FACULDADE de ENGENHARIA
da UNIVERSIDADE do PORTO

Empreendedorismo de Software Livre

Fernando Luís Ferreira de Almeida

Licenciado em Engenharia Informática e Computação pela
Faculdade de Engenharia da Universidade do Porto

Dissertação submetida para satisfação parcial dos requisitos do programa do curso de Mestrado em Inovação e Empreendedorismo Tecnológico

Dissertação realiza sob supervisão do
Doutor José Magalhães Cruz
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Porto, Maio de 2006

Agradecimentos

Um conjunto alargado de pessoas e instituições contribuiu, de uma forma mais ou menos acentuada, para a realização desta dissertação de mestrado. A todos gostaria de expressar o meu agradecimento pela ajuda que me concederam.

Entre todos aqueles que me ajudaram, gostaria de começar por agradecer ao meu orientador Profº José Magalhães Cruz pela disponibilidade com que sempre me atendeu, pela contribuição na elaboração da tese e no estabelecimento de contactos com entidades externas que usam o software livre no seu processo de comercialização.

Aproveito também para agradecer aos vários docentes e director do curso do MIETE (Mestrado em Inovação e Empreendedorismo Tecnológico) o Profº João José Pinto Ferreira. Entre todos os docentes do MIETE que me ajudarem na elaboração desta tese, destacam-se a Profª Aurora Teixeira e o Profº José Mendonça através das várias sessões de inovação no âmbito da disciplina de PIT (Plano de Inovação e Tecnologia).

Por último, aproveito para agradecer a receptividade por parte das empresas contactadas no âmbito da análise de cada um dos casos de estudo. Entre estas empresas destacam-se a Critical Software através do Engº André Barbosa e a PyLabs através do Engº Hugo Ramos.

Resumo

A popularização de software classificado como Software Livre, que é distribuído livremente acompanhado de código fonte, tem gerado grande curiosidade por parte de investigadores e profissionais da área da informática. Esta curiosidade, em parte, deve-se à forma particularmente simples pela qual este software é produzido. Apesar de apresentarem alta taxa de confiabilidade, evolução rápida e uma comunidade produtiva, os projectos de software livre apresentam um processo de desenvolvimento aparentemente caótico, com algumas características particulares: grande descentralização, política liberal em relação a alterações, e discussão aberta sobre todo o assunto técnico pertinente aos projectos.

Os seus modelos económicos de suporte aqui analisados não são normalmente obtidos directamente a partir da venda dos programas, mas são alcançados a partir de modelos económicos indirectos que se baseiam na venda de software adicional, hardware adicional ou serviços adicionais. Paralelamente, a articulação desse software livre com software proprietário não é pacífica, existindo um grande número de licenças no domínio de software livre que são incompatíveis com a utilização de software proprietário, sendo que algumas delas são também incompatíveis entre si.

Nesta dissertação apresentam-se também seis casos de estudo de empresas que usam ou usaram o software livre no seu processo de comercialização. Três deles são casos de sucesso na comercialização deste software (Cyclades, Wild Open Source e Critical Software), enquanto os outros três dizem respeito a casos de insucesso (Eazel, SourceXChange e PyLabs).

Enunciam-se ainda um conjunto de recomendações e procedimentos que se afiguram importantes para a obtenção de sucesso na comercialização deste tipo de software. De entre eles destacam-se uma boa interligação das diversas áreas funcionais da empresa, comprometimento dos recursos humanos com a empresa, sustentabilidade do modelo de negócio indirecto, capacidade e flexibilidade na resposta ao cliente e alterações no mercado e a aposta na internacionalização e nichos de mercados que devem estar alinhados com as competências centrais da empresa.

Abstract

The popularization of software classified as open source, which is distributed freely and with source code, have originated a lot of curiosity from researchers and computer science professionals. This curiosity, in part, is due to the way that this software is developed. Despite the high tax of trustfully, fast evolution and a productive community, the open source projects have apparently a chaotic development process, with very peculiar characteristics: large amount of decentralization, liberal politic of changes, freely discussions about all the technical process pertinent to the projects.

The supportive economic models analyzed in this dissertation aren't usually directly obtained from the source code of programs, but are achieved using some indirect economic models that are based on: selling additional software, selling additional hardware or selling additional services. At the same time, the articulation of this open source software and proprietary software isn't peaceful, existing a high number of licenses inside the open source domain that are incompatible with the use of proprietary software, and some of these open source licenses are incompatible among them too.

In this dissertation are presented six cases of studies of companies that use open source software in their commercialization process. Three of them are success cases on the commercialization of this software (Cyclades, Wild Open Source and Critical Software), while the others three are related with failure cases (Eazel, SourceXChange and PyLabs).

As conclusion, we also enunciate a package of recommendations and procedures that seems important to the success in the commercialization of this kind of software. Among them, it's important to highlight a good interconnection between the several departments and functions of a company, the commitment of human resources with the company, the indirect business model sustainability, the capability and flexibility to react to changes in the external environment and to the client and the commitment with the internationalization and market niches, that must be aligned with the internal core competences of the company.

Índice de Conteúdos

Agradecimentos II
Resumo III
Abstract IV
Índice de Conteúdos V
Índice de Figuras VIII
Índice de Tabelas IX
Abreviaturas X
Glossário XI
1 Introdução 12
1.1 Definição de Software Livre 12
1.2 Enquadramento 14
1.3 Motivação e Objectivos 15
1.4 Estrutura da dissertação 15
2 Inovação em Software Livre 17
2.1 Considerações Introdutórias 17
2.2 Modelo de desenvolvimento do Software Livre 17
2.3 Características do Software Livre 19
2.4 Vantagens e desvantagens do Software Livre 23
2.5 Impactos Culturais nas Organizações 24
3 Modelos Económicos e Análise de Mercado 27
3.1 Considerações Introdutórias 27
3.2 Atributos económicos do software 27
3.3 Cadeia de Valor 29
3.4 Modelos de Negócio 32
3.4.1 Venda de serviços adicionais 32
3.4.2 Venda de software adicional 33
3.4.3 Venda de hardware adicional 33
3.4.4 Factores limitativos 34
4 Aspectos éticos e legais 35
4.1 Considerações Introdutórias 35
4.2 Propriedade Intelectual 35
4.3 Tipo de Licenças 37
4.3.1 BSD 37
4.3.2 GPL 38
4.3.3 LGPL 39
4.3.4 MPL 40
4.4 Pirataria de Software Livre 42
5 Análise de Casos 44
5.1 Considerações Introdutórias 44
5.2 Enquadramento Teórico 44
5.3 Casos de sucesso - Cyclades 48
5.3.1 Enquadramento da empresa 48
5.3.2 Produtos / Serviços 49
5.3.3 Cadeia de Valor 49
5.3.4 Competências Centrais e Vantagens Competitivas 51
5.3.5 Estratégia de Criação de Valor 52
5.3.6 Clientes 53

5.3.7 Conclusão ... 54
5.4 Caso da Wild Open Source ... 55
5.4.1 Enquadramento da empresa ... 55
5.4.2 Produtos / Serviços ... 55
5.4.3 Cadeia de Valor ... 56
5.4.4 Competências Centrais e Vantagens Competitivas ... 57
5.4.5 Estratégia de Criação de Valor ... 58
5.4.6 Clientes ... 58
5.4.7 Conclusão ... 59
5.5 Caso da Critical Software ... 60
5.5.1 Enquadramento da empresa ... 60
5.5.2 Produtos / Serviços ... 60
5.5.3 Cadeia de Valor ... 61
5.5.4 Competências Centrais e Vantagens Competitivas ... 63
5.5.5 Estratégia de Criação de Valor ... 64
5.5.6 Clientes ... 64
5.5.7 Conclusão ... 66
5.6 Caso da Eazel ... 66
5.6.1 Enquadramento da empresa ... 66
5.6.2 Produtos / Serviços ... 67
5.6.3 Análise SWOT ... 67
5.6.4 Competências Centrais e Vantagens Competitivas ... 68
5.6.5 Estratégia de Criação de Valor ... 69
5.6.6 Clientes ... 69
5.6.7 Conclusão ... 70
5.7 Caso da SourceXChange ... 70
5.7.1 Enquadramento da empresa ... 70
5.7.2 Produtos / Serviços ... 71
5.7.3 Análise SWOT ... 71
5.7.4 Competências Centrais e Vantagens Competitivas ... 72
5.7.5 Estratégia de Criação de Valor ... 72
5.7.6 Clientes ... 73
5.7.7 Conclusão ... 73
5.8 Caso da PyLabs ... 74
5.8.1 Enquadramento da empresa ... 74
5.8.2 Produtos / Serviços ... 74
5.8.3 Análise SWOT ... 76
5.8.4 Competências Centrais e Vantagens Competitivas ... 77
5.8.5 Estratégia de Criação de Valor ... 77
5.8.6 Clientes ... 78
5.8.7 Conclusão ... 78
6 Práticas e Regras para o Sucesso Empresarial ... 80
6.1 Considerações Introdutórias ... 80
6.2 Aspectos Genéricos das Organizações ... 80
6.3 Negócios na área da Internet ... 82
6.4 Posicionamento na Cadeia de Valor da Indústria ... 83
6.5 Estratégia e Vantagens Competitivas ... 85
6.6 Capacidades Dinâmicas das Empresas ... 87
6.7 Inovação Empresarial ... 88
6.8 Outsourcing Estratégico ... 90

7 Novas tendências para o Software Livre 91
7.1 Considerações Introdutórias 91
7.2 Limitações do Open Source 91
7.3 Implicações Económicas e Sociais 94
7.4 Visão do Futuro 96
8 Conclusão 99
8.1 Principais Contribuições 99
8.2 Trabalho Futuro 101
9 Bibliografia 103
ANEXOS 107
Anexo A: Licença BSD 108
Anexo B: Licença GPL 109
Anexo C: Licença LGPL 115
Anexo D: Licença MPL 123
Anexo E: Lista dos 20 projectos mais activos no *SourceForge* em Abril de 2006 131
Anexo F: Entrevista da Critical Software 134
Anexo G: Entrevista da PyLabs 136
Anexo H: Contrato de serviços de suporte técnico da Cyclades 138
Anexo I: “Technical data sheet” da edgebox 140

Índice de Figuras

Figura 1: Esquema do funcionamento dos projectos *open source* 18
Figura 2: Motivações dos desenvolvedores de soluções open-source..... 20
Figura3: Classificação dos utilizadores e desenvolvedores do software livre 22
Figura 4: O modelo "iceberg" de uma organização 26
Figura 5: Cadeia de valor do software..... 30
Figura 6: Tipo de estratégias para criação de valor em software livre..... 31
Figura 7: Modelos de negócios indirectos do software livre..... 32
Figura 8: Distribuição das licenças dos projectos *Open Source* sedeados no *SourceForge* 37
Figura 9: Esquema geral da cadeia de valor..... 45
Figura 10: Tipo de estratégias de negócio..... 47
Figura 11: Soluções de out-of-band da Cyclades..... 48
Figura 12: Arquitectura da *edgeBOX* 61
Figura 13: Esquema da aprendizagem adaptativa 88
Figura 14: Esquema da aprendizagem criativa..... 88
Figura 15: Sistema de mercado do software proprietário..... 92
Figura 16: Sistema de mercado do software livre 93

Índice de Tabelas

Tabela 1: Comparação da abordagem tradicional e de open source 25
Tabela 2: Comparação das mais importantes licenças de software open source..... 42
Tabela 3: Tipo de estratégias de negócio 53
Tabela 4: Principais clientes da Critical Software..... 65
Tabela 5: Análise SWOT da Eazel..... 67
Tabela 6: Análise SWOT da SourceXChange..... 71
Tabela 7: Análise SWOT da PyLabs..... 76

Abreviaturas

BPO – Business Process Outsourcing

BSD – Berkeley Software Distribution

CRM – Customer Relationship Management

FSF – Free Software Foundation

GPL – General Public License

LGPL – Lesser / Library General Public License

MPL – Mozilla Public License

NCO – Network Operation Center

OSC – Open Source Community

OSI – Open Source Initiative

ROI – Return on Investment

SWOT – Strenghts, Weaknesses, Opportunities and Threats

TCO – Total Cost of Ownership

TI – Tecnologias de Informação

WAN – Wide Area Network

ZPL – Zope Publish License

Glossário

Benchmarking – processo contínuo de medir produtos, serviços e processos com relação aos concorrentes mais fortes ou aos líderes internacionais reconhecidos no sector.

Best-of-breed – termo em inglês para determinar aquele que é o melhor em um grupo ou uma categoria.

Copyleft – obtido a partir de um trocadilho com a palavra original "copyright". Trata-se de um princípio de que o conhecimento deve ser disponibilizado para todos os que se interessam por ele.

Copyright – termo em inglês que indica o direito de cópia ou propriedade reservada.

e-Business – nome que se dá a processos e transacções realizados electronicamente através da Internet.

Firmware – conjunto de instruções gravadas permanentemente em *hardware*. Os programas de *firmware* correspondem normalmente a tarefas que o computador executa rapidamente.

Outsourcing – contratação a uma entidade externa de uma função ou área de suporte ao negócio da organização.

Player – empresa ou entidade que se encontra que actua e/ou influencia as acções no mercado me que se encontra inserida.

Releases – lançamento de uma versão consolidada de um software.

ROI (Return on Investment) – termo que descreve o cálculo do retorno financeiro em uma política ou iniciativa de negócios que implica algum custo. O ROI pode ser medido e várias maneiras, sendo a mais comum calculada em termos de um período para a recuperação do investimento.

Trademark – protecção de um produto ou bem baseada na sua marca.

Upgrade – pacote de melhoria ou actualização de um determinado parâmetro de um produto.

WAN – rede que utiliza tecnologias de telecomunicações para conectar computadores ou redes sobre longas distâncias.

1 Introdução

1.1 Definição de Software Livre

A noção de software livre não é consensual e existem algumas dúvidas relativamente à sua definição mais apropriada e que convém esclarecer desde o início desta tese de dissertação. Por isso, começar-se-á por apresentar o que é aqui entendido por software proprietário, gratuito, aberto e livre, de acordo com as definições mais aceites.

Nas últimas décadas, as empresas de software têm utilizado, em quase a totalidade dos casos, todos os meios disponíveis para proteger a propriedade intelectual dos seus programas de informática, com o objectivo de impedir o acesso ao respectivo código fonte e impedir as empresas concorrentes de desenvolverem produtos similares. Isso consegue-se, não só não entregando o código fonte da aplicação, como utilizando mecanismos de protecção de propriedade intelectual, como o *copyright* ou as patentes de software, até ao limite que a lei permite. A este tipo de software dá-se o nome de software proprietário.

A diferença entre o software proprietário e gratuito é bastante simples de ser evidenciada, pois essa diferença só existe no processo de comercialização. Quando se cria software proprietário assume-se que se espera obter rentabilidade financeira com o produto, ou seja, tem-se a expectativa que o cliente irá pagar para poder usufruir da sua utilização. No software gratuito o utilizador não terá necessidade de despender qualquer dinheiro na sua obtenção e utilização, mas tal não implica que o utilizador tenha acesso ao código fonte do programa. Aliás, na maior parte dos casos, o utilizador não obtém qualquer acesso ao seu código fonte do software nem é autorizado a realizar qualquer tipo de alterações na aplicação.

No software de código aberto, os mecanismos de propriedade intelectual são utilizados para garantir os direitos de autoria do software, mas sem impedir o seu conhecimento ou limitar seu uso. O código aberto não só garante o acesso ao código fonte, mas também a possibilidade de distribuir a aplicação livremente e utilizá-la para qualquer propósito; inclusive, permite explicitamente alterá-lo e adaptá-lo a qualquer necessidade específica.

Entretanto, quando chegamos ao ponto de distribuição do software existe uma importante discrepância entre movimento de código aberto e de software livre, até agora coincidentes. Enquanto o movimento do código aberto permite que uma aplicação de código desse tipo seja modificada e não exige a publicação do código fonte dessas modificações, o movimento de software livre exige explicitamente que qualquer melhoria realizada e publicada seja disponibilizada de forma idêntica.

A *Free Software Foundation* define que um software pode ser considerado livre quando oferece quatro liberdades fundamentais:

1 Liberdade de execução, por qualquer utilizador e para qualquer uso;

2 Liberdade de estudo do funcionamento de um programa e de adaptação a necessidades específicas;

3 Liberdade de redistribuição de cópias;

4 Liberdade de efectuar melhoramentos e publicação das modificações.

A 1ª liberdade garante liberdade de acção com as ferramentas disponíveis. Assim, por exemplo, evitam-se restrições legais à utilização do software como o *benchmarking* de aplicações que representam um risco legal, muitas vezes desconhecido.

A 2ª liberdade garante a independência relativamente ao fornecedor. Esta independência é possível porque esta liberdade permite a uma empresa corrigir ou adaptar o seu software às suas necessidades. Mais importante do que isso, permite a criação de um mercado de empresas que fornecem serviços de consultoria e manutenção num regime de livre concorrência.

A 3ª liberdade permite o crescimento do parque informático de uma empresa sem aumento dos custos de software.

A 4ª liberdade permite participar num desenvolvimento em cooperação com outros. Esta forma de desenvolvimento permite reduções de custos no desenvolvimento e manutenção do software que são partilhados com outras entidades. Nesta situação as modificações efectuadas à aplicação são tornadas públicas de modo a que toda a comunidade beneficie da melhoria.

No mesmo seguimento, e com vista a esclarecer algumas questões deixadas em aberto pela *Free Software Foundation* na componente de distribuição do software, a *Open Source Initiative* (OSI) estabeleceu dez pontos que a distribuição do software livre deve atender:

1. Redistribuição gratuita;
2. Disponibilização do código fonte;
3. Permissão de se efectuar alterações no código fonte;
4. Preservação da integridade da autoria do código fonte;
5. Não discriminação contra pessoas ou grupos;
6. Não discriminação da sua utilização em qualquer área de negócio ou campo de pesquisa;
7. A licença aplica-se a todos os distribuidores e utilizadores do software;
8. A licença acompanha o software independentemente do conjunto onde ele é distribuído;
9. A licença não pode impedir a utilização de outro tipo de software que o acompanha;
10. A licença deve ser tecnologicamente neutra.

Esta última definição de software livre, descrita pela *Open Source Initiative*, é a que será adoptada ao longo desta dissertação, e será usada de forma indiferenciada a terminologia "open source" e "software livre" quando se refere a este modelo de software.

1.2 Enquadramento

Durante a última década tem surgido um interesse crescente por parte do modelo de software livre e da sua sustentação, não só do ponto de vista científico e social, mas também do estudo da sua viabilidade económica.

A popularização de software classificado como Software Livre, que é distribuído livremente acompanhado de código fonte, tem gerado curiosidade por parte de investigadores e profissionais da área da informática. Esta curiosidade, em parte, deve-se à forma particularmente simples pela qual este software é produzido.

Apesar de apresentarem alta taxa de confiabilidade, evolução rápida, e uma comunidade produtiva, os projectos de software livre apresentam um processo de desenvolvimento aparentemente caótico, com algumas características particulares: total descentralização, política liberal em relação a alterações, e discussão aberta sobre todo o assunto técnico pertinente aos projectos. Os projectos sobrevivem utilizando um mínimo de infra-estrutura disponibilizada pela Internet, sendo o correio electrónico a ferramenta mais utilizada pelos participantes.

Em 1998, Raymond publicou um conjunto de artigos que descrevem informalmente algumas características do processo de desenvolvimento que muitos projectos seguem, as mais importantes sendo: *releases* frequentes, propriedade de código compartilhada e revisão de código massiva (Raymond, 1999). Nestes artigos, Raymond utiliza o termo "bazaar" para descrever a forma de organização e desenvolvimento dos projectos.

Nos últimos anos, paralelamente ao interesse no processo de desenvolvimento do software livre tem surgido um crescente interesse pela área de comercialização deste software. Empresas como IBM, Apple, Sun, Netscape/AOL e Novell têm reconhecido e investido em projectos de software livre. No ano de 2001, foi realizado o primeiro workshop para desenvolvimento *open source* como parte do *International Conference of Software Engineering*.

O interesse da indústria e da comunidade académica é sustentado pelo próprio mercado de software. Em alguns segmentos importantes, projectos de software livre são líderes de mercado. O servidor web mais utilizado actualmente é o Apache, que é software livre. A consultora *JD Edwards* atribui 80% do volume total de e-mail Internet ao *Sendmail*, um programa de software livre para transmissão de correio electrónico. O núcleo do sistema operacional *Linux*, associado a um conjunto enorme de componentes de software livre tem mais de 29 milhões de instalações estimadas mundialmente.

As liberdades garantidas pelo software livre beneficiam a sociedade, evitando a criação de monopólios e barreiras para a liberdade de expressão e de investigação. As suas repercussões sociais, políticas, filosóficas e culturais têm mantido vivo e em constante crescimento o movimento de software livre iniciado por Richard Stallman na década de 1980.

1.3 Motivação e Objectivos

O objectivo principal deste trabalho é encontrar um conjunto de regras e boas práticas indicativas que as empresas que se desejam aventurar no domínio de software livre devam seguir. Este conjunto de boas práticas e caminhos de orientação é obtido a partir da análise de casos de sucesso de empresas que adoptaram o modelo de negócio baseado em software livre e também a partir da análise de casos de insucesso de empresas que adoptaram este mesmo modelo.

Ao longo desta dissertação pretende-se analisar modelos de software livre que uma empresa de software pode adoptar de forma a ter rentabilidade. Um dos resultados a retirar diz respeito à possibilidade de sustentação do seu negócio a médio e longo prazo quando fornece unicamente soluções de software livre ou pelo contrário, a constatação de que o modelo de software livre só poderá ser sustentado quando a empresa se dedicar também à comercialização de software proprietário.

Para além disso, torna-se necessário analisar os diversos tipos de licenças de software livre existentes, as respectivas consequências para a empresa da adopção dessas licenças, o respectivo impacto para o cliente e também as diversas formas de interligação entre software livre e software proprietário e as correspondentes implicações de implementação de soluções mistas.

Por último, pretende-se lançar um olhar sobre o conjunto de empresas que comercializam software livre e quais as vantagens e inconvenientes para a empresa que segue este modelo de comercialização.

1.4 Estrutura da dissertação

Esta dissertação encontra-se dividida em 6 capítulos: Inovação em Software Livre, Modelos Económicos e Análise de Mercado, Aspectos Éticos e Legais, Análise de Casos, Práticas e Regras para o Sucesso Empresarial e Novas Tendências para o Software Livre.

No capítulo *Inovação em Software Livre* é feito um levantamento dos modelos e processos de desenvolvimento do software livre. É realizada uma análise sobre quem cria o software livre, as suas respectivas motivações, e o surgimento de vastas comunidades que alguns dos projectos de software livre possuem.

No capítulo *Modelos Económicos e Análise de Mercado* é realizada uma análise sobre os modelos de negócio que permitem sustentar um negócio de software baseado em software livre. Neste capítulo é também realizado um estudo comparativo da cadeia de valor do software livre face ao software proprietário. Por último, é também realizado um estudo sobre os impactos tecnológicos da utilização do software livre para a própria empresa e para a sociedade em geral.

No capítulo *Aspectos Éticos e Legais* é feito um estudo sobre as normas legais existentes no domínio do software livre através do estudo das suas várias licenças disponíveis. É dada uma especial atenção às questões da utilização do software livre misturado com software proprietário, nomeadamente ao nível dos impactos legais e outros aspectos que é

necessário atender. São também tecidas algumas considerações sobre a propriedade intelectual e sobre a pirataria de software livre.

No capítulo *Análise de Casos* são analisadas diversas empresas que usam o software livre no seu modelo de negócio. Trata-se de empresas que não só usam software livre na sua actividade, como providenciam soluções de *open source* para outras empresas através da sua actividade de comercialização. Neste capítulo são analisados casos de sucesso e insucesso na adopção de software livre por essas empresas.

No capítulo *Práticas e Regras para o Sucesso Empresarial* pretende-se extrair dos casos analisados anteriormente um conjunto de boas práticas e regras que podem conduzir ao sucesso empresarial na área de software livre.

No capítulo *Novas Tendências para o Software Livre* pretende-se obter uma visão sobre o futuro desenvolvimento na área do *open source*, as suas respectivas potencialidades e limitações.

2 Inovação em Software Livre

2.1 Considerações Introdutórias

Neste capítulo apresenta-se uma visão global do modelo de inovação em software livre. Numa primeira instância (Secção 2.2) analisa-se o modelo de desenvolvimento das soluções *open source* e apresenta-se as motivações que estão por base do vasto conjunto de desenvolvedores destas soluções, analisando-se o processo de criação de comunidades e redes de conhecimento. Seguidamente (Secção 2.3) define-se um conjunto de características que são comuns às soluções de software livre. Posteriormente (Secção 2.4) analisa-se algumas das vantagens e inconvenientes das soluções *open source*. No final (Secção 2.5) disseca-se alguns dos principais impactos nas organizações que advém da utilização do modelo de negócio baseado em software livre.

2.2 Modelo de desenvolvimento do Software Livre

O desenvolvimento de soluções *open source* pode ter lugar nos mais variados contextos organizacionais (universidades, empresas, institutos de I&D, …). Como regra, os desenvolvedores de software livre juntam-se voluntariamente a um projecto e, dependendo da sua contribuição, podem estar mais ou menos envolvidos nele e na sua estrutura de decisão. O surgimento destes projectos e a sua organização emerge de uma forma espontânea. Do mesmo modo, a participação nestes projectos é voluntária e não pretende tirar quaisquer contrapartidas financeiras.

Contrariamente ao modelo tradicional de desenvolvimento de software, os projectos *open source* não começam a partir da identificação de uma necessidade do mercado, mas surgem, na grande maioria dos casos, como uma ideia ou um problema específico que o desenvolvedor tem. Um projecto de *open source* começa pela publicação do seu respectivo objectivo ou fundamento e pela publicação do seu código fonte inicial. Se o projecto se mostrar interessante para a comunidade, mais desenvolvedores aparecem e se lhe juntam. Com o crescimento do projecto, as suas estruturas de suporte tendem também a ficar mais complexas, e surge então o que se chama "núcleo de desenvolvedores", constituídos pelos responsáveis pela maior fatia de contributo para o projecto e que tomam as decisões estratégicas relativamente ao seu futuro desenvolvimento.

Tanto dentro do núcleo de desenvolvedores como fora dele existem muitas formas possíveis de participação no desenvolvimento desse software: o simples teste do programa ao desenvolvimento de novas funções no código fonte, a correcção de "bugs", etc. A comunicação no interior destas equipas dá-se através do uso do correio electrónico, listas de discussão e grupos de notícias, sendo normalmente abertas ou moderadas.

Eric Raymond (1999) no livro "The Cathedral and the Bazaar" apresentou uma visão sobre o modelo de desenvolvimento do software livre. Nele, o processo de desenvolvimento do *open source* é comparado com um "bazaar", enquanto que o software

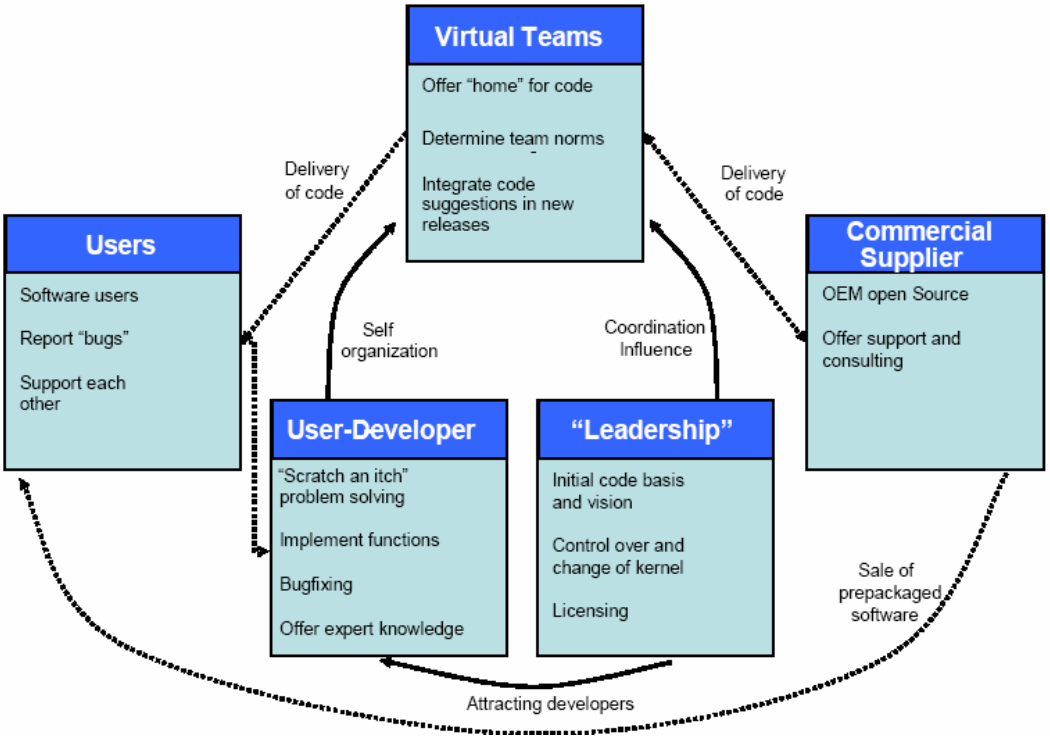
proprietário se assemelha ao processo de construção de uma catedral. Neste livro o software livre é descrito por Eric Raymond (1999) como:

> "…great babbling bazaar of differing agendas and approaches out of which a coherent and stable system could seemingly emerge only by a succession of miracles."

Por outro lado, o software proprietário é entendido como a "catedral" numa metáfora à sua divisão hierárquica, autoridade formal e à rigidez no estabelecimento de funções pré estabelecidas.

Contudo, este modelo de associação voluntário do software livre conduz muitas vezes ao risco de os projectos não terem uma continuidade assegurada. Se os desenvolvedores tornaram estes novos projectos como unicamente seus e tentarem somente atender ao seu ponto de vista sobre os mesmos, os projectos podem ser abandonados por falta de suporte da própria comunidade. Como exemplo, dos programas correntemente registados no sourceforge.net, apenas 16% deles atingem um estado de desenvolvimento designado por estável e apenas 2% atingem o estado de estável. (no sítio web "sourceforge.net", 2005).

Figura 1: Esquema do funcionamento dos projectos *open source*



Fonte: Boston Consulting Group (2002)

A figura 1 apresenta o processo de desenvolvimento de uma solução *open source* como sendo estritamente modular com tarefas bem definidas. Na prática um mesmo utilizador pode desempenhar vários papéis simultaneamente ao longo do ciclo de desenvolvimento de um produto.

Segundo (Kooths e Langenfurth, 2003), não existe qualquer canal de "feedback" do mercado que permita escutar e perceber as necessidades dos consumidores contrariamente ao software proprietário que se encontra especialmente vocacionado para atender às necessidades do público alvo. Segundo o mesmo autor, no software livre só tem oportunidade de introduzir novas ideias quem se tornar desenvolvedor.

Contudo, e segundo a minha própria opinião, esta afirmação não traduz verdadeiramente o que se passa no software *open source*, já que a maior parte dos desenvolvedores são público-alvo. Quem quiser dar ideias e sugestões é livre de o fazer e não raramente são tidas em conta. Contudo, a contabilização destas contribuições é extremamente difícil de se fazer, uma vez que não são feitos estudos de mercado nesta área.

2.3 Características do Software Livre

Os projectos de software livre possuem muitas características comuns entre si. As características abaixo são consideradas como atributos chave por estarem presentes em grande parte dos projectos *open source*.

Comunidade

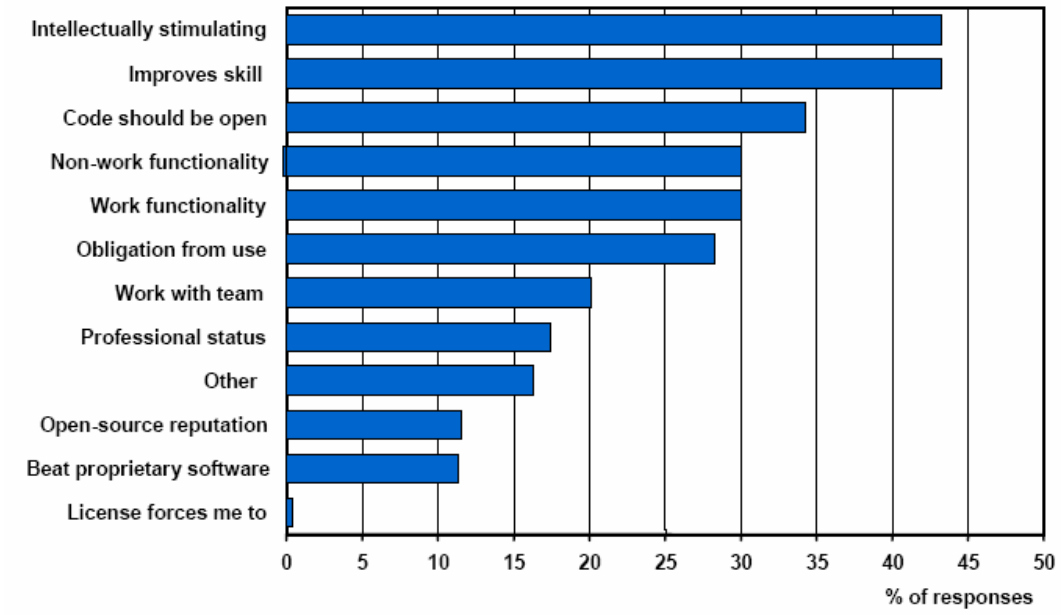
Todos os projectos de software livre de sucesso têm de ter uma comunidade bem definida e com interesses comuns. Torna-se necessário que exista o envolvimento desta comunidade em torno da utilização e desenvolvimento do produto (software).

Motivação

Uma das grandes questões por detrás do modelo *open source* diz respeito à motivação que leva um vasto conjunto de pessoas a interessar-se por este tipo de software. Existem diferentes tipos de contribuidores: individuais e empresas. A nível individual temos motivações ligadas a satisfação pessoal, investigação, projectos científicos e progressão profissional. As empresas normalmente envolvem-se com o intuito de conquistar um novo segmento de mercado, obter novas vantagens competitivas relativamente aos seus concorrentes e gerar novos produtos, que não poderiam ser desenvolvidos apenas internamente.

Em 2002, o *Boston Consulting Group* conduziu um estudo para análise das motivações dos desenvolvedores registados no "SourceForge.org". Os resultados obtidos estão ilustrados na figura 2:

Figura 2: Motivações dos desenvolvedores de soluções open-source



Fonte: Bóston Consulting Group (2002)

Da análise da figura 2 podemos ver que os dois mais importantes factores de motivação são o estímulo intelectual e aprofundamento de conhecimentos. Daí que não seja de estranhar que a maioria dos desenvolvedores da comunidade *open source* estejam ligados à área académica. Para além desses, merece destaque o facto de muito desse software ser destinado a uso pessoal e não unicamente de trabalho (*Non-work functinality versus work functionality*).

Desenvolvedores são também utilizadores

O conjunto das pessoas que contribuem para a geração do código fonte são também utilizadores do código produzido. Isto significa que o grupo de desenvolvedores é um subconjunto da comunidade "open source". Desta forma, desenvolvedores do software livre são também utilizadores, mas nem todos os utilizadores são desenvolvedores.

Processo de aceitação de contribuições

Um projecto *open source* envolve o recebimento de submissões dos mais variados âmbitos e sobre os mais diversos aspectos de um projecto. A forma mais comum de contributos acontece ao nível da correcção de erros, documentação e casos de teste. Existem também casos em que os contributos podem ser exclusivamente monetários, que são bem-vindos na maior parte dos casos, pois permitem sustentar as infra-estruturas dos projectos.

Ciclos de desenvolvimento

Novos melhoramentos numa solução *open source* podem-se manifestar de dois modos: inovações radicais no software ou inovações contínuas. As inovações radicais contemplam o aparecimento de novas funcionalidades. As inovações contínuas, dizem respeito ao

melhoramento de uma funcionalidade já existente na versão anterior e corresponde à consolidação e aperfeiçoamento dessa mesma funcionalidade.

Modularidade

Este vector é imprescindível no software *open source* de forma a permitir a colaboração remota de diferentes intervenientes. Os módulos devem permitir a sua posterior integração numa solução de conjunto mais vasta.

Contudo, se existem muitas características comuns ao processo de desenvolvimento de soluções *open source*, existem ainda mais características que variam de projecto para projecto. As mais importantes variáveis a serem consideradas encontram-se descritas abaixo.

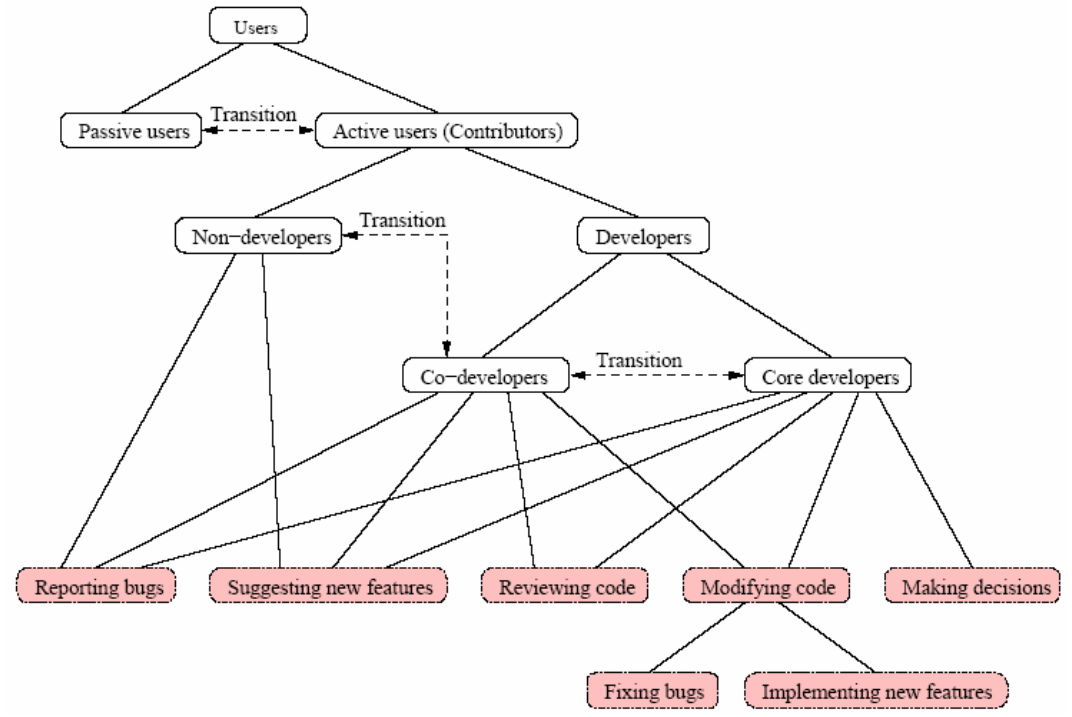
Escolha da área de trabalho

Não existem à partida áreas restritas de actuação do software *open source*, mas o seu processo de desenvolvimento depende unicamente da motivação da própria comunidade que o envolve e o suporta.

Balanceamento centralizado e descentralizado

A organização de comunidades dentro de um software *open source* pode ser muito variada. Existem projectos em que existe uma hierarquia mais rígida entre os diversos níveis de desenvolvedores (figura 3) e existem outros em que esta estrutura é praticamente inexistente. As hierarquias mais rígidas conduzem a uma maior centralização do poder de decisão sobre esse projecto.

Figura 3: Classificação dos utilizadores e desenvolvedores do software livre



Fonte: Gacek, C. , Lawrie, T. e Arief, B. (2004)

Por exemplo, da análise da figura 3 podemos verificar que os "core developers" têm um maior poder de decisão sobre o rumo a ser seguido da solução *open source* e no seu código do que os "co-developers".

Cultura do mérito

O modelo de desenvolvimento do software *open source* baseia-se na livre percepção do mérito de cada um dos desenvolvedores para o projecto. Esta informação é pública e é visível para toda a comunidade. Do mesmo modo, utilizadores passivos de uma solução *open source* podem passar a utilizadores activos através de contribuições efectuadas para o projecto.

Modelo de negócio

Dependendo do domínio e do interesse suscitado pela comunidade, os projectos *open source* podem motivar o envolvimento de empresas comerciais, investigadores, desenvolvedores individuais e utilizadores. Contudo, não existe necessariamente um modelo à partida para a distribuição deste tipo de software e esta é uma das situações que será analisada ao longo do capítulo 3 desta dissertação.

Documentação e teste

A documentação e teste são importantes aspectos a ter em conta no processo de desenvolvimento do software livre. Boa documentação permite que as pessoas mais facilmente o utilizem e para ele possam contribuir. O nível e extensão da documentação é

grandemente variável, sendo por vezes extensa e clara, e outras vezes inexistente ou dispersa por vários locais.

O processo de testes permite que os utilizadores e desenvolvedores tenham uma maior confiança no software que se encontram a usar e desenvolver e validam o seu processo de codificação.

Licenciamento

O processo de distribuição e comercialização do software livre pode contemplar diversas formas de licenciamento. Os tipos possíveis de licenciamento encontram-se descritos detalhadamente no capítulo 4 desta dissertação de mestrado.

2.4 Vantagens e desvantagens do Software Livre

Usualmente, a primeira percepção que temos da vantagem do software *open source* diz respeito ao facto de ele ser disponibilizado de forma gratuita ou com um baixo custo. Contudo, esta característica não é exclusiva do software *open source*, já que algum software proprietário é também disponibilizado de forma semelhante (ex: navegador web Internet Explorer da Microsoft). O que de facto distingue o software livre do software proprietário é a combinação das características descritas na secção 2.3. Todas elas combinadas produzem um conjunto de impactos e sinergias que podem constituir a superioridade do modelo *open source* em muitas circunstâncias.

As características descritas na secção 2.3 permitem a concretização das seguintes vantagens:

- Disponibilidade do código fonte e a liberdade de o poder alterar. Permite que se façam diversas melhorias e alterações sobre o código, de forma a adaptá-lo a um novo hardware ou a novas necessidades das organizações;
- Liberdade de redistribuir modificações e melhoramentos efectuados no código. Permite que estas novas alterações sejam divulgadas por uma larga comunidade de potenciais utilizadores;
- Liberdade de usar o software de qualquer forma. Possibilita que o software seja usado por uma larga comunidade de utilizadores e em inúmeras situações;
- Não existe ninguém no topo da hierarquia que possa restringir o uso do software dentro dos limites da licença. Evita-se, por exemplo, a situação em que a empresa detentora dos direitos do software proprietário não queira lançar uma nova versão de "upgrade" para uma determinada versão mais antiga;
- Não existe dependência de uma única entidade sobre a qual o futuro dessa solução de *open source* dependa unicamente. Esta é uma preocupação que existe ao nível do software proprietário, uma vez que existem situações em que uma empresa fecha e não dá continuidade a um produto;

- Não existência de "caixas pretas". A existência do código fonte disponível permite inspeccionar e avaliar a correcção, eficiência e segurança da aplicação de um determinado algoritmo;
- Possibilidade de aparecimento de "forking" no desenvolvimento de soluções *open source*. Um "fork" é uma subdivisão do código base em duas partes distintas, cada uma delas geridas por grupos diferentes. Esta ocorrência pode ser interpretada com uma vantagem, uma vez que permite atender às necessidades específicas de um determinado grupo de desenvolvedores, mas pode também ser entendida com uma desvantagem, pois existe uma divisão de esforços por ambos os projectos;
- O elevado nível de cooperação dentro de um projecto *open source*. Isto permite que o desenvolvimento da solução seja compartilhado aumentando a rapidez do processo e tornando mais fácil e rápido o trabalho de cada um dos intervenientes.

Existem contudo alguns problemas no modelo do software livre que convém atender e dos quais uma empresa deve estar consciente antes de adoptar o software livre:

- Não existem garantias de que o desenvolvimento do projecto vai manter-se consistente. Por outras palavras, não é possível antever à partida quando e se um projecto vai chegar ao seu estado de maturidade;
- Pode haver problemas significativos relacionados com o nível da protecção da propriedade intelectual, caso não se tenha especial cuidado com as diversas licenças que acompanham o software. Este ponto é especialmente importante numa altura em que alguns países começaram a aceitar o patenteamento de software, ideias e algoritmos;
- Pode haver dificuldade na obtenção de informação sobre a existência de um determinado projecto *open source* e do seu respectivo estado de desenvolvimento. Esta situação acontece, pois não existe uma grande divulgação do tipo de soluções *open source* existentes e existem dificuldades em criar um espaço único no qual se tente disponibilizar todas as soluções *open source* nos mais variados âmbitos.

2.5 Impactos Culturais nas Organizações

A adopção do software livre por parte de uma organização levanta algumas questões e problemas a que convém dar especial atenção, de forma a se poder com eles lidar. Neus e Scherf (2005) afirmaram o seguinte:

"…Based on our experience, we believe that very few challenges are the result of limits in technology, tools or process – rather, they are often the result of limits and boundaries in people's minds."

O paradigma do modelo *open source* assume que o resultado do trabalho produzido é resultado de um processo massivo de colaboração. A abordagem *open source* acredita na evolução e melhoramentos segundo uma série de versões e lançamentos, tirando vantagens do baixo custo dos mecanismos electrónicos de comunicação.

Este paradigma do modelo *open source* é totalmente oposto ao que acontece nos modelos tradicionais de colaboração, como se pode ver na tabela 1.

Tabela 1: Comparação da abordagem tradicional e de *open source*

Abordagem Tradicional	Abordagem Open Source
Brooks' Law	Linus' Law
Hierarchy	Network
Experts	Peers
Teams	Communities
Cathedral	Bazaar
Perfection	Improvement
Construction	Evolution

Fonte: Neus, A. e Scherf, P. (2005)

Da análise da tabela 1 podemos ver que o software proprietário baseia-se numa metodologia de desenvolvimento designada por Brooks' Law, enquanto que o software *open source* segue a metodologia de desenvolvimento Linus' Law.

Brooks em 1975, com base na sua vasta experiência na gestão de grandes projectos de software, sugeriu que a complexidade de coordenação dos projectos cresce de uma forma exponencial à medida que se aumenta o número de colaboradores envolvidos nesses projectos. Como dizia Eric Raymond "…Brooks' Law predicts that a Project with thousands of contributors ought to be a flaky, unstable mess,…" (Raymond, 1999).

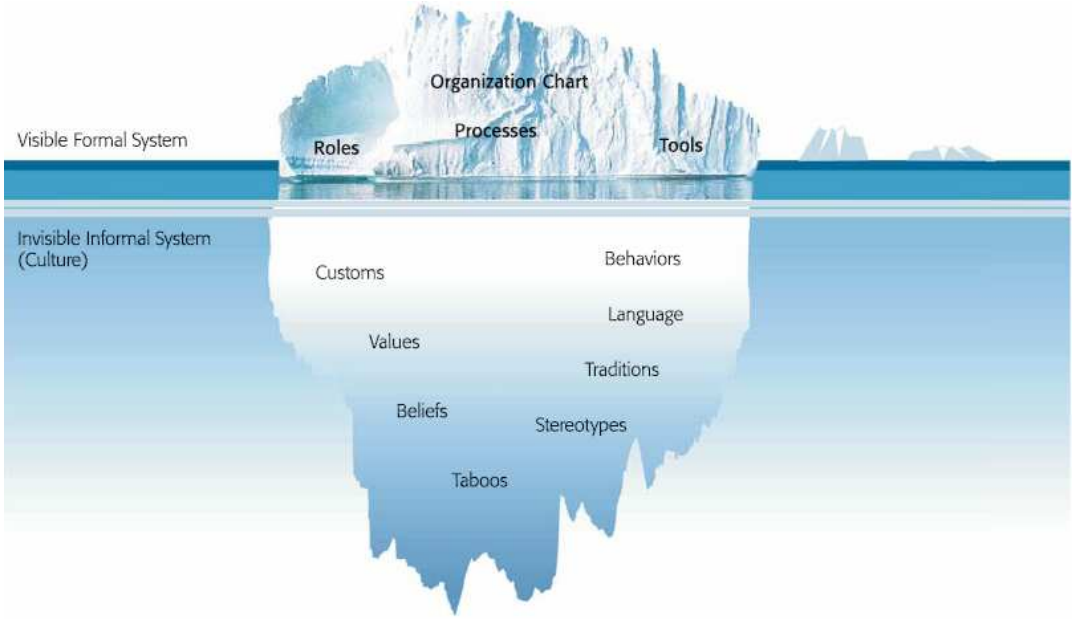
Do outro lado, a Linus' Law entende que "the key to the success of the collaborative-development model is based on the lower transaction cost of information, enabling the separation of the identification and solution components of quality problems and spreading these tasks over a much larger population…" (Raymond, 1999). Desta forma, o software *open source* baseia-se num processo de construção iterativo, no qual o seu processo de evolução se baseia no lançamento de uma série de novas contribuições, tirando partido para tal do actual custo praticamente negligenciável dos meios de comunicação electrónica.

Uma mudança de perspectiva da abordagem tradicional de colaboração para uma abordagem *open source* requer uma mudança também ao nível das mentes das pessoas.

As ferramentas, processos, papéis e sistemas organizacionais encontram-se disponíveis gratuitamente na Internet. Contudo, estes métodos não podem ser directamente transpostos para uma dada organização, sem que para tal se tenha em conta a sua própria cultura.

A parte visível da organização diz respeito aos seus modelos de organização formais, processos e tarefas. Em suma, é um tipo de organização designada por "iceberg" (figura 4). A cultura da organização identifica o que se chama "abaixo da linha do iceberg".

Figura 4: O modelo "iceberg" de uma organização



Fonte: Neus, A. E Scherf, P. (2005)

A cultura é um aspecto complexo do sistema que contempla as crenças, valores, linguagem, costumes, comportamento e artefactos que um membro de um grupo usa na interacção com esse mesmo grupo. Este aspecto cultural tem uma forte influência na maneira de pensar e de agir e nas consequentes decisões realizadas pelos indivíduos. A cultura é o resultado de um longo período de aprendizagem através da observação, interacção e experiência.

Conclui-se, por isso, que a adopção do modelo *open source* pode ser bem sucedida numa dada organização e não o ser em outra.

3 Modelos Económicos e Análise de Mercado

3.1 Considerações Introdutórias

Neste capítulo apresenta-se os principais modelos económicos que permitem rentabilizar o modelo de negócio baseado em software livre. Inicialmente (Secção 3.2) analisa-se os principais atributos económicos do software e verifica-se o impacto da utilização do software livre para cada um deles. Seguidamente (Secção 3.3) analisa-se a cadeia de valor do software e efectua-se um estudo comparativo da cadeia de valor do software livre e do software proprietário. Por último (Secção 3.4) indica-se alguns modelos de negócios possíveis para o software livre.

3.2 Atributos económicos do software

O software possui um conjunto de custos, que lhe estão inerentes, quer se opte pela utilização do software livre ou software proprietário. Contudo, para a organização que o vai usar, o impacto de cada um destes custos não é mesmo, caso a escolha do software recaia sobre um ou outro modelo.

Mas antes de prosseguir torna-se importante apresentar os seguintes dois conceitos:

- Custo total de posse de software (TCO) – representa o cálculo dos custos tecnológicos de implementação e manutenção do software ao longo do tempo;
- Custo estratégico – o custo estratégico tem em conta outros factores para além dos custos que podem ser directamente imputados à tecnologia. Por exemplo, tem em conta factores, como: impacto na produtividade dos colaboradores, acessibilidade à informação e valor estratégico que traz para a organização.

O TCO é calculado com base nos custos totais de implementação de uma solução tecnológica. Isto inclui os custos iniciais de aquisição do software (preço de aquisição ou licenciamento), custos de hardware, custos de instalação, custos de treinamento do utilizador final, custos de manutenção do software (valores anuais de manutenção e suporte e custos de *upgrade*).

Para além do TCO torna-se importante considerar os custos estratégicos. Este tipo de valor é difícil de quantificar, mas tem-se tornado cada vez mais importante no processo de decisão. Deve ter em conta situações nas quais o software proprietário não é uma alternativa para a implementação da solução, quer através de critérios de segurança e robustez quer através de personalizações muito específicas que têm que ser garantidas pela empresa.

O mais óbvio critério no qual o software livre leva uma inequívoca vantagem verifica-se em termos dos custos de aquisição, custos de manutenção e custos de *upgrade*. O software livre é, em grande parte dos casos, disponibilizado gratuitamente, não existindo à partida valores de manutenção a serem pagos e os *upgrades* são também geralmente gratuitos. Contudo, existem outros conjuntos de custos (custos de consultadoria, treinamento dos

colaboradores, administrativos, etc.) que convém considerar e que podem originar que o software livre não seja necessariamente menos dispendioso em termos de TCO que o software comercial.

Custos de aquisição do software

Os custos de aquisição do software proprietário são muito distintos entre si. Algum software proprietário, como ferramentas simples ou pequenas aplicações, têm custos muito baixos de aquisição. Outros, porém, como complexos sistemas de gestão de bases de dados, pacotes de soluções financeiras ou sistemas operativos têm custos muito altos de aquisição.

Outro factor que se deve ter em mente é o facto de em alguns casos se tornar necessário pagar por pequenas soluções complementares à ferramenta principal, o qual não se verifica no software open source. Por exemplo, o *Microsoft Windows Server* é uma ferramenta de automatização de "backups" que tem que ser paga individualmente, enquanto que no sistema Linux as ferramentas de gestão automática do "backup" são incluídas no pacote global.

Custos de instalação

Para algum software, a instalação é um processo bastante simples e intuitivo, não demorando mais de 10 a 30 minutos. Algumas soluções mais complexas (pacotes financeiros, soluções de gestão de clientes, …) podem demorar alguns dias a serem instaladas e configuradas, nomeadamente se necessitarem de migração de dados de um sistema mais antigo.

No caso do software open source convém estar atento ao facto de muitos deles serem mais difíceis de ser instalados do que o software proprietário. Esta situação não advém do facto de estas soluções serem mais complexas, mas da falta de documentação e suporte técnico aquando da instalação.

Por último, um outro aspecto importante a ter em conta diz respeito à necessidade de uma empresa em ter especialistas para o suporte da aplicação, porque nem sempre é fácil encontrar especialistas que estejam familiarizados com soluções open source.

Custos de hardware

Em grande parte das situações, as soluções de software são implementadas sobre hardware existente, daí que não existam quaisquer encargos adicionais com hardware. Contudo, caso se esteja a implementar uma rede de trabalho pela primeira vez, torna-se necessário definir também a estrutura dessa rede. Por exemplo, estudos provam que servidores Linux conseguem lidar com um maior volume de tráfego do que o Microsoft Windows, o que implica que se efectuam poupanças no hardware quando se utiliza o Linux. (Horowitz, 2004)

Custos de treinamento

Para soluções destinadas ao cliente final (aplicações de escritório, pacotes financeiros,…) o custo de treinamento do pessoal é o custo mais significativo na implementação de uma nova solução tecnológica.

A maior parte dos colaboradores não envolvidos directamente com a área tecnológica não estão familiarizados com soluções fora do espectro Windows. Desta forma, torna-se necessário fornecer formação específica a esses colaboradores.

Despesas de manutenção

Os custos de manutenção são comuns a algumas soluções de software proprietário. Estas despesas são calculadas anualmente como sendo uma percentagem do custo de aquisição. Estes valores são também imputados no cálculo do TCO.

As soluções de *open source* normalmente não têm este custo de manutenção associado. Contudo algumas empresas de distribuição do Linux (como a RedHat) têm um custo de manutenção anual usualmente associado ao suporte da aplicação.

Custos de *upgrade*

Os custos de *upgrade* dependem muito do fabricante do software e do número de cópias instaladas desse software. Em alguns casos estas soluções de *upgrade* são disponibilizadas gratuitamente ou têm um custo mais acessível por unidade, para quem tiver um maior conjunto de soluções instaladas.

No domínio do *open source* estas soluções são disponibilizadas gratuitamente e podem ser obtidas fazendo o *download* da versão de *upgrade* e posterior instalação num número qualquer de postos de trabalho.

Custos administrativos e de suporte

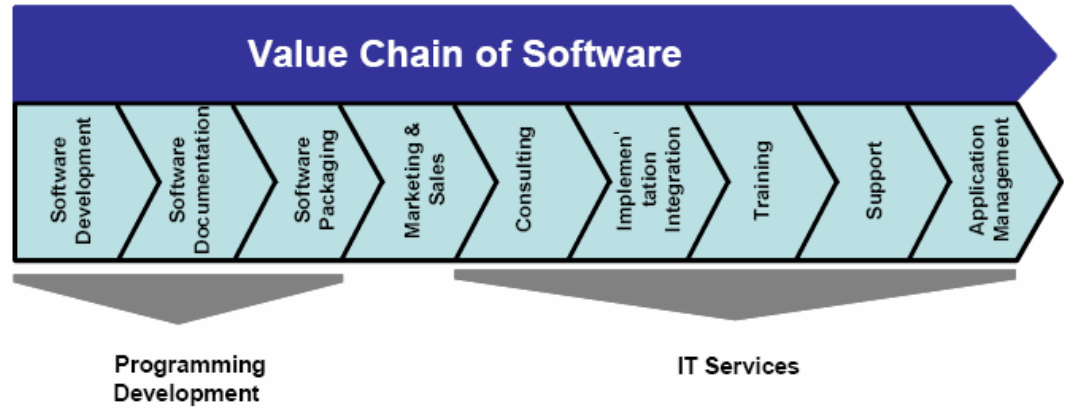
Todo o software (aplicações, base de dados, sistemas operativos) requer alguma forma de suporte administrativo de algum tipo. Em alguns casos, torna-se necessário ter colaboradores que providenciem suporte, em outros casos são contratados consultores.

As características do software determinam o nível de suporte necessário, como por exemplo: a vulnerabilidade a problemas de segurança e o nível de complexidade da utilização dessa ferramenta para a sua administração e para os seus utilizadores finais.

3.3 Cadeia de Valor

As diferenças no processo de desenvolvimento do software proprietário e *open source* conduzem a mudanças na forma como os dois modelos cobrem a cadeia de valor do software.

Figura 5: Cadeia de valor do software



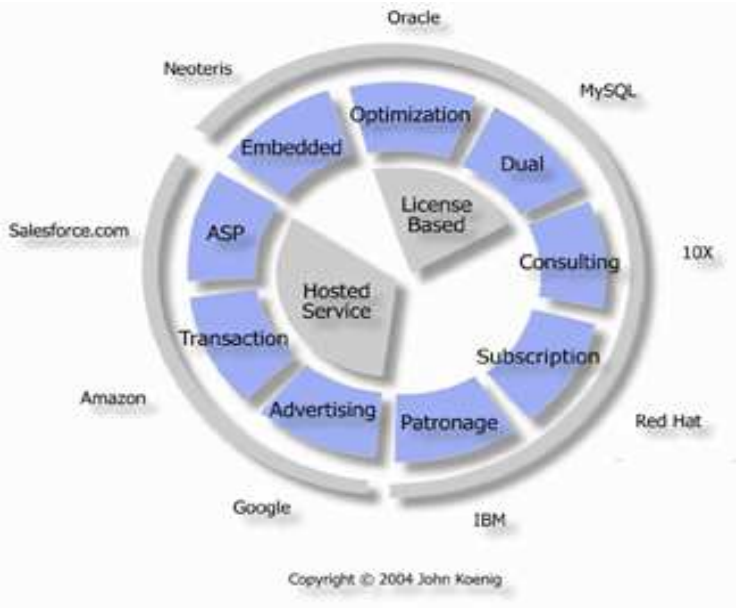
Fonte: Berlecon Research (2002)

Na cadeia de valor (ver figura 5), o desenvolvimento do software é seguido pelos serviços de software, no qual se incluem a consultadoria, a implementação, o suporte e a gestão da aplicação.

Existem diferenças nos modelos de cadeia de valor do software proprietário e do software livre. Todos os elementos descritos na figura acima podem ser encontrados em ambos os casos; contudo, a dimensão e a importância de cada um deles pode ser significativamente diferente. Para além disso, no caso do software livre não é possível gerar valor a partir desses elementos individuais da cadeia de valor, mas o seu valor só pode ser obtido da combinação desses factores.

Existem diversas estratégias no domínio de software *open source* que podem ser usadas com o intuito de criar valor para um produto, atrair clientes e gerar receita. Em 2004, John Koening apresentou sete estratégias que foram usadas com sucesso por diversas empresas.

Figura 6: Tipo de estratégias para criação de valor em software livre



Fonte: Koening (2004)

Da análise da figura 6 podemos ver que o Koening considerou as seguintes estratégias:

- Optimização – garantia de melhor desempenho de serviços usando para tal uma solução no domínio do software livre;
- Dual – são oferecidos dois tipos de soluções: uma solução no domínio pública e outra que é taxável. A solução do domínio pública contém um conjunto limitado de funcionalidades;
- Consultoria – são oferecidos serviços de consultoria no domínio de soluções *open source,* aproveitando o facto de 70% dos custos de software para uma empresa se centrarem na área de implementação da solução (Mckinsey Consulting, 1999);
- Subscrição – existência de uma subscrição pela utilização e suporte da respectiva solução;
- Patrocínio – estratégia que visa patrocinar diversos projectos *open source* numa área estratégica para a empresa, de forma a tirar vantagens competitivas no médio e longo prazo. Usualmente estas vantagens competitivas centram-se ao nível da inovação, criação de *standards* ou obtenção de mais valias competitivas a jusante da cadeia de valor da empresa;
- Hospedagem – as empresas servem-se do software *open source* para minimizar os custos da utilização da sua estrutura tecnológica de base. A obtenção de valor não é obtida através da comercialização do software, mas através de diversos serviços que se centram em outras áreas de negócio.
- Integração – utilização do software livre como forma de potenciar a criação de mais valias, em termos de portabilidade, desempenho, rapidez de implementação de outras soluções que são lhe são integradas e que podem ser vistas como complementares.

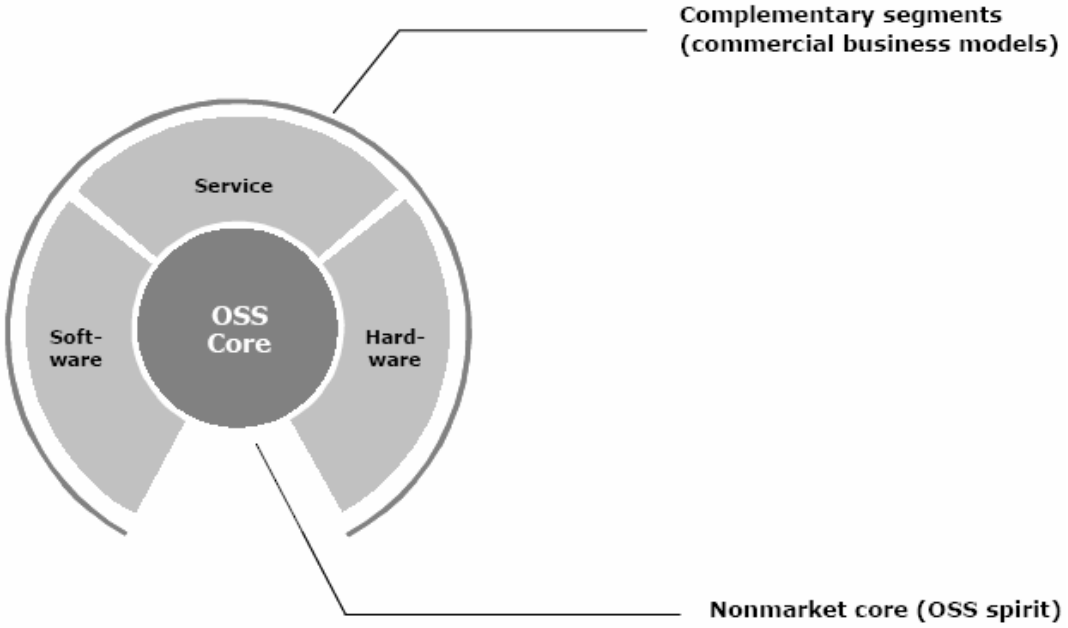
3.4 Modelos de Negócio

Um importante factor no desenvolvimento do software livre é a prevenção de que alguém possa ter direitos sobre esse software e o possa usar comercialmente. O facto de o software livre poder ser legalmente copiado e efectuado o seu *download* a partir de locais públicos impede que seja explorado comercialmente.

Contudo, modelos comerciais de exploração do software livre têm emergido, no qual o software livre é usado como produto base, ao qual são adicionados serviços ou produtos. Do ponto de vista económico, investimentos em software livre só são rentáveis para companhias que consigam gerar lucro com outros produtos ou serviços acoplados à solução base.

Os modelos de negócio baseados em software livre são muitas vezes descritos como modelos de negócio indirectos, porque as vendas não são geradas directamente do produto actual, mas a partir de serviços e produtos adicionais fornecidos.

Figura 7: Modelos de negócios indirectos do software livre



Fonte: Schimidt e Schnitzer (2002)

A figura 7 mostra as diferentes formas de se conceber soluções comerciais baseadas no modelo de software livre. Aí é realizada uma distinção entre os modelos que ofereçam serviços adicionais, software adicional e hardware adicional.

3.4.1 Venda de serviços adicionais

Grande parte dos distribuidores de software livre oferecem um conjunto de serviços adicionais que são adicionados à solução base de *open source*. A empresa acrescenta valor a uma ferramenta de *open source* através da disponibilização de serviços adicionais, como consultadoria e manutenção.

Neste modelo existem duas variantes. A primeira diz respeito ao facto de uma companhia poder actuar como um consultor pago e desta forma definir qual a melhor solução que se adequa à empresa. Neste esquema, qualquer companhia pode actuar como consultor, já que não existe qualquer limitação em termos de competências técnicas e de experiência no trabalho com soluções *open source*. A segunda opção contempla a participação da empresa no desenvolvimento da solução *open source*. Esta empresa pode actuar como consultora e fornecedora de serviços adicionais, mas possui uma inequívoca vantagem relativamente à opção anterior. Esta vantagem diz respeito ao tempo necessário para personalizar e adaptar esta solução face às reais necessidades dos clientes, uma vez que esta empresa já tem um conjunto de experiência e capacidade técnica que um seu concorrente não possui.

3.4.2 Venda de software adicional

Os modelos de negócio baseados na venda adicional de software usam o *open source* como um ponto de partida base para a venda de produtos de software proprietário e oferta de serviços de suporte. As empresas que implementam este tipo de modelo de negócio têm normalmente uma relação muito próxima com projectos *open source* específicos. Em muitos casos, esta empresa assume posições de iniciativa dentro do projecto e/ou assumem importantes funções no seu desenvolvimento, existindo o caso em que colaboradores desta empresa são também os principais desenvolvedores das solução *open source* (Ex: Sun e OpenOffice).

Contudo, quanto maior for o foco comercial da empresa, maior será a dificuldade em atrair um conjunto de desenvolvedores voluntários, já que os interesses comerciais desta empresa não vão ao encontro dos interesses da comunidade de desenvolvedores, que contribuem de forma espontânea e gratuita. Um exemplo desta situação ocorreu com a Netscape.

Os aspectos básicos deste modelo consistem em desenvolver novas funcionalidades ou módulos sobre um software *open source* que se encontra acessível a toda a comunidade. Este tipo de estratégia permite disponibilizar um software *open source* para a comunidade e explorar comercialmente este novo software, que contempla o conjunto de funcionalidades adicionais. Em algumas situações, esta versão comercial é quase idêntica à versão livre, e tal é perfeitamente legítimo, desde que esta solução livre possa ser usada gratuitamente, e que não implique o investimento adicional numa solução proprietária (Ex: Ghostscript e Alladin).

3.4.3 Venda de hardware adicional

Uma outra alternativa consiste em usar projectos *open source* para promover a venda de hardware adicional da própria empresa. A IBM é um exemplo bem conhecido de tal estratégia. A IBM não é unicamente uma empresa de hardware, mas também um dos maiores produtores de software. Por sua conta, a IBM investiu 1 bilião de dólares americanos em vários projectos *open source* no ano 2001, o qual incluiu a adaptação do Linux e do Apache a diferentes plataformas de hardware IBM (Schimidt e Schnitzer, 2002).

3.4.4 Factores limitativos

Os modelos de negócios baseados em software livre dirigem-se a um mercado aberto e concorrencial, no qual já se encontram as soluções proprietárias. As empresas que investem em modelos de negócio *open source* só se podem tornar rentáveis do ponto de vista financeiro se conseguirem oferecer um conjunto de serviços e produtos que possam ser financiados pelo próprio mercado, ou seja, que possa ser traduzível em número e valor de vendas.

Um dos factores limitativos do sucesso de um modelo de negócio baseado em software livre diz respeito ao seu processo de desenvolvimento e à sua adesão no interior de uma comunidade: o objectivo da empresa é conseguir implementar um conjunto vasto e difícil de reproduzir de serviços e produtos adicionais; em contrapartida, o objectivo da comunidade de desenvolvimento *open source* é manter esse conjunto o mais possível dentro do fórum de software livre.

À medida que existe um crescente interesse por parte das empresa em tornar comerciável este produto, a comunidade científica tende a abandonar este projecto. Isto de facto, origina que sejam necessários por parte da empresa maiores recursos humanos alocados ao projecto e o estabelecimento de uma equipa de desenvolvimento permanente. Contudo, é preciso ter em atenção que este trabalho da equipa de desenvolvimento fica também disponível para toda a comunidade, daí que seja necessário tomar medidas que restrinjam o acesso livre à restante comunidade. Uma destas medidas diz respeito à escolha criteriosa do processo de licenciamento desta solução.

Outro factor limitativo diz respeito ao facto de a solução base *open source* se encontrar disponível para toda a comunidade. Nesta comunidade, existem todo o tipo de *players*, incluindo concorrentes que também podem facilmente construir produtos e serviços adicionais sobre esta solução base. Desta forma, o processo de replicação de uma solução torna-se mais simples de ser realizável do que numa solução totalmente comercial.

Kooths e Langenfurth (2003) afirmaram que:

> "…if a business model is based on an open-source product, the market position of that product can easily be contested by imitator (fast erosion of pioneering advantages)…"

A criação de barreiras à entrada de novos concorrentes fica desta forma mais fragilizada, já que o ciclo de desenvolvimento do produto é reduzido drasticamente. O facto de a empresa apoiar a vantagem competitiva unicamente na solução de software pode torná-la mais vulnerável aos seus concorrentes. Esta vantagem competitiva não deve advir unicamente do software, mas deve ser uma mistura de outros factores tais como: serviço ao cliente, política de marca, I&D, etc.

4 Aspectos éticos e legais

4.1 Considerações Introdutórias

Neste capítulo apresenta-se os principais modelos de licenciamento do software livre. Inicialmente (Secção 4.1) expõe-se a problemática da propriedade intelectual. Seguidamente (Secção 4.2) apresenta-se os principais modelos de licenciamento do software livre, as suas potencialidades e respectivas limitações. Para além disso, aborda-se a problemática da escolha do melhor modelo de licenciamento pelo qual uma dada organização deve optar e quais são os problemas e compatibilidade da utilização conjunta de diversos modelos de licenciamento. Por último (Secção 4.3) analisa-se o impacto da pirataria no software livre e compara-se este mesmo impacto com o caso do software comercial.

4.2 Propriedade Intelectual

A questão em torno da propriedade intelectual é um dos assuntos mais debatidos junto da comunidade de desenvolvedores de software *open source*. Existem quatro mecanismos previstos na legislação internacional para proteger a propriedade intelectual, dos quais apenas três (*copyright*, patentes e *trademarks*) são aplicáveis ao software livre. O quarto elemento (*trade secret*) não é adequado ao modelo de software livre, uma vez que se baseia na manutenção em segredo do código fonte, que é exactamente o oposto ao modelo de acesso ao código fonte do software livre.

Neste sentido passemos a apresentar o que se entende ao longo desta dissertação por *copyright*, patentes e *trademarks*.

Os direitos de autor ("copyright") referem-se às criações do espírito, que podem ser do domínio literário, científico ou artístico, por qualquer modo exteriorizada. Esta protecção acontece ao nível da expressão do pensamento e não de ideias. Para adquirir um direito de autor não é preciso cumprir nenhum tipo de formalidade, é apenas necessário criar uma obra.

Uma patente de inovação representa um título jurídico que é atribuído ao inventor por uma entidade competente, que no caso de Portugal é o *Instituto Nacional da Propriedade Intelectual* (INPI). Esse título confere ao titular um direito exclusivo de explorar a invenção, impedir um terceiro de explorar, fabricar e utilizar sem consentimento a invenção durante um determinado período de tempo (actualmente definido em 20 anos) e num determinado território onde foi solicitada a protecção.

O registo de uma marca tem por finalidade proteger um nome do produto e não uma invenção. A marca pode ser constituída de diversas formas, nomeadamente através dos recursos a conjuntos de sinais, nomes de pessoas, números ou sons. Contudo, esta marca terá obrigatoriamente que ser constituída por dois requisitos:

- Susceptibilidade de representação gráfica;

- Capacidade distintiva.

O registo de uma marca dá ao seu titular um direito exclusivo que permite impedir terceiros de, sem o consentimento do titular, usar no exercício de actividades económicas qualquer sinal igual, ou semelhante, em produtos ou serviços idênticos ou afins àqueles para os quais a marca foi registada (no sítio web pmelink, 2005).

A questão do *copyright* no software livre pode ser bastante ambígua. Segundo o *United States Copyright Office,* o copyright encontra-se associado à pessoa que escreveu o código fonte ou que reescreveu um código fonte antigo, cuja versão é diferente da do original. Contudo, com múltiplos actores no desenvolvimento de uma solução *open source* o seu mecanismo de *copyright* encontra-se também distribuído, o que vem colocar novos desafios ao modelo de *open source* baseado exclusivamente nas leis do copyright. Podemos pensar em três situações típicas:

- Distribuição do desenvolvimento sem coordenação – neste caso, cada um dos intervenientes possui os direitos de copyright da sua contribuição;
- Desenvolvimento controlado e centralizado – neste caso, os direitos de cada contribuidor são vistos como um todo;
- Reescrita de um trabalho existente – neste caso, quem efectuou a reescrita do código fonte possui os novos direitos de copyright sobre o trabalho efectuado.

De facto, as licenças de *open source*, de uma forma ou de outra, usam as leis do copyright "standard", de forma a construir licenças de software *open source* que têm em conta as preocupações em torno da comunidade de software livre (Vadén, 2003). Estas preocupações centram-se especialmente nos seguintes 4 níveis:

- Não permitir o uso do seu código em software proprietário, já que os desenvolvedores lançaram e publicaram o seu código para todos usarem. Neste caso, o uso do código é visto como um acto de confiança, já que se alguém quiser usá-lo terá que jogar com as mesmas regras;
- Proteger a identidade de autoria do código. As pessoas têm um grande orgulho no seu trabalho e não querem que alguma outra pessoa retire o nome do seu respectivo autor e que assuma perante outros a autoria desse trabalho;
- Distribuição de código fonte. Muitas licenças de software livre forçam a distribuição do código fonte. Isto protege o utilizador permitindo que eles personalizem o software conforme as suas necessidades;
- Forçar qualquer trabalho que englobe o seu a usar a mesma licença.

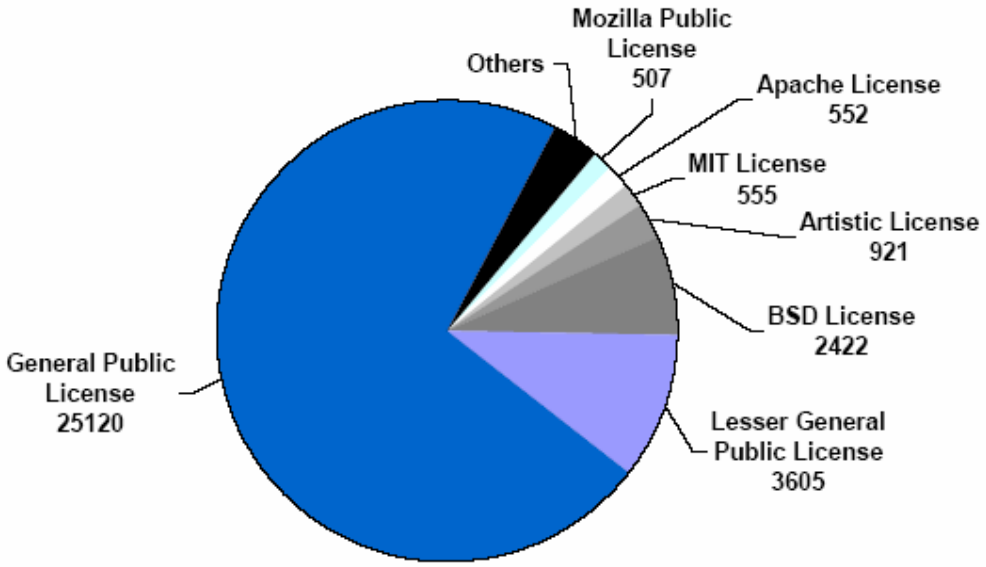
No caso desta dissertação não serão considerados os outros modelos de protecção do software aplicadas ao software livre, respectivamente as patentes e *trademarks*. Estes métodos são similares ao que acontece no caso do software proprietário, os quais já foram apresentados genericamente na *secção 4.2* desta dissertação.

4.3 Tipo de Licenças

Existem um grande número de licenças de software livre, grande parte delas muito similares. No sítio web "http://www.opensource.org/" pode ser consultado a lista completa de licenças de software livre.

No âmbito deste trabalho apenas são apresentadas as licenças mais amplamente divulgadas e usadas em projectos *open source*. No ano de 2004 a distribuição de tipos de licenças pelos projectos *open source* sedeados no *SourceForge* está representada na Figura 8.

Figura 8: Distribuição das licenças dos projectos *Open Source* sedeados no *SourceForge*



Fonte: Berlios (2004)

Como se pode ver a licença General Public License (GPL) detém mais de 50% do conjunto, sendo que a LGPL e a BSD são as licenças que lhe seguem, mas a uma significativa distância.

4.3.1 BSD

A licença BSD (*Berkeley Software Distribution*) é uma das mais usadas na distribuição de software livre. Aproximadamente 7% dos projectos *open source* licenciados no repositório *Sourceforge* usam a licença BSD. É a terceira licença mais popular, logo a seguir à GPL e LGPL. Estas três licenças juntas detêm mais de 80% do mercado de licenciamento do software livre. Este modelo de licença é utilizado em muitos programas, inclusive uma versão do Unix e o próprio software BSD, que deu origem à licença.

A universidade da Califórnia em Berkeley tem uma larga história de pioneirismo na concepção de modelos de distribuição e de desenvolvimento de software. A BSD surgiu no interior deste ambiente tecnológico e de inovação no início da década de 80, sendo a

mais antiga licença das licenças de *open source*. Posteriormente surgiu a licença BSD modificada que veio retirar a "cláusula de propaganda" que fazia alusão ao facto de o software ter sido desenvolvido pela universidade da Califórnia em Berkeley.

A licença BSD é uma das mais curtas existentes no domínio das licenças *open source* e é uma das mais simples de ser interpretada sem que surjam inconsistências jurídicas na sua análise (OSS Watch, 2005).

Em termos sucintos, a licença BSD permite:

- Uso, cópia e distribuição de código fonte ou executável não modificado do programa licenciado;
- Uso, cópia e distribuição de código fonte ou executável modificado do programa licenciado.

Esta licença exige que sejam respeitadas as seguintes situações:

- Todas as cópias distribuídas são acompanhadas da respectiva licença;
- O nome ou referências a antigos contribuidores não são usados para promover quaisquer modificações efectuadas no programa sem o consentimento deles.

Em suma, a licença BSD permite que os programas originários de software livres sejam distribuídos como proprietários. O autor de software pode inclusive comercializar a nova versão do programa, cobrando pela sua utilização.

A versão do BSD aqui apresentada corresponde à licença modificada BSD e a mais comummente usada. Esta licença é compatível com a GNU GPL, mas o mesmo não acontece com a licença BSD original devido à sua "cláusula de propaganda".

4.3.2 GPL

A licença GPL (*General Public License*) é a mais usada como modelo de licenciamento *open source*. Mais de dois terços dos projectos no repositório *Sourceforge* usam este tipo de licença.

O projecto GNU (significa *GNU's Not Unix*) teve o seu início em 1984, quando Richard Stallman, um programador que trabalhava no *Massachusetts Institute of Technology*, decidiu criar e promover a criação de software que pudesse ser distribuído e adaptado livremente. Esta ambição deparou-se com grandes problemas, uma vez que a cultura no mercado até essa altura se baseava unicamente na obtenção de dividendos financeiros do código de software, sendo o software uma indústria altamente rentável.

Stallman discordava desta visão do mercado e pretendia criar uma licença que permitisse a redistribuição e utilização do código fonte. Em 1985 Stallman fundou a *Free Software Foundation* (FSF), uma organização sem fins lucrativos que tinha por objectivo criar suporte ao projectos no âmbito de software livre como o GNU e formulou licenças de software livre que fossem de encontro às necessidades e desejos do movimento de software livre.

A licença GPL, tal como qualquer licença, estabelece um conjunto de direitos mediante certas restrições. A licença GPL permite a:

- Cópia e distribuição do código fonte de programas não modificados (Ponto 1);
- Modificação do código fonte do programa e distribuição do código fonte desse novo programa (Ponto 2);
- Distribuição de versões compiladas do programa, quer modificados quer não modificadas (Ponto 3).

Por outro lado pretende assegurar as seguintes situações:

- Todas as cópias distribuídas (modificadas ou não) são acompanhadas da respectiva identificação de copyright e exclusões da sua utilização (Ponto 1 e 2);
- Todas as cópias modificadas são distribuídas segundo a licença GPL (Ponto 2);
- Todas as versões compiladas do programa são acompanhadas do respectivo código fonte ou é garantida a sua disponibilização em domínio público.

Em suma, a GPL possui um mecanismo conhecido como *copyleft* (o oposto de copyright) que exige que qualquer programa desenvolvido a partir de um software livre também deve ser licenciado em GPL, garantindo a abertura do código.

4.3.3 LGPL

A LGPL (*Lesser ou Library General Public License*) é uma licença idêntica à sua licença mãe GPL que lhe deu origem. Ela foi escrita em 1991 por Richard Stallman tendo sido posteriormente actualizada por Eben Moglen em 1999. Trata-se essencialmente de uma versão menos restritiva que a GPL, que permite ao programador criar aplicações proprietárias utilizando bibliotecas de código livre.

O objectivo desta licença é preservar a possibilidade de bibliotecas públicas poderem ser usadas em software que não é do domínio público. Qualquer distribuição segundo uma licença LGPL deve conter a seguinte informação:

- Código fonte do trabalho que usa a biblioteca;
- Incluir uma facilidade que permita que o trabalho que usa a biblioteca possa continuar a sê-lo com versões modificadas dessa mesma biblioteca, ou seja esta biblioteca deve manter a mesma interface.

Esta última facilidade é extremamente importante, pois permite o desenvolvimento de programas "semi abertos" onde parte do código do programa está disponível e o restante das funções é proporcionada por alguns binários que não possuem seu código aberto. Estes binários são ligados aos programas no momento da sua compilação, de uma forma análoga às *Dynamic Link Libraries* (DLL) do Windows (Laurent, 2004).

Na prática, este tipo de licença permite que o programador aproveite códigos livres para criar um programa fechado.

4.3.4 MPL

A licença MPL (*Mozilla Public License*) é considerada como uma licença que se situa no meio termo entre a rigidez da GPL e da tolerância da licença BSD (OSS Watch, 2005). Não é ainda tão usada como a licença GPL ou BSD, mas é bastante flexível e análises recentes no domínio das licenças existentes permite-nos concluir que se tem tornado cada vez mais popular.

A origem desta licença encontra-se associada à empresa *Netscape Communications*, quando, no início de 1998, decidiu abrir à comunidade o seu navegador web (*Netscape Communicator*). Esta decisão baseou-se no facto de a empresa ter perdido grande parte da quota do mercado para a Microsoft (*Internet Explorer*), apesar de já ter sido a líder no mercado. Neste sentido, a empresa decidiu divulgar o seu código fonte mediante uma licença *open source*.

Contudo, este processo não foi simples uma vez que certas partes do seu código encontravam-se licenciadas a entidades terceiras e noutros casos estavam sujeitas a certas restrições (como por exemplo, o seu módulo de criptografia que não podia ser livremente exportado para fora dos Estados Unidos). Neste sentido, a *Netscape Communications* criou uma nova licença designada por NPL, que posteriormente viria dar origem à MPL (*Mozilla Public License*). Esta nova licença permitiria ir de encontro aos objectivos da *Netscape*, que pretendia que a comunidade de desenvolvedores *open source* efectuasse modificações no seu código fonte e que pudessem ser relicenciadas e usadas na sua solução comercial.

Ao contrário das outras licenças, a MPL divide os direitos das licenças em duas secções – uma dizendo respeito aos direitos do autor inicial e outras estabelecendo direitos para as outras pessoas que adicionaram código à solução inicial.

Os autores iniciais do projecto possuem os seguintes direitos:

- Uso, reprodução, modificação, visualização, sub-licenciamento e distribuição do código fonte original e de versões modificados desse código fonte;
- Criação de patentes sobre o código fonte original;
- Distribuição do trabalho que contenha o código original e licenciamento desse novo código da forma que o distribuidor quiser.

Os contribuidores, segundo o modelo de licenciamento MPL, possuem os seguintes direitos:

- Uso, reprodução, modificação, visualização e sub-licenciamento e distribuição do código fonte das suas modificações;
- Criação de patentes sobre o código fonte das suas modificações e de todo o trabalho, ou seja, contém o código original mais as modificações introduzidas;
- Distribuição do trabalho que contenha as suas modificações ao original e licenciamento desse novo código da forma que o distribuidor quiser.

Contudo, estes direitos encontram-se sujeitos às seguintes condições:

- Todas as cópias distribuídas (originais ou modificadas) devem incluir o seu respectivo código fonte ou indicar a forma como o pode obter;
- Todas as modificações realizadas são descritas e acompanhadas por documentação;
- Qualquer direito de patentes incluídos estão perfeitamente descritos na documentação de suporte;
- Todas as cópias do código (originais ou modificadas) tem os respectivos direitos de copyright em anexo;
- Todo o código modificado deve ser distribuído segundo a licença MPL.

O MPL é considerado um modelo liberal em termos de *copyleft*, tendo este modelo já sido adoptado por algumas empresas e adaptado por outras para criar as suas próprias licenças, como foi o caso da *Sun Microsystems*, que a usou como base para criar a *Common Development and Distribution License* para o sistema operativo *OpenSolaris* (OSS Watch, 2005).

O MPL é uma licença incompatível com a GNU GPL, ou seja, um módulo coberto pela GPL e um módulo coberto pela MPL não podem ser usados conjuntamente.

De seguida apresentam-se as principias implicações da escolha de cada uma das licenças.

Tabela 2: Comparação das mais importantes licenças de software *open source*

Licença	Ligação com software proprietário	Distribuição do "trabalho"	Redistribuição de alterações	Compatível com GNU GPL
GPL	Não permitido (se houver incorporação numa solução de conjunto).	Não permitido (com software cuja licença não seja compatível com a GNU GPL).	Permitido (só se o trabalho derivado for licenciado segundo GNU GPL).	Sim.
LGPL	Permitido (desde que o software que liga à biblioteca não for considerado um trabalho derivado).	Permitido (mas sujeito a restrições, tal como a disponibilização do código fonte da biblioteca e suas eventuais alterações).	Permitido (só se o trabalho derivado for licenciado segundo a licença LGPL ou GPL).	Sim.
BSD	Permitido.	Permitido.	Permitido.	Sim (mas unicamente no caso da versão modificada BSD).
MPL	Permitido.	Permitido.	Permitido (mas só segundo o modelo de licenciamento MPL).	Não.

Fonte: Rusin, Z. (2006), no sítio web http://gnu.org (2006) e no sítio web http://opensource.org (2006).

4.4 Pirataria de Software Livre

A pirataria de software prejudica a todos, desde desenvolvedores de software, proprietários de lojas de distribuição e até os próprios consumidores do produto. A replicação e a distribuição ilegal de software têm um impacto significativo sobre a economia, resultando em prejuízos enormes aos países, reduzindo empregos e originando evasões de divisas e impostos, recursos que poderiam ser revertidos para a própria população (no *site web* Microsoft, 2005)

As empresas que são vítimas da queda de facturação em decorrência da pirataria são obrigadas a cortar despesas inicialmente destinadas ao desenvolvimento de novas tecnologias e forçadas a reverter seus recursos financeiros na protecção de seu software e de seus utilizadores legítimos.

Contudo, as empresas de software e de distribuição não são as únicas afectadas pela pirataria em geral, mas os próprios consumidores são também vítimas de todo este processo, em termos das seguintes variáveis:

- Qualidade – O software pirata geralmente está desprovido dos principais recursos e documentação, não proporcionando aos utilizadores quaisquer garantias ou opções de actualização;

- Vírus – os meios de distribuição que não são devidamente testados poderão ser infectados com vírus que poderão danificar o disco rígido do utilizador ou contaminar a sua rede, para além de não contarem com os inerentes benefícios de suporte técnico do software original;
- Riscos para as empresas – a cópia ou o uso de software copiado ilegalmente no local de trabalho coloca toda a empresa sob risco de infringir as leis de direito de autor.

De acordo com a *Business Software Alliance* (Price Water Coopers, 2003), 36% do software instalado em computadores do mundo inteiro foram pirateados em 2003, embora essa taxa seja ainda pior em alguns países.

Convém nesta altura fazer a distinção do que se entende por pirataria de software livre em relação ao software proprietário. A pirataria de software livre infringe do mesmo conjunto de problemas para o utilizador final, que não se encontra preocupado com o desenvolvimento desse software. Acontece porém que a comunidade de desenvolvimento do software livre pode ser muito alargada e não se encontra restrita ao ambiente da própria empresa, daí que se coloquem maiores dificuldades no estabelecimento dos direitos de autor de um dado software. Assim, é possível a um desenvolvedor embutir parte ou a totalidade do software livre em um pacote de software proprietário, sem que sejam respeitadas as respectivas normas de licenciamento.

No software proprietário, os problemas de pirataria encontram-se associados quase integralmente à existência de cópias não autorizadas. Com o software livre, isto não é um problema; pelo contrário, é até pedido que se difundam as cópias. Segundo Richard Stallman, a pirataria com software livre é quando algum programador ou empresa se apropria do código fonte do programa para incorporá-lo em algum outro programa que não seja livre (Stallman,1999). Desta forma, a questão da pirataria em software livre põe-se no desrespeito pelas licenças que acompanham o software. É neste contexto que a definição de Richard Stallman se aplica.

O próprio impacto da pirataria no software livre não é consensual dentro da comunidade de desenvolvimento *open source*. Segundo (Amadeu, 2005) a pirataria fortalece o monopólio, uma vez que permite divulgar um programa e criar dependência desse programa para o utilizador.

Segundo o movimento de software livre do Brasil o software livre pode ser visto como uma oportunidade para combater a pirataria. Neste caso, o software livre é uma oportunidade para todos aqueles que não podem pagar pelos altos custos de uma licença de software e permite um maior acesso à informação dentro da legalidade.

5 Análise de Casos

5.1 Considerações Introdutórias

Neste capítulo apresentam-se alguns casos de estudo sobre empresas que utilizam o software livre no seu processo de comercialização. São apresentados três casos de sucesso (Cyclades, Wild Open Source e Critical Software) e três casos de insucesso (Eazel, SourceXChange e PyLabs) do processo de comercialização deste tipo de software. Inicialmente (Secção 5.2) faz-se o respectivo enquadramento teórico da abordagem e conceitos utilizados na análise de cada um dos casos, que são apresentados na (Secção 5.3 a 5.8). Para cada caso de estudo, faz-se uma breve introdução sobre a área de negócio de cada uma das empresas; seguidamente apresentam-se os principais produtos e serviços disponibilizados por cada uma das empresas; posteriormente constrói-se a sua cadeia de valor e identificam-se as suas competências centrais que servem de base à sustentação das suas vantagens competitivas. Estas vantagens competitivas permitem delinear a estratégia de cada uma das empresas; seguidamente apresentam-se ainda os principais clientes de cada uma das empresas; por último extraem-se as conclusões mais importantes. Todos estes casos estudados têm uma organização semelhante de forma a facilitar o seu estudo comparativo e para facilitar a extracção de um conjunto de regras e práticas que são posteriormente revistas no capítulo 6 desta tese de dissertação.

5.2 Enquadramento Teórico

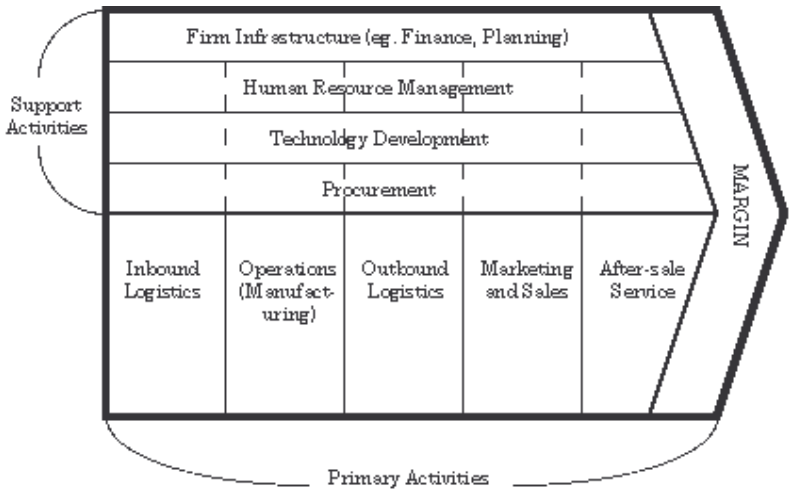
A abordagem utilizada consistiu na análise da cadeia de valor e identificação das competências centrais e vantagens competitivas de cada uma das empresas segundo estabelecido por Porter nos inícios da década de 90.

A análise da cadeia de valor permite desagregar as actividades estrategicamente relevantes para uma empresa, no sentido de se procurar perceber como ela pode construir vantagens competitivas. As actividades presentes na cadeia de valor situam-se a dois níveis:

- Actividades primárias – actividades envolvidas na criação física do produto ou serviço, na sua venda e na assistência pós-venda;
- Actividades de suporte – actividades que permitem que as actividades primárias se realizem, através do fornecimento de *inputs*, tecnologia, recursos humanos e informação.

Na figura 9 apresenta-se um esquema genérico da "cadeia de valor":

Figura 9: Esquema geral da cadeia de valor



Fonte: Sítio *web* da "marketingteacher.com" (2005)

Como se pode ver da figura 9 o estudo da cadeia de valor de uma empresa compreende os seguintes itens:

- Infra-estrutura – actividades de carácter geral que suportam o funcionamento da empresa. Usualmente estas actividades centram-se ao nível da gestão geral, sistemas de informação, gestão financeira e assessoria jurídica;
- Gestão de recursos humanos – actividades relacionadas com o recrutamento, contratação, formação, desenvolvimento e remuneração dos colaboradores da empresa;
- Desenvolvimento tecnológico – actividades desenvolvidas no sentido de melhorar o produto e processo através do qual é produzido;
- Compras – função de compras de todos os *inputs* necessários à cadeia de valor da empresa;
- Logística de entrada – actividades associadas com o recebimento, armazenamento e distribuição dos *inputs* necessários à produção;
- Operações – actividades associadas com a transformação dos *inputs* no produto (bem ou serviço) final;
- Logística de saída – actividades associadas com a recolha, armazenamento e distribuição física dos produtos até chegarem aos clientes;
- Marketing e vendas – actividades associadas com a criação de meios para que os clientes possam comprar os produtos;
- Serviço – Actividades associadas à prestação de um serviço que mantenha e aumente o valor do produto. Dentro destas actividades encontram-se a instalação de um produto, sua reparação e formação ministrada.

Um outro passo extremamente importante e consequente da cadeia de valor é o que diz respeito à identificação das competências centrais da empresa. Este conjunto restrito de recursos estratégicos e de capacidades serve de base ao desenvolvimento de vantagens competitivas. São estes recursos e capacidades que:

- Contribuem decisivamente para o valor percebido pelo cliente em relação ao produto ou serviço oferecido pela empresa;
- São únicas e diferenciadoras face à concorrência;
- São difíceis de imitar pelos concorrentes.

A vantagem competitiva de uma empresa é a capacidade dessa empresa no sentido de obter um desempenho superior ao dos seus rivais de uma forma sustentada. Estas vantagens competitivas permitem oferecer uma solução de valor única para os seus clientes, dificilmente imitável por parte dos concorrentes, que permita à empresa obter lucros acima da média de uma forma sustentada.

Segundo (Porter, 2000), uma vantagem competitiva resulta sempre de uma situação de desequilíbrio entre as empresas concorrentes, já que uma delas tem que ter um melhor desempenho. Essa vantagem competitiva pode resultar:

- Do aproveitamento das condições do meio ambiente externo (posicionamento nas indústrias mais atractivas);
- Das condições do meio ambiente interno (desenvolvimento de competências centrais).

Desta forma, a vantagem competitiva resultará de um ajustamento dinâmico entre competências centrais e factores críticos de sucesso do mercado.

Nos casos de insucesso analisados neste capítulo da dissertação optou-se por apresentar a análise SWOT de cada uma destas empresas e extrair as suas respectivas competências centrais e vantagens competitivas.

A análise SWOT é uma forma muito difundida de se fazer o diagnóstico estratégico de uma empresa. Neste tipo de análise pretende-se definir as relações existentes entre os pontos fortes e fracos de uma empresa com as tendências mais importantes que se verificam na envolvente global da empresa, seja ao nível do mercado global, do mercado específico, da conjuntura económica, das imposições legais, etc.

A sigla SWOT é a junção em inglês dos quatros elementos-chave desta análise estratégica, a saber:

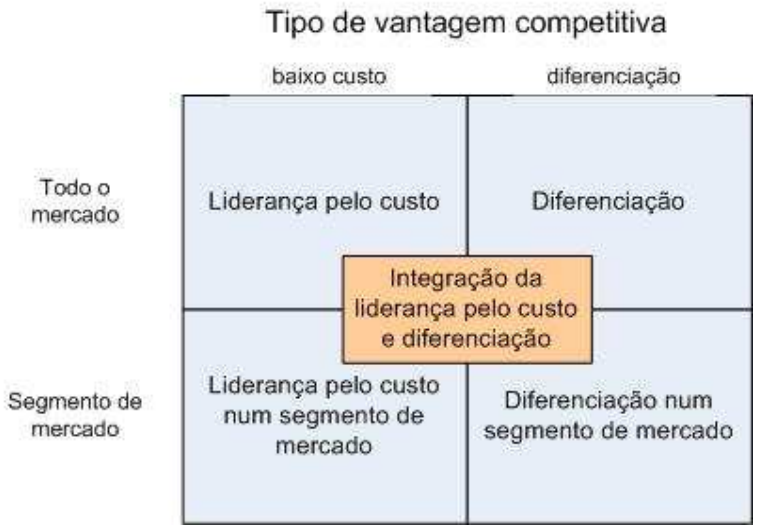
- *Strenghts* – pontos fortes: vantagens internas da empresa em relação às empresas concorrentes;
- *Weaknesses* – pontos fracos: desvantagens internas da empresa em relação às empresas concorrentes;
- *Opportunities* – oportunidades: aspectos positivos da envolvente que potenciam o crescimento da vantagem competitiva da empresa;

- *Threats* – ameaças: aspectos negativos da envolvente com o potencial de comprometer a vantagem competitiva da empresa.

A escolha da metodologia acima citada, para os casos de insucesso é, no meu entender, a mais adequada, pois a análise SWOT permite identificar os principais factores internos e externos que têm impacto numa dada organização. Nos casos de insucesso é muito importante identificar e compreender as ameaças e oportunidades oriundas de factores externos, que muitas vezes são elementos decisivos no sucesso ou não das empresas.

Para todas as empresas estudadas neste capítulo apresenta-se a sua estratégia competitiva. Segundo Porter (1996), uma empresa pode optar pelas seguintes estratégias de negócio:

Figura 10: Tipo de estratégias de negócio



Fonte: Patrício (2005)

Segundo o esquema acima, uma empresa pode seguir uma das três estratégias de negócio:

- Liderança pelo custo – a empresa opta por ter um produto de baixo custo na sua indústria, actuando num escopo alargado;
- Diferenciação – a empresa procura ser única no seu ramo, ao longo de algumas dimensões amplamente valorizadas pelos clientes. A empresa selecciona um ou mais atributos que muitos clientes na indústria consideram importantes, posicionando-se singularmente para satisfazer essas necessidades. Nesta situação, o cliente pode estar disposto a pagar um preço prémio por estas características diferenciadoras;
- Foco num "nicho" de mercado – a empresa selecciona um grupo alvo dentro do grande segmento da indústria. Este grupo alvo é atendido de uma forma privilegiada pela empresa, através de uma estratégia de liderança pelo custo ou liderança pela diferenciação.

Ainda segundo Porter (1996) uma empresa que se encaixa em cada estratégia genérica mas não alcança a liderança em nenhuma delas, está no "meio-termo". Uma empresa que

está no meio-termo será absorvida pelas líderes dessa indústria e apenas sobreviverá se a empresa tiver a sorte de ter os concorrentes também no "meio-termo", sendo que esta empresa será, em geral, menos lucrativa do que rivais que alcançam uma das estratégias genéricas.

No capítulo 6 desta tese de dissertação "Práticas e Regras para o Sucesso Empresarial" analisa-se mais em detalhe algumas das características do modelo de Porter e identificam-se algumas das suas principais lacunas, que permitem compreender o facto de muitas empresas actualmente conseguirem ter sucesso com estratégias simultâneas de liderança pelo custo e diferenciação e analisa-se o impacto das tecnologias de informação na criação de vantagens para uma empresa.

5.3 Casos de sucesso - Cyclades

5.3.1 Enquadramento da empresa

A Cyclades é uma empresa que fornece produtos para centros de gestão de *data center* através da disponibilização de acesso tipo consola, controle de alimentação de energia e gestão de sistemas que aumentem a tolerância a avarias.

A área de negócio da Cyclades passa pelo desenvolvimento de soluções de gestão da rede *out-of-band*. Este tipo de rede usa uma técnica de comunicação, que tipicamente é usada para sinais de controlo, que se posicionam em paralelo com comunicações normais (*in-band*) sem que com elas interfiram.

Uma solução completa de gestão da *data center* requer uma combinação de tecnologias *out-of-band* diferenciadas para complementar a capacidade *in-band*, como se apresenta na figura 11:

Figura 11: Soluções de out-of-band da Cyclades

Fonte: Site da "Cyclades" (2006)

Uma solução *out-of-band* para uma empresa costuma incluir quatro principais tecnologias:

- Consolo série – para prover um acesso de emergência, de baixo nível, ao equipamento controlado, incluindo servidores Unix e Linux;
- Teclado, Vídeo, Rato (KVM) – para disponibilizar uma interface gráfica aos utilizadores de sistemas operativos Unix e Linux, assim como para algumas aplicações em servidores Unix;
- Controlo de energia – para permitir ligar e desligar equipamentos remotamente;
- Administração – para assegurar a aplicação de regras de segurança e como infra-estrutura de suporte para gestão do equipamento de *data center*.

A Cyclades foi fundada em 1991, possuindo uma longa história de contribuição, envolvimento e compromisso com o Linux e o movimento Open Source, tendo sido a primeira empresa de *hardware* no mundo a desenvolver um *device driver* para Linux. No ano 2000 a Cyclades inovou no mercado de gestão de servidores e redes, sendo a primeira empresa no mundo a lançar um servidor baseado em Linux e a oferecer um pacote de desenvolvimento para os produtos. Desta forma, os clientes podiam adaptar os seus produtos às suas necessidades através da disponibilização do código fonte.

A Cyclades possui uma forte componente internacional, encontrando-se presente em dezasseis países distintos na América do Norte, Europa, América Latina e Ásia. A sua empresa sede encontra-se localizada em Fremont na Califórnia.

5.3.2 Produtos / Serviços

Os produtos disponibilizados pela Cyclades centram-se nos seguintes três níveis:

- Gestão de *data centers* – fornece soluções de servidores de consola, servidores de KVM corporativo, gestão de dispositivos série, gestão de processadores de serviços e unidades inteligentes de gestão de energia;
- Aplicações de WAN – disponibiliza um conjunto de encaminhadores ("routers") que possuem em comum as seguintes características: multiprotocolo, alta disponibilidade e desempenho, qualidade garantida de serviços, instalação e configuração remota e mecanismos de segurança;
- Cabos e adaptadores – disponibiliza também um conjunto de cabos e adaptadores para produtos de *data center* e também para dispositivos gerais de conectividade, nomeadamente para servidores de acesso remoto, placas multi-seriais, placas síncronas e réguas inteligentes de distribuição de energia.

5.3.3 Cadeia de Valor

Apresenta-se de seguida a cadeia de valor da empresa Cyclades:

Actividades de Suporte

Infra-Estruturas:

- Líder mundial em conectividade Linux;
- Presença de filiais da empresa em diversos países na Europa, América do Norte,

América do Sul e Ásia;

- Redes de comunicação fiáveis e de alto débito;
- Gestão global da qualidade dos serviços prestados.

Recursos Humanos:

- Elevadas competências técnicas no domínio da tecnologia de redes;
- Elevada especialização dos seus colaboradores.

I&D:

- Linha inovadora de servidores de consola (Cyclades-TS);
- Prémios obtidos (1ª empresa de hardware no mundo a desenvolver plataformas de comunicação para Linux (1998) e primeira empresa a nível mundial a lançar um servidor de consola com base no Linux (2000)).

Compras:

- Definição e estabelecimento de contratos com fornecedores de placas e circuitos digitais;
- Parcerias estratégicas com os fornecedores.

Actividades Primárias

Logística de Entrada:

- Parcerias com empresas desenvolvedoras de software específico;

Operações:

- Controlo de qualidade rigoroso;
- Deliberação de software submetidas a uma auditoria de mecanismos de segurança por uma empresa externa;
- Construção de um acesso de baixo nível para providenciar acessos de emergência;
- Integração de circuitos electrónicos;
- Interface gráfica para o utilizador;
- Controle de energia para ligar e desligar equipamentos remotamente;
- Infra-estrutura necessária para gestão do equipamento de *data center*.

Logística de Saída:

- Venda para distribuidores autorizados;
- Parcerias com empresas de telecomunicações para o fornecimento de produtos e serviços.

Marketing e Vendas:

- Imagem de marca, difundida pelos seus próprios clientes:

"Having information at hand to analyze the problem and the ability to remotely access the equipment replaces the need to travel to data centres, saving us time and money" (Montanaro R., Director of IT Operations at Fastweb);

"With Cyclades technology, we have significantly reduced our recovery time from unexpected hardware or software crashes, which helps us avoid late night emergency trips to the data centres" (Kulzer T. CEO of Aweber Sytems).

- Os seus clientes representam 85% do total das 100 maiores empresas a nível mundial;
- Personalização dos produtos desenvolvidos de acordo com as especificações do cliente;
- Parcerias com revendedores e estabelecimento de ligações privilegiadas.

Serviço pós-venda:

- Actualização via web do *firmware* para os servidores console;
- Contratos PSA (Priority Support Agreement);
- Diferentes níveis de serviço para atender as necessidades dos clientes (PSA Platinum, PSA Gold, PSA Silver e PSA Bronze).

5.3.4 Competências Centrais e Vantagens Competitivas

No caso da Cyclades consideraram-se as seguintes competências centrais:

1. Personalização de soluções;
2. Monitorização das actividades da rede;
3. Disponibilização de um acesso remoto alternativo;
4. Parcerias com fornecedores e distribuidores
5. Parcerias com empresas desenvolvedoras de software;
6. Parcerias com *Telcos*;
7. Conhecimento profundo da tecnologia de redes;
8. Reputação da marca;
9. Recursos de segurança;
10. Equipa interna de I&D;
11. Construção de soluções independentes do sistema operativo do cliente.

Da concretização destas competências centrais podem-se construir vantagens competitivas que se centram, essencialmente, em 8 níveis:

- Focalização na área de negócio da tecnologia de redes [7] – motivado pelos profundos conhecimentos da área das tecnologias de redes por parte dos seus colaboradores;

- Desempenho e funcionalidades superiores [3, 7, 9, 10] – no caso de a rede de uma empresa apresentar algum problema, as soluções disponibilizadas pela Cyclades possibilitam aos colaboradores de TI da empresa cliente uma resolução mais rápida e fácil, deixando a rede funcionando com o mínimo de interrupções;
- Simplificação da administração da rede [1, 2, 9, 11] – através do fornecimento de uma plataforma única, segura, integrada e abrangente que integra a gestão de infra-estrutura independentemente do sistema operacional, fornecedor da plataforma ou arquitectura de hardware;
- Optimização dos recursos de TI [2] – aumenta a eficiência dos colaboradores de TI de uma organização com ferramentas de gestão remoto e automação. Desta forma, é possível que técnicos dedicados a equipamentos específicos possam ser divididos pela empresa, reduzir os custos com tempos de viagem e produzindo respostas rápidas;
- Redução da necessidade de intervenção e erros humanos [2, 3] – as soluções disponibilizadas pela Cyclades automatizam a detecção de falhas de TI e recuperam as actividades, ajudando a acelerar a disponibilidade de serviço e reduzindo o tempo de inactividade de uma rede causado pelos erros do operador. Segundo a Gartner "only Out-of-Band management technology can address the 20 percent of infrastructure incidents and up to 40 percent of operator errors that cause unplanned network downtime" (Gartner, 2005).
- Diminuição do tempo de inactividade [3] – o cliente obtém uma maior visibilidade do seu equipamento crítico de TI, podendo assim monitorar mais de perto a infra-estrutura de rede. Frequentemente, as soluções Cyclades alertarão os seus funcionários de TI quanto aos problemas e falhas de relatório antes de se tornarem incidentes que resultem em tempo inactivo, antes mesmo dos próprios funcionários da empresa cliente perceberem;
- Facilidade de personalização [1, 5] – torna-se possível de uma forma bastante simples e intuitiva adaptar cada uma das soluções standard da Cyclades às necessidades específicas de cada cliente. A existência de parceiros ao nível de empresas que desenvolvem soluções de software, permitem introduzir novas funcionalidades nas soluções Cyclades que não se enquadram no âmbito das competências centrais da Cyclades;
- Escalabilidade dos seus produtos [1, 5, 10] – as soluções de monitorização da rede podem ser redimensionadas de acordo com a dimensão e características da rede da empresa. Simultaneamente, estas soluções são perfeitamente escaláveis à medida que a rede da empresa cresce e se torna mais complexa. A construção de uma nova solução não origina o desenho completo de um novo *layout*, mas é integrado em forma de *upgrade* ao anteriormente existente.

5.3.5 Estratégia de Criação de Valor

A Cyclades segue uma estratégia de liderança pela diferenciação. Os seus produtos

distinguem-se da demais concorrência fundamentalmente através do seu desempenho e funcionalidades superiores, facilidade de personalização, escalabilidade das suas soluções e através da gestão simplificada da rede.

5.3.6 Clientes

A Cyclades possui um gama muito diversificada e invejável de clientes. Tal como foi previamente mencionado, cerca de 85% das 100 maiores empresas a nível mundial são clientes da Cyclades.

Na tabela 3 podemos ver alguns dos principais clientes da Cyclades por cada uma das áreas de negócio das empresas clientes:

Tabela 3: Tipo de estratégias de negócio

Automóveis	**Indústria**	**Educação**	**Entretenimento**
Audi AG	Clorox	Columbia University	Disney
BMW	Eastman kodak	MIT	DreamWorks SKG
Ford Motor Corporation	FedEx	Stanford University	Reuters
General Motors	Xerox	Yale University	Time Warner
Bancos	**Retalho**	**Governo**	**Tecnologia**
Bank of America	Dillard's	General Dynamics	Apple
Deloitte & Touche	Gap	NASA	Dell
Deutsche Bank	Home Depot	Sandia National Laboratory	IBM
HSBC Bank	Wal-Mart	US Army Corps	Nokia
Saúde	**Telecomunicações**	**Internet**	
Eli Lilly	British Telecom	FastWeb	
Genentech	Deutsche Telecom	Google	
WebMD	Vodafone	Lycos	
Wyeth	Telefonica	Yahoo	

Fonte: Site da "Cyclades" (2006)

De seguida passamos a analisar muito resumidamente o impacto das soluções Cyclades nas organizações que a adoptaram. Para tal, seleccionaram-se dois casos de estudo: a *Vodafone* e o *Yahoo*.

A *Vodafone* é a maior empresa de telecomunicações móveis do mundo, que oferece uma extensa faixa de serviços, que inclui comunicações de dados e voz, com presença significativa na Europa Continental, Reino Unido, Estados Unidos e Extremo Oriente.

A *Vodafone* escolheu a empresa Cyclades quando necessitou de substituir os servidores de consola no núcleo da infra-estrutura *out-of-band*. A nova solução desenvolvida teria que dar condições para que os administradores de rede pudessem aceder e gerir os sistemas operativos e o hardware de TI de forma independente da rede. Estes, teriam também que ser compatíveis com a arquitectura de rede e os sistemas actuais. Para além disso, teriam que ser garantidos recursos de segurança com mecanismos de autenticação e controlo por auditoria.

Esta solução produzida possibilitou à *Vodafone* a obtenção das seguintes mais valias:

- Redução para poucas horas, da perturbação causada por determinadas falhas e aumento da eficiência de toda a organização de TI;
- Efectuação do *login* em um servidor a partir de um único local, sem necessidades de ir a locais remotos para resolver problemas;
- Visualização, acesso e controlo de vários dispositivos a partir de um único sistema, ajudando a diminuir a necessidade de instalação de equipamentos redundantes.

O *Yahoo* é um provedor líder em produtos e serviços on-line a clientes e empresas em todo o mundo, atingindo por mês, 238 milhões de pessoas no mundo.

O *Yahoo* criou uma infra-estrutura de rede global impressionante para atender a um dos maiores públicos no mundo, precisando de manter uma rede que contém no seu total mais de 15000 elementos de rede, incluindo servidores web e outros dispositivos de rede, como por exemplo, *routers* e *firewalls*.

O *Yahoo* necessitava de mais de 40 portas seriais em uma unidade de espaço rack. Até à altura, não existia um produto assim no mercado, daí que tivesse sido necessário criar o primeiro servidor de console de 48 portas, que foi o resultado de um grande processo de interacção entre as duas empresas.

A solução desenvolvida pela Cyclades trouxe os seguintes benefícios para o *Yahoo*:

- Gestão dos centros de processamento de dados pelo mundo a partir de um *Network Operation Center* (NCO);
- Diminuição do tempo de resposta a problemas detectados na rede e aumento da disponibilidade da rede;
- Diminuição dos custos operacionais da empresa.

5.3.7 Conclusão

A Cyclades desenvolve os seus produtos na área da tecnologia de redes a partir de tecnologias *open source*. Todos os seus produtos são desenvolvidos a partir de plataformas *open source*, que posteriormente são integradas em soluções de conjunto muito mais complexas, que em alguns casos são adaptadas à medida de cada cliente.

A grande vantagem competitiva da Cyclades provém das suas elevadas competências técnicas que lhe permite fornecer produtos com um grande conjunto de mais valias para os seus clientes ao nível das facilidades de gestão da rede e redução dos custos operacionais que lhe estão inerentes. Paralelamente, a sua forte presença mundial permite-lhe atender cerca de 85 das 100 maiores empresas mundiais, o que lhe traz inequívocas mais valias em termos de conhecimentos adquiridos e casos de sucesso que podem ser usados como ponto de partida na conquista de novos clientes.

Segundo (Blackshaw, 2005) esta nova abordagem de marketing é poderosíssima, especialmente quando estamos a lidar com marketing industrial, na qual o sucesso de uma boa política de aquisições tem influência na actividade da própria empresa. Este autor afirma "Make no mistake, Word-of-mouth marketing's potential is enormous, and it's one of the obvious and logical next steps for advertising….Word-of-mouth marketing works because it comes from a trusted source" (Blackshaw, 2005).

Existem também importantes barreiras à entrada de novos concorrentes, pois o mercado de gestão de redes empresarial é extremamente competitivo e encontra-se dominado pela Cyclades. Para além disso, os profundos conhecimentos na área da tecnologia de redes da Cyclades e a sua experiência passada na concepção desse tipo de soluções para grandes empresas multinacionais permite-lhe manter-se à frente de novos potenciais concorrentes.

5.4 Caso da Wild Open Source

5.4.1 Enquadramento da empresa

A Wild Open Source é uma empresa que providencia um conjunto de serviços na área das soluções *open source*, particularmente ao nível do sistema operativo Linux. O seu objectivo passa por fornecer serviços com o intuito de ajudar os seus clientes na área de planeamento, adopção, desenvolvimento e gestão de soluções *open source* em ambientes empresariais.

No início de 2005 a companhia alemã *Soeren Research* efectuou um estudo em 50 companhias que adoptaram software livre, e conclui que elas pouparam cerca de 30% nos seus custos de software. É neste sentido que aparece a Wild Open Source que, segundo o seu CEO, Henry Hall, encoraja as empresas a migrarem do software proprietário para o software livre, através do apoio de consultores com larga experiência neste campo.

A empresa foi fundada em 2001 e possui um largo conjunto de especialistas na área de soluções *open source*. Mais de metade dos seus colaboradores são oriundos da empresa LinuxCare, que foi a primeira empresa "pure services" para o sistema operativo Linux.

A sua sede encontra-se localizada nos Estados Unidos da América na área de Massachussets, possuindo também escritórios na costa oeste do EUA (San Francisco), no Canadá, na Suiça (Genebra) e no Japão (Tóquio).

5.4.2 Produtos / Serviços

A Wild Open Source disponibiliza serviços no domínio de soluções *open source* que se centram nas seguintes áreas:

- Desenvolvimento de produtos – assistência ao nível do desenvolvimento de soluções *open source*;
- Integração de soluções – providencia *know-how* de forma a permitir a integração de diferentes programas e módulos *open source* num todo coerente e numa solução de conjunto para uma empresa cliente;
- Estratégia e planeamento – consultadoria ao nível da definição dos melhores projectos *open source* para uma empresa e estabelecimento de processos de desenvolvimento de software que permitam minimizar o seu ciclo de desenvolvimento e o tempo de entrada no mercado;
- Desenvolvimento personalizado – criação de soluções à medida de cada cliente. Neste caso, a empresa parte de soluções existentes no mercado no domínio *open source* e adapta-as à medida das necessidades e especificidades de cada cliente;
- Teste de soluções – avaliação de cada solução desenvolvida no domínio do software *open source* e detecção e correcção de eventuais anomalias no código;
- Formação de recursos humanos – disponibilização de formação para a correcta utilização de cada uma das soluções *open source* existentes na empresa cliente.

5.4.3 Cadeia de Valor

Apresenta-se de seguida a cadeia de valor da empresa Wild Open Source.

Actividades de Suporte

Infra-Estruturas:

- Presença de sucursais da empresa em pontos estratégicos a nível mundial (Europa do Norte, Europa Central e Ásia);

Recursos Humanos:

- Elevada experiência e conhecimento acumulado no desenvolvimento e integração de soluções *open source*;
- Certificação das competências dos seus colaboradores;
- Formação contínua dos seus colaboradores;
- Flexibilidade e capacidade de adaptação dos seus recursos humanos.

I&D:

- Empresa pioneira no lançamento de iniciativas “enterprise Linux”;
- Lançamento da iniciativa “ESC” apresentada na “LinuxWorld Expo 2003” em Nova Iorque.

Compras:

- Definição das plataformas de base para concepção e desenvolvimento das soluções;
- Parcerias estratégicas com a IBM, SGI e Agilsys.

Actividades Primárias

Logística de Entrada:

- Integração de eventuais módulos disponibilizados pelos parceiros.

Operações:

- Desenvolvimento e integração de soluções "best-of-breed";
- Capacidade de personalização das soluções à medida de cada cliente.

Logística de Saída:

- Instalação da solução no cliente;

Marketing e Vendas:

- Preço mais baixo que os seus concorrentes;
- Adaptação de cada solução à medida de cada cliente.

Serviço pós-venda:

- Contrato *Premium Service* que permite definir o tempo de resposta máxima, gestão da conta do cliente e estabelecimento de recursos dedicados;
- Subscrições técnicas;
- Formação ministrada na empresa cliente.

5.4.4 Competências Centrais e Vantagens Competitivas

No caso da Wild Open Source consideraram-se as seguintes competências centrais:

1. Elevada qualificação da sua equipa técnica;
2. Integração de diversas soluções oriundas de plataformas *open source* distintas;
3. Posicionamento dos seus produtos;
4. Personalização das soluções;
5. Testes e avaliação do código;
6. Parcerias mantidas com líderes do mercado;
7. Optimização de aplicações *open source*;
8. Gestão da gama de suporte técnico

Da concretização destas competências centrais pode-se construir vantagens competitivas que se centram, essencialmente, em 6 níveis:

- Integração de soluções *open source* e comerciais [1, 2, 6] – as soluções desenvolvidas pela WOS não utilizam unicamente plataformas e ferramentas *open source*, mas são integradas, em muitas situações, com outras ferramentas do domínio

comercial. Tenta-se tirar partido das melhores soluções específicas para cada cliente e integrá-las num todo coerente;

- Personalização do software às necessidades da empresa [1, 2, 3, 4] – cada módulo desenvolvido é personalizado segundo um processo anterior de análise de requisitos estipulado para cada uma das empresas cliente;
- Redução do custo de aquisição e posse do software [2, 3, 4] – através da utilização de soluções *open source* consegue-se disponibilizar soluções com um custo inicial de aquisição muito mais baixo e consegue-se diminuir também os custos relativos a novas soluções de *upgrade* e lançamento de novas versões;
- Redução do ciclo de desenvolvimento do produto e "time-to-market" [1, 2] – a vasta experiência dos colaboradores da WOS e as metodologias em termos de engenharia de software utilizadas permite diminuir o tempo de desenvolvimento de uma nova solução e a consequência entrada no mercado de uma nova ferramenta;
- Experiência da equipa de desenvolvimento [1] – a elevada qualificação dos seus colaboradores e o saber acumulado ao longo dos anos permite construir soluções cada vez mais adequadas e robustas para cada cliente;
- Apoio técnico personalizado [8] – através da disponibilização de três tipos de suporte adaptado às necessidades de cada utilizador, respectivamente "platinum services", "gold services" e "silver services".

5.4.5 Estratégia de Criação de Valor

A Wild Open Source segue uma estratégia de gestão simultânea baseada na liderança pelo custo e na diferenciação.

Existe uma liderança pelo custo, pois as soluções por si desenvolvidas têm um custo mais baixo que a dos seus concorrentes, que usam soluções comerciais.

Existe uma liderança pela diferenciação, pois as soluções desenvolvidas pela WOS são personalizadas de acordo com as necessidades e preferências de cada cliente e através dos serviços de apoio técnico disponibilizado à medida de cada cliente.

5.4.6 Clientes

A Wild Open Source possui um conjunto de grandes empresas em várias áreas de indústria como seus clientes. Destacam-se os seguintes clientes:

- Sybase – uma das dez maiores empresas de software independentes a nível mundial e líder em tecnologias de bases de dados e infra-estruturas empresariais móveis e *wireless*;
- HP – um dos principais fabricantes de *hardware* e tecnologias móveis;
- Avaya – uma das principais empresas na área das telecomunicações;
- Intel – maior fabricante mundial de chips e igualmente líder na produção de computadores, produtos de rede e comunicações;

- Dell – fabricante de computadores que oferece soluções inovadoras ao nível da customização de cada computador;
- Oracle – uma das principais empresas de maior renome na área de gestão de bases de dados;
- Nokia – líder mundial em comunicações móveis;
- SGI – empresa desenvolvedora de soluções de gestão de informação;
- Scala – empresa líder mundial no desenvolvimento de soluções multimédia para a televisão.

Como exemplo, a Scala precisou de efectuar o *upgrade* dos seus servidores internos de correio e serviços de notícias, devido ao facto de ao longo dos últimos anos a manutenção da administração dos seus servidores se ter tornado demasiado complexa. O seu objectivo consistia em aumentar o rendimento do sistema e tirar vantagem de novo hardware lançado no mercado.

Neste sentido, foi necessário desenvolver uma solução personalizada à medida da Scala. Esta solução teve em conta a necessidade de replicar a informação já existente no antigo sistema, configurá-la e adicionar novas funcionalidade à solução desenvolvida. Esta nova solução foi posteriormente instalada num nova plataforma de hardware e procedeu-se à configuração do software de forma a facilitar a integração de cada um dos novos módulos.

5.4.7 Conclusão

A Wild Open Source optou pelo desenvolvimento e integração de soluções *open source* e comerciais numa única solução de conjunto disponibilizada pela empresa aos seus clientes. Estas soluções são desenvolvidas à medida de cada cliente e de uma forma geral têm um custo mais baixo que a dos seus principais concorrentes que utilizam unicamente soluções comerciais.

As suas principais vantagens competitivas residem ao nível da qualificação e experiência dos seus consultores e da sua forte relação com o cliente. Para a Wild Open Source a manutenção dos seus quadros técnicos é de vital importância. Existe portanto uma forte ameaça de sucesso nesta empresa que passa pela saída de quadros qualificados.

A Wild Open Source encontra-se numa área de negócio extremamente competitiva que sofre uma enorme pressão por parte das empresas instaladas, não só em termos de preço como também ao nível do fornecimento de soluções diferenciadas. Existe um claro risco ao nível de perda de clientes para empresas que seguem uma estratégia diferenciada e focalização num sector de negócio, pois estas podem servir uma dada área de negócio simultaneamente com critérios de qualidade superiores e com custos muito competitivos.

Para além disso, existem muito poucas barreiras à entrada de novos concorrentes, pois as soluções *open source* encontravam-se disponíveis para toda a comunidade, que obviamente podem também ser usadas por outras empresas que entrem nesse mercado. As únicas barreiras à entrada de novos concorrentes situam-se ao nível da maior ou menor

dificuldade de personalização e integração de cada uma das soluções e ao nível dos serviços de suporte disponibilizados.

5.5 Caso da Critical Software

5.5.1 Enquadramento da empresa

A Critical Software é uma empresa portuguesa que fornece soluções, serviços e tecnologias para os sistemas de informação críticos das empresas. Actualmente a Critical Software é uma das empresas tecnológicas portuguesas de maior sucesso a nível internacional, tendo sido reconhecida como uma das dez empresas que mais cresceu no último ano de 2005.

A Critical Software foi fundada em 1998, numa incubadora de empresas do Instituto Pedro Nuno, ligado à Universidade de Coimbra. A empresa iniciou desde cedo o seu processo de internacionalização, conseguindo atingir em dois anos os objectivos traçados para cinco. Actualmente, os negócios realizados fora de Portugal representam cerca de 75% da sua facturação. Em 2002, as vendas de exportação ultrapassaram, pela primeira vez, as vendas domésticas.

Em 1999, a empresa abriu uma subsidiária em São José, a capital do *Silicon Valley*, na Califórnia (EUA), possuindo também um representante de vendas em Inglaterra. Perante o grande volume de negócios gerados na zona de Lisboa, em 2003 a Critical Software abriu aí um outro escritório situado no Parque Tecnológico de Lisboa. Em cerca de seis anos, uma empresa de três pessoas transformou-se numa organização que conta hoje com mais de 90 funcionários.

A empresa está estrategicamente organizada em três unidades de negócio: 1) Rede e Telecomunicações, 2) Aeroespacial & Defesa e 3) Soluções Empresariais, possuindo um vasto conjunto de produtos e serviços, como soluções de gestão para o sector industrial e um vasto *know-how* em qualidade de serviço para redes IP, segurança e computação móvel.

Esta empresa desenvolve produtos na área de software *open source* e comercial. No âmbito desta tese de dissertação analisa-se com maior detalhe a solução *edgeBOX* que utiliza diversas soluções na área do *open source* para a sua concepção.

5.5.2 Produtos / Serviços

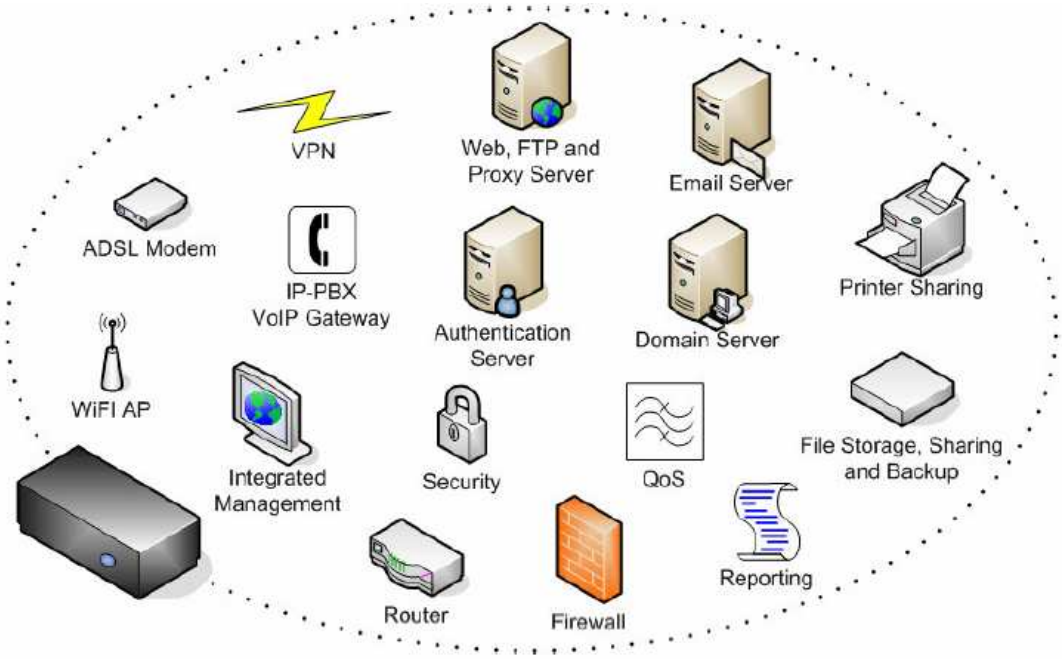
A Critical Software já teve oportunidade de desenvolver inúmeras soluções à medida de cada cliente sobre cada uma das áreas de negócio indicadas no ponto anterior. Para além disso, a Critical Software tem desenvolvido um conjunto de produtos inovadores, nomeadamente:

- CoreCompetence – uma solução inovadora na área de gestão de competências de recursos humanos;
- EdgeBox – solução completa de gestão de infra-estruturas de acesso a rede de dados;

- WMPI – solução de *middleware* de gestão de mensagens para *High Perfomance Computing* (HPC);
- Xception – *framework* para verificação e validação dinâmica de software.

No caso desta dissertação, vai-se dar particular atenção à solução edgeBOX. A solução edgeBOX é uma completa plataforma multi-funcional de rede, que se destina às pequenas e média organizações. A sua arquitectura contempla os seguintes módulos:

Figura 12: Arquitectura da *edgeBOX*



Fonte: Site do projecto "Edgebox – Critical Software" (2006)

A EdgeBOX oferece uma solução única em termos de proposta de valor. A combinação deste tipo de serviços num único dispositivo físico permite inúmeras vantagens destacando-se a simplicidade de configuração da rede, implementação de mecanismos robustos de segurança, integração de dados e voz e *backup* de informação.

5.5.3 Cadeia de Valor

Apresenta-se de seguida a cadeia de valor da empresa Critical Software:

<u>Actividades de Suporte</u>

Infra-Estruturas:

- Delegações da empresa em território nacional e na Califórnia (Silicom Valley – San Jose);
- Líder mundial no desenvolvimento de aplicações críticas de software;
- Taxa de crescimento da empresa, indicada pela seguinte afirmação retirada do seu *site web*.

"…the company has grown 80% per year from 1999 to 2001, and expects to achieve a sustained

growth of 50% per year in the period 2002-2004" (in site www.critical software.com, 2006)

Recursos Humanos:

- Grau de escolaridade elevado dos seus colaboradores;
- Aposta na formação contínua dos seus funcionários.

I&D:

- O investimento em I&D representa cerca de 10% do seu volume total de vendas;
- Parceiros de I&D através de programas comunitários;
- Publicações científicas;
- Certificação de qualidade ISO 9001:2000;
- Prémio para a melhor aplicação de banda larga em Madrid – IEC Infovision Award.

Compras:

- Definição das tecnologias a serem usadas em cada projecto;

Actividades Primárias

Logística de Entrada:

- Integração de módulos com empresas desenvolvedoras de soluções complementares.

Operações:

- "Enterprise Application Integration (EAI) – compreende a integração de diversas aplicações e transferência de dados entre diversos sistemas de gestão de base de dados;
- "Dependability" – contempla a integração de serviços ao nível da sua disponibilidade, confiabilidade e segurança.
- "Ground Segment Software" – software ao nível de interacção com a máquina nas suas camadas mais baixas ("low layers");
- "Networking" – disponibilização de serviços na área de redes e sua interligação com as aplicações que a empresa disponibiliza;
- "High Perfomance Computing" – elevado desempenho das suas aplicações, tornando-se um requisito fundamental para algumas aplicações dos seus principais clientes;
- "Manufacturing Execution Systems" – actividades de planeamento, execução e controlo de tarefas.

Logística de Saída:

- Encaminhamento das soluções para os parceiros.

Marketing e Vendas:

- Endereçamento de diversos mercados (telecomunicações, sector público e defesa);

- "Word of mouth";
- Relação mantida com o cliente;
- Notícias em jornais;
- Revistas científicas da especialidade.

Serviço:

- Instalação da solução é realizada por parceiros da Critical Software;
- Equipa de suporte que dá formação às equipas técnicas dos parceiros.

5.5.4 Competências Centrais e Vantagens Competitivas

No caso da Critical Software consideraram-se as seguintes competências centrais:

1. Parcerias com grandes entidades na área de I&D;
2. Elevadas competências técnicas;
3. Integração de soluções "best-of-breed";
4. Personalização das soluções desenvolvidas;
5. Integração de soluções *open source* e comerciais;
6. Concepção de sistemas críticos;
7. Robustez das soluções desenvolvidas;
8. Formação dos seus recursos humanos;
9. Forte investimento em I&D.

Da concretização destas competências centrais podem-se construir vantagens competitivas que se centram, essencialmente, em 7 níveis:

- Focalização vertical num âmbito geográfico alargado [2, 6] – a Critical Software aposta no desenvolvimento de sistemas críticos para áreas de actividade distintas e mercados geográficos alargados. A sua competência central em termos operacionais centra-se no desenvolvimento de sistemas críticos que podem ser usados em vários tipos de indústrias e actividades;
- Vanguarda e pioneirismo em tecnologias inovadoras [1, 2, 3] – a Critical Software usa soluções inovadoras, que muitas vezes são o resultado directo de projectos de parcerias transnacionais ao nível de I&D;
- Independência de fornecedor [3, 4] – a Critical Software não define à partida uma plataforma base para a concepção das suas soluções. Em primeiro lugar definem-se as necessidades específicas de cada cliente e em segundo lugar tenta-se encontrar no domínio das soluções comercias e *open source* módulos que possam ser integrados, com maior ou menor necessidade de personalização, para cada um dos projectos;
- Qualidade das soluções desenvolvidas [2, 7, 8] – todas as soluções desenvolvidas têm um elevado enfoque ao nível da qualidade de processos utilizados e resultados

produzidos. Uma prova desta situação verifica-se pelos prémios obtidos, como foi o caso da solução *edgebox* que ganhou o prémio de melhor aplicação na área da banda larga no *Broadband World Forum Europe*, realizado em Outubro de 2005 em Madrid;

- Flexibilidade e agilidade [3, 8] – A dimensão ainda reduzida da empresa permite-lhe ter uma maior agilidade, de forma a ir de encontro às necessidades de cada cliente e reduzir os custos e tempo de produção de uma nova solução. Paralelamente, o facto de se ter criado uma organização bastante "flat" a nível hierárquico permite-lhe estar em maior contacto com as necessidades do mercado;
- Cumprimento de prazos [2, 8] – a Critical Software faz um grande esforço para o cumprimento dos prazos estabelecidos nas reuniões com os clientes e o acompanhamento da evolução das soluções por esses clientes.
- Relação mantida com o cliente [4, 7, 8] – a Critical Software dá muita importância ao "feedback" disponibilizado para cada cliente e ao acompanhamento da solução do cliente, após a sua instalação e configuração na empresa cliente.

5.5.5 Estratégia de Criação de Valor

A Critical Software segue uma estratégia de liderança focada na diferenciação num segmento de mercado através do desenvolvimento de soluções críticas para vários sectores de actividade apostando num ambiente geográfico alargado. Estas soluções distinguem-se pela sua qualidade, pela utilização de tecnologias de ponta e pela excelência de relacionamento mantido com o cliente.

5.5.6 Clientes

A Critical Software possui também um conjunto muito alargado de clientes a nível nacional e internacional como se pode ver na tabela 4:

Tabela 4: Principais clientes da Critical Software

Espaço Aéreo	**Defesa**	**Indústria**
Alcatel Space Industries	Ministério da Defesa Nacional	Soporcel
NASA	NATO	Revigrés
ESOC	Westland Helicopters	Biotecnol
ESA		Daikin Industries
Sector Público	**Telecomunicações**	
COTEC	Vodafone	
Cranfield University	Siemens	
Instituto de Informática	TDM	
Ministério da Economia	PT Comunicações	

Fonte: Site da "Critical Software" (2006)

A Critical Software ajudou a Siemens no desenvolvimento de uma solução de gestão da rede para mercadas globais. A Critical Software, como parceira tecnológica privilegiada no desenvolvimento de soluções de gestão de rede de telecomunicações com elevada complexidade de requisitos e altos conhecimentos tecnológicos, foi responsável pelo módulo de gestão de tráfego.

As principias funcionalidades deste módulos compreendem a gestão automática do tráfego numa rede a partir de uma interface gráfica amigável. Esta solução está actualmente em exploração em vários operadores de telecomunicações em Taiwan, Grécia, Portugal e Polónia.

A ESA é também um dos principais clientes da Critical Software. Ao longo dos últimos anos foram desenvolvidos vários projectos para a ESA, dos quais merece especial destaque o projecto GSS-web-3.0.

Este projecto consistiu no desenvolvimento de uma infra-estrutura de software que permitisse a publicação e gestão de conteúdos *via web*. Este projecto foi desenvolvido seguindo uma estrutura modular, baseada num conjunto robusto e completo de tecnologias *open source*.

Este projecto trouxe inequívocos benefícios para a ESA, destacando-se os seguintes:

- Melhoramentos ao nível da gestão de utilizadores e controlo de acessos;
- Uma nova secção de notícias;
- Criação de um fórum de discussão;
- Secção de gestão de documentação;

- Armazenamento e gestão do conhecimento.

No caso da "edgebox", temos que esta começou por ser desenvolvida com um cliente alvo que era a Soporcel, tendo a partir daí servido como ponto de partida para a divulgação desta solução por outras empresas e tendo a mesma sido difundida pelos diversos escritórios da *Soporcel*.

Actualmente esta solução está a ser disponibilizada por vários parceiros escolhidos pela Critical Software. Existem, por exemplo, parceiros no Reino Unido, no Brasil, no Chile e em Portugal. As próprias empresas parceiras têm equipas de suporte que são os responsáveis por dar formação às empresas clientes e por responder a dúvidas mais básicas que os clientes finais tenham. Contudo, quando surge uma dúvida mais técnica, pode ser encaminhada para a equipa de suporte da Critical Software e, se mesmo assim for necessário, esta pode ser encaminhada para a equipa de desenvolvimento.

5.5.7 Conclusão

A Critical Software optou por focalizar a sua área de negócio no segmento de mercado das soluções críticas apostando num âmbito geográfico alargado. A Critical Software utiliza no desenvolvimento das suas soluções ferramentas *open source* e comerciais, mediante as necessidades e especificidades de cada cliente.

A Critical Software possui uma forte componente de internacionalização possuindo já um leque invejável de clientes. Para esta empresa o seu processo de conquista de novos mercados é crítico, pois o mercado português não tem só por si uma dimensão que justifique uma elevada rentabilidade para uma empresa portuguesa que se focalize na concepção de sistemas críticos.

Desta forma, e em especial no caso da *edgebox*, a aposta é claramente na utilização de parceiros com elevado conhecimento dos mercados alvos, ou seja, os mercados a que se destinam as soluções *edgebox*. Para além disso, a Critical Software utiliza muito a subcontratação para a instalação e manutenção dos seus produtos.

No mercado endereçado pela Critical Software existem importantes barreiras à entrada de novos concorrentes, pois esta é uma área em que as competências técnicas são críticas, não sendo simples encontrar no mercado colaboradores com formação e elevada experiência na concepção deste tipo de soluções. Para além disso, as soluções desenvolvidas neste domínio necessitam de ter uma elevada qualidade e certificação ao nível dos processos usados na sua concepção e ao nível de mecanismos de segurança e confidencialidade da informação.

5.6 Caso da Eazel

5.6.1 Enquadramento da empresa

A Eazel foi uma empresa privada criada em Abril de 2000 e que cessou a sua actividade 16 meses depois de a ter iniciado. Trata-se de um dos casos de insucesso mais debatidos

pela comunidade *open source*, uma vez que se tratou de um projecto com um elevado nível de investimentos e que reuniu um grupo de colaboradores com uma elevada experiência na área do software e no mundo dos negócios. Dentro destes colaboradores destacam-se o seu COO (Mike Boich), que se juntou à Apple em 1982 tendo sido um dos responsáveis pelo projecto Macintosh; Andy Hertzfeld que também foi um dos responsáveis por este projecto na Apple; e Mike Homer que foi presidente da America Online.

A Eazel foi criada recorrendo a diversos capitais provenientes de diversas entidades, entre as quais:

- A empresa de capital semente *Accel Partners* que investiu 11 milhões de dólares;
- Duas organizações denominadas por "Business Angels" (*Ron Conway's* e *Sippl Macdonald Ventures*);
- Dinheiro proveniente dos seus fundadores.

A área de negócio da Eazel passava por desenvolver software e serviços que tornassem os computadores mais simples de serem usados pelos utilizadores que utilizassem o sistema operativo Linux. De certa forma, procurava-se encontrar uma nova abordagem que possibilitasse que o sistema operativo Linux se tornasse cada vez mais popular e que fosse de encontro aos utilizadores *desktop* dos computadores.

5.6.2 Produtos / Serviços

A actividade da Eazel basicamente consistia na divulgação de um produto gratuíto designado por *Nautilus* e agregar valor em produtos e serviços que pudessem ser acrescentados a esta solução base.

O *Nautilus* é uma aplicação de rede que integra acesso a ficheiros, aplicações, media e serviços Internet de uma forma prática e extremamente funcional. Segundo Pitts (2000), "The propose of Nautilus is to deliver a dynamic and rich user experience".

Este projecto funciona como uma extensão da interface GNOME para Linux, cujo proposta consiste no desenvolvimento de uma interface gráfica simples de usar, como nos populares sistemas Macintosh e AOL. Desta forma, este programa seria o principal componente de acesso a informação dentro do GNOME, tal como acontece com o *Windows Explorer* dentro do *Microsoft Windows*.

Este projecto encontra-se ainda activo, apesar de a empresa Eazel já ter deixado do existir. Este projecto encontra-se a ser desenvolvido segundo a norma GPL.

5.6.3 Análise SWOT

Tabela 5: Análise SWOT da Eazel

Pontos Fortes	Pontos Fracos
- Provas dadas e elevada experiência dos seus fundadores;	- Dependência do sucesso comercial do Linux;

- Interface gráfica agradável e fácil de usar (foco na usabilidade); - Credibilidade da sua equipa de gestão; - Parceria com a Red Hat.	- Dependência de financiamentos "venture capitals"; - "Layoff" de trabalhadores; - Dependência do sucesso da plataforma de base Nautilus; - O valor da solução depende exclusivamente da oferta em termos de "add-on" à solução Nautilus.
Oportunidades	**Ameaças**
- Alguns sinais de crescimento do Linux para os utilizadores "desktop"; - Contribuições de outros potenciais desenvolvedores de soluções open source	- Concorrência do sistema operativo Microsoft; - Concorrência das interfaces da Corel e MandrakeSoft; - Grau de adesão da comunidade open source.

5.6.4 Competências Centrais e Vantagens Competitivas

No caso da Eazel consideraram-se as seguintes competências centrais:

1. Experiência e passado dos seus colaboradores;
2. Rede de contactos dos seus fundadores;
3. Interface gráfica agradável;
4. Credibilidade da sua equipa de gestão;
5. Parceria com a *Red Hat*;
6. Notificação de *updates* automáticos;
7. Foco no desenho de soluções com elevados padrões de usabilidade.

Da concretização destas competências centrais podem-se construir vantagens competitivas que se centram, essencialmente, em 3 níveis:

- Características inovadoras do seu produto [1, 6] – o *Nautilus* possui um conjunto de funcionalidades que o tornam único e diferenciador face aos demais concorrentes de interfaces gráficas e gestão de ficheiros disponíveis na versão Linux;
- Facilidade de utilização [3. 7] – enfoque na construção de uma interface gráfica muito simples de usar e poderosa que substituísse com vantagens a do sistema operativo Windows;
- Curva de aprendizagem muito reduzida [7] – a curva de aprendizagem para um novo

utilizador *desktop* do Linux é mínima, facilitando desta forma a sua adopção e diminuindo as resistências iniciais à utilização de uma ferramenta desconhecida do mercado até ao momento.

5.6.5 Estratégia de Criação de Valor

A Eazel apostou no desenvolvimento de uma estratégia de criação de valor apostando na liderança pela diferenciação. A solução Nautilus é uma solução inovadora e possibilita que utilizadores *desktop* Linux passem a dispor de uma interface gráfica fácil de usar e com grande ênfase na usabilidade que irá permitir automatizar todo o processo de gestão de ficheiros.

5.6.6 Clientes

A Eazel encontrava-se muito dependente dos parceiros que conseguiria reunir em torno deste projecto. O seu mercado alvo não era directamente o cliente final, mas sim o mercado primário que é constituído por todas as empresas distribuidoras de software Linux e empresas de software integradoras de soluções *open source*.

Consequentemente, é raro que o conhecimento do mercado dos clientes imediatos (mercado derivado) seja suficiente para uma boa análise da evolução do mercado. Na realidade são os diferentes mercados finais (mercados primários) que orientam a evolução do mercado derivado (Coutinho, 2005).

Destacam-se as seguintes três parcerias mantidas pela Eazel no âmbito do seu projecto *Nautilus*:

- Parceria com a *RedHat* e a *Sun Microsystems Inc.*;
- Parceria com a *Dell*;
- Parceria com a *Linuxcare*.

A parceria com a *RedHat* iria permitir instalar previamente a solução *Nautilus* na distribuição Linux da *RedHat*, que controla actualmente cerca de 80% das soluções Linux no mercado norte-americano. Ainda segundo o vice presidente da Eazel "the most exciting thing in this deal from our perspective is that Red Hat is validating the model of delivering services to desktop users of Linux" (Christensen, 2001). No mesmo sentido, surge a parceria com a *Sun* que iria possibilitar a instalação prévia do *Nautilus* nos sistemas operativos *Solaris Unix* da *Sun*.

A *Dell*, líder mundial na concepção e venda directa de sistemas informáticos, estabeleceu uma parceria com a Eazel, devido às expectativas crescentes por parte da companhia de crescimento da adopção de software livre nos seus computadores. Desta forma, o software *Nautilus* seria instalado nos seus sistemas operativos Linux.

A parceria com a *Linuxcare* iria permitir disponibilizar serviços de suporte de correio aos clientes do ambiente de rede da Eazel, para além dos serviços já disponibilizados pela Eazel aos utilizadores do sistema *Nautilus*. Para além disso, a *Linuxcare* iria disponibilizar um serviço de gestão de conhecimento através da captura de documentação, actualizações de software e manutenção do sistema de FAQ's.

5.6.7 Conclusão

A Eazel apostava no desenvolvimento de uma única solução *open source* base denominada por *Nautilus* e na concepção de serviços de suporte a esta solução. São nestes serviços de suporte que se baseava o modelo de negócios da Eazel.

O *Nautilus* é um projecto inovador que foi concebido de forma a atender às necessidades dos utilizadores de soluções *desktop*. Uma das razões mais apontadas para a fraca adesão ao sistema operativo Linux por parte da comunidade de utilizadores de computadores *desktop* prende-se com o grau de dificuldade de habituação à interface Linux e às dificuldades de realização de tarefas rotineiras. Assim, este projecto vinha de encontro a este público alvo, através da criação de uma interface extremamente atractiva e funcional para a gestão de ficheiros.

Contudo, o sucesso deste projecto encontrava-se dependente do sucesso de vendas do sistema operativo Linux para a comunidade de utilizadores de computadores *desktop*. Na altura, existia a forte convicção que iria existir uma taxa de crescimento muito acentuada desta comunidade, motivado pelo seu rápido crescimento ao nível de servidores: "Linux is moving much more rapidly than we thought. We had projected it would be No. 2 in 2002 or 2003. It happened in 1999". (Kusnetzky, 2000).

Contudo, este crescimento não tinha sido traduzido em termos de rentabilidade para as empresas distribuidoras de sistemas operativos Linux, como se pode verificar no seguinte artigo "But sales of Linux brought in only $32 million for the whole year, less than 1 percent of the $5,7 billion market". (Shankland, 2000). Para além disso, existiam apenas alguns sinais de crescimento de utilizadores *desktop* do Linux, permanecendo o Linux a ser predominantemente usado ao nível dos servidores.

Simultaneamente a concorrência era extremamente feroz, não só por parte de companhias como a *Corel* e a *Mandrakesoft* que também estavam a fazer esforços no lançamento de uma versão Linux "user-friendly", mas também por parte da Microsoft que não queria deixar escapar a sua hegemonia no mercado de utilizadores de computadores *desktop*.

Apesar de a empresa Eazel ter fechado as portas 16 meses após a sua fundação, este projecto continua activo dentro da comunidade *open source*. Esta é tida como uma das principais vantagens dos projectos *open source*, já que o encerramento de uma empresa na significa necessariamente o abandono do projecto.

5.7 Caso da SourceXChange

5.7.1 Enquadramento da empresa

A Collab.Net é uma empresa na área das tecnologias de informação voltada para o desenvolvimento e fornecimento de serviços na área do software colaborativo e que baseia grande parte das suas soluções nos princípios de software livre.

A Collab.Net foi fundada em Julho de 1999 e tem recebido ao longo dos últimos anos

importantes investimentos por parte de grandes companhias na área de hardware, software e de serviços, nomeadamente a *Dell Ventures*, a *Intel Capital*, a *Novell* e a *Oracle corporation*.

Actualmente a Collab.Net centra a sua actividade no desenvolvimento de soluções para as seguintes áreas:

- Mercado colaborativo de desenvolvedores;
- Repositório de informação com capacidades de gestão colaborativa de projectos;
- Serviços profissionais que facilitem o desenvolvimento de software em equipas geograficamente dispersas.

5.7.2 Produtos / Serviços

Nesta dissertação pretende-se estudar unicamente uma das soluções lançadas e promovidas pela Collab.net, que diz respeito ao SourceXchange.

O SourceXchange foi um projecto lançado no ano de 1999 e que foi oficialmente cancelado em Abril de 2001, devido à falta de empresas dispostas a pagar pelos serviços oferecidos pelo SourceXchange.

Este serviço foi idealizado pelo co-fundador da *Apache Software Foundation*, Brian Behlendorf, e foi concretizado com o auxílio da *O'Reilly & Associates* e *HP*. A ideia era um criar um espaço na web para que as empresas contratassem desenvolvedores na área de *open source* para os seus projectos.

Este projecto apesar de ter sido cancelado voltou a ser aberto por outra empresa na Alemanha, denominada por *OpenSourceXperts*. A ideia desta empresa é pegar na ideia original do projecto do SourceXchange e torná-lo num projecto mais abrangente e mais ambicioso para que possa suscitar uma maior taxa de adesão por parte das empresas.

5.7.3 Análise SWOT

Tabela 6: Análise SWOT da SourceXChange

Pontos Fortes	**Pontos Fracos**
- Ideia inovadora de "marketplace" de desenvolvedores open source; - Suporte e apoio financeiro por parte de diversas empresas tecnológicas.	- Tempo de entrada no mercado; - Credibilidade da empresa; - Falta de casos de sucesso e "provas" das suas competências centrais; - Problema no controlo e segurança da propriedade intelectual; - Falta de aposta na área do marketing.
Oportunidades	**Ameaças**

- Outsourcing de projectos à escala global; - Sistema de contratação de recursos humanos; - Aumento do número e da importância dos projectos open source para as empresas; - Talento e capacidades técnicas da comunidade open source.	- Taxa de adesão da comunidade; - Cópia do modelo de negócio; - Falta de barreiras à entrada de novos concorrentes.

5.7.4 Competências Centrais e Vantagens Competitivas

No caso do SourceXchange consideraram-se as seguintes competências centrais:

1. Inovação de processo;
2. Gestão de plataformas web;
3. Experiência na concepção de soluções colaborativas;
4. Gestão de currículos de desenvolvedores *open source*;
5. Notificação automática dos currículos por perfis.

Da concretização destas competências centrais podem-se construir vantagens competitivas que se centram, essencialmente, em 4 níveis:

- Primeira empresa no mercado usando este modelo de negócio [1, 3, 4] – existe claramente uma inovação ao nível do processo de recrutamento de colaboradores por parte de uma empresa;
- Número de desenvolvedores *open source* registados [1, 4] – a adesão por parte da comunidade *open source* que viu nesta solução como uma forma de divulgar as suas actividades, procura de novas oportunidades e valorização profissional;
- Custo dos seus serviços [2, 4] – os custos dos serviços por si disponibilizados são mais baixos do que as empresas de recrutamento especializadas neste domínio;
- Simplicidade de utilização [4, 5] – tanto para as desenvolvedores como para as empresas torna-se muito simples usar este portal e contratar um novo colaborador que vá de encontro ao perfil desejado pela empresa cliente.

5.7.5 Estratégia de Criação de Valor

O SourceXchange foi lançado recorrendo fundamentalmente a um estratégia de criação de valor baseada simultaneamente na liderança pelo custo e na diferenciação. A sua opção estratégica passou pelo lançamento de um *web site* bastante simples de usar no qual os desenvolvedores *open source* poderiam afixar o seu currículo e as empresas poderiam divulgar as suas ofertas de trabalho. Fundamentalmente, procurava-se diminuir os custos com o processo de recrutamento e agilizar e tornar mais rápido todo este processo.

5.7.6 Clientes

A Collab.net possui um grande número de clientes dividido por diversos sectores de actividade, respectivamente empresas de tecnologias de informação e software, telecomunicações, governo, sector financeiro e serviços. No conjunto destes sectores destacam-se empresas como a *Intel*, *Sun*, *Sybase*, e *grupo Allianz*.

No caso específico do projecto SourceXchange torna-se importante distinguir dois tipos de clientes, que possuem estratégias distintas de entrada em cada um desses mercados. Um dos mercados diz respeito à captação do interesse da comunidade *open source* e dos seus colaboradores, de forma a estes disponibilizarem os seus currículos para as empresas, que se mostram interessadas no desenvolvimento de projectos *open source*.

Não foi possível obter um valor preciso sobre o número de desenvolvedores *open source* registados no sítio *web* do SourceXchange, mas segundo informações recolhidas no sítio *web* da Collab.net o número de desenvolvedores registados situava-se na ordem dos vários milhares de participantes. Ainda e segundo a empresa OpenSourceXperts.com, que deu seguimento à ideia original deste projecto, esta empresa regista actualmente 2684 colaboradores *open source* registados no início de Fevereiro de 2006.

Um outro mercado diz respeito à captação do interesse por parte das empresas que se mostram interessadas em recrutar colaboradores com competências no domínio de soluções *open source*.

Também, neste caso não foi possível obter informações sobre o número de empresas registadas no sítio web do SourceXchange. Contudo, foi possível obter informações sobre o número de empresas e ofertas de emprego registadas no início de Fevereiro no sítio web da OpenSourceXperts.com. A informação aí presente indica a existência de 63 empresas registadas, 107 ofertas de emprego e 70 cursos de formação (OpenSourceXperts, 2006).

5.7.7 Conclusão

O projecto SourceXchange apostou num projecto inovador ao nível da inovação de processos de negócio que consistiu na concepção de um portal web para divulgação de ofertas de trabalho para desenvolvedores na área do software *open source*.

Contudo, este projecto viria a ser cancelado pela empresa Collab.net em Abril de 2001, devido à fraca adesão por parte da comunidade empresarial.

Segundo (Mills apud Weiss, 2001):

"The big problem was that many companies preferred to seek aplication development help from known entities with whom they already had established working relationships. The model they wanted to use was the model of a trusted contractor, rather than canvassing an online collaborative site such as SourceXchange".

Esta afirmação não deixa de ser surpreendente atendendo ao facto de cada vez mais as empresas optarem pelo *outsourcing* de actividades e pelo aumento do número de vínculos com carácter provisório a cada uma das empresas e pela contratação de pessoas alocadas a cada projecto e não à empresa. Uma analista do *Hurwitz Group* afirmou "I was surprised

by the site's closure, especially because the softening economy makes it more likely that companies will seek outside Project help instead of hiring full-time employees. Maybe it was a case of being early to market" (Corbo apud Weiss, 2001).

Simultaneamente surgiram suspeitas de reduzido empenhamento por parte da empresa responsável pelo projecto, a *Collab.net*, que focou a sua atenção em outros projectos, nomeadamente no projecto *SourceCast*, que tinha por missão o desenvolvimento de um "online marketplace".

A SourceXchange teve ao longo de todo o seu ciclo de actividade uma forte concorrência por parte das empresas de recrutamento instaladas, que possuem uma superior imagem de marca e qualidade dos seus serviços junto das empresas, comparativamente ao projecto SourceXchange. Para além disso, não existiam barreiras à entrada de novos concorrentes no mercado, o que se viria a confirmar anos mais tarde com o lançamento da empresa *OpenSourceXperts*.

A *OpenSourceXperts* tenta dar seguimento a este projecto e alargar o leque de serviços disponibilizados, nomeadamente nos seguintes níveis:

- Promover o software livre nas mais variadas áreas de negócio;
- Encontrar desenvolvedores especialistas *open source* por região, país ou estado;
- Anunciar ofertas de emprego e especificação dos seus requisitos;
- Anunciar prémios e tendências dentro da comunidade OS;
- Divulgar acções de formação no domínio de tecnologias OS.

5.8 Caso da PyLabs

5.8.1 Enquadramento da empresa

A Pylabs foi uma empresa portuguesa fundada no início de 2003 e que cessou a sua actividade em Janeiro de 2004. Esta empresa tinha a sua sede no Tagus Campus em Lisboa e foi formada a partir de alguns técnicos oriundos da *Oniway*.

A PyLabs surgiu como um projecto idealizado pelo que foi o CEO da empresa, Engº Hugo Ramos, e posto em prática por mais quatro sócios, com o objectivo de colmatar uma falha na produção de tecnologias *web* em Portugal viradas para o *open source*.

Esta empresa usava na concepção das suas soluções apenas tecnologias do domínio *open source*, que normalmente passavam por uma arquitectura genérica do tipo Zope mais MySQL. Contudo, a esta arquitectura de base eram adicionados novos módulos no domínio *open source*, muitos deles desenvolvidos internamente recorrendo à linguagem de programação Python.

5.8.2 Produtos / Serviços

Nesta dissertação analisa-se apenas uma solução destinada à gestão de documentos em

portais colaborativos que foi desenvolvida pela Pylabs. Esta solução destinava-se a médias empresas, que possuem diversos departamentos e colaboradores distribuídos geograficamente e que precisavam de ter um repositório *web* para armazenamento e gestão dos seus documentos. Para além disso, esta solução deveria disponibilizar uma interface web de acesso a esta informação por parte de utilizadores externos a cada um dos projectos; teria pois, de haver diferentes vistas sobre a mesma informação e permissões de controlo de acessos.

Das ferramentas *open source* que foram usadas para a concepção desta solução destacam-se o Zope e Plone.

O Zope é um acrónimo para "Z Object Publishing Environment" (Ambiente Z para publicação de objectos que foi desenvolvido pela *Digital Creations Inc*., hoje *Zope Corporation*.

O Zope tem aproximadamente 85% do seu código escrito em Python, tendo o resto sido convertido para C++, de forma a maximizar o desempenho da sua aplicação. Encontra-se registado com a licença ZPL (*Zope Publish License*), que é uma licença open source compatível com a GPL.

Esta plataforma é basicamente um ambiente para construção e gestão de aplicações web com especial foco ao nível da administração de conteúdos. Segundo (Martins, 2005) no Zope não existem páginas estáticas, ou seja, todas as páginas de um servidor são geradas dinamicamente (no momento em que o servidor recebe o pedido do navegador).

O Zope possui o seu próprio servidor web (*ZServer*) e possui duas linguagens internas de apresentação e *script,* denominadas respectivamente *DTML* e *ZPT* e possui a sua própria base de dados (ZODB), que é um banco de dados transaccional orientado a objectos, e que pode também interagir com outros bancos de dados relacionais como Oracle, MySQL e PostgreSQL.

Por seu lado, o Plone é um sistema de código aberto de gestão de conteúdos que funciona sobre o servidor Zope. Possui uma larga e crescente comunidade de suporte e é uma solução especialmente vocacionada para uma *intranet* ou *extranet* como um sistema de publicação de documentos, um servidor de portal ou uma ferramenta de colaboração para áreas geograficamente distantes.

O Plone encontra-se licenciado sob a licença pública geral (GNU), a mesma licença que o Linux usa. Tal como o Zope, o Plone interage com a maioria dos sistemas d bancos de dados relacionais e pode ser executado em diversas plataformas, incluindo Linux, Windows, Mac OS, Solaris e BSD.

Ao nível de bases de dados, verificou-se que grande parte das soluções desenvolvidas pela PyLabs baseavam-se no sistema de gestão de base de dados (SGBD) MySQL, que é compatível com a utilização do Zope e Plone.

O MySQL é um sistema de gestão de base de dados, que utiliza a linguagem SQL como interface de acesso aos dados. É actualmente um dos bancos de dados mais populares com mais de quatro milhões de instalações pelo mundo.

As suas características mais importantes são:

- Elevada portabilidade (suporta praticamente qualquer plataforma actual);
- Grande compatibilidade (existem *drivers* ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Java, C/C++, Python e Perl);
- Bom desempenho e estabilidade;
- Reduzida exigência em termos de recursos de hardware;
- Facilidade de uso.

Para além disso, a empresa MySQL AB disponibiliza uma versão de software livre do seu popular sistema de gestão de base de dados, tendo sido esta a principal razão para a sua escolha.

Contudo, o sistema de gestão de base de dados MySQL apresenta alguma limitações face a problemas complexos ao nível do modelo relacional, com interdependências entre várias tabelas e quando se torna necessário armazenar um grande volume de informação e garantir mecanismos de *backup* automáticos.

Apesar destas limitações, a PyLabs optou pela utilização do MySQL, pois a complexidade das suas soluções não eram muito elevadas e não se colocava em causa em nenhum dos seus projectos as limitações anteriormente apresentadas. Em qualquer dos casos, seria relativamente fácil optar por se utilizar um SGBD mais robusto como o PostgreSQL que também se enquadra no domínio do software livre.

5.8.3 Análise SWOT

Tabela 7: Análise SWOT da PyLabs

Pontos Fortes	**Pontos Fracos**
- Uso de tecnologias open source e comerciais; - Utilização de tecnologias open source inovadoras; - Suporte técnico posterior à implementação.	- Falta de experiência de gestão dos seus colaboradores; - Espaço geográfico de intervenção da empresa bastante restrito; - Dificuldades de internacionalização; - Falta de visibilidade da empresa.
Oportunidades	**Ameaças**
- Aumento da complexidade da gestão da informação nas empresas;	- O Plone encontrava-se ainda em fase de desenvolvimento, não existindo uma versão

- Aumento da importância dos arquivos digitais; - Dispersão das organizações e das suas actividades; - Aposta no software *open source* por parte das entidades governamentais.	perfeitamente estável; - Elevado grau de concorrência por parte das empresas instaladas; - Reduzido grau de adesão das empresas clientes.

5.8.4 Competências Centrais e Vantagens Competitivas

No caso da PyLabs consideraram-se as seguintes competências centrais:

1. Integração de soluções *open source*;
2. Dedicação e competência técnica dos seus colaboradores;
3. Estrutura organizacional flexível da empresa;
4. Experiência de desenvolvimento em Python;
5. Experiência de programação em ambiente *web*;
6. Suporte de múltiplas base de dados.

Da concretização destas competências centrais podem-se construir vantagens competitivas que se centram, essencialmente, em três níveis:

- Flexibilidade da empresa [2, 3] – A dedicação dos seus colaboradores aliada à sua dimensão e estrutura organizacional da empresa permite-lhe ter uma maior flexibilidade na resposta ao cliente e um menor *lead-time* na resposta a uma proposta do cliente;
- Personalização das soluções à medida de cada cliente [4, 5, 6] – cada solução é desenvolvida e adaptada de acordo com as especificações de cada cliente;
- Custos mais baixos das suas soluções [1, 2] – o facto de se usar unicamente soluções *open source* permite à PyLabs ter custos mais baixos que as restantes concorrentes no mercado.

5.8.5 Estratégia de Criação de Valor

A PyLabs seguiu uma estratégia de negócio baseada simultaneamente na liderança pela diferenciação e pelo custo. A opção pela utilização do software livre foi uma estratégia deliberada na criação de soluções diferenciadas e de maior qualidade para o cliente.

Um dos aspectos mais importantes, diz respeito à flexibilidade de encontrar no domínio *open source*, a ferramenta mais apropriada para cada cliente específico. Isto permitia ir de encontro às especificidades do cliente e construir soluções robustas mas simultaneamente extremamente flexíveis.

Uma outra vantagem diz respeito à independência de fornecedores, o que lhe permitia ter custos no desenvolvimento do software muito mais baixos e utilizar as tecnologias que

melhor se adequava a cada necessidade. Isto era determinado pela PyLabs e pelos seus clientes, e não por qualquer fornecedor de software existente no mercado.

Simultaneamente, existia também uma liderança pelo custo, pois como se referiu anteriormente a não existência de um fornecedor comercial permitia-lhe ter custos no desenvolvimento do software muito mais baixos, pois não existiam despesas de licenciamento de software. Este é um dos factores que a empresa crê ir-lhe permitir captar novos clientes no mercado das PME's.

5.8.6 Clientes

Os principais clientes da PyLabs foram o Estado Português e também algumas PME's, que se localizavam na zona metropolitana de Lisboa.

Ao abrigo de acordos de exclusividade e confidencialidade não se torna possível apresentar uma lista de entidades que foram clientes da PyLabs. Apenas se torna possível mencionar que estes projectos envolveram várias áreas das empresas, como é o caso da gestão da intranet de uma instituição, integração com módulos de gestão de projectos e projectos de financiamento.

Em termos de soluções tecnológicas foram sobretudo usadas arquitecturas que se baseavam na integração das seguintes tecnologias e plataformas, respectivamente: Linux, Python, Zope e MySQL.

5.8.7 Conclusão

A PyLabs foi uma empresa que apostou na criação de soluções diferenciadoras face aos seus concorrentes. Para tal, a sua aposta na diferenciação baseou-se no facto de usar na concepção das suas soluções software livre.

Esta empresa viria a cessar a sua actividade em Janeiro de 2004, devido a várias razões de índole pessoal, mas o mais importante foi o facto de a empresa não estar a lucrar de acordo com o que estava previsto no plano de negócios. As expectativas de desempenho da empresa ficaram claramente abaixo do esperado, daí que a melhor opção segundo o seu fundador passaria pelo seu desmantelamento.

As razões para tão fraco desempenho passavam sobretudo pela fraca adesão dos clientes e também ao facto da existência de inúmeras empresas competidoras no mercado de software.

Os clientes mostraram sempre muitas reticências pela opção de utilização de software livre em detrimento do software comercial. A confiança no software livre é ainda bastante reduzida no domínio das pequenas e média empresas (PME), especialmente em Portugal, pois a sua utilização é ainda bastante recente e muitas vezes encontra-se delimitada a um ambiente restrito. O CEO da Pylabs afirma que "este tipo de mentalidade em Portugal está ainda na idade média" (Ramos, 2006).

Neste sentido, e apesar das vantagens competitivas em termos do custo de software, o cliente não se encontrava convencido das vantagens e confiabilidade do software livre. O

preço mais baixo das soluções era visto com desconfiança e encarado como se fosse um produto inferior ao da concorrência.

A concorrência foi implacável para com a PyLabs e esta não arranjou forma de conseguir contornar esta grande adversidade. A disputa de cada cliente nesta área de negócio é extremamente feroz e grande parte das vezes a escolha das empresas sobre as soluções tecnológicas de suporte não recai sobre a melhor proposta em termos da relação qualidade - preço.

Segundo (Ramos, 2006) a Pylabs não conseguiu fazer face aos *lobbies* e compadrios das grandes empresas que, de uma forma ou de outra, ganhavam todos os concursos e propostas que surgiam.

6 Práticas e Regras para o Sucesso Empresarial

6.1 Considerações Introdutórias

Neste capítulo apresentam-se um conjunto de regras e boas práticas que devem ser atendidas, para facilitar o alcançar do sucesso empresarial numa organização que usa o software livre no seu processo de comercialização.

Inicialmente (Secção 6.2) apresentam-se os principais aspectos que afectam directamente o sucesso empresarial de uma empresa. Seguidamente (Secção 6.3) descrevem-se as principais particularidades de que se revestem os negócios na área da Internet. Posteriormente (Secção 6.4) analisa-se qual o melhor posicionamento na cadeia de valor da indústria e apresentam-se as mais adequadas fontes de vantagens competitivas e estratégias empresariais (Secção 6.5). Por último, analisam-se as novas formas de vantagens competitivas empresariais que surgiram nos últimos anos, respectivamente: capacidades dinâmicas das empresas (Secção 6.6), inovação empresarial (Secção 6.7) e *outsourcing* estratégico (Secção 6.8).

6.2 Aspectos Genéricos das Organizações

O sucesso de uma empresa na área das tecnologias de informação, quer se dedique à comercialização de software livre ou de software comercial, tem em comum diversos factores. Aliás, a grande maioria destes factores são também comuns às restantes indústrias e aplicam-se em grande parte das situações, às empresas em geral.

Desta forma, e recorrendo a uma análise da literatura nesta área, extraíram-se os principais factores que são fundamentais a um sucesso empresarial:

- Adequação e correcta integração das várias políticas funcionais da empresa, compreendendo as seguintes áreas funcionais: marketing, gestão das operações, gestão financeira e recursos humanos;
- Implementação do processo de gestão nas seguintes áreas transversais: planeamento, organização, liderança e controlo.

O <u>marketing</u> é uma das políticas funcionais mais importantes para qualquer empresa que se situa numa indústria extremamente competitiva, como é o caso das empresas na área das tecnologias de informação. Contudo, esta área é vista ainda com alguma desconfiança por parte de alguns empresários e clientes, que olham para esta abordagem como uma forma de "empurrar" os produtos para o cliente.

Esta é uma ideia errada e que convém ser alterada. O marketing é essencial como forma de captar as necessidades e preferências dos clientes face a um determinado produto, o que deve ser sempre feito no início da actividade empresarial ou no lançamento de novos produtos. O marketing permite, então passar de uma abordagem orientada para a produção para uma abordagem orientada para o mercado (Kotler, 1999).

No caso do software existe um conjunto de políticas de comunicação que podem ser usadas em favor da empresa, sendo as mais importantes a presença em feiras e exposições, a participação em conferências com os parceiros, as apresentações nas empresas e a divulgação dos produtos e serviços em revistas técnicas especializadas (sobretudo em empresas que adoptam estratégias de focalização). Para além disso, deve-se dar especial atenção à concepção e manutenção do *sítio web* da empresa, que é muitas vezes a porta de entrada de novos clientes na empresa. Aspectos como acessibilidade, usabilidade e actualização da informação são muito importantes para uma boa imagem da empresa junto do cliente.

A gestão de operações é vista como o núcleo das organizações. No caso do software esta noção é ainda verdadeira, mas a sua importância é manifestamente inferior ao caso das indústrias tradicionais. No caso do software esta área passa pelo desenvolvimento do código, integração de diferentes módulos e personalização da solução para o cliente.

Na área das tecnologias de informação verificamos que a grande maioria das empresas segue uma estratégia de produção "pull", na qual o início da concepção de uma solução só se verifica após o lançamento de uma ordem de encomenda, ou seja, depois de já existir um determinado cliente e depois de já se ter realizado a respectiva análise de requisitos da solução.

Um outro aspecto muito importante em toda esta fase diz respeito ao controlo de qualidade desse software. Por esta noção entende-se toda a respectiva certificação da qualidade do código e procedimentos associados. Cada vez mais o mercado olha para esta certificação, não como um aspecto diferenciador entre os diversos "players" no mercado, mas sim como um factor crítico para a escolha de uma dada empresa.

A gestão financeira é crítica para uma empresa e é a causa principal do encerramento da grande maioria das empresas. Contudo, em grande parte dos casos a gestão financeira não é primeira causa do problema, mas sim a consequência visível de uma má gestão aos mais diversos níveis.

No caso das empresas ligadas às TI torna-se importante mencionar que esta área é muitas vezes descurada, uma vez que aqui a grande maioria das empresas são PMEs, nas quais os seus fundadores possuem profundos conhecimentos técnicos, mas reduzidas competências ao nível da gestão. É neste caso que é amplamente recomendada a existência de um colaborador na área de gestão, de forma a garantir um equilíbrio em termos de competências técnicas e de gestão na equipa de topo que chefia e coordena o projecto.

Os recursos humanos são a matéria viva de uma organização. Eles são extremamente importantes para o sucesso da empresa e deve-se ter especial atenção às suas políticas de contratação e de manutenção.

Tal como se mencionou anteriormente torna-se fundamental que uma organização possua recursos humanos com diversas áreas de competências, nomeadamente em termos técnicos e de gestão. Os recursos humanos contribuem com a sua experiência, conhecimentos, motivação e dedicação para a empresa, daí que se torne fundamental assegurar a sua correcta gestão ao longo do ciclo da empresa.

Contudo, estas áreas funcionais só por si não conseguem resolver os problemas e desafios das empresas. Torna-se necessário incluir o respectivo processo de gestão que é transversal a cada uma das áreas funcionais anteriormente descritas, que compreende o respectivo planeamento, organização, liderança e controlo.

O planeamento tem por objectivo definir os objectivos da organização, para que seja possível estabelecer um conjunto de estratégias para atingir esses objectivos. Este planeamento não deve ser rígido e excessivamente formal, mas deve funcionar como linhas orientadoras do futuro da empresa.

A organização de empresas na área das tecnologias de informação costumam ser bastante achatadas ("flat") em termos de níveis hierárquicos, como forma de diminuir a burocracia interna, aumentar a sua flexibilidade e fomentar o espírito de inovação e cooperação.

O estilo de liderança contém essencialmente dois vectores de orientação: comunicação e motivação. No caso de empresas na área das TI a melhor opção passa por adoptar um estilo de liderança denominado por "teoria situacional de liderança", definido por Hersey e Blanchard em 1986. Esta teoria, apesar de já ter sensivelmente 20 anos ainda se adequa perfeitamente aos dias de hoje e defende que o líder deve ajustar o seu estilo de liderança à preparação dos seus subordinados e à sua autonomia.

O controlo é uma das últimas etapas na gestão das organizações que é muitas vezes lamentavelmente esquecido. É através deste processo que se procura assegurar que o desempenho corresponde aos objectivos e padrões de actuação definidos para a organização na fase de planeamento. O seu objectivo principal passa pela gestão da incerteza, detecção de irregularidades e identificação de oportunidades a nível interno e no mercado.

6.3 Negócios na área da Internet

Um grande número das empresas tecnológicas que usam software livre no seu processo de comercialização utilizam modelos de negócio predominantemente baseados na Internet. Este tipo de posicionamento acarreta diversas consequências para as empresas, muitas das quais positivas, mas outras também que podem constituir ameaças para essas empresas.

A tecnologia da Internet dá melhores oportunidades às empresas para estabelecerem posicionamentos diferenciados dos que davam as anteriores gerações de tecnologias de informação (Porter, 1996). Ainda segundo Porter, muitas empresas bem sucedidas são aquelas que usam a Internet como um complemento dos meios tradicionais de competição e não aquelas que colocam as suas iniciativas na Internet longe das operações estabelecidas. Em suma, a utilização da Internet para negócio, por si só, apenas excepcionalmente será uma vantagem competitiva (Porter, 1996).

Porter aponta que a Internet trouxe inúmeros pontos negativos e encara-a com alguma desconfiança. Segundo Porter a Internet tenderia a enfraquecer a lucratividade sectorial e a destruir toda a possibilidade de manter vantagens competitivas. A Internet teria vindo

então alterar o ambiente competitivo da indústria e originar as seguintes consequências nefastas:

- Deu demasiado poder ao lado do comprador;
- Reduziu barreiras à entrada nos negócios;
- Intensificou a rivalidade infra-industrial;
- Empurrou as empresas para guerras de preços suicidas;
- Expandiu o mercado geográfico trazendo ainda mais concorrentes para a empresa.

Contudo, esta visão especialmente negativa não é partilhada por outros autores como Peter Cohan (especialista em *e-business*) e Don Tapscott (responsável pela criação do conceito "Economia Digital"). Segundo estes autores a Internet não pode ser vista como uma mera ferramenta tecnológica, mas como um novo tipo de comunicação estruturante de uma sociedade relacional e de uma plataforma económica. A Internet não foi adicionada como complemento, mas como elemento central estruturante de um novo modelo híbrido (Tapscott, 1999).

Para além disso, a Internet foi um elemento fundamental na *webização* de empresas estabelecidas e é um dos principais elementos no modelo de negócios de novas empresas. O seu baixo custo, a sua facilidade de utilização e o seu enorme mercado alvo potencial torna-se numa ferramenta poderosa a ser usada pelas empresas. Contudo, a sua utilização revelou-se frágil especialmente em termos do modelo de receitas indirectas provenientes da publicidade e das comissões (Porter, 2001).

6.4 Posicionamento na Cadeia de Valor da Indústria

O correcto posicionamento de uma empresa na respectiva cadeia de valor é um dos aspectos mais críticos no sucesso de uma dada organização. O modelo de Porter é uma ferramenta que veio ajudar as empresas em todo este processo e que trouxe consigo diversas mais valias, nomeadamente:

- Compreensão profunda do ambiente competitivo em que a empresa opera e se posiciona;
- A rivalidade competitiva fornece o incentivo fundamental para a inovação e a tecnologia pode assegurar competências distintivas e melhores produtos e serviços;
- Uma competência distintiva deve ser sustentável a longo prazo, pressupondo a análise da concorrência e das condições de mercado.

Contudo, o modelo sobre o qual se baseou a análise de casos desta dissertação possui algumas lacunas que convém esclarecer e que permite atender às especifidades de cada uma das indústrias e empresas em questão.

Um dos aspectos que este modelo subestima é o poder das mudanças tecnológicas para transformar as estruturas industriais. As empresas e tecnologias não se encaixam

ordeiramente em estruturas industriais estáticas, mas são afectadas por desenvolvimentos tecnológicos que decorrem dos avanços da ciência e do saber, e este avanços têm sempre consequências para as empresas. São comuns avanços e descontinuidades em termos tecnológicos na área do software, que é uma área em constante mutação e em que se torna necessário ter uma elevada flexibilidade em termos de adopção da melhor tecnologia, adopção de "standards" e modelos de negócio.

As empresas possuem ainda diversos constrangimentos na escolha das suas estratégias de negócio e de inovação. Existem diversos factores que condicionam esta escolha, nomeadamente:

- A dimensão da empresa – que influencia a escolha entre estratégias tecnológicas "frente larga" (tip. Grandes empresas) e "focadas" (tip. PME's);
- A base estabelecida de produtos da empresa e as competências tecnológicas inerentes influenciam as áreas tecnológicas e os sectores industriais nos quais a empresa espera competir no futuro;
- A natureza dos produtos e dos clientes influencia fortemente as opções qualidade/custo dos produtos e serviços.

Segundo Porter, e com base no que foi considerado ao longo deste relatório, estabeleceu-se que uma empresa pode seguir três estratégias empresariais distintas, a relembrar: liderança pelo custo, liderança na diferenciação e focalização. Contudo, uma empresa não tem necessariamente de seguir uma destas estratégias, mas pode seguir uma das seguintes duas opções:

1. Liderança simultânea em várias componentes – cada vez mais e face à grande competitividade do mercado a nível global, as empresas vêem-se obrigadas a seguirem estratégias simultâneas de liderança pelo custo e liderança na diferenciação. Isto não quer dizer, que uma empresa possa ser simultaneamente a número um em ambas as componentes, mas não pode descurar completamente uma delas;
2. Posicionamento intermédio em cada um dessas componentes – empresas posicionadas com produtos de custo e qualidade intermédios podem obter ROI's mais elevados do que as que adoptam estratégias de custo baixo/alta qualidade ou custo alto/alta qualidade. Isto acontece, pois estas empresas endereçam diferentes segmentos de mercado com níveis de poder de compra intermédios.

Um outro factor em ter em conta diz respeito aos problemas de implementação da estratégia, que muitas vezes aparecem em diversas empresas. Torna-se necessário garantir a correcta e contínua transferência de informação de uma forma transversal à organização, de forma a atravessar as barreiras funcionais e organizacionais. Em suma, não basta ter uma boa estratégia, é também necessário implementá-la com sucesso.

Por último, assiste-se cada vez mais ao surgimento de um novo conceito que diz respeito ao conhecimento colaborativo em detrimento de uma ênfase excessiva no conhecimento organizacional. São hoje evidentes os benefícios das relações de cooperação (e não de

rivalidade) com fornecedores e clientes que se traduzem num aumento de competitividade que decorre não só de uma melhor satisfação do cliente, mas também de uma melhor eficiência de fornecimento.

Segundo o relatório geral sobre as actividades da União Europeia para a investigação e inovação (*in site europa.eu.int*, 2006), mais de 35% das empresas europeias afirmam que o conhecimento técnico que obtêm dos seus clientes e fornecedores é muito importante para as suas actividades de inovação e permitem-lhe definir directrizes para as suas estratégias competitivas.

6.5 Estratégia e Vantagens Competitivas

Para cada um dos casos de estudo analisados no capítulo anterior extraíram-se um conjunto de vantagens competitivas e estabeleceu-se uma estratégia competitiva para cada uma das empresas. Algumas das vantagens competitivas apresentadas são comuns a várias empresas e são simultaneamente apresentadas na literatura como pontos bastante importantes no sucesso empresarial na área das tecnologias de informação. Assim, passo a apresentar algumas das características citadas na literatura e simultaneamente presentes nos casos estudados no âmbito desta dissertação.

O primeiro factor diz respeito à focalização numa determinada área de negócio, que se encontra presente no caso de estudo da Cyclades e da Critical Software. A focalização é fundamental especialmente para as PMEs que optam por colocar os seus produtos num âmbito geográfico alargado. É esta focalização alinhada com as competências distintivas que permite à empresa servir melhor os seus clientes relativamente aos demais concorrentes.

Um outro factor mencionado, diz respeito à flexibilidade e agilidade no lançamento de novos produtos e de atender às necessidades do mercado. Torna-se necessário ter uma grande agilidade a nível organizacional para se conseguir estar atento ao mercado e repercutir esta informação ao longo de toda a cadeia organizacional da empresa, que pode ter à partida uma maior ou menor predisposição para assimilar esse conhecimento consoante a sua estrutura organizacional interna.

No âmbito deste factor, verificamos que as PMEs têm uma maior apetência para serem mais flexíveis que as grandes empresas, pois os níveis hierárquicos entre as chefias e os restantes colaboradores da empresa são muito menores, chegando em alguns casos a ter organizações totalmente planas.

A vanguarda e o pioneirismo tecnológico pode ser entendido como um factor com alguma importância, essencialmente na área de comercialização do software livre. O software livre não deve ser unicamente visto como uma alternativa para as empresas que querem produzir software e produtos mais baratos, mas deve ser visto como uma alternativa para a concepção de novos produtos que não existem até ao momento no mercado.

O software livre aparece assim como uma capacidade diferenciadora, pois só a integração de diferentes módulos permite ter um produto mais completo e simultaneamente inovador. Este produto distinguir-se-á da restante concorrência, não com base no seu processo, mas sim com base nas suas características únicas e diferenciadoras. Contudo, surge nesta altura um novo problema: será que estas características inovadoras são difíceis de copiar?

No meu entender, dificilmente estas características poderão ser sustentadas no longo prazo, mas tal situação também ocorre com o software comercial. É certo que grande parte do trabalho pode já estar acessível para toda a comunidade e desta forma um possível concorrente pode diminuir o ciclo de desenvolvimento desse produto para competir com a empresa instalada, mas uma situação idêntica acontece também com o software comercial, que vê existirem no domínio *open source* inúmeras soluções alternativas ao seu software.

A qualidade das soluções desenvolvidas é um ponto fundamental no sucesso empresarial de uma empresa que opta pela introdução de software livre no seu sistema de comercialização. E isto é especialmente importante no caso do software livre, pois este é ainda encarado por algumas empresas com alguma desconfiança, associando-lhe normalmente inúmeros problemas em termos de "bugs" e estabilidade.

A facilidade de personalização das soluções é também um ponto com alguma importância e que pode ser encarado de duas formas distintas:

- Personalização das soluções no cliente;
- Personalização de módulos de base de concepção das soluções.

Na 1ª situação o produto do cliente deve ser construído de forma a ter em conta a facilidade de introdução de novas alterações requisitadas pelo cliente. Uma nova funcionalidade exigida pelo cliente deve acarretar um esforço mínimo para a empresa.

Na 2ª situação tenta-se criar um conjunto base de módulos sobre os quais as soluções assentam e são desenvolvidas para os clientes, ou seja, uma nova solução pedida pelo cliente não deve originar o começo a partir do ponto zero do desenvolvimento de toda uma nova solução. Esta deve poder ser incorporada a um conjunto de módulo básicos, que visam essencialmente diminuir o "time-to-market" do produto no mercado.

A escalabilidade do produto é também um aspecto a ter em conta e que se encontra associado ao anterior. A facilidade de personalização não deve ser unicamente atendida na fase de concepção e instalação da solução no cliente, mas deve ser assegurada ao longo de todo o ciclo de vida de um produto, que contempla também a sua fase de manutenção no cliente.

O cumprimento de prazos é um factor igualmente importante no caso do software livre e software comercial. É um factor de grande importância nas mais variadas áreas industriais em que o factor de marketing mais importante se baseia no sucesso do atendimento a clientes passados. Neste tipo de mercados torna-se muito importante atender o primeiro cliente de um dado mercado com critérios de excelência.

A competência e experiência da equipa de desenvolvimento é um factor que tem repercussões imediatas na qualidade das soluções desenvolvidas pela empresa. Isto não

significa que não devam existir colaboradores jovens e dinâmicos ao serviço da empresa, apenas afirma a importância da experiência dos principais membros da equipa de desenvolvimento e também o grau de envolvimento desses colaboradores em projectos *open source*.

O apoio técnico personalizado é outro factor com grande importância no domínio da indústria. As soluções precisam de ser mantidas no cliente e este cliente deve poder optar pela melhor forma de fazer esta manutenção. Devem portanto serem acordados diferentes níveis de manutenção das soluções dependendo do cliente em causa.

Por último, a relação mantida com o cliente é um ponto fulcral. O cliente condiciona, só por si, a implementação de uma dada solução e pode condicionar fortemente o número de vendas futuras da empresa. Um cliente exigente é um dos melhores contribuidores para o sucesso empresarial, pois vai permitir aumentar o grau de exigência da organização, fomentar a construção de produtos cada vez mais robustos e, por consequência, atender novos mercados e clientes (Tidd, Bessant e Pavitt, 2005).

6.6 Capacidades Dinâmicas das Empresas

A capacidade de resposta face a alterações de mercado pode ser uma importante fonte de vantagem competitiva. Esta vantagem competitiva assenta:

- Num recurso ao nível da informação;
- Numa capacidade ao nível da flexibilidade.

A informação é essencial para se conhecer e acompanhar as alterações do mercado. Torna-se necessário criar redes de informação entre a empresa e os seus parceiros de forma a se ter acesso a um conhecimento privilegiado. Para além disso, e aliada à informação, a capacidade de a empresa ser flexível permite-lhe ajustar rapidamente a sua estratégia às alterações detectadas na envolvente externa.

As organizações são então vistas como sistemas abertos nos quais a sua gestão é influenciada pelo seu meio envolvente. Simultaneamente, os vários elementos da organização estão inter-relacionados e funcionam como um todo na implementação da estratégia da empresa.

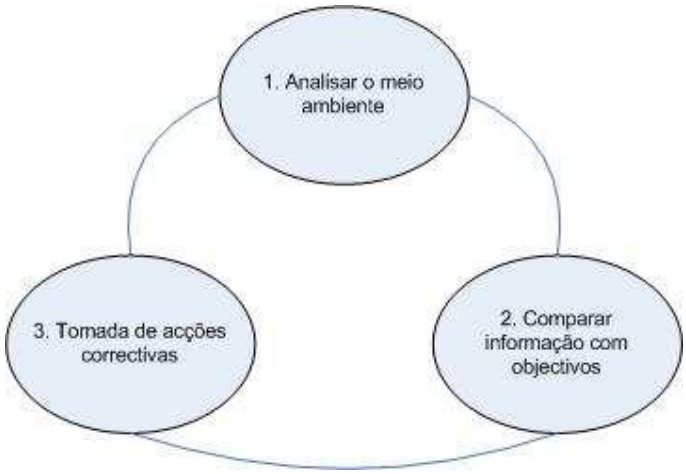
Desta forma, pode-se afirmar que não existe uma única forma de criar estratégia, nem há uma única melhor forma de organização. Formas diferentes funcionam bem em contextos específicos (Mintzberg e Quinn, 1996).

Uma outra abordagem que tem ganho crescentes adeptos nos últimos anos diz respeito à teoria "Learning Organization" estabelecida inicialmente por Peter Senge em 1990. Segundo este autor a capacidade de aprendizagem é vital para o sucesso das organizações e estabeleceu dois tipos de aprendizagem que as empresas podem seguir:

- Aprendizagem por adaptação;
- Aprendizagem criativa.

A aprendizagem adaptativa pode ser descrita com base no seguinte modelo:

Figura 13: Esquema da aprendizagem adaptativa

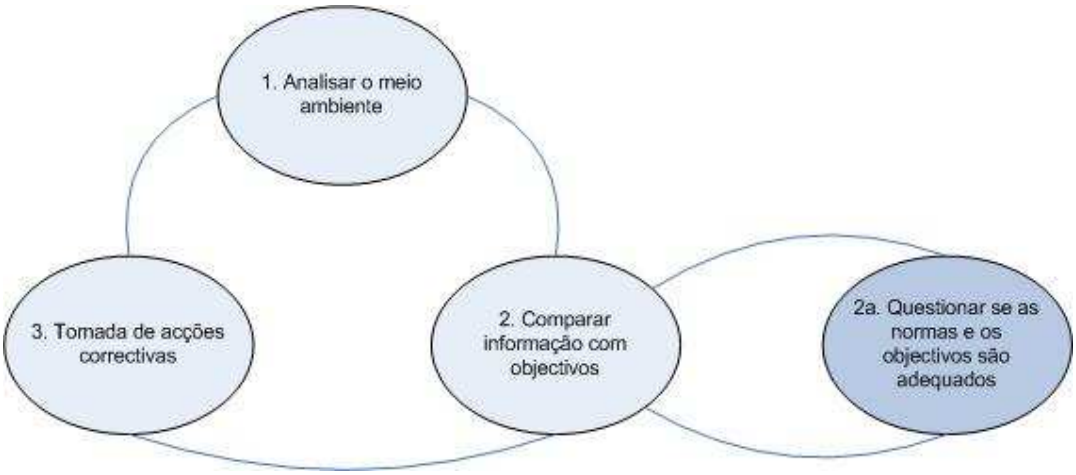


Fonte: (Senge apud Smith, 2001)

Da análise da figura 13 podemos verificar que a aprendizagem neste caso é totalmente reactiva e é vista como uma capacidade de reagir a alterações do meio envolvente. Assim, a organização aprende sem alterar significativamente os seus pressupostos de base.

Por seu lado, a aprendizagem criativa pode ser modelada da seguinte forma:

Figura 14: Esquema da aprendizagem criativa



Fonte: (Senge apud Smith, 2001)

Neste caso, e atendendo à figura 14, a aprendizagem é vista como criação e inovação, envolvendo pró-actividade e antecipação da mudança, pelo que a organização necessita de reformular os seus pressupostos básicos.

6.7 Inovação Empresarial

A inovação é outra das novas formas de vantagens competitivas e que tem sido largamente difundida pelos órgãos de comunicação social e pelos diversos Estados europeus.

A inovação permite não só criar vantagens competitivas, mas também alterar a estrutura da indústria em favor da empresa. Muitas das empresas mais bem sucedidas têm quebrado as regras de jogo da sua indústria, inovando ao nível de produtos e processos.

Segundo Porter, existem duas estratégias de posicionamento de uma empresa face à inovação segundo Porter, respectivamente:

- Liderança pela inovação;
- Seguidores na inovação.

Na liderança pela inovação o objectivo da empresa passa por liderar o mercado através da superioridade tecnológica. Para isto ser possível, têm que existir as seguintes condições:

- Comprometimento forte em criatividade e assumir de riscos;
- Existência de laços fortes às principais fortes de conhecimento;
- Capacidade de resposta rápida às necessidades dos clientes.

Este tipo de posicionamento traz diversos benefícios para as empresas nomeadamente:

- Primeira escolha ao nível do posicionamento e segmentos de mercado a atingir;
- Como pioneiros, ditam as "regras do jogo";
- Vantagens ao nível da distribuição dos seus produtos;
- Economias de escala e experiência.

Os seguidores na inovação têm por objectivo a chegada tarde ao mercado, mas de uma forma deliberada, e baseia-se no processo de imitação (aprendizagem) da experiência dos líderes tecnológicos. Desta forma existe:

- Comprometimento forte na análise da concorrência;
- Ênfase na engenharia inversa;
- Redução de custos e aprendizagem no fabrico optimizado.

Este tipo de posicionamento traz consigo os seguintes benefícios:

- Habilidade de se tirar partido de erros de posicionamento dos pioneiros;
- Habilidade de se tirar partido de erros ao nível do produto dos pioneiros;
- Habilidade de se tirar partido de erros de marketing dos pioneiros;
- Habilidade de se tirar partido da falta de recursos dos pioneiros;
- Habilidade para se retirar mais valias dos últimos desenvolvimentos tecnológicos.

6.8 *Outsourcing* Estratégico

Nas suas variadas formas o *outsourcing* está entre nós há vários anos, representando um sector económico que, globalmente, muitos estimam valer entre 100 e 150 milhões de euros apresentando uma taxa de crescimento anual entre 10 e 20% (Gonçalves, 2003).

Se bem que a maior notoriedade do *outsourcing* provenha da sua implementação ao nível dos sistemas de informação, o recurso a contratos desta natureza vai muito para além desta área, entrando em áreas contabilísticas, financeiras, recursos humanos, logísticas, etc. Toma, neste casos, a designação de BPO – *Business Process Outsourcing*.

No mundo actual de elevada complexidade e exigências crescentes, onde a especialização assume um extraordinário relevo, o recurso a qualquer uma das variantes do *outsourcing* pode garantir a renovação e a sobrevivência futura de qualquer organização.

A opção por se efectuar o *outsourcing* de determinadas actividades permite vulgarmente trazer benefícios para as empresas, tais como: redução de custos, focagem no "core business", melhor qualidade de serviço, poupança de mão-de-obra e acesso a recursos especializados. Estes benefícios resultam de uma atitude "passiva" na decisão do *outsourcing*.

Outra abordagem, muito mais importante e com cada vez mais aderentes é a atitude "activa", isto é, a procura objectiva de um parceiro que seja responsável pelo *outsourcing* de uma área da organização, mas que traga consigo *know-how* fundamental para a conquista de vantagens competitivas. Hoje o *outsourcing* tem que ser valorizado não apenas como uma via para resolver problemas de custo e eficiência, mas também pelo seu enorme potencial para injectar novas ideais, conceitos e estratégias na organização. Assiste-se hoje, portanto, e cada vez mais a uma passagem do *outsourcing* operacional para o *outsourcing* estratégico (Gonçalves, 2003).

Contudo, torna-se importante ter bem presente quais são as áreas da empresa em que se pretende realizar o *outsourcing,* que nem sempre são simples de seleccionar. Torna-se necessário construir uma solução de compromisso entre as mais valias decorrentes da minimização dos custos e a preservação das actividades críticas para a empresa internamente. Uma má política de *outsourcing* pode originar a passagem de conhecimento para um fornecedor ou um cliente e daí resultar um conjunto de novas ameaças para a empresa instalada, decorrentes de possíveis entradas de novos concorrentes no mercado ou até mesmo de ameaças ao nível de integrações a montante e a jusante.

Assim, convém que a empresa concentre esforços num pequeno conjunto de competências centrais e que opte pela subcontratação das actividades que não são estratégicas, ou que a empresa não desempenha particularmente bem. Deve-se portanto fomentar a criação de alianças estratégicas, criando-se redes de empresas que contribuem de forma cooperativa para o produto final que é oferecido ao cliente.

7 Novas tendências para o Software Livre

7.1 Considerações Introdutórias

Neste capítulo pretende-se fornecer uma visão sobre o impacto e o futuro de desenvolvimento de soluções *open source* a nível mundial.

Inicialmente (Secção 7.2) apresentam-se os principais limites das soluções *open source* e do seu actual esquema de desenvolvimento. Seguidamente (Secção 7.3) abordam-se as principais implicações económicas e sociais deste novo modelo de desenvolvimento, nomeadamente ao nível das empresas e da sociedade em geral. Por fim (secção 7.4) pretende-se lançar uma visão sobre o futuro do *open source* e o seu impacto nos próximos anos.

7.2 Limitações do Open Source

O modelo de desenvolvimento e comercialização de soluções *open source* apresenta também algumas limitações que não são consensuais na literatura consultada para esta tese. Há contudo, alguns aspectos limitativos que são considerados relativamente consensuais, apesar de o seu impacto para cada projecto ser também bastante discutível. Dentro destes factores merecem especial destaque:

- Desenvolvimento demasiado orientado ao desenvolvedor;
- Inadequada alocação de recursos;
- Necessidade de sustentabilidade de estratégias complementares de *open source*;
- Baixa capacidade de inovação;
- Maior dificuldade no estabelecimento de "standards".

No domínio *open source*, aquando da elaboração de um novo produto, o aspecto fundamental não é o que é tecnologicamente exequível ou que o mercado pretende, mas sim a visão pessoal de um desenvolvedor, que pode ter um maior ou menor fundamento com base no mercado.

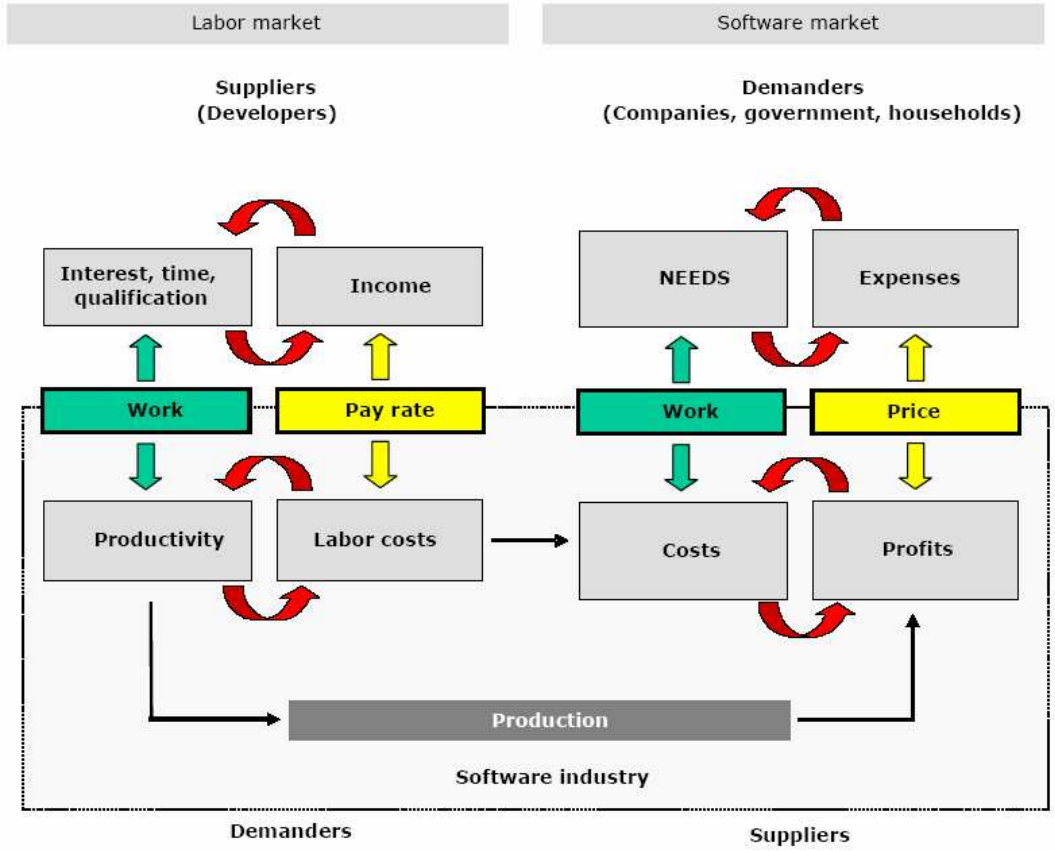
Mas esta visão é restrita e apenas foca nas áreas de competência e interesse do desenvolvedor, tendo este tipicamente um perfeito desconhecimento do valor do resto do mercado.

Por exemplo, engenheiros que apenas se focam nos aspectos tecnológicos do seu desenvolvimento, têm tendência a criar produtos sofisticados de alto desempenho e de elevado custo, que podem não ser procurados por um consumidor do mercado, pois estes não reconhecem valor a este tipo de produto ou não conseguem ter disponibilidade financeira para o adquirir (Backhaus, 1999). Esta situação é descrita por Backhaus no seu livro com o termo "Happy Engineering", ironizando estes desenvolvedores que ficam apaixonados pela sua tecnologia.

Acontece porém em muitos casos que o desenvolvedor é o próprio cliente do software, daí que a sua solução desenvolvida vai de encontro às suas necessidades. Contudo, e tal como se referiu anteriormente, a utilização desse software a uma escala mais alargada pode ficar comprometida.

Um aspecto determinante na distribuição de produtos não é a sua sofisticação em termos tecnológicos, mas a sua criação de valor para o consumidor. No caso do software proprietário, temos uma correspondência que pretende ser perfeita entre as capacidades e funcionalidades da tecnologia e as necessidades dos consumidores.

Figura 15: Sistema de mercado do software proprietário

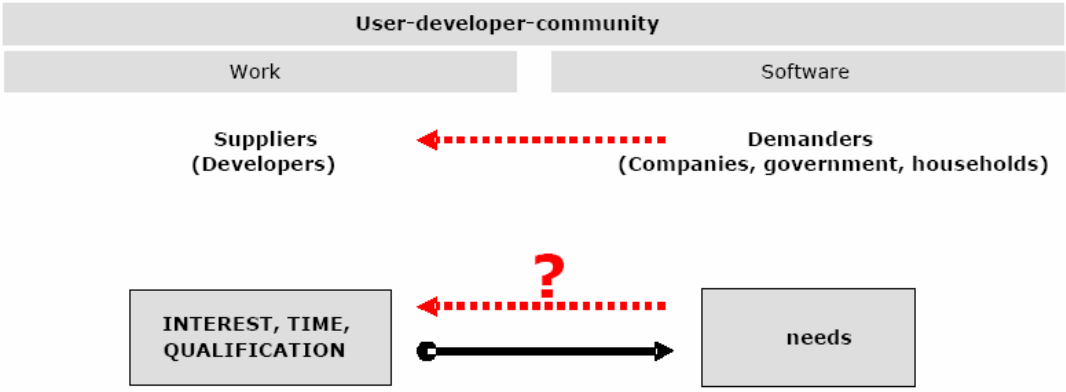


Fonte: (Kooths e Langenfurth, 2003)

Da análise da figura 15 podemos constatar que no software proprietário existe uma tentativa de se fazer uma clara correspondência entre os esforços necessários em termos de desenvolvimento de software (em termos de tempo e custos de recursos humanos) e a sua comercialização (em termos de custos e necessidades satisfeitas).

No caso do software livre esta situação não ocorre existindo um modelo com várias interrogações em aberto.

Figura 16: Sistema de mercado do software livre



Fonte: (Kooths e Langenfurth, 2003)

Da análise da figura 16 podemos verificar que existe uma clara dificuldade em identificar as necessidades do mercado e consequentemente dos custos e tempo de desenvolvimento necessários.

Uma outra limitação prende-se com a inadequada alocação de recursos. Os desenvolvedores de software *open source* consideram a sua actividade como uma forma de lazer e de melhorarem as suas actividades (ver capítulo atrás: motivações), não existindo um plano de alocação de tempo aos projectos em que se encontram envolvidos. Contudo, mesmo que não sejam imputados custos directos ao desenvolvimento de uma dada solução, existem sempre custos indirectos decorrentes de um custo de oportunidade do desenvolvedor que gastou parte do seu tempo nesse projecto.

Para além disso, não está estabelecida uma hierarquia de prioridade de projectos. Um dado desenvolvedor pode estar interessado no desenvolvimento de diversos projectos *open source* e pode alocar o seu tempo a eles da forma que bem entender, ou seja, não existe uma visão de conjunto de todos os projectos. Isto origina resultados muito díspares em termos de tempo de desenvolvimento de uma dada solução, que pode ser extremamente curto ou pode se alongar no tempo de uma forma indeterminada.

Contudo, esta situação não se aplica às empresas que apostam e desenvolvem software livre. Neste caso, os recursos humanos da empresa são afectados aos projectos *open source* da mesma forma que acontece com o software proprietário.

A sustentabilidade das estratégias de complemento das soluções *open source* é um dos problemas que se coloca às empresas que adoptam o software *open source* no seu processo de comercialização. Esta problemática foi largamente estudada ao longo desta dissertação e a resposta a ela baseia-se no grau de sustentabilidade que cada empresa consegue criar como suporte aos seus projectos *open source*. A área específica de cada projecto *open source* pode ser um factor primordial de sucesso para a empresa, mas a mesma deve estar associada a um conjunto de serviços que são comercializados e que não podem ser directamente obtidos através da comunidade *open source*.

Uma outra limitação situa-se ao nível dos procedimentos de inovação empresarial. No caso do software *open source* o actual modelo de inovação que se baseia principalmente

na protecção da propriedade intelectual e estabelecimento de patentes pode ser posto em causa.

Como foi visto nesta dissertação (capítulo 6.7) a inovação é muito importante para as empresas, pois estas conseguem retirar dividendos do facto de serem a primeira empresa no mercado a oferecer determinado produto ou solução. Contudo, no modelo de software livre as barreiras à entrada de um novo concorrente são mínimas, daí que esta vantagem tende a não ser sustentada no médio e longo prazo. Contudo, isto não significa que não exista inovação no software livre, apenas significa que os dividendos desta inovação não podem ser directamente obtidos a partir do código fonte do programa.

Por fim, uma outra limitação aparece ao nível da dificuldade no estabelecimento de "standards". Um "standard" é visto aqui como um padrão no desenvolvimento de soluções de uma dada classe, e este padrão serve de base aos restantes desenvolvedores e empresas no mercado na concepção das suas soluções.

No caso do software *open source* podem existir situações em que não existe consenso entre o rumo de desenvolvimento de uma dada solução, aparecendo uma rivalidade entre os desenvolvedores que "lideram" este projecto. Daí resultam situações de "fork" em que o projecto toma rumos diferentes o que origina diversos standards (Feller, Fitzgerald e Lakhani, 2005).

Contudo, se esta problemática é vista de uma forma consensual, o mesmo não acontece com o processo de criação de um "standard". O processo de criação de um standard é mais simples no caso do software *open source*, pois não existe uma rivalidade e competição entre empresas, daí a existência à partida de um standard dominante. E este standard é facilmente difundido e aderido por toda a comunidade, enquanto que no software comercial podemos ter várias empresas que seguem o seu próprio standard. Tal situação pode ser comprovada com base na seguinte afirmação:

"The department of justice in United States estimates there are 16,000 different methods used by local governments to classify the same 300 bits of information in their databases." (Kooths e Langenfurth, 2003).

7.3 Implicações Económicas e Sociais

O software livre tem tido ao longo destes últimos anos grandes impactos económicos e sociais. Este novo tipo de software veio alterar aspectos económicos base das organizações e veio possibilitar um maior acesso a tecnologias e informação que de outra forma não eram possíveis.

Ultimamente verifica-se que cada vez mais organizações privadas e entidades governamentais se interessam por este tipo de software e verifica-se uma gradual passagem da utilização do software proprietário para software livre. Em muitos casos estes dois tipos de software coexistem numa mesma organização em âmbitos distintos.

Actualmente os custos dispendidos em software e licenças são enormes, daí que este mercado se revele bastante atractivo para um grande número de empresas. Por exemplo,

no ano fiscal de 2003 o governo dos Estados Unidos da América despendeu mais de 58 biliões de dólares em produtos e serviços na área das tecnologias de informação. Existem mais de 4 milhões de computadores pessoais e portáteis ao serviço das agências federais. No total estimam-se que sejam gastos mais de 100 milhões de dólares anualmente associados a processos de licenciamento (Walker, 2004). Contudo, este número é apenas a ponta de um gigante icebergue, já que segundo a *IDC Corporation*, no conjunto as diversas entidades federais e locais de cada Estado despende no seu conjunto mais de 34 biliões de dólares por ano (Walker, 2004).

Desta forma, existe um enorme potencial de mercado que pode ser ganho pelas soluções *open source* e já existem largos sinais no mercado de que esta batalha começa a ser ganha em alguns dos seus segmentos. Este impacto não se reflecte unicamente no número de utilizadores finais, mas também nas empresas que começam a disponibilizar soluções de *open source* para os seus clientes e a cooperarem com outras entidades públicas e privadas em todo este processo de comercialização.

Existem claros sinais de que existe um crescimento por parte do software livre. Uma evidência desta situação é a empresa *Hewlett-Packard* (HP), em que 40% do seu volume de negócios provém da venda de servidores contendo o sistema operacional Linux e serviços que lhe estão associados. A HP é ainda responsável por 28,1% dos servidores Linux existentes no mercado a nível mundial, o que significa que a HP investiu fortemente neste mercado (Walker, 2004).

Aliás o Linux é um dos casos de sucesso do software livre como se pode verificar na seguinte informação:

"Linux server sales of nearly $3 bilion in 2003 are expected to rise to nearly $9 bilion by 2007, according to market research IDC. Over the same period, IDC believes Linux server shipments will increase from about 800,000 to about 2,5 bilion". (Cnet, 2004)

Na Europa encontrámos também diversos exemplos que comprovam a adesão ao software livre por parte de entidades governamentais. Um exemplo desta situação pode ser encontrado na cidade de Munique (Alemanha) que adoptou o sistema Linux nos seus computadores, não apenas por razões meramente económicas, mas segundo um estratégia qualitativa (Unilog, 2003). A empresa de consultadoria Unilog efectuou a seguinte recomendação às entidades governamentais germânicas:

"updating the Microsoft products currently in use to the current XP versions constitutes the technically easiest and economically soundest alternative for LHM. The monetary advantages of the Windows XP alternative outweight the next-expensive equipment alternative by approx. 2.46 million euros in principal value and by approx. 1.76 million euros in total value. The cost advantage of Windows XP as opposed to a pure Linux solution amounts to approx. 11.9 million euros in principal value and approx. 11.6 million euros in total value". (Unilog, 2003)

Da análise desta afirmação podemos constatar que de um ponto de vista meramente financeiro a solução proprietária é claramente superior. Contudo, a solução *open source* do ponto de vista qualitativo é encarada como a melhor solução, pois vai permitir aumentar o

nível de segurança dos seus sistemas e reduzir a sua dependência dos fornecedores de soluções de software (Acohido, 2003).

O software livre tem tido também fortes impactos em economias emergentes, nomeadamente no sudeste asiático, destacando-se países como a China e a Índia. A China possui actualmente uma forte comunidade *open source* e tem existido uma clara estratégia de independência das suas soluções tecnológicas de grandes grupos económicos localizados nos EUA e Japão. Como exemplo desta situação encontra-se a seguinte notícia:

"Aside from developing its own wi-fi standard, China is actively promoting open-source computing to reduce its dependence on Microsoft Windows". (Ásia Times, 2004).

Contudo, o software não possui apenas fortes impactos no tecido económico e empresarial, mas também possui uma elevada capacidade de produzir um impacto ainda mais significativo a nível social. O software livre pode e deve ser usado no combate à exclusão digital e a tecnologia deve estar disponível para a população em geral. A cidadania na era da informação impõe o direito a se comunicar, armazenar e a processar informações velozmente, independentemente de condição social, capacidade física, visual, auditiva, sexo, idade, raça, ideologia e religião (Silveira, 2005).

Existe portanto uma dimensão social do software livre especialmente nos países em vias de desenvolvimento, em que este software pode servir de base ao seu progresso. Assim, o acesso a determinado conjunto de software encontra-se facilitado, existe uma redução de custos na transferência tecnológica entre países desenvolvidos e em vias de desenvolvimento e existe ainda a possibilidade o desenvolvimento tecnológico e social destas populações.

7.4 Visão do Futuro

Fazer uma análise futura da evolução da tecnologia na área do software é sempre um risco muito grande. Contudo, optou-se por uma abordagem baseada em diversos cenários possíveis sobre a evolução do software livre e a sua adesão por parte das empresas e das entidades governamentais. Deu-se um especial destaque aos apoios provenientes de entidades governamentais, uma vez que são estes que possuem um maior número de responsabilidades ao nível da promoção de projectos na área da sociedade de informação e integração tecnológica das populações. Isto não quer dizer que o software livre não consiga progredir só por si mesmo, algo que realmente já foi demonstrado em diversos projectos *open source* de grande sucesso.

Optou-se por considerar os seguintes três cenários para a evolução do software livre:

- Cenário I - Nenhuma acção de apoio é tomada pelas entidades governamentais;
- Cenário II - Algumas acções de suporte são tomadas para promover o software livre;
- Cenário III - Diversas e variadas acções de suporte são tomadas para promover o software livre.

Cenário I

Neste cenário considera-se que não vão ser tomadas quaisquer medidas de apoio por parte de entidades governamentais ao software livre. Neste cenário a promoção do software livre passa unicamente pelas entidades privadas e pelas pessoas individualmente.

Neste sentido, o software livre poderá ser usado em nichos de mercado e em diversas entidades governamentais, que vêem neste software uma oportunidade para diminuir os custos de licenciamento. Contudo, não existirá uma visão estratégica do software livre por parte destas entidades, que apenas usam esse software e não fornecem o respectivo *feedback* ou apoios financeiros (directos ou indirectos) para a restante comunidade.

Este é aliás o cenário actual que se começa a tornar cada vez mais visível na grande maioria dos países a nível mundial, sobretudo na União Europeia. Segundo o grupo de estudo sobre o software livre:

" In fact, this trend is already starting to take shape in several countries (at least in Europe). Open Source will be preferred in many situations simply on economic or technical grounds, as a lower cost or technically better alternative to proprietary systems. In this case no specific benefit will be reaped by governments, or by the societies they serve, except for the direct cost saving on license" (working group on libre software, 2000).

Neste cenário podem surgir algumas ameaças para o movimento *open source*. As mais importantes, e que têm sido discutidas pela comunidade OS, são as seguintes:

- A estratégia designada por FUD (*fear, uncertainty, doubt*) seguida por organizações comprometidas com o software proprietário, que sentem em risco a sua supremacia e quota de mercado. Até agora, estas técnicas não têm sido um problema sério para o software *open source*, mas têm certamente condicionado a sua rapidez de expansão;
- A dissolução das principais vantagens do modelo *open source*. Isto pode acontecer à medida que surgem cada vez mais licenças de software livre que tendem a aproximar este tipo de software do software comercial. Isto pode originar fortes divisões no seio da comunidade *open source*, cujos membros podem seguir rumos diferentes;
- A falta de visão global dos membros da comunidade *open source*. Isto pode acontecer à medida que se aumenta o número de membros desta comunidade, que só se encontram interessados por determinados projectos específicos;
- O aparecimento de impedimentos legais, que podem condicionar e limitar a progressão do movimento *open source*. Uma destas ameaças diz respeito ao patenteamento do software.

Cenário II

Neste cenário faz mais sentido questionar a sociedade, administrações e governos em geral, de como estes podem beneficiar do software livre e como estes podem usar este software para criação das suas vantagens competitivas e de uma melhor sociedade, em vez de se olhar para a tradicional abordagem que questiona unicamente de como o software livre pode beneficiar de apoios governamentais.

Nesta situação, estas entidades (públicas e privadas) investirão algum do seu tempo e recursos para analisar a adequabilidade do software livre às suas áreas de intervenção e procurarão identificar as principais barreiras que podem impedir a adopção de tecnologias *open source*. Com o tempo, estas organizações irão identificar quais os projectos *open source* para os quais pretendem contribuir e irão entender mais claramente quais os benefícios trazidos em termos de flexibilidade, ao nível do ciclo de produção de software e em termos de adaptabilidade.

Para além disso, as entidades governamentais terão outras vantagens ao nível da sociedade de informação, tornando este software acessível para toda uma vasta comunidade. A melhor forma de fomentar o interesse pela tecnologia e de criar competências tecnológicas na sociedade em geral é criar um padrão de software que permita a sua mais ampla difusão. Desta forma, as entidades governamentais servirão como uma alavanca em todo este processo.

Cenário III

Este é claramente um cenário mais agressivo, que contempla o apoio directo ao software livre por parte de diversas entidades, nomeadamente as entidades governamentais. Contudo, esta visão "agressiva" da evolução do software livre pode ser posta em causa pela legislação, e esta pode tornar o que hoje parece uma visão "agressiva", num cenário perfeitamente comum nos próximos anos.

Como exemplo deste cenário, podemos considerar acções legislativas por parte de entidades governamentais e da União Europeia que visem dar preferência às soluções *open source* face às soluções comerciais na mesma área tecnológica. Uma outra acção interessante poderá ser o estabelecimento de fundos para o desenvolvimento de soluções *open source* alternativas às soluções comerciais em áreas que sejam consideradas relevantes para os governos, devido à sua posição estratégica, social ou económica.

Esta medida iria criar um enorme mercado na área de desenvolvimento de soluções e consultadoria *open source* e iria melhorar significativamente os conhecimentos tecnológicos na área das tecnologias de informação. Para além disso, como menciona o grupo de estudo em software livre criado no âmbito da Comissão Europeia para a sociedade da informação, existiriam alguns importantes impactos para os países europeus, nomeadamente ao nível da sua balança comercial, em termos de importação e exportação, que se encontra bastante desfavorável em relação aos Estados Unidos da América.

8 Conclusão

8.1 Principais Contribuições

O empreendedorismo de software livre enfrenta um conjunto de desafios, muitos dos quais, são também comuns ao processo de empreendedorismo de software proprietário. Contudo, existem alguns aspectos que são distintos, e sobre os quais convém dar um particular destaque por serem úteis a quem pretende lançar um negócio usando software livre. Estes aspectos induzem a um conjunto de regras e boas práticas que melhor irão preparar uma dada organização para lidar com os mais importantes desafios na área da comercialização deste tipo de software.

À partida, não se pode garantir que uma empresa que opte por software livre no processo de comercialização possa ter sucesso, mas o mesmo também se passa com o software proprietário. Da análise dos casos de estudo apresentados nesta dissertação (ver capítulo 5) verifica-se que, em grande medida, o insucesso não advém de causas relacionados com o software livre em si mesmo, mas sim de vários factores em termos organizacionais, financeiros e de recursos humanos, que são comuns a todas as organizações.

Um dos principais aspectos a ter em conta diz respeito ao estabelecimento de um plano de acção para cada uma das áreas funcionais de uma empresa. A empresa deve ser a mais homogénea possível e os seus mais diversos departamentos devem estar em consonância com a "missão" da organização. Todos eles devem estar alinhadas com os objectivos da empresa e não unicamente voltados para os seus desempenhos individuais. Aliás, isto aplica-se também às empresas de dimensão mais reduzida, que são a que têm maior expressão no domínio da comercialização do software livre.

Assim, é extremamente importante que exista um plano de marketing, rigor na gestão financeira e um excelente ambiente e qualificações dos recursos humanos. Estas várias vertentes devem ser suportadas transversalmente ao longo de todo o processo da gestão, que implica o seu respectivo planeamento, organização, liderança e controlo. Destes vários aspectos, destaca-se particularmente a liderança e controlo que são a causa primária de muitos dos problemas das empresas neste sector de actividade. A liderança deve ser assumida à partida de uma forma natural e não como sendo imposta e deve ser compreendida por todos os colaboradores da empresa. O controlo serve para corrigir eventuais situações futuras, pois uma empresa não deve ter complexos de assumir os seus erros e deve conseguir medi-los e quantificá-los (Hitt, 2003). Só assim se poderá tomar medidas futuras que visem corrigi-los.

A definição do modelo económico que vai suportar a comercialização do software livre é também um aspecto muito importante, sendo que esta situação só se aplica no caso da comercialização deste tipo de software. Como os dividendos não se podem extrair directamente do código fonte de um programa, é extremamente importante definir um modelo de negócio indirecto que permita trazer dividendos para a empresa. No capítulo 3 desta dissertação apresentaram-se três modelos de negócio indirectos, que permitam viabilizar economicamente o software livre, a saber: venda de serviços adicionais, venda

de software adicional e venda de hardware adicional. A empresa pode optar por um destes modelos, em simultâneo ou não, mas a escolha deste modelo encontra-se limitado, e deve estar alinhado, face às competências centrais da empresa (Kooths e Langenfurth, 2003).

Um outro aspecto que só se aplica ao software livre diz respeito à escolha do modelo de licenciamento desse software. Esta escolha é fortemente condicionada pelo mercado de soluções no âmbito do software livre, uma vez que certas licenças exigem um certo conjunto de restrições na sua aplicabilidade (ex: GPL). Existem diversas licenças que são incompatíveis com outras (ver capítulo 4), não podendo, por isso, ser usadas conjuntamente. Para além disso, devem-se salvaguardar as situações em que se pretende usar simultaneamente software livre e software proprietário. Estas situações são perfeitamente viáveis, desde que se tenha em atenção o tipo de licença que se pretende usar e também as características de cada um dos módulos que compõem o software livre. Por exemplo, são comuns situações em que se torna necessário licenciar determinados módulos de software livre, para que possam ser usados legalmente nas soluções proprietárias (ex: caso da Critical Software).

Um correcto posicionamento na cadeia de valor assume também uma relevância fulcral. A empresa deve estar preparada para competir com as demais existentes no mercado e deve possuir um conjunto de vantagens face a esses concorrentes. A empresa deve optar por um posicionamento que vá de encontro às suas características e objectivos, podendo seguir uma estratégica de liderança pelo custo, diferenciação e focalização ou até mesmo optando intencionalmente não ser líder em nenhum desses segmentos. Contudo, e perante os casos estudados, pode-se estabelecer que uma das estratégias que garante à partida maiores possibilidades de sucesso passa por apostar na focalização num nicho de mercado num âmbito geográfico alargado, ou seja, assumir à partida um forte comprometimento com a internacionalização.

Qualquer que seja a estratégia adoptada pela empresa, esta tem que ter uma grande capacidade de resposta a alterações da sua envolvente externa. Para tal, esta deve assegurar recursos ao nível do tratamento da informação e capacidade interna ao nível da sua flexibilidade. Esta informação e flexibilidade são essenciais para se conhecer e acompanhar as alterações do mercado (Ramachandran, 2004).

Um outro aspecto que normalmente se encontra associado às empresas que comercializam software livre é o seu forte comprometimento com a inovação tecnológica. As soluções de software livre potenciam o aparecimento de inovações do ponto de vista tecnológico, apesar de não serem facilmente sustentáveis ao longo do tempo (Kooths e Langenfurth, 2003). Contudo, é sempre possível para uma empresa ser pioneira na oferta de um determinado produto e personalizá-lo de acordo com a área de actuação da empresa e do próprio cliente. Neste sentido, uma empresa que adopta este tipo de estratégia deve ter fortes laços de relação com a restante comunidade *open source* e deve ser um colaborador activo desta mesma comunidade.

Em termos técnicos e ao nível da concepção de soluções, uma estratégia de desenvolvimento de soluções *open source* assenta sobre dois pilares: personalização das soluções e qualidade das soluções.

As soluções *open source* devem poder ser facilmente personalizadas para o cliente e este cliente deve ter um elevado grau de liberdade no estabelecimento dos requisitos do projecto. O produto deve ser construído de forma a ter em conta a facilidade de introdução de novas alterações requisitadas pelo cliente. Uma nova funcionalidade exigida pelo cliente deve acarretar um esforço mínimo para a empresa. No mesmo sentido, a escalabilidade do produto é também um aspecto de grande importância. A facilidade de personalização não deve ser unicamente atendida na fase de concepção e instalação da solução no cliente, mas deve ser assegurada ao longo de todo o ciclo de vida de um produto, que contempla também a sua fase de manutenção no cliente.

A qualidade das soluções é um dos aspectos que merece um especial destaque. Usualmente, as soluções *open source* são ainda encaradas por algumas empresas com alguma desconfiança, nomeadamente em termos de estabilidade e na existência de "bugs" (sintomas de mau funcionamento). As soluções *open source* podem ser tão estáveis e robustas como as soluções proprietárias, mas para que tal aconteça é necessário que exista internamente uma equipa técnica preparada para lidar com estas situações. Se à partida já existir um grande grau de comprometimento da empresa sobre a solução de software livre que se encontra a desenvolver, todo este processo é facilitado e existirá um grande domínio de conhecimento por parte da empresa da solução que pretende disponibilizar. Sugere-se também que exista um processo de teste e correcção de eventuais módulos que, por questões de prazos, ficaram inacabados ou que não se adaptaram totalmente às necessidades previsíveis dos clientes para o futuro.

Por último, um outro facto a considerar diz respeito à qualificação e formação dos recursos humanos. As soluções no âmbito de software livre exigem por parte dos seus colaboradores uma grande flexibilidade e capacidade de aprendizagem. Paralelamente, os recursos humanos, nomeadamente a equipa de gestão de uma empresa, devem ter bons conhecimentos de gestão e não apenas capacidades técnicas em termos de desenvolvimento de software, o que, em algumas situações, acontece nomeadamente em "start-ups" tecnológicas. A empresa deve estar preparada em termos de recursos humanos para atender a todos os desafios que lhe são colocados em termos técnicos e comerciais.

8.2 Trabalho Futuro

No âmbito desta dissertação de mestrado não foi analisada a problemática da viabilidade e as mais valias que pode trazer a utilização do software livre numa sociedade cada vez mais dependente da informática. Assim, não foi estudada a problemática da utilização do software livre como forma de desenvolver, no quadro da nossa sociedade capitalista, uma sociedade de competição sã e não monopolista, que respeite a legalidade e o bem-estar dos seus cidadãos.

Esta situação seria interessante de se analisar, já que não tem unicamente impacto na sociedade, mas também no seio das empresas privadas e organizações sem fins lucrativos. Actualmente assiste-se cada vez mais a uma preocupação por parte do meio empresarial na gestão da sua imagem perante a sociedade, que não se esgota unicamente nos seus proveitos financeiros. Aliás, a procura desses proveitos financeiros são apenas uma vertente, porventura a mais visível, da missão da organização.

Outro trabalho futuro interessante passaria pela realização de um estudo que avaliasse a actual dimensão da penetração da tecnologia *open source* no tecido empresarial do país.

9 Bibliografia e Referências

Acohido, B. (2003), "Linux took on Microsoft, and won big in Munich", *USA Today*, publicado na edição 13/7/2003.

Amadeu, S. (2005), "Valor Económico, Empresas & Tecnologia", *Instituto Nacional de Tecnologia da Informação do Brasil*: 1-3.

Almeida, I. (2006), "Software livre não é software gratuito", no sítio web http://www.itmaster.com.br, acedido em 11/03/2006.

Backhaus, K (1999), "Happy Engineering", *Manager Magazine*, vol8: 130-133.

Bonaccorsi, A. e Rossi, C. (2003), "Why Open Source can succeed", *Research Policy*, vol. 32: 1243-1258.

Cohan, P. (1999), "Net Profit: How to invest and compete in the real world of internet business", *Jossey-Bass*, 1ª edição.

DiBona, C , Raymond, E. , Stallman, R. e outros (1999), "Voices from the Open Source Revolution", *O'Reilly*, 1ª edição.

El-Eman, K. (2001), "Ethics and Open Source", *Kluwer Academic Publishers*, vol6: 291-292.

Feller, J. , Fitzgerald, B. , Hissam, S. e Lakhani, K. (2005), "Perspectives on Free and Open Source Software", MIT Press, 1ª edição.

Ferraz, N. (2002), "Vantagens Estratégicas do Software Livre para o Ambiente Corporativo", *Monografia apresentada para a conclusão do curso Master Business Information Systems*.

Frost, J. , Leffler, K. , Gomulkiewicz, R. e Laster, D. (2005), "Some Economic & Legal Aspects of Open Source Software", *Department of Economics*, University of Washington.

Gacek, C. , Lawrie, T. e Arief, B. (2005), "The many meanings of Open Source", *Department of Computing Science*, University of Newcastle.

Gonçalves, J. (2003), "Outsourcing Estratégico e Desempenho Organizacional", *Semanário Económico*, nº863.

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

Hitt, Michael (2003), "Strategic management: competitiveness and globalization", McGraw-Hill, 5ª edição.

Horowitz, M. (2004), "Linux vs. Windows: A comparison of Linux and Windows", no sítio web http://www.michaelhorowitz.com, acedido em 17/12/2005.

Karaganis, J. e Latham, R. (2005), "The politics of Open Source adoption", *Social Science Research Council*, version 1.0.

Kenwood, C. (2001), "A Business Case Study of Open Source Software", *Washington C3 Center Bedford*, Massachusetts.

Klincewicz, K. (2005), "Innovativeness of open source software projects", *School of Innovation Management*, Tokyo Institute of Technology.

Koening, J. (2004), "Seven open source business strategies for competitive advantage", *IT Manager's Journal*, vol.1: 1-8.

Kooths, S. e Langenfurth, M. (2003), "Open-Source Software: An Economic Assessment", *MICE Economic Research Studies*, Vol.4: 41:48.

Kotler, P. (1999), "Marketing Management", *Pearsons US Imports & Phipes*, Millenium Edition.

Shankland, S. (2000), "Linux sales surge past competitors", CNET news, publicado em 9 de Fevereiro de 2000.

Larsen, K. , McInerney, C. , Nyquist, C , Santos, A. e Silsbee, D. (1996), "Learning Organizations", no sítio web http://home.nycap.rr.com/klarsen/learnorg/ acedido em 6/04/2006.

Laurent, A. (2004), "Understanding Open Source and Free Software Licensing", *O'Reilly*, 1ª edição.

Lerner, J. e Tirole, J. (2000), "The simple economics of Open Source", *Harvard Business School California Research Center*, Paperback version.

Mintzberg, H. e Quinn, J. (1996), "The strategy process", Prentice Hall, 3ª edição.

Moody, G. (2001), "Rebel Code", *The Penguin Press*, 2ª edição.

Morris, Peter. (2004), "Is Japan losing its tech edge to East Asian rivals", no sítio web http://www.atimes.com, acedido em 26/03/2006.

Murrain, M. (2004), "Choosing and Using Open Source Software: A primer for Nonprofits", *Nonprofit Software Initiative*.

Neus, A. e Scherf, P. (2005), "Opening Minds: Cultural change with the introduction of open-source collaboration methods", *IBM Systems Journal*, Vol. 44, No. 2: 215-226.

Patricio, L. (2005), "Apontamentos de Fundamentos de Gestão", FEUP.

Porter, M. (1996), "Competitive Advantage: Creating and Sustaining Superior Perfomance", *Free Press*, 3ª edição.

Ramachandran, S. (2004), "Strategy in a competitive landscape", CEO refresher publications.

Raymond, Eric S. (1999), "The Cathedral and the Bazaar", *O'Reilly*, 2ª edição.

Schimidt, K. e Schnitzer, M. (2002), "Public Subsidies for Open Source: Some Economic Policy Issues of the Software Market", *CEPR*, University of Munich.

Shaw, R. (2005), "The TCO of Open Source", no sítio web http://www.cio-today.com, acedido em 3/04/2006.

Silveira, S. (2005), "Software Livre e Exclusão Digital", *Jornal da AEUDF*, Vol.3: 5-8.

Tapscott, D. (1997), "The Digital Economy", *MacGraw-Hill*, 1ª edição.

Tidd, J. , Bessant, J e Pavitt, K. (2005), "Managing Innovation: Integrating Technological, Market and Organization Change", *Wiley*, 3ª edição.

Vaden, T. (2003), "Intellectual Property, Open Source and Free Software", *University of Tampere*: 1:5.

Valimaki, M. e Oksanen, V. (2003), "Evaluation of Open Source Licensing Models for a Company Developing Mass Market Software", *Helsinki Institute for Information Technology*, Paperback version.

Walker, T. (2004), "The Future of Open Source in Government", *Government Open Source Applications*: 1-5.

Weber, S. (2000), "The Political Economy of Open Source Software", BRIE *Working Paper Series*, No. 140.

Weiss, T. (2001), "Open-source collaboration web site shuts down", ComputerWorld, publicado em 9 de Abril de 2001.

Apontadores:

http://www.criticalsoftware.com – sítio web da empresa Critical Software, acedido em 7/01/2006.

http://www.cyclades.com – sítio web da empresa Cyclades, acedido em 10/12/2005.

http://www.edgebox.net – sítio web do projecto edgebox da Critical Software, acedido em 18/01/2006.

http://www.fsf.org – sítio web da Free Software Foundation, acedido em 23/11/2005.

http://www.marketingteacher.com – sítio web que providencia um conjunto de informação na área de marketing, acedido em 16/10/2005.

http://www.microsoft.com – sítio web da maior empresa mundial fornecedora de software proprietário, acedido em 19/11/2005.

http://www.opensource.org – sítio web da Open Source Initiative, acedido em 15/11/2005.

http://www.opensourcexperts.com – sítio web do projecto opensourcexperts, acedido em 15/12/2005.

http://www.pmelink.pt – sítio web de gestão de negócios online destinado aos gestores de pequenas e média empresas, acedido em 18/11/2005.

http://www.oss-watch.ac.uk – sítio web que contém diversas referências sobre os principais projectos "open source" e respectivos *standards*, acedido em 4/03/2006.

http://www.softwarelivre.citiap.gov.pt/ - sítio web do software livre do Estado Português, acedido em 14/11/2005.

http://sourceforge.net - sítio web do repositório de software livre em termos de código e aplicações, acedido em 11/04/2006.

http://www.wildopensource.com – sítio web da empresa Wild Open Source, acedido em 22/12/2005.

ANEXOS

Anexo A: Licença BSD

Copyright 1994-2006 The FreeBSD Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE FREEBSD PROJECT ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

Anexo B: Licença GPL

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on

the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Anexo C: Licença LGPL

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into

another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to

exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library

and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and an idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

Anexo D: Licença MPL

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. ''Contributor'' means each entity that creates or contributes to the creation of Modifications.

1.2. ''Contributor Version'' means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. ''Covered Code'' means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. ''Electronic Distribution Mechanism'' means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. ''Executable'' means Covered Code in any form other than Source Code.

1.6. ''Initial Developer'' means the individual or entity identified as the Initial Developer in the Source Code notice required by **Exhibit A**.

1.7. ''Larger Work'' means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. ''License'' means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. ''Modifications'' means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. ''Original Code'' means Source Code of computer software code which is described in the Source Code notice required by **Exhibit A** as Original Code, and which, at the time of its release

under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. ''Source Code'' means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You'' (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You'' includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control'' means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section **2.2**. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section **6.1**, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section **3.5**.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available

via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims.
If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL'' which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.
If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.
Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.
You must duplicate the notice in **Exhibit A** in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a

Contributor to the notice described in **Exhibit A**. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section **3.1-3.5** have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section **3.2**. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.
If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section **3.4** and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.
This License applies to code to which the Initial Developer has attached the notice in **Exhibit A** and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation (''Netscape'') may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases ''Mozilla'', ''MOZILLAPL'', ''MOZPL'', ''Netscape'', "MPL", ''NPL'' or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in **Exhibit A** shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.
COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS'' BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.
8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually

agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a ''commercial item,'' as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of ''commercial computer software'' and ''commercial computer software documentation,'' as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent

necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as Multiple-Licensed. Multiple-Licensed means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the MPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

Anexo E: Lista dos 20 projectos mais activos no *SourceForge* em Abril de 2006

		Contribuições					
Nome do Projecto	**Licença**	**Bugs**	**Features Request**	**Patches**	**Fóruns**	**Public Messages**	**Descrição**
Asterisk@Home	GPL	196	106	13	2	24687	The Asterisk@Home project enables the home user to quickly set up an Asterisk PBX. A web GUI makes configuration and operation easy. We also provide an xPL (home automation) interface for easy interaction with other devices in the home.
Gaim	GPL	11963	3520	2590	1	16905	Gaim is a GTK+ instant messaging application. It supports multiple protocols, including AIM, ICQ, Yahoo!, MSN, Jabber, IRC, Napster, Gadu-Gadu, Zephyr, and SILC. It has many common features found in other clients, as well as many unique features
Azureus	GPL	2316	N/D	N/D	4	38393	Azureus is a powerful, full-featured, cross-platform java BitTorrent client.
PDFCreator	GPL	569	191	29	2	6473	PDFCreator easily creates PDFs from any Windows program. Use it like a printer in Word, StarCalc or any other Windows application.
phpMyAdmin	GPL	2124	919	301	3	20621	phpMyAdmin is a tool written in PHP intended to handle the administration of MySQL over the Web. Currently it can create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage keys on fields.
7-Zip	LGPL	557	505	60	2	16414	7-Zip is a file archiver with the high compression ratio. The program supports 7z, ZIP, CAB, RAR, ARJ, LZH, CHM, GZIP, BZIP2, Z, TAR, CPIO, RPM and DEB formats. Compression ratio in the new 7z format is 30-50% better than ratio in ZIP format.
Guliverkli	GPL	608	382	5	2	5173	Home of VobSub, Media Player Classic (MPC) and other misc utils.
Zk – Ajax	GPL	74	95	0	4	1779	ZK is Ajax Java Web framework enabling rich UI for Web

							apps with no JavaScript and little programming. With event-driven feature-rich components, developing as simple as programming desktops. With markup language, designing as simple as authoring HTML.
PhpGedView	GPL	1002	583	330	3	14064	PhpGedView parses GEDCOM 5.5 genealogy files and displays them on the internet in formats and charts that you are familiar with. It also allows relatives to edit their genealogy online and collaborate together on their research.
FileZilla	GPL	1052	886	46	3	3437	FileZilla is a fast FTP and SFTP client for Windows with a lot of features. FileZilla Server is a reliable FTP server.
WebCalendar	GPL	1556	454	436	2	20252	WebCalendar is a PHP application used to maintain a calendar for a single user or an intranet group of users. It can also be configured as an event calendar.
OpenWFE	BSD	436	34	59	8	6330	OpenWFE is an open source java workflow engine / environment. It is a complete Business Process Management suite, with 4 components : an engine, a worklist, a webclient and an 'apre' (Automatic Participant Runtime Environment).
JasperReports	LGPL	307	201	133	2	21667	Free Java reporting library. XML report templates are used to generate ready to print documents using data from customizable data sources, including JDBC. The output can be delivered to the screen, printer, or stored in PDF, HTML, XLS, CSV and XML format
KeePass	GPL	381	694	10	2	4753	KeePass Password Safe is a free, open-source, light-weight and easy-to-use password manager for Windows. You can store your passwords in a highly-encrypted database, which is locked with one master key or key-disk.
Compiere ERP	MPL	2244	589	66	15	23667	Smart ERP+CRM solution for Small-Medium Enterprises in the global market covering all areas from order and customer/supplier management, supply chain to accounting. For $5-500M revenue companies looking for "brick and click" first tier functionality.
iReport-Designer	GPL	434	137	43	2	8588	iReport is a visual reporting tool based on JasperReports written in 100% pure java. You can manage charts, images, subreports,... Data can be retrived using JDBC, TableModels, JavaBeans, XML,Hibernate, CSV...It

							supports output in PDF,RTF,XML,XLS,CSV,HTM
MinGW	OSI	929	33	295	13	2674	MinGW: import libraries and header files for use with GCC to build native Windows applications; now with added extensions to the MSVC runtime to support C99 functionality.
Gallery	GPL	1763	1915	198	5	N/D	A slick, intuitive web based photo gallery with authenticated users and privileged albums. Easy to install, configure and use. Photo management includes automatic thumbnails, resizing, rotation, etc. User privileges make this great for communities.
PHP For Applications	GPL	67	73	N/D	5	1082	P4A (PHP For Applications) is a PHP RAD and object oriented PHP framework for building event-driven stateful web applications. Features tableless HTML, accesskey, point&click app generator, auto data type recognition, UTF-8, i18n/l10n, PEAR integration.
FCKeditor	LGPL	1253	338	244	3	11796	Online text editor (DHTML editor), for ASP, ASP.NET, ColdFusion, PHP, Java and JavaScript brings to the web many of the powerful functionalities of known desktop editors like Word. It's XHTML compliant and works with Firefox, Mozilla, Netscape and IE.

Anexo F: Entrevista da Critical Software

Data: 20 de Fevereiro de 2006
Destinatário: Engº André Barbosa (colaborador no projecto *edgebox*)

1- De quem é a responsabilidade da instalação e configuração da edgebox? Existem serviços de suporte disponibilizados pela Critical Software?
A venda da edgeBOX é feita ao cliente por parceiros escolhidos pela Critical Software. Todas as instalações são feitas por esses parceiros estratégicos. Existe, por exemplo, um parceiro em UK, outro no Brasil no Chile, em Portugal e de certeza mais alguns que não tenho conhecimento. Aqui em Coimbra temos uma equipa de suporte que é responsável por dar formação às equipas de suporte dos parceiros.
Numa primeira fase a formação passa por ensinar como instalar e utilizar uma edgeBOX. Explicar cenários reais de utilização e resolução de alguns problemas. As próprias empresas "parceiras" tem equipas de suporte que são as responsáveis por responder a dúvidas mais básicas que os clientes finais tenham.
Quando surge uma dúvida mais técnica esta pode ser encaminhada para a equipa de suporte da Critical Software, e se mesmo assim for necessário pode ser encaminhada para a equipa de desenvolvimento.
Para dar um caso concreto de como isto funciona. Eu próprio numa primeira fase tive de dar formação a vários colegas aqui da Critical sobre a secção VoIP da edgeBOX que é da minha responsabilidade.
Também fiz apresentações e dei formação a alguns parceiros como por exemplo a VCOMM (parceiro em UK). http://www.vcomm.co.uk/
No momento esta formação é feita por colegas da área de suporte que tem reuniões comigo quando são disponibilizadas novas *features*.

2- Qual a vossa política de distribuição da edgebox? Têm distribuidores e revendedores ou fazem a venda directa através da vossa empresa?
Como respondi em cima a Critical Software não faz venda directa da edgeBOX. Não é a sua área de negócio e não pretende adquirir *know how* nesse ramo. Tem portanto uma rede de parceiros que tratam da venda, instalação das caixas e dão um primeiro nível de suporte aos clientes.

3- Quem são os vossos principias clientes da edgebox? Já têm alguns casos de sucesso?
A edgeBOX começou por ser desenvolvida com um cliente alvo que é a Soporcel. Neste momento eles contam com várias edgeBOXs em cada um dos seus escritórios.
Quanto à versão actual 4.0 está neste momento a ser disponibilizada por vários parceiros, mas quanto a clientes finais não sei quantos existem neste momento.
Sei que por exemplo vão estar algumas escolas 2\3ciclo e secundários equipadas com edgeBOXes para uma primeira fase de *trials* que podem resultar em algo mais concreto se se proporcionar.

4- Qual a vossa política de conquista de um novo mercado internacional? Têm parceiros nesses mercados?
A aposta é claramente em parceiros com conhecimento do mercado em questão.

5- Qual o vosso contributo para a comunidade open source? Fazem algum desenvolvimento interno de soluções OS e partilham estas novas versões com a restante comunidade OS?

O desenvolvimento da edgeBOX é por razões comerciais closed-source. Todas as aplicações são usadas legalmente e são colocadas as respectivas licenças. Fazemos até acordos como no caso da Digium ("proprietária" do Asterisk) onde colocamos uma interface simples para activar licenças do seu codec g.729 (pago) de forma a suportar o projecto deles.

Também no caso da Digium, todo o nosso hardware VoIP é feito por eles e é o hardware certificado e aconselhado para uso na edgeBOX. Falo concretamente deste caso porque me é mais próximo.

Anexo G: Entrevista da PyLabs

Data: 11 de Março de 2006
Destinatário: Engº Hugo Ramos (antigo CEO da PyLabs)

1- Qual a data de fundação e encerramento da PyLabs?
A empresa foi fundada no início de 2003 e viria a cessar a sua actividade em Janeiro de 2004.

2- Qual a área de negócio da PyLabs?
Concepção de soluções para a web usando software open source. O objectivo era de colmatar uma falha na produção de tecnologias web em Portugal viradas para o open source.

3- Qual o vosso "background" em termos científicos e profissionais?
Os fundadores da empresa eram oriundos da Oniway. Já tínhamos alguma experiência de trabalho em conjunto e juntamos a essa experiência mais três estagiários da Universidade Técnica de Lisboa na área das tecnologias da informação.

4- Apresente, como exemplo, uma arquitectura típica de uma solução desenvolvida pela PyLabs?
As tecnologias que usávamos em cada solução dependiam das necessidades e especificações do cliente. Usávamos essencialmente tecnologias na área de open source, e grande parte das nossas soluções seguiam uma arquitectura tecnológica com base no Linux, Python, Zope e MySQL.

5- Quais foram os vossos principais clientes?
Os nossos principais clientes foram o Estado Português e também algumas PME's na área metropolitana de Lisboa.

6- Quem foram os vossos principais concorrentes?
Existem grandes concorrentes na área de desenvolvimento de software, nomeadamente os que usam soluções comerciais com base no Java e Microsoft. Um dos nossos principais concorrentes foi a Accenture.

7- Quais eram as maiores valias apresentadas pela vossa empresa?
Para além do preço mais baixo das nossas soluções, a personalização das soluções era um dos nossos pontos fortes, que se aliava também à grande flexibilidade da nossa empresa que nos permitia ir de encontro às necessidades específicas de cada cliente e diminuir o ciclo de desenvolvimento de uma solução.

8- Quais os vossos principais obstáculos deparados no decurso da existência da empresa?
A mentalidade dos clientes quando se deparam com soluções open source. A este nível, este tipo de mentalidade ainda se encontra na idade média.
Um outro obstáculo diz respeito aos concursos públicos que eram ganhos de forma alternada por grandes empresas. Não foi possível fazer frente aos grandes lobby's e compadrios das grandes empresas.

9- Quais as principais razões para o encerramento da empresa?

Quando fechámos a empresa, esta não estava a dar prejuízo, mas o seu lucro era bastante baixo. Não estávamos a tirar os dividendos esperados que se apresentavam nos planos de negócio, daí que resolvemos fechar a empresa e encarar novos desafios profissionais.

Anexo H: Contrato de serviços de suporte técnico da Cyclades

Os contratos PSA (Priority Support Agreement) têm por objectivo proporcionar novas opções de serviços contratuais aos clientes que já possuem ou, que estão adquirindo um produto Cyclades.

Principais características do PSA:

- Telefone exclusivo para abertura de chamados de software para esclarecimentos de dúvidas quanto à configuração e instalação de nossos produtos, com tempo de reposta muito reduzido em relação ao oferecido na garantia padrão.

- Suporte de hardware com opção de atendimento nas instalações do cliente e ainda, garantia de tempo de solução através da troca do equipamento.

- Se optar por continuar com suporte de hardware em balcão, oferecemos também tempo de reparo reduzido em relação ao oferecido na garantia padrão.

Benefícios para a sua empresa ao adquirir o contrato PSA:

- Redução dos tempos de "parada" ocasionados por problemas de configuração incorrecta ou necessidade de manutenção de hardware ou software;

- Atendimento prioritário para clientes com contrato de suporte;

- Tempos reduzidos para atendimento e resolução de problemas;

- Não há necessidade de aguardar ou aprovar orçamentos;

- Redução de investimento em equipamentos sobressalentes;

- Minimiza risco de perdas de produtividade e financeiras;

- Pacotes de serviços flexíveis.

2. NÍVEIS DE SERVIÇO

A Cyclades oferece vários níveis de serviços para melhor atender as suas necessidades:

PSA PLATINUM

Suporte de software por telefone, disponível 24 x 7 x 365, com tempo de resposta de até 2 (duas) horas para até 3 (três) chamados por mês por equipamento contratado.

Suporte de hardware com atendimento nas instalações do cliente, disponível 24 x 7 x 365, com de tempo de solução de até 6 (seis) horas através da troca do equipamento.

PSA GOLD

Suporte de software por telefone, disponível em horário comercial, com tempo de resposta de até 2 (duas) horas úteis para até 3 (três) chamados por mês por equipamento contratado.

Suporte de hardware com atendimento nas instalações do cliente, disponível em horário comercial, com de tempo de solução de até 1 (um) dia útil através da troca do equipamento.

PSA SILVER

Suporte de software por telefone, disponível em horário comercial, com tempo de resposta de até 2 (duas) horas úteis para até 3 (três) chamados por mês por equipamento contratado.

Suporte de hardware com atendimento em balcão, disponível em horário comercial, com tempo de solução de 1 dia útil, contados a partir da entrada do produto na Cyclades.

PSA BRONZE

Suporte de software por telefone, disponível em horário comercial, com tempo de resposta de até 2 (duas) horas úteis para até 3 (três) chamados por mês por equipamento contratado.

Suporte de hardware com atendimento em balcão, disponível em horário comercial, com tempo de solução de 5 dias úteis, contados a partir da entrada do produto na Cyclades.

Obs.:

- Nas modalidades SILVER e BRONZE, eventuais despesas com envio e devolução de equipamentos, serão de responsabilidade do cliente.

- As modalidades PLATINUM, GOLD e SILVER estão disponíveis somente para equipamentos em linha de produção.

- Para equipamentos fora de linha, está disponível exclusivamente o contrato BRONZE.

- Os contratos não cobrem defeitos causados por mau-uso ou perda total.

Anexo I: “Technical data sheet” da *edgebox*

Software Features

Security

Stateful Inspection Firewall

- Filter on source and/or destination IP address/port value
- Filter on SYN, ACK flags and ICMP
- Stateful inspection
- Provides enterprise-grade firewall protection
- Full-time Stateful Packet Inspection with built-in support for most popular applications
- No predefined limit on the number of rules that can be created and applied
- All firewall messages can be logged to syslog servers
- AAA, Authentication, authorisation, and accounting (external and internal)
- Group based access control

Virtual Private Networking

- IPSec, no pre-defined limit on VPN tunnels
- Internet Key Exchange (IKE) including Aggressive Mode
- 3DES (168-bit) encryption
- Supports Perfect Forward Secrecy (DH Group 2 and 5)
- Provides protection from replay attacks
- IPSec VPN Firewall to control tunnel access
- PPTP Server with local or remote authentication

User Authentication

- Local, RADIUS server, or LDAP server (e.g. W2K)

Configuration and Management

Configuration

- Easy setup through a browser-based user interface
- Wizard based setup
- Default basic configuration
- Configuration and management using HTTP, serial console and SSH
- Syslog server support

Secure Management

- Local and remote management
- RADIUS management authentication support
- RADIUS support for accounting (time & traffic)
- SSH secure management channel
- SNMP support

Software Update Management

- Automatic update from central update server (update.edgebox.net)
- Manual configuration option which downloads available updates and waits for user confirmation before install
- Notifications for service shutdown and service restart for planned maintenance
- Update service access based on yearly subscription

Anti-piracy protection

- Proprietary software is encrypted (AES 256 bit)
- Software key lock tied to hardware signature to deter piracy, renders edgeBOX software useless in other hardware
- License key controls access to update server on update.edgebox.net

IP Quality of Service (IP QoS)

- DiffServ traffic prioritisation (ToS byte marking)
- Inbound and Outbound QoS
- CBQ traffic prioritisation
- Configurable traffic prioritisation policies by: addresses, users, groups, port, protocol

High Availability

- Data backup to remote FPT site or USB disk
- Automatic, configurable regular backups
- Manual restore of backed up data from restore list
- Failover *
- Dual-homing*

* To be released in Q1 2006

Networking

- PPP over Ethernet and PPP over ATM
- Automatic IP and DNS assignment
- Static routing on the LAN and/or WAN
- DHCP server and client
- DNS Server (Master and Forward)
- Dynamic DNS client
- Network Address Translation (NAT)
- Port Forwarding
- Supports public Web and e-mail servers with NAT

Collaboration Services

- HTTP Server with user home pages
- FTP Server
- Email server supporting: POP3, IMAP, SMTP
- Web mail
- LDAP Intranet Mail routing
- Email remote storage
- Email aliases management
- Email antivirus (interfaces with Sophos™)
- Email antispam (based on RBLs)
- SMTP relay for road warriors
- Print Server (for USB printers)
- Windows file sharing (SME)
- Web Transparent Caching
- User Disk Quotas

VoIP and IP PBX

- SIP and IAX support
- Analogue and Digital PSTN gateway
- Least Cost Routing
- Interactive Voice Response (IVR)
- Automatic Call Distribution (ACD)
- Telephone Number Mapping (ENUM)
- Voice Mail with optional email forwarding of voice messages
- Call routing between remote edgeBOX switches (branch offices)
- Digital Receptionist
- Incoming Call Rules
- DID Call Routing
- Sound Manager
- Follow-me feature for user extensions
- Call Parking
- Call Forwarding
- Conference Bridging
- Music on Hold
- Call Queuing
- Hunt Groups
- User Profiles with call permissions
- Support for all common codecs

Hardware Features

edgeBOX runs on standard hardware and supports a range of hardware features listed below. For support of specific hardware configurations please contact us.

WAN Interface

- Cable Modems
- ADSL modems (internal or USB)
- UMTS
- Ethernet-based modems

LAN Interface

- Primary Ethernet
- Secondary Ethernet for DMZ or bridging
- WiFi (802.11b/g) AP mode

EWAN Interface

- Ethernet port for Enterprise WAN

Serial Interface

- One asynchronous serial console port

Telephony

For interconnection with digital and analog telephony equipment, edgeBOX supports and auto-detects a number of hardware devices from various manufacturers including BRI boards (1, 4 and 8 ports), PRI (up to 124 channels) and POTS 2+2 boards from Digium and Beronet.
No additional hardware is needed for VoIP.

Open Source Software

The edgeBOX software distribution includes a number of open-source packages that are distributed according to their specific licenses, including: Linux 2.6 kernel, Openldap, Apache, php, sendmail, OpenSSL, bind, dhcpd, OpenSwan, Squid, Samba, Iptables, Iproute2, Asterisk, qpopper, imapd, postgreSQL, tomcat, proftpd, net-snmp, freeradius, OpenSSH, mrtg, mailscanner.

Critical Software is an ISO 9001:200
Tick-IT certified company