

UNIVERSITEIT GENT

FACULTEIT ECONOMIE EN BEDRIJFSKUNDE

ACADEMIEJAAR 2011 – 2012

Configuration and Transition of Business Models in the ICT Industry: a Study of Content Management Systems

Masterproef voorgedragen tot het bekomen van de graad van

Master of Science in de

Toegepaste Economische Wetenschappen: Handelsingenieur

Reinout Denys

Gert Vanhaverbeke

onder leiding van

Prof. dr. Bart Clarysse



UNIVERSITEIT GENT

FACULTEIT ECONOMIE EN BEDRIJFSKUNDE

ACADEMIEJAAR 2011 – 2012

Configuration and Transition of Business Models in the ICT Industry: a Study of Content Management Systems

Masterproef voorgedragen tot het bekomen van de graad van

Master of Science in de

Toegepaste Economische Wetenschappen: Handelsingenieur

Reinout Denys

Gert Vanhaverbeke

onder leiding van

Prof. dr. Bart Clarysse

Permission

The authors hereby grant Ghent University the right to make this master thesis available for consultation. The reader is granted the right to copy parts of this master thesis for personal use only, explicitly mentioning the source when referring to the contents of this master thesis. Redistribution or publication of this document, partly or entirely, in any form, with or without modification, without prior written approval of both authors, is forbidden.

Reinout Denys

May 2012

Gert Vanhaverbeke May 2012

Acknowledgements

This master thesis is the final step to obtain the certificate of Master of Science in Business Engineering. Without help we would not have been able to finish this master thesis. Therefore we would like to say special thanks to several people.

First and foremost, we would like to thank our promoter Prof. dr. Bart Clarysse for giving us the opportunity to write this master thesis and gain an in-depth knowledge in the interesting world of business models and ICT firms.

Furthermore, we are very appreciative of the support from research assistant Aarti Mahajan from the department of management, innovation and entrepreneurship. As she was always prepared to meet with us, her support kept us on the right track during the whole execution of our master thesis.

We are also very appreciative of the input of Prof. dr. ir. Frank Gielen. Both his classes and the extra information he provided us with, were very useful.

This work was only possible thanks to the input of multiple business professionals: Bart De Waele, Ben Hodous, Brian Brown, Dean Smith, Eddy Lalou, Eric Kuhnen, Franz Maruna, Rob Gietema, Sam Keller, Sjoerd van Elferen and Tim Geyssens. All of them provided us with insights on the structure and functioning of their companies, which constitute the fundamentals of our work.

We also want to thank Annelies Vandenberghe for proofreading this master thesis, and Piet Vanhaverbeke, Sarah Praet and Gertjan Verdickt for reviewing parts of this master thesis.

To conclude we also want to express our appreciation to our parents and friends for supporting us during the stressful and intense period for completing this master thesis.

Reinout Denys & Gert Vanhaverbeke, May 2012 In deze thesis gaan we na hoe verschillende bedrijven zich organiseren om hun product aan de klant te brengen. We focussen ons hiervoor op ICT bedrijven en meer in het bijzonder op aanbieders van Content Management Systemen. Een belangrijk onderscheid dat moet gemaakt worden zijn de twee grote strekkingen binnen de softwarewereld. Aan de ene kant betreft het de 'closed source'-bedrijven, waarbinnen de broncode van het product geheim blijft en aan de andere kant vindt men de 'open source'-bedrijven die hun broncode vrijgeven en met behulp van een 'community' deze verder ontwikkelen.

In een uitgebreid literatuuronderzoek bespreken we de belangrijkste thema's die noodzakelijk zijn voor het verdere verloop van dit onderzoek.

Eerst en vooral wordt het concept omtrent business modellen behandeld. Dit concept is nog vrij nieuw aangezien het pas midden de jaren negentig in literatuur opgedoken is. Sinds de internetzeepbel is het echter een zeer populair begrip geworden. Ondanks de populariteit van dit begrip is in de literatuur geen algemeen aanvaarde conclusie over business modellen te vinden.

In de eerste plaats werden vooral toepassingen van business modellen van internetbedrijven besproken, maar al snel werd duidelijk dat het concept ook voor andere bedrijven interessant is. Een business model speelt een grote rol in de prestaties en levensvatbaarheid van een bedrijf aangezien het de grote lijnen van het waardecreatieproces en het leveren van die waarde aan de eindklant beschrijft.

Door de grote verscheidenheid aan visies is het moeilijk een eenduidige betekenis aan het begrip te binden. Toch komen verschillende aspecten meermaals terug. Op basis van die aspecten kan de volgende definitie opgesteld worden: *"Een business model is een raamwerk dat de basislogica van een bedrijf omtrent het creëren en leveren van waarde aan zijn klanten bespreekt. Rekening houdend met de bijdrage van noodzakelijke middelen, activiteiten en partners voor het bedrijf en de financiële consequenties van deze acties."*

In de literatuur wordt dit raamwerk dikwijls verder opgesplitst in specifieke componenten met elk hun belangrijke bouwstenen. Deze onderverdelingen maken het mogelijk om een business model te begrijpen en er over te kunnen communiceren. Volgens het werk van Osterwalder (2010) kunnen er over het algemeen 4 grote componenten geïsoleerd worden: product, klanten, infrastructuur en financiën. De bouwstenen bespreken binnen elke component hoe ze bijdragen tot de algemene werking van een onderneming.

- product:
 - ✓ waardepropositie: welke waarde in de vorm van een product of service het bedrijf de klanten wil bieden op basis van de klantbehoeftes
- klanten:
 - ✓ klantensegmenten: welke doelgroep(en) het bedrijf tracht te bereiken
 - ✓ distributiekanalen: op welke manier het bedrijf zijn klant zal bereiken
 - ✓ klantenrelaties: op welke manier het bedrijf met zijn klanten communiceert
- infrastructuur:
 - ✓ middelen: welke middelen en vaardigheden het bedrijf moet bezitten om te kunnen voldoen aan hun waardepropositie
 - ✓ activiteiten: de activiteiten die waarde creëren en leveren aan de klant
 - ✓ partnernetwerk: betreft het netwerk van leveranciers en partners en welke activiteiten door het bedrijf zelf en welke extern worden gedaan
- financiën:
 - ✓ kosten: welke kosten gemaakt moeten worden voor het creëren en het leveren van de waarde aan de klanten
 - ✓ inkomsten: de inkomsten die het bedrijf genereert door het leveren van de waardepropositie aan de klant

Verder moet nog opgemerkt worden dat een business model niet gelijk is aan een strategie. Magretta (2002) merkt hierbij op dat elk bedrijf vroeg of laat geconfronteerd zal worden met concurrentie. Hoe een bedrijf hier mee omgaat, wordt bepaald door de strategie. Op die manier zullen strategie en een business model elkaar dus aanvullen.

Het tweede deel van de literatuurstudie bespreekt 'open source'. Dit principe gaat in tegen de standaarden van 'closed source'-softwarebedrijven. Dergelijke bedrijven bieden hun software aan in de vorm van licenties voor het gebruik ervan, bijvoorbeeld Microsoft. 'Open source' houdt over het algemeen in dat de broncode vrij beschikbaar gemaakt wordt.

Het beschikbaar stellen van de broncode in 'open source' wordt ondersteund door het bestaan van verschillende licenties waaronder de code wordt uitgebracht. Deze licenties kunnen van verschillende aard zijn. Algemeen worden de licenties onderverdeeld onder de categoriën 'permissive' en 'restrictive'. Dit komt neer op het al dan niet beschikbaar maken van de code om gebruikt te worden in 'closed source'-software. Deze licenties zijn nodig om twee redenen. Enerzijds, wordt op die manier het initiële eigendom van het auteursrecht gevrijwaard. Anderzijds zorgen deze licenties dat verdere ontwikkelingen van de software ook gedeeld worden met de hele 'community'.

Bedrijven kunnen op verschillende manieren 'open source'-software gebruiken om inkomsten en zelfs winst te generen. Ze kunnen de code gebruiken bij het ontwikkelen van een nieuw product dat ze dan aanbieden aan hun klanten of ze kunnen het volledige 'open source'-product bundelen met bestaande producten en/of services en op die manier aanbieden aan hun klanten.

Het derde deel binnen de literatuurstudie behandelt 'content management'. We spitsen ons hierop toe omdat binnen dit segment een breed aanbod van zowel 'closed source'- als 'open source'-bedrijven beschikbaar is.

'Content management'-systemen hebben over het algemeen drie basis functionaliteiten, nl. verzamelen, managen en publiceren van 'content' (informatie). Specifiek wordt het gebruikt voor websites waarbij de controle in handen is van de eigenaar van de informatie in plaats van in handen van de informaticus. Met een 'content management'-systeem kunnen personen zonder tussenkomst van een informaticus de inhoud van hun website aanpassen. Dit principe blijkt niet enkel interessant te zijn voor websites, maar ook bedrijven zien het potentieel van het principe. Zij gebruiken dergelijke systemen voor het managen van informatie binnen alle lagen van het bedrijf.

Een laatste thema binnen de literatuurstudie omvat een beperkte uiteenzetting omtrent 'Software as a Service (SaaS)'. Dit principe wordt binnen de ICT wereld gezien als belangrijke trend binnen de industrie en is dus zeker ook van toepassing op 'content management'-systemen.

SaaS kan kort samengevat worden als software die beschikbaar gesteld wordt en toegankelijk is via het internet. Het principe is interessant voor bedrijven omdat op die manier een nieuwe groep klanten interessant wordt vanuit economisch standpunt. 'Selling to the long tail' bestaat erin dat bedrijven door kostenefficiënter hun product aan te bieden, een groter aanbod kunnen leveren aan een groter doelpubliek en op die manier meer inkomsten kunnen genereren. In het tweede deel van deze thesis gaan we na hoe ICT-bedrijven hun business model definiëren en leggen we de nadruk op het verschil tussen 'open source'- en 'closed source'bedrijven. Verder proberen we evoluties binnen de 'content management'-sector vast te stellen en welke invloed ze hebben op de 'open source'- en 'closed source'-bedrijven. We doen dit aan de hand van interviews met experten uit de sector. Mensen van verschillende 'closed source' en van verschillende 'open source'-bedrijven werden geïnterviewd om op deze manier tot een algemeen business model van hun strekking binnen de software wereld te komen. Uit de analyse van deze interviews konden we gelijkenissen en verschillen tussen de twee soorten bedrijven bekomen. Verder werden ook enkele opmerkelijke uitspraken uit de interviews besproken en toegelicht.

De gelijkenissen die konden gevonden worden betreffen voornamelijk de distributiekanalen die de softwarebedrijven gebruiken. Zowel 'open' en 'closed source'bedrijven maken hoofdzakelijk gebruik van een partnernetwerk om hun product aan de eindgebruiker aan te bieden. Verder zijn ook op vlak van waardepropositie, kostenstructuur en inkomstenstromen verschillende overeenkomstigheden vast te stellen.

De grootste verschillen zijn voornamelijk te vinden tussen 'closed source' en 'community open source'-bedrijven. Deze bedrijven hebben initieel geen winstoogmerk. De verschillen die merkbaar zijn situeren zich voornamelijk op vlak van waardepropositie. 'Closed source'-bedrijven bieden namelijk professionele diensten aan bij hun, meestal op maat gemaakte, softwarepakketen, terwijl dit niet het geval is bij 'community open source'bedrijven. Echter, 'corporate open source bedrijven', 'open source'-bedrijven met een winstoogmerk, komen in dat opzicht dichter aanleunen bij 'closed source'-bedrijven. Deze twee soorten bedrijven komen ook op financieel vlak meer overeen, aangezien kosten en inkomsten bij deze bedrijven van een andere orde zijn dan bij 'community open source'bedrijven.

Verschillende opmerkelijke uitspraken uit de interviews leiden tot verschillende vaststellingen omtrent conclusies, transities en evoluties in de content managementsystemen-industrie. Op vlak van nieuwe functionaliteiten, bv. sociale media, 'web analytics', SaaS... komen de meningen tussen de verschillende strekkingen grotendeels overeen. Echter, de visies verschillen onder andere op vlak van veiligheid, professionele diensten en prijskaartje tussen 'open' en 'closed source'-bedrijven. Een duidelijke evolutie in de markt is dan ook niet waar te nemen en hoe de verhouding tussen 'open' en 'closed source'-bedrijven in de toekomst zal zijn kan bijgevolg ook niet voorspeld worden.

Table of Contents

Acknowle	dgements	I
Samenvat	ting (Dutch Summary)	II
Table of C	ontents	VI
Utilized A	bbreviations	VIII
List of Fig	ures	IX
Introducti	ion	1
PART 1: L	ITERATURE	2
1. Bu	siness models	2
1.1.	Introduction	2
1.2.	Definition	4
1.3.	Components of a Business Model	7
1.4.	Business models and strategy	
1.5.	Conclusions	
2. Op	en source	
2.1.	History of open source	
2.2.	Definition	
2.3.	The need for licenses	21
2.4.	Open source licenses	21
2.5.	The open source business model	25
2.6.	Conclusions	
3. Co	ntent Management Systems	
3.1.	What is content?	
3.2.	What is content management?	
3.3.	How does it work?	
3.4.	The Market	
4. Sof	ftware as a Service	

Z	ł.1.	Definition	
PART	' 2: RE	ESEARCH	
1.	Res	earch method	
2.	Ana	lysis	
2	2.1.	Closed source companies	
2	2.2.	Open source companies	63
3.	Res	earch conclusion	
	3.1.	Closed source business model	
	3.2.	Open source business model	
	3.3.	Comparison	90
PART	' 3: GE	ENERAL CONCLUSIONS	
Refer	ences	5	Х
Appe	ndice	S	XIV
1.	Inte	erview	XIV
2.	Tra	nscript interview – Ben Hodous, Accrisoft	XVI
3.	Tra	nscript interview – Eric Kuhnen, Anonymous X	XXII
4.	Tra	nscript interview – Brian W. Brown, AppState/phpWebsite	XXVII
5.	Tra	nscript interview – Franz Maruna, Concrete5	XXXI
6.	Tra	nscript interview – Bart De Waele, Fork CMS	XLV
7.	Tra	nscript interview – Dean Smith, Hannon Hill	L
8.	Tra	nscript interview – Sam Keller, Oxcyon (Centralpoint CMS) (part 1)	LXI
9.	Tra	nscript interview – Sam Keller, Oxcyon (Centralpoint CMS) (part 2)	LXVII
10.	Tra	nscript interview – Sjoerd van Elferen & Rob Gietema, Plone	LXXVI
11.	Tra	nscript interview – Eddy Lalou, Sitecore	XCII
12.	Tra	nscript interview – Tim Geyssens, Umbraco	CVII

Utilized Abbreviations

AGPL	Affero General Public License
AppState	Appalachian State University
ASP	Active Server Pages
BSD	Berkely Software Distribution
CEO	Chief Executive Officer
CMS	Content Management System
CRM	Customer Relationship Management
CS	Closed Source
CSS	Cascading Style Sheets
СТО	Chief Technology Officer
EU	European Union
FBI	Federal Bureau of Investigation
FTP	File Transfer Protocol
GPL	General Public License
GUI	Graphical User Interface
HTML	HyperText Markup Language
IaaS	Infrastructure as a Service
IBBT	Interdisciplinair instituut voor BreedBand Technologie
ІСТ	Information and Communication Technology
IRC	Internet Relay Chat
IT	Information Technology
LGPL	Lesser General Public License
MIT	Massachusetts Institute of Technology
MPL	Mozilla Public License
NASA	National Aeronautics and Space Administration
000	Online Channel Optimization
OS	Open Source
OSL	Open Source License
OSS	Open Source Software
PaaS	Platform as a Service
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
POS	Professional Open Source
R&D	Research and Development
SaaS	Software as a Service
SCSL	Sun Community Source License
SEO	Search Engine Optimization
SQL	Structured Query Language
USA	United States of America
XML	Extensible Markup Language

List of Figures

Figure a: Business Model Components Framework (Pateli A. et al., 2003)	8
Figure b: Components of business model affinity diagram (Shafer S.M. et al., 2005)	9
Figure c: The nine business model building blocks (Osterwalder A., 2004)	10
Figure d: Business Model Ontology (Osterwalder A., 2004)	11
Figure e: Components of business models (Kroon M., 2009)	15
Figure f: The 9 building blocks (Osterwalder A. & Pigneur Y., 2010)	15
Figure g: Different OSS 2.0 licenses (Fitzgerald B., 2006)	22
Figure h: Non-GPL model (Krishnamurthy S., 2005)	25
Figure i: GPL-model (Krishnamurthy S., 2005)	26
Figure j: Dual licensing model (Välimäki M., 2003)	29
Figure k: How does a CMS work? (Boiko B., 2005)	34
Figure I: Magic Quadrant for Web Content Management (MacComascaigh M. et al., 2011)	35
Figure m: The "long tail" (Chong F. & Carraro G., 2006)	38
Figure n: Business Model Accrisoft	46
Figure o: Business model Anonymous X	49
Figure p: Business model Hannon Hill	53
Figure q: Business model Oxcyon	59
Figure r: Business model Sitecore	63
Figure s: Business model Appstate/phpWebsite	66
Figure t: Business model Concrete5	71
Figure u: Business model Fork CMS	74
Figure v: Business model Plone	79
Figure w: Business model Umbraco	83
Figure x: Business model Closed Source	86
Figure y: Business model open source	89

Introduction

The IT industry is characterized by rapid evolutions and innovativeness, so keeping up with the number of changes is almost impossible. For software companies it is important to be well-organized so they can adjust to changing environments quickly. Often the vision on how to be organized, diverges from company to company.

In general, enterprises' visions on the way to organize themselves for doing business can be summarized in their 'business model'. Definitions on business models vary significantly, but the general idea is that it includes all the core logic of a profit-making organization.

In the software industry a distinction can be made between open source and closed source companies. Both types of companies have their own believes in the way they should be offering their product, which is of course translated in the business models of this firms.

Active parts of the software industry are the companies focusing on content management. These companies are developing processes and system technologies, which support the collection, management and publication of information in any form or medium.

In this master thesis we investigate the business models of these ICT firms based on the distinction between open and closed source companies. Therefore we focus on firms who offer content management systems, as this is a large industry with sufficient number of open and closed source companies.

In the first part we will discuss the topics of business models, open source, content management systems and Software as a Service (SaaS) based on our literature research. In the second part of this thesis we present our findings on the research we have executed. Consequently, we discuss the research method, the expert interviews, the analysis and, at last, the research findings. In the last part of this text, a general conclusion is provided together with the limitations of our research and some suggestions for further research.

PART 1: LITERATURE

1. Business models

In this chapter we will go into further detail about business models. We will discuss what a business model exactly is and elaborate on different theories found in literature. An overall introduction on business models can be found in §1.1. Further, definitions of business models from literature will be compared in §1.2, in which we present our own definition on the subject. In §1.3, the different components of a business model will be discussed and will be used as the ontology for the rest of our research. In §1.4 we will deal with the distinction between business models and strategy. Conclusions on the literature research on business models can be found in §1.5.

1.1. Introduction

A business model is a relatively recent concept. When searching for scientific literature, there is few to be found before the mid-nineties. Since the dot-com era however, there has been a lot of buzz around the concept. Ever since, much research has been done, but no clear definition has come forward.

With the emergence of the internet, people were looking how to make value from it. Initially, business models were used as a managerial tool while setting up e-business companies. (Osterwalder A. & Pigneur Y., 2002; Osterwalder A., Pigneur Y. & Tucci C., 2005) Not only the growth of the Internet and e-commerce, but also the emerging knowledge economy, the outsourcing and offshoring of many business activities and the restructuring of the financial services industry around the world were driving factors behind the appearance of business models in public consciousness. (Teece D. J., 2010)

The concept of business models is not only applicable to the online world, but also relevant to all sorts of firms. It's not surprising that many 'brick and mortar' companies were forced to rethink their strategies, if not their whole business models, because of the Internet. Shorter product life cycles, global competition and the use of ICT forced managers to find new ways of doing business in this complex environment. Exploiting new opportunities offered by the internet was something that had to be integrated in the existing business models. (Shafer S., Smith J. & Linder J., 2005; Teece D. J., 2010, Osterwalder A. & Pigneur Y., 2002)

In modern market economies, there are a lot of factors that have to be taken into account. Factors such as consumers' choice, transaction costs, heterogeneity amongst consumers and producers, but also competition, supplier selection, etc. approve the importance of a firm's business model. Despite this obvious prove, the study of business models is a topic which has been neglected for a long time. (Teece D. J., 2010)

Nowadays, there is still much confusion about business models in general, what they are and how they can be used. Although several authors presented different definitions of 'business models', none has appeared to be accepted in general. (Shafer S. et al., 2005) However, this has not stopped people discussing business models in practice, on the contrary.

Although everyone expresses their own point of view on business models, several similarities can be recognized in the different opinions. Generally, the basis of a business model can be seen as the identification of the core logic of the firm. More specifically, it concerns the elements and relationships which are responsible for creating and capturing value for a firm. (Osterwalder A. et al., 2005; Shafer S. et al., 2005)

Despite this view on business models, some authors believe a business model has to be more than just a good logical way of doing business. They believe a business model has to be a source of competitive advantage for the firm. This way, exploiting the firm's business opportunities contributes to the performance and survival of the company. (Teece D. J., 2010; George G. & Bock A., 2011)

The performance and consequently the survival of companies largely depends on the quality of their business models. Depending on the sector in which the firm is active, several criteria determine whether or not a company has designed a good business model. Teece (2010) describes a good business model as a value proposition compelling to customers, achieving advantageous cost and risk structures which enables the business that generates and delivers the products and/or services to capture significant value.

Of course it has to be said that there is not one single good business model. A good design comes from an iterative process and is highly dependent on the situation the firm is in at the moment. Furthermore, changing technology and enhanced competition also requires the constant evolution of a firm's business model. Sometimes it is even necessary to abandon a business model and replace it by an improved model which incorporates the advantages of organizational and technological innovations. (Teece D. J., 2010)

Overall, the value of business models can be found in different fields. First of all, a business model helps companies implementing their business strategies. Furthermore, a good comprehension of the concept of a business model allows firms to clearly understand, communicate and learn about the different aspects of their business. (Osterwalder A. & Pigneur Y., 2002)

In the next section, we will discuss different definitions of a business model found in literature and we present our own definition on the subject.

1.2. <u>Definition</u>

"A business model describes the design or architecture of the value creation, delivery and capture mechanisms employed." (Teece D. J., 2010, p. 191)

"A description of the roles and relationships among a firm's consumers, customers, allies, and suppliers that identifies the major flows of product, information, and money, and the major benefits to participants." (Weill P. & Vitale M.R., 2001, p. 34)

"A business model is a representation of a firm's underlying core logic and strategic choices for creating and capturing value within a value network." (Shafer S. et al., 2005, p. 202)

"A description of a complex business that enables study of its structure, the relationships among structural elements, and how it will respond in the real world." (Applegate L.M., 2000, p. 53)

"A business model is a conceptual tool that contains a set of elements and their relationships and allows expressing the business logic of a specific firm. It is a description of the value a company offers to one or several segments of customers and of the architecture of the firm and its network of partners for creating, marketing, and delivering this value and relationship capital, to generate profitable and sustainable revenue streams." (Osterwalder A. et al., 2005, p. 17-18)

The above cited definitions are just a few of the ones that can be found in literature. This also illustrates that there is not a generally accepted definition of the business model

concept. The consequence of this variation in literature is the misuse of the term in different contexts.

From their research on literature on business models, Osterwalder et al. (2005) show that there is a continuum between the uses of business models in literature. This continuum goes from authors who use the term to simply refer to 'the way companies organize themselves in a specific business' (Galper J., 2001; Gebauer J. & Ginsburg M., 2003) to authors who emphasize the modeling aspect of business models (Gordijn J. & Akkermans H., 2001; Chesbrough H.W., 2003; Osterwalder A. & Pigneur Y., 2002). This second group of authors tries to conceptualize rather than describe the way an organization does its business. This conceptualization contains the picture of how the elements and the interconnections of a company function. That way a business model can be seen as a tool to transcribe business complexity to an understandable level.

The different definitions found in literature vary from very generic descriptions to very specific ones. This creates an extra difficulty in understanding the concept of a business model. In order to obtain a better comprehension of a business model it is useful to set up an ontology around the subject. Osterwalder and Pigneur (2002) and Fensel (2001) define an ontology as a rigorously defined framework providing a shared and common understanding of a specific domain by defining its elements and the relationships between these elements. Such an ontology would make it possible for managers to easily understand, share, communicate, design and evaluate business models in general. (Osterwalder et al., 2005)

To build up a definition Osterwalder et al. (2005) had the idea of examining the two words in the term itself. Hereby they came to the following raw definition: "A business model is a conceptual tool containing a set of objects, concepts and their relationships with the objective to express the business logic of a specific firm. Therefore we must consider which concepts and relationships allow a simplified description and representation of what value is provided to customers, how this is done and with which financial consequences." (Osterwalder et al., 2005, p. 5)

Magretta (2002) does a good job on making this definition less abstract. She defines a business model as follows: "They are, at heart, stories – stories that explain how enterprises work. A good business model answers Peter Drucker's age-old questions: Who is the customer? And what does the customer value? It also answers the fundamental questions every manager must ask: How do we make money in this business? What is the underlying

economic logic that explains how we can deliver value to customers at an appropriate cost?" (Magretta J., 2002, p. 87)

By observing the definitions from the beginning of this chapter and the descriptions above it's clear that a business model describes several specific elements and relationships in an organization. Again, in literature, a wide range of possibilities in order to create a correct definition are suggested. Several definitions only cover the financial part and see a business model as a revenue model, focusing on the way of how to generate revenue and profit. On the other hand, as Osterwalder and Pigneur (2002) rightly mention, there are also a large number of authors who only focus on other aspects of a business model without picturing the whole image. In existing business model literature, a lot of authors only discuss the revenue- and product-specific, business actor- and network-specific or marketing-specific aspects of a business model. Rarely, two or all of the different aspects are handled in the same research.

Taking this into account, it is important to see a business model in the broad sense. In the words of Osterwalder et al. (2005): *"For business models, the quest is to identify the elements and relationships that describe the business a company does."* (Osterwalder et al., 2005, p. 5)

When exploring literature on business models, an overall description of the concept can be made. Combining several definitions of business models and generalizing this input identifies recurring aspects such as business logic, value creation and delivery, product, customer, costs and revenues. Research from Shafer et al. (2005) shows that business people see these aspects as important factors for firm performance and survival, organizational structure and opportunity exploitation of a company.

Teece (2010) defines a business model as the logic of how a business can create and deliver value to customer, while taking into account the architecture of revenues, costs, and profits associated with the business enterprise delivering that value. In general this reflects the hypothesis made by management about what customers want, how they want it and how an organization is organized to best meet those needs, gets paid for doing so, and makes a profit.

Combining all of the above, we define a business model as follows. This definition will be used throughout the remainder of this research:

A business model is the framework that contains the company's basic logic on value creation and delivery to its customers and suppliers, obtained by the input of its available resources, activities and partners, while keeping in mind the financial outcomes of these operations.

Finally, as Shafer et al. (2005) and Osterwalder et al. (2005) point out, a business model has to be seen as a structure connecting the firm's core activities to service a specific set of goals. When a company defines a business opportunity, it is a matter of implementing the business model in concrete things containing business configuration (e.g. departments, units, human resources), business processes (e.g. workflows, responsibilities) and infrastructure and systems (e.g. buildings, ICT).

1.3. Components of a Business Model

In literature, business models are regularly described through different basic building blocks. These building blocks are used to show the logic of how a company intends to make money. (Osterwalder A. & Pigneur Y., 2010)

In their book 'Business Model Generation' Osterwalder en Pigneur (2010) describe 9 building blocks which cover the four main areas of a business, namely customer, product, infrastructure and financial viability. This would facilitate the description and discussion on what is understood by the business model concept. Doing this, they tried to keep the concept simple, relevant and intuitively understandable, without oversimplifying all the complexities of how an enterprise functions.

The idea of business models consisting different building blocks is described in several research papers. Although different authors identify different business model components, again, a lot of similarities can be found.

Osterwalder & Pigneur (2002) indicate that most authors writing about business model components only discuss a few of them. They put forward that in literature rarely all four of following aspects concerning a business model are handled: financial aspects, product aspects, customer aspects and/or actor & infrastructure aspects. Regularly, authors only deal with financial/product aspects (Tapscott D., Ticoll D. & Lowy A., 2000), actor & infrastructure aspects (Timmers P., 1998; Gordijn J. & Akkermans H., 2001) or customer specific aspects.

George & Bock (2011) frame a business model comprising 3 dimensions. Those are resource structure, transactive structure and value structure. They define resource

structure as the static architecture of the firm's organization, the product technology and core resources, while transactive structure consists of the configuration of key transactions with partners and stakeholders. Thirdly, value structure is the system of rules and mechanisms that determine the value creation and capturing of the firm. Dependent on the relative dominance of one or more dimensions, George & Bock (2011) believe this will have implications on the organizational effectiveness, strategic fit and structuration within the environmental context of the company.

Some authors go a step further and elaborate business model components even more. Pateli & Giaglis (2003) attempted to combine and synthesize a number of standard components identified in other literature in a generic framework (figure a). This framework consists of a revised and extended version of Alt and Zimmermann's (2001) proposed construct for the six elements of a business model. The horizontal frame includes the primary components of a business model: Mission (Strategic Objectives), Target Market (scope and market segment), Value Proposition (product/ service offering), Resources (capabilities, assets), Key Activities (intra- and inter-organizational processes), Cost and Revenue Model (cost and revenue streams, pricing policy), Value Chain/Net (alliances and partnerships). The vertical frame supports the underlying components of a business model; business and social environments such as market trends, regulation and technology. As this structure increases the complexity of the framework significantly, we believe using this model would not contribute to the general understanding of a business model.



Figure a: Business Model Components Framework (Pateli A. et al., 2003)

Somewhere in between these two descriptions of business model components, we can situate the framework of Shafer et al. (2005) As they define a business model as "a representation of a firm's underlying core logic and strategic choices for creating and capturing value within a value network", they identify 4 key components for their framework (figure b).



Figure b: Components of business model affinity diagram (Shafer S.M. et al., 2005)

The first key term is 'strategic choices' which contains the assumptions concerning the value proposition of the firm. More specifically this can be seen as the bundle of products and services the company wants to offer. 'Create value' and 'capture value' grasp the fundamental functions an organization must perform to remain competitive. 'Create value' displays the way the company wants to operate and use its resources, while 'capture value' is the component where costs and profits are situated. To make this value creation and delivery possible, interactions with suppliers and partners are necessary within the 'value network'.

Nonetheless, the most cited author on the work of business model components is Osterwalder. As stated before, the goal of his research was to construct a framework which contributed to the general understanding of the business model concept. As he defines a business model as *"the rationale of how an organization creates, delivers and captures value"* (Osterwalder A. & Pigneur Y., 2010, p. 14), several similarities with the work of Shafer et al. (2005) can be recognized.

In the work of Osterwalder & Pigneur (2002) and Osterwalder (2004), the business model ontology is founded on four main pillars: products and services, infrastructure and network of partners, relationship capital and the financial aspects (figure c).

Pillar	Building Block of Business Model	Description
Product	Value Proposition	A Value Proposition is an overall view of a company's bundle of products and services that are of value to the customer.
	Target Customer	The Target Customer is a segment of customers a company wants to offer value to.
Customer Interface	Distribution Channel	A Distribution Channel is a means of getting in touch with the customer.
	Relationship	The Relationship describes the kind of link a company establishes between itself and the customer.
	Value Configuration	The Value Configuration describes the arrangement of activities and resources that are necessary to create value for the customer.
Infrastructure Management	Capability	A Capability is the ability to execute a repeatable pattern of actions that is necessary in order to create value for the customer.
	Partnership	A Partnership is a voluntarily initiated cooperative agreement between two or more companies in order to create value for the customer.
Financial	Cost Structure	The Cost Structure is the representation in money of all the means employed in the business model.
Aspects	Revenue Model	The Revenue Model describes the way a company makes money through a variety of revenue flows.

Figure c: The nine business model building blocks (Osterwalder A., 2004)

The first pillar covers all product-related aspects. The value proposition of what a firm wants to offer to specific target customer segments and the capabilities the firm should possess to assure the delivery of this value are the main elements of this pillar.

Secondly, infrastructure management comprises the configuration of the activities within a firm and the relationships with the firm's partner network to assure the correct resources and assets are possessed to create and deliver value.

Further, customer relationship handles the information strategy towards the customer, the channels used to deliver the value and the creation of trust and loyalty through the firm's relationship with the customer.

The last pillar consists of the revenue model and cost structure of the firm. As these determine the profit model of the company, this will determine whether or not the company has the ability to survive.

To get a more detailed image of this business model ontology, the pillars of this figure are subdivided into nine components. These components can be retrieved from the following figure (figure d).



Figure d: Business Model Ontology (Osterwalder A., 2004)

The relationships between all the building blocks can easily be described. An organization realizes a value proposition via their core capabilities and partner network, which both entail certain costs. This value proposition is offered to the targeted customer segments by means of particular distribution channels and is based on the relationship with the client. This way the organization generates revenues. The difference between these revenues and the cost structure of the company will determine the company's profit.

Below, the different building blocks will be discussed in more detail.

Value proposition

The first building block, value proposition, is found in the product pillar of the business model framework of Osterwalder and Pigneur (2010). In general, this is the overall description of the value the firm wants to offer to a specific customer segment. The company offers a bundle of products and/or services to satisfy a customer's need. Eventually, this value proposition will be the way of how the company creates a competitive advantage and is the reason why people become a company's customer.

Value can be offered in several ways as it can fulfill a distinct set of customers' needs. Value can be created in a quantitative (e.g. price, speed) or in a qualitative (e.g. design) way. Possible factors customers take into account concerning a product or service are: newness of the offer, performance, customization, accessibility, design, price et cetera. (Osterwalder A. & Pigneur Y., 2010)

Customer segments

In the customer pillar, the customer segments concern the different groups of people a company wants to reach and serve. Good customer segmentations offer the company the possibility to assign resources and investments to the groups of clients which are most attracted to the value proposition of the company. The separate segments can be distinguished by different needs, different distribution channels, different types of customer relationships or different profitabilities. A business model can mark one or more small or large customer segments. Some examples are: mass markets, niche markets, segmented markets, diversified markets and multi-sided markets. (Osterwalder A. & Pigneur Y., 2010)

Distribution channels

The second building block within the customer pillar affects the distribution channels of a company. This describes how a company communicates with and reaches their customer segments. Communication, distribution and sales channels are the ways in which a company links the value proposition to the customer segment. The goal is to deliver products and services in the right amount, on time, and to the right customer segment. Several methods can be distinguished: own direct channels such as sales forces and/or web sales or indirect channels such as own stores. Another way is through indirect partner channels such as partner stores or wholesalers. (Osterwalder A. & Pigneur Y., 2010)

Customer relationships

Customer relationships is the last building block in the customer pillar and details the types of relationships a company establishes with the specific customer segments. Interaction and communication with the clients can result in certain customer relationships. These relationships can have the goal of customer acquisition, customer retention or upselling. This can be done in several ways: personal assistance, self-service, automated services, communities, co-creation and so on. (Osterwalder A. & Pigneur Y., 2010)

Key resources

The first building block in the infrastructure pillar is capabilities or resources. This block handles the necessary assets important for the company to make the business work. Dependent on the business model different, key resources are needful. Key resources are not necessarily owned by the company, but can also be obtained from partners. Resources can be of physical, financial, intellectual or human nature. Physical resources contain assets such as manufacturing facilities, buildings, machines, systems, distribution networks... Intellectual property is often difficult to develop, but can offer considerable value to the company. Examples are brands, patents and copyrights, customer databases, partnerships and proprietary knowledge. Human resources are the people needed to create value with tangible and intangible resources, these are especially important in knowledge-intensive industries. Financial resources can be seen as financial guarantees such as cash, credits and/or stock option pools. (Osterwalder A. & Pigneur Y., 2010)

Key activities

The second infrastructure block is value configuration, which depends on the key activities of the company. To ensure the business model works, these key activities are the most important things a company must do. In order to realize the value proposition, it is fundamental to have an elaborate configuration of internal and external activities and processes. This configuration will lead to value creation and delivery. As with key resources, key activities also hinge on the type of business model. In manufacturing companies, production activities will dominate, while in service organizations the key activities relate to finding a solution to the problem of each individual customer. (Osterwalder A. & Pigneur Y., 2010)

Key partnerships

The last part of the infrastructure pillar is the key partnerships. These concern the network of suppliers and partners of a company and describe which key activities of the firm are done by external parties. Four different types of partnerships can be distinguished: strategic alliances, coopetition, joint ventures and buyer-supplier relationships. The motivations for these partnerships can vary. A first one is optimization and economy of scale, which optimize the allocation of resources and activities in order to reduce cost (e.g. outsourcing). A second motivation for a partnership can be the reduction of risk and uncertainty, as sharing knowledge and resources minimizes the possibility to fail. Because not many companies possess all the resources and activities in their business model, a last motivation is the acquisition of particular resources and activities. Companies enlarge their own capabilities by relying on partners for delivering specific resources or execute certain activities. (Osterwalder A. & Pigneur Y., 2010)

Cost structure

In the financial pillar, we find the cost structure building block. This contains all the costs the company makes to create and deliver value to their customers. Companies not always tend to minimize costs. Value-driven firms primarily focus on the value creation and are less concerned about the cost implications (e.g. luxury hotels). On the other hand, cost-driven companies try to minimize costs wherever possible (e.g. low-cost airlines). (Osterwalder A. & Pigneur Y., 2010)

Revenue streams

In the last financial building block we find the revenue streams of a company. They represent the way a company generates revenues through the delivery of the value proposition to each customer segment. It is possible to have different revenue streams with different pricing mechanisms within a business model. In general, two different types of revenue streams can be distinguished: transaction revenues from one-time customer payments and recurring revenues from ongoing payments for after-sales service. The ways a company generates revenue streams also diverge: asset sales (selling ownership to physical good), usage fees (use of specific service), subscription fees (selling access to service), lending/renting/leasing (temporarily granting right of use for a particular asset for a fixed period), licensing (permission to use protected intellectual property), brokerage fees (intermediation services) and advertising.

Pricing mechanisms can make a huge difference in terms of generating revenues. The two main pricing mechanisms are fixed and dynamic pricing. Fixed pricing concerns predefined prices based on list price, product features, customer segments or volume. With dynamic pricing prices change dependent on negotiation, yield management, realtime-market or auctions. (Osterwalder A. & Pigneur Y., 2010)

As Kroon (2009) rightly mentions, there are several authors who name the different building blocks of a business model. Osterwalder (2004) and Osterwalder et al. (2005) are the most extensive works but also Jansen, Jägers, Steenbakkers & Melger (2003) and Chesbrough & Rosenbloom (2002) are authors who cover different components of a business model. The results of their research can be compared in the following figure (figure e).

Osterwalder et al. (2005)	Chesbrough &	Afuah & Tucci (2001)	Markides (2000)
	Roosenbloom (2002)		
Value Proposition	Value Proposition	Customer Value	What
Target Customer	Market Segment	Scope	Who
Revenue Model	Revenue Mechanism	Revenue Source	How
Value Configuration	Value Chain	Implementation	
Cost Structure	Cost Structure	Pricing	
Partner Network	Value Network	Connected activities	
Core Capabilities	Competitive Strategy	Sustainability	
Customer Relationship		Capabilities	
Distribution Channel			
Jansen et al. (2004)	Mitchell & Coles (2003)	Bouwman & MacInnes	Linder & Cantrell (2000)
		(2006)	
Strategie	Who	Service Offering	Value Proposition
ICT	What	Technology	Channel Model
Processen	When	Financial Arrangements	Commerce Relationship
Besturing	Why	Organization	Commerce Process Model
	Where	Arrangements	Pricing & Revenue Model
	How		
	How Much		

Figure e: Components of business models (Kroon M., 2009)

It is clear that a lot of similarities can be found in the choice of the building blocks between the different researchers. Often, dependent on the level of abstraction, the number of building blocks diverge. This can lead to differences in naming, but in essence, these components describe the same things.

During the remainder of our research we will apply the business model of Osterwalder (2004) comprising the 9 building blocks discussed earlier (figure f).



Figure f: The 9 building blocks (Osterwalder A. & Pigneur Y., 2010)

1.4. Business models and strategy

An important distinction that has to be made, is the distinction between business models and strategy. As with literature on business models, also literature on this topic is diverse. Many authors have a different vision on the distinction between both concepts. For many authors the difference between strategy and business models is not really clear and the terms are often used interchangeably (Magretta J., 2002). Business models and strategy complement one another, but they are not the same thing.

As described in §1.2 a business model is "a framework that contains the company's basic logic on value creation and delivery to its customers and suppliers, obtained by the input of its available resources, activities and partners, while keeping in mind the financial outcomes of these operations." (Supra, p. 7)

For a company it is not sufficient to have a successful business model as they also have to take into account how they will be better than or different from the rest. A business model on its own is rather easy to copy, so it is just a matter of time before someone imitates it if the model turns out to be successful. (Teece D. J., 2010) As Magretta (2002) rightly mentions every company will have to deal with competitors sooner or later and dealing with this competition will be crucial for the performance of a company. This is where strategy comes in.

Porter defines strategy as follows: "Strategy is the creation of a unique and valuable position, involving a different set of activities. [...] The essence of strategic position is to choose activities that are different from rivals." (Porter M.E., 1996, p. 68)

Magretta's vision on strategy comes down to the same logic: *"How you are going to be better by being different."* (Magretta J., 2002, p. 91)

How a company will be different from its competitors is determined by the strategic choices the company makes. These choices concern "performing activities differently or performing different activities than rivals." (Porter M.E., 1996, p. 64) More specifically, this relates to determining "which products or services are offered in which markets based on differentiating features." (Schafer M. et al., 2004, p. 203) The (operational) implications of these choices are translated in the business model as this is the "representation of a firm's underlying core logic and strategic choices for creating and capturing value within a value network." (Schafer M. et al., 2004, p. 202)

The implementation of a well-designed (new) business model with the right strategy will result in a competitive advantage for the company. (Teece D. J., 2010) Having this competitive advantage means you do something in a way no other company can duplicate and this unique position will lead to superior performance. (Magretta J., 2002) From the above, it becomes clear that strategy and business models complement each other.

1.5. Conclusions

In this chapter we have introduced the concept of business models. First we have situated the emergence of the concept in a short introduction. Based on definitions from literature, we have generated our own definition on a business model, which can be found in the second paragraph. In the third part of this chapter, we have discussed the different components of a business model and motivated our choice for using the Osterwalder business model for the remainder of this thesis. In the fourth paragraph we have emphasized the distinction between a business model and strategy.

For the remainder of this research we will focus on what are the business models in the ICT industry. More specifically, we will try to find an answer to the following question:

What are the differences and similarities in business models in ICT firms and what are the transitions and evolutions in the sector?

2. Open source

In this chapter we will explain the concept of open source. It is the counterpart of closed source, which actually is the traditional software vendor business model. The best known example of closed source software is Microsoft Office. A fee is paid for a license, which allows the user to install and make use of the software. As closed source is an easy concept to understand, this chapter will only deal with open source. §2.1 defines a short history of open source, while §2.2 presents its official definition. In §2.3, the need for licenses is discussed. §2.4 gives a brief overview of the different licenses available to software developers. We limit ourselves to the most important licenses, as there is a very large number of licenses available. §2.5 then includes the business model concept. We elaborate on the different business models existing in open source. Both original, older business models of open source as well as the most contemporary ones are included. The final paragraph (§2.6) summarizes the conclusion of this chapter.

2.1. <u>History of open source</u>

The origins of open source (OS) lie in the history of UNIX, Internet free software and the hacker culture. The first author writing about open source was Eric Raymond (1999), who published his paper "The Cathedral and the Bazaar" in 1997. The term 'open source' has been in use since February 1998, when it replaced the word 'free software'. The switch became necessary as 'free software' brought along a pile of negative connotations. According to Hecker (1999), "The word 'free' has traditionally led to commercial software vendors to think 'no revenue' and customers of those companies to think 'no support'." (Hecker F., 1999, p. 46)

2.2. <u>Definition</u>

The most basic definition of open source is 'source code made freely available'. The official definition of open source can be found on the site of The Open Source Initiative and entails more than just access to source code. The definition is based on ten criteria:

"Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

Integrity of the Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface." (Open Source Initiative)

Some authors have noticed a shift in open source software (OSS). Fitzgerald notes that "*the* open source phenomenon has undergone a significant transformation from its free software origins to a more mainstream, commercially viable form – OSS 2.0". (Fitzgerald B., 2006, p. 587) OSS 2.0 is a term used by Fitzgerald to identify how open source can be commercially profitable and at the same time keep the community values. Another term used to describe this phenomenon is 'professional open source (POS)'. POS has gone through a few phases already, and will continue to develop as a business model (Watson R., Wynn D. and Boudrea M., 2005). The shift from OSS to OSS 2.0 or POS is in our view undeniable and plays an important role in the change of business models as well, as we will discuss later in this chapter. This shift also sets the basics for the distinction between community open source and corporate or commercial open source. The fundamental difference between these two categories primarily lies in the number of stakeholders. Community OS is controlled by a community of stakeholders, while corporate OS is in hands of one stakeholder. This stakeholder has the full copyright to the code and related intellectual property (e.g. trademarks and patents). (Riehle D., 2009)

Open source software is often confused with public domain software and shareware. Therefore, it is useful to make a careful distinction between them. Public domain software for one does not have an owner and so no one owns the copyright. Thus, it is possible for anyone to change public domain code without any restrictions. The risk attached to releasing software in the public domain is that a third-party can reuse the freely available software to create a proprietary product and make money with it. On the contrary, OSS is always licensed software. This means that there is always someone (or often a group of individuals) who retains the copyright ownership of the source code. The need for licenses and the different licenses themselves will also be discussed later in this chapter. Shareware is in some way a bit similar to open source software, as it is generally licensed software. However, where OSS usually is free, shareware tends to use a 'try before you buy' principle, giving the user a certain trial period. (Driver, M., 2010) It should however be remarked that shareware does not share source code in any way, which is where the fundamental difference with open source software lies.

2.3. <u>The need for licenses</u>

The need for licenses is a two-fold issue. On the one hand, there's the previously discussed problem of releasing software in the public domain. On the other hand, there is the copyright issue. The problem is that original developers are always privileged because they 'own' the original code. As a consequence they are free to set license terms for the use and redistribution. This problem becomes significant when taking into account the point of view of a developer. If other developers are privileged and receive compensation, why would they contribute to the open-source program? This question immediately gives rise to the need for licenses that restrict developers from taking the credit and profit for another's work. This is where the different open source licenses come in, which aim to prevent developers from obtaining a privileged position over another. In the next section we give a brief overview of the most important licenses available.

2.4. Open source licenses

Over the past decade, a lot of open source licenses have been written. These licenses are available to use, but one can also modify an existing or even write a completely new license. Licenses can be divided into several categories. Fitzgerald (2006) defines 4 categories of licenses in his OSS 2.0 concept: Reciprocal licenses, Academic Style licenses, Corporate Type licenses and the Non-approved licenses (figure g).

Table 2. A Typology of OSS 2.0 licenses		
Reciprocal	GPL, LGPL, Open Source License (OSL)	
Academic Style	Academic Free License, Apache License, BSD, MIT	
Corporate Type	MPL, Qt Public License, Sun Public License, IBM Public License, Apple Public License, Eclipse Public License	
Non-Approved (e.g., Shared Source family)	Microsoft Shared Source Initiative Licenses: (Microsoft Community License, Microsoft Permissive License), Sun Community Source License (SCSL)	

Figure g: Different OSS 2.0 licenses (Fitzgerald B., 2006)

Reciprocal licenses (also called viral licenses) traditionally mean that derivatives of the original software must be licensed under the same license. An important term in this regard is 'copyleft'. The term is actually a parody to the term copyright, which is used with closed source, where the code is being protected as intellectual property. Copyleft, on the contrary, aims to keep intellectual property freely available to the public. When discussing licenses, a distinction can be made between weak and strong copyleft. Reciprocal licenses are of the second category, meaning these are more restrictive.

Academic licenses guarantee the retention and acknowledgment of previous contributors' work, but further impose few restrictions.

Corporate-style licenses aim to benefit corporate interests rather than the open source community.

The non-approved category of licenses pushes the boundaries of proprietary software as the sector wants to accommodate the open source model. (Fitzgerald B., 2006)

For each of these categories, we will now briefly discuss the most influential licenses.

BSD License and BSD-style licenses

Originally released to use for UNIX distributions of Berkeley University, the BSD (Berkeley Software Distribution) license has the following features:

- Explicit grant of the right to unlimited use in source or binary form
- The developers' copyright notice should be retained in modified versions
- The developers should be credited in advertising materials using the software
- Neither the name of the organization, nor the name of its contributors may be used to endorse or promote products.

Modified versions of the original BSD license are called BSD-style licenses, and contain the minimum terms and conditions that a license should have to be effective. BSD-style licenses ensure maximum freedom to developers to use the software under the BSD-license and create a proprietary product, for which the source code is not necessarily openly available. For this reason, one can categorize BSD-style licenses as weak copyleft. Therefore, it is often not recommended using the BSD license, but to use a license, which ensures that derivative software will be open source as well. (Hecker F., 1999)

GNU General Public License

The GNU General Public License (GNU GPL or GPL) was originally written by Richard Stallman for the GNU project and has been the most widely used and documented open source license since. Over 80 percent of the open source software is licensed under GPL. (Bonaccorsi A. & Rossi C., 2003) Since its initial release, two follow-ups have been launched already. GPL2 & GPL3 are the successors for the original GPL. The GPL is designed to prevent open source projects being used to create proprietary software, which was possible with the BSD license. An example of this is Microsoft. They have acknowledged that they incorporated BSD-code in its operating systems windows 2000 and XP. The GPL avoids this through so-called copyleft provisions in the GPL, which means that software licensed under GPL must be distributed without a license fee and with the source code made available. Furthermore, derivative works of a program licensed under GPL must also be licensed under GPL. The term derivative works is specified in GPL as 'any work that you distribute or publish that in whole or in part contains or is derived from the program or any part thereof". So not only modified versions of the software are categorized under these derivative works, but also software which uses simple fragments of the source code.

GPL can give problems for developers who wish to convert their proprietary software into open source. If the proprietary software contains third-party technology or shares source code with other products, then the GPL would be a difficult license to implement, as it is impossible to put only a part of the code under GPL and still use that code in other products. (Hecker F., 1999)

GPL has a number of variants. One of the variants is the 'Lesser GPL' or in short LGPL, which allows a greater flexibility concerning the incorporation of commercially licensed pieces of source code. (Lerner, J. & Tirole, J., 2002) Another well-known variant is the Affero GPL or AGPL. This license was originally written to cover a loophole in the original GPL license. As technology and software evolved since the GPL was written, a very specific
problem was not covered. The original GPL describes only ways of physically distribution of software, such as software burned on a CD. However, new ways of delivering software have been developed since a number of years. Software as a Service (SaaS) for example makes it possible to deliver software over the internet. As distribution over the network wasn't covered within GPL, third-parties could build proprietary software from OSS without infringing the GPL license. AGPL solves this issue, and is widely used today by (SaaS) applications.

Mozilla Public License

The Mozilla Public License (MPL) was created by Netscape together with the related Netscape Public License as part of the project to release the Netscape Communicator source code. The MPL was thus created by a commercial company, which was not the case for GPL or BSD.

MPL is similar to GPL regarding source code modifications. When a piece of code falls under MPL, the modifications and use of this code also fall under the MPL. This way they prevent open source code being used for proprietary software. The MPL does differ from the GPL on another level. MPL explicitly permits the use of MPL-ed code to be combined with proprietary code to create a proprietary program that doesn't fall under the MPL. It may be licensed for a fee and the proprietary source code doesn't have to be publicly available. (Hecker F.,1999)

The Microsoft Shared Source Initiative

Being rather a framework than a license, the Microsoft Shared Source Initiative is gaining importance in OSS 2.0. Microsoft originally designed this framework to comply with the growing demand of increasing transparency. Therefore, they committed to three core shared source licenses: (Fitzgerald B.,2006)

- Microsoft Reference License: permits licensees to only view the source code.
- Microsoft Community License: based on the MPL and intended or collaborative projects
- Microsoft Permissive License: similar to BSD and allows licensees to review, modify, redistribute and sell works without having to pay royalties to Microsoft.

2.5. The open source business model

After discussing the need for the licenses, and the various licenses themselves, in this section we try to address the question 'How do we make a business in open source environments?' Many different open source business models are documented in literature. Nevertheless, there are many commonalities to be found. Generally we can divide the OS business models between the early classical business models and the more recent business models existing in OSS 2.0. Originally, business models have often been split up between GPL and BSD models. (Onetti A. & Capobianco F., 2005) Some authors even take it further, and make a distinction between a GPL business model and a non-GPL one. (Krishnamurthy S., 2003). Krishnamurthy (2003) takes into account different actors and takes the point of view of a software producer to analyze the business.

Software producers can exploit the open source community in two ways. First, they can use OS code to create a new product. Second, they can also use the entire OS product, and bundle it with existing products. If the license is not GPL, the source code does not need to be released. An earlier mentioned example is the use of BSD code in Microsoft's operating systems Windows 2000 and XP. All they had to do was to acknowledge they used BSDlicensed code in their products.

What we see in this model, depicted in figure h, is that software producers probably use open source code, because of the low cost. The price they pay for this, is the fact that the used open source is still freely available to the end users. If not much added to the derived work, this may cause problems for them (in terms of copying their software).



Figure h: Non-GPL model (Krishnamurthy S., 2005)

In the model shown in figure i, software producers are legally bounded to release the source code of the derived product as well. There can still be money involved in the selling of the product, but the source code is freely available; thus a free version of the product might emerge easily. Another possible revenue stream is the offering of several services to the end user.



Figure i: GPL-model (Krishnamurthy S., 2005)

One can conclude that the difference between the two models is what the seller expects from the user. If the software producer is looking for an empowered user that engages in a *"two-way conversation"*, it should choose a GPL-model. The non-GPL model generally relies on an end user that just uses the product, and does nothing else with it. (Krishnamurthy S., 2003)

The original open source business models, before the emergence of OSS 2.0, can be found on the website of The Open Source Initiative. (Open Source Initiative)

Although there are more recent business models that are more in use today, it is still quite interesting to look at the original ones as they greatly influence the new business models. (Hecker F., 1999)

Support Sellers

Support sellers is the original free software model introduced by Richard Stallman in the GNU Manifesto. The model is usually used with the GPL out of historical reasons; however, most of the open-source licenses could be used with this model.

Revenues are generated in this model by the selling of physical goods and services. Goods can be for example hard copies of software, while services usually refer to post-sales services such as technical support or maintenance. Vendors thus can differentiate themselves on points such as quality of service. (Hecker F., 2000)

Loss Leader

In this model the open source product is used as a loss leader for traditional commercial software. The generation of revenue for the open source product can be very little or even none and is usually pretty similar to the revenue generation discussed in the 'Support Sellers' model; through the selling of goods and services. However, the open source product can be very useful in addition to the existing (proprietary) products. The open source product can enlarge the brand's name and reputation. It can make traditional products more functional and useful. It can increase the base of developers and end users familiar with and loyal to the existing product line.

As source code in the open source product might be used in other proprietary software within the firm, the right license should be chosen wisely. The GPL for instance, wouldn't be a great choice in this case. The MPL might be more appropriate. (Hecker F., 2000)

Widget Frosting

This model is intended for hardware selling companies. The hardware is in this case called the 'widget'. The open source software usually is enabling software, such as drivers, which are distributed without any charge together with the hardware. This is the 'frosting' part.

The generated revenue in this model obviously comes almost exclusively from the hardware sold. However, the use of open source software could increase the hardware sales as in the 'Loss Leader' model. (Hecker F., 2000)

Accessorizing

This business model is for companies who distribute books and other physical items associated with open source software. The only license feature required for this model is the ability to bundle free software with the item being sold. Hence, any open source license can be used in this model. Revenue is of course being generated by selling the goods themselves. (Hecker F., 2000)

Service Enabler

In this model, open source software is used to support access to online services. For this model, a company should choose an open source license that minimizes the risk of its code being used in proprietary software. So GPL, or MPL could do well for this.

Revenue is generated by the online service and thus vendors have the possibility to differentiate them based on the attributes of this service. (Hecker F., 2000)

Dual licensing

The most active business model for open source software producers nowadays, is without a doubt the dual licensing model. The model exists of two layers. On the one hand, there is the open source license, usually being GPL, which is freely available and modifiable as long as derivative works are licensed under GPL as well. On the other hand, there is the commercial license, which is just the same as a closed-source license. So an end user is presented with both an open source and a closed source option. (Gomulkiewicz R., 2004)

The difficulty for successfully implementing the dual-licensing model lies within creating a meaningful gap between the OS version and the premium-paid version, which is often called an 'enterprise edition' of the software. This gap can entail for example 24/7 support or just more advanced features that are not available in the OS product. This model with a free and premium version is often also called the 'Freemium business model'. (Driver M., 2010)

The advantages of the dual licensing are numerous. The free edition makes faster adoption of the product possible and thus broadens the user base. The possibility to have a trial period prior to the purchase of a product and take a look at the source code makes the dual licensing model a pain-free path for adopting companies. When choosing the OS license, a company can even use the software internally and further develop it without any restrictions. This is often more valuable than a money-back guarantee. (Koenig J., 2004)

The model creates a win-win situation. On the one hand, the open source community benefits from the influences of another strong product. Open source software is generally considered as strong and secure because of the thousands of people that have reviewed and improved it with a different mindset. On the other hand, commercial customers receive a product that is of a high price-quality standard. (Gomuliewicz R., 2004)

Many dual licensing models have developed from an original pure open source model to the new structure. A good example of this case is JBoss, which was initially solely a free product in accordance with open source principles. It evolved from a zero-revenue structure, through an education and consulting based model, towards the dual-license standard. (Watson R et al., 2008) Other influential examples of the model are MySQL, Sleepycat, Trolltech and Funambol.



Figure j: Dual licensing model (Välimäki M., 2003)

Figure j visualizes the dual licensing model in detail. The reader should look at this figure from the bottom up. At the lowest layer, end users can be divided into copyleft users (using the open source license) and customers (using the commercial license). The arrow from copyleft users to customers means that a certain percentage of copyleft users will buy a commercial license and become paying customers. Above the core product, there is a layer of developers. On the one hand there is the development community, which usually contributes by for example fixing bugs. On the other hand, there are the commercial development partners who develop essential components of the core products. (Välimäki M.,2003)

2.6. <u>Conclusions</u>

In this chapter, we have explained the concept of open source. We have defined the concept, looked at the different licenses available and discussed the several business models that have existed since its emergence. The business model that is most popular

today is undoubtedly the dual licensing model. This model has been explained extensively in the last paragraph. Open source is still gaining popularity these days, and a lot of literature is still being written about it.

3. Content Management Systems

To research the business models in ICT firms, we chose to focus on the market of content management systems (CMS). Although it is a relatively recent concept, there is a broad range of companies – both open and closed source – available. Therefore it was a very interesting sector for us to focus on. Nevertheless, not much scientific literature exists about this topic. One author, however, Bob Boiko, has managed to write a whole book about it. In his book 'Content Management Bible', published in 2005, he tries to give a standardized definition of content management. Therefore, this chapter is largely based on his book. First, we have a look at what content itself exactly is. Next we define content management and thus what a CMS should do. Finally, we base ourselves on a research conducted by Gartner to take a look at the market of CMSs.

3.1. What is content?

Content is often confused with data. The easiest way to create a distinction between the two is to incorporate the concept of human value. Data can be seen as bits of information that can be interpreted by a computer. Numbers, figures, words, letters, sounds ... they all represent the concept of data. Data however, is stripped from any context and does not have any real meaning. At this point, the aspect of human value comes in. Data does not have any contextual meaning towards human beings. This is where the difference with content lies. Content is, just like data, information, but information with human meaning and context. So content is actually raw information, which is given a usable form (often in terms of giving it a name). Information is not content by default. It becomes content when someone tries to make use of it.

Although the difference between content and data should be noticed, this does not mean that the two do not interact. When considering a content management system, content cannot go without data as a computer can only interpret data. Content therefore can be considered as information that is tagged by data, in that way that a computer is able to organize and publish it.

The most important attribute of content is without a doubt structure. If you don't control the structure of content, you won't be able to manage it. Therefore, creating clear structures is key to managing the content. It should be looked at the way that different components of content are put together and fit into an integrated framework. (Boiko B., 2005)

3.2. What is content management?

Before we define content management, it is useful to take a brief look at the history of web development. Initially, IT-specialists or web designers were building static websites. They were building websites writing plain HTML in a simple text editor. This situation evolved in tools making it easier for non-technical users to make websites. Programs such as Microsoft FrontPage and Macromedia Dreamweaver delivered environments that handled a 'what-you-see-is-what-you-get approach'. Text could be just typed into boxes, while images could be dragged into the right places, all while the back-end was providing the HTML code. Static websites are really the simplest kind of websites, suitable for small situation was the fact that content was actually being published by a programmer rather than for example a marketer. This resulted in poor quality of content. On the plus side, static websites are known to be very fast, as processing requirements are very limited. (Boiko B., 2005)

The second generation of websites is the dynamic websites, sometimes also known as database-driven websites. Whereas static websites were only plain files of HTML, dynamic websites consist of several components. A database (or XML structure) receives a request when a user clicks a link. The link activates a template page, which can contain HTML as well as programming scripts or other data sources. After the template creates the appropriate HTML page, the web server sends it back to the user's browser. (Boiko B., 2005)

The third generation of websites, is the world of content management systems. The basic idea behind the use of a CMS is described by Browning & Lowndes (2001): *"Fundamentally a CMS devolves control over content to the owners of that content, rather than the technician."* (Browning P. & Lowndes M., 2001, p.2) Further they describe the goal of a CMS as *"the increased integration and automation of the processes that support efficient and effective Internet delivery".* (Browning P. & Lowndes M., 2001, p. 4)

Content management systems consist of 3 basic functionalities: collection, management and publishing of content. Collection means that you can acquire information from different existing sources. Management means creating a repository that contains database records and/or files regarding content components and data. Publishing means making content available to, for example, web sites or printed media by extracting it from the repository. (Boiko B., 2005)

When publishing content is limited to the web, content management can mean just web content management. A definition of web content management found in literature is:

"Web content management (WCM) incorporates the activities involved in the creation and deployment of digital content to web based audiences, where these audiences may consist of customers, suppliers, partners and staff accessing web content via extranet, Internet or intranet." (McKeever S., 2003, p. 686)

In this definition, it already becomes clear that the potential of a content management system reaches far beyond the World Wide Web. A CMS can do more than just publish content to the web. This is also what developers of CMSs have realized. More and more firms are moving towards a system of what is called 'Enterprise Content Management (ECM)'.

A content management system provides a framework that makes it possible to deliver different information to different audiences. Web Content Management is only one of these types of information. ECM provides tools not only to publish content to the web, it can be any organizing system for content you want it to be. Organizations are more and more realizing that it is important to have a clear common structure in both printed publications as web publications. The ultimate idea behind ECM is to have all information resources of an enterprise flow into a single giant repository. Furthermore, this information should be available to the right people at any time. Although this is an extreme situation, and probably impossible to achieve, the idea has a lot of potential and is being executed by many companies across the globe.

3.3. How does it work?

The working of a web content management system is explained in figure k. The different components of the CMS have their own function:

- Repository: the repository lies behind the web server and is a relational or XML data source. It actually holds the content, the data and any of the resources needed to build the sites (e.g. graphics or style sheets)
- CMS-application: also behind the web server, this application collects the content from the contributors.

- Set of flat HTML files: the CMS manages and deploys the files to the static part of the website
- Live data source: this is the CMS-generated database. It is located on the web Server and is intended for the dynamic parts of the site.
- Other data sources: other sources of data can be connected to the web site that are not connected to the CMS. They can run independently of the CMS that accesses the sources.
- Templates: templates take care of putting data from any sources in the right format it needs to be in for the site.



(Boiko B., 2005)

Figure k: How does a CMS work? (Boiko B., 2005)

3.4. <u>The Market</u>

To analyze the market of CMSs, Gartner periodically publishes a Magic Quadrant for Web Content Management. The result of the most recent analysis, performed by MacComascaigh, Murphy and Tay (2011), can be found in figure l. Further findings from Gartner's research include the following: (MacComascaigh M. et al., 2011)

- The WCM market is growing fast: Gartner projects an annual growth rate of 14 % from 2009 to 2014.
- WCM now focuses on Online Channel Optimization: OCO is the discipline of maximizing the impact of engagements with the target audiences over a variety of communications media.



Figure I: Magic Quadrant for Web Content Management (MacComascaigh M. et al., 2011)

Although not mentioned in Gartner's quadrant, WordPress, Joomla and Drupal actually have the biggest market share in web content management. With about 54 % of the sites (that are in fact using a CMS) using WordPress, they have the biggest market share. Joomla and Drupal are the next in line with respectively a 9.2% and 6.7% market share. In comparison, OpenText and Sitecore only have a market share of respectively 0.3% and 0.1%. Nevertheless, OpenText and Sitecore are considered as market leaders thanks to their out-of-the-box solution. (W3Techs, 2012)

Four trends are being noted in the WCM Market: (MacComascaigh M. et al., 2011)

- 1. The need for business impact: In the past overall efficiency and productivity was a main goal, but nowadays WCM is more shifting towards an emphasis on business outcomes: revenue and profit.
- 2. Technology that works across channels: The growing importance of social media and mobile channels cannot be ignored and so enterprises need to incorporate multiple touchpoints into their online strategies. In the future, links with Facebook or mobile apps will be a minimum requirement for WCM.
- 3. Integration with related technologies: To be able to be used as a tool more, WCM should be closely integrated with adjacent technologies such as CRM, ERP and ECM.
- 4. Growing interest in cloud-based WCM: Companies such as CrownPeak and Clickability have already highlighted the benefits of Software as a Service (SaaS) for many years, but many vendors continue to take a 'wait and see' approach.

Gartner also supplies us with a more contemporary definition of WCM:

"WCM is the process of controlling the content to be consumed over one or more online channels through the use of commercial, open-source or hosted management tools based on a core repository." (MacComascaigh M. et al., 2011, p. 4)

4. Software as a Service

Software as a Service (SaaS) has already been mentioned a few times throughout this thesis. Both industry watchers and experts we had the chance interviewing, pointed this out as a trend in the industry. Therefore, it seems useful to briefly define what SaaS is.

4.1. Definition

SaaS is one of the most popular topics in business models in ICT firms today, particularly when considering the delivery component. SaaS is one of the three basic models within cloud computing, next to Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). SaaS is of particular interest to our research as it is not only a general trend within software, but also especially popular in the sector of content management systems. Chong & Carraro (2006) start defining SaaS by using its most simplistic definition: *"Software deployed as a hosted service and accessed over the Internet."* (Chong F. & Carraro G., 2006, p. 2)

By this definition, even web-based email services such as Gmail or Hotmail are seen as SaaS. Two major categories can be distinguished within SaaS applications:

- Line-of-business services, which are offered to enterprises mostly as a customizable and modular product. They are often sold on a subscription basis.
- Consumer oriented services, which are offered to the general public, usually at no cost and supported by advertising.

Three main characteristics can be identified for SaaS applications, and cloud computing in general:

- Scalable: an application should be able to maintain the desired quality level for changing system loads by adding or removing servers without the applications have to be adjusted.
- Configurable: an application should allow a certain level of modification. This
 offers the different customers the possibility to use separate instances of the same
 application code. This way, the vendor is able to fulfill the different needs of each
 customer.

 Multi-tenant efficient: multiple users (tenants) should be able to access the software at the same time. A multitenant architecture makes it possible for a software application to be designed to virtually partition its data and configuration. Furthermore, each client's organization works with a virtual application instance.

An important advantage of SaaS is described by Chong & Carraro (2006), namely 'selling to the long tail', a concept originally described by Chris Anderson (2004). The idea is that online sellers such as Amazon can serve a very broad public in a cost-effective way thanks to their unique positioning. The classic example describing the phenomenon, is that of a traditional book or cd store that is not able to fill in demand of less-popular cd or book titles. They concentrate on selling the most popular products, as they have limited stock space available and cannot be in danger of being overloaded with an unwanted stock of a particular item. Online retailers are not being faced with this problem and can sell a rarefound article as well as a popular one. The access to this 'long-tail' of low-volume sales generates a huge amount of revenue. (Chong F. & Carraro G., 2006) This is being clarified in figure m, taken from Chong & Carraro's paper.



Figure m: The "long tail" (Chong F. & Carraro G., 2006)

How this relates to SaaS applications is quite obvious. Because of the economies of scale that a SaaS vendor can achieve and the lowered requirements towards servers and other hardware from the customer's perspective, SaaS providers get access to a new range of customers compared to traditional software vendors. The 'long tail' becomes a target group that is now cost-effective to serve.

PART 2: RESEARCH

1. Research method

Our research is of qualitative nature. To start our research, Mrs. Mahajan shared with us a database of contemporary content management systems. The database has been split up in three parts. A first distinction that has been made, is obviously between closed and open source. A second distinction is the one between community based open source and corporate open source. Generally, corporate OS are the ones making a profitable business out of open source, usually by offering both a free version and an enterprise version of their product. Community OS, generally are the CMSs developed by a community and offered under a free price tag solely.

The goal of our research is to find an answer to the following research question:

What are the differences and similarities in business models in ICT firms and what are the transitions and evolutions in the sector?

Therefore we wanted to interview a number of companies in each of the sectors. We made a basic email to convince companies to allow us to conduct an interview. By preference, we wanted to do some face-to-face interviews, but due to travel distance, most of the time this was not possible. In this case, we usually first sent a small basic questionnaire to the company. Following to that mail we then conducted an interview via Skype or by phone in which we asked some in-depth questions based on the answers of the questionnaire.

To find out what is the business model of a certain company, we set up a list of indirect questions, which allowed us to obtain a detailed insight in the four pillars of our business model described in the literature review. For each pillar (product, infrastructure, customer and financial) we made a set of questions that could be used during the interview as a checklist of criteria. This list of questions can be found in appendix 1.

The companies were selected based on the following criteria: the number of downloads, the number of releases and the country of origin. After making a broad selection in every field and based on the response we got, we ended up with the following list of companies:

- For closed source we interviewed Sitecore, Hannon Hill, Oxcyon, Anonymous X and Accrisoft.
- For corporate open source we had meetings with Umbraco, Concrete5.
- As far as community OS is concerned, we had appointments with Fork CMS, Plone and Appalachian State University's phpWebsite.

The analysis of the interviews was done by applying each company to the 9 components of Osterwalder's business model. This analysis can be found in the next chapter. Per company we made both a detailed analysis and a summarizing overview. The detailed analysis not only entails the 9 business model components, but also a section with important remarks from the interviews. These are some quotes or visions that we consider as remarkable and discuss in detail.

Next, we made a conclusion for both open source and closed source based on what was found in common between the interviews. This way, we could make two distinct business models. The conclusions also include some important insights or remarks we thought to be interesting. Finally, the most important evolutions noticed by the interviewees are summarized as well.

Accrisoft

Accrisoft is a proprietary company that offers both a CMS (called Freedom CMS) and a CRM (customer relationship management) product. Accrisoft takes care of the hosting, and offers free training and support. They offer a Software as a Service solution, and pricing is based on a monthly subscription fee. They also offer a 30-day free trial of their CMS. We had an elaborate interview with Ben Hodous, vice president of communications at Accrisoft.

AppState/phpWebsite

phpWebsite is an open source CMS developed by Appalachian State University. Its main goal is to provide Universities, corporations and individuals with a complete site content management solution that is flexible, robust and extendable. The phpWebsite project started out as a program that allows students to explore technologies beyond what is covered by the standard curriculum. So it is mainly developed by IT students. Therefore it is somewhat an outlier in the field of CMSs and so it was interesting to have an appointment with them. Brian Brown, the initiator of the phpWebsite project, was kind enough to make some time available for us to ask him some questions by phone.

Anonymous X

Anonymous X is a company offering a XML-based enterprise content management system. An XML-driven content strategy really is a niche market in the world of CMSs. Their business model is also SaaS-based. For these reasons, it was interesting to contact them. Eric Kuhnen, director of product management at Anonymous X, provided us with answers to our questions. Their product is primarily aimed at developers having knowledge of XML.

Concrete5

Concrete5 is an open source CMS that offers a free version of the CMS and a lot of services. Hosting, support, consulting, training and enterprise services are available options. Concrete5's slogan is: "CMS made for Marketing but built for Geeks'. An important part of their product is the marketplace. A great number of add-ons and themes are available. These are sometimes free, but can come at a cost as well. Their showcase includes websites such as Glimmerglass Festival, R magazine, Mail Away and Genome British Columbia. Franz Maruna, CEO of concrete5, was available for an interview via Skype.

Fork CMS

Fork CMS was selected for a number of reasons. First of all the original creators were based in Ghent, so it was possible to set up a face-to-face interview. Second, it is a very new CMS, and so it would give us the possibility to gain insights on how communities are being built and developed. Fork CMS is growing rather rapidly, being adopted by a number of web developers in Belgium and communities are spreading over the world. Examples of websites running Fork CMS are the sites of Ancienne Belgique, Vorst Nationaal and IBBT. Bart De Waele, CEO at Wijs, the web design company that originally developed Fork CMS, was the one who was available to us for an interview.

Hannon Hill

Hannon Hill is another closed source company offering a professional enterprise CMS. Their product is called Cascade Server and primarily targets the sector of higher education. Their CMS is therefore certainly optimized for use at schools and universities. They offer a lot of services such as hosting, support, consulting and migration services. Their client list includes University of Scranton, California University of Pennsylvania and North Carolina State University. Dean Smith, a sales consultant at Hannon Hill, had a very interesting conversation with us about their product.

Oxcyon

Oxcyon is another American closed source company offering an all-in-one CMS, called Centralpoint. They claim to have a unique approach towards content management, bringing a Service-Oriented-Architecture to content management. Oxcyon can be considered a pioneer in the world of content management systems, and has a patented technology called 'horizontal propagation'. Sam Keller, the CEO of Oxcyon, did not only have a very interesting conversation with us, but also provided us a lot of useful ideas towards our research.

Plone

Plone is a community based OS company that has been around since 2001. Plone is among the top two percent of all open source projects worldwide. They claim to have the best security track of any major CMS. With organizations and companies such as Connexions, the Brazilian government, the FBI and NASA using Plone CMS, this is not hard to believe. Two board members of Plone, Sjoerd van Elferen and Rob Gietema are working at Four Digits, a company focusing on web design and development via Plone CMS based in Arnhem, The Netherlands. They were kind enough to have us come over for a face-to-face interview.

Sitecore

Sitecore is a well-known closed source CMS. They claim to be the best ASP .NET CMS available. As a proprietary solution, they have a high-end product, offering an enterprise website and an intranet portal. Furthermore, they offer a digital marketing system and have an 'App center'. Their product can be tightly integrated with Microsoft SharePoint, which is an attractive order winner towards many companies. We had a very interesting interview with Eddy Lalou, the regional sales manager for Belgium and Luxemburg.

Umbraco

Umbraco is an open source ASP .NET CMS. In the sector of .NET based CMSs it is one of the biggest players. Next to their freely available CMS, they offer commercial products which can be tailor-made. The company employs a number of people full time. They work in what's called Umbraco HQ. One of these people is Tim Geyssens, who works in Ghent and was happy to meet us to conduct an interview. Some well-known examples of sites running Umbraco are Ijsboerke, Heinz, Jaguar and Sandisk.

2. Analysis

2.1. Closed source companies

Accrisoft¹

Customer Segments

In general, Accrisoft doesn't aim for a specific target **market segment**. To cite the Ben Hodous, "*any company or individual who develops a site, is seen as a potential customer.*" Nevertheless, certain markets are more represented in the customer base. These include parts of governmental groups, and membership-based organizations.

Although there is no concrete **market segmentation**, a distinction between low-, middle-, and high-end customers is made through the offering of 3 distinct products.

Value Proposition

Depending on the specific needs of a customer, Accrisoft offers 3 different kinds of distributions of their software. These differ mainly in the availability of features and number of modules. Their different packages are the starter package, the basic one and the unlimited package. Ben Hodous told us the unlimited package is chosen the most. Next to this, customers also have the opportunity to try out the software of Accrisoft via an online demo, which is available for trial during 30 days. This is a good example of software that can be classified under shareware.

An important part of the value proposition is Accrisoft's **quality assurance**. Accrisoft believes quality can be assured better in closed source than in open source. The underlying reason being mainly that developers not only are often better qualified, but it is very important that those are the same developers. The people on their team have built not only one part of the software, but all of it. This results in a better aligned end product.

Another key part of Accrisoft's value proposition is its impeccable **support**. As customers are expecting this from a closed source software vendor, they just have to perform in this field.

¹ All citations made in this analysis are from the interview with Ben Hodous (appendix 2), April 20th, 2012.

Accrisoft uses a **SaaS-based delivery** strategy. They believe that this delivery strategy offers extra value to their customers, as the end users do not have to worry about server management and so on.

For the future, Accrisoft believes **automating processes** can offer significant value to the customer. As an example, one click-installation offers an instantaneous automated service without the customer going through the whole (technical) process.

Channels

To raise awareness about the company's product, Accrisoft mainly uses **social media**. They also go to trade-shows and sometimes put ads in some magazines. Nevertheless, their most important marketing tool is **word-of-mouth**, which is stimulated by Facebook, Twitter and other micro-blogging sites.

The most important part of their channel strategy, is the indirect contact with the enduser. Every interaction concerning the purchase and deployment of the software, goes through their **partner network**.

As Accrisoft offers the possibility to the web development bureaus to deliver the software as **SaaS**, this can be seen as the way the value proposition is delivered to the customer.

Customer Relationships

Accrisoft has a relationship with web development companies, and not with the clients of those companies. Direct feedback from the end-user of the product is **non-existent**. However, they do gain feedback via their partners through the **Ecosystem**. The Ecosystem also provides a feature request form, through which Accrisoft enquires about the new needs of the customer regarding the product.

Another important relationship with the customer is concerning the **support** Accrisoft offers. On the one hand, they have a kind of personal assistance with the web development companies to set up and make a certain customer project possible. On the other hand, they are also trying to automate the support they offer through videos and workbooks with demo sites.

Key Resources

The **quality of the developers** emerged as an important aspect through the interview, as they depend on the skills and knowledge of their employees for the development of their product. For a closed source company, it is important to have **intellectual resources**, and for a company such as Accrisoft, patents, copyrights and proprietary knowledge are key components of their business model.

At Accrisoft they are convinced of the importance of having a **good support** channel. Picking up the phone on the first ring and being able to answer customers' questions are seen as key factors in the competition with open source.

Key Activities

"We try to do all development in house." This way, Accrisoft wants to assure the quality of the product. This can be seen as *the* most important activity. Furthermore, they offer **training for new people**. This training is free of any charge and happens directly (via meetings or on the phone) or indirectly (via videos and workbooks with demo sites). As sales happen through partners, another key activity is setting up new and maintaining their existing partner relationships.

<u>Key Partners</u>

Accrisoft solely relies on partners regarding the sales of their product. These partners are web development companies. The process of becoming a partner is rather simple. Accrisoft just wants to make sure a company or individual has the right web development skills. They call their partners 'solution providers'. To maintain contact with them, they have set up a reseller partner portal, which is called the 'Ecosystem'. It is a community for the partners, where they have marketing materials, best practices, etc. at their disposal. This community works in two-ways, it is also a way for Accrisoft to gain information and feedback on their end-customer via their partners. It is, however, no kind of community in which development and code-sharing are available, as all development is happening inhouse.

On the other hand they work closely together with **hosting companies**. As they have a SaaS based product, this certainly is an important part of their business model.

Cost Structure

As Accrisoft uses a third-party **hosting** network to offer their SaaS-based product, their main cost is hosting. Next to this, the most important costs include **personnel** and **development**.

Revenue Streams

Accrisoft mainly raises revenue through its **monthly fee** charged for their SaaS product. This fee includes the software license, hosting, server maintenance, upgrades and support. The different products have a different pricing structure. Furthermore, there are **nonrecurring revenues** generated through the sales of professional services.

Other important remarks from the interview

Support is seen as very important to closed source software vendors, which most often cannot be guaranteed in open source solutions. An evolution noticed by Accrisoft is that *"people are realizing they need more training, support, hosting and somebody to help them"*. It is impossible to do all these things by themselves.

Most of the clients at Accrisoft have a background of building a business with open source. *"They have tried to build a business with open source and bumped into various problems, especially when the business was growing to a large size."* This pushed them to make the switch to a closed source solution.

As a final remark, Accrisoft is seeing an increasing number of companies offering support to an existing open source product, and companies building on top of open source. Many organizations nowadays are also starting to build a hosting business around an open source product. As this lies closer to what Accrisoft is doing, they are facing increasing competition from these kind of businesses.

Key Partners	Key Activities	Value Proposition		Customer Relationships	Customer Segments
Sales Partners	In-house development	Different packages		No direct relationship with	No specific market segment
Reseller Partner Portal	Training	Quality Assurance		end-user	
					Low-, middle-, and
Hosting companies		Impeccable support		Ecosystem partner community	high-end product
		SaaS-based delivery			
				Offered support	
	Key Resources	Automating processes		Channels	
	Quality of developers			Social media & word-of-mouth	
	Intellectual resources			Partner Network	
	Good support channel			SaaS	
Cost Structure Mainly third-party hosti	ng	Internet	Revenue Streams Monthly fee	e for SaaS	
Personnel & development			Non-reccuring revenue for professional services		

Figure n: Business Model Accrisoft

Anonymous X²

Customer Segments

Anonymous X is aiming at a **niche-market** with their XML CMS. The market is rather small, but there are a number of other players in the segment. As far as different industries are concerned, Anonymous X aims at **medical device**, **high technology**, and **discrete manufacturing** industries. For the different markets they go after, they may choose a different price target; however it is the same product.

Value Proposition

Anonymous X's CMS is really a niche player in the CMS market, as they offer an **XML CMS**. The market is even this small that there is not really a possibility for open source to work innovatively in the segment. Eric Kuhnen mentioned that "*There is almost no competition form open source. There are one or two XML content management systems, but I could not name them.*"

One of the key features of Anonymous X's CMS is the fact that it operates **very reliably**. They put a lot of effort in developing a quality product. Their main focus is on the product itself, not on professional services. Along with the **SaaS-based** service, this is where Anonymous X excels. Anonymous X's SaaS based model is rather particular. They work with a **single-tenant** model, so security can be assured to the customer.

<u>Channels</u>

They both have **direct and indirect** channel strategies to provide their product to the customer. Although this could be seen as being competitors with their own partners, the indirect channel gives Anonymous X the opportunity to operate in markets in which they have no expertise.

Customer Relationships

Due to a small market, it is possible for Anonymous X to have a close relationship with their clients. If possible, they try to meet clients on a **weekly** basis. Because of this personal contact, Eric Kuhnen could assure us that *"Anonymous X has very good customer relationships and a very loyal customer base."* These conversations lead to information regarding new customer needs and features they would like to see being integrated in the product.

² All citations made in this analysis are from the interview with Eric Kuhnen (appendix 3), May 4th, 2012.

Key Resources

Anonymous X uses both **in-house and external human resources** for development. Developers need the necessary skills and knowledge (about XML). Furthermore, they employ skilled local salespeople in different areas. This way they believe they can operate more effectively in different areas.

Anonymous X does not have a lot of physical resources, as they outsource hosting of their SaaS-application to **third-party hosting** companies.

Key Activities

A large part of **development** happens in-house. That way they can both ensure the quality of the product and the path Anonymous X wants to take. Both external contracted developers and in-house developers are brought together in a **community**. Managing this community is an important activity.

Anonymous X also offers a number of **professional services**. Although they stress the fact that their business model is mostly based on the use of the system, such as like training and support are of vital importance.

<u>Key Partners</u>

Anonymous X works with multiple sources of partners. They have partners at the **development** side and at the **sales** part of the business. Anonymous X has many contracted developers working on the software. To bring the product to the market, Anonymous X also chooses an indirect way via their partner network as they believe "*It is more efficient to use local resources than to put our own resources in a certain geography.*" Becoming a partner of Anonymous X usually happens through reference.

Cost Structure

Anonymous X did not want to elaborate a lot on their cost structure as they consider it proprietary information. They certainly have to take into account **personnel costs** and costs for their **third-party hosting** network.

Revenue Streams

Anonymous X's most important revenue stream is the one generated from the selling of their **service**. They may set different prices for different markets. Their pricing is based on usage. This means the price tag is calculated based on the number of users on the system.

Another revenue stream that should be mentioned is their offering of professional services. Although not considered as important as the core product, they should still be incorporated in the model.

Other important remarks from the interview

Eric Kuhnen made a remarkable comment on open source. He told us that "Open source is not really a business model as much as a development model". He believes that open source is just a way of developing software, but from the moment there is a usable product, their business becomes the same as a proprietary one. "You still have to make money if you're going to service it in some way. Otherwise people can't continue to develop it for nothing. So there is some revenue model, and I think they perform relatively the same [as proprietary]."

Considering innovation in the future, Eric Kuhnen believes *"the market is figuring out two things."* The first one involves the collaboration of end users in the creation of content without polluting it. The second one deals with simplifying the contribution of content by subject experts. This means that people without extensive XML-knowledge should be able to manage specific content in the XML-CMS.

Key Partners Development partners Sales partners	Key ActivitiesImage: Construct of the second secon	Value Proposition XML CMS Reliability Profession Single-tena security as	al services ent model: surance	Customer Relationships Face-to-face contact on weekly basis Channels Both direct sales and indirect sales	Customer Segments Niche market (XML CMS) Different industries: medical, high- technology and discrete manufacturing
Cost Structure Personnel Costs Third-party hosting net	work		Revenue Streams Different pi	rice targets for different n	narkets

Figure o: Business model Anonymous X

Hannon Hill³

Customer Segments

Hannon Hill is primarily focused on **higher education**. Over 175 colleges and universities are using Cascade Server as their CMS. Cascade Server is available under three different licenses that are aimed to target both smaller and bigger universities. There is the **standard** single-CPU license, which is used in about 75 percent of the cases and is aimed at smaller and medium-sized colleges or universities. Next to this, there is the **enterprise** license, which is for a machine running Cascade on 2 processors. This is aimed at the larger schools like Indiana University and Michigan State University. The newest option is the **hosted** license, which is becoming more and more popular.

Next to higher education, Hannon Hill has a few more big clients, but this is certainly not where their focus lies. It is important to notice that Hannon Hill is targeting primarily **end-users**, and not developers.

Value Proposition

Cascade Server offers a lot of value to the end customer. It comes as a true **enterpriselevel CMS** with unlimited users and unlimited sites at no additional charge. Next to the product license, there is **support and maintenance** available, which annually comes at 20 percent of the license cost.

The three different licenses available, briefly discussed in the previous paragraph, make it possible for Hannon Hill to offer the right value to the right customer. The **standard** license uses a single dual- or quad-core CPU, while the **enterprise** version makes use of two dual- or quad core CPUs. The **hosted** version is growing more and more popular nowadays. It offers significant value to the customer, as clients can become less dependent on their IT department. Hannon Hill offers each customer choosing the hosted license an own **dedicated server**. This is what is fundamentally different to their service and software as a service. *"This offers as advantage that clients can choose not to upgrade, whereas in SaaS you do not have that option."*

Another part of Hannon Hill's value proposition is their so-called "**QuickStart** package". They offer new schools this possibility "to get started up and running with Cascade as *quickly as possible.*" The package includes two days of training and one hundred hours of professional services. QuickStart has proven to be quite popular. "Over the last two years alone, we've accomplished over 50 QuickStart packages for our clients."

³ All citations made in this analysis are from the interview with Dean Smith (appendix 7), April 25th, 2012

Following to what they believe is the newest market trend, Hannon Hill has developed their own web-marketing tool called **Spectate**. The product has been built from the ground up, based on feedback from their higher-education clients. *"Spectate is going to show if your marketing efforts are paying off."* Spectate is offered to Hannon Hill's existing clients for free and will be integrated with Cascade in the future.

A last important feature of Hannon Hill is its very useful **client community**. As they have a very specific client base, it is useful for them to have the possibility to interact and share visions and thoughts.

<u>Channels</u>

Hannon Hill primarily uses a **direct way** of going to the customer. They focus on marketing efforts and cold calling. Next to this, Hannon Hill also uses **some partners** to do some reselling, but most clients are approached directly

Customer Relationships

A very important part of Hannon Hill's vision is managing customer relationships. They try to be on top of the feedback they receive from their clients and do this in various ways.

Firstly, they have a **client advocate team**, which sole job is to keep the clients happy. Dean Smith believes that from what they have seen *"spending that extra money to that client advocate will pay off in the long term"*.

Secondly, clients get a voice in the future development of Cascade Server. Through the **client community**, it is possible to give feedback and suggestions, start discussions and even share code.

Finally, Hannon Hill organizes face-to-face meetings with their clients through user **conferences**. Dean Smith describes it as *"the highlight of our year"*. They discuss new ideas, learn from each other and just have a great time with their clients.

Key Resources

A first important resource for Hannon Hill concerns the physical infrastructure, more specifically the **servers** for their hosting service. Furthermore a lot of attention is paid to the selection process of the **personnel**. Positive, self-starting and supportive people are employed and can be seen as important human resources. At last, as for every closed source software company, it is important to protect **intellectual property**. This is mainly done by patents, trademarks and/or copyrights.

Key Activities

Development for the Cascade Server is certainly a key activity for Hannon Hill. Next to development, **marketing and sales** are, of course, an important part of Hannon Hill's daily routines.

What is very important to Hannon Hill, is the ways of getting **feedback** from the client, and maintaining its loyalty. Therefore the support staff closely manages the client community.

<u>Key Partners</u>

Hannon Hill works closely together with different partners. Despite the fact that most of the sales of Hannon Hill are done directly to the customer, several resellers are seen as **strategic partners**, as they can service another geographical region within the United States. Hannon Hill does not focus on the design services of the software. For this part they have to count on the collaboration with **design partners**. *"If a school comes to us for design work, we will definitely send them to a design partner. If a partner design company is talking to a school wanting a CMS, they send them our way."*

Cost Structure

The main costs of Hannon Hill concern both hosting costs as personnel costs. As the hosted solution of Hannon Hill becomes more and more popular, the costs for the **servers** are very high. Also the costs for the development, marketing, sales and client advocate **staff** are considerable.

Revenue Streams

Hannon Hill's revenues are generated in different manners. First of all, they have revenue from the **license** sales. Furthermore, customers have to pay twenty percent of the license fee for **maintenance and support** each year. A popular option in the offering of Hannon Hill is the **QuickStart package**. This package aims for new customers (schools) to set up and get a Cascade system up and running as fast as possible, while two days of training and one hundred hours of professional services are also included. This popular package can be seen as a third revenue stream for Hannon Hill. Additionally, also separate purchased **service hours** is a way of generating revenues for Hannon Hill.

Other important remarks from the interview

In their market segment, Hannon Hill has noticed that their clients want their IT departments to focus more on highly technical tasks, and less on content management.

Therefore they chose to include the hosted license option, which *"gives more ownership of the CMS to the marketing department instead of having to go to the IT department"*. This part of the offering has seemed to be quite popular in the last few years.

Although they believe in the concept of SaaS, they feel that their hosted Cascade customers need more flexibility than a SaaS-based solution can offer. They are convinced that having a dedicated server for each client offers a lot of advantages over having running a single instance.

As an evolution in the CMS market, Dean Smith mentioned not only web-marketing tools such as their own Spectate, but also the integration of social media and responsive designs. These kinds of designs make it possible for websites to adapt themselves when the user shrinks or enlarges the browser window.

There is not really a higher-education centric open source CMS in the market available right now. Hannon Hill does notice that some schools are using Drupal or Joomla. Therefore they can be seen as their main competitors in open source. The fact that commercial support often lacks at those companies is a competitive advantage for Hannon Hill.

Key Partners Strategic partners (resellers) Design partners	Key Activities Image: Constraint of the second of	Value Proposition True enterprise-level CMS Support & maintenance Standard, enterprise and hosted version Hosted version: dedicated server Quickstart package Web-marketing tool: Spectate	ise-level CMS aintenance aterprise and on: dedicated ackage ing tool:	Customer Relationships Client advocate team Client community Conferences Channels Both direct way and partners	Customer Segments Higher education Segmenting between small and big companies via different licenses Rather end-users than developers
	Patents, trademarks & copyrights	community			
Cost Structure Servers Staff			Revenue Streams License sales Maintenance Quickstart po	e and support (20 % of license ackage	e fee)

Figure p: Business model Hannon Hill

Oxcyon⁴

Customer Segments

Oxcyon's only real segmentation is that it targets the **larger companies or organizations**. They have developed a **high-end product**, and this of course comes at a price. For the big players, three different licenses are available. A 'Master Enterprise License' is being used to license a very big company (e.g. 3M or Ford Motor Company). It is synonymous with Microsoft SharePoint Portal Services, where under one server any number of portals can be created. Next there is the 'Enterprise Portal License' for the large multinationals or school districts, which is priced lower than the previous. To target the 'smaller' players (e.g. a 200-men company), there is a third license, the 'Single Site License', which has a significant lower price than the previous two.

Value Proposition

Oxcyon offers an out-of-the-box solution. They have the **largest available module gallery** with over 230 modules and tools. Oxcyon has some unique features and uses its own ways to deliver a quality product. Sam Keller believes that *"building content management is a top-down process."* This means they put a lot of effort to know the needs of even the smallest client and bring a scaled product in a **modular** way.

Another key differentiator in Oxcyon's CMS is the use of their patented technology **horizontal propagation**. This entails they install a portal of Centralpoint at one of their clients' facilities, which has the ability to create 'children-Centralpoints'. That 'child' can even have grandchildren and great grandchildren sites. *"Centralpoint is very core; it creates portals, which create portals, which create portals..."* What is extremely valuable to this situation, is that if Oxcyon releases a module, they can release it to all Centralpoints and all the children and grandchildren at the same time.

Using the partner network, Centralpoint is being **verticalized** to a specific sector. The partner figures out what types of content the companies or organization manage, and the types of audiences that the portal serves. This way, a very specific product is being delivered to different sectors such as health care, legal departments, scientific communities...

Oxcyon offers different licenses. As previously discussed, they have licenses for **small**, **medium** and **large** enterprises. Common in these three licenses is that all modules come

⁴ All citations made in this analysis are from the interview with Sam Keller (appendix 8 & 9), April 30th, 2012.

in a *"kind of buffet style, all-you-can-eat"*. Sam Keller believes that modules *"should not be offered 'à la carte', like a Chinese menu"*. So they give their clients all the modules.

In the pricing, **service** is also being included when they are selling directly. Oxcyon tries to focus its sales on the partners though, to avoid they become competitors with their own resellers.

<u>Channels</u>

Oxcyon is both using direct and indirect channels to deliver Centralpoint to the end customer. Using the indirect channel, partners are **verticalizing** Centralpoint to their market segment. This concept has already been explained in the value proposition. Oxcyon is also delivering **directly** to the end customer. This is often happening because the client wants it. They make contact with Oxcyon to ask for a direct sales channel and prefer not to use one of their partner. Oxcyon does not focus however on these direct sales. They want to avoid becoming competitors with their own distributors. Therefore they focus on building a better product.

Customer Relationships

Customer relationships are also being handled both directly and indirectly. Via their **partners,** clients get to tell Oxcyon what they need. It's what they call the voice of customer, and Oxcyon believes this is key for their business. Horizontal propagation makes it possible to quickly incorporate clients' needs all over the world.

Oxcyon also makes use of an **issue management system**. They try to centralize all information coming from their clients in this system. Centralizing information makes it possible to process it more efficiently. So phone calls, interviews, surveys are happening, but all the info is ending up in the issue management system. To stress the importance of client feedback, Sam Keller made the following comment: *"What should happen is, listening to voice of customer so that it organically becomes the research and development."*

Key Resources

Oxcyon can protect its intellectual property through a remote connection with all of its products. Through this connection Oxcyon can update and add new tools to their software. Although, when they verify a system via that connection as not legitimate, they have the ability to deactivate an installation all over the world. This way people are not able to infringe the **licenses** of Oxcyon.

The need for **infrastructure** is also quite big, as Oxcyon needs development and production management infrastructure and hosting facilities to fulfil the evolution and offering of the product.

"Our most important resources are our people." This illustrates the importance of the **staff** at Oxcyon, e.g. development staff, architecture staff, sales staff, etc. They all have to possess the necessary skills within their field.

Key Activities

"You should really just focus on building a better product, instead of competing with your own partners for the delivery of your product." The **development** of the product is the most important activity for Oxcyon, while delivering the product to the end users is done via partners. Development is done in two ways: the first, directly by Oxcyon, are the general release modules aiming for the whole horizontal system. The second is the development of specific features uniquely for a particular vertical.

Training the partners is a second key activity for Oxcyon. "Our partners are given full training in terms of everything they need to know to manipulate the technology, in terms of verticalizing these solutions for their local market." They are also provided with samples, case studies and documentation to successfully present the products to the clients.

A last important activity for Oxcyon is the **gathering of feedback** from their customers. This is all collected in one central database, the issue management system. *"By funnelling all the issues (coming directly from the customers via phone calls, interviews or surveys, or indirectly via the partner network) through one central hub, you will be able to see patterns, and the response to that intelligence can be very accurate."*

Key Partners

Basically, Oxcyon works together with two kinds of partners. On the one hand, they cooperate with **technological partners**. These partnerships aim for optimizing their product to a specific technology, e.g. integrating the system with lab systems. The second kind of partners, are **value-added resellers**. These are seen as very important because these partners that will relicense the product to the end user, while offering additional services such as consulting services. *"When I say verticalize, I mean that our partners translate those vertical types in the market they serve."* They will build modules that add and create value for the product in their own vertical. When they are contacted by a client for a specific project in a vertical they do not have a partner in, initially they will develop

the project themselves for strategic reasons, but afterwards, when the segment shows enough potential, they will look for a partner within that vertical.

Cost Structure

"The most important cost for our company is our people, our staff." These are development **staff**, architecture staff, sales staff etc. These people keep Centralpoint running, while developing, handling and responding to issues. Furthermore, **infrastructures** for development and production and for hosting services are some big costs as well.

Revenue Streams

When a partner is interested in a specific vertical, he will have to pay money to Oxcyon to get the **privilege of that vertical**. Moreover, for every client a partner attracts, Oxcyon receives **a fee for every sublicense**. These are two of the revenue streams for the company. Sometimes they also offer a product directly to the customer, as they may not have a specific partner in that vertical yet. For this **service and license**, Oxcyon will also generate revenue. Three different licenses can be distinguished in the offering of Oxcyon: Single Site License, Enterprise Portal License, and Master Enterprise License. They all aim for a specific size of customer with specific needs. Oxcyon does not sell their licenses based on the number of users, modules or processors. On the contrary, pricing is based upon size, scope and structure of the overall enterprise: custom development, integration of third party data, and the overall breadth of the solution.

Other important remarks from the interview

During the interview, Sam Keller had a clear opinion on open source. He identified himself as an opponent of the concept, making multiple statements to exemplify his vision. "Open source is not really free. You still require an architect to implement and finish the project." With this statement he wants to counter the general opinion of open source being cheaper than closed source or even free. Furthermore, he thinks the lower price comes at an important quality cost as he says "Open source is really the worst. It is not properly architectured. It is unfinished." This is also the result of open source developers without foresight, vision and experience in the evolution of such kind of product.

Another remark concerning open source deals with the job security of developers in companies using open source software. He believes that they are convincing their company of using open source, because they will be the only one having the required knowledge about the specific implemented system. That way they build job security. Sam Keller makes the following metaphor for this: *"To most companies, this is a very attractive*

offer, until they realize they didn't buy the bread they needed, but instead they bought the 'baker', who is required for a long time, to keep the bread coming, and hope his recipe is good."

Sam Keller's final remark regarding open source is that it is "too loose, not controlled and it can fail".

Another strong opinion noticed during the interview, is Sam Keller's vision on the concept of Software as a Service and cloud-based solutions in general. *"You cannot expect clients to have their valuable content stored somewhere in Nebraska. That is not going to work."* He emphasizes the dangers of storing content in the cloud. Some content is just too confidential or sensitive and should be stored locally and highly secured. This issue is being handled in Oxcyon's idea of horizontal propagation, which has a lot of commonalities with SaaS, but excludes the downside of having content exposed. Sam Keller further believes that *"SaaS is only a temporary measure to fill a market need, but eventually organizations need housing at their own facility and need to realize that information needs to relate with all of their other information, otherwise it is worthless."* Even though they do not strongly believe in the concept, Oxcyon does also offer a cloudbased solution, but Sam Keller would never recommend this to the clients. However, it is necessary to remain competitive. *"If people are eating hot dogs, you have to serve them hot dogs, even though they are not good for the people."*

A remarkable comment made by Sam Keller is *"pricing in closed source is often ridiculous"*. He does not believe companies should charge their clients based on number of users, number of servers, per disk usage and operate it as a metered tool. At Oxcyon, they do not charge on usage, but charge based on the size and structure of the portal. Sam Keller again uses a metaphor here: *"Imagine you bought an Audi for \$ 30.000, but if you drove it 300.000 miles you would have to pay an additional \$90 000. This is ridiculous. They should not be charging based on usage. It is either the car works, or it doesn't work."*





Sitecore⁵

Customer Segments

Sitecore uses a partner network to target the end users. As far as end users are concerned, they aim for **mid-market and enterprise-market**. They have a high-end product and have become a **leader** in the market.

To differentiate further in the market segment, Sitecore offers room for scalability, by making available some basic configuration packages. There is the **primary** packet, the **professional** version and the **enterprise version**. Transitions from a smaller to a bigger package are possible and occur often. Sitecore also segments based on the operating system platform. As they have a very close partnership with **Microsoft**, they only make their product available in Microsoft environments. This means that the platform can only be installed on Microsoft servers. Mac or Linux users can still access the interfaces via a browser.

⁵ All citations made in this analysis are from the interview with Eddy Lalou (appendix 11), February 2nd, 2012.
Value Proposition

"Sitecore offers the added value of a total out-of-the-box solution." Sitecore offers a high-end product that is **feature-rich and powerful**. Features including multi-channel efficiency, email campaign manager, marketing automation and web analytics are available to the end users. The product is not only international, multi-lingual and bug-free, but also highly rated regarding aspects such as usability, functionality. Furthermore Sitecore identifies the marketing channels that are most effective for their clients. All this comes as a **tightly integrated** and strong product.

The product offered is of **modular** nature. The CMS comes as a core-product that is customizable and can be personalized to customer needs, as a large range of different modules is available. The modularity of the product is also represented by the different ways it is being used in. Both an intranet as an e-commerce site belong to the list of possible applications.

An enormous asset to Sitecore is its complete integration in the **Microsoft** environment. Their alliance with Microsoft proves to be very valuable. Also integration with Microsoft's SharePoint is being delivered to the customer.

Although Sitecore's core-competence is the delivery of an **out-of-the-box**, quality product, they do offer support and maintenance contracts, which include software updates and such. The Sitecore team is available **24/7** and offers support world-wide. To summarize the value proposition with Eddy Lalou's words: *"For us, product quality is the most important, an all-time sustainable functioning platform in all circumstances"*

<u>Channels</u>

Getting the product to the end-customer is mostly done via Sitecore's **partner network**. Sitecore does make contact with the end-user, but follow-up usually happens via the partners. The selling of the Sitecore CMS usually happens in two phases. There is the phase of buying licenses and software. In this phase there is a direct contact with the customer. In the second phase there is the developing of the website itself, the configuration and deployment. This happens through the partners. *"We're a product owner, responsible for the product, while our partners are responsible for the projects."* The partners are trained to do this in an efficient and professional way.

Eddy Lalou did mention however that Sitecore often has to deal with major clients directly. These enterprise-level clients ask for a **direct relation** with the software vendor and often with the major developers.

Customer Relationships

Same as for the channel strategy, customer relationships happen in both direct and indirect ways. Indirectly, information is being gathered via the **partners**, who are sometimes asked to write cases about clients. Sitecore is also checking up on clients leaving partners. The reasons for this usually vary from clients being unhappy, to clients switching from a technical to a marketing partner to clients just looking to get a new partner to get fresh ideas after several years.

Furthermore, direct contact with the end users is being done by yearly surveys. A back office survey organization is calling customers globally or if necessary regionally.

Key Resources

As Sitecore has worldwide support centers they need people and infrastructure all over the world. This means there is a need for physical resources, concerning **buildings and communication facilities**. Furthermore, there is a need for human resources, such as **good technical workers** and product **specialists**, who need the right skills and knowledge for the development of the software and the support. At last, the protection of their software, the contracts with clients & developers and the partnership with Microsoft on-site in Redmond can be seen as **intellectual** property.

Key Activities

"Our core activity is the product. We are a sales and marketing company for CMS software, we are not a service provider." This illustrates the vision of Eddy Lalou. They believe to be **product responsible**; developing the product is the most important activity. The elaborate **R&D department** is the biggest proof if this. Furthermore, they provide a platform to the developers community, as their accredited developers are important partners for the company. Their other partners, specifically their marketing partners, are presented with a partner community platform where documentation on the products can be found. Providing these **community platforms** can be seen as an important activity as well. Finally, also product communication and **marketing** directly towards the market are key activities for Sitecore.

Key Partnerships

For Sitecore, relationships with different partners are very important. There are the partnerships with **marketing companies**, and partnerships with **technical companies**.

The end-customer is free to choose which partner they want to collaborate with dependent on the marketing or technical needs.

Furthermore, Sitecore has several **deployment** and **consultancy** service providers as partners. Some of them are preferred partners, which are chosen by Sitecore, while others approach Sitecore themselves. It is important for Sitecore to support and train their partners.

Cost Structure

The biggest costs for Sitecore concern **personnel** and **R&D costs**. Employing 200 people purely in research and development certainly is a huge cost. These people are responsible for the evolution, development and support of the product's future.

Revenue Streams

The biggest revenue stream for Sitecore is **new sales**, which means the selling of new licenses for their software. Next to this, another revenue stream is the **maintenance and support contracts** Sitecore has with its customers. This entails twenty percent of the initial license fee. As the number of license sales is growing, revenue from new sales and maintenance contracts will keep increasing in the near future. Pricing of the software is based on the **scalability** of the licenses. Dependent on the number of websites, the number of content and the number of databases necessary, one or more licenses are required. Taking those differences into account, a distinction between three configurations can be made: basic configuration primary, professional configuration and enterprise configuration. Furthermore there are some **extra modules** which are not inside the out-of-the-box solution, but which can be bought on-demand. This will also influence the pricing of the software system. Although it should be noticed that Sitecore does not generate revenues from the selling of modules, which are developed externally.

Other important remarks from the interviews

An evolution noticed by Eddy Lalou is that content management systems are more and more becoming a commodity. For this reason Sitecore has looked at going into other software markets such as marketing automation, e-commerce, web analytics and social media. These software solutions have already been integrated into one strong framework. A trending evolution right now is the offering of cloud-based solutions. But Eddy Lalou comments that *"The market will always have new expectations. What is new right now, will become a commodity in the future."*

Eddy Lalou thinks that open source is too often being touted as free and evolving faster than closed source. He says this is a false economy as open source most often does not have the necessary roadmaps available to provide a well-directed functional evolution of the product. At Sitecore, roadmaps foresee both the functional and technical development of the product for the coming 18 months, which are being reviewed every 3 months. Partners are given access to the roadmaps to ensure a fine-tuned sustainable relationship now as well as for the future. If necessary, the company adjusts its strategy or business model to be able to adapt to future customer needs.



Figure r: Business model Sitecore

2.2. Open source companies

AppState/phpWebsite6

Customer Segments

Appalachian State University does not target a specific segment with phpWebsite. They developed it in the first place to use **themselves**. They use it at the university as both a CMS and a web application development platform. They decided to make it open source, so everyone that wants to make use of it has the possibility to do so.

⁶ All citations made in this analysis are from the interview with Brian W. Brown (appendix 4), April 6th, 2012.

Value Proposition

phpWebsite is not only being used as a **CMS.** Internally at Appalachian, it is also being used as a platform to develop **web applications**. They have developed applications such as club management software and a housing system. They are currently considering open sourcing these applications as well.

The fact that phpWebsite is open source improves the **stability** and the **security** of the software. The great thing about open source *is "getting a lot of eyeballs on the source code"*. The few times that they had security-issues, the community has in this way been a wonderful resource at pointing out the flaw and helping Appalachian fix it.

<u>Channels</u>

As Appalachian State University is developing the phpWebsite project with **public and student funds**, they have to be extremely sensitive in using these. They cannot in any way use these funds to promote phpWebsite. Therefore, they do not spend a lot of time to marketing. *"It's more like, 'here it is, if you find it useful, that's great."* They can only develop it and give it away to the public at large, because that is a productive thing to do.

The **community** can be seen as a channel, as it raises the awareness of the product through word-of-mouth. It is also a way to get in touch with the end users.

There are some companies who are making a **commercial business** out of phpWebsite.

Customer Relationships

As there is no commercial purpose to phpWebsite, it does not have a real customer relationship management program set up. However, there are possibilities to contact the developers of the phpWebsite project to ask for some support. For example there are **forms** created that make it possible to ask questions. Next to this the developers are available in the **IRC channel** every day. They do try to answer the questions, but Brian Brown stresses they "*do not actually provide external support*".

Key Resources

The main key resources for phpWebsite are the development skills of the **people** within AppState. As it concerns a university, and the purpose of phpWebsite is mainly focused on education and own use, the development team and the IT students are the most important resources. The people from the **community** can also be seen as valuable because they are an active player in the process of solving any problems with the software.

Regarding physical resources, it is clear that the organization is facilitated on the university premises, so there is no real need on other infrastructure.

The software of AppState is protected by the license used for the source code (GPL). The first objective of the program is primarily the own use, so they do not spend a lot of attention to these intellectual resources.

Key Activities

The key activities for phpWebsite are particularly aimed at the development of the CMS and the web-applications that are complementing phpWebsite. Next to the **development** itself, it is important to **monitor** questions, suggestions and bug reports. This way a lot of problems can be identified and also needs for the future can be mapped. This information will be used in the **roadmap**, which is largely driven by the university's teams and foresees the coming evolutions of the software.

<u>Key Partners</u>

The key partnerships of AppState can be found in the cooperation with **external developers**. They are mostly found in the community, but also other universities cooperate in this. These people regularly come up with new ideas which are ending up improving the software. For AppState, this works at a very good cost/benefit ratio.

Cost Structure

As phpWebsite is a project organized by a public university, there is no profit-making business organized for it. As everything at the university is happening through public money, they do not mean to seek profit from it. They try to **minimize costs** as much as possible as a lot of the activities are done in an educational setting.

Revenue Streams

AppState does not generate any revenues from phpWebsite. The project is financed with **public money**, so that is why they do not have a commercial business organized. By giving it away for free they believe they have given something back to society.

Other important remarks from the interview

The phpWebsite is a project that has been running for over 12 years now. Back then, there was not a broad selection of CMSs available. The decision to build their own CMS was therefore driven by the need of having a platform they could control; primarily to assist them in web application development.

Brian Brown notices two trends in the CMS world. First, there is the upcoming trend of putting things into the cloud and SaaS-based solutions. Brian Brown believes *"content management systems are a kind of cloud services by their very natures"*. He thinks that SaaS and CMS are in a certain way destined for each other and really complement one another. Second, there is the growing need to offer better mobile computing support. Due to the exploding number of mobile devices, this is an evolution that cannot be ignored.

Brian Brown repeatedly mentioned that the sensitivity of the source of funding has been a constraint for the evolution of the project. He doesn't think that it has been the best thing for the project, but in terms of education and own-use purposes it has certainly proven its worth.



Figure s: Business model AppState/phpWebsite

Concrete57

Customer Segments

Concrete5's slogan *"Made for marketing, but built for geeks."* actually summarizes the segments they are targeting. They try to find the balance between a technical CMS appealing to **IT people** and a user-friendly CMS attractive to **marketing managers**.

⁷ All citations made in this analysis are from the interview with Franz Maruna (appendix 5), May 1th, 2012.

Concrete5 does not segment in the businesses they are targeting. *"You can build anything with concrete5"*. Franz Maruna informed us that their client base ranges from **small churches** and non-profit organizations, who considered the product mainly because it is free, to **big well-known companies** spending six figures on large web presences.

Value Proposition

Concrete5 offers a very **user-friendly** CMS for non-technical people. *"They can actually edit stuff without having to call the geeks."* Their in-context editing, is a feature that really sets Concrete5 apart. People do not have to go into the back-end of the software to edit stuff. It is sufficient to just put the page in "edit-mode".

To the engineer, Concrete5 is offering a CMS that is **modularly controllable**, **object-oriented** and **very well architected**.

As Concrete5 has a background as an interactive media firm, they are really trying to architect Concrete5 as *"a toolbox for building design-centric websites."* They are offering a CMS that is not only easy-to-use, but also very good looking.

Another important part of Concrete5's value offering is their **Marketplace**. It actually is similar to Apple's Appstore, in which both Concrete5 and third-party developers can offer their apps. Again, the Marketplace is trying to be very user-friendly. For example, add-ons can be installed through a one-click install process. This makes it very convenient for non-technical people.

To enterprises, Concrete5 is offering a number of extra features. For example they offer **services and hosting**.

<u>Channels</u>

"Honestly, we do not spend a tremendous amount of money on marketing." Franz Maruna believes it is very difficult to see a meaningful return on investment on marketing efforts. What they try to do instead, is making sure their site has clear messaging.

"An enormous advantage of being open source is that you have that huge group of advocates that want to go out and sell for you." Concrete5 really tries to enable those **(value-added) resellers**, and gives them all they need. Concrete5 has made some attractive PDFs, supplies the resellers with stickers if they want them... That in fact, is their marketing cost.

Next to this indirect way of getting the product to the customer, Concrete5 also offers their product **directly** to the customer. Also via the community, there is contact with the end

customer. Furthermore, the **community** raises awareness through word-of-mouth so it can be looked at as a channel.

Customer Relationships

Concrete5 uses **WHMCS** to manage the relationships with their clients. It's not only a client management package, but also a billing & support solution for online businesses. What they further do, is help third-party developers, who are offering add-ons via the Marketplace, to offer support on the add-ons. In the Marketplace you can ask support for every different add-on via a single click. Concrete5 then either helps the customers themselves, or sends them to the right developer.

Another important tool in the interaction with clients is of course the **community**. *"The community helps one another."* It is an important mechanism for Concrete5, as it simplifies keeping clients happy.

Key Resources

As for most software development companies, the most important resource for Concrete5 is **human capital**. It is important for an open source company such as Concrete5 to have talented people within the small core team, including an important community manager, as having intelligent, creative and skilled people in the community. The quality of development coming from crowdsourcing depends on the competences of all the people programming code for Concrete5.

Facilities to offer their hosting services, primarily **server infrastructure**, can be seen as the required physical resource.

Regarding intellectual resources, the license used for Concrete5 concerns the MIT-license. This license does not provide a significant amount of protection for the product. Although, they believe the strong **brand** name is something other companies, offering the same product as Concrete5, do not have.

Key Activities

"It is better to focus to make your product better." This illustrates the vision of Franz Maruna on the key activities of Concrete5. The **development** of the product is key for the viability of the company. Core development is severely **monitored** and together with quality input from the community, the contribution of the core team of Concrete5 is very important. Also development in the field of themes and add-ons can be seen as significant. They can add these themselves, but they also offer the possibility of crowdsourcing.

Next to assuring the development on Concrete5, another important activity is the support they supply to the community. **Managing** everyone and everything around the Concrete5 **community**, is an important task of the community manager and the rest of the core team. A major instrument for this is the **Marketplace**, as this is the place where everyone comes together. Furthermore, this is an important source of income for Concrete5. Therefore, making sure the Marketplace runs well, containing quality software, which is controlled by the peer review board, is an important responsibility for Concrete5.

Key Partners

"We are looking more to **system integrators** and **value-added resellers**, than to a form of distributor reseller model." Franz Maruna primarily focusses on partnerships with companies which resell the Concrete5 product with additional value-added services and/or products. These companies will offer the Concrete5 product to the end customer. That is why it is important that Concrete5 foresees these partners with knowledge and education about the product so these companies understand the value of the software. That way they can keep fulfilling the end users' demands.

External developers can be seen as an important partner as well, because this cooperation has a significant influence on the evolution of the product.

Cost Structure

The major cost for Concrete5 definitely is **payroll**. The contracted people working for the company take a big chunk out of the budget. The second largest cost is the **infrastructure** for offering their hosted service, specifically servers etc.

Revenue Streams

About 50 till 60 percent of the generated revenues come from the M**arketplace** where they have a twenty five percent cut on all sales of add-ons and themes. The remaining 40 till 50 percent comes from **hosting** and other **services**.

Other important remarks from the interview

"Running the community part is like a whole additional thing you have to do." Franz Maruna thinks that this part can be done in several ways. He mentions examples (e.g. Joomla) where control is rather tight, and examples where there is not much control. *"Crowdsourcing is not the answer to everything."* This is what Maruna believes in, and this is reflected in the way they manage the community. Still having that small team involved, often leads to success stories.

Concrete5 has grown out of a full service media shop. They used to be a dozen developers and had some small projects going on. At that point it was just ConcreteCMS, and this was a closed source solution. As many of their clients were start-ups that didn't get through the second or third round of funding, they decided to work on a brand-new version of the CMS. They believed it looked awesome, and decided to make it open source. They moved out of their expensive offices, staffed down to just 3 people and then things actually started taking off. Due to popularity, over the years, they have staffed up again.

Concrete5 does not offer a specific enterprise-version of their CMS, although they do offer specific enterprise-services. What they want to avoid is having two different products, which often implies that one product is getting more focus than the other one. The customers of the open source one might start to think that they are getting the crappy version of the product. Concrete5 wants to avoid that image of selling so-called 'crippleware'.

An interesting point Franz Maruna made was: "A disadvantage of open source is that it involves politics." He explains this by referring to the "volunteer thing". Franz Maruna thinks it's really cool that "someone from New Zealand or so, is spending two days building some feature for Concrete5". But if this feature turns out to be crap, he has a problem. "If they are volunteering their time, the time you have to spend to make them feel good, while at the same time you are not happy with what they have done. That is just a political nightmare." Keeping those external developers happy and motivated, thus is an important task and is not easy to fulfill.

"I think that the smaller guy sees open source and links it to free. The bigger guy sees open source and links it to 'I have independence'." This can be linked to the fact that open source avoids a vendor lock-in and gives more freedom to the customer. "We are working with clients who would rather own what they are paying for." Furthermore, Maruna presents himself as an opponent of the GPL license. "I think the GPL is really problematic as it steers a lot of people away from open source, because no one really knows what the GPL is going to do with that whole redistribution clause."

Evolutions noticed by Franz Maruna are the growing number of open source companies offering a commercial product. Also prices for commercial stuff are decreasing, as everyone always has to rebuild their website every 2-3 years. That way it is pointless to ask for a 5 or 6 figure license fee. As a last evolution, he noticed that many large companies are choosing for open source solutions.

Key Partners System integrators	Key Activities Development	Value Proposition User friend	ly ly	Customer Relationships WHCMS	\heartsuit	Customer Segments IT people
Value-added resellers	Managing the community	Modularly controllabl oriented ar	e, object- nd very-well	community		Marketing managers Both very small and
External developers	Organizing the marketplace	architected Design-cen	tric			large well-known companies
	Key Resources	Marketpla	ce	Channels	P	
	Human capital 🎽	Services an	d hosting	Value-added resellers		
	Server infrastructure			Directly		
	Strong brand name			Community		
Cost Structure Payroll		(IIII)	Revenue Streams Marketplac	e sales (50-60%)	(J
External developers			Hosting and	d services (40-50)%)	



Fork CMS⁸

Customer Segments

Fork CMS embraces the vision that a content management system should not be ITfocused. They believe content should be posted by a **marketing and/or communication manager** and therefore see a CMS as a marketing and communication tool. The targeted segment thus is the marketing and communication market.

Value Proposition

As Fork aims at the **marketing and communication** segment, the key features of Fork CMS are related to this. In developing Fork CMS, much attention has been paid to key differentiation points such as **usability**, **traceability**, and **SEO** (search engine optimization).

Another key reason why clients go with Fork CMS is the avoidance of a **vendor lock-in**. This is a true benefit for all open source systems and is often a key reason for web development companies to go with open source.

⁸ All citations made in this analysis are from the interview with Bart De Waele (appendix 6), February 10th, 2012.

<u>Channels</u>

Fork CMS is being delivered to the end customer via **web development companies**. Bart De Waele is CEO of such a web development company, called 'Wijs'. At the moment of the interview, 15 companies in Belgium were already using Fork CMS as their primary product.

Fork CMS is trying to make a name in the CMS world, mostly via social ways of marketing (Facebook, Twitter...). The main groups they are targeting are **developers** and **students**. Furthermore, word-of-mouth is important as **community** members are talking about Fork CMS and promoting it this way.

Customer Relationships

Via the **Github** site, improvements to the code can be made and suggested by the community. As Fork CMS is a very recent project, the **community** itself actually takes care of the customer relationship management. As there are many enthusiasts of the project, the community is very active and people are helping each other. The core team at Fork CMS is just supporting the existence of the community.

Key Resources

The core team of Fork CMS consists of usability **developers** and **design specialists**. These people can be seen as important human resources as they decide on the future evolution of the system. Furthermore, having skilled developers within the community can also be seen as human resources.

Key Activities

Key activities of Fork CMS mainly focus on decisions on the **evolution** of the software. The core team sets the directions in which development will be guided. They decide whether improvements sent from the community are **approved** to be added to Fork CMS. Furthermore, when there are 'Fork CMS'-communities emerging all over the world, e.g. China, the core team tries to **support** these and aims for people from these communities to **engage** them within the core team.

<u>Key Partners</u>

The key partnerships for Fork CMS can be seen as the cooperation with **web development companies** who are using Fork CMS for their business. These companies are building modules and applications for the system, which they, over time, release back

to the whole Fork CMS community. This way the features of Fork CMS enlarge, which benefits to the viability of the system.

Cost Structure

Currently, costs are **negligible** for Fork CMS, as mostly all of the development comes from the community. In the long run, when there should be a minimal revenue stream, they plan to employ some people to work on the system full time.

Revenue Streams

As Fork CMS is a rather young project, revenue streams are still **absent**. For the future they plan to set up an appstore, where people can buy applications for their Fork CMS. These applications would cost an amount of money and a percentage of it would go to the core team, which would regulate this appstore. These revenue streams should ensure that Fork CMS can support itself in the future. Nevertheless, they do not have plans to make profit out of it, a break-even operation would suffice.

Other important remarks from the interview

As Fork CMS is a very recent CMS, it was interesting for us to see what the early stages of developing a community-based software system are like. Bart De Waele commented that *"3 to 5 years ago, almost every web development company in Belgium was developing its own CMS."* Nowadays, more and more companies are switching to Drupal, although at Fork CMS they chose an opposite direction. Open sourcing their CMS was a decision largely based on their belief in the concept and power of open source. Everything they were doing at Wijs was open source and so this simplified their decision. Next, the commercial pressure to go open source was becoming greater as clients wanted to avoid a vendor lock-in. A last reason was the fact that the cost of solely deploying a proprietary system for a business can be very high.

Across the different web development companies, developers are working on Fork CMS. These companies use a percentage-based time allocation structure, which allows their developers to work specifically on the development of Fork CMS. This usually is being represented in full-time equivalents. Both web development companies and Fork CMS benefit from this, as web development companies using Fork CMS cannot afford being left behind on the evolutionary path, so the future of the system is being assured.

A trend being noticed by Bart De Waele is the growing number of companies offering a SaaS-based solution of an open source CMS. Companies are offering a hosted version and extra support on open source solutions. This might also be a possibility for Fork CMS in the future. Other trends being spotted are the integration of analytics, email and social marketing.



Figure u: Business model Fork CMS

Plone9

Customer Segments

Plone is an enterprise-ready CMS aiming for **small- and medium-sized enterprises**. *"Although it is done, using Plone for a simple and small personal website is kind of overkill."*

web development companies might segment in the different markets they are targeting, but Plone itself is not targeting any specific markets.

Value Proposition

Regarding the value proposition of Plone, a distinction can be made between the value delivered to the end-customer and the one reserved for developers.

Towards the end-customers, Plone is an **enterprise-ready** CMS, promoting its key features, **security** and **stability**. Plone already has a set of customers that prove this point.

⁹ All citations made in this analysis are from the interview with Sjoerd van Elferen and Rob Gietema (appendix 10), February 27th, 2012.

Organizations such as NASA and FBI would not use Plone if this were not the case. Also the **usability** and the possibility to set up a well-structured **authentication** and **authorization** system is a very valuable asset.

To the developers, the main features of Plone are its possibilities to use it in different ways. *"The idea of Plone is to be as lean and mean as possible. The core of Plone has to become as compact as possible. Afterwards, a lot of features can be installed to that."*

An interesting point that came forward through the interview is that using open source avoids a **vendor lock-in**. This can be seen as a positive point for both the developers and for the end-users. Open source always gives the end-user the freedom to choose between different providers of the software. When the relationship with the current supplier is troubled, they can go to another one without having to change their current platform. The advantage for the development companies, is that they can use this idea to convince new clients, and keep their personal motivation for delivering a quality product.

<u>Channels</u>

The raising of **awareness and promotion** of Plone is organized directly by the Plone foundation. The marketing team takes care of this by, for example, being present at trade-shows or conferences.

The deliverance to the end-customers is mostly being done by the **web development partners**. Although there is a possibility to download Plone right away from the site, a certain complexity and skillset is necessary to get the most out of Plone.

Via the **community**, members inform their peers about the product and its features. This way, awareness is being raised and so the community can be seen as a channel to attract customers

Customer Relationships

Most of the information coming back to Plone is happening via the **web development companies**. They are passing on the feedback they get from their clients to the community.

There is also a possibility to direct feedback. An **issue tracker** is available on the main site of Plone, were both end-users and developers can enter the problems they encounter when using Plone. *"This issue tracker can be seen as the ideal situation, because there is one central point where all the issues are being collected."*

Key Resources

First, there are the intellectual resources in Plone. These are the existing **trademarks and licenses**, which are protecting the product. *"Thanks to the fact that we had an elaborate look at the available licenses in a very early stadium of the project, we almost never have issues related to that."* Furthermore, there are the **contribute agreements**, which every developer has to sign and results in a transfer of rights concerning written code to the Plone foundation.

Second, there are the human resources in Plone. These are all discussed in the key activities section and involve the different skills and knowledge of the **people** who are active in either the foundation or the community.

Key Activities

To analyze the key activities of Plone, it is important to make their structure clear. First, there is the Plone foundation. The Plone foundation exists of members who have proven themselves towards Plone as they have had an active contribution towards its evolution. Each year, these foundation members decide on the board, which is elected from candidates found in the community. Next, the foundation also entails the marketing team. Second, there is the community. This consists of different teams based on the activity they do. For instance, there is a GUI-team, a framework team, a documentation team and a roadmap team. These two parts, the foundation and community cannot function separately, but influence each other in both ways.

The key activities for the board are not development-related. Their first key task is to **support the community** as much as possible. They do this by offering the platform and by protecting the product, concerning licenses, trademarks and copyrights. They are responsible for the **legal** side of Plone as well, as they will deploy attorneys in case of any legal issues. The marketing team is responsible for the **promotion** of the product.

The key activity for the community is **development** of the product. Both core and noncore development is done by the community. Core-development is **reviewed** by the framework-team and the release manager. *"The release manager is a bit of an odd one, as he is paid by the foundation, but the foundation doesn't give him directions."* Non-core development is also being reviewed by peers in the community. Sometimes there is a socalled 'sprint' or 'Plone tune-up' being organized. This is a period of heavily focused development where in a short time a lot of bugs are being fixed or a certain aspect of Plone is being improved. **Conferences** are an important part of the way Plone works. Each year the Plone conference brings people together. Both foundation members, who have long been active in the community and more recent community members are present and share their opinions on the future evolution of Plone. Training, vision sharing, development, are all phases in this conference.

<u>Key Partners</u>

Plone's main partners can be found in the **community**. An important part of the community consists of people active in **web development companies**. They do two separate things for the community. On the one hand, they deliver the product to the end customer, often in some kind of an implementation project, accompanied by a service level agreement. On the other hand, they are major developers of the Plone project as they benefit themselves from Plone being improved. Developers can be freelancers as well, who are not contracted by a specific web development company.

Cost Structure

The costs of development in Plone are almost negligible, as they are counting on the free efforts of intrinsically motivated community members to take on the different necessary tasks. The only cost regarding development is the payment of the **release-manager**.

Then there is the cost of **underwriting** for the conferences, as the revenues are only made at the conference itself. Furthermore, **marketing activities** and sometimes **transportation costs** for major developers visiting the conferences or sprints take a chunk out of the budget of Plone.

Revenue Streams

The foundation receives money for the entrance at **conferences**. Sometimes they also receive money from **sponsoring**. So although the foundation does have money, it should not be looked at as an objective. Achieving a break-even situation is more than enough for Plone.

Other important remarks from the interview

An evolution being noticed by Sjoerd van Elferen is that social media has been integrated a lot the last few years. At this moment, adding geo-information is being spotted as a trend and for the future Rob Gietema listed real-time collaboration as an important feature. Most of the time, modules for features are added in the following way: *"It starts with new functionality. Then different people have different visions towards details of the development*

of the module. If there are many common functionalities, they will try to connect and develop the module together."

"You've got to be quite a large proprietary company to compensate all the developing hours being done by the developers of a community like Plone's." Due to intrinsic motivation in open source projects, there is no need for compensations, which is a huge difference, compared to closed source companies. The interviewees were developers themselves, and they told us that intrinsic motivation often derived from the number of downloads for a developed module. "If you see that the download-counter is over 1000, you know that is more than 1000 sites, not 1000 users, which are running your module."

A remarkable fact about Plone is the existence of the yearly elections for a new board. It seems like this is a bit of a downside towards creating a general direction in the evolution of the project. However, they do have a roadmap team in place, which may compensate for this problem and ensures the technical evolution for the next couple of years.

Plone works more democratically than other open source projects where a small group of people is in charge. This is also represented by the yearly elections of the board. Even one of the original founders of the project, is now no longer part of the board.

The interviewees believed that Plone does not have the biggest community (compared to for example Drupal or WordPress), but certainly a very active one. Also the well-defined structure (with the existence of the Plone-foundation) of Plone can help to the success. "If all goes well with an open source project, it will continue to go well. If it goes wrong, a kind of institution like in our case 'the foundation' can be helpful to get back on track." This illustrates the importance of the foundation in the success of the Plone-project.

The fact that it is free is often the least positive aspect of open source. Much more the aspect of security thanks to peer-review by an active community can be seen as the most important aspect. *"Being closed-source doesn't mean that the software is more secure. On the contrary, having more people looking at the source code (in open source) is just that extra benefit."*

Many web development companies participating in an open source project apply some kind of percentage rule regarding development in the project. This means that a percentage of the time (usually referred to in full-time equivalents) is dedicated to open source development.



Figure v: Business model Plone

Umbraco¹⁰

Customer Segments

Customer segments for Umbraco can be divided into end-customers and developers.

For the developers, Umbraco aims for **.NET** specialists, as their CMS is based on this platform.

As far as the end users are concerned, they do not target a specific segment. To quote Tim Geyssens: "It can go from somebody setting up a small website as a hobby to intranets for big news gathering companies."

Value Proposition

The value proposition can be looked at in two ways. First there is the value proposition towards the end user of Umbraco. Second, and maybe more importantly, there is the value proposition towards the developers.

For the end user, the main value offered is **simplicity and usability**. Everything is being made as easy-to-use as possible. Umbraco's CMS is constructed in a way that it can be

¹⁰ All citations made in this analysis are from the interview with Tim Geyssens (appendix 12), February 23th, 2012.

applied in many ways. It can be used in the range from individuals running their own small website to intranets for giant news websites. In the future, the product will be available as a service in the cloud as a **SaaS** solution. This makes both set-up and maintenance processes easier for end users.

Concerning the developers, Umbraco also has a unique value proposition. Their main feature is that their product offers a lot of **flexibility** to a developer. Umbraco is known in the CMS-world as a system that gives a lot of freedom to a developer without 'pushing him/her into a corner'. The use of the **MIT license** extends the idea of freedom when taking into account licensing. This also offers the developers the opportunity to use Umbraco's source code in other software applications.

A final important part of Umbraco's value proposition is the possibility to engage in a **support contract**. Every domain name using Umbraco has the possibility to file for a support contract. This happens via the partners, who cannot use a single support contract to help out different clients.

<u>Channels</u>

The most important channel undoubtedly is the **partner network**. The product is being delivered to end customers through web development companies. As far as delivery to these web development companies is concerned, Umbraco uses a direct strategy. Support contracts are also being delivered via the partners.

"The community is self-organizing, it grows gradually." So through word-of-mouth, users will inform each other about the availability and features of the product. This way, the **community** can be looked at as a channel as well.

Customer Relationships

Although Tim Geyssens told us that *"there is no direct contact with the end-customer"*, and all contact goes through the **partner network**, he also mentioned that *"you actually get feedback from the community"*. That way, they are in some way interacting with the end customer, as both Umbraco HQ members and end customers are active in the **community**. The community can therefore be looked at as a kind of customer relationship tool.

Key Resources

First of all, Umbraco needs some **physical resources**. They have their own site, and a community site, which are running on a server.

Second, they rely on human resources which can be found in the community, as they don't have an elaborate development team in their structure. There is however, a small group of developers, called **the HQ**. This is a group of 12 people who are full-time employees of Umbraco and ensure the future of the company. They all have their own specific function within the company. For example Tim Geyssens was responsible for a commercial form builder, which is a tool for building forms without having to dispose of code-knowledge. Active community members are approached by these HQ-members to join them. This way, Umbraco can be ensured that a member of the HQ possesses the necessary skills, knowledge and motivation.

Very specific in open source companies is their licensing structure. This can be seen as a form of intellectual resources, as it protects their core product. Umbraco uses the **MIT-license**, which offers a lot of freedom to developers. License infringements are therefore rare, and Umbraco doesn't make a problem of people 'rebranding' their product, as they believe their own name is already very well known. The MIT-license just protects them from any liabilities, claims, damages, etc.

At last, the **generated revenues** can be seen as a financial resource for future development.

Key Activities

Most key activities of Umbraco are related to the **community**. On the one hand, this community is self-organizing, but Umbraco has to offer an interactive platform to make this possible. Although being self-organizing, a community sometimes needs steering. For example, when several different developers are working on incorporating the same function, Umbraco brings them together.

Product development is of course a very important activity for any software solution provider. Umbraco keeps some development in-house, but mostly relies on the input of the developers, active in the community.

A last key activity is training. Umbraco offers a lot of **training** to its customers. They give training session themselves, but also outsource to external trainers. Furthermore, there are subscriptions available to video-tutorials.

<u>Key Partners</u>

The main partners for Umbraco are **web development companies**. Umbraco's product is delivered to the end customer via these types of companies. This is a buyer-supplier kind

of relationship and is key to Umbraco's business model. They do not have direct contact with the end customer.

Next to these web development companies, Umbraco sometimes agrees to a strategic alliance with other companies in the ICT industry. For example, **Microsoft** has sponsored Umbraco in the past via some Google-ad campaigns. In return they can request to add some features in Umbraco.

Umbraco plans to release their product as a service in the cloud. For this SaaS-application they will work closely together with Microsoft. **Hosting** will be taken care of by Microsoft.

As Umbraco is community-based, development is often done by external developers. All these developers in the **community** can be seen as partners. Acquiring these free resources is of vital importance to Umbraco.

Cost Structure

The major costs are pretty straightforward for Umbraco. Their main costs are the **servers** for running their sites and the **remuneration** of the HQ-members for core-development and other activities.

Revenue Streams

"75 percent of our revenue comes from the commercial packages". This in fact means that Umbraco takes a percentage of revenue generated by a third-party selling a **commercial add-on** via their site. Next to this, they obviously create commercial packages themselves as well. Umbraco further has revenue from their **training** methods. They offer direct training, via external trainers (they get a percentage on this) and video-tutorials online.

Furthermore there are custom **support contracts** available, which are valid for three to six months. Their pricing is based on the amount of time they have to spend on a project.

Other important remarks from the interview

Tim Geyssens often stressed the fact that *"Umbraco has grown organically"*. Umbraco started as a one-person project in Denmark. Someone engineered the project for himself and his clients. Gradually, more and more people became involved in the development of Umbraco. Currently they are 12 people in Umbraco HQ, while original plans included a maximum of 9 people. For the future they don't aim to expand a lot, but try to take on more work with the same amount of people.

An evolution noticed by Umbraco is that confidence in open source has been growing a lot. Tim Geyssens believes it may also have to do with the current financial crisis. Maybe companies try to reduce their costs by avoiding license costs. A fact they have noticed in Denmark, is companies using Sitecore, are making a switch to Umbraco. They provide a viable alternative for Sitecore, as they are both using the .NET framework.

Key Partners Web development companies Microsoft Hosting partner Community	Key ActivitiesImage: Constraint of the second of the seco	Value Proposition Simplicity & SaaS Flexibility Freedom fr license Support co	with a constraint of the const	Customer Relationships Via partner network Active community Channels Partner network Community	Customer Segments .NET developers No specific end user segment
Cost Structure Remuneration HQ-mem Servers	bers	A A A A A A A A A A A A A A A A A A A	Revenue Streams Commercio Training m Support co	al add-ons ethods ntracts	Circle Ci

Figure w: Business model Umbraco

3. Research conclusion

3.1. Closed source business model

Customer Segments

A commonality found between the different closed source companies is that they use **different licenses**. These different licenses are usually in place to be able to target smallas well as medium-sized or large companies. The segmenting in the kind of industry is different however for all companies.

Value Proposition

A key issue for every closed source company is the offering of **professional services**. For some companies, these are included in the licenses offered, while for others they are sold separately. The **different licenses** make it possible to offer enough value to both smaller and bigger companies. All closed source companies are offering a **hosted version** of their product as well. For some companies, this is indispensable (as their main product solely comes as a hosted version), while for others it is just to broaden their range of solutions or to remain competitive (e.g. Oxcyon).

Channels

Every researched closed source company has an extensive **partner network** set up. Next to this form of indirect sales, they are also offering their products **directly**. For bigger companies, this usually happens to a lesser extent. They only have direct relationships with their largest clients, as they are specifically demanding it.

Customer Relationships

There is a **big diversity** in the way that the researched companies are managing the relationships with their clients. Some companies (e.g. Anonymous X) deploy customer relationship management as a differentiator and have face-to-face meetings with their clients on a weekly basis, while others (e.g. Accrisoft) are more relying on the input from their partners. Many companies are also using a **client community**, in which clients can both help each other and be helped by staff.

Key Resources

For closed source companies the most important resource concerns **personnel**. Having quality developers with the necessary skills and knowledge are indispensable for such

type of company. In order to offer the promised support and services, closed source companies need **infrastructure** for hosting and communication (e.g. servers). In the field of intellectual resources, their software product is protected by **patents, trademarks and copyrights**.

Key Activities

The major key activity for a closed source company is indisputable **in-house** (research and) **development**. Furthermore, an important part of the activities concerns offering **training** and other **professional services** to the customers.

Key Partners

Essential for closed source companies are their partnerships with **sales partners**. These partners can simply be resellers of the software or they can be value-added resellers of the product. It is mainly through these partnerships the closed source companies offer their product to the end users. Another important partnership is the cooperation with **technology partners**. Both design and development partners contribute to the evolution of the product.

Cost Structure

For closed source companies the biggest cost concerns **payroll**. The people working on research and on the development of the product take the biggest chunk out of the budget. Further, the cost of **infrastructure** is also very high. Either in-house hosting facilities or costs for third-party hosting services are very expensive.

Revenue Streams

The most important revenue stream for closed source companies is mainly **license fees**. Some companies include **services and maintenance** in their license fee, while others consider revenue from support as a different stream.



Figure x: Business model Closed Source

3.2. Open source business model

Customer Segments

Usually the product can be for **anyone**. As the product is freely available to download from the Internet, most of the time it is used by different sorts of people. Sometimes, a company is aiming at a specific segment such as marketing or communication and develops tools specifically for that segment.

Value Proposition

An important value that can be offered to clients, which is in the nature of open source, is the fact that a **vendor lock-in** can be **avoided**. Open source offers a lot of freedom to the end customers as they can always switch between suppliers of the open source solution or even choose to do it themselves. At the same time, **freedom** is also assured to the developers, as they can modify source code anyway they like. Licenses can play a role in this and are restrictive in some cases. Although different open source solutions can have a different focus, such as a marketing-focus or an IT-focus, 4 out of 5 researched companies are mentioning **user-friendliness** as one of their key aspects.

Channels

Most of the researched companies were somehow working via a network of **value-added resellers**. Those are usually focusing on extra development or design and services. Another important channel in open source companies is the **community**. Word-of-mouth is often important to them and many community members will convince new people of a certain open source solution. This way, a larger community is being built and thus it can be seen as a channel.

Customer Relationships

The **community** in an open source company is usually an important customer relationship management tool. In the community, people can help each other out, or ask staff-members for support. This means that offering the platform that makes this community work is a key task for the open source companies. Often, feedback is also being collected via the **partner network**.

Key Resources

For the development of the product, open source companies need **skilled**, **creative**, **and talented people**. These can be active within the company itself, e.g. in the core team, or can be found in the community around the product. Concerning physical resources, a

server to support the community is almost the only requirement open source companies need. Intellectual resources in open source companies are special. Dependent on which **license** accompanies the software they can work restrictively or can be used to give freedom to the developers and users. For corporate open source companies, the **brand name** is very important as it is used as a protection against other suppliers who use the same software. This is not really the case for community open source companies as they tend to focus less on profit making.

Key Activities

Key activities for open source companies focus on two things. In the first place, they have to assure the evolution of the product by contribution of own **development**. This can be core development as well as non-core development. The second important activity for an open source company is **supporting the community**. They have to facilitate the community with the necessary platforms to give developers the opportunity to contribute to the software.

Key Partners

Open source companies mainly have partnerships with **web development companies**. These companies are often value-added resellers and offer the core product of the open source company accompanied by additional modifications and services to the end users. Besides, also **external developers** can be seen as a partnership for the open source companies. In general, these external developers are people from the community who make valuable contributions to the development of the software.

Cost Structure

Regarding cost structure for open source companies, a distinction has to be made between corporate open source and community open source companies. Corporate open source have contracted employees, mainly developers, so **payroll** can be seen as a big cost. For community open source companies costs are almost **negligible**.

Revenue Streams

Also for the revenue streams the distinction between corporate and community is important. For corporate open source companies, revenues are generated from **sales** (e.g. market place, add-ons) and from **hosting** and other (support) **services**. Regarding community open source companies, revenues are **limited** and are used to cover all expenses. These companies aim to operate break-even and do not have a profit-making target.

Key Eartners Contract Companies External development	Key Activities (Core)Development Supporting community	Value Proposition Avoidance of a vendor lock-in Freedom to end- customers and developers User-friendliness	Customer Relationships Partner network Community	Customer Segments Usually for anyone
	Key Resources Skilled & creative people Server License Brand name		Channels Calue-added resellers Community	
Cost Structure Payroll for corporate OS Negligible for community O	sc	Revenue Streams Sales, hosti Limited reve	ng & (support services) for co enues for community OS	rporate OS

Figure y: Business model Open Source

3.3. <u>Comparison</u>

Similarities in business models

Some similarities can be found in the products offered by open source and close source companies. For example, most interviewed companies used a network of sales partners for the distribution of their product (channel). Next to the partner network, there are some more commonalities to be found, primarily between closed source and corporate open source companies. Business model components that are often very similar are value proposition, cost structure and revenue streams. Community open source seems to have fewer things in common with closed source.

Differences in business models

The main differences between closed source and (to a larger extent) community open source companies can be found in the value proposition. While closed source companies always offers professional services, which are sometimes included in the license fee and sometimes sold separately, community open source software often lacks professional support. Corporate open source however, usually does offer a set of professional services. Furthermore closed source solutions are generally more tailor-made, whereas personalized customization in open source happens through value-added resellers. Although we did not make a separate business model for corporate OS and community OS, this is a difference that can be found. Further there is an important remark regarding financial aspects: corporate OS has both bigger incomes and larger costs than community OS. This is only logical, as corporate OS is trying to make a profit-making business out of open source.

Other important conclusions

A remarkable fact in open source is related to licensing. All of the studied corporate open source companies were using the BSD or MIT license. At first, that seemed like a peculiar point of view, as it allows reuse of the source code in a proprietary way. On the contrary, two out of three community open source companies were using GPL, which actually is more restrictive. When enquiring about this, the corporate OS companies replied that they believe they have already created a brand name that is strong enough.

"We believe it will be difficult for them to do a better job than we are doing. If we can't do a better job at it; then what are we doing here? We wrote it ourselves!" (Conversation with Mr. Maruna, 01/05/2012, Skype)

They point out they really believe in their brand name and in the service offered.

A next conclusion is about customer service in an open source company. OS almost automatically gives customers 'a voice' by the use of their communities. Customers do not only help each other in a community, but can also often be helped by staff members if necessary. This way an OS company is in a way in direct contact with their clients.

Many end customers still see two downsides in open source software: security (as source code is freely available) and professional services.

"Getting a lot of eyeballs on the source code is a really a positive thing" (Conversation with Mr. Brown, 13/04/2012, Skype)

Bugs or flaws in the source code are spotted faster and can be fixed rapidly if a structured feedback system is in place. The lack of professional services is a fact usually stressed by the closed source companies. However, there are a growing number of businesses being built around support for an open source product, often in the form of a value-added reseller. This trend is also recognized by some closed source companies, and is looked at as increased competition. It seems like a potentially very profitable business.

"A company that offers hosting and other services for an open source CMS; that is closer to what we [closed source company] do." (Conversation with Mr. Hodous, 20/04/2012, Skype)

Although open source is partly aimed at the division of work, many people can be working separately on the same function or module. This seems very inefficient. Companies, however, are often aware of this problem and try to bring these people together as much as possible.

"Pricing is often ridiculous in closed source." (Conversation with Mr. Keller, 26/04/2012, Skype)

This interesting remark actually came from a CEO of a closed source company. He was convinced that pricing should not be based on usage (e.g. number of users or servers). At his company, Oxcyon, they do not charge per user, but charge based on the size and structure of the software. Sam Keller enlightened his vision by giving a metaphor:

"Imagine you bought an Audi for \$ 30.000, but if you drove it 300.000 miles you would have to pay an additional \$90 000. This is ridiculous. They should not be charging based on usage. It is either the car works, or it doesn't work." (Conversation with Mr. Keller, 26/04/2012, Skype)

An important order winner for many open source companies is the avoidance of a vendor lock-in. When choosing an open source solution, clients have the possibility to switch between suppliers when they are not completely satisfied, without having to start all over with a completely new system. An interesting remark in this regard is the following quote from the conversation with Concrete5:

"The smaller guy sees open source and links it to free. The bigger guy sees open source and links it to 'I have independence'." (Conversation with Mr. Maruna, 01/05/2012, Skype)

"A disadvantage of open source is that it involves politics." (Conversation with Mr. Maruna, 01/05/2012, Skype)

This has a lot to do with intrinsic motivation of the external developers, active in the community. It is not always easy to keep those external developers, who are of vital importance, happy and motivated. Sometimes it happens that the work they do is just not good enough. It can be painful to explain this to the developer concerned, as he might have spent a lot of time and effort in what he did.

"If they are volunteering their time, the time you have to spend to make them feel good, while at the same time you are not happy with what they have done. That is just a political nightmare." (Conversation with Mr. Maruna, 01/05/2012, Skype)

In community open source, development is often being done by developers who are active in web development companies that are reselling the open source software. These resellers usually apply a percentage-based time allocation structure to allow their personnel to work on open source projects. Generally, this is represented in full time equivalents.

An interesting trend was the fact that closed source companies are using client and/or partner companies. We believe that they have recognized the benefit of having such a community by examining the business of open source companies.

The free price tag was a point that was never stressed during the interviews with open source companies. 'Free' often has a bad connotation; there are many other advantages you can mention when selling an open source product. That is why we did not incorporate the free price tag in the value proposition of an open source company. "The fact that it is free, might even be the least positive feature about open source." (Conversation with Sjoerd van Elferen and Rob Gietema, 27/02/2012, Arnhem)

Transitions and evolutions

During the interviews, we always enquired about evolutions in the CMS market. Most interviewed experts mentioned the same trends. The most important evolutions regard the functionality of the software: social media integration, mobile computing support, web marketing and analytics, geo-information, responsive designs and real-time collaboration.

Next to the adding of functionalities, an important transition is towards the cloud or SaaSbased delivery models for software companies. It is certainly a trending topic and there is a lot of buzz about the concept. However, cloud-based solutions are being served in many different forms. Different companies have different ideas about the concept. Some companies, particularly closed source companies, do not embrace the idea of multitenancy in SaaS. They do offer a cloud-based version of their product, but apply a singletenant structure.

"Each client on a dedicated hosted instance can really decide the way he wants to move forward" (Conversation with Mr. Smith, 25/04/2012, Skype)

The SaaS issue seems security related; security, in this sense, of customers feeling safe. The same as customers might not feel safe having a software solution of which the source code is available; they might not feel safe having their data on another server either. Therefore it seems that closed source customers will not as easily choose a SaaS-based solution as open source customers. Hannon Hill, Anonymous X and Oxcyon all were using a single-tenant model to approach their customers. It can be concluded that there is still a lot of room for competitive differentiation in SaaS. Every company has its own idea about it and probably different sets of customers will embrace a different idea.

As a final conclusion of this research, we believe that there is no clear direction the market is evolving to. During our interviews, employees of closed source companies informed us of people moving away from open source, whereas active community members in open source projects were convinced that open source was gaining popularity and thus an opposite trend was being noticed. We therefore conclude that it is hard to predict which way the market will go.

PART 3: GENERAL CONCLUSIONS

In this master thesis, we started off by extensively examining literature on four separate subjects. The first chapter dealt with business models. We presented an overview of the available definitions and elaborated on our own definition:

"A business model is the framework that contains the company's basic logic on value creation and delivery to its customers and suppliers, obtained by the input of its available resources, activities and partners, while keeping in mind the financial outcomes of these operations." (Supra, p. 7)

This framework can be subdivided in several important building blocks that describe the key elements of a business model. These 9 components are: customer segment, value proposition, channel, customer relationship, key resources, key activities, key partners, cost structure and revenue streams. This framework of components was originally proposed by Osterwalder in his book 'Business Model Generation' (2010). It served as a basis to analyze the interviewed companies in our research.

In the next chapter we clarified the concept 'open source'. We gave a brief history of how open source originated. The most important different licenses available were displayed, in which a distinction was made between strong copyleft and weak copyleft licenses. The chapter also gave an overview of the different business models existing in the world of open source software. The chapter concluded with an extensive explanation on the dual licensing model, which is nowadays a popular business model in open source.

The next chapter of the literature review handled content management systems. It briefly defined what we mean by content. We further discussed the history of web development that lead to the existence of content management systems. After having explained how a contentment management system works, we elaborated on the current market situation.

The final chapter dealt with one of the most important trends in software engineering: Software as a Service. In this chapter we defined the concept, summarized its most important characteristics and addressed the issue of 'the long tail'.

To investigate the business models being used in ICT firms we focused on the sector of content management systems. We selected this specific market because of the wide range of both open and closed source solutions available. In our research, we looked for fundamental differences and similarities in how open and closed source companies work. Therefore we had expert interviews with 10 different companies. Based on the 9 different components of a business model, we analyzed the business of each company. These analyses were the foundations to derive a business model for both closed and open source companies.

To conclude our research, we discussed the differences and similarities of the business models of open and closed source companies. The most important commonality found in the business models is the use of a partner network in the distribution of the product. The main differences were to be found in the value proposition of the companies. Professional services often lack in open source software, whereas this usually is an order winner for closed source companies.

Some other important conclusions were made based on the interviews. These were either very remarkable statements made by the interviewees, transitions we noticed in open and closed source companies or general evolutions in the market of content management systems. The final conclusion stated that we believe there is no general trend to be noticed in the ICT market. Some people are moving away from open source software, while others are embracing the concept. The same was true for closed source software.

The limited number of companies willing to cooperate with this research certainly has an influence on this master thesis. Although we believe to have covered the broad scope of the different existing companies, it would be beneficial to increase the number of companies. Due to this limitation, we did not make a distinct business model for corporate and community open source software. The differences between corporate open source and community open source definitely is an interesting topic for further research.
References

Afuah, A., & Tucci, C. L. (2000). *Internet Business Models and Strategies: Text and Cases*: McGraw-Hill Higher Education.

Alexander, O., & Yves, P. (2002). An e-Business Model Ontology for Modeling e-Business: 15th Bled Electronic Commerce Conference, e-Reality: Constructing the e-Economy. Bled, Slovenia, June 17 - 19, 2002.

Alt, R., & Zimmermann, H. (2001). Introduction to Special Section - Business Models. *Journal on Electronic Market*, *11*(1), 3-9.

Anderson, C. (2004). The long tail. Wired 12(10).

Applegate, L. M. (2000). E-Business Models: Making sense of the internet business landscape, In Information Technology and the New Enterprise: Future Models for Managers. In G. Dickson & G. DeSanctis (Eds.), *Prentice-Hall* (pp. 49-101).

Boiko, B. (2005). Content Management Bible. Wiley Publishing.

Bonaccorsi, A., & Rossi, C. (2003). Why Open Source software can succeed. *Research policy*, *32*(7), 1243-1258. doi: 10.1016/s0048-7333(03)00051-9

Bouwman, H., & MacInnes, I. (2006). Dynamic business model framework for value webs. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06).

Browning, P., & Lowndes, M. (2001). JISC TechWatch Report: Content Management Systems. *JISC*.

Chesbrough, H. W. (2003). *Open Innovation: The New Imperative For Creating and Profiting from Technology*: Boston: Harvard Business School Press.

Chesbrough, H. W., & Rosenbloom, R. S. (2002). The role of the business model in capturing value from innovation: evidence from Xerox Corporation's technology in spinoff companies. *Industrial and Corporate Change*, *11*(3), 529-555.

Chong, F., & Carraro, G. (2006). Architecture Strategies for Catching the Long Tail. MSDN.

Driver, M. (2010). What Every IT Practitioner Needs to Know About OSS. Gartner Research.

Fensel, D. (2001). Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. *Springer-Verlag*.

Fitzgerald, B. (2006). The Transformation of Open Source Software. *Management Information Systems Quarterly*, *30*(3), 587.

Galper, J. (2001). Three Business Models for the Stock Exchange Industry. *Journal of Investing*, *10*(1), 70-78.

Gebauer, J., & Ginsburg, M. (2003). The US Wine Industry and the Internet: An Analysis of Success Factors for Online Business Models. *Electronic Markets*, *13*(1), 59-66.

George, G., & Bock, A. J. (2011). The Business Model in Practice and its Implications for Entrepreneurship Research. *Entrepreneurship Theory and Practice*, *35*(1), 83-111.

Gomulkiewicz, R. (2004). Entrepreneurial Open Source Software Hackers: MySQL and Its Dual Licensing. *Computer Law Review and Technology Journal, IX*, 203-212.

Gordijn, J., & Akkermans, H. (2001). Designing and Evaluating E-Business Models. *IEEE Intelligent Systems archive*, *16*(4), 11-17.

Hecker, F. (1999). Setting Up Shop: The Business of Open-Source Software. *IEEE Software*, *16*(1), 45-51.

Hecker, F. (2000). Setting Up Shop: The Business of Open-Source Software Retrieved February 10, 2012, from <<u>http://www.hecker.org/writings/setting-up-shop</u>>

Jansen, W., Jägers, H. P. M., Steenbakkers, G. C. A., & H., M. (2003). Business models - ontwerpen voor de toekomst. *Den Haag: Ten Hagen & Stam*.

Jansen, W., Jägers, H. P. M., Steenbakkers, G. C. A., & Melger, H. (2004). Business Models in Nederland - Onderzoek naar het voorkomen van business models. *Primavera Working Paper, Universiteit van Amsterdam, Nederland*.

Koenig, J. (2004). Seven Open Source Business Strategies for Competitive Advantage. *IT Manager's Journal*.

Krishnamurthy, S. (2003). An Analysis of Open Source Business Models. *University of Washington*.

Kroon, M. (2009). Constructie van innovatieve business models - Een andere kijk op innovatie. *Thesis, Universiteit van Amsterdam, Nederland*.

Lerner, J., & Tirole, J. (2002). The Scope of Open Source Licensing.

Linder, J., & Cantrell, S. (2000). Changing Business Models: Surveying the Landscape. *Accenture Institure for Strategic Change.*

MacComascaigh, M., Murphy, J., & Tay, G. (2011). Magic Quadrant for Web Content Management. *Gartner Research*.

Magretta, J. (2002). Why business models matter. *Harvard business review*, 80(5), 86-92.

Markides, C. (2000). All the Right Moves: A Guide to Crafting Breakthrough Strategy. *Boston: Harvard Business School Press*.

McKeever, S. (2003). Understanding Web content management systems: evolution, lifecycle and market. *Industrial Management & Data Systems, MCB UP Limited, 103*(686-692).

Mitchell, D., & Coles, C. (2003). The ultimate competitive advantage of continuing business model innovation. *The Journal of Business Strategy*, *24*(5), 15-21.

Onetti, A., & Capobianco, F. (2005). Open Source and Business Model Innovation: The Funambol Case. *First International Conference on Open Source Systems*, 224-227.

Open Source Initiative. Open Source Case for Business Retrieved March 1, 2012, from <<u>http://www.opensource.org/advocacy/case_for_business.php</u>>

Open Source Initiative. The Open Source Definition Retrieved February 22, 2012, from <<u>http://www.opensource.org/docs/osd</u>>

Osterwalder, A. (2004). *The Business Model Ontology - a proposition in a design science approach.* Thesis, University of Lausanne, Switzerland.

Osterwalder, A., & Pigneur, Y. (2010). Business model generation : a handbook for visionaries, game changers, and challengers.

Osterwalder, A., Pigneur, Y., & Tucci, C. L. (2005). Clarifying Business Models: Origins, Present, and Future of the Concept. *The Communications of the Association for Information Systems*, *15*(1), 1-40.

Pateli, A., & Giaglis, G. (2003). A Framework For Understanding and Analysing e-Business Models. *Proceedings of the Bled Electronic Commerce Conference*.

Porter, M. (1996). What is Strategy? Harvard Business Review, Nov-Dec, 61-78.

Raymond, E. (1999). The Cathedral and the Bazaar. *Knowledge, Technology & Policy, 12*(3), 23-49.

Riehle, D. (2009). The Commercial Open Source Business Model. *Value Creation in E-Business Management*, *36*(1), 18-30.

Shafer, S. M., Smith, H. J., & Linder, J. C. (2005). The power of business models. *Business horizons*, *48*(3), 199-207.

Tapscott, D., Ticoll, D., & Lowy, A. (2000). Digital capital - Harnessing the Power of Business Webs. *Boston: Harvard Business School Press*.

Te Fu, C. (2009). Building a platform of business model 2.0 to creating real business value with web 2.0 for web information services industry. *International Journal of Electronic Business Management*, *7*(3), 168-180.

Teece, D. J. (2010). Business Models, Business Strategy and Innovation. *Long Range Planning*, *43*(2–3), 172-194.

Timmers, P. (1998). Business models for electronic markets. *Journal on Electronic Market, 8*(2), 3.

Välimäki, M. (2003). Dual Licensing in Open Source Software Industry. *Systemes d'Information et Management, 8*(1), 63-75.

W3Techs. (2012). Usage of Content Management Systems for Websites Retrieved May 10, 2012, from <<u>http://w3techs.com/technologies/overview/content_management/all</u>>

Watson, R. T., Boudreau, M.-C., York, P. T., Greiner, M. E., & Wyn, D. J. (2008). The Business of Open Source. *Communications of the ACM*, *51*(4), 41-46.

Watson, R. T., Wynn, D., & Boudreau, M.-C. (2005). JBOSS: The Evolution of Professional Open Source Software. *MIS Quarterly Executive*, *4*(3), 329-341.

Weill, P., & Vitale, M. R. (2001). *Place to Space: Migrating to EBusiness Models*. Harvard Business School Press.

Appendices

1. Interview

General CMS & market

- Where do you excel? What are the CMS' key features? How to distinguish from competitors (competitive advantage)?
- What are the differences between Open/Closed Source?
- What are the advantages of this?
- What are the disadvantages of this?
- What are the differences between big and small players on the market concerning policies, governance?
- How do you react to competition in the (CMS) market?
- How did your firm's business model evolve during the last years/decades?
- How did the market evolve during the last years/decades? (in general (no difference between open en closed source), specific markets)
- What will probably be future innovations concerning business models in the sector?
- What will probably be the future innovations concerning the specific CMS product?
- Where are the main differences, according to you, between OS community, OS corporate and CS in this four fields. How do you think this 4 key components will evolve?

<u>Product</u>

- Does the firm aims for a specific target customer segment (which (geographic) market/ (product) segments), B2B or B2C?
- What value is aimed to deliver to the customer?

- What are these main capabilities the company has/should have? (e.g. error and complaint handling, customization...)
- What were the main evolutions on this field in the past? And what's expected in the future (innovations)?

<u>Infrastructure</u>

- How are the resources and assets of the firm managed (human, intangible, tangible)? In-house/Outsourced?
- What are the most important resources and assets for the firm?
- How are relationships with partners managed?
- What are the most important relationships? Which partners? Why?

Customer Relationship

- Who is your customer? (especially for OS)
- How do you communicate with the customer? How to get info from them?
- What do you do with information gained from the customer? How?
- Do you try to measure trust and loyalty? Do you follow up your customers?
- How do you get the product to the customer (what is the channel strategy?) Direct/Indirect?

Revenue model:

- Where do you get revenue from? (Licensing, maintenance?)
- How is the licensing structure organized and monitored?
- Pricing structures?

<u>Cost structure:</u>

- Focus on core competencies & outsourcing for cost savings?
- New/innovative opportunities for cost savings in future?

2. Transcript interview – Ben Hodous, Accrisoft

Participants:

Reinout Denys	Ghent University
Gert Vanhaverbeke	Ghent University
Ben Hodous	Accrisoft
Location:	Internet call, Skype
Date:	Friday, 20 th April 2012
Time:	09.00 - 09.40 (EDT)

Questionnaire

What do you believe, are the advantages of a closed source CMS over an open source one? What are the disadvantages?

The primary advantage of a proprietary system is that everything was developed by the same group of people. Major benefits of this are quality assurance, consistency of the user interface and tight integration between parts of the CMS.

An open source CMS usually relies on extensions, which may not be compatible with the core system upon upgrade. These extensions can't usually integrate with other extensions, because each website is configured differently and the installed extensions are never the same.

One possible disadvantage of a proprietary system is that development is limited by the number of developers on the staff. An open source system can potentially be developed by hundreds or thousands of people around the world (for better or worse).

Does Accrisoft aim for a specific target segment? What are its key differentiation points, order winners or order qualifiers?

Accrisoft's target market is any company or individual who develops websites. Typically, if a potential client is interested in our software for their own site but isn't experienced in web development, we refer them to one of our partners. Open source generally works with a community. Do you have a sort of developers community for the evolution of the product? Are there any other partnerships or important relations concerning research development?

Instead of the members of our community developing the main product as in open source, our customers develop templates for many of the modules in our system, and share them with one another. This requires knowledge of HTML/CSS rather than PHP, and allows people to improve the system without harming the core functionality of the software. Client feedback is also very important for us, and we maintain an active feature request list from our clients, allowing them to influence the direction of development.

How do you manage your relationship with clients? Is there a direct contact, or do you work with intermediary partners?

Our clients are web development companies; we do not deal with their clients, the endusers of the software. We provide support to our clients, who maintain their own relationships with their customers.

In addition, have a sister company called Accrinet which is a web development company and, essentially, one of our customers.

How are revenues generated? What kind of different streams are there? (Licensing, maintenance,...?) How is pricing done?

Revenues are generated primarily through monthly fees for our software. This fee includes the software license, hosting, server maintenance, upgrades and support. We also generate non-recurring revenue through professional services.

Are there any general evolutions to be expected in the CMS market?

We feel that there has been a growing trend of people moving away from completely free CMS systems that don't offer hosting, support and integration. We expect this trend to continue, with companies building proprietary frameworks upon open source software, with companies offering hosting and support for open source systems, and with people leaving open source for proprietary systems.

Interview

You told us one of the major benefits of a CS CMS was quality assurance. Why do you think this is more in CS than in OS? Because in OS, maybe more developers are looking at the code and quality can be assured as well.

It's not just the quality of the developers that you can assure; it's also the same developers you can provide. The people on our team have built one part of the system, so they are familiar with it when they have to build the other part. So the whole system is designed from the beginning to work interactively.

About the revenue model, there are different packs you can buy at Accrisoft. What are the different percentages? What's the percentage of what package is bought?

I don't have the specific data, but it's primarily the unlimited package that our customers buy.

And it's mostly via partners. Are there individuals as well that buy Accrisoft, or is it only the partners?

It's only via the partners. However, the process of becoming a partner is very simple. So if someone wants a website, we just make sure that they are capable of web development. It doesn't matter whether it is a person or company. Then we allow them to become what we call a "solution provider", which is actually a partner or reseller.

Do you foresee some training for people who lack experience in web building who want to use your packages?

Yeah, we do a lot of training for new people. We try to automate that by increasing the numbers of videos. We also provide a workbook that they can go through with demo sites. We'll also spend time going to meetings or on the phone with the customers. But generally we require them to have certain knowledge of HTML and CSS skills to begin with.

And so revenue is generated through training as well?

No, training is for free.

Are there some other partners, whom you work with, not only downstream with clients, but also upstream with for example developers you use for the software?

We don't use external developers. We do all the development in house. We partner with a lot of hosting companies to offer our product. We have relationships with a number of hosting companies.

So we believe a new trend in that is software as a service. Is that why you work with hosting companies?

That's a big part of it. We do server management, so companies don't have to worry about that. When they place an order for one of our sites, it's usually ready within an hour. It's already set up on the servers. They don't need to know how they are set up through apache and server work etc. You just place an order and it is ready.

Was that before SaaS was trending, or was it recently added to the service you offer? Or was it something from the beginning?

At some points, people were installing the software on their own servers. But just looking at the situation, we decided this was ideal.

So you adapted your business model throughout the years?

Yes.

Can you see some evolutions in your kind of business model in the future, with new ways of doing business? Maybe bigger differentiation of products or new ways of going to the customer?

Yeah, I think one thing we'd like to do, is increasingly automate things. For example, when you order the software, you can maybe install it with one-click. Don't go through the process, just instantaneous and automated.

You also answered in the questionnaire that you basically aim for every customer that wants to build a website. So you don't really aim for a customer group?

We tend to have certain markets that are more represented. We work with a lot of chambers of commerce around the US. Typically for the US is that many organizations work on membership base. We don't limit ourselves in any way. We feel it's for any type of website.

How do you reach your customers? Which kind of marketing tools do you use?

We try to do a lot of the social networking. Twitter, Facebook, etc. We have gone to some tradeshows recently. We blog, we put an ad in a magazine last year, but we're mostly focused on web 2.0. We get a lot through word of mouth.

Did your customer base grow a lot recently, or is it kind of the same for the last few years?

It's been growing well. More and more people heard of us.

When we look at the other side, at the cost structure, what are the main costs for Accrisoft?

Hosting is a very large cost. We have a lot of servers. I am sure that would be the number 1 cost. Otherwise, personnel and development. I don't have the exact numbers...

Is there a relationship with your customers in two ways, to have some feedback and hear what day have to say about what features they want in the software?

We have a feature request form. We get a lot of information from that form. On our ecosystem, which is a sort of the reseller partner portal, we allow people to vote and rank features that people are requesting. They are very open about what they want, so we get a lot of feedback.

What are in your view the most important resources or assets that your firm has to have to offer a good service?

We need a good support, a good product and a good development. We provide good support to our customers. We try to pick up the phone on the first ring, and give them that personal touch that if they have a question we're able to answer. In my point of view, that's pretty important and something you can't always get with open source products.

Do you believe there is a different perception from clients towards open and closed source?

Yes, most of the clients that we have, have tried to build a business with open source. They've bumped into various problems, especially when the business was growing to a larger size. So most of them already know what the problems are. They really appreciate the difference that we can offer them.

Do you consider OS as large competitors for you or not?

Most of the people we're targeting are indeed the type of customers that are considering Open Source. They are not exactly a direct competitor, because they are in a different business. They are just providing free software. I think the products themselves are what we're competing against.

Do you have a view on the differences between some big players on the market and how they get their product to the customer?

I can't really comment on that. I'm not an expert on that.

Is there some kind of community in Accrisoft, or is this the Ecosystem?

Yes, that is indeed a community for our partners, and we have forums and release notes. We have marketing materials, best practices, etc.

How big is this community? How should we picture that?

It's difficult to say. We have 100 000s of people, but they are not all active members.

Are there any important evolutions you notice in the CMS sector, or evolutions you saw in the past that are now standard?

I think people are realizing that they need more training, support, hosting, they need someone to help them. They need someone who knows what they are doing. So even we're seeing companies that are offering support for open source, we see companies building on top of OS, or companies like ours that are CS, but offer those things that people need. People are just realizing that trying to do all things by themselves, especially when the business is growing large, is just impossible.

So an OS company who offers hosting, you consider them as a competitor rather than an OS one without?

Yes, it's closer to what we do.

3. Transcript interview – Eric Kuhnen, Anonymous X

Participants:

Reinout Denys	Ghent University
Gert Vanhaverbeke	Ghent University
Eric Kuhnen	Anonymous X
Location:	Internet call, Skype
Date:	Friday, 4 th May 2012
Time:	08.30 – 09.00 (Pacific time zone)

Questionnaire

What do you believe, are the advantages of a closed source CMS over an open source one? What are the disadvantages?

I don't think this is the correct question. I believe the principal factors affecting any CMS are the pace of innovation, the ability to respond to customer needs, and the ability to derive a profit. The relative success of any CMS project depends entirely on these factors alone. Also, what is the definition of "advantages" (and "disadvantages") in this context? Open- and closed-source are licensing terms, and each has its advantages and disadvantages in relation to a specific context. What's the context?

Does Anonymous X aim for a specific target segment? What are its key differentiation points, order winners or order qualifiers?

The Anonymous X platform targets the creation, management, and publication of technical product documentation. Our qualification process and points of differentiation are proprietary.

Open source generally works with a community. Do you have a sort of developers community for the evolution of the product? Are there any other partnerships or important relations concerning research development?

Anonymous X is a closed-source platform. The development community is bound by a non-disclosure agreement.

How do you manage your relationship with clients? Is there a direct contact, or do you work with intermediary partners?

Anonymous X has both direct and indirect client relationships.

How are revenues generated? What kind of different streams are there? (Licensing, maintenance,...?) How is pricing done?

Anonymous X is sold as a subscription service ("Anonymous X On-Demand") and as a licensed software package ("Anonymous X On-Premises"). Our revenue streams, licensing terms, and pricing algorithms are proprietary.

Are there any general evolutions to be expected in the CMS market?

The pace of innovation is brisk and on many fronts. One important area of investigation is the market's desire to integrate mass-user feedback into the content production cycle.

Interview

You said content management systems are specified by the pace of innovation, the ability of responding to customer needs and the ability to derive a profit. Do you believe open source and closed source perform evenly well on all these fields?

Yes, I think they do. OS is not really a business model as much as a development model. We choose a closed source developer model. But an open source solution probably will have greater major developers. Though, you still have to make money if you're going to service that platform in some way. Otherwise people can't continue to develop it. So there has to be some revenue model, and I think they perform relatively the same.

You said Anonymous X is a closed source platform, but you do use a development community. Is this the community of company developers, or are there also some external developers?

It's a closed community, we will either use in-house resources or contract-developers from other geographies who are assigned to Anonymous X and work only on Anonymous X.

About the relationship with your clients, you said you have direct and indirect relationships. Don't you think you can become competitors with your partners? No, us going indirect has nothing to do with our ability to operate. It has to do with our ability to operate in certain market niches in which we have no expertise. So we choose an indirect way, because it's more efficient to use local resources than to put our own resources in that geography.

How does one become a partner with Anonymous X?

It's usually through reference. Either by common consent or some other sort of reference, we find people to form a partnership.

You said already that you have partners for development and to go to the client. Are there any others?

No.

What do you believe are the most important resources and assets you should have? (Skills, knowledge,..)

It's mainly a technical form of resource. You need to have some technical expertise as part of the offering process. For any company you should have a product that operates reliably. Many vendors get in trouble when they are trying to sell a product in another segment than initially aimed for.

You said is kind of technical to use the XML CMS. Is it correct to assume you aim for a customer who has more technical knowledge?

There are several XML CMS vendors. Each vendor in the market requires the customer to have some technical knowledge of XML in order to appreciate the benefit that comes with an XML CMS. Without knowledge of XML, the value proposition of an XML CMS is more difficult.

Do you one way or the other look at your competitors, of what they do, what features they add?

Oh yes, we always look at each other.

Are there many OS competitors for the XML CMS?

There are just one or two OS XML CMSs. I know of one company that has built an XML CMS on top of the existing OS CMS, which is not specifically a XML CMS. I think there are some OS solutions out there, but they are nowhere in size in comparison to CS CMS.

So you don't feel much competition of OS then?

No, there's not much competition. The market is very small, so I don't think there is enough driving innovation for OS. Unlike say Drupal, which is an OS web CMS, where the opportunity is large enough for OS development. The same is not true for the XML CMS market.

That market, how did it evolve the last couple of years, and how do you think it will evolve in the future?

The market has largely been driven by the adoption of the standard called DITA. A secondary, not so strong in the US, but more in the EU, is another standard S1000B. Those standards drive the most innovation in the market.

You offer your services through third party hosting services, so you don't have your own servers?

That's correct. We made a conscious decision not to invest our money in building our own data center infrastructure. We use a third party data center.

It's clear that you are offering a SaaS product, are there any security issues with that as there is data of several customers stored on one and the same server?

Well that's a different issue than SaaS. SaaS is only providing the service available over the internet. What you're describing is multi-tenancy. We don't make any claim about that. In fact we use a single-tenant model.

Are there some different kinds of types of software? A very elaborate version perhaps and a limited version for example?

Our services are homogeneous. We offer a single service and we may choose different price targets for different markets we go after, but it's the same service.

The main costs for the customer are the yearly fees for the service, or is it the initial fee for the software?

Well, the way we do our pricing for our system is based on usage. There are several vendors in the market who have a variety of fee-based modes. Anonymous X's model is based on the number of users on the system. Another company has three license model. Some of them are per minute, such as cellphone usage, some of them are per users. They have a wide variety. But most of these SaaS vendors are charging per user.

And are there any other revenue streams, such as revenues from training perhaps?

Yes, we certainly collect revenues for professional services, of which training is a part. The majority of our focus though, is not on professional services. Our business model is more based on the use of the system than on the professional services to extend that system.

Concerning the costs of your company, what do you see as the main cost of operating Anonymous X?

That's proprietary information.

Do you do a lot of outsourcing or do you keep things in-house?

Anonymous X prefers to do most of its own work.

How do you measure trust and loyalty? Do you follow up your customers somehow?

We don't use measurements in that sense. We do have very good customer relationships and a very loyal customer base. The means by which we test and assess that loyalty are much more personal than some sort of metric or a loyalty program.

And how do you get information from the customer concerning new needs or features they would like to add?

Yes, that's one of the reasons that customers stay with Anonymous X. We have a very close relationship with them. We meet with our customers weekly, and we've been doing that for multiple years.

Do you have some kind of roadmap where you want to go with Anonymous X between for example this and five years?

Sure, we have roadmaps.

What do you believe will be important for the next coming years in the CMS market?

I think that the market is trying to figure out two things. Firstly, how to get end users to collaborate in the creation of the content without polluting that content. Second thing is how to get subject matter experts to contribute content, especially structured XML content without having to know very much about XML. I think all of the vendors, including Anonymous X, are trying to innovate along those two fronts. As to the specifics of Anonymous X's roadmap, that's proprietary.

4. Transcript interview – Brian W. Brown, AppState/phpWebsite

Participants:

Time:	10.00 – 10.30 (EDT time zone)
Date:	Friday, 6 th April 2012
Location:	Internet call, Skype
Brian W. Brown	AppState/phpWebsite
Gert Vanhaverbeke	Ghent University
Reinout Denys	Ghent University

What are the advantages of an OS CMS over a CS CMS?

We've been running the phpWebsite project for over 12 years now. Of course, 12 years ago the selection of available CMSs wasn't that broad like today. Our decision to create our own CMS was driven by the need of having a platform that we could control, primarily to assist us in web application development. Although phpWebsite is a content management system, internally at Appalachian, it serves as a framework for application development, where we can design custom applications to use at the university. We have several very large applications, which are currently not open source, but we are currently considering of moving OS with those. The initial idea was to have a flexible framework which we could control. We open sourced the project, because, being a public institution, we use public funds to operate and so that was an easy way for us to give back to the community. By open sourcing it, we had some very good collaborations with other individuals and universities. We're not a profit-making business and so there really wasn't any downside to just giving it away.

Is there any commercial purpose towards phpWebsite?

I think there are some people using it commercially, but we are a public university and we have no commercial side other than serve as a service framework to develop applications which the university uses.

But development is in your hands? Or is it done by the people who commercialize it as well?

There are people who use it in a commercial sense, who have contributed to the development. Of course, the primary development efforts are done out of my office. I have a handful of developers who work on it. They also develop web-applications that are used by the university. But we are the primary developers.

But are there any disadvantages to open sourcing it?

I think there is absolutely no disadvantage. One of the great things of OS is that we get lots of eyeballs on the source code. The very few times that we have got problems with the code, security-related or so, the community has been wonderful at contacting us and pointing out the flaw so that we could fix it. I feel that by having the code OS, that actually increases its stability and security.

Is there a specific market you aim to with your CMS?

We don't spend a lot of time promoting phpWebsite. It's more like "here it is, if you find it useful, that's great." It is a fine line; we are a university and we have to serve the students. So we can't be spending public funds marketing phpWebsite for people to use it. That's not right. We can develop it for our own use and give it away, because that is different and the productive thing to do. We really don't have a market.

So for you at the university, there is not really competition with closed source. It's just 'here it is, and we don't worry about the competition'?

That would be it in a nutshell, yes.

Is the community managed in any way?

We don't actually manage the community. There are people who have created forms for external questions. We have an IRC channel where the developers sign in every day. When they log in to IRC and they have questions, we do our best to answer them. But we don't actually provide external support. The community is probably one of the smaller ones, but I feel it's very helpful and we have a very organic kind of approach to it. We let the external community be what it is as there is not a great deal of active management. Again, this is because we are very sensitive for making sure that public funds are being used appropriately. We're not the ones going to support it. The people who are making commercial use of it in hosting environments and stuff, they do that and tend to provide support.

The platform which are offered to the community to discuss and share their views. Who is responsible for offering that?

We set up things like the Sourceforge forms and stuff. Different members of my team will monitor those for questions, suggestions and bug reports. There are external forums and things that are being set up. I know several people who have based a hosting company around our software. Of course they provide technical support for a fee. But our main goal with the product is for our own use and sharing it is secondary.

How is research done? Is there any kind of roadmap?

Yes, there is a roadmap. That is largely internal and largely driven by the university's internal teams. What we plan is to release some of these web-applications that we've developed. We've developed a system for tracking internships for our students. There are some laws in the US to what's called state authorization and we have created software on top of phpWebsite to handle that. We have a co-curricular transcript and club management software which we're going to open source. We also have a housing system that takes care of housing applications and related things that we use to manage our on-campus resident halls. These things complement phpWebsite. We feel that sharing these things will help other universities.

Is there any relationship towards the end-user?

We really do keep them at arm's length and refer them to the external community. That's again because we have an extreme sensitivity for the fact that the project is being paid for by the university using public and student funds. I don't think this has been the best thing for the project, but given the source of the funding, we have to be mindful for that. It's just one of the constraints we have to work with. On the plus side, we do give something back to the public at large.

Is there another kind of partner network? Like web design firms etc.?

Yes, we cooperate with external developers. We have several other developers who also work at other universities. Certainly, we have those external conversations for people who have good ideas. Those conversations can result in a sort of relationship and result in improvements for the software. Ultimately we are getting free work done. People are improving the code, getting the software better,... That of course works out to be a very good cost/benefit ratio for us.

Is there any kind of revenue generate?

There are no revenue streams associated to our product.

What are the general evolutions in the CMS market we can expect e.g. cloud computing?

Things like content management systems are a kind of cloud services by their very nature. I think that sort of technology is a complement to a CMS and a CMS is in some sense a cloud-type service. Internally, when we create something like the housing management system, we see the potential of SaaS. We implement SaaS internally at the university. We have a lot of other applications which we purchase commercially, which are in the cloud as SaaS. I certainly think phpWebsite as a CMS and as a framework for web application development has a role to play in that. Although we provide it only internally. Things like the housing management system in a commercial environment would be ideal products to sell to a university.

Are there any other trends regarding CMSs, new things in functionality that can be important in the future?

I think that we're going to see better mobile computing support as it is becoming more and more important. When you have a web service or website that's being served and created by a CMS, the breadth of devices which can be coming to that site or service all have to be supported. Content management systems will have to keep up with the exploding number of devices that can access that content.

5. Transcript interview – Franz Maruna, Concrete5

Participants:

Time:	11.00 - 12.00 (Pacific time zone)	
Date:	Tuesday, 1 st May 2012	
Location:	Internet call, Skype	
Franz Maruna	Concrete5	
Gert Vanhaverbeke	Ghent University	
Reinout Denys	Ghent University	

Questionnaire

What are the advantages of an open source CMS over a closed source one?

An active community will use your software in ways you never intended. An active community will find edge cases you never considered. Being open source makes the sales process less adversarial. "Hey, it's free, if you don't want to pay us to help you - don't."

What are the disadvantages?

Any open source project involves politics. Programming and politics are challenging. Design by committee is a bad way to design.

Does Concrete5 aim for a specific target segment? What are its key differentiation points, order winners or order qualifiers?

No. We're a building material for the web. Just like our namesake, you can build anything out of it. People choose concrete5 for a wide variety of reasons. It's well coded, it's got a very easy to use interface, etc. Really the heart of it is we focus on the balance between a site owner's and site developer's needs. It's all about that relationship for us. The people getting paid to build the house, and the people who are going to have to live there for years.

How is the community managed?

With a copy of concrete5 we've customized extensively and a staff of about half a dozen.

How do you manage your relationship with the end user? Is there any customer relationship management tool that is being used?

We use WHMCS to manage our hosting and services customers. It's more of a billing system than a CRM, but it covers most of what we'd need out a CRM.

How are revenues generated? What kind of different revenue streams are there?

About half of our revenue is generated from marketplace sales. The other 50 is split between hosting and services.

Are there any general evolutions to be expected in the (open source) CMS market?

Continued consolidation as more of the larger clients who used to buy commercial software to be "safe" get tired of their annual license fees that provide little value and look to rebuild with open source.

Interview

What are the key differentiators for you product?

The key differentiators for our product is obviously in-context editing. No other CMS is doing it as gracefully as Concrete5 does. We start to see other systems that adopt the idea of having a gateway to get to the right spot in the back-end from the page on. That ability to have a site, see a typo and put the page in 'edit'-mode, change the typo and you are done, it just makes a lot of sense to customers, as it is very much the opposite of the Drupal or WordPress model, where they have to find it in the back-end. Taking that intimidation out of the process for a site owner really helps a lot.

I think the deeper answer to that is that we really try to architect Concrete5 as a toolbox for building design-centric websites. We started as an interactive media firm and we needed this toolbox for our own clients. We always thought of the balance between the needs of a site owner and the needs of a site developer as fundamental of what we do. We often use the metaphor of home construction or building and once we first started, we thought it was ridiculous that every time a client came along, you say 'you need a house, the first thing we are going to do is like build a forge and make a hammer.' It just seems that there should be a slightly better set of tools available. We also found it very frustrating that any client that we ran into that had any existing web presences; typically it ended up being trash. Even if they were using something that we knew, like Drupal, it took so long just to figure out how the last shop had used that framework and very often we came to the spot where we would better of starting from scratch. This way, we wanted to introduce some kind of consistency to the building materials for the web. One agency could work on a project, finish it and another agency could pick it up. A client could have a site built with something and it would work the same way as any other site built with it. And really, you could have the same trust that you would get from a certified home builder like 'look if you built my home it is going to be built in a sane way and I would not have to call the same person that built this thing just to fix the plumbing five years from now, anyone could fix the plumbing. That is what we want to do. Every decision we make has that balance between 'will this help the site owner' and 'will this help the site developer?' and 'is that relationship healthy and does this feature improves that relationship in some way?' I don't think there is a lot of our competition that looks at this that way. I think they look at it like there is stuff on one side or the other; WordPress is very consumer-focused: you could build a blog drunk, Saturday night; it still is going to work on Sunday. Drupal on the other hand is much more technology-focused. If you are a developer there is all sort of stuff that you can implement, but when you give it to your customer it is like 'oh my god, what am I looking at.' That balance is fundamentally what sets us apart.

In the questionnaire you mentioned that there wasn't really a target segment, but on the site it says that it is made for marketing. Is that the main target group, those marketing people?

No, our slogan is: 'made for marketing, but built for geeks.' We have found that website projects typically are driven by one of those groups. So it is either the marketing folks that need a new site or it is the IT folks that are tired of updating what exist. Both of those parties have to buy off on the choice of platform. Concrete5, actually, appeals very well to both. We get involved with projects were either party is championing the use of Concrete5. Our slogan is kind of out there to show that balance of look 'this is something an engineer is going to like, because is modularly controllable, it is object-oriented, it is well architected, and this is something that a marketing manager is going to like as the can actually use it and edit stuff without having to call the geeks.' In terms of a vertical; 'is Concrete5 really great for dentists' websites, but not so much for lawyers? No.' you can build anything with Concrete5 and that doesn't matter to us. In terms of scale; 'is Concrete5 really great for tiny projects, but not big ones? No.' It is open source, and what we find going open source is that immediately when you go open source you get a lot of small clients; you get a lot of churches and non-profit organizations. Those types of folks just think about you because you offer a free product. But we are also working with huge,

well-known companies, which are spending six figures on large web presences and using Concrete5 very successfully. It is a screwdriver; you can use it to pull anything.

You say that one of the advantages of an open source CMS is that the active community will find new ways for the use of your CMS. Isn't this also a disadvantage because maybe you, at Concrete5, have another vision than where the community is going to?

It is definitely a challenge. Running the community part is like a whole additional thing that you have to do. When you are thinking about the differences between commercial and open source, the best metaphor I can think of is that it is basically like adding a political campaign to your business. You have to sell people on the ideas instead of just selling them on the products. That is definitely extra work and you have to be aware of that. I do think that you will find vast differences in how different projects treat that. I believe Joomla is run by a revolving committee of folks, which actually sprung out of another project called Mambo. Mambo was open source as well but it was really mismanaged for a while and they didn't do very good. Version releases of which it wasn't clear what was going on, it was kind of owned by this one company that didn't do a good job with it, so people split it off and called it Joomla. They wouldn't want to make the same mistakes again and they have swung the pendulum in the other direction and now everything that they do is based by a committee. I actually believe that every six months they elect a new core team that is in charge of deciding what goes in to the application. That is not really my style. I tend to think that the more cooks you get in the kitchen, the more likely it is going to taste bad. I am a big fan of the Mythical Man-Month concept; a task that is going to take six hundred hours for one person to do, having six people do it, that doesn't mean that it will be done in one hundred hours. That means it is going to get done in not much longer. The more people to add on something, the harder it is for people to understand what is going on and what their role is. That is very difficult to balance with 'it is a community and we are all crowd-sourcing this work'. I think you have to decide in the aspect of crowd-sourcing, which is only one part of what open source means, where does crowd-sourcing makes sense for your business. To me, I see a lot of success stories where you still have that small team involved. A story that is often told concerning crowd-sourcing is the story of Netflix; they had a project of figuring out a better algorithm. It is a few years ago now, but they came out with a million dollar price, for people who would found a better way to provide good suggestions for what someone would like to watch. If you were able to improve it a few percentages you would win the price. They had a lot of teams competing with their algorithms and models for it. They got some really clever thinking and someone succeeded. Immediately the press looks at that and says that crowd-sourcing is the answer

to everything. Sure, you have ten thousand people working on that, but they are all grouped up in sets of three, solving problems in their own ways. That works spectacularly well and that is why we have a market place with add-ons and themes. We spend a lot of energy creating a curation process around that market place, so when you submit an addon or theme, the peer review board will take a look at it. They volunteer their time as well to review their submission; 'your code is really sloppy here, or, you need to add ... statements there.' The community helps one another to keep the bar of excellence high for those submissions. Which products are going to work well in there, which one is going to sell? I don't know, no one knows. And it is up to the ecosystem to figure that out and I think crowd sourcing does a great job at that. On the flip side, we often are at odds with our community when it comes to managing the core. You guys won't have to do a lot of research to find threads in our forums, where we are arguing about this stuff. Their eager will have us approve poor requests that they are submitting and often those poor requests seem innocuous enough but they just touch different things and they are going to create bugs later. We can see that because we have a perspective that is different from where they are at and we have to not accept them, just because of the ecosystem that we live in. This way, it can be a very difficult balance because here you have a group of volunteers who are saying 'hey, I have spent a lot time working on this'. For them that is an afternoon's work, but we think 'we really didn't ask you to.' And we have spent a lot of time as well, as in thousands of hours per year. What you are doing is going to break what we have done so 'no thank you'. And that is hard, but the flip side for that for Concrete5 is that the design works well, the upgrades work. You kind of have to choose where you going to be on that scale. And I think it is definitely a scale and not an 'absolute committee always rules' or a kind of dictatorship; it is somewhere in the middle.

You were just talking about the market place. I believe it is pretty keen in Concrete5? If someone makes a module or an add-on, does the money go back to that person or does he gives the rights to Concrete5?

It works a lot like the Appstore. We take a twenty five percent cut on all the sales in the market place. We handle all the transactions and you can buy stuff immediately from within your own websites. If you buy a theme, you give us 30 bucks. Since the marketplace has modules from third-party developers, you can see the balance, developers can request a pay-out, and we pay out every week and we take a twenty five percent cut. That works well for everybody.

Is it only possible to get these add-ons via Concrete5, or is there another way where people can get add-ons from a place that is not organized by Concrete5?

You can do that. There are a couple of different ways to install add-ons. You can install an add-on by hand; you can download a zip-file and stick it in the packages directory. That will work. You can also get a package, which is a structured way to get an add-on, from anywhere and just install it; you would just have to know how to use FTP to get it in that incoming packages directory. A lot of the value that we provide is just the one-click install process. Speaking back on that balance between a site developer and a site operator. If you install Concrete5 on your budget-web hosting service, as it's available as a one-click installation, you get into this overlay within your site. Every day it might tell you some news and it might show you some themes that you may say 'hey, that, I would love to install on my site', you can purchase that right from within the CMS. This will initiate a project page on our site that will connect it for you, that will download it for you and within one click you have that installed. Technically there is really no reason why you would have to use our marketplace; it is really a matter of convenience.

Does it happen Concrete5 does not give permission for add-ons from external developers to sell them in the marketplace?

We have a soft rule on that. We strongly frown on people who are selling things outside of our market place if they are selling them in our marketplace. Our thoughts on that are 'we have this peer review board and you have taken advantages of their energy to get your add-on or theme in a nice fashion, looking good. Furthermore we are spending money marketing your stuff and promoting it so it is not very fair for you to go sell it on the side as well. Apple has an absolute approach on that with their Appstore, we don't. There are some exceptions that we have, like Brandy. A little chat that sits on the bottom of your browser, kind of like Facebook has. Of course, they have a product outside of Concrete5; we just help them build the Concrete5 version of it. You can buy that comment chat on your own or you can buy the comment chat for Concrete5 through our site. That kind of stuff is doable, but we generally push people away from that. We have had some experiences, typically with theme developers, in the past, when they were selling stuff on their site as well. We reprimanded them and said 'we would prefer that you wouldn't do that', and after a little bit of back and forth they said 'ok, will do it through your site'. We also have a section of special offers on our site, where as a developer you can take your add-on or theme, you can bundle it with someone else's add-ons and themes as well, and you can put different licenses together and offer them to a special price. So people use that

for saying 'buy an unlimited version of my theme and you will get 900 copies for the price of 10, or whatever. People often will also bundle similar themes and add-ons, like an addon for e-commerce that Concrete5 have developed, and a theme on e-commerce that somebody else has developed. We tell third-party developer to use the special offers section on our site to make whatever bundle he wants to offer on his blog. Then, we actually help them promote it and they will end up with increased sales. When using the market, the support for third-party developers is also a lot easier. Customers can create a support ticket through our site and you can see that actually purchased that. There is infrastructure to manage that, so the third-party developers will end up making more money.

How do you get in contact with third-party developers and how is cooperation managed? Is it only through the market store or are there also other ways you get in touch with them?

Developers are typically contacting us, more than us contacting them. There is a private forum for folks that have add-ons and themes approved. We got like a mechanism where they can ask questions and chitchat in there. We have got some email-distribution lists of everyone who is selling stuff in our market place, so if we need to pass down information via email, we can do that. And then there is a lot of activity in the peer review board. This board is largely made up by people who also have add-ons and themes in the market place, so if new works come in, each one of those submissions has a discussion attached to it, and a series of tests it has to go through. There is often a lot of chitchat in those discussions.

Fur us; the structure of an open source company is often complicated as it seems a bit like a large web. Can you give some kind of structure for Concrete5?

Well, we are an S-corporation. Here in the States, that means that we can have a limited number of shareholders and we cannot carry our revenue over from one year to the next. We never raised funding, so it is entirely employee-owned. In terms of the company, where it is at today. At first, before we were open source, we were a full service media shop. Back in 2007, we had around a dozen people, mostly developers. It was early 2008, and most of our clients at that point were startups. We were building online communities with what was just ConcreteCMS at that point. And a lot of our startups didn't get in the second or third round of funding as it was the beginning of the financial collapse. So we had been working on a brand new version of ConcreteCMS in the basement, and as we believed it looked awesome, we decided to just make it open source. So when we did that, we scaled things way back. We went from around twelve employees to just three. We moved out of our expensive offices and had a 'let's see what happens with this' idea. And it

started really taking off, so over the last few years, we have staffed back up. Right now we are about with ten people in that room. We are still mostly developer-focused, we have an in-house designer and we have a community manager that also manages a lot of our small services work.

Nowadays, is Concrete5 only about the open source CMS or are there other things Concrete5 does?

We are basically focused on Concrete5 right now, but we will be reposition ourselves as a general creative lab at some point in the not too distant future. We love Concrete5 as it obviously assumes the vast majority of our time, but we also do some interesting projects on the side.

You talked about the peer review group. Who is in this group?

All people from the community who have asked to be in it. There are about 30 members I think and when you are inactive in it for a while, we dump you out and put someone else in who wants to be in it.

Is there a specific reason why you do not offer an enterprise version of your CMS?

We have some tools built for the enterprise that would extend Concrete5 and these will come out later this year. But what I want to avoid is having two different products. You see this happen a lot; where you start with the open source one. And everybody loves that one. You think 'god, we really have to make some money from it', and so you start pulling features out of the open source one and putting them in the enterprise version. Then you have to maintain these two different code bases and you have to migrate from one to the other. Then the open source community members will start thinking, rightly I believe, 'hey, we are getting screwed, we get the junky one and the enterprise guys get the good one. We want to try to avoid that, at least that image. What we have found is that what the enterprise needs is typically very different from what the small organization wants anyway. It shouldn't have to be that way, and the way we feel it is that enterprise tools are value-added things that go on top. Our internationalization add-on; we have a free one that works just fine for an individual, and we also have one for the enterprise, that mostly adds tools around the team of people working on translation. So the bones are the same underneath each one, but the value-added factor for the enterprise one is all about having more tools to manage process as this is really the challenge that a large group has. I think we can do it that way. That we can give things the enterprise can choose to buy, but not offer crippleware to the rest of the world.

Concerning revenue streams; on the one hand you have got the services, on the other hand, the market place sales. Are there some other kinds of revenue streams?

There is hosting. About 50-60 percent of what we net is market place. The remaining 40-50 percent is the combination of services and hosting. We sell t-shirts, but that not really important.

How about cost structures, what are your main costs?

Definitely payroll. Payroll is by far the biggest cost. Rent is pretty cheap. The second largest cost is server infrastructure.

Do you practice a form of marketing and how is it organized?

Honestly, we do not spend a tremendous amount of money on marketing. We have danced with it in the past, but it is very difficult to see a meaningful return on the investment. We try to make sure that our site has clear messaging. We try to provide material for folks that go out and promote Concrete5. A tremendous advantage of being open source is that you have that huge group of advocates that want to go out and sell for you. If you can give them some attractive PDFs that they can use to acquaint your product, they will beat on doors for you. We do some energy around that. If people want to do a local meet up or if they want stickers in the mail, so they can go to their PHP-meet up and hand out those stickers, we will happily send them that and eat the cost on it. That, basically, is our marketing cost. The other options that we have; online advertising is not particularly valuable in our industry. The amount of money it takes to take someone to come to the site is just hard to track. Also deriving how much you will make off it is almost impossible. We also have done some tradeshows, but I think tradeshows are in trouble. It is so easy to find information on the internet these days. The times that you have to go to trade shows and meet channel partners are kind of past. I think SaaS is also making these channels a little bit tricky. The other real advantage is that open source is a free product. I don't need to have some bunch of sales guys convincing you to use it. Just use it and if you like it: 'cool, give me a call and maybe we can help', if you don't like it: 'cool, I didn't spend any money trying to help you'. That really changes the sales cycle. It is better to focus on making our product better. We have spent very little money on marketing for the last 3 or 4 years.

Is there some kind of vision for the future you have? And is there a kind of roadmap to achieve it?

Absolutely. We continue to want to be a leading building material for the web. That means we have to serve the two sides of the equation. We have to continue to add features for the small church sites that build themselves; and we also have to add features for the large enterprise that needs more accountability built in to Concrete5. Our own efforts this year are focused on adding more of those enterprise tools, but you also see us adding those one-click installers and that sort of stuff. I think we are doing a pretty good job on the smaller consumer with our market place and the in-CMS shopping experience. We see this stuff is working and we expect these things to mature. I think the new stuff you will see from us is more positioning near the enterprise and help people understand that Concrete5 works better as it is built by one person instead of hundred people working together on an own website.

Do you think it is realistic to both serve the small and the big players and not get stuck in the middle?

It seems to work. I don't see why not. I know, traditionally, everyone will tell you that it is very easy to pick a vertical and built towards it when offering software. But there are just many, many examples that you can point at where something has been built for one and then embraced by another. WordPress is a blog, but it is used powering some of the largest sites on earth, that's insane. If you invented a screwdriver, it would be easier to bring a screwdriver to the market if you called it a 'schlibidu' and you only market it for plumbers. However, it's a screwdriver, so anyone can benefit. It is worth your while to educate people on that.

You mentioned Software as a Service. Your hosted version, is that SaaS?

It is in the sense that you are paying by the month and we will maintain it for you. It is not in the sense that they are all managed through one centralized install.

You said in your mail that a disadvantage of open source, is that open source projects involves politics. What do you mean with that?

ConcreteCMS is been around since 2003 and when we first built it, it was me and Andy. We architectured it on cocktail napkins in a bar. There were a lot of ideas that weren't that good, there were some that were great, but the only way we learned that was by do projects for paying clients. The site would work but not really the way we wanted it to, and we said 'man, the next time we do one, we really have to redo this and undo that.' We spent 5 years, as commercial software, trying ideas out; keeping the good ones en throwing away the bad ones. When you have half a dozen guys in a room and you don't

have investors and I am just paying these guys, it is very easy to throw the bad ideas out. There is an afternoon when you evaluate and realize that a part of it was bad. Then you throw it out as it was not going to work, and you feel a little bad about it, but I'm paying you. If it would be fun, I wouldn't be paying you, so let's go on to the next problem. That is business; everyone is a grownup so you can do that. As soon as you go open source, you have got that volunteer thing. It is really cool that someone in New Zealand or so, volunteers two days to build some feature. But if he builds that feature poorly, then I have a problem. I have got my CTO saying 'this sucks' and I have this dude in New Zealand saying 'I really want this in there'. There is no money involved in any of that, so in many ways it is a waste of energy. I look at some of these projects where a guy in his basement builds something, it kind of works, he calls it open source and then they start to build a community. Sometimes I think 'that will be really though'. For us, the reason that there is a five in our name is that it is the fifth major rebuild with no back works compatibility that we did. And it was not just until that rebuild that we were completely happy with it. So much of software architecture thinking these days is about highly iterative development and bring something to the market and be willing to completely rethink it as you go and move on. The more people you add, the harder that is. Especially when they are volunteering your time, the time you have to make them feel good while at the same time you are not happy with what they have done and that is just a political nightmare. It is worth doing if you are at the right spot and you can see a real benefit from it. We would never be able to have all these things in the marketplace if we had to write it ourselves. It is totally worth there. Looking back when Concrete was not half a million lines of code, but fifty thousand and full of stuff we didn't like, we would have made a tremendous mistake to open source that. First we had to have a better product. I reject the idea that a thousand people that had never met each other somehow would make something better. I have yet to see that happen.

How do you deal with competition? Do you look at the features of other open source and closed source CMS systems or do you just do your own thing?

To be honest with you, we do not spend a tremendous amount of time looking at other CMSs. Way back in the day, before we build Concrete5; we were selling and installing other people's stuff. So I am pretty familiar with how painful those are to use. Occasionally, we look at Mambo/Joomla and Drupal. We see WordPress because all of our wives have WordPress blogs. But no, there are so many things out there that are called a CMS, that really sitting down and doing a general competitive review and saying 'these guys have that feature, so maybe we can build it better and win' is really reactive and doesn't really

help us. We tend to spend a lot of time looking at what our clients are doing and understanding the challenges that they are having and try to figure out how to better architecture things to better solve stuff in the ecosystem we actually live in. But I don't care what Drupal does.

You said not to believe in partner channels anymore, do you still have any?

That is maybe too strong of as statement. I think there is a lot of changes happening with the way software sales work, mostly tied to Software as a Service. But channel partners are definitely important, as we do more enterprise stuff. We are looking more to the system integrators and value-added reseller side than kind of distributor reseller model. Obviously, it is a content management system, so they don't really do a lot for you until someone builds you a website. Finding ways to partner with those folks, who are becoming your channel partners, system integrators, that is really important. Motivating them involves money, but not primarily. Something like a kickback is nice, but really it is about knowledge and education, and helping them understand why Concrete5 would be a good choice for their clients, so their clients continue to like them. It is important, but I think it is a really changing world for the moment. So it is an interesting one to watch.

You mentioned that a lot of large clients using commercial software for their company in the past are shifting to open source. Do you think it is only in that direction or is it in both ways?

I don't know. My experience has been that for big companies open source means a different thing to them than for the smaller guy. Frankly it is more of what open source meant to me. I think that the smaller guy sees open source and links it to free. The bigger guy sees open source and they link that to 'I have independence'. I can't count the number of times I have run into clients that have bought into MS SharePoint, built something with it and then been surprised at their recurring annual license fee, realizing 'oh my god, we have to pay this forever, we just didn't do our total cost of ownership that well'. Having that, beholding to Microsoft and hope that they make a better product, that is a tough one for big clients. We are working with clients who would rather own what they are paying for, as you always will end up paying either way. I'm sure there are folks moving in either direction. I know that it can be difficult in a large company with open source. Purely licensing can be tough for people to understand. I think the GPL is really problematic for that and it really causes a lot of people to steer away from open source, because no one really knows what the GPL is going to do with that whole redistribution clause. That is a big chunk of the reason why we are offering under the MIT license. Even putting that aside, the concern that there is with bundled licenses inside of your applications is pretty

strong from people who know what they are doing at the enterprise level. We certainly use third-party libraries and such stuff within Concrete5. 'But have we done a good job on making sure that all of the licensing inside of our application is actually ok for the enterprise?' Of course we have. That is why you have bought enterprise support; that we will look in there and continue to make sure it always is. I think what you will see is for things that can't really work well as SaaS there will be a continued push to open source. Not because it is cheap, just because it feels like a more honest way of doing business when you know you are going to have to change and owning the core product anyways. I think that you will also see, for things that you don't need, more and more Software as a Service. The traditional model where you charge a big license fee up front and then an ongoing maintenance fee, while not being able to look what is under the hood, that is will become hard, because that is going to be replaced with SaaS.

You said to be working under the MIT license. Do you have problems with that; e. g. people will just resell your software?

Yeah, that's annoying, but what are you going to do? Where we stand is that it is difficult to believe that they are going to do a better job than we are. If we can't do a better job at it, then what are we doing here; we wrote the damn thing. We do see people repackaging Concrete5 and doing whatever they think they need to do with it. If they can make money doing that, good for them. Don't ask me to help, but go for it. When we do run into those, we typically see that they are pretty punkie groups, so I am not filled with fear too much yet. The product is MIT license, but the brand is certainly not, so you have to go build a community, everything else we have done, you have to make that work. The product is really the beginning and not the end. We have built features into the latest version of Concrete5 to do white labeling and approve the fashion that we like. This way you can turn it on and replace our logo with yours and then you get a little powered by Concrete5 label in the toolbar. To bring this back to the channel partners argument, as we continue to go forward, we add more features to our marketplace and to our enterprise stuff. 'But if you want to do it your own way, that is fine. But if you want to own your client, and that is fair as you convinced that client they need a website and you walked through the whole process, we don't want them to call us with questions. 'That is for you, small agency over there.' So it makes perfect sense to say that it is your own content management system that happens to be powered with Concrete5. That is the way I like to see it happen, but is it in my interest to go sue people who are doing it in another way, even if I had the power? No, it is a waste of time and energy. I think this comes down to the MIT and GPL licenses, like how does freedom feels like to you? This is tough and there are a lot of heated debates

between MIT and GPL about this. The way it adds up to me is that GPL is really wrong, really confusing and no one is going to court to defend it so no one can really tell you exactly what will happen. And it really treats this redistribution freedom problem as if freedom were this really, really fragile flower that has to be protected at all costs. I don't think of freedom that way. When you say 'hey, this is free, do whatever you want with it, just don't sue me', it is a dandy line and that thing is going to grow forever. If anything, you are going to have a hard time controlling it once you get it passed this stage. That is how freedom feels to me. Like in the Middle East, once you have planted a seed, it is hard to control, but I don't think you need to protect it, I believe you just have to make it happen in the first place and it will protect itself. That is kind of the underlying philosophy you need to have if you are going to go MIT versus GPL

What are the main evolutions you have noticed in the market the last couple of years?

In the market of just CMS, from my perspective, I have seen prices for the commercial stuff go down. I have seen interest in adopting purely commercial products wane. I have seen more open source things that have a commercial end, so where they get you with: 'here is the free product. If you need the enterprise one, I will do what you need.' I think you are starting to see more of that. Everyone will always rebuild their website every two years or so. I think it is getting more difficult to justify ongoing five and six figure license fees for a commercial tool set to do that, when you know you are going to rebuild it again three years from now and you weren't that happy with the tool in the first place. We have seen a few large companies just completely leaving commercial stuff behind and just choosing open source solutions for this.

6. Transcript interview – Bart De Waele, Fork CMS

Participants:

Reinout Denys	Ghent University
Gert Vanhaverbeke	Ghent University
Bart De Waele	Fork CMS
Location:	Voorhavenlaan 31/003
	9000 Gent
	Belgium
Date:	Friday, 10 th February 2012
Time:	15.00 - 16.00

Waar situeren jullie zich als je de markt van CMS'en bekijkt?

Je kan de markt onderverdelen in verschillende stukken. Enerzijds heb je de grote enterprise *proprietary* systems: Ektron, Tridion,... Anderzijds zijn er de *open source* systemen, die zijn meestal kleiner en daar zitten wij. Daarbinnen heb je bijvoorbeeld Drupal, Wordpress (vaker voor kleinere dingen) en Expression Engine. Wij hebben vanalles getest, maar onze zin niet gevonden, dus dan besloten we om ons eigen CMS te maken. Bijna elk bureau heeft in Belgie zijn eigen CMS gemaakt 3 tot 5 jaar geleden. Fork CMS hebben we in september 2010 *open source* gemaakt. Veel webbureaus in Belgie zijn nu naar Drupal aan het switchen. Wij hebben een beetje een tegendraadse richting gekozen.

Waarom hebben jullie gekozen voor open source?

Er waren verschillende redenen. Eers en vooral geloven we zeer sterk in de kracht van *open source*. Alles wat we hier doen, ligt ook in die lijn. We gebruiken zelf veel *open source* software. De tweede reden had te maken met commercieel en *business* denken. Commercieel was de druk vanuit de klanten groter en groter. Ze begonnen meer en meer bang te zijn van een *vendor lock-in*. Verder was er nog de *business*-logica. We zijn wel een groot bedrijf, maar de kost om enkel met je eigen systeem bij te blijven is zeer hoog.

Welke zaken onderscheidt Fork CMS van ander CMS'en?
Er waren zeer specifieke redenen waarom we met ons eigen systeem zijn gestart. Ten eerste waren we heel erg bezig met 'vindbaarheid' en SEO (*search engine optimization*), en dat zat zeker 3 jaar geleden niet goed bij de meeste CMS'en. Het tweede punt is het feit dat we zeer harde *usability* freaks zijn. Verder was ook het gebruik van de *backend* van de meeste systemen een hel. De derde reden is dat wij CMS niet vanuit techniciteit benaderen, maar wij willen marketing, communicatie-gericht werken. We vertrekken van de wensen van een marketing- en/of communicatiemanager. Dat is een groot verschil met de meeste CMS'en. Zij zetten die persoon centraal.

In het core-team van fork, die de kernbeslissingen nemen, daar zitten zowel usability engineers, developers als designers in samen. Bij drupal zijn het vooral developers. Dat is voor en door developers gemaakt.

Zijn er volgens jou verschillen tussen grote en kleine spelers, in de manier waarop ze naar de markt toe gaan?

Enterprise levels hebben een volledig intern team. Kleine spelers overleven daar niet. Bij *open source* zie je meer en meer dat het zich centraliseert rond een commerciële entiteit die het voornamelijk trekt. Bijvoorbeeld bij Drupal heb je Acquia, en bij Wordpress heb je Automatic. Het zuivere *open source* model voor CMS, waar er geen mensen zijn die er een loon uit halen, dat is vaak niet meer overleefbaar.

Fork denkt erover van de BSD licentie te gaan gebruiken. Niet GPL omdat ze de *core* open willen houden. Bij BSD laten ze de mogelijkheid om modules proprietair te houden. We hebben binnen Wijs ook een aantal modules, die we ook niet willen open maken omdat we er competitief voordeel mee hebben. Er zijn al 15tal bedrijven in Belgie die met Fork werken. We gaan ook langzaam aan die modules die vandaag nog een competitief voordeel leveren in de toekomst misschien releasen.

Is er een specifiek klantensegment waar Fork zich op richt?

We richten ons op marketing-communicatie. Dat is over sectoren heen, maar wel gericht op de marketing/communicatie manager. Wij zien een website als een tool om *leads* te genereren en om aan marketing en *sales* te doen.

Welke waarde wordt er geleverd aan de klant?

Voor Netlash bSeen is het een interne *productivity tool*: we kunnen zelf sneller websites maken. Het is ook voor ons een positioneringstool, omdat de meesten Drupal gebruiken.

De belangrijkste waarden zijn de sterke *usability* en het gebruik als marketing- en communicatietool.

Wordt de community op een bepaalde manier georganiseerd?

Er is een centraal team dat binnen die core alle beslissingen neemt. Via Github worden er commits gedaan. Github is een versiebeheersysteem. Je kan dat zien als een website waarop je code kan downloaden en verbeteringen kan doorsturen, en dan worden die verbeteringen door het coreteam goedgekeurd. Verbeteringen worden dus gesuggereerd.

Die core-groep, kan je die zien als een voltijdse job? Is dat een klein bedrijfje?

Over de verschillende bedrijven die met Fork samenwerken heen, werken er verschillende developers aan Fork. Hier bij Wijs zitten we met één voltijds equivalent dat volledig aan Fork wordt gespendeerd. Wijs heeft er belang bij dat Fork evolueert omdat ze dat als tool gebruiken.

Zijn er plannen om een soort enterprise version van Fork uit te brengen?

Er is een plan om een stukje *appstore* te maken. Waar iedereen, zowel wij als anderen, hun proprietaire modules te koop kunnen aanbieden. Een module zou dan bijvoorbeeld 200 euro per download kunnen kosten. Die appstore gaat door het coreteam gereguleerd worden. Dat is het plan voor binnen een jaar. Eerst is er 2 jaar aan de *community* gewerkt, zonder business model. Op het moment dat die *appstore* er is, is het de bedoeling dat er een klein percentage op gaat gepakt worden. Met het idee dat er dan een soort eigen entiteit kan ontstaan die dan die opbrengst gebruikt om mensen fulltime op Fork te laten werken. Dit zou dan het businessmodel kunnen zijn. Het is niet de bedoeling er een winstgevend model van te maken. Break-even zou goed zijn.

Zijn er plannen met Fork om er een soort advertising tool van te maken?

Juist omdat we andere bureaus willen aanmoedigen om er in te stappen, willen we het niet teveel Wijs-branded maken.

Hoe is de evolutie van Fork in bedrijven die het opnemen en hoe kan dit verder gaan?

Ik zie daar een grote evolutie in. Plots duikt er nu een Chinese *community* op en heel de backend is nu vertaald in het Chinees. Hetzelfde geldt voor een Finse, Roemeense en Poolse *community*. Overal duiken er dus mini-communities op. We proberen die wel in de gaten te houden, ondersteuning te geven, en een stukje uit te recruteren voor het coreteam. Vanuit de Chinese *community* zijn er ook al aanpassingen gekomen, die dan moeten goedgekeurd worden.

Welke mogelijkheden zie je nog in Fork en de sector van content-managementsystemen in het algemeen?

Er zijn mogelijkheden genoeg. Bijvoorbeeld zoals een bedrijf als Acquia. Die bieden een *hosted* version, een SaaS version aan, die gratis is, maar waar zij dan advertenties op hebben. Die leveren diensten en services aan een bedrijf dat Drupal implementeert en zij bieden dan expert-services aan 'hoe kan je dat optimaliseren'. Zij bieden ook *security tools* aan. Er is dus een hele *community* van mensen die Drupal gebruiken, en zij gaan dan op die *community* werken.

Wordt er reclame gemaakt voor Fork?

Voorlopig beperken we ons tot developers en studenten. We stappen daar effectief naartoe. We promoten en ondersteunen dat. We proberen dat op een moderne marketingmanier op de markt te brengen (door gebruik van social media). Nu richten we ons eerst op de bouwers, dat zorgt voor die eerste laag in het bouwen van een *community*.

Hoe gaat Fork om met concurrentie?

Voornamelijk door differentiatie op marketing en communicatie. Er zijn 2 soorten klanten. Enerzijds is er een groep klanten die zegt "Ik wil iets van deze aard, *make it work*." Anderzijds is er een groep klanten die zegt "Wij willen een Drupal (of ander) systeem." Dat doen wij niet meer. Wij proberen deze mensen dan te overtuigen om Fork te gebruiken.

CMS gaat niet noodzakelijk alleen over content an sich aan de buitenkant. Het gaat ook over gebruiksgemak, waarmee je SEO-technische ingrepen gaat doen. Stel, je wil Google Analytics implementeren, dan volstaat het om die code 1 keer te plakken, en dan gaat dat over heel de site. Er is zo een SEO-tabje in de pagina's.

Verder zit er ook een mailmotor in het CMS. Dankzij de integratie van Google Analytics kan je zien waar je bezoekers vandaan komen. Dat wordt bedoeld met die marketing/communicatie tool. De meeste CMS'en zijn bedoeld om content te publishen, terwijl we met Fork leads willen genereren en informatie verzamelen door het publishen van content.

Is het een trend dat CMS'en meer op iets specifieks beginnen focussen?

Er is een trend van verschuiving naar *open source*, en naar integratie van *analytics*, email of social marketing. Er is verder ook een grote verschuiving naar aggregatie in plaats van publishing (vb. Tumblr, wat een voorbeeld is van republishing). Dit is een totaal andere aanpak dan het zelf maken van content. Hier gaat men content verzamelen.

Een trend in *open source* is iets dat volledig *open source* begint, maar dan naar een commerciële strategie evolueert die wel rond een *open source* strategie gebouwd is. Die verdienen er geld mee, maar moeten ook moeite blijven steken in *open source*.

7. Transcript interview – Dean Smith, Hannon Hill

Participants:

Reinout Denys	Ghent University
Gert Vanhaverbeke	Ghent University
Dean Smith	Hannon Hill
Location:	Internet call, Skype
Date:	Wednesday, 25 th April 2012
Time:	13.30 - 14.00 (EDT time zone)

Questionnaire

What do you believe, are the advantages of a closed source CMS over an open source one? What are the disadvantages?

Here is some helpful that you might find helpful as to why Cascade Server is a better fit over Open Source CMS products.

Buying a commercial CMS tool like Cascade Server offers a number of distinct advantages, not the least of which is commercial support and well-defined service level agreements.

- Cascade Server contains built in "out of the box" functionality...you have to customize the open source product to meet your specific needs at your organizations which takes up even more time and resources.
- Cascade Server will be faster to implement and our documentation and training is significantly stronger.
- Our main strength is Higher Education. Hannon Hill has been working with Higher Ed institutions for over 11 years and the Cascade Server was actually created with Higher Ed organizations in mind.
- Open source products are not created with Higher Ed. in mind.
- Cascade Server is being utilized campus wide by over 175 prominent Higher Ed organizations

- Numerous product features have actually come directly from specific comments and suggestions from our Higher Ed clients.
- We offer quite a few number of resources geared around our Higher Ed. clients that we feel lack on the Open Source CMS side
- We offer a full Knowledgebase (hannonhill.com/kb) Client Community Site (help.hannonhill.com) Annual User Conference, Higher Ed. Litserv, and a free web marketing took called Spectate (www.spectate.com)
- Spectate was built from the ground up based off of the feedback from our Higher Ed. community.
- We also have been praised for our outstanding support staff on cmswatch.com
- Cascade Server has the ability to hook in with any 3rd party "Best of Breed" tools through our SOAP based web-services API.
- We also allow you to use any type of custom scripting within the Cascade Server.
- Cascade Server is put in place as an investment. As your site grows over the years, you will have a solid product that will only get better over time. Since we are Higher Ed. driven and Higher Ed. focused, our clients feel secure that Cascade Server will be their long term needs. This is why we have such a high client retention rate.
- Cascade Server can easily accomplish multiple site management.
- We're technology neutral and can publish content out as any language and to any web server using any technology

We find that the reason that many users originally try an open source solution is price, which is understandable if you don't have the budget. An open source CMS will be significantly cheaper than a commercial CMS, but as you know not everything is "free". Product support, documentation, and user training are often subject to the whims of volunteer developers. As a result, there is no brand name or customer service department to offer assurances or assistance in maintaining CMS stability and security and the product implementation may take considerably longer.

Does 'your firm' aim for a specific target segment? What are its key differentiation points, order winners or order qualifiers?

Our main client base is surround around Higher Ed. institutions:

To date, over 175 Colleges and Universities use Cascade Server campus wide: http://www.hannonhill.com/customers/industries/education.html

Open source generally works with a community. Do you have a sort of developers community for the evolution of the product? Are there any other partnerships or important relations concerning research development?

Our <u>Cascade Server Community</u> is a key component of our success. As a company, Hannon Hill places great emphasis on being transparent, providing open access to our resources (including our customers, our prospects, the Higher Ed. community, and our competitors), and fostering our user community. **Help Forum.** Everybody can search our help forum for articles and FAQ, ask questions, and participate in discussions with our Product Engineers, Support Team, Trainers, Services Developers, and of course, other Cascade Server users.

Knowledge Base. Our KB is another resource open to anybody. You'll find a plethora of tutorials, videos, and both high-level and in-depth explanations of concepts and features.

Code-sharing. Our Services, Support, and Engineering teams are very open and not afraid to share their code with our customers. Therefore, we created a GITHUB repository, where anyone can download examples (including, but not limited to, migration tools, XSLT, Velocity, Publish Triggers, and Web Services). Of course, we also encourage our customers to share their code as well.

Hi-Ed Listserv. The Hi-Ed Listserv is a fantastic resource maintained by our large Higher Education community. If you're looking for tips, additional code examples, or you just want to compare notes with other organizations, be sure to subscribe to the listserv.

Webinars. Hannon Hill has been offering webinars for years. Not only do we offer webinars on best practices, tips and tricks, and product features, but our Cascade Server community also continues to step up to the plate and host webinars on how they leverage the power of Cascade. It's a true community effort, and something that we are looking forward to providing on an ongoing basis.

Idea Exchange. User experience and our customers' needs have always been our number one priority, which is why several years ago, we implemented our idea exchange, where current customers can discuss their ideas and suggestions for Cascade Server, and vote up the features that are most important to them. Many of the requested features have already made it into the product, and we encourage to keep the ideas coming, as they always play a vital role in our product road map.

User Conference. Our annual user conference is the highlight of our year, since we get to interact with our customers face to face, discuss new ideas, learn from each other, and just have a great time. And, as you would expect from us, we make our conference videos available to anyone.

Regional User Groups. Our customers have formed regional user groups, which is a great opportunity to network and brainstorm with higher education organization in your area.

How do you manage your relationship with clients? Is there a direct contact, or do you work with intermediary partners?

Yes, there is a direct contact. We also have a dedicated Client Advocate whose primary job it is to maintain relationships with our clients.

How are revenues generated? What kind of different streams are there? (Licensing, maintenance,...?) How is pricing done?

Pricing can be found here: http://www.hannonhill.com/products/pricing.html

Finally, are there any general evolutions to be expected in the CMS market?

A major trend at the moment is an increase in web marketing. Google recently updated their search algorithm putting more emphasis on content engagement, social media, and freshness factor.

This is why we built Spectate from the ground up. Spectate is included free of charge with the purchase of a Cascade Server License and Spectate is a compliment to Cascade Server on campus.

Here are some of the ways in which Spectate can help you optimize your marketing:

• SEO scoring: Spectate gives you an average SEO score for your website, as well as a detailed SEO score for each individual page. It checks for title tags (and their lengths), meta descriptions, links, h1 headings, broken links, HTML size, load time, and number of include files, providing you with a way to immediate spot potential problem areas with regard to SEO. In addition, Spectate shows you the load times for all of your pages and alerts you to problem links.

- Keyword tracking and analysis: Spectate allows you to optimize your keywords by showing you the search volume, cost per click, and ranking difficulty for each of your keywords. In addition, you can see the keyword density for your pages and blog post, which helps you optimize your content and prevent keyword stuffing.
- Social Media platform: Spectate provides you with a comprehensive social media platform that allows you to manage and track as many Facebook, Twitter, and LinkedIn accounts as you need. You can send and schedule posts to all of those social media channels, automatically create custom tracked links, and monitor social mentions and respond to them from within the system (you can even assign specific users to respond to a social conversation, thus streamlining your process). In addition, Spectate allows you to track your social media goals and objectives, calculate your social media ROI, generate social analytics reports, and track social sentiment for your organization. Furthermore, via RSS Broadcasts, Spectate can automatically promote your new content for you.
- Competitor tracking: Spectate delivers an analysis of how your organization's website compares to your competitors' sites in terms of indexed pages, inbound links, and PageRank over time.
- Identifying Visitor Companies: Spectate doesn't just show you the IP addresses of visitors on your site, but it also identifies visitor companies, which allows you to see when a competitor or a media outlet is on your site. You can track their behavior and even set up email alerts.
- Landing Pages and Forms: With Spectate, you can create a landing page with an associated form with just a few clicks. Our form builder is very easy to use, but very sophisticated from a functionality standpoint. For instance, it allows you to set up progressive profiling, which means that if someone has already filled out the form before, you can ask them for different information the second time. Form submissions are captured within the system and also be submitted to your database. Email alerts and auto-responders can be set up as well.
- Reporting: Spectate provides you with a variety of reports, including a 30 day comparison of your visitors and page views, inbound links, referral sources, search terms, and traffic sources. It also gives you a 24 hour performance snapshot (which includes the most active pages, most active social posts, top searches, etc.).

Furthermore, the system shows you when a visitor comes to your site via a new search term.

Interview

A first question about the pricing. You offer both a standard version versus the enterprise version, and also propose a hosted license. Which is most often used or chosen?

I'll just give you some background information regarding the popularity and the difference between the licenses. Based on our 11 years of experience we can conclude that the standard single-CPU license is the most popular option. Up to 75 percent of our clients will actually choose this license running this on a Cascade Server with 1 dual-core or 1 quadcore processor. The nice thing about our licenses is that since Cascade Server is a true enterprise-level CMS, unlimited users and unlimited sites are possible at no charge. This allows our customers to host all their users, all their sites, even the entire universitywebsite all within one CMS license.

That license cost consists of a one-time cost of 40 000 dollars and a yearly recurring maintenance and support cost of 8.000 \$ (20%) This explains why this single CPU license is so popular.

Next to the single-CPU we also have the enterprise license. Both are licenses that schools will purchase and actually install on-site at their location. Some schools are very large with over 50.000 students. Bigger Schools opt for the larger license; our enterprise license. This will actually deliver a better performance, running Cascade on 2 dual-core or 2 quad-core processors.

Next to these two above options we also offer a fairly new possibility of a hosted license. It's been around now for a little bit over one year and we offer this hosted standard license at a price of 2500 dollars a month. It becomes rapidly more and more popular. Many schools want to cut out the IT department and will allow a CMS-company to host this for them. It gives more ownership of the CMS to the marketing department instead of the ITdepartment.

It allows our company to manually update for our clients. In the hosted license, we handle the installation as well as all the future upgrades. We manage both the Operating system as the daily backups... That's in a nutshell pretty much what we provide. So that's becoming more and more popular. In the last 2 months, we signed 7 new clients, and 3 of those clients went for the hosted license.

You explained that the hosted license is quite new. How did you prepare yourself as a company to offer that to the client?

We're working with specifically higher-education institutions. We want to make sure that we are well informed and we're on top of the feedback we receive from our clients. We concluded that more and more clients wanted to their IT department to focus more on highly technical tasks on campus and decide for themselves about their CMS processes; they didn't want other departments to be involved in their CMS processes. That's primarily why we have seen the push towards the hosted license. Until now many CMS companies in the market are also offering Software as a Service (SaaS). But the problem with SaaS is that many clients won't have their own dedicated server but may be on one specific instance of Cascade. So we offer every customer their own dedicated instance and only they will have access to it.

So your hosted version isn't SaaS?

No it's not, it's a dedicated instance.

What advantages does this have over SaaS?

For example, an advantage is if a client doesn't want an upgrade. They can choose not to. In SaaS they can't. If other clients want to upgrade you as the provider have to make the decision to upgrade the software or not. Now each client on a dedicated hosted instance can really decide the way he wants to move forward. From the perspective of Hannon Hill, offering this kind of SaaS solution was really the best decision to move forward.

Are there any kinds of other revenue streams?

Next to the licenses we also provide services. We have a full in-house professional services team at our corporate office. This section of our office is dedicated to accomplish integration projects for our clients. We have a project called Quickstart package. This is aimed at new schools to get started and up and running with Cascade as quickly as possible. This package includes two days of training and one hundred hours of professional services. Over the last two years alone, we've accomplished over 50 Quickstart packages for our clients. So it's really a popular option. In addition to the Quickstart package, we can accomplish anything for our clients. Whether it is a full site integration or an integration of a section of their site. Maybe an integration of a calendar.

We have the expertise. We do not accomplish however, design services. We have many partners that will design websites for higher-education institutions. We can integrate those new designs, but in terms of the actual design market, we recommend having the school work with a vendor locally or one of our partners. The way that Cascade handles design is very straightforward.

On the other side, what are the main costs for your customers?

The Quickstart package (hundreds hours service and the two days of training) will cost 22000 dollars. Services hours can actually be purchased in blocks of 15 hours at a rate of 175 dollars an hour. Primarily, after training, a school has all the information to fully manage their entire site with Cascade.

And for your own company, what are the main costs for you? Is that the hosting part or the maintenance or...?

Hosting and of course staff. We have a full sales staff, a marketing staff and a client advocate team, which sole job is to keep our clients happy. Their job is to check in with our clients on a quarterly basis. That's something we found to be very, very successful. So we have within our staff someone who is almost "the advocate for our client" and takes care that they stay happy. It assures that those renewals are coming every year. That's really helping to the success of our company and pays off in the long term.

You talked about partner companies. How does one become a partner?

We have two partnership agreement documents. They are very straightforward. A partner will receive a 20 percent discount on the Cascade Server license. So the partner can actually mark-up the Cascade Server any way he likes. That's if they want to resell Cascade Server. As for other partners, like of our design partners, companies like m-stoner or company like BarkleyREI. Those companies work specifically with schools who want to redesign their website. Many schools come to our partners to integrate a CMS and have a design at the same time. So our partner companies will refer business to us and we will also refer business to them. If a school comes to us and they come for design work, we will definitely send them to a partner. If a partner company is talking to a school wanting a CMS, they sometimes send them our way.

Are there also other sorts of partners or do you only have design partners?

We do, in addition to design partners, we also have strategic kind of partners that are very familiar with working with Cascade Server. We have a partner called Cascade Factory,

whose sole job is working directly with Cascade. They may work within a certain geographical area of the United States, for instance Texas. We have many organizations around the country that benefit from the flexibility that Cascade offers, but the majority of our clients are higher-education institutions.

Most of your sales, are they done directly or via the partners?

Definitely directly. We have a couple of deals brought to us through partners, but the vast majority is through direct sales (cold calling) and our marketing efforts.

So what do you think are the most important resources and assets that your company has to have?

What's great about our product at Hannon Hill is that we focus most of our time and efforts on just one product. So our clients know that all the features and functionality added over time, are always going to be added in the best interest of our clients. We're not allocating resources to other products and industries. That's been very beneficial to us. We've also been branching off, well not really branching off, but we offer a free product that is available to our clients. It's a free web-marketing product called Spectate. That product was built from the ground up, based on the feedback we got from our higher-education clients. Many schools around the country are not only looking to have a CMS to manage their campus, but they want a CMS in the place to be able to track their marketing records and marketing success. So more and more schools are busy with Facebook and Twitter and LinkedIn. The nice thing about Spectate is that it will allow you to provide a centralized platform for a marketing member tracking all their marketing activities. So that's what we're seeing. More trends are heading towards social media, the management of it and analytics and data.

Is it your intention to create via Spectate a sort of new customer segment or is it's still aiming for higher-education institutions?

It can be for anybody, but we of course are offering Spectate to all of our clients free of charge. So think of Spectate just as a complement to Cascade on campus. So Cascade will help you manage the core website content, Cascade will allow you to log in to the CMS and manage the core content. If you have an excellent CMS in place like Cascade and you're out there managing your site, having all that effort put in your site; what's the point if you can't harvest the analytics and data. So that's why we built Spectate. To be able to look if your marketing efforts are paying off. So as an example, Spectate can post news and events from the Spectate dashboard, send it out to Facebook and Twitter and Spectate will create

tracked links. So every time someone clicks on it, you can actually see the data in Spectate, and that's really going to show if your marketing efforts are paying off. So we've actually decided to integrate products together. So a part of our next release, we create a Spectate connector within Cascade Server. So you'll be able to create Spectate tracked links from within Cascade Server.

When looking at the competition, is there another product which offers a product like you?

There are many companies out there that offer similar products. But really what makes us different from a CMS-company perspective, is that we focus a hundred percent of our time and efforts on higher-education institutions and the way that our software is built, we can easily allow to schools share assets across different sites, and also with the addition of Spectate. Having a free web-marketing tool, that is something that is taking content management to the next level. We like to call it content empowerment. Being able to empower your marketing teams to manage the full content life setting.

Are there any open source software solutions to that?

In terms of OS, we come across schools that will use Drupal or Joomla. But if you look at any OS CMS, they will all aim at different industries. I don't believe there is a highereducation centric OS CMS product in the market right now. I'm not sure how successful that would be. The market is quite saturated right now, with a number of commercial higher-education CMS. For the moment most open source competition is from Drupal or Joomla. We have seen school migrating away from Drupal to Cascade Server. The problem that schools will find using an OS CMS is that OS products often don't offer commercial support. They are looking for a CMS company that is specifically geared for highereducation.

Do you look at the open source solutions and take some things from their functionality?

Not really, we already have a very large higher-education client base; we have over 175 higher-education clients now. Clients have a voice in what they want to see added in a future release. So we really base ourselves on feedback from our clients. We find that many CMS companies follow us when we release a feature, they will copy it. We also investigate look at our competitors, we look at what features are being added, so we can be competitive, but overall the majority of our add-on features are based on the feedback we get from our clients.

How do you manage that client feedback?

We have a client community site that is put up. That allows clients to provide feedback and suggestions, they can start discussion, and they can share code.

Does someone manage that community?

Our support staff manages that community. They will interact with clients looking for answers.

Do you think there are some other evolutions next to the analytics being incorporated?

Yes, web marketing is becoming very popular. That's where we foresee the future for content management. Also more and more schools are moving to responsive designs. For instance when you shrink your browsers window, you see that the site will adapt itself. This is strengthened by the evolution of tablets and smartphones. The popularity of mobile devices is only going to increase, so that's important.

Is there anything we missed that is important to your BMS?

It's important for our company to hire individuals that are positive, self-starting and supportive. Part of our success is being very thorough in our selection process. It is only by hiring the right personnel with the correct characteristics, that your company will evolve and grow together with your clients.

8. Transcript interview – Sam Keller, Oxcyon (Centralpoint CMS) (part 1)

Participants:Reinout DenysGhent UniversityGert VanhaverbekeGhent UniversitySam KellerOxcyonLocation:Internet call, SkypeDate:Thursday, 26th April 2012Time:11.00 - 11.30 (EDT time zone)

Preliminary information

For your thesis, the contrast between open source and closed offer many opportunities of discussion...besides the obvious (closed is more finished solution than open, requiring development)....there is a lot under the hood.

Some points for you and your partner's consideration which would help to make your thesis truly unique...

- Open Source is touted by the developers themselves, offering a 'free' price tag to the company (in exchange) for their job security....as the end result will be dependent on the 'person' coding it. To most companies, this is a very attractive offer, until they realize that they didn't buy the bread they needed, instead they bought the 'baker', who is required for a long time, to keep the bread coming, and hope his recipe is good (to use a metaphor).
- Closed is often ridiculous, demanding license fees per user, per server, per disk usage, and literally operates as a metered tool.
- Our approach is different, we offer the source code for the 'finished' product....meaning no programmer is needed to finish the job, and the client maintains ownership rights, and moreover the ability to hyper customize themselves....This requires a remote update (so clients may install locally, yet inherit new tools by us, if under maintenance), yet provides for local custom development around 'virtually' everything 'except' those elements designed to protect clients 'from

themselves' (yes, from themselves). Often in an open source environment, everyone assumes that the architecture of the programmer is accurate...when it might not be. There are foundational considerations which must be met to preserve integrity. Our technology has identified those elements (audience, roles & taxonomy) and those things may NOT be changed by our hyper customized (local) clients....in this way they can play and customize but cannot hurt themselves from myopia, or short sightedness. Ours is the only platform like this in the world, with all three (polyhierarchical classification rules) which cannot be changed from the architecture.

- As the old saying goes 'who is guarding the guards'.....well in this case I see our company (or technology) governing or guarding those who 'claim' status of architect, when they are merely web technologists.
- My favorite point.....'What is content?'....Too often companies regard content as webpages, documents to be managed by a 'document management system', and a multitude of other servers to do their part (think SharePoint or Active Directory) to manage 'just' authentication or single sign on, and then a bridge to the (disparate) content management system.....arcane thinking. Instead I like to say 'Content is a fancy, overused term for 'data'.' This said, documents (and their versioning), videos, podcasts, blogs, x-ray images, anything is content. To that end, since we need to manage the content (yes to a webpage) but also down to the granularities of 'Who can see what content' (DRM- Digital rights management), it is imperative for any 'sustainable' system, to include such things as Roles (filtration), Audience (filtration) and/or Taxonomy (filtration) for any singular 'record' (albeit an x-ray image, or policy document)....With this structured approach, we can now specify who may see which record, and content ends up encompassing 'everything' (per Mr. Bill Gates' comment, 'Content is King'....you may start to see it as I do). The research firms categorize Document Management systems (documentum) from basic content management tools (like Drupal or Joomla), and then again another sector of research is 'Authentication or single sign on)......Even the research firms are in the dark ages, and need to understand the power of 'unification'.
- I would suggest looking up the theoretical discussions on the topic of 'Knowledge Management' (a buzz term from the late 90s, not used as much today, as it seems blue sky). This is the era I come from.....Now, when you look at our product, its unique differentiators, and then evaluate the contrast of open and closed CMS systems (available today) you will see a huge differentiator, and our futuristic thinking.

Interview

What are the key differentiation points for Centralpoint? And who are your main competitors? What makes you different from the competitors?

Our primary competitors are the biggest vendors, simply because there is the most money available, in terms of the options and the size. Our largest competitors are MS SharePoint, IBM WebSphere and also Filenet. But we also have to compete with every small open source CMS in the market. Our value proposition falls squarely in the middle of your 4 pillars. On the one end there are, proprietary, very expensive, high end systems and on the other hand, there are these very cheap –sometimes free- open source systems. But these latter ones are not really for free since you require an architect to finish these products. These people working on open source platforms do often not have the vision, foresight and long term experience in which to develop a winning and lasting architecture.

Our strategy: Highest common denominator on one end, lowest common denominator on the other. When we serve a client, whether he is a small, medium size or big company, we are coming with all of the tools in the arsenal. With now already over 230 "out of the box" modules or tools, we have more than is necessary to satisfy all needs. Our architecture is extreme, meaning we planned it for a multinational company, but we can easily scale it down for the smallest common denominator. So the differentiator here is, instead of trying to start from the existing low level of the customer we believe the portal building content management building is clearly a top down process. We go with our product all the way figuring out where all the challenges are going to end up and bring it to the market in a modular way, scaled down to meet the smallest client.

What is your vision on the 'job security' point of view concerning open and closed source software vendors?

Interesting question: A lot of companies choose today for a content management system, a portal. But the next choice for open source is often done by their developers. , not because of the advantages an open source can offer but merely as a job security issue. Indeed their work will hyper customize the rough open source CMS into the business's needs. The minute that programmer touches the open source code and starts tweaking it, no one can fire this guy. He is the only one who can keep it alive.

So often we are up against the developers within the company who are against Centralpoint CMS. The reason is clear. With Centralpoint everyone can build his own portal, even a kid can do the job. So the position of the web developer is threatened. Thus he creates a major problem in this whole market. They like to reinvent over and over the same thing for every new client. Why are all these programmers required to reinvent the same thing. And furthermore, what are the credentials of these programmers to actually state that they know and understand the end-tier (entire) architecture and security. These people aren't those guys.

That paper on 'Horizontal propagation' from 2005, is this still relevant in the current CMS market?

It is still very relevant as will show following metaphor. In the early nineteen hundreds cars were built by hand. The maintenance of the cars was very difficult and the long term costs of the cars were ridiculous. And then came Henri Ford who used an assembly line to lower costs drastically. Since then the options of a vehicle became modular; when you wanted a better car, you could add options out of a seemingly endless list.

The assembly line is the only way to guarantee quality, it is the only way to guarantee a better architecture and it is the only way to guarantee low long term maintenance costs. This is the American way.

Today in IT, we seem to forget these lessons from the past: ship building during World War II, the automotive sector at the turn of the century. Nowadays in IT, building your own portal with free open source is like building your own car by hand. ". No, no, no, no. If you build that car, you will not want to drive it on the highway. It seems attractive because it is free but the developers that you need to build /maintain your portal will cost yearly 60 000 to 80 000 \$ per year minimum, maybe up to 150 000 \$/€ per year, to keep that so called free open source going. It doesn't make sense.

Do you think that horizontal propagation has become the new standard in the sector?

We are helping to make it the standard as it has not become the new standard yet. We see a new direction for all business. Let me use the same metaphor of the car industry. Only twenty years ago Ford had different manufacturing facilities for different models. They used a different chassis for each car. Now they realize that they can use the same chassis but that it needs to be adjustable, so that it can be used for different models.

What you are seeing is a modular thinking which still allows you to personalize. We are introducing the same model into CMS. It has not become the standard yet, and we are still considered visionary in this way, but it is clearly the future. Once the modular model has

been proven, which we think we have already, it is just a matter of time before one will stop using free open source CMS.

You have the patent on the horizontal propagation, is it not difficult to make it the standard in the industry when you are the only one having it? And how is competition coping with the fact that they realize they have to go that way, but that they have to look for an own solution?

We are not trying to make Horizontal Propagation the standard in the industry. But it is the standard for us. Centralpoint has the ability to create 'children-Centralpoints'. When we install Centralpoint at any one of our client's facilities around the world, it is an offspring of our parent installation. They talk towards each other. That child installation, in turn, can create grandchildren sites and great grandchildren sites, and on and on. Centralpoint is very core; it creates portals, which create portals, which create portals...

This business model allows us to create extremely quick solutions at clients. Because there is this parental relationship, if we decide to release a new module (may it be from own development or improvement upon request from a client), all of the offspring generations all around the world inherit en pull down those new features. This means that nobody is left behind; this means that obsolescence is no longer possible.

This is a clear advantage over the open source model where the developer builds customer solutions over many years. It becomes impossible to retro-actively update each client, as each one is lost in its own point in time. That is why most web development companies have difficulties. Although they can easily create many sites with open source, it becomes extremely difficult to update them continuously, in two, five or ten years it will all be garbage.

What is the difference between horizontal propagation and Software as a Service, because it has many similarities we believe?

No, no, not at all. Software as a Service (SaaS) assumes that your software is posted in a central location once it is finished. But content is valuable and sensitive. You cannot expect clients to have their valuable content stored somewhere in Nebraska. That is not going to work. The ability to install locally at the client site, and have them managed their content locally, is still software as a service, but of another kind. In the old days they called it ASP (Application Service Provider). Centralpoint is the ASP for ASPs. Centralpoint empowers organizations to launch their own Software as a Service, that's the difference. It is a subtle one, but an important one.

Do you see potential in SaaS in the CMS market as a lot of companies are implementing it?

Absolutely, but here is a down side. There are many SaaS applications which are tempting for a client to sign up for. Let's say that there is a web based job board for careers at a subscription fee of 300 \$ a month and immediately they have their online job board. It will do the job perfectly, but down the road, when those job listings have to integrate with some other reporting system, problems will start.

What I am claiming is that SaaS is only a temporary measure to fill a market need. Eventually organizations need housing at their own facility; eventually all information needs to interrelate. SaaS acts like a band aid to fix a wound rather than cure the disease. Yes there is money to be made, but the longevity of these models is questionable.

9. Transcript interview – Sam Keller, Oxcyon (Centralpoint CMS) (part 2)

Participants:

Reinout Denys	Ghent University
Gert Vanhaverbeke	Ghent University
Sam Keller	Oxcyon
Location:	Internet call, Skype
Date:	Monday, 30 th April 2012
Time:	11.30 - 12.30 (EDT time zone)

Can you explain your quote "who is guarding the guards?"?

It refers to my notion that we put too much trust in IT and IT technologists. So when we let the technologists select the platform, this becomes a potential dangerous scenario, where they select a platform that first meets their needs - job security- before the company's overall information management needs. It becomes "the blind leading the blind". Since the business owners do not know the technology they depend on the technologist. These often will take the company in a direction the company does not want to go in. By doing so, they remain in control.

In my experience developers want ultimately more control over the company than the business owner. To avoid this Centralpoint, was developed; to liberate organizations from being misled. That may sound crazy but it is true. This open source is really the worst; it is not properly architectured and unfinished. Just because a lot of people are using it, is no proof of their intelligence.

Is that what you mean with technocracy as well?

Yeah, it is the same thing with technocracy. Too many technologists are misleading the companies they work for.

Concerning the infrastructure pillar of your business model: Are there any partners involved with Oxcyon? What do they do?

Oxcyon exits now 11 year. Our partners relicense our technology for their local business verticals. So Centralpoint becomes a horizontal content management and portal framework, which has all of the best practices built in to it. Now it is up to our partners to take that horizontal solution and verticalize it. A school district or a university or hospital managing information, it is all the same. The things that differ are the types of content they manage, the types of audiences that that portal serves and the different roles of the people who need to get to it. For example for a school district the tasks and grades are the different types of information, the schools themselves are the audiences and the people accessing it will be the students, their parents or the faculty members.

Our partners translate those vertical types in the market. Some partners service health care, some service publishing, some service the scientific communities or attorneys or military and their only job is to define, in that vertical market segment, what are the types of content, what are the different roles we need to serve and what are the different audiences. Once they verticalize that solution for let say a big health care provider, now they just have to replicate that for every other health care provider in that industry.

Is there a sort of procedure companies go through to become a partner of Oxcyon?

They do; they must convince us about a new vertical or a vertical in their region that they have identified, which is the market potential for them to sublicense our technology; they must give us examples of their own expertise to help clients in their vertical' to modify Centralpoint to that customers' specific needs. At that point we can identify a partner opportunity. Depending on this vertical, they have to pay money for the privilege of having that vertical. In addition to that, every client they sign we receive a sublicense.

For instance, if you can start your own technology company in your region (Belgium), as there is nothing like this today, except maybe for SharePoint and Filenet. You can compete with the largest vendors in the world (IBM & MS), providing the local customers with a low cost alternative, and arguably a better architected solution. You can do that without paying any money to start that business. It just boils down to sales and presentations in front of those customers. Horizontal Propagation actually propagates horizontally and vertically. We are looking at how to carve up that new niche, so that that even a new technology company can start building and building and building on a recurring revenue base and building revenue for themselves and inadvertently for Oxcyon without big investments.

How are partner relationships maintained? Do you offer them training or other services?

Absolutely. Our partners are given full training in terms of everything they need to know to understand and work with the technology; everything they need to know in terms of verticalizing these solutions for their local market. We also provide them with samples of previously built sites so they can understand better. Furthermore we provide all documentation so they have from boiler plate license agreements up to PowerPoint presentations, to present the products successfully to their clients. In this way they can sign the new clients under their own license agreement and under the name of their company.

We take a very comprehensive approach and support these guys because we really want them to succeed. This creates for us the necessary revenues, but also justifies the whole concept of propagations. Also the interaction with all these different local market segments is very rewarding. It is what we call the voice of the customer (VOC). Their clients start telling them what is needed. If we believe that this might be of interest for the complete vertical we will develop that set of tools and then next week everyone's server updates, just like Microsoft update, and then they receive the new tools that they can take to market.

You talked about your partners relicensing your technology. Is there any way they can infringe your license?

No, it works like Microsoft Update. If you are running new versions of Microsoft, and have an invalid license of Microsoft, Microsoft, back in Washington, knows about this; there is a remote connection, even though these portals are housed e.g. in Germany. For that portal to be able to inherit new tools in the update process, every week, it is a key process. The same as with your desktop computer, when you install Windows 7, Win7 is always looking back to MS in order to control if there are new updates available. At the same time, there is also verification if the version is legitimate or not. We also have the ability to remotely meter the usage or deactivate any installation in the world.

I believe the most of your partners are some kind of middleman towards the end user. Are there also other partners like e.g. external developers?

Certainly, we have a lot of different partners and certainly also technology partners. There is the example of our partner for the vertical of animal health care space (for veterinarian); they have integrated our product to work with lab systems for animals. We have others that do the same kind of thing, related to health care technology. We are building more and more technology partnerships now. But my preference goes to the value-added reseller or sales group.

Do you also get your product to the end users directly or only indirectly by using the partners?

We do go also directly. For that we have our own sales force who will identify some strategic type opportunities. Sometimes clients contact us because they want to work directly with Oxcyon and not through a partner. We are avoiding this because we do not want to compete with our distributors. Vignette Company had a similar model many years ago and they ended up selling directly to clients to the point where they were competing with their own distributors. They ended up losing most of their distributors because of that. We believe there is a Bell curve for this business model, where there is a certain point when your distribution channels is bringing in enough business, you really should just focus on building a better product and not competing with them.

What are the most important resources/assets for your company?

We need to understand from our client, who they are serving, who is our client's client? What types of content do they need to manage for those various types of clients they serve? And finally, what specific roles, very specific kinds of clients do they serve? That is the most important thing to know when implementing our solution. We have a very simple formula really. What we bring to them is a scalable architecture that allows, regardless of what their audiences types are, to be serviced.

You talked about the voice of customer. Does this only happens via the partner network or is there a direct feedback system towards the end user?

We also maintain a direct feedback system; we call it our issue management system. I'm obsessed with centralization as I believe all things have to relate back to the central point. The issues that we manage work the very same way. We have the centralized issue management system, where all the requests, whether it comes from a partner or a client, funnel trough this one system so that we can analyze the chatter. It is very much like the CIA. By funneling all issues through one central hub, you will be able to see patterns, and your response to that intelligence can be very accurate.

Another example I like to use is the shipping company "the Federal Express". Fred Smith who started Federal Express, developed this main idea that if you have one central hub to ship everything through, instead of multiple shipping locations, you can run better operations, deliver things faster and everything would move with almost perfect efficiency. He still is the CEO and chairman of the board of Federal Express and has proven to the world that by funneling it all through one central point everything runs like clockwork. The minute you have more than one brain, or more than one hub it is a mess. The central theme here between FedEx and how to centralize content, is true for any of us. By centralizing the reporting of gauging of what customers are looking for, you can improve your working.

Most companies will structurally invest in R&D to make their product better", but who are they to do that? What are they going to do? Invest in new things the public might not want? That is crazy. What they should be doing, is listening to voice of customer so that it organically becomes their R&D.

Is this issue management system the only way you receive feedback from your customers or are there also other ways?

We try to funnel it all through that. Off course, we make phone calls, interviews and surveys, but ultimately the results of these interviews or surveys end up in the issue management system. If it is not in the issue management system it is basically lost and worthless.

Is there a kind of system where you measure trust and loyalty? Do you follow up your customers as well?

Absolutely, we follow up with our clients. After every issue that we close the user is able to survey how helpful that was and we register this in our issue management system, This is a byproduct of content management and it works like records in a portal You can compare it with Facebook. Clients can respond to everything and can like or dislike applications and tools. The incentives of our employees at Oxcyon are based on how many positives they receive. So when we are getting low marks of a client, and this happens also, that is all the more reason for us to listen more carefully to that voice of customer. Sometimes you get a thought leader client who sounds like he is crazy, but as I like to say "the only difference between an idiot and a genius is about 8 years." It is how long it takes for their wild dreams to come through.

Do you follow the competition in the market or do you just do your own thing?

We are very aware of our competition. Our primary competition is Microsoft SharePoint, more specifically SharePoint Portal Services, which is a server license to create multiple portals, identical to what we do. So, it is our intention to provide a better solution than SharePoint Portal Services. We also track FileNet and WebSphere. Because of WebSphere's platform is on a different database, we do not worry too much about them.

Are there any market evolutions you have seen the last couple of years, that are now standard? And are there any evolutions you see for the future, the coming years?

I do. The latest trend is what is called cloud computing. 'Put it all upon the cloud.' seems very attractive to most people. However, there is a danger to this trend, because the content that you are putting on the cloud is often content that belongs locally and highly secured.

We have seen this in many cases in the past with the internet where something will come up and it is very easy to participate, like Facebook or (My)Google. Everybody is participating, feeding information about yourself and your ideas. The next thing you know, is that government, your employer, your wife, your neighbor is tracking you and abusing the given information.

So, you have to think about the value of information and how cloud-based computing can really expose private information. I mean, if you were cheating on your girlfriend and your girlfriend posted a picture on her Facebook of you two at a party, you are busted. This kind of stuff happens every day. I don't think people are aware of the dangers and how private this should be. It is part of the reason that I do not participate to Facebook or LinkedIn. It seems ridiculous; why would a CEO of a technology company not participate? Well, that's because I see the dangers.

Are you hopping on the cloud train with Oxcyon or are you sticking with Horizontal Propagation?

Well, Horizontal Propagation remains our backbone. Off course, we do offer a hosted cloud-based solution, because we have to remain competitive. If the people are eating hot dogs, you have to serve hot dogs, even though they are not good for the people. But I would not recommend it to anybody.

In your software you work with different modules that you can choose and add. Who is developing these modules? Is that all from Oxcyon itself or is there a kind of external development?

We have both. We have our general release modules, which are horizontal. They are the Central Point modules as released by Oxcyon. Any module we will cater to any vertical. It replies to a horizontal demand that is universal to all verticals.

Alternatively at the vertical level, partners are welcome to customize those horizontal models to be very specifically named even at a field-by-field level for their vertical. (E.g. health care versus attorneys). In this way our partners have been building to add and create their own modules which are unique to only that vertical. Those modules only propagate downward; they do not come back to the hub.

How is pricing organized for the modules or the software product in general?

First of all, modules can be offered 'à la carte', like a Chinese menu. But I do not believe in that, because most of the times people prefer cheap and do not buy the modules they really need. So we prefer the buffet style; 'eat all you can '. So we want to give them all of the modules.

We have to predicate price in another way and ours is based on the size and structure of the portal. In this way a very small company might have a hundred or two hundred employees and needs only a very simple, basic portal. We will supply them with a singular portal license. This will cost maybe 25.000 \$. Alternatively we have portals (e.g. for school districts) which serve 10 children portals under it, so it would be a multiple factor of the number of sublicenses under it. The price goes up to 125.000 \$. Finally, there is also, what I call the big kahuna; a big customer who will need a 'Master Enterprise License' (comparable to MS SharePoint Portal Services) e.g. 3M or Ford Motor Company. The price with MS starts at 700.000 \$, ours is only half this amount. This multinational company can create under that server any number of portals that they want to. Each one of those portals can have any number of children and grandchildren of sites that it needs. This 'Master Enterprise License' is clearly the most expensive. . But remember, all our prices are for finished products. The product is built, developed, finished and completely done for that price

Why do you claim that pricing in closed source companies is often ridiculous?

Unlimited number of users, unlimited number of servers. No one should be punished to use something more. Image that you bought an Audi for 30 000 euro but if you drove it 300 000 miles you would have to pay 90 000 euro. If you invite five friends to ride in your car, there shouldn't be an extra fee. Everybody would understand this is ridiculous. Once you have bought the car you can do whatever you want with it. So by analogy one should

not be charged based on usage or number of servers. You have bought the product and can do with it what you want without being punished.

Which other different revenue streams are there, except for license fees?

That's about it. We do not get involved in any kind of advertising revenue share, nothing like that. It is just straight software license fee; if you want to use the software, here is the license to use it. That is it.

Is your service also included in this fee?

Yes. We always include service in our offering when we sell direct. Now, our partners, they may wish to choose to charge extra for custom services; they often provide consulting services, to come in and to study the organization and build a portal around that. The clients will pay them gladly for all those services. As already discussed I do not like that model for our business. But some of our partners love it. Central Point becomes a great way for them to sell all kinds of services to their clients. But on the other way Central Point make the customers think about their own organization. You would be surprised that most companies have not given that any thought at all. So really, if you think about 'how much help companies need to determine just that simple basic fact (who are you servicing, what are the content types, what are the roles)'. It is probably the most valuable service you can provide a company.

If you both, you and your partners, are offering services, doesn't that make you a competitor with your own partners?

Yes, it is. That is why we are selling less directly to clients. As mentioned before we only serve who look us up. These are mostly strategic opportunities where we have no partners. Like e.g., a US city government, a really big city from Florida, wanted to use our software for a lot of community services, so that all the citizens could pay their traffic bills online, etc. so we built that portal. Since we have no partners in this vertical our team here has done a good job reaching out to ten or fifteen different big cities in America and we are now signing two of them, who like our city portal. Now that we have sold a few licenses in this vertical it will be very, very easy for us to find a partner to serve this vertical. Now they have three bona fide examples, and we give them a running start. Yeah, we make some money there initially, but eventually we need to turn this over to someone who wants to just make that his business and then off they go.

What are the most important costs for your company?

Our largest single resource is also our biggest cost; this is our workforce which consists of developers, architects and sales people. Our biggest expenses are development and production management infrastructure. These are the people in systems which allow us to keep Central Point running remotely, manage those issues, respond to these issues, manage all the customer service elements or help us build new partnerships around the world.

Next to it we have to have a very large hosting infrastructure as well. That is a big bill, a big cost. But it is not my favorite, nor, would I say, the most important, because, you can host everywhere.

Let me finish with a metaphor to illustrate the differences between open and closed source companies:

If you look at the difference between a communal type of organization and a strong centralized federal empire. You will have to conclude that an economy tends to flounder in a communal system, but will thrive once it is more federated. This is a really interesting metaphor with open versus closed source. Open source is too loose, it is not controlled and it could more easily fail.

10. Transcript interview – Sjoerd van Elferen & Rob Gietema, Plone

Participants:

_	
Reinout Denys	Ghent University
Gert Vanhaverbeke	Ghent University
Sjoerd van Elferen	Plone, Four Digits
Rob Gietema	Plone, Four Digits
Location:	Willemsplein 44
	6811 KD Arnhem
	Netherlands
Date:	Monday, 27 th February 2012
Time:	13.00 - 14.30

Ik zit aan de ene kant als *board* member bij Plone *Foundation*. Anderzijds projectleider en *developer* bij *Four Digits*.

Eerste vraagje: voor wat dient Plone? Is dat een specifiek product die op een bepaald marktsegment richt?

Plone is een *enterprise ready CMS*. Dat betekent dat we ons vaak richten op MKB (KMO), MKB+ en groter. Als je een simpele website wilt maken is Plone eigenlijk een beetje overkill, maar het gebeurt wel eens. Ondertussen is Plone ook uitgegroeid tot een platform dat gebruikt wordt als een soort van framework, bv. videoplatform is erop gebouwd. Maar de core blijft een *enterprise ready CMS*. Bij *Four Digits* bewerken we de MKB+ en groter. We hebben kleine locale bedrijven tot de FBI of de NASA wat megabedrijven zijn. En daar leent onze toepasing zich ook goed voor, het komt pas goed tot zijn recht vanaf een bepaald niveau van gebruikersniveau en complexiteit.

Als grooste kenmerk wordt vak zijn nauwkeurigheid geroemd.

Het is inderdaad één van de grote kenmerken. Het is één van de veiligste pakketten en dat wordt wel aangetoond door de gebruikers, al daar zijn bijvoorbeeld de FBI Is er een verschil tussen productdifferentiatie, CMS'en die zich differentiëren naar een meer marketing CMS of intranet CMS ofzo, hoe gaan jullie daar dan mee om?

Wanneer wij een offerteproject hebben, waarbij we weten dat een SharePoint of aanverwante producten ook in de running zijn dan maken we sterkte/zwakte analyse van de producten in de running. We zetten dan zwaar in op het gedeelte waar Plone sterk is en de rest belichten we vaak minder. We gooien niet alles in de offerte, maar we kiezen wel uit. *Security* is altijd een groot pluspunt, naast een goed gebruikerssyteem waarmee je erg gemakkelijk bepaalde rollen kan aanduiden.

Zijn er bepaalde evoluties binnen CMS die je opmerkt (vb social media integratie)

Ja. Je kan in Plone een aantal modules zetten die dan bijvoorbeeld voor Twitter integratie zorgen. Die hebben we zelf ontwikkeld en een van onze collega's onderhoudt dat dan ook. De vraag ernaar is nog steeds erg groot. Eigenlijk is het tegenwoordig wel standaard voor overheid en overige instanties dat ze over een Twittermodule en/of over een Facebook "Like Button" beschikken Die dingen zie je nu overal opduiken

Zijn er dingen die nog op dit moment niet veel gevraagd zijn, maar die je op termijn wel ziet komen?

Ja er zijn al wel wat trends te herkennen . Geo-informatie begint nu een beetje te komen, Sociale media moeten steeds vaker geïntegreerd worden. Als je verder naar de toekomst kijkt , dan verwacht ik dat *real time collaboration* in CMS belangerijk zal worden. Dus met meerdere mensen aan één document werken. Met andere woorden dus wat Google Docs nu al een beetje doet. Alhoewel die vraag er nog niet echt is , ondersteunt Plone dit ondertusen al wel.

Worden de modules door de community aangeboden?

Je hebt een aantal manieren waarop we modules beschikbaar maken. Die Twittermodule bijvoorbeeld hebben wij zelf ontwikkeld (*Four Digits*.portal.Twitter) en blijft hier in beheer, maar we bieden ze wel aan als *open source*, zodat iedereen hiermee kan doen wat ze willen.

Daarnaast hebben we ook de zogenaamde *collective*. Daar kan je alles meedoen, code toevoegen, dingen toevoegen in tegenstelling tot het *Four Digits* pakket waar wij de dingen doen. Maar in theorie kunnen ze ook die code pakken, die aanpassen, en dat dan gebruiken. Het is volledig gratis, volledig door iedereen te gebruiken en aan te passen wat je wil.

En hoe denken jullie met Four Digits daar dan meer van te kunnen profiteren? Marketinggericht of qua naam ofzo, want dat staat wel in de naam van die module dan?

Ja, bij *python packages* is het wel gebruikelijk dat dat in de naamspace staat. Als er bijvoorbeeld 'collective' of 'Plone' staat,, dan snap je wel dat dit van de Plone *foundation* afkomstig is. Dat is wel een beetje marketing natuurlijk . Zo gebruiken we voor social media integration onze eigen *Four Digits* Twitter module, dat is natuurlijk wel een leuk marketing effect.

Als er meerdere modules tegelijk voor hetzelfde doel ontwikkeld worden (vb die Twitterapplicatie) blijven er dan meerdere bestaan of hoe evolueert dat?

Dat is heel afhankelijk van de functionaliteit. Bijvoorbeeld wij hebben een module ontwikkeld vanuit invalshoek X en iemand anders heeft het ontwikkeld vanuit invalshoek Y, dan kan het best zijn dat die jaren naast elkaar blijven bestaan, maar het kan ook best zijn van dat er op een bepaald moment door de partijen samengewerkt wordt en tot één module wordt samengevoegd. Als die partijen elkaar kennen zal dat natuurlijk sneller gaan.

Uiteindelijk is het gegeven van een community toch dat je niet opnieuw het wiel wilt uitvinden?

Uiteindelijk worden ze samengevoegd of blijft enkel de beste over. Bijvoorbeeld bij formulieren generatie tool, waren er ook eerst meerdere, maar nu blijft er maar ééntje over, en iedereen gebruikt dan die. Zo ontstaat het meestal, het begint met nieuwe functionaliteit, en dan zijn er verschillende invalshoeken, en dan als er veel gedeelde functionaliteiten zijn, dan proberen ze dat samen te doen.

Op termijn worden die modules dan opgenomen in de core van Plone?

Niet altijd, in principe heb je een soort driesplitsing. De idee van Plone is om zo *Mean en Lean* mogelijk te zijn. De core van Plone moet eigenlijk juist heel compact zijn. Daarnaast zijn er nu al héél wat modules die beschikbaar zijn. Tenslotte waar we nu naartoe willen is 'recommended by Plone' packages mee aanraden, die niet volledig door *core developers* ontwikkeld worden, maar die wel van dermate goeie kwaliteit zijn.

Wordt bij een zelfontwikkelde module, waarbij een bug wordt vastgesteld, de module aangepast?

Een tijd geleden was er een discussie op de mailinglijst waar ze zeiden dat Plone een *duocracy* was. De Plone *community* bestaat uit mensen die dingen doen, en als er een *bug* gevonden wordt, dan wordt er vaak door de ontdekker een *patch* geschreven. Dit wordt gemeld aan de *developers*, die het kunnen implementeren in een nieuwe versie. Natuurlijk is dit de ideale situatie . In de praktijk is de bug niet altijd heel makkelijk op te lossen en bovendien bestaat de *community* niet alleen uit *developers*, maar ook uit marketinggebruikers , of *user interface* gebruikers. Die zorgen niet voor patch maar zenden enkel een email naar ons en dan kunnen wij ernaar kijken of bijvoorbeeld bijvoegen in de issue tracker.

Dit laatste is fijn , omdat je dan 1 centrale meldpunt hebt. Maar aangezien het om *open source*, gaat is het dus niet zo dat die dan de volgende week al opgelost is. Als hebben alle Plone-bedrijven er wel baat bij dat het ticket snel wordt opgelost.

Die tickets komen bij de developers terecht?

Ja in principe wel, dat kan specifiek bij 1 *developer* zijn, maar in het principe blijft het gewoon in het systeem zitten. Er zijn *Plone tune-ups*, dat zijn korte *remode-sprints* waarbij je een bepaald onderwerp samen oppakt. Die *Plone tune-ups* zijn eigenlijk bedoeld om elke 2 weken met mensen over de hele wereld tegelijk te werken om bepaalde *bugs* te fixen. Dat wordt dan gecoördineerd door iemand. Die gaat dan kijken of een beginnende ontwikkelaar of echt *core-developer* vereist is en op basis daarvan gaat die tickets toewijzen ..

Hoe wordt dan beslist wie dat moet managen of coördineren ?

De *duocracy* speelt ook hier. Iemand neemt de beslissing dat er teveel tickets zijn en die gaat dan een *tune up* aansturen.

Dat is dan het developers team?

Neen dat kan iedereen zijn die vindt dat er iets moet gebeuren. Het hoeft zelfs niet noodzakelijk een *developer* te zijn dat dat doet.

Hoe staat open-source daartegenover en wat zijn de voor en nadelen. Wie ontwikkelt zich sneller, open of closed source en wat zijn de verschillen op vlak van ontwikkeling en functionaliteit? Een van die dingen die meteen in me naar boven komt is 'niet het wiel opnieuw uitvinden'. Als een ander bedrijf een nieuw CMS ontwikkelt, dan, vraag je je af waarom, het wiel bestaat al, is het dan een beter wiel ?.

Plone bestaat al 11 jaar, zolang wordt er al ontwikkeld met steeds meer *developers* en verdere doorontwikkeling. Dat heeft ons een grote breedte gegeven. We kunnen ondertussen een heleboel met Plone, want er zijn heel wat sterke functionaliteiten gebouwd. Je kan ze gebruiken, maar hoeft ze niet te gebruiken.

Bovendien heb je ook een veilige en erg stabiele *core* net omdat ze al zolang bestaat en verbeterd is en zoveel gebruikers heeft. Je moet al een flink bedrijf zijn om alle *developer* uren die er per jaar aan besteed worden, die te vergoeden, naar analogie met *closed source*.

Closed source lijkt ons wel meer gecentraliseerd. Bovendien pleit men voor een duidelijke visie enzovoort. Geldt dat voor open source minder?

Het is wellicht minder strak aan deadlines gekoppeld. Maar er is ondertussen wel een *roadmap*-team, die een richting moeten bepalen waar we over vijf jaar of tien jaar willen staan. Daarnaast heb je een *framework* team wat echt meer de review doet van de actuele status. Bij *open source* heb je geen uitgesproken alleen beslissingsrecht . het is veeleer en democratie waarbinnen een hele hoop mensen samen tot een compromis komen om de weg uit te stipplelen. Je kan minder sterk je eigen stempel drukken maar anderzijds heb je wel advies van een hele hoop vakmensen, die ook weten waar ze mee bezig zijn.

Dat geldt in elk geval zo voor de Plone *community* maar daarom niet voor alle andere *communities*. Op ons jaarlijks congres hebben we deelnemende *developers* die echt van wereldniveau zijn. Als je je realiseert wat voor ervaring die mensen hebben en hun tijd en ervaring in Plone stoppen, kan je begrijpen dat er hoogwaardige feedback bereikt wordt. Dat is een gedegen basis om je mening op te baseren. En dat heeft wel voordeel bij je visie. Dat vind ik een voordeel tegenover een *closed source* bedrijf waar een leidinggevende de beslissing neemt.

Die developers krijgen er uiteindelijk niets voor terug? Hoe motiveer je die mensen dan?

Je motiveert ze doordat andere mensen het gaan gebruiken omdat ze het goed vinden. Als je als *developer* kan vaststellen dat er 1000 sites met je toepassing werken, dan geeft dat enorm veel voldoening en motivatie. Als je via *closed source* werkt en 3 of 4 klanten hebt (groot verschil) Weet ook dat wanneer je iets voor de *core* van Plone ontwikkelt er bedrijven zijn als FBI, CSI, NASA, Nokia, die hebben dan hun internet op jouw software draaien. Dat is toch iets dat je niet snel kan bereiken met een *closed source*.

Een community staat of valt uiteindelijk wel met vrijwilligers. Dat is vanuit economisch standpunt moeilijk te begrijpen.

De meeste vrijwilligers doen het natuurlijk ook voor het werk.

Ja, jullie hebben er bij Four Digits toch baat bij dat Plone zich verder ontwikkelt? Want dat is het enige systeem dat jullie gebruiken naar klanten toe?

Wij promoten ons ook heel hard als Plone specialisten Door ons heel actief binnen de *community* op te stellen, , dan gaan ze eerder voor ons kiezen. Zo krijgen wij er ook wat voor terug. Ook kennisdeling is gebruikelijk. We gaan heel veel naar congressen, *sprints*, ... Je ontmoet heel veel mensen en juist omdat iedereen heel open is, omdat je samen die dingen ontwikkelt, leer je heel veel van elkaar en til je elkaar naar een hoger niveau.

Die congressen, wie organiseert die dan?

Eigenlijk is dat vrij onwillekeurig. . Toevallig doen wij het dit jaar in Arnhem. De Plone*conference* wordt nu sinds 10 jaar, elk jaar gehouden . Vorig jaar was dit in San Francisco en werd door 2 *freelancers* georganiseerd. Het jaar voordien was het een Engelse Plonepartij, die het in Bristol lieten doorgaan.

Moeten jullie daarvoor een toestemming vragen?

Je hebt binnen de *foundation*, de *board*, die vraagt aan de *community* om *bids*: *proposals* van de leden. Dit jaar waren er 2 *proposals*: wij en Parijs. Vervolgens gaat de *board* die bekijken en mogen de *members* (ongeveer 130) de *proposal* toewijzen. De *members* kiezen ook uiteindelijk de *board* tijdens deze jaarlijkse conferentie. Dit jaar waren er bijvoorbeeld 9 kandidaten, en daar zijn er dan 7 van gekozen voor de *board*.

Dus elk jaar verandert de board?

Die kan elk jaar veranderen maar dat hoeft niet. Je wordt *boardmember* voor de periode van één jaar maar je kan je herverkiesbaar stellen en bij gevolg langer in de *board* zetelen.

Toch raar als je op lange termijn wil bouwen, je elk jaar de board gaat wijzigen ?

Het blijft wel democratisch. De *board* is enkel het uitvoerend orgaan van de *foundation members*. Het doel van de *board* is ook het beschermen en promoten van Plone. Er zijn geen tegengestelde belangen binnen de *Board*.
Hoe ziet het organigram van de Plone Foundation er dan uit?

Officieel gezien heb je de *board*, de *foundation members* en het marketing team. Dat zijn de daadwerkelijke officiële dingen vanuit de *foundation*. Daarnaast heb je natuurlijk de *release manager* ook nog. Die maakt de *releases* van Plone.

Worden al die functies betaald ?

Enkel de *release manager* wordt betaald vanuit de *foundation*, maar het is geen fulltime salaris. De *foundation* heeft inkomen vanuit de *members* en via de jaarlijkse conferentie. Hiermee sponsort ze ook evenementen.

Maar het product, CMS van Plone brengt geen geld in het laadje?

Neen, alle gelden die overblijven van sponsoring en dergelijke worden terug in de *foundation* gestoken. Het is eigenlijk meer een break-even operatie.

Het is niet zo dat er aan advertising binnen Plone gedaan wordt, om wat extra binnen te krijgen?

Neen, in principe wordt het geld binnen de *foundation* gebruikt om conferenties te organiseren, deelnames aan marketing activiteiten of aan sprints (bv vliegkosten) te financieren Voor het overige krijgt de *release*-manager 1000 dollar voor een *major-release* en 500 voor een kleinere In principe is het de technisch eindverantwoordelijk die moet zorgen dat alles klaar is op de moment dat het goedgekeurd wordt.. Maar in verhouding met de tijd die zo iemand erin steekt stelt dat niet veel voor.

Dus als jullie zeggen, 1 april moet er een nieuwe versie uitkomen, moet hij daar voor zorgen?

Zo strak zijn de deadlines niet. Maar om op het organigram terug te komen: Bovenaan heb je de *board* die door de members gekozen wordt. . De *release manager* die zit eigenlijk boven het framework team Daar heb je dan een heleboel teams onder : het *framework* team, het marketing team, die je kan beschouwen als de staff-functie van de *board*; het documentation maakt alle how-to's en manuals.; het GUI team (user interface), het *roadmap* team, enz. . Daarnaast heb je de *release manager* en het membership commity, die beoordelen mogelijke nieuwe lidmaatschappen. Dan is er nog één heel belangrijk onderdeel en dat zijn de *contributors*. Dat kan iedereen zijn. Hiertoe teken je een contract en draag je jouw rechten op alle codes die jij maakt voor Plone af aan de Plone *foundation*.

Is dat met een bepaalde licentie?

Dat heeft inderdaad met een license te maken, met GPL.

(Andere spreker) Dat is niet helemaal waar. Langs de ene kant geef je je rechten af zodat de Plone *foundation* het onder een willekeurige licentie mogen uitgeven. Dus de *foundation* mag het onder GPL of BSD of wat dan ook uitgeven en tegelijkertijd krijg je zelf ook een licentie . Dus je geeft eigenlijk je *copyright* niet af. Maar wel mag de Plone *foundation* er mee doen wat ze willen. Dat is belangrijk want op die manier kan de Plone-*foundation* alle codes die ze hebben internationaal waarborgen en alle trademarks verdedigen (ook in China en India). Dat is ook één van de rollen van de *foundation* en dus van de *board*.

Veel van die teams zijn *community-driven*. Daar is de *foundation* niet verantwoordelijk voor of mengen ze zich niet in. Dat is de *community* die dat drijft. Enkel marketing wordt gedreven door de *foundation*. De *Foundation* houdt zich helemaal niet technisch bezig maar enkel met promotie en bescherming.

Dus die Foundation moet zich eigenlijk aanpassen aan wat de community drijft. Als de community naar userfriendlyness evolueert, dan moeten jullie die meer gaan marketen?

Ja, dat zal waarschijnlijk wel gebeuren. Bijvoorbeeld Plone 4 kwam uit. Daar zaten bepaalde dingen in die nieuw en hip waren. Dan wordt daarmee gemarket. Uiteindelijk is het een beetje een samenspel.

Maar het is dus niet de Foundation die de route uitzet en bijvoorbeeld voorstelt om zich op snelheid te gaan focussen. ?

Neen, helemaal niet. Daar houdt zich juist de *foundation* NIET mee bezig, dat is vanuit de *community*.

Wordt die community dan op één of andere manier gemanaged, want nu klinkt het een beetje dat de foundation gemanaged wordt door de community.

De *foundation*, en dan heb ik het wel echt over de *board*, die wordt niet echt door de *community* gemanaged, en de *foundation* probeert ook helemaal niet de *community* te managen, ze probeert de *community* te ondersteunen. De *board* ondersteunt de *community* door ze te beschermen met trademarks, copyrights, etc. en door Plone te promoten .

Bijvoorbeeld is er binnenkort Pycon, het grootste python congres ter wereld. Dan zorgen wij ervoor dat wij als Plone er ook wat gaan doen als *open source community*. En dat kan best zijn dat dat een bedrijf is dat daar toevallig in de buurt zit, waar we mee samenwerken?. Maar dat is meer de rol van de *foundation*. Dat zijn niet echt dingen waar de *community* zich direct mee bezig houdt. Bijvoorbeeld komen er vragen vanuit de *community* voor budget voor marketing. Maar om echt te zeggen dat de één op de ander invloed probeert uit te oefenen dat gebeurt niet echt.

Staat het dan volledig los van elkaar ?

Het is eerder een samenspel . Ik zit in de *board* maak ook deel uit van de *community*. Sommige mensen hebben ook zoiets van 'de *board*, hartstikke leuk, maar ik ben blij dat ik het niet hoef te doen'. Sommige mensen boeit het niet wat voor trademarks we hebben in India, maar 't moet wel gebeuren.

Het geld dat de Foundation werft wordt dit door gesponsorde conferenties verzameld?

De jaarlijkse Plone conferentie wordt op twee manieren gesponsord : Enerzijds, worden er toegangskaarten verkocht en 10 procent van alle inkomsten van de conferentie gaat naar Plone *foundation*. Daarnaast wordt een congres ook gesponsord, en ook hiervan gaat 10 procent naar de Plone-*foundation*.

Is dat dan de enigste inkomstenstroom?

Niet helemaal, je kunt als bedrijf ook Plone sponsoren, en dan krijg je een eigen pagina op de website van Plone.

Zo'n conferentie, hoe groot moeten we ons dat voorstellen?

Tussen de 400 en 500 *developers*. De grootste tot nu toe was 420 ofzo. We zitten hier in Europa redelijk centraal en alle Europese congressen zijn drukker bezocht dan de Amerikaanse. Er zitten flink wat bedrijven bij zoals wij, maar ook heel wat freelancers.

Hoe moeten we ons het verloop van deze conferentie voorstellen ?

De conferentie dit jaar is 7 dagen. De eerste 2 dagen is training. Dan zijn er mensen vanuit de *community* die zeggen 'ik wil wel een training geven'. Dat is dan ook betaald. Als je een training wil volgen moet je daarvoor betalen en dat geld gaat dan naar die trainer. Zo heb je wel heel specifieke trainingen over bepaalde pakketten en heel wat kennisoverdracht. Daarna is het 3 dagen conferentie. Dat zijn dus 3 dagen talks (lezingen). Op voorhand kunnen mensen inzenden dat ze over een bepaald onderwerp een talk willen geven, en dan wordt er op gestemd. Er wordt getracht om hel uiteenlopende onderwerpen te behandelen. Zo is er wat vn alles : voor beginners, gevorderden, marketing mensen, *developers*, ... Nadien zijn er nog twee dagen sprints. Daar zal ik het later nog over hebben.

Waar en hoe is Plone eigenlijk ontstaan?

Er zijn 3 mensen begonnen, waarvan na korte tijd enkel nog een Noor en een Amerikaan overbleven. De ene was een GUI-*developer*, die was alleen maar met usability en dergelijke bezig. Hij was op zoek naar een CMS, maar de meeste CMS'en waren toen allemaal gedreven op techniek, en hier kan hij niets mee. ,dus toen heeft hij zelf een paar dingen bedacht. Omdat hij ze zelf niet kon implementeren heeft hij contact gezocht met een Amerikaan en die heeft vervolgens heel de backend gemaakt. Ze hebben het anderhalf jaar lang ontwikkeld zonder dat ze elkaar ooit gezien hebben. Daarna hebben ze op het eerste congres elkaar ontmoet. Het is dus duidelijk helemaal als *open source* begonnen.

Is er ooit een plan geweest om er een soort 'PRO-versie' van te maken om er geld mee te verdienen?

Nee. Ondanks dat er een groot internationaal bedrijf flink wat geld wou neerleggen voor een licentie en dan een Pro versie onder eigen naam wou draaien naast Plone (*open source*). Dat geld zou dan aan de *foundation* betaald worden.

Dat is wel een trend die je vaak ziet. Dat er zowel een free als Enterprise versie is.

Ja, dat zou in dit geval niet helemaal waar gewest zijn. Zij zouden normaal wel wat eigen dingen hebben toegevoegd, maar de Plone, volledige versie zou wel gewoon gratis blijven. Ze zouden gewoon de code gebruiken die *Open source* is, maar ze zouden het dan ook gewoon zelf als een soort van *closed source* willen gebruiken.

Wat je bij veel partijen wel vaak ziet, is dat ze naast een gratis versie ok een betalende versie hebben (*upselling*). Als klant kan je dan gratis straten en doorgroeien naar een zwaardere betalende versie. Het is dan vaak zo dat er een commercieel bedrijf achter zit en bij Plone is dat niet zo. De *Foundation* geniet wel van de naamsbekendheid maar heeft geen doel tot doorverkopen. Een *enterprise* versie betekent vaak dat je een stuk ondersteuning krijgt, en updates. Dat is nu net iets wat de *foundation* niet doet. Er zijn wel heel veel Plone-leveranciers zoals wij, wij leveren wel *enterprise*, als je er een label wilt opplakken. Wij leveren Plone met een implementatieproject en een *service level agreement*. Wij begeleiden ze dus om dat stukje support en een heleboel andere partijen doen dat ook.

De foundation, was die er al bij de oorsprong van Plone, of pas na enkele jaren? De community is van 3 mensen naar een 500tal developers gegaan, hoe is dat gekomen? Is er een bepaald kritisch punt ofzo?

Het is geleidelijk aan gegroeid. Dus in het begin waren er 3 mensen, en dan werd het groter en groter te groot om het nog langer met z'n 3en te doen. Toen een extern bedrijf belangstelling toonde voor de opmaak van een enterprise versie (zie boven) hebben ze er dan al wat geld in gestopt, en met dat geld is de *foundation* dan gestart, naar model van de Apache Software *Foundation*. Dat bedrijf heeft dus eigenlijk betaald voor de Plone *foundation*.

Het lijkt me moeilijk wat dat bedrijf wou doen, omdat je onder GPL werkt, dus alle modules die ze er op zouden willen uitbrengen moeten dan toch ook onder GPL gebeuren.

Klopt, er zijn een aantal dingen die ook onder BSD zijn uitgebracht. Andere partijen mogen dat niet, wij mogen dat wel omdat het onze code is. De *foundation* mag de code uitbrengen onder welke licentie ze dat willen.

Een aantal jaren geleden is er een onderzoek geweest om een aantal modules ook onder BSD uit te brengen. Voorwaarde was wel dat het niet een core-Plone library was en dat het een library was die generiek was en die ook voor andere *python frameworks* kan dienen. Onder die voorwaarden kan je een aanvraag indienen om dus een module onder BSD uit te brengen.

En wat is dan het voordeel?

Dat ze meer compatibel zijn met andere frameworks. Bijvoorbeeld met GPL 2 én GPL 3, immers GPL 2 en GPL 3 zijn niet compatibel en Plone is onder GPL 2 uitgebracht. GPL 2 is een stuk strenger dan 3. Dergelijk onderzoek is dan ook werk voor de *board*.

Er is destijds al in een heel vroeg stadium uitgebreid gekeken naar licenties. Waardoor eigenlijk, we heel veel voordeel hebben t.o.v. andere *open source* projecten. We hebben bijna nooit gezeur. Dat komt ook doordat we die *contribute agreements* heel vroeg hebben ingevoerd. Iedereen moet dus gewoon een contribute agreement tekenen, en alles wat er dan ontwikkeld wordt is dan standaard van de Plone *foundation*. Zo kan het nooit dat er iemand de rechten van opkoopt., en daarin ligt dus het grote voordeel.

Je krijgt ook pas toegang tot alle *source code* als de contributor's agreement getekend is. Je kunt ze wel bekijken, je kan ze zelfs downloaden en gebruiken, maar je kan ze niet uploaden Het gaat dus gewoon niet terug Plone in.

Elke maand krijgen we een paar van die aanvragen tot contributor's agreements.

Naar wat wordt gekeken om die agreements te tekenen. Moet je bepaalde skills hebben?

Neen, daar moet je niet over beschikken .Het komt wel pas in de core wanneer er naar is gekeken. Voor er iets in het core-product komt wordt dit allemaal wel gereviewed door het framework team en de *release manager*.

Je kan dus niet zomaar iets toevoegen voor onmiddellijk gebruik?

Je kunt het wel inchecken. Van elke *check-in* komt er ook een mail in de mailinglist en die worden opgepikt door de *active members*. Dan krijg je wel feedback. De peer-review is wel enorm wat dat betreft natuurlijk.

Dan kunnen we ook even terugkomen op de indeling van de conferentie met name op de sprints. waarbij er dus alleen maar gecoded wordt met iedereen en vaak tijdens de conferentie wordt er al volop geprogrammeerd .Er ontstaan spontaan teams van mensen die aan dezelfde zaken willen werken. Er wordt dan ook altijd gekeken naar wat er wordt toegevoegd. Die sprints leveren vaak de meest productieve discussies op wanneer de mensen gewoon samenzitten en allerlei dingen aan het bedenken zijn. Het aantal *contributors* dat er bijkomt en ook het aantal *check-ins* is na een conferentie altijd wel veel groter.

Nog een vraagje omtrent de structuur van die foundation enz. Is die er gekomen omdat de community te groot werd en er vrees was dat niet alles nog in goede banen zou blijven?

Het is wel belangrijk om te differentiëren. De *foundation* wil niet de *community* in banen leiden, ze wil ze ondersteunen. De *foundation* zegt helemaal niks van 'we moeten nu en nieuwe versie uitbrengen, maar ze faciliteert de comunity om te kunnen ontwikkelen. Dat is het belangrijkste wat de *foundation* doet, het beschermen en het promoten. Beschermen door bijvoorbeeld' de *release manager* aan te stellen. De *community* doet zijn eigen ding , daar is het net *open source* voor. De *Foundation* zal enkel beschermen , faciliteren en promoten .

Kan dat veralgemeend worden naar andere open source CMS bedrijven / communities?

Er zijn een heleboel andere communities die het helemaal anders doen. Plone is wat dat betreft heel erg democratisch. De meeste *open source* pakketten zijn dat de facto niet? Ze zijn vaak ontstaan vanuit 1 bedrijf of 1 of 2 personen en alles moet bij hen worden ingecheckt. Zij willen heel erg veel controle op die dingen houden. En bij Plone willen we dat niet. Iedereen die iets wil doen , willen we ondersteunen. Je moet er dan wel voor waken dat je de kwaliteit ervan steeds controleert. Het *framework* team controleert dus dingen, en het *roadmap* team zorgt vaak voor ideeën van 'je kan dat misschien proberen'. Dus het is veel meer van 'doe het maar' en wij zorgen dat er een organisatie omheen staat zodat hetgene wat je maakt een beetje geshaped wordt zodat het een kwaliteitsvol product wordt. Dat is wel anders dan bij veel andere communities.

Maar uiteindelijk is een foundation of iets anders toch nodig om een community levensvatbaar te maken. Een open source community kan die zichzelf uiteindelijk draaiende houden?

Technisch wel. Het goeie is dat je juist mensen gaat motiveren binnen zo'n organisatie. Als je het net andersom doet, zoals bij python 3 bijvoorbeeld, doen slechts 2 mensen de checkins doen. Zo wordt je gigantisch afhankelijk van deze personen. Als er dan eentje weggaat valt 50 procent van je capaciteit weg. Bij Plone is het zo dat de *board* elk jaar 50 tot 60 procent vernieuwd wordt. Het framework team roteeert om de zoveel tijd, de *release manager* is altijd een nieuwe persoon. Zo heb je elke keer een doorstroom van mensen. Je moet er wel voor waken dat je elke keer een gedegen kennisoverdracht waarborgt en er dus voor zorgen dat al je procedures goed beschreven zijn. Dan maakt het niet meer uit wie het doet.

Zijn de oprichters nog in de board?

Neen, wel tot afgelopen jaar was de Noorse oprichter nog in de *board*, maar hij vond het niet meer nodig. Er zijn voldoende experten aanwezig en hij heeft zich opnieuw op de GUI binnen Plone gestort.

Wij hebben ook met Fork-CMS een gesprek gehad. Daar bestaat geen foundation en geen begeleiding . Hoe kan dat dan overleven?

Als het goed gaat met zoiets, dan blijft het goed gaan. Zo'n *foundation* grijpt in als het dreigt slecht te gaan of om het weer op de goeie rails te krijgen. Een echt probleem omtrent trademark of copyright issue kan een *community* kapot maken.

Een andere reden ligt ook in de grootte van een *community*. Het aantal mensen dat aan Plone werkt is vrij groot maar als je dat vergelijkt met een een Drupal of Wordpress, dan is ons cijfer relatief klein voor het product/gebruik dat we leveren. Dan heb je daar per *developer* een totaal andere verhouding. Ook de structuur/organisatie is van belang , als enkelen alles moeten regelen , dan blijft het niet duren. Wie spreekt met advocaten ? Binnen de *Foundation* kunnen we heel wat vervelende dingen wegnemen, en dat levert een hele hoop voordelen op.

Hoe lang zijn jullie al bezig met Plone?

Bij *Four Digits*, sinds 2005 officieel en sinds 2007 fulltime met Plone.

Ik kan me inbeelden dat er bij mogelijke klanten van Four Digits een angstgevoel bestond tegenover open source CMS. Vroeger was OS toch veel minder populair? Gratis heeft een slechte naam...

Ja dat klopt, maar het hangt een beetje van je soort klanten af en natuurlijk ook hoe je het verpakt. Dat het gratis is, is in dat opzicht misschien het minst positieve aspect aan OS. Er zijn heel wat andere voordelen bij OS die je klanten kan meegeven. Een mogelijk voordel voor de klant is dat hij makkelijk van het ene bedrijf naar het andere kan overstappen, want iedereen mag met Plone werken. Een aanpassing kost ook geen fortuinen zoals vaak voorheen. Het motiveert ons als bedrijf natuurlijk ook om goed werk te blijven leveren zodat de klant niet wegloopt. Bovendien heb je bij Plone het feit dat heel veel mensen naar de code kijken en er juist heel veel lekken al uit gehaald zijn. Daardoor is de beveiligings track record van Plone ondertussen enorm goed. Als bedrijven als NASA en FBI er gebruik van maken, dan toon je wel aan dat het veilig is.

De userbase is bij een *closed source* pakket bijna altijd veel kleiner dan bij Plone , wiens kracht ligt in de getallen, net omdat zoveel mensen ermee werken zijn alle bugs eruit. Wat dat betreft, is het bij ons alleen maar positief uitgepakt. Met Four Digits leveren we heel veel support, regelen we zelf de servers en maken zelf de nodige aanpassingen aan de broncode . Als wij SharePoint zouden leveren, dan kunnen we inderdaad prima support leveren op SharePoint, maar we kunnen geen bugs aanpassen, want we hebben geen toegang tot de broncode.

Ee,n standaard vraag voor een closed source bedrijf is , werken jullie met partners/ outsourcing ? Is dat in een community dan niet langer nodig?

Dat is een rekbaar begrip. Alhoewel we bij Fourdigits alles zelf ontwikkelen leveren we geen design. Daarvoor werken we dan weer wel samen met partners. Hetzelfde geldt voor usability en copyrighting en werken we dan ook samen met een vaste partner . Daar hebben we dan veel communicatie mee ook tijdens de conferenties enz.. Ze hadden op een gegeven moment een module ontwikkeld om een koppeling met iets van Microsoft te realiseren,. Die module wilden ze gaan *open source*n, alleen was dat zo complex en dan hebben wij ze weer geholpen , maar dat is eerder uitzondering dan regel.

Geldt dat dan ook voor de community of foundation?

Dat hangt een beetje af van de grootte van de opdracht. Indien de opdracht te groot is kan je een samenwerking opzetten met een andere Plone partij. Voor de *Foundation* kan samenwerking enkel op technisch vlak en nooit op commercieel vlak, dat dient dan via de bedrijven zelf te gaan Er is wel èèn uitzondering. Er bestaat een organisatie waarvoor de *foundation* ontwikkelt. Een samenwerkingsverband tussen een 30tal Europese niet concurrerende partijen , maar dan alleen op zakelijk vlak, bijvoorbeeld personeelszaken enz.,

Echte outsourcing zoals ondermeer opdrachten uitbesteden aan lagelonenland gebeurt niet. Noch vanuit de *community* en ook niet vanuit de *foundation*.

Er zijn wel bedrijven die zeg maar mensen inhuren om aan Plone te werken. Dan wordt een nieuw onderdeel van/voor Plone ontwikkeld. *Four Digits* heeft bijvoorbeeld de 10procent regel. 10 procent van onze tijd wordt aan Plone gespendeerd. , dat kan *development* zijn ,maar ook tijd gespendeerd binnen de *board* , het *framework* team of meewerken aan tune-ups kan hieronder vallen.

Heeft Plone een soort van klantenfeedback systeemvoor de eindgebruiker (geen developer) Is dat dan ook dat ticket systeem? Volgen jullie op of klanten blijven of voor iets anders kiezen ?

Je hebt aan de ene kant Plone bedrijven en dan aan de andere kant de *foundation*. De *foundation* heeft tickets en mailings , etc... De meeste Plone installaties gebeuren toch via bedrijven., zoals hier bij *Four Digits*. Ieder bedrijf heeft zijn eigen systemen dan.

En brengen jullie die feedback dan terug naar Plone foundation?

Naar de *community* wel. Bijvoorbeeld van 'dit is een reden waarom klanten Plone niet hip vinden', dus hier gaan we iets aan doen. Ik vind Plone onder bepaalde omstandigheden best wel traag, en dan kan je daar iets proberen aan te doen. Dat zie je veel. Als een *community* het niet eens is, gaat het niet gebeuren, dan gaan ze het niet maken.

Maar als je zelf je eigen site maakt via Plone, wordt dat dan op één of andere manier opgevolgd?

Ik zou niet weten hoe we dat kunnen implementeren. Er is uiteraard wel mogelijkheid tot feedback, zoals het ticket systeem, maar het moet wel altijd vanuit de persoon zelf geïnitieerd worden. Een terugmelding zoals bij SharePoint waar alles naar Microsoft gaat, zo'n systeem is er niet nee.

Zijn het dan vooral bedrijven die Plone implementeren en niet zozeer privé-personen?

Er zijn wel een hoop freelancers, maar meestal is het inderdaad wel een bedrijf die het gaat opzetten voor een klant. Dat heeft ook te maken met de complexiteit van Plone. Het goed configureren is wel vrij complex. Het vraagt echt wel tijd om iemand om te scholen van by Linux naar Plone.

11. Transcript interview – Eddy Lalou, Sitecore

Participants:

Reinout Denys	Ghent University
Gert Vanhaverbeke	Ghent University
Eddy Lalou	Sitecore
Location:	Barastraat 175
	1070 Brussels
	Belgium
Date:	Friday, 2nd February 2012
Time:	10.00 - 11.00

Hoe zie je de sector momenteel en waar plaats je jezelf erin?

Gartner bespreekt de *open source* space en de *solutions* space waar wij in zitten. Ondertussen is er ook een nieuw Gartner magical quadrant gemaakt. Daar staat een analyse van hun in over open en *closed source* als neutrale analyst.

Hoe zijn jullie tot een close source software aanbieder gekomen?

Door een aantal .Net en content management specialisten die een eigen bedrijf opgericht hebben (heden ten dage nog steeds een privately owned bedrijf en staan niet op de beurs). Door hun knowhow over de .net ontwikkelingen en hun visie over Microsoft evolutie naar digitale kanalen, hebben ze zich daar op geënt om een oplossing te bouwen, een *out-of-thebox solution* voor web content management systemen. Met dat product zijn wij in het marktsegment naar een leiderschapspositie doorgegroeid. Die positie hebben we bereikt door een aantal zaken: het product opzicht; technisch, functioneel, user experience, (implementatie) efficiëntie, efficiëntie om het te integreren in een algemeen applicatielandschap dat op Microsoft gebaseerd is.

Drie jaar geleden heeft het management de visie naar de toekomst wat gewijzigd. CMS systemen zijn commodities geworden, de evolutie zal op technisch vlak zeker ook verder gaan, zodanig dat in de toekomst ook *cloud* enabled is. Maar dat is een *commodity*. Als wij dan tegenover OS bedrijven in competitie zitten dan gaat het om een prijsdiscussie, die wij niet willen hebben. Wij hebben het over de added value van ons totale *out-of-the-box*

solution. De strategische *shift* die gekozen is betreft naast de CMS ook de ontwikkeling van enkele extra (*out-of-the-box*) modules die we als *out-of-the-box solution* kunnen verkopen, maar die een meerwaarde kunnen bieden boven op het CMS-platform. Op die manier moeten we niet negotiëren over prijs tegenover *open source* of andere competitors, maar dat we een value ad hebben en dat we daarmee wel succesvol op de markt zijn.

Blijven die modules nog altijd binnen het CMS gebeuren of legt het linken met andere onderdelen van een onderneming?

Het legt linken met andere domeinen van applicaties. Omdat alles wat online en via de digitale kanalen gebeurt meer en meer multi-kanaal wordt en onze klanten hun klanten meer en meer multi-kanaal willen benaderen met eenzelfde user experience, hebben wij verschillende blokken uitgebouwd om ook het mobiele kanaal met dezelfde content te laten werken, ook email campaign managers en marketing *automation* functionaliteiten. Heel dat lappendeken hebben wij als *added value solutions* bovenop de CMS gebouwd en de totaliteit zelfs tot en met *analytics* op websites maakt dat we een differentiator hebben op de markt. Met ons business model, ons product en onze pricing hebben we meer een discussie over de fit ten opzichte van requirements en over de voordelen van een *out-of-the-box solutions* eerder dan een prijsdiscussie over een *commodity* als een CMS. Dat geeft ons een voordeel ten opzichte van OS alleszins bij die klantsegmenten waar dit een voordeel kan zijn. De KMO markt zal sneller voor een *open source* product van 2000 euro of 0 euro kiezen dan voor een *out-of-the-box solution*. Daar is niets mis mee, maar dat is ons marktsegment niet.

Wat definieer je juist als uw marktsegment?

Midmarket en enterprise market.

Een kenmerk van de OS-markt is dat het misschien sneller aanpast en sneller evolueert dan CS. Wat is uw visie hierop?

Dat is een valse economie, dat is niet juist. De OS-wereld maakt wel een *patchwork* en groot pakket aan *frameworks* klaar die mensen kunnen gebruiken om hun eigen omgeving op te zetten, maar ze moeten die nog altijd opzetten, bouwen, integreren en onderhouden. Die *frameworks* evolueren praktisch niet. Wij hebben met ons *out-of-the-box* product een *roadmap* van 18 maanden vooruit waar om de 3 maanden een nieuwe functionele evolutie in zit. Bvb, de recente upgrade heeft de hele integratie naar de social network gebracht. Onze klanten die onze licentie al hebben, krijgen gratis en voor niets - omdat ze het maintenance contract hebben - nieuwe functionaliteiten om ook met social media

campagnes te gaan vieren via en met hun website(s), closed loop. Wij evolueren dus functioneel, OS heeft misschien wat meer technische evolutie maar dat is een heel andere *trade-off*, dan die waar wij voor kiezen. En dat is ook onderdeel van een keuze die een bedrijf maakt, een bedrijf dat een grote IT afdeling heeft en zelf die evolutie van een dergelijk OS-framework zelf kan beheren en bekostigen gaat daar misschien voor kiezen, daar is niets mis mee. Wij komen dit natuurlijk ook in de markt en in de competitie tegen, het statement van OS is gratis. Dat is ook niet juist, dat is een valse economie.

Merk je verschillen in de manier van werken tussen de grote en kleine bedrijven binnen closed en/of open source? Hoe gaan jullie als grote speler op de markt om met kleine bedrijven die dan meer differentiëren in het product dat ze aanbieden?

Dat legt het Gartner model heel goed uit. Er zijn bedrijven die niet in het *leading quadrant* zitten met hun oplossing, die zitten daar omdat ze niet internationaal, *cross border* georganiseerd zijn of omdat ze met hun CMS-product een heel beperkte niche van mogelijkheden of een verticaal marktsegment innemen bvb CMS-en voor ziekenhuizen, omdat daar een elektronisch passportintegratie inzit. Goed, daarom dat ze ook daar zitten waar ze zitten. Niet in het *leading quadrant*, maar misschien wel in hun nichemarkt een goed product aanbieden. Dat is een onderscheid dat wij zien, wij proberen globaal, internationaal, holistisch een *out-of-the-box solution* te zijn die veel dingen kan. Dat is een andere visie.

Dit is dan misschien een belangrijkere vraag voor kleine spelers dan voor grotere spelers?

Door de evolutie die wij doorgemaakt hebben met ons product hebben wij natuurlijk een stap gemaakt in aanverwante software markten; *marketing automation, e-commerce, web analytics, social media...*

Is dit ook een evolutie die je ook in het algemeen ziet in heel de CMS-markt?

Wij zien daar followers in die dat ook doen met hun product. Sommige van onze concurrenten doen dat ook. Die gaan nu ook mee in die richting. Dat maakt dat we *open source* competitie hebben in ons segment, maar ook in die vier of vijf andere aanverwante segmenten. Een klant die zegt "ik wil een volledig nieuw CRM-systeem", daar heb je *out-of-the-box solutions* versus *open source solutions* versus … Wij willen dat ook koppelen aan ons online kanaal dus we moeten een *solution* hebben die zowel marketing automation als online en web content management aan moet kunnen sturen. Dan heb je natuurlijk plots andere concurrenten in die andere markt. Ons voordeel is dat alles op elkaar afgestemd en geïntegreerd is.

Een meer programmeergerichte vraag: gebruiken jullie bij Sitecore ook stukken code die open source zijn, maar wel de licentie hebben die ervoor zorgt dat je ze mag gebruiken?

Dat denk ik wel, want wij hebben als Sitecore, net als de *open source* wereld, een community van certified Sitecore *developers* opgestart. Daar zitten momenteel een stuk of 5 à 6000 *developers* in. Die kunnen dus producten ontwikkelen, of modules of functies die zij in een *shared source* omgeving kunnen opladen en onder elkaar kunnen verdelen. Bij sommigen daarvan zal zeker gebruik gemaakt worden van *open source* code. Die producten/modules/ functies zijn dan wel enkel beschikbaar voor de certified Sitecore *developers*.

Net zoals de *open source* leveranciers soms een community hebben en beweren dat hun *community* een *lever* is, hebben wij dat ook, ook al zijn wij een *closed source* bedrijf.

Worden de mensen binnen uw community hiervoor dan betaald?

Sommigen wel, sommigen niet. Wij hebben bijvoorbeeld ook een landschap wereldwijd van Sitecore-partners. Dat zijn bedrijven die onze technologie verkopen maar ook deployment en consulting services daarrond leveren. Die leveren wij niet als software leverancier, die levert ons partner netwerk. Sommige van die partners ontwikkelen ook een stuk oplossing; dit kan een connector zijn naar een of ander technologie, dat kan een stuk extra functionaliteit zijn... Die kunnen zij zelf commercialiseren, op voorwaarde dat zij die ontwikkelde componenten laten valideren bij Sitecore en een creditatie krijgen zodat zij Sitecore enabled en Sitecore *support*ed zijn.

Kunnen de mensen uit die community dan aan die source code? En is die community dan niet gevaarlijk?

Neen, de mensen van de community kunnen niet aan de source code. Zij bouwen enkel maar componenten bij. In onze oplossing zitten ook componenten zoals bvb een geïntegreerde search engine of een blog of een poll module, sommige van die modules zijn *open source* producten. De eindklant die uiteindelijk onze licenties koopt, die weet dat, maar die heeft daar ook geen last van of geen onderhoud aan. Dat wordt allemaal achter de schermen verzorgd.

Wordt die community op een bepaalde manier gemanaged binnen Sitecore?

Ja, wij hebben enerzijds een portal gemaakt specifiek naar die community toe, waar die allerlei technische documentatie, *roadmaps*, vergelijkingen, Gartner rapporten kan downloaden. Dat is dus een *developer community portal*. Anderzijds, degene die een *accredited Sitecore developer* zijn die hebben ook access tot onze wereldwijd georganiseerde *support* desk. Die accepteert alleen aanvragen en bug reports van de accredited Sitecore *developers*. Niet van andere mensen omdat die de technologie onvoldoende kennen om een gedetailleerde diagnose en rapport af te leveren.

Wordt het portal gebruikt om vanuit Sitecore te zeggen; is er iemand om dit of dat te ontwikkelen? Of gebeurt het eerder in de omgekeerde richting zodat developers uit eigen initiatief bepaalde dingen ontwikkelen?

Dit gebeurt voornamelijk in de tweede richting. Er zijn creatieve *developers* die iets ontwikkeld hebben waarvan ze zeggen dat het interessant is om in de *shared source* bibliotheek te uploaden. Sitecore gebruikt het portal niet om eigen vragen betreffende bepaalde ontwikkelingen naar de community te stellen. Het zijn vooral de *developers* die interessante en leuke dingen doen, maar ze zijn daar vrij in.

Worden die componenten die geüpload worden op een of andere manier gecontroleerd?

Op het moment dat mensen die componenten willen commercialiseren, moeten ze een accreditation van onze technologiecel krijgen. Als men die componenten niet wil commercialiseren, mag men die delen, maar worden die componenten niet gecontroleerd of ze volledig sound proof zijn. Zodra men ze wil commercialiseren gebeurt dit wel, want voor commerciële producten willen we wel dat die technologie gecheckt is. Voor eigen gebruik is dit niet nodig.

Legt Sitecore zelf de link naar de klanten of gebeurt dit via het partnernetwerk?

We leggen zelf ook wel de link naar de klant, maar de opvolging gebeurt dan meer via partners. Het implementeren van een Sitecore-platform heeft altijd twee luiken. Enerzijds de aankoop van licenties en de relatie met Sitecore op vlak van onderhoud van die licenties, software en evolutie. Anderzijds, een project dat zegt "we gaan nu een website ontwerpen, deployen, configureren, vullen met content en integreren met andere toepassingen." Dit is een project en gebeurt via de partners. Zij zijn dus *'project delivery'* verantwoordelijk, wij hebben alleen een product verantwoordelijkheid naar de markt. Wij moeten zorgen dat we bug free, internationaal en multi-language zijn en ook meegaan met nieuwe evoluties, meegaan met nieuwe markten (bvb China). Wij zijn *product owner* en verantwoordelijk, onze partners zijn project verantwoordelijk.

De relatie met die partners, vertrekt die van jullie uit of komen de partners eerder naar jullie?

Dat gebeurt in twee richtingen. Wij hebben natuurlijk een visie betreffende wat de beste *competence mix* is om ons product en de evolutie van ons product goed te implementeren. Wij kiezen daarvoor *prefered partners* die wij benaderen en partner proberen te maken. Zij moeten daarvoor aan een aantal voorwaarden voldoen in termen van competenties die ze opbouwen rond ons product, het kunnen geven van demo's enzovoort. Anderzijds zijn er agencies, digitale bedrijven, consultancies en onafhankelijke freelancers die ons benaderen om een partnership te kunnen verwerven. Dat is dus de andere richting waarin wij kunnen filteren; Past dat? Is het een *sustainable* relatie? Of is het een opportunistische relatie blijkt te zijn, steken we daar natuurlijk geen tijd en energie in. Alle andere partners die professioneel met ons werken, trainen en coachen wij. Wij hebben net als voor de *developers*, ook voor de partners een wereldwijd portal. Daar kunnen zij commerciële productinformatie, prijsinformatie, case stories en dergelijke verkrijgen, zaken die hen helpen. Op voorwaarde dat zij een professionele en accredited Sitecore partner zijn. Dit wordt dan ook contractueel toegezegd.

Is er dan ook een financiële link tussen Sitecore en de partners of geven jullie gewoon de partners de mogelijkheid jullie product aan te bieden?

Wij gaan geen venture aan met die partners, wij worden dus geen aandeelhouder van die partners, dat blijft neutraal. Maar natuurlijk zoals in alle *license sale* business modellen, krijgt de partner voor die projecten die hij verkoopt en waarbij onze software aangekocht wordt, een commissie, dat is maar normaal. Certified Microsoft partners krijgen bvb ook een commissie als ze Microsoft producten verkopen, dit is zo in de software-industrie in het algemeen.

Zijn er verschillen in het product dat jullie aanbieden?

Er is eigenlijk één groot CMS met dan de verschillende modules errond die ervoor zorgen dat het een aangepast product wordt naar de verschillende klanten toe, maar het vertrekt altijd vanuit de ene CMS. Onze technologie blijkt dermate modulair te zijn dat onze technologie op verschillende manieren kan worden toegepast. We hebben een CMS waarmee verschillende klanten en partners van ons een online e-shop hebben gemaakt. De functionaliteit die er standaard inzit, leent zich daartoe. Er zijn andere partners die daarmee intranetten voor bedrijven hebben gemaakt, omdat ze de communicatie naar bepaalde employees of naar afdelingen kunnen personaliseren en een tweewegscommunicatie met blogs, scores, en polls kunnen realiseren met hun employees. Daarnaast hebben wij ook 'out-of-the-box' een module specifiek om een e-commerce product te maken. Daar zit meer functionaliteit achter, voor het registeren van online orders, voor het betalen, voor het afleveren en opvolgen van de logistiek... Zo hebben we ook een module die een intranet accelerator heeft. Als je een intranet wilt, krijg je daar 100 objecten in waarmee je al binnen de week een intranet *up and running* kunt hebben. Er zijn verschillende wegen naar Rome, en sommige klanten vinden innovatieve manieren om ons product te gebruiken, sommige partners vinden innovatieve manieren om met ons product meer te doen dan voor wat wij het functioneel en commercieel maken. Dat vertrekt dan wel meestal vanuit de partners.

Wij hebben dus een *roadmap* die wij ook delen via onze partners, ons partnerkanaal, zodat zij zouden beseffen dat wat wij vandaag verkopen eigenlijk de toekomst is van onze klant van morgen. Wat wij vandaag in een project in de *scope* definiëren en commercialiseren houdt al rekening met de toekomstvastheid van die keuzes, want straks komt er iets nieuws aan zodat we weten wanneer er iets nieuw komt. Op die manier kunnen partners hun project baseren of minimaal iets maken dat op termijn toch standaard opgevangen wordt door het product. Wij hebben een *roadmap* die 18 maanden vooruit plant en ook daarover communiceren we met het partnerkanaal. Wij gaan daar altijd mee met de *fast followers* zowel in het functionele domein (social media integratie bvb) als in het technisch domein (dan kan ons product straks in de *cloud* geladen worden, dat is nu al in bèta test).

Op korte termijn is het nieuwe momenteel dan de modulaire structuur, maar wordt dat dan in de toekomst aanzien als de nieuwe standaard?

Dat klopt, wat nu nieuw is wordt in de toekomst een *commodity*, maar dan zijn er andere verwachtingen van de markt. En dan wordt integratie met de *cloud* waarschijnlijk iets nieuws.

Wat interessant is over onze organisatiestructuur: wij zijn dus globaal georganiseerd. Op het gebied van product R&D en product marketing hebben wij 200 mensen die dagdagelijks bezig zijn met de evolutie, ontwikkeling en *support* van ons product. Dat betekent dat wij continu kijken naar 'wat zijn de nieuwe marktbehoeften van morgen?' en hoe moeten wij ons product binnen de *roadmap* gaan aanpassen om aan die behoeften te kunnen voldoen. 200 man enkel daarvoor is uiteraard een serieuze investering. Maar dat is wat een *out-of-the-box solution* provider altijd moet doen als hij professioneel werkt en een leader wilt zijn. Dat is wat *open source* bedrijven minder doen. Daar zitten wel creatieve mensen achter die ook kijken naar de markt en die ook creatieve frameworks in de lucht brengen, maar die zijn niet zo gestructureerd als wij. Ook op vlak van *support* hebben wij 24/7 mogelijkheid. Dat betekent dat wij world wide *support* centra en product specialisten moeten organiseren; Oekraïne, Singapore, Amerika (West- tot Oostkust). Zodanig dat onze klanten 24/7 altijd bij ons terecht kunnen.

Wordt R&D gecentraliseerd vanuit één visie of gebeurt dit ook vanuit verschillende hoeken?

Dit wordt gecentraliseerd vanuit één visie. In ons hoofdkwartier zitten een aantal R&D mensen en product marketeers die het centraal ontwikkelen en verder uitdragen. Onze CEO niet te na gelaten, die zelf ook een heel scherpe visie heeft van waar hij met zijn product naar toe wil. lets wat voor ons een 'asset' en 'competitive advantage' geeft in de markt is onze alliance met Microsoft. Microsoft heeft wereldwijd 40 000 technologiepartners. Daarvan zitten er 10 on-site in Redmond, om joint R&D en productintegratie en technolgie-integratie aan te sturen. Sitecore zit bij die 10. Ook daar zijn we een fast follower en leading-edge bedrijf tegelijkertijd. Als Sitecore morgen zegt: de smaak waarmee we in de cloud gaan is zo, dan zullen we binnen de twee maanden daarna ons product ook met die smaak in de cloud beschikbaar kunnen stellen. Denken ze daar binnen een jaar anders over en zeggen ze dat onze Microsoft-technologie zo werkt met satellieten bvb, dan zullen wij - omdat we in die scope zitten - ook ons product kunnen enabelen op dat vlak, zowel technologisch als functioneel. Dat doet een open source bedrijf minder, aangezien het daar van individuele figuren afhangt die open source frameworks maken en bouwen, hoe scherp zij mee zijn of hoeveel tijd zij zelf nog over hebben om scherp mee te zijn.

Het CMS van Sitecore is dan ook niet beschikbaar voor Mac- of Linux-platformen, de keuze is gemaakt om enkel te focussen op Microsoft. Ook geen Java of andere operating systemen. Op vlak van database systemen ondersteunen wij enkel SQL 2010 en Oracle.

Is er een manier waaromp Sitecore met hun klanten contact legt?

Daar is een evolutie in gekomen de laatste twee jaar. Tot voor twee jaar, beschouwden wij ons partnerkanaal als hetgene dat de face to face contacten met de klanten onderhield. We trainden onze partners om dat goed te kunnen en dat professioneel te doen. Dat heeft altijd goed gewerkt en gebeurt de dag van vandaag ook nog altijd voor een groot deel. Maar door het feit dat ons product die meerwaarde biedt, krijgen wij meer en meer eindklanten die zelf ook globaal en enterprise-level klanten zijn. Die verwachten van een software vendor een directe relatie, zij willen de productexperten soms bij een project betrekken, zij willen soms bij een project een commerciële en legale relatie rechtstreeks met de software vendor. Zij willen ook een aantal contractuele zaken rechtstreeks met de *software vendor* kunnen aangaan, van betalingstermijnen tot risico van wanneer iets fout loopt, of tot het scopen van welke *support* men verwacht wereldwijd. In die zin is er een vorm van directe relatie met onze eindklanten, zeker met de enterprise level klanten. Daarnaast, omdat ons product de laatste jaren geëvolueerd is van een 'plain vanilla' CMS naar meer een breder platform met andere mogelijkheden, hebben wij zelf ook direct naar de markt, en niet alleen via onze partners, moeten communiceren en hebben wijzelf met de grote bedrijven contacten gezocht en hebben wij een stukje organisatie moeten opbouwen rechtstreeks met de markt, naast onze partners die een evangelisatie doet van wat Sitecore allemaal kan. Als die mensen daardoor geïnteresseerd worden en een project willen doen met ons product en uiteindelijk ons product selecteren, moeten zij via het partnerkanaal de nodige projecten en licenties en andere zaken laten scopen en dan kunnen zij bij ons licenties kopen en met de partner een project doen. Het is niet de bedoeling om op termijn het directe kanaal te gaan uitbreiden, enkel grote bedrijven worden op die manier benaderd. Dit omdat zij de vragende partij zijn om een één op één relatie te hebben met de software-aanbieder. Dat is normaal als je in die grootte en in die klanten actief bent. Anderzijds omdat we in onze productcommunicatie ook een beetje naar de markt moeten werken, los van ons partnerkanaal (die dat ook doet), maar wij proberen dat ook te stimuleren om misschien al vroeger dan onze partners te communiceren over onze *roadmap* en over nieuwe trends en over de aansluiting van Sitecore naar die nieuwe trends, want zij hebben ook andere projecten en veel werk en zijn misschien iets minder bezig met het communiceren. Dat is onze rol, dat moet je ook als product marketeer doen voor een software bedrijf.

Wordt klantentevredenheid op een of andere manier gemeten?

Dat gaat op bepaalde vlakken heel formeel, op andere vlakken veel minder. Wij hebben jaarlijks een *customer survey* van klanten die in de loop van dat jaar ons product kochten en een project realiseerden. Dat is een *back office survey* organisatie die dan klanten wereldwijd of regionaal belt als we dat willen. We hebben natuurlijk ons partnerkanaal dat wij ook bevragen om *cases* te schrijven over klanten. Dan hebben we op dat moment een directe klantenrelatie, al is het maar communicatief om die *case* proper te beschrijven en op onze website te krijgen. Wij merken ook soms, in sommige markten, dat een klant beslist om van partner te switchen. Dat is altijd een signaal om die klant eens op te bellen en te vragen waarom men dat doet. Sommige klanten zeggen dan dat ze al 3 jaar met een partner werken, altijd tevreden waren, maar nood hebben aan nieuw bloed, een nieuwe visie. Voor ons is dat geen probleem. Anderen zeggen dat ze niet tevreden zijn, maar dan willen wij wel weten hoe dat komt. Nog andere zeggen dat ze initieel kozen voor een technische partner die ons allemaal leert hoe we alles in gang moeten zetten, maar nadien willen ze een marketing partner. Dat zijn twee andere soorten partners, maar voor ons maakt dat niet uit. De surveys zijn het meest passend bij de vraag naar klantentevredenheid aangezien zij het meest direct van de klanten komen. Het feit dat we ook via onze partners bepaalde signalen opvolgen is dan ook een informele vorm van opvolging van klantentevredenheid.

Zijn er veel mensen die eerst gebruik maken van een Sitecore product, maar er na een tijdje van af stappen en een ander product gaan gebruiken? Of blijft de klant trouw aan Sitecore?

De loyaliteit is zeer hoog bij Sitecore. Mede doordat we ervoor zorgen dat we meeevolueren met de marktbehoeften.

Verschilt de kostprijs van een systeem van klant tot klant? Of is er een standaard prijsstrategie?

We hebben eigenlijk een prijsstrategie die gebaseerd is op de schaalbaarheid van onze licentie. Daar zit een stuk technische schaalbaarheid in, die te maken heeft met hoeveel websites, hoeveel content, hoeveel databases nodig zijn, afhankelijk daarvan heb je één of meerdere licenties nodig. Er is een opschaalmogelijkheid om van een basisconfiguratie primary, naar een professional, naar een enterprise level door te groeien of als je groot genoeg bent direct een enterprise level pakket aan te kopen omdat daar 7 serverlicenties mogelijk zijn, daar ook een development omgeving bij zit en ook een stukje back up en recovery functionaliteit die interessant is. Dat kan ook. Daar zit een pricing-strategie achter in die zin. Weeral, doordat we meer en meer voor internationale bedrijven zijn gaan werken, hebben we dat pricing-model wat moeten aanvullen met, wat wij noemen, een organisatiefit. Bvb, als je met P&G een project doet, gaat P&G een project doen voor één van zijn merken, of voor al zijn merken of voor een categorie van producten of voor de hele wereld of alleen in Europa? Dat zijn dingen die dan ook medebepalend gaan zijn in de eindprijs die wij aanbieden aan de klant. Maar standaard is er dus een pricing politiek die zegt: wij hebben serverlicenties, dus geen klantlicenties. Afhankelijk van hoe een klant van ons zijn infrastructuur wil opzetten, of internationaal werken of qua failover werken hebbe zij meer serverlicenties nodig. Om die reden schalen zij in een bepaalde bandbreedte en kost en dat een bepaald bedrag. Dan hebben we ook een aantal extra modules, die on-demand aankoopbaar zijn, byb een connector met MS SharePoint, om de content van SharePoint ook op de website te kunnen gebruiken, een CRM-integrator, misschien een module om emailcampaign management te optimaliseren... Alle klanten nemen die niet standaard 'out-of-the-box'; die hebben dus een 'features and options'-menu,

waar zij deze modules kunnen opnemen in hun standaardlicentie en daar ook het onderhoud op te betalen en ook het onderhoud voor te verkrijgen op termijn.

Een vraagje specifiek over de modules; zijn de modules nog aanpasbaar binnen zichzelf of zijn deze standaard?

Die modules zijn één pakket en kunnen dus niet meer aangepast worden.

Als je een project aanbiedt aan een klant tegen een bepaalde prijs, zitten onderhoud en ondersteuning daar dan altijd standaard in of zijn er op dat vlak ook verschillende gradaties die dan elk hun eigen prijs hebben?

De politiek daar is dat bij de eerste aankoop, de initiële aankoop van een licentie van ons, de kost voor onderhoud voor het eerste jaar is inbegrepen. Vanaf het tweede jaar betaalt die klant altijd 20 procent onderhoud en ondersteuning kosten. De klant kan dat wel opzeggen indien hij dit wenst, maar dan moet hij daar ook de gevolgen van dragen (volgens de terms & conditions). Hij heeft dan geen toegang meer tot nieuwe releases, tot nieuwe functionaliteiten, maar hij behoudt wel zijn product. Dat heeft hij gekocht, dus dat is dan van hem. Zo werkt ook de standaard binnen de software industrie.

Uit welke inkomstenstroom genereren jullie het meest inkomsten? Kun je daar een percentage op kleven?

Laat ons zeggen dat de totale turnover van Sitecore op jaarbasis in 2010 iets van 75 miljoen euro *license sale* alleen is, daar zit vooral new sales in, dus *maintenance* is daar een fractie van. Uiteraard, als je dit uitzet op een tijdspad, hoe meer new sales je maakt, hoe groter de inkomstenstroom voor maintenance zal groeien in de toekomst. Pas op, die inkomstenstroom hebben we ook nodig, het is een heel ander verhaal om een maintenance- en supportorganisatie op te zetten voor Europa dan wereldwijd, daar komen extra kosten, mensen en infrastructuur bij kijken.

Daarom zou je toch denken dat die inkomstenstroom (maintenance and support) de grootste inkomstenstroom zou zijn, niet?

Neen, toch niet. Als je je product met een *roadmap* van 18 maanden wil laten evolueren, groeit de R&D inspanning om die 200 mensen te betalen ook. Je hebt die inkomstenstroom dus professioneel echt wel nodig om de goede dingen te blijven doen voor uw klanten en uw producten. Maar sinds de laatste 4 à 5 jaar hebben wij jaarlijks een turnover groei van 30 of 40 procent, sommige jaren soms meer. We dachten dat dit fiscaal jaar er met de huidige economie misschien wat afvlakking zou zijn, maar de eerste helft van het fiscaal

jaar hebben we terug meer dan 50% groei gemerkt in de license sale. Dus dit product is dermate goed en goed gepositioneerd dat we blijven groeien met een stabiele ratio.

Is die groei te vergelijken met andere bedrijven binnen de sector? Hebt u daar een zicht op?

Die vraag krijgen we altijd bij requestfull proposals, maar er zijn nergens wereldwijde geconsolideerde zichten of cijfers op dat vlak. We kunnen wel zeggen dat we een marktaandeel hebben van 5, 10 of 20 procent, maar het is niet kwantificeerbaar. Zelfs voor België bestaan dergelijke cijfers niet. Er is geen vakliteratuur of analist en specialisten die ons dat cijfer kunnen geven van: hoe groot is de markt voor CMS- en web content management in België in producten en in diensten?

Kun je stellen dat jullie zich focussen op de kernactiviteiten en de bijkomende activiteiten outsourcen, bvb via het partnernetwerk?

Onze kernactiviteit is productie. Wij zijn een sales-en marketingbedrijf voor CMS-software, we zijn geen dienstenleverancier. We moeten wel goede productarchitecten en techneuten hebben om ons product te ondersteunen naar de markt toe en om het te laten ontwikkelen natuurlijk. Maar onze *core* competentie is ervoor zorgen dat we een software bedrijf zijn die dat CMS-product kan aanbieden.

Op vlak van kosten zijn personeelskosten en kosten voor R&D dan waarschijnlijk de grootste kosten?

Dat klopt.

Zie je op een of andere manier kansen voor de toekomst om kosten te verkleinen? Bvb personeel outsourcen?

Dat is geen doel op zich en is iets dat in onze internationale expansiestrategie sowieso zit. We doen nu ook zaken in China. Het model dat we in België en andere Europese landen hanteerden, hebben we daar op grote schaal, maar in juist dezelfde stappen gehanteerd. Kunnen we een paar klanten lokaal winnen vanop afstand? Kunnen we er een salesorganisatie neerzetten? Of kunnen we er een product consultancy cel neerzetten? Door die organisatie daar neer te zetten, ontstaat er een partnernetwerk. Dat is onze *roll-out* strategie op dat vlak. Een groot deel van onze R&D- en *support*- mensen zitten gegroepeerd in Oekraïne. Dat is maar in mindere mate omdat we daar goedkopere resources vinden, want dat blijft toch niet duren op termijn, maar dat is voornamelijk omdat we daar heel goede, professionele IT'ers vinden. De markt hier is te klein, we vinden in West-Europa in het algemeen niet genoeg specialisten. In de Oost-Europese landen zijn er heel veel goede technische mensen die we kunnen gebruiken. Er zijn nog zulke pools of competences wereldwijd; dat heb je ook in India, in Bangladesh, ook in China op termijn. Nu goed, wij hebben ervoor gekozen om een team uit Oekraïne op te starten. Maar dat is geen uitbestede organisatie, die zijn allemaal werknemer van Sitecore.

Jullie analyse zal uiteindelijk de analyse maken tussen wat vertegenwoordigt een '*out-ofthe-box*' software vendor op dat vlak en hoe zit het ecosysteem van *open source* bedrijven of leveranciers in elkaar.

We merken zelf dat meer en meer open source bedrijven evolueren naar een stukje closed source onderdeel met daarrond dan de community. Ik denk niet dat het omgekeerd ook gebeurt?

Dat zal nooit gebeuren, denk ik. Wat wel al gebeurd is, is dat *solution* bedrijven, '*out-of-thebox*' software leveranciers componenten van de *open source* wereld, integreren in hun product.

Met betrekking tot de evolutie naar onder andere cloud enzovoort, is jullie structuur en jullie manier voorzien op de aanpassingen die nodig zullen zijn om dergelijke wijzigingen te kunnen verwezenlijken? Verwacht je bepaalde evoluties op vlak van bedrijfsstructuur binnen dit en 10 jaar?

Onze structuur is opgebouwd om 18 maanden vooruit te kunnen kijken naar de markt en dan te definiëren wat onze *roadmap* moet zijn, zowel functioneel als technisch. En dan een tweede en een derde *roadmap* op te stellen om te weten 'wat moeten we intern doen om die *roadmap* te realiseren' en 'wat moet ons commercieel model zijn?' Vandaag hebben wij een *pricing* politiek, maar als we bvb over een jaar volledig *cloud-oriented* ons product gaan aanbieden, dan moet onze prijspolitiek wel aangepast zijn. Of misschien ook niet, maar dan moeten we wel kunnen uitleggen waar niet. Of moeten we toch kunnen inspelen met onze *price setting* op de *perceived* meerwaarde van een *cloud* omgeving ten opzichte van een *on-premise* omgeving. Maar dat moet elk bedrijf doen, of je nu auto's of software verkoopt. Op dat vlak zijn wij wel een vrij '*agile*' organisatie om in het zog van onze *roadmap*-plannen ook aanpassingen te doen (ook in onze eigen structuur) en aanpassingen te doen in ons business-model indien dit nodig zou zijn.

Hoe wordt de roadmap samengesteld? Is dit puur vanuit het hoofdkwartier of komen verschillende kantoren uit verschillende landen tot deze roadmap? (Enkel downstream of ook upstream?)

Ik denk dat er een stuk 'bottom up'-informatiegaring is, vanuit alle regio's, vooral vanuit de grote regio's: Amerika, Europa, Verre Oosten... Vooral op vlak van wat zijn de trends en de kritische succesfactoren die we vanuit het partnerkanaal horen? Wat zijn de elementen waar klanten naar vragen die we nog niet beschouwd hebben? Dat gebeurt allemaal in een organisatie die wereldwijd georganiseerd is; productmarketing. Die maakt daar een perceptie van, van hoe de *roadmap* zou moeten zijn de komende 18 maanden. Dat is meestal niet volledig anders dan 3 maanden eerder, maar toch moet er enige evolutie inzitten. Vervolgens beslist het corporate management of dat in orde is en of we dat kunnen bekostigen. Soms zijn er waarschijnlijk dingen die we niet kunnen bekostigen of die we later of sneller willen ontwikkelen, daar is dan een *intern governance process* voor om natuurlijk onze *roadmap* en de enhancements en de impact daarvan voor de organisatie en business modellen te beheren.

Zijn er globaal trends in de internetindustrie? Of merk je verschillen tussen verschillende regio's?

Dit merken we wel in de projecten en de vragen van onze partners. In Amerika en in Engeland worden sneller nieuwere technologieën geadopteerd, vooraleer die op het oude continent worden aangenomen. Een voorbeeld: QR-codes. In het Verre Oosten zijn daar al toepassingen voor. Tesco als Engels bedrijf mocht in de Aziatische landen geen keten openen. Ze hebben dan maar in alle metrostations een grote poster gehangen met daarop wat ze verkopen met QR-codes bij. En mensen in het station kochten in het station op weg naar huis de producten die dan bij hen thuis werden geleverd. Dit is het gevolg van het visionair bezig zijn van een bedrijf uit Engeland of uit Amerika. We merken wel dat de adoptiesnelheid van nieuwe technieken in die landen sneller verloopt dan hier.

Hebben jullie mobiele plannen voor een applicatie waarmee mensen de content voor hun website kunnen aanpassen?

- Hebben we de nodige technologie in ons product in gebouwd om op mobile en tablet devices een website of content te kunnen presenteren? Ja
- 2. Bouwen wij zelf *apps*? Nee, dat is onze *core* competence niet.
- 3. Bouwen onze partners *apps*? Sommige. Wij enabelen uiteindelijk de mogelijkheid met ons product, maar we doen het zelf niet. Er zijn dan ook geen plannen in die richting voor de toekomst.

Zijn bij developer community (de commercialisatie van de modules) de inkomsten volledig voor degene die een dergelijke module ontwikkelt?

Op dat vlak hebben wij geen inkomstenstroom, alles gaat naar de *developers*. We zien dit namelijk niet als onze *core-business* en hebben geen franchisemodel zodat anderen met onze naam mogen verkopen.

Voor ons is productkwaliteit het belangrijkste; een *all-time sustainable functioning platform* in alle omstandigheden!

12. Transcript interview – Tim Geyssens, Umbraco

Participants:

Reinout Denys	Ghent University
Gert Vanhaverbeke	Ghent University
Tim Geyssens	Umbraco
Location:	Raas van Gaverestraat 83
	9000 Ghent
	Belgium
Date:	Thursday, 23 th February 2012
Time:	14.00 - 15.45

Hoe situeren jullie je product in de markt van CMS'en. Is dat een zeer gedifferentieerd publiek of eerder een breed publiek?

Waar wij ons op richten, wij zijn een open source CMS, in de .NET wereld. We gaan vooral richting de .NET developers. In de CMS-wereld, heb je ofwel PHP ofwel .NET, dat zijn de 2 grote werelden. PHP is wel veel groter dan .net. Omdat we nu een .net-framework zijn, richten we ons op die markt. Dus op bedrijven die die technologie willen gebruiken.

En het pakket zelf, is dat toepasbaar op veel vlakken, of is het puur marketing gericht of voor intranetten?

Het wordt voor alles gebruikt. Het kan gaan van iemand die als hobby een site opzet tot intranetten voor grote nieuwswebsites.

Wat is de grootste troef van Umbraco?

De grootste troef is, dat als je Umbraco gebruikt, je niet gaat vastzitten. Maar het gaat wel veel sneller gaan om een site te maken zonder dat je tegen een muur gaat lopen. Je behoudt de flexibiliteit van maatwerk.

Usability dus eigenlijk?

Ja langs één kant, usability van de eindgebruiker, omdat het heel simpel is om te gebruiken. Maar zeker ook voor de developer, hij gaat nergens tegenaan lopen. Hij kan alle technologieën die hij wilt gebruiken integreren.

Er zijn heel wat CMS'en die met modules gaan werken. Is dat bij jullie ook het geval?

Wij hebben eigenlijk een totaal andere architectuur. Dus bij velen is het zo dat je een basispakket installeert, en dan wil je bijvoorbeeld een nieuwsmodule, en dan ga je daar naar op zoek. Maar bij die nieuwsmodule zit je vast aan bepaalde zaken. Bijvoorbeeld elk artikel heeft een titel en foto. Dus die content staat wat vast en als je dat wil aanpassen moet je de broncode aanpassen. Bij Umbraco ga je eigenlijk zeggen 'wat heb ik nodig?' en dan ga je je CMS daarop definiëren.

Dus er is niet echt een basiskern? Iedere CMS is dus verschillend?

Het draait Umbraco natuurlijk, maar de documenten die de eindgebruikers gaan beheren zijn allemaal op maat van.

En dat gebeurt dan via partners, dat jullie de eindgebruiker bereiken?

Wijzelf hebben eigenlijk nooit contact met eindgebruikers, dus met de mensen die de site beheren. Wat wij wel doen, één van onze inkomsten, is opleidingen geven. Aan developers en bedrijven. We hebben dus een partnerprogramma. Vanaf dat een bedrijf 4 certificaten heeft, dus als er 4 mensen de opleiding gevolgd hebben, worden we partners.

En dan zijn het die partners die feedback geven naar ontwikkeling toe van het CMS?

Die feedback krijg je eigenlijk van de community, dat moeten daarom geen partners zijn.

Zijn het dan vooral bedrijven die Umbraco gebruiken, of kan ik zelf ook de gratis versie downloaden en gebruiken?

Iedereen kan het gebruiken eigenlijk. Het zijn wel voornamelijk .net developers die het gebruiken. Dat kan gaan van een student die in zijn vrije tijd een site opmaakt, tot een freelancer die het gebruikt. Maar dat kunnen ook grote bureaus zijn. Dat is zeer breed.

En hoe kun je ongeveer de structuur van Umbraco beschrijven, qua directie, een soort organigram en waar situeer je dan juist de community?

Enerzijds hebben we gewoon onze site waar er gewoon info wordt geboden van welke producten we aanbieden. Dan hebben we ook een community site, waarlangs de meeste communicatie gebeurt. Anderzijds is er nog een bron van inkomsten, het support contract dat bedrijven kunnen aangaan. Dit is een open source CMS. Soms willen bedrijven support, en dat bieden wij dan aan als een product.

Dat gebeurt dan via partners, of dat is Umbraco zelf?

Dat is Umbraco zelf.

Is dat een soort van Enterprise-versie dan?

Neen, de versie is altijd hetzelfde. Het enige wat je kan doen is dat je er support bijkoopt, maar dat is dezelfde versie.

Zijn er dan ook andere partners waar jullie mee samenwerken? Op andere vlakken, dus naast die die je gebruikt om het naar de eindgebruiker te brengen?

Omdat we op Microsoft technologie werken, werken we tamelijk samen met Microsoft ook. Wat er dus soms gebeurt, is dat zij sponsoren om bepaalde features te implementeren. Een paar jaar geleden kwamen ze bijvoorbeeld met een nieuw soort database engine, dat er uit bestond dat je gewoon een file moest uploaden. Ze wouden dan dat wij dat ondersteunden. Maar het is niet zo dat wij alles gaan doen, wat zij willen implementeren.

Ik zal snel iets zeggen over de geschiedenis van Umbraco. Eigenlijk is het begonnen door één persoon in Denemarken. Dat was gewoon een zelfstandige webdeveloper die het systeem voor zichzelf maakte om het dan uit te rollen naar klanten. Hij zat op kantoor met andere mensen en die anderen zagen dat het interessant was, en dan zijn verschillende developers samen beginnen bouwen voor hun klanten. Op een bepaald moment hebben we dan beslist het open source te maken. En zo zijn er altijd meer mensen dat beginnen gebruiken.

En de community, wordt die op een bepaalde manier gemanaged vanuit Umbraco?

Die groeit eigenlijk zelf zonder dat wij er veel aan moeten doen. Die onderhoudt zichzelf.

En wat er ontwikkeld wordt, dat groeit ook vanuit de community zelf van 'dat is er nodig'? Of gebeurt dat vanuit Umbraco?

Dat hangt er een beetje vanaf. Soms kan er iemand met een goed idee komen, die dat dan maakt en het ons voorstelt om in de core toe te voegen. Dan kan dat gebeuren.

En alles wordt gecontroleerd door jullie, de board, of hoe heet dat?

Wij noemen dat de HQ. En dat zijn 12 mensen.

Maar jullie zitten niet wekelijks samen?

Neen, er zijn er 3 in Denemarken, 2 in UK, 2 in VS,... Het is tamelijk verspreid. We kunnen dus niet samen op kantoor zitten. De communicatie gaat via Skype etc.

En heeft iedereen binnen die 12 mensen dan zijn eigen rol in die HQ?

Ja, er is iemand die enkel support doet bijvoorbeeld. Ikzelf ben verantwoordelijk voor een commercieel product waarmee je zelf dingen kunt bouwen waarvoor je geen code moet schrijven.

En alles wat door de community gemaakt wordt komt dan bij de juiste persoon bij jullie terecht?

Op de community site kunnen mensen gewoon hun eigen ding releasen zonder dat wij er moeten tussenkomen. Dat zijn dan standalone zaken die kunnen geïnstalleerd worden.

Zonder controle dan?

Zonder controle ja, maar de community kan er zelf op reageren, je kan punten geven enz. Als we zien dat er echt iets is dat iedereen gebruikt, dan gaan we die persoon contacteren en vragen of we het mogen toevoegen.

En onder welke license werken jullie?

Wij werken onder de MIT-license. Wat eigenlijk betekent dat je er mee mag doen wat je wilt, no strings attached. MIT is copyleft, je mag ervan kopiëren wat je wil. Er zijn zelfs bedrijven die het helemaal gaan branden. Dat mag ook allemaal.

Vind je dat dan geen nadeel, dat zij gewoon jullie werk mogen 'stelen'?

Dat is niet echt stelen, ze rebranden dat.

Daarstraks heb je al enkele inkomstenstromen vermeld, zoals die support, welke zijn er nog?

De CMS zelf is dus volledig gratis. De grootste bron van inkomsten, komt eigenlijk van commerciële add-ons die we doen. Eén daarvan is die formbuilder. Dan hebben we ook nog iets dat het mogelijk maakt om te syncen tussen verschillende omgevingen. We hebben dan ook die supportcontracten. Opleidingen geven we ook. Hier in België doe ik ze om de 6 maanden. Maar er zijn ook over de wereld opleidingen. We hebben trainers die in Umbraco HQ zitten, maar we hebben ook nog externe trainers. Die betalen dan per training die ze doen een bedrag aan ons, dat is dus ook een bron van inkomsten. Verder zijn er ook video-tutorials waar je een abonnement op kan nemen.

En wat gebeurt er met dat geld? Gaat dat allemaal rechtstreeks naar development?

Die producten zijn er om de core mee te helpen, voor core-ontwikkeling. We hebben onlangs bijvoorbeeld het CMS herschreven naar een ander framework. We trekken daar rechtstreeks geen geld uit, maar misschien moeten we die andere inkomsten gaan gebruiken om dat te gaan financieren.

Zijn er bepaalde richtingen die je verwacht waar de CMS-markt naartoe zal gaan?

Wat je ziet is dat mensen nu wel vertrouwen beginnen krijgen in open source. Misschien dat door de crisis mensen geen geld willen geven aan licenties. We zien in elk geval dat in België bijvoorbeeld meer grotere spelers Umbraco gaan gebruiken.

Er zijn ook veel meer voordelen dan de kostprijs aan open source. Gebruiken jullie dat om mensen te overtuigen?

Neen, wij gaan eigenlijk niemand overtuigen. We worden wel uitgenodigd op sommige events om een sessie te geven.

Jullie gaan je partners niet trainen in marketing en dergelijke?

Aan marketing doen we eigenlijk niet.

Maar er zijn waarschijnlijk wel een aantal webbureaus die enkel Umbraco gebruiken? Staan jullie daar dan mee in contact?

Er zijn er een paar waar we nauw mee samenwerken, maar we zijn nog altijd ons eigen bedrijf. De groei van Umbraco is misschien niet zo snel. Maar de mensen die het gaan gebruiken, gebruiken het wel vaak en blijven het gebruiken en zo gaat het verspreiden.

En houden jullie je dan bezig met de klantenretentie? Monitoren jullie op één of andere manier dat de mensen die Umbraco zijn gaan gebruiken dat ze dat blijven gebruiken?

Ik denk niet dat we dat monitoren nee. We zien gewoon hoe onze inkomsten stijgen en daar gaan we ons dan op baseren hoe we verder gaan.

Jullie ontwikkelen niet zo snel, maar jullie zijn toch één van de meest gedownloade CMS'en in de .Net wereld?

Ja, maar je moet rekening houden dat het al 7 jaar bestaat.

Hoe groot is de community?

Ik denk rond de 60.000. Het aantal wereldwijd actieve installaties van Umbraco ligt rond de 400.000. Wat een groot getal is, maar als je kijkt op wereldbasis is dat natuurlijk niet zo'n groot percentage.

Je zegt dat ontwikkeling niet zo snel gaat, hoe komt dat?

De groei gaat niet zo snel, niet de ontwikkeling. De groei is tamelijk organisch. We hebben al veel aanbiedingen gehad voor kapitaal te injecteren, maar we hebben dat nooit aanvaard.

Daarstraks sprak je van modules die geld kosten, is er ook een mogelijkheid dat iemand uit de community die dat ontwikkelt daar ook geld voor gaat vragen?

Ja, we hebben een community site waar mensen hun eigen modules kunnen aanbieden, wij noemen dat packages. Daar is ook een mogelijkheid om je eigen packages aan te bieden. Als ze verkopen via onze site, krijgen wij daar ook een percentage van.

Is er al inbreuk gepleegd op jullie licenses?

Het enige dat we ooit hebben gezien is dat iemand het probeerde te verkopen als zijn eigen CMS. Daar zijn we dan wel wat tegen ingegaan, maar nu kan ons dat eigenlijk weinig schelen, omdat we al die eigenheid hebben.

Als we dan een soort percentage maken, is opleidingen en support dan de grootste inkomstenstroom?

Nee, eigenlijk komt dat van de commerciële pakketten, dat is ongeveer 3/4e.

Heeft Umbraco een specifieke visie t.o.v. de concurrentie

Wij proberen om niet naar de concurrentie te kijken.

Maken jullie bepaalde marketingcampagnes?

Neen, dat gebeurt niet. Wel heeft Microsoft ons ooit gesponsord voor google-ads en dergelijke.

En hoe is de community zo gegroeid? Is dat door de sterkte van het product?

Ja ik denk wel door de sterkte van het product. We zien als mensen het gebruiken, blijven ze het gebruiken. Er is een site 'CMSwatch', die maakt elk jaar een balans op van de CMS wereld, en bepaalde sterktes en zwaktes. Waar wij in uitblonken was de tevredenheid van de developers. Dus ze gebruiken het graag.

Richten jullie je dan meer echt naar developers dan naar eindgebruikers? Dat is toch een rare keuze, want ook al gebruiken developers het graag, als het dan te moeilijk is voor de eindgebruiker...?

We zijn er van uitgegaan om alles zo simpel mogelijk te maken. Kijk maar naar de ontwikkeling die we gedaan hebben voor de eindgebruiker, dat is de laatste jaren vaak gebeurd.

Zijn de functionaliteiten dan ook over het algemeen constant gebleven de laatste jaren?

Voor de eindgebruiker wel ja.

Maar de laatste jaren is er toch veel gebeurd op vlak van sociale media enz., gebeurt dat bij jullie dan trager?

Neen, het grote voordeel is dat Umbraco je niet in een hoek duwt. Als developers sociale media willen integreren, kunnen ze dat perfect doen. Ze hebben die vrijheid om dat te integreren.

Dus als de klant kiest om met één van jullie partners te werken, dan gaat die partner eigenlijk Umbraco actief maken, en zij kiezen dan enkele modules erbij?

Ja, zij gaan dan het CMS op maat maken en dan kan het zijn dat ze vanuit de community nog wat extra's willen.

Het gebruik van die packages, wordt dat ook getraind, of zijn daar tutorials bij en dergelijke?

Als mensen er iets opzetten, dan is er meestal wel wat documentatie bij, of via Twitter of forums.

Ik kan mij inbeelden dat er verschillende mensen aan een module werken met dezelfde functionaliteit, bijvoorbeeld Facebook-integratie. Groeit dat altijd naar één module?

Wat dat wij proberen doen, is als we merken dat verschillende mensen die hetzelfde aan het doen zijn bij elkaar te brengen. Soms lukt dat, soms lukt dat niet. Je kan bijvoorbeeld een package opzetten, en dan kan iemand er een vinkje aanzetten dat die daar ook wilt aan meewerken. Uiteindelijk is dat de bottom line van open source: "Niet het wiel opnieuw gaan heruitvinden". Maar als je dan hoort dat er heel veel developers naast elkaar producten met dezelfde functionaliteiten gaan ontwikkelen, lijkt dat wel inefficiënt. Maar tegelijk worden developers niet betaald natuurlijk. Dat is een beetje het dubbelzinnige en vreemde vanuit economisch standpunt. Developers zijn daar zelf voor gemotiveerd. Is dat ook hoe u begonnen bent?

Ja, ik ben begonnen in een klein webbureau en we waren op zoek naar een CMS. Eerst dachten we zelf één te schrijven, maar ik had al snel door dat dat niet goed was. Ik ben dan zo bij Umbraco uitgekomen en zelf cursussen gaan volgen.

U bent lid van de HQ, hoe geraak je daar dan juist in?

Aan het hoofd staat dus de Deen die Umbraco gestart heeft. Het hangt er een beetje van af hoeveel werk en inkomsten er zijn.

Die HQ is wel betaald?

Ja, dat is een commercieel bedrijf.

Die worden dan 'geheadhunt', of heb jij zelf de stap gezet naar hen?

Neen, die hebben mij benaderd. Ze zien wie er veel ontwikkelt en wie wat kan. Dan moet je geen tijd meer verdoen met trainingen enz.

Dat zijn dan die 12 mensen die werknemer zijn. Maar dus geen ander team en dergelijke?

Neen, dat zijn de enige mensen.

Dat zijn dan de enige kosten?

Ja, tesamen met de servers waar onze eigen site op draait.

Zie je evoluties in de zin van bedrijven die van closed naar open source gaan?

Wat we in Denemarken wel merken, daar is Sitecore onze grootste concurrent. Wat we zien is dat veel bureaus die dat gebruiken, gaan overschakelen naar Umbraco. Hun systeem is gelijkaardig maar veel complexer. De laatste paar jaar merk ik in België dat ik meer aanvragen krijg van grotere bureaus ook. Dan ga ik daar een demo geven, of zij komen naar de opleiding. We hebben nu gepraat over die 4 pijlers van ons business model. Is er volgens jou iets dat we over het hoofd hebben gezien wanneer we praten over BM van Umbraco? Iets wat bij jullie het verschil maakt en dergelijke?

Ik denk vooral die organische groei. Niels (oprichter) is begonnen als freelancer, en heeft Umbraco uitgerold voor zijn eigen klanten. Dan is er die training bijgekomen. Dan deed hij minder eigen projecten, maar meer trainingen. En zo is er altijd een shift geweest. Dan maakten we een commercieel product.

Is er nu al een idee over hoe het er over 2 jaar zal uitzien. Een soort roadmap?

We gaan Umbraco gaan aanbieden als een service in de cloud. Dan kunnen mensen zich registreren, en dan wordt de instantie gestart en dan betalen zij een bedrag aan ons om dat te hosten.

Dus de cloud is iets waar je merkt 'daar moeten we naartoe'?

Het maakt het gemakkelijker. Zeker voor die freelancers die Umbraco gebruiken die dan hun eigen server nog nodig hebben enz., voor hen zal dat wel gemakkelijker zijn. Echt grotere bedrijven, die hebben vaak een eigen IT-team en ik denk niet dat het daar gemakkelijk naartoe zal gaan.

Is dat dan een extra product?

Ja.

Is pricing afhankelijk van de grootte van elk project?

Hoeveel tijd moeten we er zelf insteken, daar gaat het meestal om. Dat valt meestal goed mee. Ons voordeel is dat we nieuwe klanten kunnen aannemen zonder dat we nieuwe mensen moeten aannemen.

Als je support aankoopt, is dat dan beperkt in tijd?

Ja er is één geldig voor drie maanden, en één voor zes.

En die support wordt dan aangevraagd door het webbureau dat dat dan doorrekent naar de klant?

Ja. Dus dan is het webbureau zeker van 'Oh, als er hier een bug is, dan kunnen wij naar Umbraco gaan'. De eindklant gaat nooit een contract aan met ons. Het supportcontract is altijd per domeinnaam, dus een webbureau kan niet met 1 supportcontract verschillende klanten voorzien.

Zijn er bepaalde evoluties geweest in de structuur van Umbraco, in die tijd dat je hier was?

Vooral dat groeien eigenlijk. Ik was de derde die er bij kwam. En 4 jaar later zijn we tot 12 mensen gegroeid.

Zouden er per vakgebied op termijn misschien bepaalde departementen komen, of zou het dan te groot worden?

Ons plan was eigenlijk van niet met meer dan 9 te zijn, en we zijn nu al met 12. Dus het is niet de bedoeling van met veel meer te zijn. Je weet natuurlijk nooit wat het gaat worden, maar dat verwachten we niet. Wij proberen vooral om meer werk aan te kunnen met evenveel man.