelen waar ze aan willen werken en realiseren zo een nieuwe versie van de software. Deze komt weer tot de beschikking van de gebruikers, waarna het proces zich herhaalt.

**Voetnoot 21 - Hoofdstuk 4, paragraaf 4.4:**

De mate van "Open Source-heid" is een verbastering van de mate waarin er aan de definitie van Open Source voldaan wordt. Dus de mate waarin *de broncode, tegen de in de licentie gestelde voorwaarden, beschikbaar is gemaakt voor een bredere groep dan alleen de houder van het intellectuele eigendom*. Hier is namelijk wel wat voor nodig (zie hoofdstuk "De Methode").

**Voetnoot 23 - Hoofdstuk 5, paragraaf 5.1.3:**

Modulaire opbouw wil zeggen dat de code is opgesplitst in op zichzelf staande onderdelen die onderling met elkaar communiceren. Op die manier kan een verandering in de functionaliteit, doorgevoerd worden door slechts één module aan te passen. Dit heeft het grote voordeel dat niet alle code opnieuw bewerkt en gecontroleerd hoeft te worden.

**Voetnoot 24 - Hoofdstuk 5, paragraaf 5.1.3:**

Een style guide is een document waarin is vastgelegd op welke manier de code geschreven moet worden. Er staat in hoe er met naamgeving van variabelen moet worden omgegaan, hoe het inspringen hoort te gaan, etc.

**Voetnoot 30 - Hoofdstuk 7, paragraaf 7.1:**

"Welke factoren bepalen het succes van ICT projecten en wat kunnen we daar van leren?"

Project Informatiemanagement

Minor Bedrijfskundige Informatica - 4e jaars DT

I. Adams, F. Beunder, M.N. Bierman, I. Muller, B.J. Verweij

ICA Juni 2006

#### 8.4 Geraadpleegde literatuur

The Cathedral & the Bazaar

Eric S. Raymond

O'Reilly 1999

ISBN 1-56592-724-9

Gelezen als internetversie op :

<http://www.catb.org/~esr/writings/cathedral-bazaar/>

Nederlandse versie intussen beschikbaar op:

<http://www.opensource.nl/bazaar.html>

Understanding Open Source communities

An organizational perspective

Ruben van Wendel de Joode

Proefschrift, in 2005 in eigen beheer uitgegeven onder

ISBN: 90-5638-138-5

Trends in IT 2004| 2005

Peter Noordam, Aart van der Vlist, Barry Derksen

IT Trends institute, tenHagenStam september 2004

ISBN 90-440-1021-2

Open Source

The unauthorized white papers

Donald K. Rosenberg

IDG Books Worldwide inc. 2000

ISBN: 0-7645-4660-0

Trends in IT 2006| 2007

Peter Noordam, Aart van der Vlist, Barry Derksen

IT Trends institute, SDU Uitgevers april 2006

ISBN 90-440-1021-2

Open Source Jaarboek 2006-2007

Freek Blankena, Syb Groeneveld, Gijs Hillenius,

Frits de Jong, Sanne te Meerman,

Mathieu Paapst, Hans Sleurink, Wouter Tebbens

Media Update Vakpublicaties 2006

ISBN: 90-78730-01-3

Understanding Open Source Software development

Joseph Feller & Brian Fitzgerald

Addison Wesley 2002

ISBN: 0-201-73496-6

## 8.5 Geïnterviewden

Dhr. R. van Wendel de Joode

Assistent Professor TU Delft

<http://www.tbm.tudelft.nl/live/pagina.jsp?id=2a8b20dc-ce5f-492f-886b-c73837b76afc&lang=nl>

Dhr. M. Methorst

Senior ontwikkelaar

Infa B.V.

[www.infa.nl](http://www.infa.nl)

Dhr. J. Verwaard

Beleidsadviseur Informatievoorziening gemeente Almere

[www.almere.nl](http://www.almere.nl)

Dhr. M. Dashorst

Senior ontwikkelaar

Topicus B.V.

[www.topicus.nl](http://www.topicus.nl)

Dhr. M. Bommelje & M. Wobben

Consultants van BCP software

[www.bcp-software.nl](http://www.bcp-software.nl)

Dhr. M. Krans

(voorheen) Projectleider Emaxx

[www.emaxx.nl](http://www.emaxx.nl)

Dhr. J. Willems

Senior ontwikkelaar

Topicus B.V.

[www.topicus.nl](http://www.topicus.nl)

## 8.6 Bronvermelding van de afbeeldingen

Cathedral:

[http://www.lets-go-europe.com/Germany/Ulm/cathedral\\_pan\\_web.jpg](http://www.lets-go-europe.com/Germany/Ulm/cathedral_pan_web.jpg)

Bazaar:

[http://www.affordablehousinginstitute.org/blogs/us/grand\\_bazaar\\_istanbul.jpg](http://www.affordablehousinginstitute.org/blogs/us/grand_bazaar_istanbul.jpg)

## 8.7 Reviewers

Deze scriptie is gereviewd door:

Dhr. M. Kok (Unit manager Topicus B.V.)

Dhr. J. Verwaard (Beleidsadviseur Gemeente Almere)

Dhr. S. Tempelman (Docent informatica ROC)

Dhr. M. Krans (Projectleider Topicus B.V.)

Dhr. M. Dashorst (Senior ontwikkelaar Topicus B.V.)

Dhr. J. Willems (Senior ontwikkelaar Topicus B.V.)

Marteniek Bierman

4<sup>e</sup> jaar Bedrijfskundige Informatica

Informatie & Communicatie Academie ICA

Hogeschool van Arnhem en Nijmegen HAN