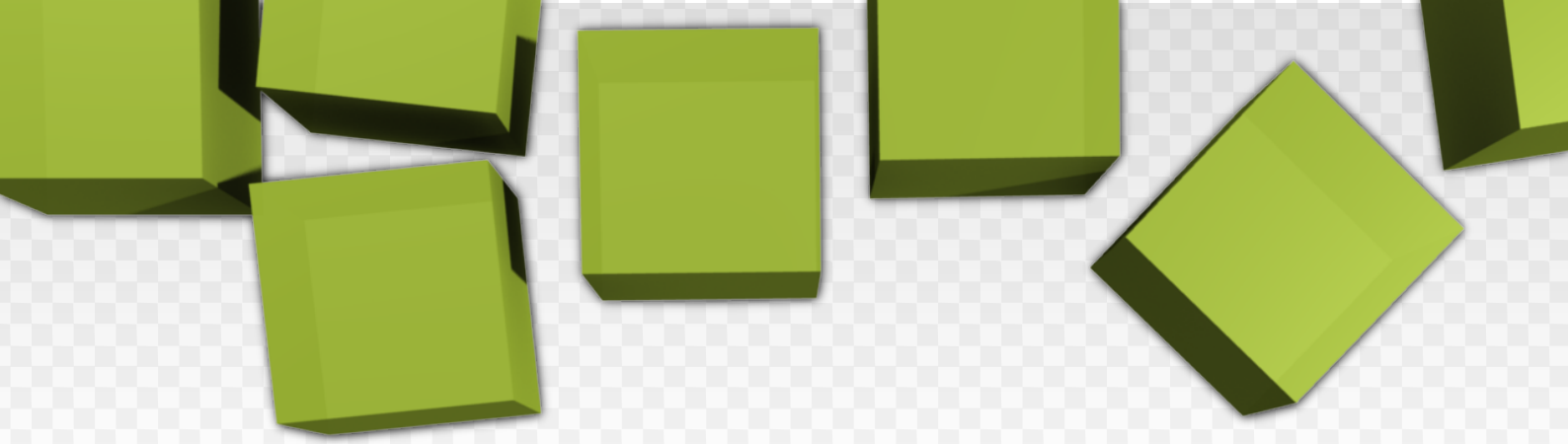


IMPLEMENTATION OF FREE SOFTWARE SYSTEMS

AUTHOR:
A. ALBÓS RAYA
M. D'ELIA BRANCO
M. LEÓN MARTÍNEZ
A. NOVO LÓPEZ
A. OTERO GARCÍA
Ó. SÁNCHEZ JIMÉNEZ

COORDINATOR:
D. MESÍAS JIMÉNEZ

FREE TECHNOLOGY ACADEMY

3

Implementation of free software systems

David Megías Jiménez (coordinator)
Amadeu Albós Raya
Marcelo D'Elia Branco
Mónica León Martínez
Alejandro Novo López
Alberto Otero García
Óscar David Sánchez Jiménez

PID_00148363

UOC
Universitat Oberta de Catalunya
www.uoc.edu

David Megías Jiménez

Degree in IT engineering from the UAB (Autonomous University of Barcelona). Master's degree in advanced process automation techniques from the UAB. Doctorate in IT from the UAB. Lecturer in IT and Multimedia Studies at the UOC.

Amadeu Albós Raya

Degree in IT engineering from the Universitat O de Catalunya (Open University of Catalonia UOC) and the University of Andorra with Security networks and distri systems as his specialist su He currently teaches coordinates and administrates IT systems at an educational institution is a mem of the Grupo de Investigación en Seguridad Informática y Software Li (IT Security and Free Software Research Group) of the University of Andorra and acts as a consultant to the Master's Degree in Free Software of the Universitat O de Catalunya.

Marcelo D'Elia Branco

Marcelo has worked in information technology for the last 25 years. He the technical manager of PROCERGS (an IT company in the state of Rio Grande do Sul) in 1999 quickly taking over the position of vice-chairman. In 2002 he coordinated the IT Division of the UERGS (Río Grande do Sul State University). Between 2003 and 2004 he acted as special projects advisor in the prefecture of Porto Alegre. He now works as a free software technology consultant and is organiser of the Projeto Software livre Brasil coordinating mem of the Projeto Software livre RS and a mem of the Science Board of the International Master's Degree in Free Software of the UOC.

Mónica León Martínez

Mónica holds a diploma in li sciences from the University of Granada and a degree in documentation from the University of Extremadura. Since 1998 she has worked on actions to promote and implement information technologies in Extremadura. She is currently the project technician of the Fomento de Nuevas Iniciativas (Promotion of New Initiatives) centre Programa de la Consejería de Infraestructuras y Desarrollo Tecnológico (Programme for the Department of Infrastructures and Technological Development) and FUNDECYT (Foundation for the Development of Science and Technology in Extremadura) which acts as a centre for gnuLinEx support and development and as a regional o of the information society.

Alejandro Novo López

Degree in IT engineering specialising in software engineering from the Pontificia University of Salamanca. Junior Degree in computer science from the University of Houston. He has pu two on the Unix environment: *Entorno UNIX y Protocolos de Internet* and *Diseño e implementación en sistemas UNIX*. He is currently in charge of the cross-sector development of emerging technologies at Sun Microsystems I

Alberto Otero García

Degree in IT engineering from the Ramon Llull University. Founding partner and project manager of Cometa Technologies a company that provides IT solutions on the use of open source standards and tools. Tenured lecturer of Operating Systems Administration in the La Salle School of Engineering and Architecture and consultant to the International Master's Degree in Free Software of the UOC.

David Jiménez

Degree in telecommunications engineering from the Polytechnic University of Valencia. He is currently R&D project manager at the Institut National de Recherche en Informatique et Automatique in Rennes (France) and has participated in many international research projects on free software. He colla with the Universitat O de Catalunya as consultant to the Master's Degree in Free Software.

First edition: February 2010

Av. Tibidabo, 39-43, 08035 Barcelona
Design: Manel Andreu
Publishing: Eureca Media, S

Preface

Software has become a strategic societal resource in the last few decades. The emergence of Free Software, which has entered in major sectors of the ICT market, is drastically changing the economics of software development and usage. Free Software – sometimes also referred to as "Open Source" or "Libre Software" – can be used, studied, copied, modified and distributed freely. It offers the freedom to learn and to teach without engaging in dependencies on any single technology provider. These freedoms are considered a fundamental precondition for sustainable development and an inclusive information society.

Although there is a growing interest in free technologies (Free Software and Open Standards), still a limited number of people have sufficient knowledge and expertise in these fields. The FTA attempts to respond to this demand.

Introduction to the FTA
The Free Technology Academy (FTA) is a joint initiative from several educational institutes in various countries. It aims to contribute to a society that permits all users to study, participate and build upon existing knowledge without restrictions.

What does the FTA offer?
The Academy offers an online master level programme with course modules about Free Technologies. Learners can choose to enrol in an individual course or register for the whole programme. Tuition takes place online in the FTA virtual campus and is performed by teaching staff from the partner universities. Credits obtained in the FTA programme are recognised by these universities.

Who is behind the FTA?
The FTA was initiated in 2008 supported by the Life Long Learning Programme (LLP) of the European Commission, under the coordination of the Free Knowledge Institute and in partnership with three european universities: Open Universiteit Nederland (The Netherlands), Universitat Oberta de Catalunya (Spain) and University of Agder (Norway).

For who is the FTA?
The Free Technology Academy is specially oriented to IT professionals, educators, students and decision makers.

What about the licensing?
All learning materials used in and developed by the FTA are Open Educational Resources, published under copyleft free licenses that allow them to be freely used, modified and redistributed. Similarly, the software used in the FTA virtual campus is Free Software and is built upon an Open Standards framework.

Evolution of this book
The FTA has reused existing course materials from the Universitat Oberta de Catalunya and that had been developed together with LibreSoft staff from the Universidad Rey Juan Carlos. In 2008 this book was translated into English with the help of the SELF (Science, Education and Learning in Freedom) Project, supported by the European Commission's Sixth Framework Programme. In 2009, this material has been improved by the Free Technology Academy. Additionally the FTA has developed a study guide and learning activities which are available for learners enrolled in the FTA Campus.

Participation
Users of FTA learning materials are encouraged to provide feedback and make suggestions for improvement. A specific space for this feedback is set up on the FTA website. These inputs will be taken into account for next versions. Moreover, the FTA welcomes anyone to use and distribute this material as well as to make new versions and translations.

See for specific and updated information about the book, including translations and other formats: *http://ftacademy.org/materials/fsm/*. For more information and enrolment in the FTA online course programme, please visit the Academy's website: *http://ftacademy.org/*.

I sincerely hope this course book helps you in your personal learning process and helps you to help others in theirs. I look forward to see you in the free knowledge and free technology movements!

Happy learning!

Wouter Tebbens
President of the Free Knowledge Institute
Director of the Free technology Academy

The authors would like to thank the Foundation for the Universitat Oberta de Catalunya for financing the first edition of this work, and a large share of the improvements leading to the the second edition, as part of the Master Programme in Free Software offered by the University in question, where it is used as material for one of the subjects.

The translation of this work into English has been made possible with the support from the SELF Project, the SELF Platform, the European Comission's programme on Information Society Technologies and the Universitat Oberta de Catalunya. We would like to thank the translation of the materials into English carried out by lexia:park.

The current version of these materials in English has been extended with the funding of the Free Technology Academy (FTA) project. The FTA project has been funded with support from the European Commission (reference no. 142706-LLP-1-2008-1-NL-ERASMUS-EVC of the Lifelong Learning Programme). This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Acknowledgements

The authors would like to thank the Foundation for the Universitat Oberta de Catalunya (http://www.uoc.edu) for the funding of this project, which forms part of the International Master's Degree in Free Software offered by the University.

They would also like to thank Jordi Mas for his help in coordinating the first edition of this material.

Introduction

In recent years, the technology market has witnessed an increase in its production rates. This development has brought with it a technological revolution in information that can be observed in many areas of everyday life, particularly in society and the economy. A new market constructed on the popularisation of computers and public access to the Internet.

This technological democratisation was spied as a business opportunity by many software producers, which split the original trend towards releasing source code, creating a new branch of proprietary models based on payment for use. Nonetheless, the characteristics of free software philosophy have remained untouched over time: the collaboration of a community located in different parts of the world on the creation, improvement and redistribution of quality, robust, secure and reliable software.

In the last decade, many obstacles have been overcome to extend the democratisation of free software to the home and business environments. Moreover, its academic and teaching application is no longer restricted to a handful of university faculties. In fact, many schools now see free software as a philosophical, economic and operating model suited to their aims.

The success of free software has generated a large market of solutions with very diverse aims. There are now many areas with not one but several products based on free software. This trend is particularly evident in the operating systems market, which, besides offering the usual features for general use, specialises in providing a response to specific hardware features (for example, support to x86 or SPARC architectures) or in implementing specific functions in a given area (for example, support to scientific software or optimised security management). In all events, the primary aim is to maximise adaptation and performance in products that can be implemented directly, thus generating benefits for the end user.

The popularisation of technology and the success of free software have given rise to a new stage with new actors. Technology has become a strategic component in organisations, allowing them to reach their aims efficiently and effectively. The complexity of this stage increases in line with market demands, which perfects free, valid and viable solutions so that they can be implemented directly for any purpose.

In this context, the importance of implementing software with the characteristic rigour and methodology of scientific and technological circles becomes clear.

Nonetheless, this type of implementation should not be viewed from a strictly technological or business angle. We must also take into account the philosophy behind the free software movement, since these systems are normally implemented due to a combination of technological, philosophical, economic and practical reasons.

A methodical execution and management of the systems implementation project is essential if we are to meet the strategic aims of the organisation while also controlling the complexity and risk inherent both to the project and to the implementation environment or context.

The aim of this subject, "Implementation of free software systems", is to provide students with the knowledge required to successfully create, manage and execute projects implementing free software solutions in very diverse organisations.

The materials have been designed to provide a conceptual, methodological and practical response to the inherent complexity of systems implementation, training professionals to implement these systems in a range of contexts and situations with guaranteed success. Nonetheless, we need to remember that any project based on free software must be seen first and foremost as a software project and subsequently as an engineering project, so these materials cannot and should not act as a substitute for the necessary knowledge of these topics.

The subject is split into two teaching modules that gradually introduce the main concepts of systems implementation and free software from two markedly different points of view.

The first module deals with systems implementation from an entirely conceptual and methodological point of view. This module is divided into three units, the first of which introduces systems implementation projects as strategic actions within organisations, provides a rough classification of the different project typologies and details the main features of the functional and operational management of the projects.

The second takes a close look at free software projects from a methodological angle, detailing the main stages and phases of the life cycle of the project that should culminate in the successful implementation of a free software system in an organisation.

The third and final unit presents the free software business in three specific points: describing and detailing profitable, valid and viable business models; indicating the basic features of the plan for setting up and managing free software companies; and specifying the features of the planning and

organisation of free software production. This module has two appendices, the first covering the main licence models and the second indicating the most common open standards.

The second module is devoted entirely to the analysis of four successful case studies with free software implementation, two private and two in the public domain. Firstly, we present the case of Spain's autonomous region of Extremadura and its global project of technological actions to spread the use of the Internet, e-government, education and network integration and support to the new economy.

Secondly, we look at the Federal Government of Brazil and the diverse actions it has taken for the digital inclusion of the country's citizens and to increase both the skills and participation of citizens in new technologies.

The third case looks at the multinational Sun Microsystems and its embracing of free software and open standards, demonstrating the compatibility between collaboration with the free software community and business for profit.

Our fourth and final case is Cometa Technologies, an SME offering solutions based on the use of open source tools and open standards, formalising a business model that ranges from development and integration to consultancy and training.

We hope that the combination of the conceptual and methodological approaches of the implementation projects of the first module and the detailed practical description of the different case studies in the second module will enable you to understand the specific features of the implementation of free software systems.

Amadeu Albós Raya
Óscar David Sánchez Jiménez

Objectives

The aims for students of this subject are:

1. to know the basic concepts of the implementation of free software systems;

2. to be able to identify the different types of project to implement free software systems;

3. to be familiar with the project management areas and their specific features in application to free software;

4. to know the main concepts of free software projects and the phases of their life cycles;

5. to learn how to draft a free software project proposal;

6. to obtain a deep knowledge of the features of free software migration projects;

7. to learn how to plan and execute free software migration projects;

8. to know the business models used by free software companies;

9. to become familiar with the items of business plans and apply them to the creation of a free software company;

10. to know the main aspects of free software production;

11. to become familiar with the different bodies and free software projects carried out by the regional government of Extremadura;

12. to become familiar with the different bodies and free software projects carried out by the Government of Brazil;

13. to understand the social impact of free software and its potential for bridging the digital divide;

14. to become familiar with the development methodology of projects put into practice by Sun Microsystems;

15. to be aware of the advantages that participation in free software projects offers to companies; and

16. to confirm the viability of free software business models through the case study of Cometa Technologies.

Contents

Annex

GNU Free Documentation License

GNU Free Documentation License
Version 1.2, November 2002

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any

member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent.

An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of

transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front

cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit.

When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form.

Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their

copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Free software implementation, projects and companies

Amadeu Albós Raya
Óscar David Sánchez Jiménez

PID_00148382

UOC
Universitat Oberta de Catalunya
www.uoc.edu

Index

Introduction

This module looks in detail at the methodology used to implement free software systems in organisations and generic scenarios, establishing the main features that will guide the project and its development.

Technology is now becoming a strategic factor in both public and private organisations, whether small, medium or large. The integration of technology in all of the organisation's functional and operating processes means that its state of operation is closely related to the production of the organisation. As a result, we need to systematically monitor the efficiency and effectiveness of the system in order to control performance in line with the evolution of the strategic needs of the organisation.

The implementation of systems cannot be left at random or to the convenience of factors that are irrelevant to the organisation because the results can be unexpected and have consequences of varying magnitudes for the organisation. Hence, we can see the need for a rigorous implementation of systems with the methodology used in scientific and technological circles, by which we establish a framework of support with guarantees for management of the complexity, development of the project and management of the risks inherent both to the technology and the system implementation process.

The main aim of this module is to provide a broad and detailed vision of the main processes related to the implementation of free software systems, both in a generic or abstract context and from a perspective focusing on the characteristics of the functional and operational management of the project to implement the system, or considering the free software business as a valid and viable option for a profitable company.

All of the above must be subject to a methodical and rigorous management allowing, on the one hand, satisfactory management of the complexity of solving the specific problems of the systems implementation project and, on the other, keeping under control all of the potential risks that could cause the project to fail in one way or another, whether prematurely or while the system is already in operation, both important consequences for the organisation and its users.

The contents of this module are divided into three units that gradually introduce the key theoretical concepts of the process of implementing a system in an organisation. Its sections present the implementation of systems from the perspective of free software, analysing the main features that will produce a successful project both from a project management perspective and

from that of the implementation of free software as a profitable business. A brief summary then follows of the main features of each of the units in this module.

The first section, "Introduction to systems implementation", gives a general presentation of systems implementation and the features of free software. We define the term "systems implementation" as an action resulting from the strategy of the organisation, detailing the different reasons leading to implementation, the main stages of the process and a classification of the main typologies based on project context and aim. We present the specific features of free software, their effects on systems implementation and the functional management of the project in terms of scope, time, integration, cost, quality, human resources, communication and risks.

The second section, "Free software projects", presents the free software implementation project in detail from the point of view of its methodology. We define the life cycle of the project and its main characteristics, repercussions and relationship with the aims. The stages of the implementation project are described in detail: study of the current situation, study of implementation issues, analysis of free software solutions, formalising of the proposal, development, implementation and migration, and lastly, training, communication and support for users. Each of the stages is defined and described in detail in relation to its specific and global objectives.

The third section, "Free software companies", provides a detailed description of free software companies as a valid and viable alternative to sales of copies of software. We present the main business models for free software, based primarily on the production and sale of extra services. Lastly, we take a detailed look at the features of the business plan of free software companies, focusing on aspects such as definition, scope, organisation, human resources and materials, production, product evolution, quality control and monitoring, guarantees and user support, economics, financing and the project feasibility study.

We hope that the conceptual, methodological and practical formalising of the main aspects of the project to implement free software systems will enable you to understand the need for and importance of the process and the actions that must be taken to guarantee the strategic aims of the project.

Objectives

The aims of this module are:

1. to be aware of the basic concepts of the implementation of free software systems;
2. to be able to identify the different types of project to implement free software systems;
3. to be familiar with the project management areas and their specific features in application to free software;
4. to know the main elements of free software projects;
5. to know the life cycle phases of free software projects;
6. to learn how to draft a free software project proposal;
7. to obtain an in-depth knowledge of the features of free software migration projects;
8. to learn how to plan and execute free software migration projects;
9. to know the business models used by free software companies;
10. to become familiar with the main elements of a business plan;
11. to learn how to draw up a business plan based on free software; and
12. to know the main characteristics of free software production and its specific features.

1. Introduction to the implementation of free software systems

This first module of this subject, *Implementation of free software systems*, provides the basis for discovering the main concepts of the implementation of systems in general, and its application to free software in particular.

The module begins with an overview of the basic concepts of systems implementation projects, characterising the different phases of the process from a methodological and functional point of view and analysing the close relationship between the strategic objectives of the organisation and the implementation project.

We then provide a classification of the most common types of implementation project, indicating their basic characteristics and the differences between them, and carefully analysing the implications of this for the objectives and performance of the project.

After summarising the basic concepts of systems implementation and the different types of project it includes, we look at the specific features of the implementation of free software systems. We analyse the main factors influencing the project and the pros and cons of implementation with free software.

Lastly, we look at the basic concepts of project management and expand on the classical models in order to adapt them to the specific features of free software implementation.

1.1. Basic concepts

This first section provides an overview of the basic concepts of systems implementation, from conceptual definition to its main phases, including methodological, strategic and organisational aspects.

1.1.1. Definition

Systems implementation has always been closely associated with the evolution and popular diffusion of technology. In fact, if we consider this concept from a global perspective, any innovation – technological or otherwise – that we wish to extend beyond the borders of the context of its creators needs to undergo an implementation process.

When implementing technology, we need to pay attention to certain basic aspects, such as the impact on the organisation and direct users, but also on indirect users and clients, for example. Furthermore, technology – in the global and generic sense – is currently seen as a decisive factor in the competitive evolution of organisations.

> Systems implementation is the process by which one or more technological innovations are introduced into an organisation, as the result of an action deriving from its strategic plan.

In line with this definition, the implementation of technological systems is the result of an organisation's strategic desire to reach new milestones, whose aims can be very wide-ranging, depending on the context of the organisation.

We can illustrate this concept with two examples from different contexts:

- Companies, as profit-making organisations, can take action concerning the technology they use in an attempt to increase their competitiveness and obtain greater market shares, thus being able to offer more innovative products or ones tailored to new demands.

- Government agencies, as non-profit-making organisations, can take action concerning the technology accessible by the region they govern in an attempt to provide competitive tools that will reduce the digital divide and develop the sector's economy.

1.1.2. The organisation's strategic plan

In the previous section we mentioned the organisation's strategic plan and its important role in systems implementation.

> An organisation's strategic plan is a series of proposals setting down the future aims or directions of the organisation. It usually covers a five year period and is developed in the diverse divisions or functional departments of the organisation.

The main aim of the plan is to minimise the risks and maximise the results of implementation by marking out real, affordable and measurable directions generated by the relationship between the organisation and the area in which operates.

Very often, SWOT[1] analysis is used to diagnose the organisation's current situation. This tool summarises in a single table the main factors influencing the structural operation of the organisation in its context at a given time.

(1)SWOT is an acronym for *strengths*, *weaknesses*, *opportunities*, *threats*).

Considering that the system implemented has to meet the current and future needs of the organisation, we can understand the importance of the link between the organisation's strategic plan and the implementation project.

The organisation's strategic plan evolves over time and adapts dynamically to changes in the context in which it is carried out. Additionally, the organisation's system must evolve with strategic changes and must conclude some of the actions begun with a new systems implementation project.

1.1.3. Origin of systems implementation

In the preceding paragraphs, we saw the link between the implementation project and the organisation's strategic plan. Organisations do not introduce implementation projects without first determining that they are necessary for their specific strategies.

Thus, the implementation of a system requires having detected deficiencies in the organisation's current system, although they can also be implemented in new organisations or those without prior technological positioning.

Broadly speaking, there are four generic origins leading to the implementation of a new system:

- **Detection of problems**: there may be diverse cases in which the current system operates inefficiently, which compromises the everyday tasks of its users and the reliability of the system.
  In these cases, the strategic plan is chiefly affected by loss of performance and efficiency in the organisation.
  An example of this type of situation could be programming errors that produce inaccurate calculations, access errors or system locking.

- **System evolution**: this covers situations in which the current system is functionally obsolete, which compromises the organisation's operation due to the lack of features to solve the increasing number of issues in the organisation.
  In these cases, the strategic plan is affected by the organisation's loss of effectiveness. An example of this type of situation might be the need to increase features following a change in legislation.

- **System enhancement**: in these situations, the system in place is structurally obsolete, which compromises the organisation's operation due to the poor performance of the platform of the current system.
  In these cases, the strategic plan is affected by the loss of performance and efficiency of the organisation. An example of this situation might be the lack of integration of new operating systems or different types of hardware.

- **New strategic action**: this includes possible updates, changes or innovations in the organisation's strategic plan not covered by the current system.
  In these cases, the strategic plan is affected by the organisation's loss of effectiveness. An example of this type of situation might be an increase in the services offered or expansion of the target market.

The above list of possible origins is not all-inclusive but it does indicate the most logical reasons for systems implementation. In addition, the different cases are not mutually exclusive and their coincidence can be strongly motivated by the evolution of the organisation and its system.

1.1.4. Resources of a systems implementation project

Normally, once the need to adapt the current system to the organisation's strategic plan has been noted, resources are allocated to the new systems implementation project. Initially, these resources usually take the form of one or more individuals with free time to spend on the project, with the financial repercussions that this will have on the everyday workings of the organisation (insourcing[2]).

(2)The term insourcing refers to the internal delegation or production of a process.

Some organisations prefer to leave this task to professionals outside the company (subcontracting or outsourcing[3]) for reasons of functional objectivity, production capacity or time availability. In these instances, it is not the case that the organisation spends no time at all on the project; rather, this time is reduced by the degree to which outsourcing increases, since both the organisation and the external professionals need one another to bring the project to successful completion.

(3)The term *outsourcing* refers to the external delegation or production of a process.

One of the key points of an implementation project, regardless of its final form of execution, is the creation of a supervisory or monitoring committee for the project (executive committee in some organisations). This committee is charged with the methodological execution of the project and its adequate, gradual and sustained progress over time.

The supervisory committee is normally made up of individuals from the various divisions affected by implementation, mainly management staff and heads of department. If the organisation contracts external professionals to

manage the implementation, they will also form part of the committee. Although most of the committee members only spend part of their time on it, there is usually at least one member that works on the project full-time to ensure rigorous monitoring.

The importance of resources in the implementation of a system is twofold:

- Firstly, human resources are allocated based on the quantity and quality in the analysis and design of the system implementation.

- And secondly, material resources are allocated according to the quality and quantity of the system to be implemented.

In all events, the resources allocated to a systems implementation project will have a direct effect on the finances of the organisation.

1.1.5. The main stages of a systems implementation project

As we explained above, a systems implementation project is a methodological process designed to adapt the system to the strategic plan. This process must be performed with due care and attention in order to guarantee the success of the project.

From a generic point of view, we can break the systems implementation process down into four main phases: analysis of the current system, design of the new system, development, and implementation of the system.

Analysis of the current system

All implementation projects begin with a study and analysis of the current status of the organisation in the terms indicated by the strategic action. There are two possible initial situations:

- If the organisation has a system in place, its characteristics are assessed by collecting information on the elements affecting the strategic action and their structure. The aim of this is to create an abstract framework for adapting the system to the new strategy.

- If the organisation does not have a system in place (or it is a new organisation), we need to evaluate the features of the area of action that will be affected by the strategic scope of the organisation's operation. We will need to create an abstract framework of aims that the implementation project must meet.

In either case, this stage defines and determines the problems that the implementation project will need to overcome, given that we will select the different aims of the strategic action. We evaluate aspects such as system history, structure and operation or the evolution of issues and workload over time. Many of these results are presented in the form of charts or diagrams.

The stage ends with a presentation of the conclusions of the study and an analysis of the current status, which evaluate the extent to which the system can stand up to new strategic challenges.

Design of the new system

Once we have assessed the initial situation of the organisation, defined the main points of the strategic action and received confirmation of monitoring from the project's executive committee, we can begin to design the new system.

This stage begins with a thorough analysis of the issues that the new implementation will need to resolve, based on the strategic actions we have defined. We will come up with different solutions or alternatives for these problems that we will need to analyse individually to gauge their suitability and determine their costs, advantages and disadvantages, both tangible and intangible. Depending on the type of project and strategic action, this evaluation will cover a five-year period.

We must be objective and use a methodology when choosing our solution in order to maximise the advantages and minimise the disadvantages both of implementation of the solution and its everyday operation. Some decision-making criteria may concern the scope of the action, performance, the necessary equipment, the requirements covered, provider support, availability of equipment and support, and the maintenance required over time.

Development of the new system

After receiving confirmation of the project monitoring by the supervisory committee, we can begin to develop the new system.

The development of the system or adaptation of the proposed solutions adopts a life cycle most suited to the purpose of the project, some of which we have studied in other subjects. At the end of this stage, we obtain a system ready to be implemented in the organisation that will resolve the strategic issues observed in the previous stage.

See also

The subject on *Software engineering in free software environments* contains more information on the life cycle of software development.

Implementation of the system

Once the new system is ready, we can begin the implementation phase, which involves setting up the system in the organisation.

This phase sees the completion of the adaptation and integration of the new system into the real environment and covers user training, pilot tests and system integration with end user testing, and the conversion and final release of the new system.

If the organisation already has a system in place, we must also take into account the migration from the old system to the new one. Migration involves transferring the current status of the system in place to the new system. Data transfer is usually the most important task of migration, since this activity cannot affect the day-to-day running of the organisation.

This stage ends with the final implementation of the new system, data migration and user training. In other words, we will have fully introduced the elements needed for performance of the strategic action we defined at the start.

Although a considerable proportion of the implementation project is completed with the implementation of the system, the maintenance and continuous evaluation of the system's feasibility remain active as in any other project, particularly if the new implementation is regarded as a strategic factor in the competitiveness of the organisation.

1.1.6. Feasibility and evaluation of the project

The previous section outlined the main phases of a systems implementation project but we also need to consider the importance of at least two control points for the project.

The first is the feasibility and continuation of the project, which is assessed on the basis of two milestones:

- The first milestone occurs after the current status analysis stage, in which we analyse and discuss the current system status in terms of the strategic action.

- The second milestone takes place after the design stage, which involves an analysis and discussion of the proposed solutions.

For instance, the convenience of continuing the project could be questioned given the financial implications of its development.

The second control point is the project evaluation, which takes place once the system has been fully implemented in the organisation. At this milestone, now that implementation is complete, we analyse the true repercussions of the new system by evaluating the measurable impact factors described in the strategic action that led to the implementation process. The aim is to evaluate the new system's concordance with the strategy of the organisation. These measurable indicators are observed from a longitudinal perspective.

Longitudinal perspective

Observing measurable indicators from a longitudinal perspective means that the observation is not unique in time but is in fact repeated several times over a previously defined period.

For example, we can evaluate the impact of the new system on the organisation's production capacity or on its ability to attract new clients.

1.1.7. Project methodology

As in all strategic projects, systems implementation must be carried out in a methodological, structured and orderly way. The importance of the project outcome for the development of the organisation dictates the precision with which it needs to be carried out.

To illustrate this concept, we can contrast it with the methodology applied to life cycle in software development and the importance of the conceptual analysis and solution design phases. Here we can see the problems caused by not detecting failures in the analysis and design stages. Solving a design problem detected at the development stage is generally considered to have a possible cost ratio of one to ten.

Systems implementation projects also require maximum precision in the analysis and design phases because of the strategic implications of the new implementation on the organisation's evolution.

The stages described in the previous sections have a sequential behaviour, inherent to their aims, but we can still specialise the methodology for execution of each stage in order to achieve our aims more efficiently. One example would be the distribution below:

- The current system analysis phase may use an iterative method, with feedback obtained from the study results.

- The design and development phases may use an XP[4] method if we are developing software, or a classical schema if we are implementing an infrastructure.

(4)XP stands for eXtreme Programming, an agile software development methodology.

- The implementation phase can use an iterative process if it affects a number of units, with the possibility of maintaining more than one line of implementation for each time unit.

In all events, we will need to adapt the methodology of each stage to the convenience of the project and the organisation, always with the aim of maximising efficiency and minimising any negative impact.

1.2. Types of project

This section contains a general classification of the different types of implementation project and describes the diverse features and main implications for project development.

It offers two classifications, one based on project scope, that is, the repercussions and scope of the implementation, and a second that considers the aim of the project, that is, the contents of the implementation.

This classification should not be regarded as a rigid structure, since the true diversity of the projects allows a combination of different typologies.

1.2.1. Classification by scope

We can define three broad scopes of action for implementation projects:

1) Internal

Projects whose scope is internal are designed to implement a local system in an organisation that will primarily be used internally.

Examples of projects with an internal scope are those implementing local network services (directory, authentication or data sharing services), those implementing tools in local groups (internal mail or groupware) or those implementing internal management tools (ERP systems).

The strategic aim of this type of project is to introduce functional internal improvements to the efficiency and efficacy of work through technological renovation. These are normally used in scenarios in which the current system reveals functional deficiencies or is unreliable.

The main implications for project development are as follows.

- **Evaluation of the current system**: This stage basically evaluates the system from a functional perspective, pinpointing and evaluating the aspects of the system that could cause problems, always taking the strategic action outlined by the organisation as a reference.

- **Design of the new system**: The issues analysis carried out at the design stage must allow us to compare two alternatives for implementation:

on the one hand, evolution of the current system, and, on the other, a complete overhaul of this system.
While evolution seeks to partially update the current system in order to extend its useful life, a complete overhaul requires changing various elements of the system (hardware and/or software).

- **Development of the new system**: There are no major differences in the development stage. In all events, the local scope of the project can help us to define the latter and provide a strategic resolution.

- **Implementation**: The project's internal scope can help with progressive implementation (of time and/or services) which, along with staff training, must contribute to the creation of a mood of acceptance while guaranteeing the day-to-day operation of the organisation.

2) External

The aim of external projects is to implement a system essentially for public use in the organisation, connecting external agents to the organisation. The system can be placed locally or remotely.

Examples of projects with an external scope are those that implement corporate intranets or extranets (tailored access to workers, clients or suppliers) or implementation of electronic government services (e-government or e-voting).

The strategic aim of this type of project is to introduce functional improvements to the organisation in its external dealings, enhancing the efficiency and efficacy of its communication through technological renovation. They are normally used in scenarios in which the management of digital relations with third parties is complex or inefficient and in cases where we are looking to improve the corporate image and target market.

The main implications for project development are as follows.

- **Evaluation of the current system**: This stage basically evaluates the system from a relational, communicative and interactive perspective. Data collection cannot be limited to the internal part of the organisation, so contact with external agents will be required. Regardless of whether or not there is a system in place, the aim of the strategic action will be to highlight the shortcomings and weaknesses of the current relational implementation.

- **Design of the new system**: The design of the new system basically needs to cover two types of aim: the efficiency and efficacy of user functionality,

and considerations relating to the corporate image that the organisation wishes to convey to the target market.

- **Development of the new system**: There are no major differences in the development stage. However, it should be noted that security management is particularly important with this type of project.

- **Implementation**: The full implementation of these systems can be carried out in two phases: we can first set up the basic system and then we can progressively update the services and contents over time, taking into account the opinions of external users on the changes that have taken place (feedback).

3) Productive

The aim of projects with a productive scope is to implement a system in a different environment to that of the organisation managing and/or developing the project (outsourcing).

Examples of projects with a productive scope include core software implementation (operating systems), specialist software implementation (office automation tools, accounting, invoicing, etc.) and implementation of outsourced services (subcontracting of website services).

The strategic aim of this type of project is to meet the demands or needs for technology services of other organisations or external groups from diverse fields. It is normally used in scenarios associated with opportunities for a strategic and technological change in the target market.

The main implications for project development are as follows.

- **Evaluation of the current system**: This stage basically evaluates the system from two different angles. Firstly, specialised projects for an organisation with specific strategic needs to be met.
  And secondly, direct implementation projects (operating systems or office automation tools), designed to provide generic solutions for a large market segment. In both cases, the importance of studying the status of the current situation is clear.

- **Design of the new system**: There are no significant differences in the design stage. Specialised projects are carried out in accordance with the internal or external project considerations in the client organisation while direct implementation projects need to meet everyday operating needs

and/or offer new features. The technology used in the project is important for the corporate image of the organisation.

- **Development of the new system**: There are no major differences in the development stage. It is important for the development to meet the aims of the client organisation efficiently and effectively, and that this is done within the fixed time limit.

- **Implementation**: Specialised projects are implemented based on typology (internal or external project), with an emphasis on user training and communication. Direct implementation projects are normally implemented as a self-installing product package (on CD-ROM or DVD-ROM, or as a download from the Internet). Some services can be considered extras (training, migration, etc.) and supplied on request by the organisation or by third parties.

1.2.2. Classification by requirement aim

Three main groups of aim can be defined for implementation projects:

1) Software

The aim of software development projects is to implement applications to meet certain demands. This type can also include projects for the adaptation, reengineering or integration of software or tools, such as the adaptation of operating systems or the integration of specialised software packages.

The strategic aim of this type of project is to provide a technological response to a specific functional problem. It is normally used in scenarios where systems are implemented to automate tasks, provide efficient technological support to users or to evolve or replace obsolete software.

The main implications for project development are as follows.

- **Evaluation of the current system**: This stage basically evaluates the system from a functional perspective, evaluating overall efficiency and efficacy in terms of the strategic action that launched the project.
  In projects involving the creation of new or generic software, the evaluation focuses on the study and analysis of the current status of the target market and its main directions.

- **Design of the new system**: There are no significant differences at the design stage in the life cycle of software production that we have studied in other subjects. The use of methodology is important during the analysis of needs and system design phases to minimise the errors detected at later stages.

See also

For more information on the life cycle of software production, see the subject on *Software engineering in free software environments*.

- **Development of the new system**: There are no major differences in the development stage regarding the usual life cycle of this type of project. Aspects of importance include the mechanisms guaranteeing the quality of the code produced and the evolution of the code in versions and revisions.

- **Implementation**: We can distinguish between two possibilities at the system implementation stage. On the one hand, the development of software that needs to be installed in an organisation, which will follow the usual process, taking into account the possible need for migration if there is a system already in place. And on the other, common or generic software, which is usually implemented as a self-installing product package on a CD-ROM or DVD-ROM, or as a download from the Internet. Some services can be considered extras (training, migration, etc.) and supplied on request by the organisation or by third parties.

2) Infrastructure

The aim of infrastructure implementation projects is to implement architectural or structural systems in a given environment. These projects are usually used to implement functional equipment that provides a basic service for the organisation.

Functional equipment providing basic services

This applies to the installation and configuration of servers or clients, such as operating systems, office automation software, basic network services (DNS, DHCP, etc.) and advanced network services (mail, groupware, etc.).

The strategic aim of this type of project is to provide a functional technological basis to meet the requirements of the organisation. It is normally used in scenarios of new creations or technological renovation due to the system becoming obsolete. We need to remember that the organisation's infrastructure is its functional basic architecture, on which the rest of the technological elements will be organised.

The main implications for project development are as follows:

- **Evaluation of the current system**: This stage basically evaluates the system from a functional perspective, taking into account aspects such as efficiency and efficacy as well as reliability and adaptation to the new standards of technological evolution.

Where there is no previous system in place, we will basically need to take into account the needs (quality and quantity) of the organisation, together with current directions and standards.

- **Design of the new system**: The design stage deals mainly with the research and study of the diverse solutions available on the market, although we cannot rule out the creation or evolution of a solution if the needs of the organisation so require. By comparing these options, we should come up with the product that best fits the strategic action, taking into account evaluation criteria such as the scope, efficiency and efficacy of the solution, equipment and support requirements, availability and product maintenance.

- **Development of the new system**: The system development stage involves the preparation of procedures to implement the infrastructure, bearing in mind that it is sometimes necessary to tailor the product or adapt configuration files.

- **Implementation**: If the new system replaces an earlier one, we will need to migrate from one system to the other. Nonetheless, we can split implementation into two phases: firstly, the installation and configuration of the new system features and secondly, start up and status restore.
  User training and communication are essential for this type of project to ensure acceptance of the new features and for an assessment of its operation after start-up.

3) Systems migration

The aim of systems migration projects is to transfer the status of the current technological architecture to a different one. These projects are implemented when one or more main elements of the system are updated.

The strategic aim of this type of project is to minimise the impact of technological changes on the operation of the organisation. It is normally used in scenarios where new software systems or infrastructures are implemented, but they can also be developed independently to replace the physical technological platform in order to improve performance, reliability or in the event of obsolescence.

The main implications for project development are as follows:

- **Evaluation of the current system**: This stage evaluates the system primarily from the point of view of preserving the configuration and

stored data. It is important to adopt a methodological approach in the research and evaluation of the various elements that need to be migrated to the new system.

- **Migration design**: The design stage normally involves studying and analysing the methods and procedures that we will need to implement to transfer the status between the two systems. We can consider aspects such as the preparation of backups or the design of procedures to export or convert the data in the current system.

- **Migration development**: Migration development requires the performance of all tasks to preserve the data and configuration, and to export and convert the current system. With this task, we must take into account the importance of the temporary location, to ensure that data is transferred fully without interrupting the day-to-day operation of the organisation. We also need to consider the security guarantees of the backup medium.

- **Implementation**: Implementation covers start-up of the new system and the status restore of the organisation's previous system. The timing of this process must be planned in conjunction with the migration development stage in order to minimise the impact of the change on the day-to-day operation of the organisation, although the progressive restoring of some minor elements may be possible. Communication and collaboration with users is particularly important in this type of project if we are to meet the aims of migration successfully.

1.3. Free software systems

This section introduces the specific features of the implementation of free software systems. We will analyse the main factors affecting the project and the myths, advantages and disadvantages of using this type of software.

When we talk about free software, we generally refer to its advantages, which are fairly well-known. For instance, free software is secure and good quality, is distributed freely, uses open standards and is based on the culture of collaboration and promotes the latter. What is more, it can be used anywhere, enhances technological capabilities, helps to reduce overheads and operating expenses in IT systems, reduces dependence on providers and fosters the development of local companies.

However, the companies that implement free software systems come up against a series of problems that is hindering the sector's development (Sáez et al).

Generally, to encourage the development of a technology, we need this technology to be commercially viable – i.e. there has to be a supply and demand – and economically feasible, in that the companies from that sector must generate profits through their implementation of the technology.

With **demand**, that is, the companies that could adopt the free software, the main obstacles are piracy, fear of change and misgivings. Companies confuse free software with freeware, and some companies rule out its implementation either because free software of similar standards is not available or because they do not believe that there are companies behind this software guaranteeing its support and maintenance.

With **supply**, that is, the companies that provide the applications and free software services, the main obstacles are fear of change and the lack of cooperation.

On the whole, IT companies are used to developing tailored software solutions without giving their clients the sources, and setting up a model of licence payments for their use. And yet, with the appearance of free software, some companies are seeing that a business model based on payment for services rather than licences can be sustainable, leading them to release part of the code that was thus far kept private. Therefore, one possibility that could be used initially by many companies would be to offer two solutions to clients: ownership (with a licence fee) and free (normally without a licence fee). This is just one of many possible business models for free software.

IT companies are also used to developing software on an individual basis but a better option would be to develop the cooperative model. This would involve drawing on work done by others, collaborating, and transferring licence savings to the end client.

Free software projects may be approached as normal projects from the point of view of software engineering and project management. However, a closer look will reveal certain differences and specific features that are often only dealt with correctly with experience and whose omission can affect and even bring about the failure of some projects.

In the light of this, the use of a comprehensive methodology for the implementation of free software systems is essential, particularly because:

- It gives **clients more confidence** in the quality of the products and processes, whether we are developing a new programme or application, migrating an existing system or starting up a new one.

- It allows providers to systematise the procedure for **the implementation of free software systems** and become familiar with their features, which results in improved efficiency and allays fears of change.

1.4. Management of free software projects

Project management is traditionally divided into nine areas of knowledge:

- Scope
- Time
- Integration
- Cost
- Quality
- Human Resources
- Communication
- Risks
- Supplies

This section will look at each of these in turn and study their specific features when applied to projects implementing free software systems.

1.4.1. Scope management

Scope management involves defining the aims of the project so that we can check to see that they are being met and, if needed, change them. In other words, scope management ensures that the project carries out all necessary work – and only the necessary work – so that the initial aims can be met.

Definition of the project scope

To define the scope of a project, the project manager must establish the project work breakdown structure (WBS), which divides the project into work packages, usually represented in the form of a logical tree. A work package is the smallest unit into which a project can be divided in order to make it independently manageable.

We therefore need to identify everything that needs to be done in the project through its WBS, briefly describing its work packages and the deliverables that each one needs or facilitates.

Changes to project scope

It may sometimes be necessary to modify the scope and aims of the project during its execution. This can be due to the following:

- Shortcomings in the original project plan, especially incorrect definition of the scope.
- Changes to the needs and requirements of the client established in the initial project plan.
- Changes to the context of the project and hence, to the hypotheses considered when the initial project plan was drawn up.

These contingencies may have a significant impact on project execution, modifying and even preventing it from achieving its aims. Thus, it is essential to control changes and to take into account risk management.

Scope management in free software projects

The features of scope management in free software projects are the same as those of any other software project. It is essential to obtain and conduct a detailed analysis of client requirements – and of the current situation of the system if dealing with a migration project.

The definition of the scope of free software projects will generally depend on the motivation for the project: cutting costs, system improvement, independence of distributors or regularising the software licence situation.

Lastly, it is important to note that clients may not always be aware of the consequences of changes to the project once it is underway.

1.4.2. Time management

The purpose of time management is to ensure that the project is carried out within the set deadlines. This means that we will need to define the sequence of activities to be performed, along with their duration and coordination.

Good time planning is an essential part of project management because it establishes the model by which the project will be carried out. Moreover, it allows us to ensure that the aims are being met, it lays the foundations for integrating time, costs and resources, and it sets down a common framework for the various individuals and partners taking part in it.

Project network

The WBS we saw in the previous section is used to identify the activities needed to conduct the project, taking into account the fact that an activity is the smallest part of work into which a project can be divided for planning purposes.

Following on from this, we need to identify the order in which the activities will be carried out: independent activities can be carried out simultaneously while dependent activities require the result of a previous activity for their performance.

Gantt chart

A Gantt chart is a tool used to help solve the problem of scheduling activities (i.e. their organisation on a calendar) and represent in pictorial form the duration of each activity, its start and end dates and hence, the total time needed to complete the project. Gantt charts also enable us to monitor the project's progress because they indicate the completed percentage of each activity and detect advances or delays in the initial planning.

Gantt charts have a system of coordinates representing:

- Horizontal axis: time scale, in the appropriate units for the project (usually, days, weeks or months).
- Vertical axis: work packages, activities and subactivities identified in the WBS, whose duration is represented on the horizontal axis.

The main advantage of Gantt charts is that we do not need masses of information to be able to use them; in fact, all that is needed is a rough outline of the project. Hence, they are straightforward to use and particularly effective in the initial planning of the project. However, once the project is underway, particularly if it is very complex, Gantt charts can become confusing.

Critical path method and PERT

To overcome the limitations of Gantt charts, other tools have been developed, such as the critical path method (CPM) and the PERT method.

The critical path of a project is the sequence of activities that generates the maximum accumulated time. It determines the shortest time taken to complete the project if we have all of the necessary resources. To do so, we need to identify all of the activities correctly and know their duration.

To represent the activities and time dependencies, directed graphs are used, whereby each arrow represents an activity identified by its name and duration, so their status changes as the project progresses. Each status is represented by a node between two or more arrows. Thus, some tasks can be conducted at the same time whereas others cannot.

The main difference between CPM and PERT is their time estimates. CPM considers activity times (*m*) to be known exactly and to vary according to the resources allocated to them. PERT, on the other hand, assumes that activity times (*Te*) are determined by a probability distribution generated by the most likely time estimate (*m*), the most optimistic time estimate (*a*) and the most pessimistic time estimate (*b*). Thus, the most pessimistic and most optimistic times give a measure of the uncertainty of each activity.

$$Te = \frac{a + 4m + b}{6}$$

The following steps are taken to calculate the critical path:

1. Calculate *Te* or *m* for each activity, depending on the method used.

2. Calculate the early start dates of each activity (ES) and the last start dates of each activity (LS).

3. Calculate the float of each activity

The float of an activity

The float of an activity is the spare time we have to determine this activity:

- Float of an event: *Hs* = LS of the event – ES of the event.
- Float of an activity: *Ht* = LS of the subsequent event – ES of the previous event – activity duration.

4. Identify the critical path of the project

A critical activity is one whose start and end points cannot be changed without modifying the total duration of the project. Critical activities have no float and the sequence of these critical activities is the critical path. To put it another way, in a critical activity, the early start date will coincide with the latest start date and the earliest finish date will coincide with the latest start date of the activity.

Calculating the ES

The ES of each event is calculated as the maximum duration of the previous activities plus the ES of the previous event. The LS is the last date on which the events can take place without delaying project completion.

See also

To find out more about each of these methods, see the bibliography at the end of the module.

Time management in free software projects

The features of time management in free software projects are theoretically the same as those of any other software project.

In projects developing programmes and applications in which the community of free software developers plays a key role, correct calculation of software development deadlines is essential. To do this, we need to know the background of the community and discuss the future implementation plan. It is also a good idea to get involved in the community and learn about how it works before starting the project.

In migration projects, we need to set aside the right amount of time for training users and introduce a degree of flexibility for the migration of users.

As a result, some free software projects can be accompanied by a degree of uncertainty, so we recommend the use of the PERT technique to characterise the most optimistic and pessimistic scenarios in the project plan.

There are many free applications to create and maintain PERT and Gantt charts.

Free applications

Examples of free applications include GanttProject (http://ganttproject.sourceforge.net/) and OpenWorkbench (http://www.openworkbench.org/).

1.4.3. Integration management

The purpose of integration management is to ensure that the different parts of the project are coordinated correctly. This includes developing the project plan and the project execution plan and tracking any changes that may occur.

Integration management in free software projects

The features of integration management in free software projects are generally the same as those of any other software project, but we need to bear in mind a few points.

By and large, integration in free software based projects has certain advantages over that of proprietary software projects, mainly due to their open source and the use of open standards for interoperability between applications, particularly with those developed outside the project.

In development projects carried out in collaboration with a community not forming part of the project, it is important to make known and discuss the implementation plan both of the project and the community, in order to identify possible incompatibilities that could affect integration.

1.4.4. Cost management

The purpose of cost management is to conclude the project with the budget approved at the start. This means planning the required resources and estimating and monitoring costs.

Cost management in free software projects

Cost management in free software projects differs considerably from that of proprietary software projects.

The main difference lies in licence costs, which are normally non-existent for free software. In contrast, we will need to take into account the costs of services provided by third parties, in accordance with any of the models of business based on free software.

1.4.5. Quality management

The purpose of quality management is to ensure that the project meets the needs for which it was initially designed. We must therefore plan, assure and continuously monitor the quality of the project in accordance with these needs.

Quality management in free software projects

The features of quality management in free software projects are theoretically the same as those of any other software project.

On the one hand, we need to consider quality from the user's point of view, adopting the standards required in each case. And on the other, in the case of projects in which we work with the free software community, whether contributing to an existing project or creating a new one, we need to take into account the quality of the code produced from the point of view of the developer.

We will need to follow the recommendations on programming style, naming conventions, documentation, error logs and formats, languages, etc. With new projects, it is a good idea to circulate these recommendations.

Standard programming styles

Remember that you should follow the programming styles that have already been defined and accepted by the community, such as the Java Code Conventions or Linux C kernel style.

1.4.6. Human resources management

The purpose of human resources management is to employ the individuals participating in the project as efficiently as possible. The activities carried out as part of this management include the organisational plan, hiring of new employees and team development.

Human resources management in free software projects

The features of human resources management in free software projects are theoretically the same as those of any other software project, but you should bear in mind a few points.

Most importantly, you will need to consider the possible participation of the free software community and the effective contribution it may bring to the project. It is generally a good idea to appoint somebody in charge of relations with the free software communities connected to the project.

1.4.7. Communication management

The purpose of communication management is to ensure the correct generation, collection, circulation, storage and elimination of project information within set deadlines.

Communication management in free software projects

The features of communication management in projects based on free software are theoretically the same as those of any other software project. It is very important to ensure the communication and circulation of information within the project, particularly when collaborating with the free software community. If you are not working with the free software community but there is a possibility of releasing the code or the client would like access to the latter, it is also important for the source code to be readable and well documented.

The configuration and correct use of software forges or collaborative development environments (CDEs) will therefore play a key role. Most forges have tools for general project management, bug tracking, forums, mailing lists, etc. Public forges also have these same tools and offer the advantage of more visibility from the free software community.

Software forges

For examples of software forges or collaborative development environments, see the following websites: Gforge, LibreSource and Trac.

Communication tools include mailing lists, IRC channels, blogs, forums and wikis. Relevant decisions made through these tools should be documented correctly and made available to all developers.

Further websites

You can find an example of a public forge at the following website: http://www.sourceforge.net.

We recommend defining rules that should be followed when drafting the documentation, together with the tools that will be used for its automatic generation.

1.4.8. Risk management

The purpose of risk management is to identify, analyse and respond to events that could jeopardise the project plan in the form of delays and increased costs.

These risks must be correctly identified and quantified, and have their appropriate response mechanisms. A risk is always characterised by uncertainty – since the event associated with the risk may or may not occur

– and by loss – because if the event eventually does occur, it will result in negative consequences or losses for the project. Thus, to be able to characterise the risks, we need to evaluate their probability and associated losses correctly.

There are several risk classifications. In this initial approach, we can consider the following three types of risk:

- Management risks, related to problems with scheduling, budgets and the organisation of staff and resources.

- Technical risks, which jeopardise the quality and scheduling of the project and pose obstacles to its development and implementation. The most common technical risks concern potential problems with design, implementation, verification and maintenance. They usually arise from ambiguities in requirements and specifications and the use of outdated or very new technologies.

- Business risks, which raise questions about the feasibility of the project. For example, developing a project for too small a market or one that does not mesh with the company's sales line.

Risk management in free software projects

The features of risk management in free software projects are theoretically the same as those of any other software project.

A classic example of risk in free software projects is the fear and uncertainty – of both the organisation and users – regarding technological change on a new and possibly unfamiliar platform.

It is useful to introduce communication and training methods at the start of the project in order to cancel out any negative effects of the rejection of technological change. It is good practice to earmark a time for presentation, communication and user training that will be continued and built on throughout implementation in order to provide an outline of free software and of the specific applications and tools.

Another classic example of risk in free software projects are the possible legal incompatibilities between free licences for the use of code development and reuse.

It is wise to check at the start of the project that the licences applied to the different parts of the code are coherent among themselves and that the planned development will not generate incompatibilities. This check should

be carried out before every release. It is good practice to keep a licence chart indicating the licence under which each of the software packages is distributed and the individual interactions between them.

1.4.9. Supply management

The purposes of supply management are to ensure that the materials and resources needed to execute the project are available at the right time and in the right place.

Supply management in free software projects

The features of supply management in free software projects are theoretically the same as those of any other software project.

The migration and implementation of a free software system is usually a good time to renew equipment or to modify the structure of the organisation's network. The project plan must therefore include orders of new equipment and materials in order to take into account and prevent possible delays.

2. Free software projects

A project is an organised, structured process of managing resources in order to achieve a specific aim, usually strategic. While, in the first part, we looked at the key aspects of the functional management of resources, this module will focus on the stages that the project needs to complete in order to achieve its aims.

We can generally define seven key stages in projects to implement free software systems:

- Study of the current situation
- Study of the implementation requirements
- Analysis of free software solutions
- Formalisation of the proposal
- Development
- Implementation and migration
- User training, documentation and support

As we can see, these stages are the result of development of the phases described in the first section of this unit and they apply specifically to free software. Nonetheless, the development described is fairly generic and can be applied to other implementation processes.

This section describes the life cycle of the project and outlines the process, its stages and its relationship with the management of the project and the resources allocated to it.

The subsequent seven sections will detail the stages of the project, linking and expanding on the concepts introduced in the first part of this module.

2.1. Life cycle

This first section introduces the main methodological and functional characteristics of the life cycle of the project in order to provide an outline of the process.

The life cycle of the project links the methodical aspects, intrinsic to the development of the implementation stages, to the functional management of the project. Thus, the life cycle guides the execution of the various stages over time and with the available resources.

By and large, the life cycle of a project has two main aims:

- Firstly, it establishes the relationships and dependencies between stages, whether time-based or functional.
- And secondly, it allows us to reduce project risk by dividing its complexity.

See also

To find out more about the management of risks, see the section on risk management in the previous unit.

The project life cycle can be used to monitor the evolution of the stages, the schedule of execution and the financial cost of the project. The management of this cycle is dynamic, so decisions to modify and adapt can be taken as time passes in order to readjust our estimates of the initial parameters on the basis of actual events.

Broadly speaking, the life cycle has four key areas: the project, the stages, execution and the results.

2.1.1. The project

Like any other project, systems implementation projects are designed to achieve a series of aims within a set period and with a given series of resources.

See also

To find out more about project management, see the section on the management of free software projects in the previous unit.

Minimising the time or resources spent on a project will usually minimise the aims that can be met or affect quality, and vice versa. In contrast, minimising project time while maintaining its aims requires us to increase the resources allocated to it. Project management seeks to find the most acceptable balance between these three elements.

In all events, changes in the relationship between these three elements have direct financial repercussions that will need to be assumed in the event that we are updating. The management of the project also has a financial cost from the point at which the project begins (when it is decided to allocate time of one or more staff to take on this management).

The main factors influencing the time and resources required to conclude a project normally refer to the size and complexity of the system to be implemented. In this sense, the features of free software tend to reduce the time and financial costs of the project:

- **Variety of applications**. The maturity of the free software market means the availability of a wide variety of direct implementation products that are reliable, consistent and secure.

- **Licence cost**. Free software is usually obtainable without licence costs and can be downloaded directly from the official website or from other public pools.

- **Source code modification**. The open nature of the source code allows the expansion, modification and adaptation of products that would require a new development to evolve the product if proprietary licence models were applicable.

It is important to note that free software also helps to reduce the overall risk of the project because it provides the freedom to view, use and modify the source code, allowing evaluation and assessment of all aspects of the application in depth.

Additionally, the project can be managed and executed internally or outside the organisation. Broadly speaking, we can differentiate between two main cases:

- ***Insourcing***: this is the case when an organisation develops a project launched as a result of a strategic action. In other words, the organisation's IT department manages and executes the project.

- ***Outsourcing***: this definition applies when an organisation delegates the management and development of a project to an external organisation that manages and executes projects[5]. In other words, the organisation reduces its direct exposure to the development of the project.

(5)One example might be technology consultancies, which carry out projects for other companies.

Thus, the format of the project development will take into account the capacity and experience of the organisation that must assume the development of the project, the associated costs, the implementation schedule and the specialisation of the external organisations present in the project.

Lastly, the project is evaluated in terms of tangible and intangible benefits. Here we can come across cases in which a surcharge for time or financial costs may be feasible in order to obtain the intangible benefits – usually strategic – required by the organisation. For example, improving corporate image with the use and diffusion of free software and the free philosophy.

2.1.2. The stages

The project life cycle is implemented in the form of successive and possibly simultaneous stages. Each stage meets a clear, set aim in a scenario related to the project, with the result that, taken together, these stages meet the aims of the project.

Broadly speaking, a stage can be considered a process that receives inputs and produces certain outputs. In other words, it requires a prior scenario with information about the environment in order to produce certain results.

Hence, a relationship is established between the diverse stages of the project, since each stage achieves part of the global aims. This relationship usually takes two forms:

- **Dependence**: between two stages indicates that a stage requires the result of the execution of the other in order to complete its task. This means that the stages must be executed sequentially over time: the stage generating the results first, followed by the stage that uses these results. For example, the development stage requires the study and analysis of the systems implementation requirements in order to complete its task.

- **Independence**: between two stages indicates that two stages have no direct relationship with each other and no specific prerequisite. This means that the stages can be executed simultaneously, though more resources may be required. For example, the system implementation stage can be executed at the same time as the user training stage.

Moreover, the stages also allow for a distributed execution of the project, that is, one or more stages are assigned to different teams, either internal or external to the organisation (insourcing and outsourcing).

Extreme cases may arise if several stages are assigned to different external organisations. It will all depend on the characteristics of the project, the specialisation of the external organisations and the associated costs.

All this highlights the importance of deliverables between stages. The importance of documenting results in the form of deliverables is threefold:

- Since the documentation of the stage summarises its development and results,
- since the result of the stage is important for the stages that depend on it, and
- since the result of the stage is a result that can be evaluated for the development of the project.

The connotations of the internal or external execution of each stage highlight the importance of deliverables. It should be noted too that their importance is proportional to the complexity and size of the project.

2.1.3. Execution

Execution of the project will begin in accordance with the initial proposed planning, with careful monitoring by the organisation in which the system is to be implemented. By and large, three key parameters need to be monitored:

- **Time**. Time monitoring and management are essential for project monitoring because any change to this parameter will have direct financial consequences. It is also particularly important for the sequencing of the different stages, particularly if they are assigned to different teams.

- **Outsourcing**. Monitoring the outsourcing of the stages or perhaps the entire project is important if we are to ensure that the work and its results meet the aims of the project and the organisation. We need to pay close attention to the monitoring and quality of deliverables and make sure that we follow the schedule correctly.

- **Quality**. The quality control of the tasks completed during execution of the project will have a major impact on the end quality of the implementation. Communication and the transfer of information within the team developing the project and in the organisation itself must also be qualitative in order to guarantee the aims of efficiency and efficacy.

In practice, the execution of the stages can be delayed for a number of reasons that may or may not be linked to the project and its management. Examples of these include a lack of synchronisation in the delivery times of the necessary material, the temporary absence of analysts or programmers and the complexity of a development not initially envisaged. Delays usually have financial consequences.

When a delay occurs, two types of decision can be taken:

- Firstly, the delay can be worked into execution of the stage, concluding it and accepting a delay in all stages that depend on it and, as a result, the overall project.

- Secondly, a project delay may be considered unacceptable and more resources are allocated to one or more stages in order to keep up the pace. Nonetheless, the allocation of more resources does not always lead to production improvements proportional to the resources earmarked.

Delays should not generally have a direct effect on stages executed at the same time as the stage that has been delayed. However, it may be worth considering a schedule adjustment to take into account the delay affecting other stages.

For example, if implementation of the system is postponed because of an excessive delay in the reception of materials, we could consider the option of delaying the user training phase until implementation. This would avoid a gap between user training and application of their knowledge of the newly implemented system.

2.1.4. Results

The results of the life cycle of a project should be directly related to the strategic aims of the project and the organisation. Life cycle in itself is merely a methodical and rigorous way of coming up with a solution for a given problem by dividing the inherent complexity of the project into different stages.

To an extent, life cycle is one way of reducing the overall risk of the project. The execution of the stages as successive fine-tuning periods for solving problems contributes to the gradual adaptation and solution of problems that can be highly complex.

We need to consider the importance of the project management team, which plans and coordinates the project to ensure that it is concluded successfully. Management is a dynamic task that must help to reconcile the differences between planning and the reality of the project during its execution.

The results of a systems implementation project are generally grouped into the following areas:

- **Organisation**. For the organisation, the project needs to meet the expectations of the strategic action from which it derives. We need to stress here the qualitative operation of the system, its integration with the organisation's methodology, user adaptation and competitive improvements to the organisation.

- **System**. The system needs to meet all of the aims and expectations of the organisation and its operation needs to meet the aims of the strategic action of the organisation. The system and its interaction with direct and indirect users must meet these aims qualitatively, offering functional efficiency and efficacy.

- **Users**. One of the aims of the system is to provide technological support to the organisation's operation through its users. The importance of the inclusion of users in the implementation project is strategic, since without their participation in the process and their acceptance of the system,

implementation could prove problematic or unfeasible and have financial repercussions.

- **Documentation**. As with any project, documentation is crucial to the quality of the implemented system for its current integration and future evolution. From deliverable documents between stages to the final documentation or user manuals, all of these materials are crucial to the maintenance and support of the system.

- **Support**. The system must have a support team in place from the start of the project, although it is particularly important in the development, implementation and user training stages. The team needs to guarantee interaction and communication between all those involved in the project during and after implementation, acting as a support team for ongoing training or answering questions and solving problems.

In all events, a systems implementation project must allow the organisation and its users to evolve towards new strategic challenges. Creating a climate of confidence and acceptance of change is essential if it is to achieve its aims.

Acceptance of change

This process is usually called **change management** and covers all of the aspects and procedures that enable us to manage and solve any problems and misgivings with the implementation of a new system in the organisation, especially those implemented in free software.

2.2. Study of the current situation

This section will define systems analysis and describe its main characteristics and special features. It will detail the various phases of the study, the main factors influencing its development and the results that the analysis should produce.

See also

For more information about the relationship between the implementation project and the organisation's strategy, see the sections on the organisation's strategic plan and the origin of systems implementation in the first module.

Systems analysis is a chiefly theoretical form of investigation designed to provide a clear and accurate view of the status of the organisation's system within the scope of the project, based on the strategic action from which it is derived.

Systems analysis covers two complementary aspects:

- Part of the analysis involves the technological application of the case study, with a qualitative evaluation of the system from a methodological and procedural perspective.

Case study

Case studies are a scientific method that allows us to explore an object or circumstance in depth through the use of empirical strategies in order to understand the object of study. It is usually used in the initial exploration and in combination with other techniques such as the quantitative approach (statistics related).

- And part of the analysis involves the study of the compliance or competence of the organisation's system, with a quantitative evaluation of the system from a functional and technological perspective.

The theoretical implications of the investigation reveal the importance of a methodical, rigorous and exhaustive approach. Errors in appreciation at this stage can generate problems later on and possibly raise doubts over continuing with the project because of biases affecting the project, the current system and the organisation's strategy, with the ensuing financial repercussions[6].

(6)It is not only necessary to take into account the direct financial costs of time spent on the project, but also all indirect costs, such as the cost of abandoning a project that has been started and the cost of the lost opportunity.

Although this section describes the features of the initial study for a system that has already been implemented, this structure is equally applicable to newly implemented projects if we transfer the object of study to the scope of the organisation, the current and past markets, future technological trends and similar projects begun previously.

This first stage of the project may also bear no direct relationship with free software, since the aim is to analyse and evaluate the implemented system or the current market, regardless of the form of implantation or trend, respectively.

Broadly speaking, we can divide systems analysis into three main phases: identification of the system, preparation or development of the case study and the final evaluation.

2.2.1. Identification of the system

The purpose of identifying the system is to define the object, scope and aims of the study. The definition of these parameters is directly related to the strategic action on which the project is based and must allow us to set up the scenario for evaluation.

It is important to remember that a system already in place is not simply a set of technological parts; it is also a series of features, methods and procedures that have a direct impact on users and the organisation in general.

Hence, the study scope must cover the technological aspects of the implementation, the features currently covered by the system, the procedures and methods of action deriving from its interaction with the organisation's operation and the impact on the use of the direct and indirect users of the system.

We need to determine two key aspects of these parameters:

- Firstly, we must determine the different sources of information that will allow us to obtain data for the subsequent analysis of the system.

- And secondly, we must identify the quantitative or qualitative nature of the data we will obtain from the sources of information, since data extraction techniques vary widely.

See also

The following section on case study development describes the main differences between techniques for obtaining sources of information.

Quantitative and qualitative data

Quantitative data are numerical variables that quantify characteristics or attributes. For example, the number of active users in the system per unit of time.

Qualitative data are variables that differentiate between characteristics or attributes but do not quantify them. For example, the colour combination of an application's user interface.

The result of this phase is a working document indicating the object, scope and aims of the study, along with a list of the data that must be obtained and the source of this information.

2.2.2. Case study development

This phase is used essentially to collate all of the important study data indicated in the system identification phase.

In practice, information can be gathered from a range of sources: historical documents, detailed interviews, results of technological audits, performance counter tools, the documentation of previous projects, technological specifications and even issue reports.

As we collect data, we may come across interesting aspects that were not considered in the system identification phase. In all events, data collection needs to be rigorous and meet the criteria on structure and organisation.

Nonetheless, we can differentiate between two generic groups in data collection:

- **Quantitative data**. This type of data is usually collected directly from technological media. The implemented system may have counters for performance, transactions, capacity, volume, etc., which can generate interesting statistical results, as in the case of time or cost units.

- **Qualitative data**. This type of data is usually collected from written documents, meetings or staff interviews. The procedure for obtaining information from interviews and meetings is crucial here, because they must be painstakingly prepared and conducted if we are to obtain quality information.

It is very important to use a methodical process to obtain both quantitative and qualitative data, since the system is a support tool for staff and the organisation itself. All information is beneficial when it comes to assessing and evaluating the system.

This phase usually marks the beginning of the development of the hardware and software inventory and the current system's network diagram. Besides being useful for determining the current status, it can also be efficient and help with the planning of a possible system migration.

See also

You can find out more about hardware and software inventories and network diagrams in sections 2.7.3 and 2.7.4 of this module.

The end format of the case is usually an investigative report in which all of the aspects dealt with above are organised and evaluated. The report must back up the data and results it describes, relating them to one another and the definition of the project and seeking out possible relationships of dependence or independence.

Depending on the type of information described, it may be convenient to use statistical results, tables, graphs and charts, and generally anything that will help with the description, understanding and evaluation of the data included in the report.

One of the most common tools used for the presentation of executive summaries are SWOT[7] analyses, which present the main conclusions of the study of the current system from a strategic point of view. If the features of the project require, we can produce SWOT tables by classifying the various features of the system according to the result of their evaluation, such as if the current system's hardware is a weakness for carrying out the strategic action successfully.

(7)SWOT is an acronym for Strengths, Weaknesses, Opportunities, and Threats.

2.2.3. Final evaluation

The final evaluation of the system is the first control point of the project and its aim is to determine the feasibility of the current system in the light of the organisation's strategic actions and hence, evaluate the need to continue with the project.

Generally speaking, there are four large groups of features that we need to consider:

- **Operating**. These relate to the functional interaction of the users with the implemented system, ergonomics, performance, efficiency, efficacy and usefulness.

- **Organisational**. These concern the methods and procedures generated by the implemented system, together with their benefits and disadvantages for the organisation.

- **Functional**. These relate to the efficiency and efficacy of the tasks carried out by the implemented system, in addition to scope, reliability, performance and malfunctions.

- **Legal and financial**. These relate to the cost of the implemented system and its legalisation, covering aspects such as maintenance, licensing and system administration.

Evaluation of the system can bring us to three main groups of conclusions:

- **The system is feasible**. The study and evaluation conclude that the current system is ready to take on the strategic actions of the organisation. These conclusions are usually drawn in cases where a study has been conducted to find out the status of a large and/or complex system whose evolution could be difficult to gauge on a superficial level.
  If the current system obtains a positive evaluation, the implementation project must be cancelled because there are no indications that a new implementation is required.

- **The system is partially feasible**. The study and evaluation conclude that the current system needs minor updates before it can take on the strategic actions of the organisation. These changes usually involve updating or changing a small series of elements, such as replacing devices or updating software.
  A partially positive evaluation of the current system requires continuing with the implementation project but it is convenient to revise the aims and scope in order to adapt them to the needs detected.

- **The system is unfeasible**. The study and evaluation conclude that the current system cannot assume the strategic actions of the organisation. This type of conclusion is usually drawn in cases of migration from older systems that are unreliable or perform poorly and hence need to be completely updated.
  A negative evaluation of the current system tells us that we should continue with the project to implement a new system. It may be necessary to revise the aims and scope of the project because replacing the current system may require more resources than those initially envisaged.

Both the report on the analysis and the final evaluation of the current system are submitted to the organisation by the project monitoring committee. The final decision on whether or not to continue with the project is usually made by the management of the organisation.

The result of this stage is twofold:

- Firstly, we obtain a comprehensive report on the current status of the system, highlighting the main features of the system from the point of view of the organisation's strategy.

- And secondly, the organisation makes a decision as to whether or not to continue with the project and any actions required to adapt the system to the latter's strategy.

2.3. Study of the implementation requirements

This section will define the study of systems implementation requirements and describe its main characteristics and special features. It will detail the various phases of the study, the main factors influencing its development and the results that it should produce.

The study of the system requirements is a process requiring a methodological analysis of the problems that need to be solved.

There are two main aims to the requirements study:

- **To define the implementation**. Requirements studies allow us to detail all the features that the system to be implemented must have and permit. To a certain extent, they define the specific, functional aims of implementation.

- **To reduce risks**. Requirements studies also allow us to reduce the risk of the project and its management by specifying and progressively fine-tuning the characteristics of the solution to be implemented.

Requirements studies for systems implementation projects usually require a great deal of effort for two main reasons:

- Firstly, because it can be difficult to specify and methodologically structure the ideas and expectations of users and the managers of the organisation regarding the new system, bearing in mind that user requirements can evolve over time.

- And secondly, because they are crucial to the subsequent development of the project, since errors in appreciation made in this phase and detected in later stages will affect the timing and financial aspects of the project or generate unplanned extra costs that could jeopardise completion of the project[8].

(8)Errors made during the system design stages detected and solved in the development stages have a possible cost ratio of one to ten.

A requirement is a feature that the new system needs to have. In other words, it is an attribute that must allow the system to meet the set aims. Requirements are usually written as text but can take the form of tables and charts if these help to clarify and specify the aim.

In general, we can define four different types of requirement:

- **Strategic policy**. Requirements linked to the organisation's strategic policy cover general aspects of the project, its management, outcome, or the approach to adopt. These include corporate ethics and image or the traditional attributes of the organisation.

- **Methods and procedures**. The implementation of a new system is usually a good time to update and improve the methods and processes of the organisation and/or system. Besides a thorough revision of current procedures, we will also need to bear in mind the specification of future methods and procedures resulting from new aims or features that the new system will need to encompass.

- **Operation of the system**. The operating requirements of the system derive from the interaction of users with the system. We will need to distinguish between functional and non-functional requirements: functional requirements correspond to specific actions that the system will need to execute, whereas non-functional requirements correspond to limitations or restrictions while executing actions, which allow the actions to be linked to the methods and procedures.

- **Key factors and priorities**. Most systems have a specific number of basic elements that form part of the core of the system. In new implementations, we need to give preference or priority to the components considered essential for the operation of the system and the organisation; other components not considered essential can be given a lower priority.

The requirement collection stage is not exclusive to free software because free software systems implementation must meet the same requirements and aims as any other type of implementation.

Broadly speaking, we can divide the study of system requirements into five main phases: identification and definition, specification and structuring, verification, validation and the final evaluation.

2.3.1. Identification and definition

This first phase of the requirements study pinpoints and defines the problems that need to be solved, indicates the project typology and determines the main sources of information for data collection.

See also

See Section 1.2 of the first module to find out about the different types of systems implementation projects.

The requirements study stage uses documentary information from the analysis of the current status stage, so part of the task of pinpointing the problems has already been completed.

We will need to identify the resources and elements from the organisation that are directly or indirectly involved and define the scope of the problems for the organisation and for those resources and elements that are directly and indirectly involved, whether human, material, functional, organisational, procedural or technological. All these elements will usually allow us to recover vital information for defining the new system, which we can use, together with the report on the current status, to establish a corpus of knowledge for making decisions.

The direction in which the market is heading is also important when it comes to defining requirements. Familiarity with the features of similar systems, organisations from the same sector, recent innovations and the future trends of the project theme can all be helpful when it comes to specifying, proposing, evaluating and understanding the requests of users and organisation managers.

The result of this first phase is usually presented in a working document with a detailed definition of the aim and scope of the project, a list of the elements to consider, sources of internal information on the organisation and a list of elements or market trends that may contain relevant information for the project.

2.3.2. Specification and structuring

The aim of the specification and structuring phase is to collect relevant data on all of the elements indicated above and to organise them using methodological criteria to create a reliable corpus of knowledge that accurately represents the reality.

This phase is based on the working document produced in the previous phase, providing an initial outline of the elements and sources of information that may contain relevant information for the project. Nonetheless, the practical development of the study may lead us to consider new sources of information and new aspects of relevance to the project that were not taken into account in the previous phase. Deadlines permitting, we should try to investigate all of the details that may appear.

We can identify two tasks in the development of this phase:

- **Collection and specification**. This task attempts to resolve the functionalities of the system. In other words, what the system has to do, not how it has to do it. Some information sources tend to focus on how actions should be done instead of determining the specific features of the task itself.

- **Structuring and organisation**. This involves organising data methodically and comprehensibly. It may be useful to prioritise certain requirements over others, in accordance with the elements of the system required for its operation.

As with the study of the current status, data collection may respond to quantitative or qualitative criteria:

- **Quantitative data**. These usually come from technological media and can be bulk processed in order to obtain statistical results to verify and justify qualitative data and to model and extrapolate results to new functionalities.

- **Qualitative data**. These are usually taken from interviews and meetings with those involved and provide us with functional and non-functional results regarding the system and the procedures and methods affected by the project.

Systems implementation requirements relate to the basic elements of the whole system, such as the hardware, software, infrastructure, staff, procedures, functionalities and even languages, documentation, formats and standards.

Requirements are usually presented as text, organised according to the characteristics of the project, system and the special features of the requirements themselves. Tables, graphs and charts can also be used to improve the definition, arguments and evaluation of the ideas covered.

The result of this phase usually takes the form of a working document that describes all of the requirements, organised, structured and reasoned in accordance with the ideas of the project. The clarity and precision of this document is crucial for the entire project, since all subsequent development and the organisation's acceptance of the terms of the final implementation both depend on it.

2.3.3. Verification

The requirements verification phase evaluates the requirements contained and described in the previous phases and assesses them in the context of the system's coherence and the aims of the organisation.

This phase uses the working document from the previous phase, which methodically organises the requirements obtained in the study.

The formal verification of the requirements can be split into two main processes:

- **Technological analysis**. The technological analysis of requirements is a process aimed at analysing and concluding that all of the requirements complement one another and form a logical system whose implementation is feasible.
  Antagonistic requirements resulting from the diversity of information sources are usually detected in this phase. This conflict can be resolved by checking the other characteristics covered by the requirements and/or validating the options with those directly involved or with the organisation.

- **Functional analysis**. The functional analysis of the requirements is a process aimed at analysing and concluding that the system deriving from implementation of the requirements meets the requirements and aims of the project and the organisation.
  In this phase, we can detect inaccurate assumptions in the requirements, which can contradict the aims of the project. The conflict can be resolved by a review and consultation of the problematic aspects with those directly involved and/or with the organisation.

Requirements verification is a fundamentally technological and methodological process that analyses the coherence and feasibility of the future implementation and introduction of the system defined in the requirements.

It can be interesting to have a number of people participate in the review of the requirements. Different perspectives can help pinpoint and solve any errors or shortcomings more effectively.

The review can also stimulate the appearance of new issues or situations not previously considered, which may generate a loop between the collection of requirements and their verification until convergence and coherence have been established in the results (a similar process to top-down methodology).

Top-down methodology

Top-down methodology takes the general definition of a problem and creates a loop by specifying and refining each item until a sufficient level of detail is considered to have been reached. The result is usually displayed as a process tree.

The result of this phase is that the working document with the requirements, created in the previous phase, is updated. This updating has allowed a detailed review of the concepts of implementation and the identification and resolution of any errors in the initial requirements.

2.3.4. Validation

The requirements validation phase is designed for reaching an agreement on the requirements of implementation of the new system – following the study – with the organisation. Agreeing on the requirements is crucial to the contractual formalisation of the implementation of the system.

This phase requires the active participation of the organisation, which must conduct a thorough analysis of the requirements proposal resulting from the previous phases. The transfer of information between the two groups is essential because the characteristics of the implementation of the system depend on the understanding of the requirements study.

The internal working document containing the requirements may not be entirely appropriate for presentation to the project's monitoring committee. In this case, it may be necessary to create new documents or presentations to transmit the information in a clearer and more effective way, using charts and diagrams.

The organisation's validation of the requirements may conclude with new revisions of the requirements. These revisions or adjustments usually involve the addition of new functionalities to the system by the organisation that were not initially planned in the project.

Nonetheless, these revisions cannot continue indefinitely because they could have a direct impact on the feasibility of the project and its implementation schedule, not to mention the financial implications of the increase in the number of requirements to implement.

The result of this phase is closely related to the final evaluation phase of the systems requirements study. It usually generates the last revision of the working document with the requirements of the new system, agreed and validated by the organisation and the team that conducted the requirements study.

2.3.5. Final evaluation

The final evaluation of the requirements study is the second control point of the project, the aim of which is to determine the feasibility of the implementation project in the light of the requirements of the new system and hence, the need to continue with the project.

This phase is closely related to the validation of the requirements, since the main working document represents the last definition of the requirements for implementation of the system, agreed on by the organisation. The validation and final evaluation phases may be merged for making decisions regarding the need to continue with the project.

The feasibility of the system requirements is evaluated by taking into account their suitability for the current system, the organisation and the strategic action that launched the project. The requirements study is the first formal step towards the new system and early estimates of the volume and cost of the changes.

In a way, the evaluation of the requirements is similar to the evaluation of the current status that we looked at in the first section, which bases its assessment on financial, technological, functional and legal aspects. We can also evaluate other aspects relating to the strategy of the organisation, such as the latter's capacity to evolve, the intangible benefits of the change, management quality and corporate ethics.

The evaluation of the proposed requirements for the new system can lead to three main groups of conclusions:

- **The project is unfeasible**. The evaluation of the requirements of the new system by the organisation concludes that implementation of the new system is unfeasible and the project is abandoned.
  There may be a number of reasons for reaching this conclusion, some of which could even be outside the scope of the project. This outcome is usually related to economics, financing, competition or the abandoning of the strategy that launched the project.

- **The project is partially feasible**. The evaluation of the requirements of the new system by the organisation concludes that it will be carried out in part, meaning that only certain elements of the new system will be implemented.
  This outcome is usually related to economics and financing. In some cases, partial implementation can be carried out progressively or in stages[9]. In these cases, it is important to introduce guarantees to ensure cohesion between the different implementations over time.

(9)Some organisations prefer to spread the burden of the investment of a new implementation over several accounting years.

- **The project is feasible**. The organisation has made a positive evaluation of the requirements of the new system and the project will continue without major limitations or restrictions that could affect the basic definition of the project.

The result of this stage is twofold:

- Firstly, we obtain a full report on the requirements of the new system to implement, validated by the team and the organisation.

- And secondly, the organisation makes a decision as to whether or not to continue with the project and on the scope of the implementation of the new system.

2.4. Analysis of free software solutions

This section will define the analysis of solutions for the implementation project and describe its main characteristics and special features. It will detail the various phases of the analysis, the factors influencing its development and the results that this stage should generate.

Solutions analysis is a process whereby the various technological options available are analysed methodically and rigorously in line with the project requirements.

This analysis has three complementary aims:

- **To know the market**. The analysis of solutions allows us to study and evaluate the current market situation in terms of the definition, scope and aims of the implementation project. The variety of solutions currently available means that an deep, careful analysis is required.

- **To adapt the solution**. The analysis allows us to select the solutions that best fit the problems of the project and the implementation of the system. Moreover, the open nature of the source code inherent to free software allows for the fine-tuning and final adaptation of the chosen solutions in the form of derivative products.

- **To reduce risks**. The analysis of solutions allows us to reduce the risk of the project and the implementation of the system because the study allows us to adapt the solution and control the main repercussions of its implementation.

With free software, the solutions analysis stage takes place in a scenario limited by two main conditions:

- **Project typology**. The project typology, the characteristics and the scope of the system to be implemented will determine the scope of the search for and analysis of possible solutions, together with the feasibility of the various proposals.

- **Analysis of requirements**. The analysis of system requirements must determine the functional and operating behaviour of the solution in terms of the project aims.

As we could expect based on what has gone before, this stage uses the documents on the definition, aims and scope of the project and on the system requirements agreed with the organisation. The report on the evaluation of the current situation can also be useful here because it indicates the characteristics, special features, and conclusions on the viability of the current system. All of this must help us to focus on the characteristics of the analysis.

This stage focuses mainly on the strength and diversification of the market's current supply of free software. The search for and selection of free solutions that meet the requirements is essential for the implementation project. Hence, our analysis must reflect the competitive attributes of free software in comparison to private solutions, with a special emphasis on freedoms of use and source code adaptation.

Broadly speaking, the free software solutions analysis stage can be divided into three main phases: the search for solutions, the analysis and assessment of candidates, and the final evaluation.

2.4.1. Search for solutions

This first search phase is aimed at identifying solutions whose implementation in the framework of the project is feasible. It is used to make an initial selection of systems with a similar aim to those of the project.

Free software is a valid and viable option for systems implementation since it offers a wide range of freedoms of use, operation, access and modification of source code, and licensing of derivatives.

These features not only allow them to be used freely by organisations and private individuals, they also permit independence from proprietary providers, savings on licences and royalties, learning from the original source code, monitoring of obsolescence with guaranteed product maintenance, quality and reliability, and guaranteed security, privacy, interoperability and software convergence.

Free software implementation can traditionally be divided into two main areas:

- Implementation in infrastructure services[10]
  Since it first emerged, free software has maintained a very close relationship with systems architecture and network services. It is now the undisputed leader of certain sectors[11], ahead of proprietary software.

- Implementation for home users or clients[12].
  As a result of various past initiatives, free software has now embarked on a new path and spread to the environment of the end user, entering into competition with the *de facto* standards of the home environment.

The market now offers a wide variety of free software systems in very diverse fields. Most important projects have their own websites to promote the knowledge, diffusion, downloading of and collaboration with the project[13].

There are also public pools[14] allowing the creation and development of and collaboration on new free software projects, together with the downloading of the resulting applications. These pools often act as a launch pad for community projects.

It should be remembered that there may be no one solution for the implementation project, whether due to scope or the specialised nature of the aims. In these cases, we need to categorise our search for solutions into more generic typologies, so that a number of specialist systems can be put together to form a joint solution to the project requirements[15].

The search for solutions should generate a document with the results of the research that covers the following aspects:

- **Definition and identification of the search for solutions**. Here we indicate the characteristics used to direct our search, explaining their relationship with the project definition, aims and scope, and the definition of the agreed requirements.

- **List of solutions**. The document must contain a list of the solutions we have come across in the search process, briefly defining the solution and its relationship with the project aims for each.

- **Summary of technical features of the solutions**. The summary of technical features of each solution must tell us about the main characteristics of the system, such as the definition of the project, the pools in which it is found, the monitoring and maintenance of the product, the languages it supports, the community collaboration, its end

(10) For instance, local network servers form part of the basic infrastructure services of the organisation.

(11) For example, the implementation of *Apache Web Server* on web servers is superior to that of other environments.

(12) For example, the Ubuntu distribution competes directly with *Microsoft*'s operating systems, debunking many myths about difficulties with the installation, management or use of GNU/Linux environments.

(13) Examples include the Ubuntu distribution (www.ubuntu.com), the office suite OpenOffice.org (www.openoffice.org) and the browsing suite Mozilla (www.mozilla.org).

(14) For example, SourceForge.net (sourceforge.net).

(15) For example, if we are looking for a solution for the management of a web database, one option might be to search for a separate operating system, web server, database manager and programming tool that can work together to offer the required functionalities. This is the case of the LAMP environment (*Linux*, *Apache*, *MySQL* and *PHP*), which is currently very popular.

licence and other requirements specific to the product, such as ergonomics of use, execution requirements and the main functionalities.

If there is no single solution for the problems of the project and we need to split it into different individual solutions, the document can be organised into categories by typology (for example, operating systems or office suites) or by function (such as a database server with a web interface for access and programming).

A SWOT table is also sometimes included for each of the options found. This table is very important in meetings and executive decision-making, and its aim is to identify and summarise the main characteristics and specific advantages and disadvantages of adoption of these options.

2.4.2. Analysis and assessment of candidates

The aim of the analysis and assessment of candidates phase is to identify the solutions most suited to the project and the organisation and to the special features of the development, implementation and migration.

This phase uses the document from the previous phase, which lists the most suitable current solutions for the project and details their main characteristics. The purpose is to methodically and carefully select the best candidates for implementation from among the diverse alternatives identified.

A series of technological parameters directly related to the project, organisation and implementation process are usually analysed, considered and evaluated. The individual assessments produce an organised classification that will determine how well suited each solution is to the implementation project.

We can generally use the following parameters to assess and evaluate a solution:

- **Project and organisation**. We must assess its adaptation to the project aims, the organisation, the definition of methods and procedures, the ethics and traditional standards of the organisation and the strategic action that launched the project.

- **System and interoperability**. Here we need to assess how well it meets the needs of the system and confirm its guaranteed operation with current or future hardware, software and network resources, the feasibility of the methods and procedures, implementation of the strategic action, interoperability with existing systems and standards in the organisation and the possibilities for future expansion and evolution.

- **Functionality and ergonomics**. We need to assess how well it meets the functional and operating needs of the project and the organisation, and confirm the ergonomics of its use and implementation of the methods and procedures with guaranteed operation and that it will meet the expectations of the users and the organisation.

- **Efficiency and performance**. We need to assess whether it can guarantee efficient operation at all times, the use and performance of allocated resources and maintenance of a suitable balance between resources and performance over time, to allow evolution.

- **Efficacy and reliability**. We need to assess whether it guarantees performance of the functions set down in the project aims, maintains functional compliance with the aims at all times, preserves the balance between resources and compliance over time and that allows evolution.

- **Implementation and migration**. We need to consider how well suited it is to the system implementation process and confirm that it allows efficient and effective migration from an earlier system, minimises the consequences of impact on the everyday operation of the organisation and that it guarantees tools for the support, training and adaptation of users and any other systems currently in place.

- **Maintenance and management**. We must assess whether it will equip the system with management and configuration tools adapted to the project aims, minimise physical and logical maintenance operations, guarantee a balance between operation and maintenance over time, and allow evolution.

- **Support and commitment**. We must evaluate whether the support and monitoring of its creators for the system and its users are guaranteed, and whether it contributes to the community commitment to evolution, future improvements, problem solving and generally all aspects that could cause the system to become obsolete.

- **Licences**. We must evaluate whether it guarantees compliance with the legislation in force, establishes a scenario for use and operation, allows a clear and effective distinction to be made concerning actions that can be performed with the system and specifies the licences for any products derived from it.

- **Economics**. We need to evaluate how it adapts to the economy of the organisation, the planned project financing, the costs of management, maintenance and its efficient and effective operation throughout the expected life of the system and the costs of training and educating the users of the system.

As we can see from the above list, the parameters seek to assess candidates in different areas (technological, strategic, operating, financial, etc.), in order to detail, analyse and evaluate the impact of implementation on the organisation.

The result of this phase is a document that numerically classifies and considers the various candidates (possibly by category), organising the solutions according to how well they adapt to the project and the organisation.

Given that different solutions could obtain the same technical score, it may be necessary to conduct another study on these alternatives that, despite differing in their characteristics, have similar degrees of adaptation to the project. In these cases, it may be useful to draw up a SWOT table for each solution in order to reveal the differences that could help us to make our final choice.

2.4.3. Final evaluation

The final evaluation of the analysis of free software solutions is the third control point of the project and its purpose is to determine the development of the implementation project, i.e. the form of the project and its final implementation.

This phase uses the documents from the previous phase, which classify the existing solutions based on how suited they are to the project, and the summaries of technical features and/or SWOT tables analysing them in detail.

This phase has two main aims:

- **Final selection of candidates**. Although the previous phase indirectly proposes the most appropriate solutions for the project and implementation (by scoring and classification), it may also be necessary to evaluate the overall integration of all of the solutions.

- **Development of the project**. The choice of the solutions meeting the project requirements will have direct consequences on its development, such as its cost in terms of time and financial resources.

Remember that the components of the system will not be separate from one another, so the best solutions will be those that generate a stable and operative system besides meeting the requirements of the project.

In some cases, it may be useful to carry out integration tests to determine the quality and performance of the cohesion of the diverse elements, since the integration of solutions that have obtained high scores in their respective categories does not necessarily guarantee an overall performance of the same level[16].

(16)For example, if the connector between the two systems is inefficient or limited due to compatibility with other systems.

The choice of solutions will generate adaptations in the project, particularly during the development phase. Overall, we can consider three different types of implementation:

- **Direct implementation**. Solutions most suited to the needs of the project and the organisation can be implemented directly, either because they are of general use or because the project aims meet the specific needs of a large group.
  Nonetheless, differences between the requirements of the project and the functionalities of the solution do not automatically rule out adoption.

Solutions for direct implementation

Examples of direct implementation solutions include operating systems, office automation packages, e-mail clients or web browsers.

- **Evolution of the solution**. The solutions most suited to the requirements of the project and the organisation can only reveal significant differences at certain points. Their adoption requires an evolution of the source code to adapt the possible shortcomings of the original solution in the event of problems with the system.
  The freedoms allowed by free software make it a valid and viable option in practice, while the proprietary software alternative may require a completely new development, with the ensuing financial implications.

- **New development**. There are no solutions that adapt sufficiently to the problems of the project and the organisation, so a completely new development is required to solve them.
  These cases may occur in systems with very specialised requirements, such as robotics. In exceptional cases, proprietary solutions may not exist in the usual marketing channels.
  In all events, the freedom for studying, analysing and reusing the source code of free software allows us to improve the quality, efficiency and efficacy of the new development and cut associated costs.

The evaluation of free software solutions can lead us to three main types of conclusion:

- **Feasible direct implementation project**. The project is feasible and the implementation requirements can be met by existing solutions. Any differences between the solutions to be implemented and the problems in the project are not insurmountable[17].

(17)For example, MS Word can be replaced with OpenOffice.org and functional differences can be dealt with by user training.

- **Feasible project subject to conditions**. The project is feasible but the implementation requires the use of a process to create or adapt solutions.

Time and money implications will be dictated by the scope of the process, which will be at its largest when a completely new development is required.

- **The project is unfeasible**. Although it is not usual to abandon the project at this stage, the organisation can sometimes decide not to continue with it. For example, if financing for a specific development is not available or if the organisation changes strategy.

The result of this stage is twofold:

- On the one hand, we obtain free software solutions adapted to the requirements of the project and the organisation.

- And on the other, we obtain a preliminary evaluation of the costs involved at the development stage of the project (direct implementation or the need for further development).

2.5. Formalisation of the proposal

This section will define the formalisation of the project implementation proposal and describe its main characteristics and special features. It will detail the various phases of the process, the main factors influencing its development and the results that this stage should generate.

The formalisation of the proposal is the stage of the project in which we detail, specify and link all of the results obtained in the early methodological phases of the project – called *design stages* – with the aim of putting forward a formal solution to the problems associated with implementing systems in the organisation.

This stage has three specific aims:

- **To characterise the project**. Formalisation of the proposal collates, clarifies and lists all of the results from the initial design stages of systems implementation. The result of this stage should serve as a guide for the subsequent development of the implementation, hence it is a key part of the project.

- **To obtain validation from the organisation**. The formalisation of the project also serves to present the implementation project to the organisation, which must then evaluate it. The resulting evaluation will determine the project's future, viability, development and execution.

- **To reduce risks**. Formalising the proposal allows us to reduce the risk of the project and implementation of the system, and delimit the scenario of the project's evolution by detailing solutions and the development of the implementation.

This stage is related to the functional management of the project because it allows us to detail aspects of project planning such as management of time, resources and project costs using the information obtained from the results of the previous stages.

Overall, the proposal needs to cover two main aspects:

- **Methodological aspects**. Methodological aspects refer to the results of the analysis and systems implementation design stages that we have seen in this unit. Specifically, these are analysis of the current system, analysis of requirements and analysis of free software solutions. The aim here is to detail the characteristics of the system to be implemented using a rigorous and methodical process.

- **Management aspects**. Management aspects cover the special features of the operation and execution of the project, such as scheduling, financial planning or others relating to the human resources needed to conclude the project. The aim is to detail the project execution parameters in order to obtain a clear vision of the requirements needed to meet the objectives.

See also

See these and other aspects of project management in the "Management of free software projects" section of the first unit.

From the preceding paragraphs, we can conclude that communication and collaboration with the organisation is important if we are to convey all of the information regarding the project correctly. By and large, we can identify two main aspects of the information that need to be conveyed:

- **The project**. Understanding of the project, the results obtained from the analysis and the solution most suited to the project characteristics. The relationship between the project proposals and the strategic action of the organisation must be clear.

- **Free software**. Understanding of the use of free software, its philosophical basis and the features of its licences, together with the special features of the solutions proposed and the benefits obtained by the organisation.

Broadly speaking, we can divide the formalisation of the proposal into four main phases: drafting, design, presentation and the final evaluation.

2.5.1. Drafting of the proposal

This first phase of the formalisation of the proposal is designed to recover all of the information that can be incorporated into the proposed solution. Here, we should obtain the documents resulting from the various stages of study, analysis and project planning.

One of the main aims of this first phase is to generate advanced results from the initial data. In other words, we should create summaries, diagrams, tables, charts and graphs to clarify the concepts in the technical documents drawn up throughout the project.

Another result of this phase is the updating of the project planning using the information contained in diverse analytical documents. The most important aspects for the organisation are usually:

- **Solution to be implemented**. The definition of the solution to be implemented, the aims and scope of the project, the strategic problems solved and the systems to be implemented are all important.

- **Time costs**. Time costs are related to the scheduling of resources and of implementation or migration itself.

- **Financial costs**. Financial costs are related to the financial repercussions of implementing the system in the organisation (materials, resources, time, training, etc.) and the project management (project management team). Remember that you will need to take into account the financial costs of the project from the outset[18].

See also

See the "Time management" section for more information about the time costs of the project.

See also

See the "Cost management" section for more information on the financial costs of the project.

It may also be useful to prepare the following aspects:

- Seek out classical misconceptions about free software, particularly concerning the proposed solutions, and provide arguments for and against. The aim of this exercise is to provide a working basis for justifying and establishing the grounds for the proposal in the event of a possible reticence to change[19] observed in its presentation.

- Obtain and present information on free software licence typologies in an attempt to prove certain assumptions wrong and to transmit the free philosophy and its relationship with the proposed solution in the right way.

- Draw up SWOT tables to represent and justify situations where it has been necessary to make decisions based on the study and evaluation of different

(18) As mentioned in the "Resources of a systems implementation project" section, the project starts when it is allocated resources for the first time.

(19) FUD (*fear*, *uncertainty* and *doubt*).

options. In these cases, it may be useful to evaluate and consider each of the alternatives presented.

This phase generates a series of documents with results on two main aspects:

- Firstly, a set of references to technical analysis documents from the design stages and diverse advanced results and information relating to the project and free software.

- And secondly, an update to the project management planning, which includes all aspects deriving from the analysis and design stages, shaping continuance of the project.

2.5.2. Design of the proposal

The design phase of the proposal is used to draw up all documents and presentations that must be submitted to the organisation and which summarise the work carried out in the study and analysis stages of the systems implementation project.

This phase uses the documents from the previous phase, with references to technical documents, information on the project and free software, advanced results and updates to scheduling and financial planning. The aim is to structure, organise and present all available information so that it can serve as a guide for the subsequent development of the project.

By and large, we can generate two main types of document:

- **Reports and schedules**: these are documents presenting the results of a deep study. Their aim is to present the contents in a structured and methodical fashion using accuracy, order and rigour so that they can be used as a guide or handbook for the project.

- **Presentations**: these are documents that reveal the general lines of the study. Their aim is to describe the main concepts of the study in an orderly, synthetic and graphical way, so that they can be used as the summary of the project.

Reports should only be presented after editing and formatting and they should adopt a formal and precise language. Reports normally contain the following:

- **Executive summary**: this is a brief summary that positions the systems implementation project and its contents in terms of time, space, its aims and its solutions.

- **Introduction**: this describes the strategic position of the organisation and the need to carry out a systems implementation project in specific circumstances.

- **Analysis of the current situation**: this should include a summary of the study, an analysis and the conclusions on the organisation's current system in relation to its strategy.

- **Definition, aims and scope**: this describes the systems implementation project, the general concepts of the proposed solution, justification of the use of free software and its suitability for the organisation's strategy.

- **Architecture of the system, infrastructure and technology**: this describes the general and functional architecture of the system, the infrastructure it requires, the technology and standards it uses, and the end licences. Integration tests for the different elements may also be included.

- **Human and material resources**: here we describe the human and material resources needed for the everyday operation of the system in the organisation.

- **Implementation and maintenance**: this describes the implementation methodology used, the features of migration, user training, adaptation of the organisation and the necessary maintenance of the installation over time (usually over a period of five years).

- **Scheduling**: this details the monitoring and relationship between the stages of the project over time, taking into account the resources allocated at any one time. It usually takes the form of a chart in which the stages are plotted against the time axis.

- **Financial planning**: this describes the assessment of the cost of conducting the project, breaking down the cost of implementation into various items (e.g. cost of project management, necessary human resources, material expenses, software development and adaptation, infrastructure installation, etc.).

- **Synergies with the free software community and other projects and bodies**: depending on the characteristics of the organisation, we can include a section describing the relationship between the solutions used and the free philosophy sector, the use of similar projects and relations with other bodies or with the free software community.

- **Conclusions:** we need to highlight the project features that meet the strategic needs of the organisation and the benefits obtained from the use of free software.

- **Technical appendices**: these can come in useful for the subsequent development of the project. For example, for the analysis of the system requirements.

We obtain two documents from this phase:

- The report or project plan, which is presented as technical documentation of the project.
- The presentation, which will be used at the executive meeting in the subsequent phase.

2.5.3. Presentation of the proposal

In this phase, we present the proposal to the organisation, usually during an executive meeting with its management body. The purpose is to inform the organisation of the final results of the analytical stages so that the organisation can evaluate the proposed solution and the monitoring of the project, and consider its continuity.

This phase is usually implemented by presenting all of the aspects of the project in the form of charts and summaries, using the presentation document from the previous phase. In addition to the presentation, we must deliver the project plan with the results of the design stages.

It is particularly important in this phase to focus on the communication and transfer of information, highlighting the relationship between the proposal and the strategic aims of the organisation. Though not the norm, we may be asked to reformulate some aspects of the proposal, which will, in any case, need to be fairly minor in quantitative and qualitative scope.

The committee in charge of validating the implementation of the system may be reticent to the use of free software, so it may be wise to justify the proposal by questioning certain preconceived ideas, explaining the advantages and disadvantages of its use and even discussing the free philosophy in general and licence models. The document drawn up in the design phase of the proposal on issues related to the project and free software may prove useful here.

This phase is closely related to the subsequent phase, the final evaluation, since decisions on whether or not to continue with the project are often made at presentation meetings.

2.5.4. Final evaluation

The final evaluation phase of the formalisation of the proposal is the fourth control point of the project, designed to evaluate the work carried out in the design stages of implementation and the proposed project solution. It must also resolve the feasibility and continuity of the project for the development and system implementation stages.

This phase is closely related to the previous one because the presentation of the project plan and exchange of information between the organisation and the team that came up with the design is essential if we are to take into account all of the implications of the proposed solution. The proposal presentation and validation phases may be merged for making decisions regarding the continuity of the project.

The importance of this control point is twofold:

- Firstly, because the formalisation of a proposal closes the theoretical and conceptual cycle of the analysis and design of the system and its associated implementation project. The subsequent stages will be mainly practical.

- And secondly, because the organisation will finally determine whether or not the project will be concluded and meet the aims of the strategic action, implementing the series of changes considered in the proposal.

Hence, as explained in the previous sections, the organisation must evaluate the aspects of the project it considers to suit its strategy; for example, the characteristics of the solution to be implemented and the financial costs of the project and implementation of the system.

It might also be a good time to analyse, evaluate and adapt certain specific details of the implementation of the system in the organisation, considering and defining certain aspects of change management, implementation scheduling, pilot tests and user training.

The organisation's evaluation of the proposal can end with one of three main types of decision:

- **Feasible project**. The organisation accepts the proposed solution without making any major changes to its specifications. Any adaptations relate to the implementation schedule or the inclusion of certain financial items, for example.

- **Feasible project subject to conditions**. The organisation considers certain adaptations to the proposed solution to be necessary. These changes may or may not be minor, but part of the proposal will have to be drawn up again if the project is to be given the go-ahead. The changes may be related

to the tools implemented or to the partial execution of the project, for example.

- **Project unfeasible**. The organisation considers the project unfeasible and abandons the implementation of the system. Although this does not usually occur at this stage in the project, a drastic change in organisation strategy or the inability to secure financing to conclude the project are possible reasons for abandoning implementation.

The result of this stage is twofold:

- Firstly, we obtain the project plan, which is agreed and validated by the organisation.
- And secondly, the organisation makes a decision as to whether or not to continue with the project implementation.

2.6. Development

This section will define the development stage of the implementation project and describe its main characteristics and special features. It will detail the various phases of the process, the main factors influencing its development and the results that this stage should generate.

The development of the system is the stage in which the solutions described in the project plan are implemented, together with all of the requirements for its start-up. The purpose is primarily to adapt, collect and produce all of the necessary elements in order to perform implementation with the utmost guarantees of efficiency and efficacy, and to minimise intervention time.

This stage uses documents from the previous stage, mainly the project plan and analysis of the system requirements to be implemented. These documents are essential for the preparation, structuring and organisation of the development of all components needed to implement the project.

This stage has two specific aims:

- **To implement the solution**. The purpose of development is to implement the solution set out in the project plan and the study of requirements, regardless of the form of the development type.

- **To reduce risks**. Development also attempts to reduce the overall risk of the project by preparing, specifying and building the solution without directly affecting the operation of the organisation.

See also

The "Requirements Verification" section of this module describes three project typologies: direct implementation, the evolution of existing tools and the development of a new solution.

In general, development needs to cover two main aspects:

- **Methodological aspects**. Methodological aspects relate to the project typology, so the development stage is directly related to the purpose of implementation. In other words, the development methodology to be applied will depend on the aim of the project.

See also

For more information on project typologies based on aims, see Section 1.2.2 "Classification by project aim".

- **Management aspects**. Management aspects concern the people involved in the development, i.e. those who implement the tasks set down in the project plan. Broadly speaking, we can have internal development (insourcing) or external development (outsourcing).

See also

For more details about internal and external development, see the "Resources of a systems implementation project" section of the first unit.

In general, we can divide the development stage into three main phases: the allocation of resources, the configuration or development of the software and the final evaluation, although the individual existence of these stages will depend largely on the project typology.

The allocation of resources and software development phases can be executed at the same time, depending on the project type. It is also possible for these two phases to overlap rather than being simultaneous, perhaps because of differences in the delivery dates of the various resources.

We also need to remember that the development stage is closely related to the subsequent stage: implementation or system migration. In this case, some tasks may overlap or be carried out at the same time in order to cut short the project schedule, though this could have financial consequences that need to be taken into account, caused by aspects such as the need for more human and material resources.

2.6.1. Allocation of resources

The aim of this first allocation of resources phase is to select and obtain all of the necessary resources to implement the system in the organisation, in order to obtain all of the elements directly required for the implementation and operation of the new system.

One of the main aims of this first phase is to equip the organisation with the necessary infrastructure for system start-up, i.e. all of the material resources, organisations and configurations that will allow the new system to become operative[20]. Therefore, if the project includes the total or partial migration of the system, it may be useful to link this phase to the migration planning phase.

(20) Examples include installing a local area network, enlisting a local technician or purchasing switches, computers and servers.

See also

To find out more about migration planning, see the sections on "Migration strategies", "Hardware and software inventories" and "Network and structure diagrams".

Material resources are usually allocated after evaluating the characteristics of provider proposals based on a set of specific conditions. This set of conditions must be created using the system requirements to ensure that it adapts and integrates with the rest of the project.

Evaluation of the diverse proposals is specific to the project, system and organisation. Broadly speaking, we can assess the following:

- **System and interoperability**: we need to evaluate each element alone and its adaptation and integration with the system, project and organisation.

- **Functionality and ergonomics**: we must assess the ease of handling and configuration of each element and the knowledge required for its fine-tuning and everyday use.

- **Efficiency and performance**: we need to assess functional and operating performance in line with the needs of the system.

- **Efficacy and reliability**: we must evaluate the reliability and coverage of the aims in relation to the system requirements.

- **Implementation and migration**: we should evaluate the ease of introducing the element into the organisation and the tools it offers for migration from the current solution.

- **Maintenance, support and guarantee**: we must evaluate the operation and maintenance needs of the element, and the support and guarantees offered by providers.

- **Economics**: we need to evaluate the cost of the element in terms of its promised results and the advantages and disadvantages of the impact on the organisation's operation.

Implementation of the project may also require the allocation of human resources to the organisation, for either direct or indirect system handling. Since the project stems from a strategic action within the organisation, a change in methods and procedures may require a restructuring of the staff involved.

The biggest changes in this regard would take place in new organisations or existing ones that do not use computer equipment in their day-to-day operation. The demand for technological profiles goes hand in hand with the level of technological implementation, the type of project and the characteristics of the organisation.

In all events, the selection of human resources will have to be integrated into the typical recruitment methodology of the organisation but we should stress the need for the candidate profile to correspond to the technological requirements of the organisation.

2.6.2. Software configuration and/or development

The aim of the software development phase is to implement all adaptations required by the free software solution in order to adapt it to the project plan. This phase can also include the development of tools to help with data or file conversion for systems migration.

The main aim of this phase is to ensure that the free software solution matches the project requirements, with the adaptation and encoding of all necessary changes, taking advantage of the open nature of the source code and the freedom to develop it.

The length of this phase will depend directly on the changes that need to be made to the proposal. Generally, we can distinguish between three different cases:

- **Direct implementation**. When the software is to be implemented directly, the development scenario is limited to adaptation of the configuration files or software parameters.
  For example, the operating system configuration (language, network access, etc.) or the parameters of office automation tools (templates, language, etc.). This case requires minimal investments in time and financial resources in comparison to the other two.

- **Software evolution**. Software evolution encompasses the extension or adaptation of the source code of one or more free software solutions to adapt its operation to that of the organisation.
  The importance of the results means that we must proceed with the usual rigour of software engineering, establishing an adequate life cycle for each modification; it could also include software re-engineering.
  One example might be the introduction of additional calculations in accounting management software or the modification of the downloading manager of a web browser.

- **New development**. New code is developed when no solutions can be found to meet the specific needs of the organisation. Nonetheless, it can still prove useful to study and reuse the source code of open applications to save time and ensure reliability.
  New developments are the most laborious in terms of time and cost. This means that we must proceed using software engineering methodologies adapted to the project. For example, the creation of industrial design or architectural or electronic device control software.

See also

For more information on free software production, see the subject on "Software engineering in free software environments".

Hence, any software development project will be guided by its specific life cycle and will establish the relevant control and quality milestones, while evaluating monitoring and results using the most appropriate methodology.

Regardless of the development format, in this phase, it is useful to generate the documentation for the software adaptation process with a list and description of all details subject to modifications, using the same rigour as for new developments. In these circumstances, we need to take into account the end licence for the evolving solution, since the freedom of the source code is usually inherited from the original solution.

It may also be useful in this phase to develop the materials for training the system users and to adapt the support manuals of the free software solutions, since we will now have all of the necessary elements to perform this activity. In these cases, we also need to take into account the characteristics of the licences for the manuals we wish to edit.

Support manuals

Frequently asked questions (*FAQs*) or the standard manuals for completing specific tasks (*How to...*) which, although highly technical, can still be very useful.

2.6.3. Final evaluation

The final evaluation phase of the development stage is the fifth control point of the project and its aim is to ensure that the development meets the requirements of the project plan. It is also designed to reduce the overall risk of the project by validating the viability and adaptation of development to the organisation's strategy.

The evaluation of this stage usually depends on the typology of the development:

- **Resources, installations and infrastructures**: these are evaluated according to whether or not they meet the aims of the design and the provider's proposal. Delivery deadlines are very important here as they have repercussions on the scheduling of other stages.

- **Direct software implementation**: we must evaluate whether or not the aims are met by the proposed adaptation of the configuration and operation. The main evaluation is based on operating tests.

- **Software evolution or development**: the complexity of the evaluation is proportional to the scope of the changes made. The evaluation resulting from the software engineering process is normally taken into account.

This phase is closely related to the implementation and migration of the system since, depending on the project type, we can implement tailored solutions to conclude the development. It can also be related to user training and pilot tests because it can be useful to find out user opinions in order to fine-tune certain aspects of the development[21].

(21) For example, the graphical interface, system response or the characteristics of the procedures that it implements.

The importance of this control point is twofold:

- Firstly, because it revises, evaluates and assesses the qualitative development of the solutions and their adaptation to the strategic requirements of the organisation.

- And secondly, because it supposes the completion of the process of creating and adapting the solution, and the start of the definitive implementation and integration of the system in the organisation.

The development evaluation phase is generally a good time to introduce users to the features of the system, and the situation can be regarded as the start of the training and adaptation of users to the new environment. These actions can take place as part of the management of the change that the organisation needs to carry out in order to overcome any misgivings or fears among users[22].

(22)FUD (*fear, uncertainty and doubt*).

The final evaluation of the development stage can end in one of three main types of decision:

- **System implementation is feasible**. The evaluation determines that the system meets the requirements of the organisation and that all modifications of the original solution have been developed and integrated correctly. Implementation of the system in the organisation may begin.

- **System implementation is partially feasible**. The evaluation determines that the system partially meets the project requirements on the control date. This situation is normally due to production delays, which can occur as a result of unexpected events, such as the lack of material resources, long delivery delays or the lack of human resources for production.
  In all events, the feasibility of the project is not affected, but we will need to take into account the fact that implementation will be delayed.

- **System implementation is unfeasible**. Although it is rare for the development of a solution to be unfeasible, there may be exceptional factors affecting the decision that could not be solved during this stage. This type of problem is normally associated with factors outside the scope of the project, such as sudden changes in strategy or a lack of funds.
  Abandoning the project at this stage will have serious financial, functional and moral implications for the organisation and will make it difficult to take any further action in the future.

The result of this stage is twofold:

- Firstly, we obtain the system to be implemented (tested and integrated), which will meet the project requirements and the strategic needs of the organisation.

- And secondly, we will have assessed the feasibility of implementation and made any adjustments to scheduling as a result of the evaluation of pilot tests.

2.7. Implementation and migration

This section details the technical aspects of migration to free software systems and their implementation.

Most implementation projects are also migration projects because they start off with a scenario in which there is a computer system based on proprietary software already in production. Implementation from scratch takes place in new organisations that look to free software solutions to start up their first system or in organisations that currently have no computer system, which is rarely the case.

Implementation from scratch is always much more straightforward than migration in terms of the technical problems that the project will need to solve, mainly because there will not be any compatibility problems with existing systems. Nonetheless, we do need to bear in mind that the users of the new system will usually be familiar with proprietary operating systems and applications, so planning the training is just as important in migration projects.

This section will refer exclusively to migration projects because they are the most difficult and because projects implemented from scratch can be considered a subset of migration projects with the particular feature that they allow greater freedom to decide on the final scenario. This freedom comes with the condition that more attention must be paid to non-functional aspects of the system, such as correct dimensioning.

We will begin by describing the different types of migration project and the most crucial aspects of the planning. We will then offer advice and guidance for executing the migration and evaluating its results. And finally, we will conduct a detailed study of all the services involved in the migration project and the most popular free software solutions for each.

2.7.1. Types of migration

We can carry out a range of possible types of migration to free software systems in organisations: aims-based, or depending on the elements of the system that will be migrated; and scope-based, or according to the number of elements that will be migrated.

Aims-based:

- **Migration of services and servers**: this affects the servers of the organisation, the applications they run and the services they carry out, such as authentication or printing services, among others. In this case, there is no change to the clients' applications, so we only need to plan for system administrator training and not for end users. These are among the easiest migrations to carry out. The servers, which use GNU/Linux, operating systems from the BSD[23] family or other free systems tend to be more reliable and offer better performance, which will increase the productivity of the organisation, both by systems administrators and end users (lower server response time).

(23)OpenBSD, FreeBSD or NetBSD.

See also

The "Network and structure diagrams" section details the migration features for each of these services.

- **Migration of users and clients**: this affects the client machines of users and the applications running on them. In this case, we will need to train and accompany the end users, who are generally less receptive to the use of new applications and will be most affected by the change, which could cause a temporary loss of productivity.

- **Migration of applications**: this only affects some of the applications running on client machines or servers, the operating system of which does not necessarily have to be GNU/Linux or any other free operating system. More often than not, the operating systems continue to be proprietary. It is sometimes a preliminary step towards migration of the operating system. These are fairly straightforward migrations, such as those to OpenOffice.org or MySQL Community Server.

Scope-based:

- **Complete migration**: this is the result of combining the migration both of servers and client machines. This type of migration must be planned in such a way as to ensure that clients are not left without access to the services provided by the servers at any time. Hence, the first step is usually the total or partial migration of servers, followed by the migration of client machines.

- **Partial migration**: this is the result of a combination of the partial migration of the servers or part of the clients, which means that there will still be machines based on proprietary software in the system. One common scenario is that where a single system contains clients based on free software and clients based on proprietary software, whose configuration will depend on the needs or preferences of the end users.

- **Migration based on virtualisation**: this can be viewed as a type of partial migration where we migrate servers and client machines at the same time as we continue to install and run proprietary software applications that could not be included in the migration, whether because there are no free

software equivalents or for other reasons. With virtualisation, we can start up a proprietary operating system on a free operating system and use it as normal with proprietary software applications.

We also need to remember that while the typical migration scenario is one where we switch from a proprietary operating system to GNU/Linux, there are other possible combinations, such as:

- From a proprietary operating system to a free operating system, such as those of the BSD family.
- From a free operating system to another free operating system.

The OpenBSD operating system

The OpenBSD operating system is renowned for the quality of its security mechanisms and its integrated cryptography, making it ideal for servers or other machines whose integrity could be compromised.

2.7.2. Migration strategies

As with any project, correct planning is essential if we are to migrate successfully to a free software system. There are as many ways to plan as there are projects and they will all be valid to an extent if they meet the requirements and features of our migration scenario. Nonetheless, depending on our migration planning, we can extrapolate four main migration strategies:

- **Single-step migration**: this involves carrying out the entire migration process in a short space of time, in a single day or during public holidays if possible. The strategy requires identifying and defining all of the tasks that need to be performed, since a mistake in any of these could leave the entire system inoperable, with the subsequent risk of delays and user rejection. This is the most economical strategy and is usually used for small systems with few clients and a single server, as is the case of small companies.

- **Pilot migration**: the first step is to migrate a small part of the system, which will be used for practising and evaluating the success of the migration before we implement the rest of the system. The pilot system usually consists of a number of servers and client machines, but it can also be a single server and a single client machine. Although it is very important to plan correctly, this strategy offers greater flexibility for modifying the migration approach and correcting possible problems. The drawback to the strategy is that it requires far more resources and is therefore usually used in organisations with medium-sized or large systems.

- **Group migration**: this involves defining user groups based on their functional characteristics and carrying out migration progressively with each of these groups. One of the main advantages of this strategy is that it enables us to minimise risks and learn from each migration. Moreover, since migration only affects part of the system, we can reduce productivity losses. The disadvantage is that we often need to keep the previous systems

running while we set up the free software system. It is usually a good time to renew hardware when carrying out group migration, and vice versa.

- **User migration**: this is similar to group migration, the difference being that only one user is migrated each time. As a result, the strategy requires very few resources but is unfeasible in large and medium-sized organisations, where migration would take too long due to the high number of users. It could, however, prove useful for the migration of critical systems and users that require special monitoring.

These strategies are not mutually exclusive and several of them can be used in any one project. For example, in an organisation that has some smaller or less important departments, single-step migration can be carried out in these, while pilot migration can be carried out in others before moving on to complete implementation. Similarly, group migration can be seen as a combination of single-step and pilot migration.

2.7.3. Hardware and software inventories

In order to plan migration, we need to identify the hardware and software available in the initial situation that we wish to maintain after migration. As a result, we must carry out a detailed inventory of both hardware and software.

The hardware inventory should include all machines available for migration including those that have been withdrawn, since we may be able to use some of these again.

Hardware can be grouped into the following categories:

- **Hardware with full GNU/Linux support**: this includes hardware with out-of-the-box support in the Linux kernel or free drivers, and hardware for which we need to use proprietary drivers, either directly or using adaptors. Most hardware has good support on GNU/Linux, so it will fall into this category. We will describe each of these situations in detail later on.

- **Hardware with limited GNU/Linux support**: this includes hardware that only works with older versions of the Linux kernel not used in the more recent distributions of GNU/Linux, hardware that runs with very old drivers whose maintenance has expired, and hardware with free drivers that have functional limitations in comparison to the proprietary drivers[24].

(24) For example, graphics adapters with 3D acceleration whose free drivers are only available in 2D.

GNU/Linux distribution

The production of a GNU/Linux distribution involves checking that all the packages included in the distribution are compatible with one another and, most importantly,

with the kernel version. Thus, there will always be a gap of some months between the date of the appearance of the distribution and that of its kernel, which is older.

- **Hardware with no GNU/Linux support**: this includes hardware that does not work on GNU/Linux. In actual fact, there is very little hardware without GNU/Linux support but when this is the case, it is either because the hardware is very new and the relevant drivers have not yet been developed, or because the hardware is very old and incompatible with the newer versions of the Linux kernel or because the hardware is dependent on a specific operating system and cannot therefore be used with GNU/Linux[25].

(25) In this case, however, a virtualisation solution could still be possible.

We can also classify hardware with GNU/Linux support by the type of support, as follows:

- **Hardware with out-of-the-box GNU/Linux support**: most equipment and devices have adequate support in the recent GNU/Linux distributions so there is no need for external drivers. It is easy to find lists of hardware with GNU/Linux support on the Internet.

Website

You can find out which hardware has GNU/Linux support at: http://hardware4linux.info/.

- **Hardware with free driver support**: although they do not have out-of-the-box GNU/Linux support, many devices work correctly with drivers maintained by the free software community. The package managers included with the GNU/Linux distributions often propose the installation of these drivers when they detect the hardware.

Website

NDISwrapper is a free software project enabling the use of wireless network cards with GNU/Linux through the use of Windows drivers.

- **Hardware with proprietary driver support**: the free software community does not maintain drivers for this type of device so we will need to use proprietary drivers, often supplied by the manufacturer. This is usually the case of hardware with very specific functionalities, such as graphics accelerators. Nonetheless, support for this type of device is gradually increasing and the manufacturers themselves often open up their drivers.

- **Hardware with adaptor support**: this hardware is supported by proprietary operating systems but not by GNU/Linux. Fortunately, tools called adaptors are available so that we can use the drivers of these proprietary operating systems in GNU/Linux.

Once we have classified the hardware correctly, we will obtain a clear idea of the available resources and the need to purchase any new equipment. Moreover, as we mentioned earlier, the hardware requirements of GNU/Linux systems are considerably lower than those of proprietary operating systems, so obsolete machines can be reused for services such as printing or e-mail.

Once we have completed the hardware inventory, we will need to do the same with the software. We will have to identify all of the proprietary software applications used in the system before migration and the best free software applications for replacing them.

There are many lists on the Internet indicating equivalent proprietary and free applications. However, a detailed study of the functionalities of each application is sometimes required in order to select the best candidate from the free software options.

Website

The SourcePYME project publishes a fairly extensive and up-to-date list of applications and services (http://www.sourcepyme.org/?q=node/13). There are also many resources available in English.

There are many options for the more common applications, such as office automation, but there is often one application or package of applications that stands out above the rest. There are communities of developers and users for applications with more specific uses and it can often be a good idea to ask their advice or even get involved.

See also

The "Evaluation of migration" section contains the available alternatives for migrating the core services of an organisation.

If there are no free software applications or projects that meet the organisation's needs, it may consider developing a new application and opening it up with a free licence, provided that it has the resources to do so. The benefits are clear: the potential contribution of external developers and users and increased visibility for the organisation.

2.7.4. Network and structure diagrams

This section describes two essential elements of systems migration: the network diagram, which illustrates the connectivity between the diverse elements of the system, and the structural diagram, which indicates its physical location.

Thus, once we know which hardware and software will be affected by migration, we can represent the system on a network diagram. This diagram must contain the following elements:

Website

An excellent free software application for creating network diagrams as well as other types of diagram, is Dia (http://live.gnome.org/Dia).

- **Servers**. Indicate the name of the equipment, together with the main services offered by each server.

- **Client or user equipment**. Indicate the name of the equipment and the network devices exposed to the other systems.

- **Printers**. Indicate the name of the printer and the print server or client computer on which it depends.

- **Other network equipment**. Indicate the main equipment forming the system network enabling connectivity between the various machines. For example, hubs, routers, switches and wireless *access points*.

- **Connectivity between elements**. Indicate the wired and wireless network connections between the various elements of the system. Indicate the organisation's local network connection to external networks such as the Internet, virtual private networks (VPN) and virtual organisations (VO).

It is very important for each machine to be uniquely identified in the network diagram.

First of all, we need to draw up a network diagram illustrating the status of the system before migration. This diagram will be used to study the possibilities for optimising the current network, such as alleviating server bottlenecks or connecting certain local printers to a central print server. We will also decide which new equipment and network elements will be introduced in the system as new servers, which old equipment can work with GNU/Linux or discuss the implementation of a wireless network.

With these elements we can create a network diagram for the system after migration. This diagram will be essential for defining the planning and strategy of the migration and will serve as a guide during implementation. Thus, it is a good idea to keep the network diagram up to date, ensuring that it offers a true picture of the system status.

As explained at the beginning of the section, the structural diagram reflects the physical location of the equipment inside the organisation; in other words, it tells us what equipment there is in each room.

As we did with the network diagram, we will create a structural diagram to illustrate the status of the system before migration, which can then be used to decide on the location of the equipment after migration.

One of the most common consequences of migration is the introduction of servers for a small number of services, sometimes even exclusively. As a result, the servers are usually grouped together in the same physical location (usually a server room) that meets the specific requirements for climate control, power supply and accessibility, among others. Another example is the connection of printers (thus far local) to a print server, which can be located in the same room.

Although structural diagrams are not very important in small organisations with just ten or so machines, in bigger migration scenarios, it is essential to know the location of each piece of equipment and network component.

2.7.5. Execution of migration

When carrying out any migration, regardless of the strategy adopted, there is a series of technical tasks that almost always crop up again and again: equipment installation, data migration, production of backup files, emulation of applications, etc. Besides these technical tasks, it is important to have a management plan in place to deal with any risks that may appear during migration.

This section will look briefly at each of these tasks and offer guidelines for their completion:

- **Equipment installation**. Automatic equipment installation tools are available for easy installation and configuration of more than one in a short space of time.

One such tool is SystemImager, which automates the installation of clones of the GNU/Linux system installed on an initial piece of equipment. SystemImager also allows us to distribute new applications or data on the system equipment and make changes to the configuration or install system updates on networks with GNU/Linux equipment. However, if the hardware of the equipment is not identical, they may need to be manually configured.

Website

You can find out more about SystemImager at http://wiki.systemimager.org/.

SystemImager

SystemImager allows you to save a clone of a GNU/Linux system in production before making changes to the system, which means that you can revert to the original situation if need be.

- **Migration of user data**.

The names and addresses of users are usually stored in directory services, normally accessible through the standard LDAP protocol, which facilitates the migration of these data at system level.

However, in the case of the applications that use these data, such as e-mail clients or groupware, different data schema are often used to structure the information. As a result, the data are rarely interoperable and external programmes must be used to synchronise the data between applications.

See also

See the section on "Hardware and software inventories" for more details on the migration of directory services.

- **Making backups**.

Backup is the term generally used to refer to the copying of data to allow a system to be restored after loss of information. The implementation of GNU/Linux systems often involves the formatting and partitioning of the equipment involved in the migration, so we need to make backups of the existing data in order to restore it later on the new system.

If the organisation has an up-to-date backup mechanism, one option is to use this to recover all the information that needs to be copied to the new equipment.

If the organisation has no backup mechanism, we can set up a storage service exclusively for the storage of the data that needs to be migrated. Another option is to implement the storage service envisaged in the project plan first and then provide access to the system users so that they can store their data before the user equipment is migrated. In both cases, the participation of users is fundamental.

Once migration is complete, we will need to set up an incremental backup mechanism[26]. As a general rule, the original system and the backup must be as independent as possible so that an error in one will not affect the other.

(26)There are many solutions available, including RSync (http://samba.anu.edu.au/rsync) and Amanda (http://amanda.org).

- **Emulation of applications and virtualisation**.

Performance of the software inventory will serve to determine which applications cannot run natively on GNU/Linux and cannot be replaced by an equivalent free application. If these applications are essential and we need to continue using them, we have two possible solutions: emulation or virtualisation.

Wine, the most popular free solution

Wine is the most popular free solution for running native Windows applications on a GNU/Linux system. Although Wine (http://www.winehq.org/) is usually referred to as an emulator, it is more correct to say that Wine provides a layer of compatibility for Windows applications. Wine is in fact an acronym of Wine Is Not an Emulator.

Wine does not need to install a Windows partition but it can be useful to have some native Windows libraries in some cases. Applications that run with Wine can access the file system, network and printing services in a completely transparent way. The Wine website contains information on the supported applications and their level of functionality.

For applications that do not run correctly with Wine, it is possible to run them on a virtualised operating system. As explained earlier, virtualisation allows us to run one operating system over another. In this case, we would run the application on a virtual Windows system on a GNU/Linux system. The most popular free virtualisation solutions are QEMU, Xen and KVM. In all events, virtualisation should always be considered as a last resort because we will have to carry on using and paying for proprietary licences and because it is a big consumer of system resources.

- **Risk management**.

Migrating to a free software system is not without risks, so it is important to draw up a management plan and keep to it for the duration of the project.

See also

See the section on "Risk management" for more details on how to draft a risk management plan.

The risks and their relative importance will depend on the migration scenario and the features of the organisation. For example, some organisations may consider it a priority to guarantee the security and integrity of certain confidential data, so this contingency will need to be provided for and a plan drawn up to solve it in the event that it does occur.

As a general rule, we suggest that you keep the migration process reversible until you have fully verified the new system, i.e. that you will be able to return to the starting point in the unlikely event that migration fails or proves unfeasible.

2.7.6. Evaluation of the migration

In any migration project, it is essential to evaluate both the end system and the migration process. This evaluation can be done once we have completed the migration, but it can also be carried out during the process, if it is not conceived as a single-step process.

Hence, the project plan must include a series of clear indicators. These indicators may include some of the following, which refer to the operating system, servers, applications and users:

- **System indicators**. Have the reliability, performance and security of the system increased since migration? How has the real (as opposed to estimated) cost of system maintenance changed? Have new services been introduced into the system? How do the administrators see the new system? Has the number of problems with system services fallen since migration?

- **Operating system indicators**. How many machines have been migrated to the new system? Does all of the equipment work properly? Is all of the hardware supported by the new system? How often have virtualisation solutions been required?

- **Application indicators**. For how many existing applications has an equivalent application in free software been found and implemented? What functionalities have been gained and lost with regard to the original applications? How many applications run through emulation or virtualisation? How many applications was it necessary to modify? How many applications have had to be developed from scratch?

- **User indicators**. How many users have been migrated to the new system? What is their opinion of the functional and non-functional aspects of the new system and the new applications? How has their productivity varied

in the short and long term? Has the number of user problems fallen since the new operating system was installed?

2.7.7. Migration of the services of a system

Most organisations have a series of basic services that must be paid special attention when planning and executing migration:

- File system
- Printing service
- Directory and authentication services
- Network service
- System management and administration
- Web servers
- Databases
- Desktop environments and office automation applications
- Corporate applications

This section will describe the main characteristics of these services and indicate the most popular free software solutions. There is generally more than one alternative so the final choice will depend on the characteristics and requirements of each scenario.

Clarification

Although most of the free software solutions described in this section are in the mature stages of development and used in many scenarios, technology evolves constantly, so it is a good idea to visit the websites of the projects to obtain more recent technical information and research other solutions that could improve on existing ones.

The importance of each of these services will also vary according to the characteristics of the organisation. Some of these services may not exist in the initial situation and will not therefore be included in the migration.

Nonetheless, migration is an excellent opportunity for analysing and revising the current status of the system and designing an architecture that meets both the current and long-term needs of the organisation. Hence, we need to consider the inclusion of new services not present in the original system.

File system

We can be faced with two situations when migrating the file system, depending on whether all or only a number of the clients are being migrated:

Migration of the storage system server but not the client server

In this case, the most popular solution is Samba, a free protocol implementation used in Microsoft Windows shared file systems for Unix systems that allows computers with GNU/Linux to act as servers or clients in Windows networks.

Migration of both the storage system and client servers

In this case, we will usually use NFS or OpenAFS.

NFS allows us to access remote files in the same network as if they were local files. NFS is included in the GNU/Linux operating system by default. Similarly, OpenAFS[27] is a distributed file system generally used in clusters and distributed computing scenarios.

(27) OpenAFS is a free implementation of a file system originally developed by Carnegie Mellon University that also influenced the design of NFS.

The choice of one or the other (or the choice of another system) will depend on the migration requirements. It is possible to use NFS or OpenAFS in networks that include Windows and GNU/Linux clients.

For the migration of servers operating with GNU/Linux, there are several file systems but the most widely known are Ext3 and XFS. Their functionalities include journaling, assignment of quotas and access privileges based on ACL (Access Control List) by file and directory.

When migrating file systems, we need to pay attention to the mapping of the Windows ACLs to Posix ACLs, since we can lose granularity here. In practice, this does not usually occur because organisations usually do not make full use of the granularity permitted by Windows ACLs.

Printing service

Printing is one of the most common sources of problems in organisations, generally because printers are installed with no planning, which leads to numerous several technical issues and, very often, a waste of resources (paper, ink, electricity). Migration to a free software system is actually a good opportunity to optimise the printing service.

Of the available free software solutions, CUPS is the print server used by most GNU/Linux distributions and is in fact the best option for almost all migration scenarios. One of its main advantages is that it provides us with a print service both for GNU/Linux clients and Windows clients, because it implements *Internet Printing Protocol* or (IPP).

IPP is a printing standard for both LAN and WAN networks that supports communication between clients and servers, between different servers and between the selected server and printer. It is supported by all modern printers.

Before carrying out migration, we need to check the support and drivers for each printer.

Directory and authentication services

The purpose of a directory service is to ensure that certain information is available to all network users. This information is usually composed of objects organised hierarchically, originating from a root object. The most common access protocol is LDAP.

LDAP (*Lightweight Directory Access Protocol*)

LDAP initially referred only to the access protocol but it has come to mean the combination of the database containing the information and the protocol for accessing it.

For example, a fairly common use for a directory service is to store system user accounts together with their privileges so that all the system applications and services can access it to obtain this type of information, which should be complete given its nature.

Thus, a directory service needs to offer the following functionalities:

- The available information must be modified and organised into a hierarchical structure.
- Use of a standard data schema to ensure compatibility and interoperability with as many applications as possible.
- User authentication and guaranteed interoperability with other authentication services.
- Administration of access rights to the information in the directory service.
- Secure transfer of information between clients and the directory service.

For authentication with a directory service, the most common free software solution is a combination of OpenLDAP and Samba, where the latter serves as a database of user accounts and OpenLDAP acts as a directory service. There are many applications compatible with LDAP, including the office automation package OpenOffice.org.

The GNU/Linux system offers various LDAP tools[28] for modifying the information stored in the directory service and there are also web-based graphical interfaces available for the administration of users and groups.

(28) These include ldapsearch, ldapad and ldapmodify.

A directory and authentication service based on OpenLDAP and Samba will also allow the simultaneous use of Windows and Linux clients. In fact, OpenLDAP also acts as part of the authentication service and as an integration tool in mixed scenarios with GNU/Linux and Windows clients. If we are carrying out a complete migration to GNU/Linux, authentication is also possible with Kerberos. Kerberos is an authentication protocol that allows two computers to reveal their identities to one another securely.

Network services

The entire TCP/IP network infrastructure (DNS, DHCP, NTP, router connection, filtering, VPN) can easily be implemented with free software solutions, due mainly to the fact that all Internet protocols are open standards, both in their definition and in their implementations.

Note

For more information about open standards, see Appendix II of this module.

One aspect to consider when migrating network services is the use of open standards, even when they are not needed (as in the case of small local networks) as this does away with the need for specific modifications from hardware manufacturers, which can eventually cause compatibility problems with other systems when new services are implemented and even a dependency on the manufacturer.

The network services include the following:

- **DNS (*domain name system*)**
  The free software implementation of reference is BIND (Berkeley Internet Name Domain), currently maintained by the Internet Systems Consortium (ISC). BIND is the most popular DNS server on the Internet. Its latest version is BIND 9, which includes DNSSEC (DNS Security Extensions), TSIG (Transaction Signature), DNS notification, *nsupdate* and *Ipv6*, among other functionalities. It is available on all GNU/Linux systems.

Internet Systems Consortium

The Internet Systems Consortium is the non-profit organisation that succeeded the Internet Software Consortium, also known as ISC.

- **DHCP (*dynamic host configuration protocol*)**
  The implementation of reference in free software is `dhcpd` , also now maintained by the ISC. `dhcpd` allows the administration of individual clients and group configurations for classes and subnets. Moreover, `dhcpd` offers load balance functionalities and high availability. It is available on all GNU/Linux systems.

- **NTP (*network time protocol*)**
  NTP is an Internet protocol for synchronising the clocks of computer systems through packet routing, thus avoiding the problems caused by variable network latency. The NTP Project provides NTP support and offers an implementation of reference, available on all GNU/Linux systems.

- **WINS (*Windows Internet Name Service*)**
  WINS allows us to resolve the names of the different Windows services and systems. This function can be replaced by `nmbd`, included in the Samba package.

System management and administration

Most system control and management applications are not native to the operating system and manufacturers often supply versions of these applications for different operating systems. The downside to this is that while there are many systems management applications for GNU/Linux, they are not based on free software.

In all events, the management and control of free software systems is very different to that of systems based on proprietary software, such as Windows. Administrators of free software systems normally use a series of management tools rather than just one, each specialising in a part of the system. Thus, administrators have much more freedom to make adjustments and correct problems with their systems, which is one of the reasons for the well-known reliability and security associated with free software.

An initial option for the automation of administration tasks in GNU/Linux is the use of `cron` and `at`. The former (`cron`) is an administrator of background processes that runs programmes at regular intervals. The `at` command also allows programmes to be run at specific times.

All GNU/Linux systems offer the basic functionality of administration from a remote terminal `ssh` on another client or server in exactly the same way as if it were a local machine, even through the graphical interface of the desktop. The combined use of `ssh`, `cron` and `at` covers many of the maintenance tasks of the administrator.

Other system utilities such as `strace`, `lsof` and `netstat` offer diverse functionalities for detecting and analysing errors, and can be useful in server management.

Network management

The solutions available for the management of TCP/IP networks as free software include Nagios and OpenNMS.

Nagios allows us to monitor servers and services in order to detect network problems in systems based on GNU/Linux. A background process controls the specified services and servers and sends the information to the Nagios server, which informs the system administrator if it detects a problem. By means of a series of plugins, Nagios can actively and passively monitor typical network services such as web and mail servers, besides others such as database management systems.

OpenNMS is a network management application that uses the FCAPS model and allows us to determine the availability of the different services, store the information and generate reports, and inform of events.

However, the management of more complex systems and networks may require the use of tools that are not available as free software.

Software management

Software management involves client installation and restoring, standard and specific applications distributions and the management of updates and patches for the entire system.

The solutions available as free software include `m23`, a software package for systems based on the Debian distribution that allows initial client installation, including the definition of partitions and the detection of hardware, the distribution and updating of software and the restoring of clients.

Web server

Apache is the main alternative for migrating and implementing an organisation's web server. Apache is present in over 60% of web servers and is distributed freely under the Apache licence.

The Apache project

Apache is one of the most successful projects developed by the free software community that, due to certain features of its licence, can be used in proprietary software products.

Its functionalities and performance are excellent and have been thoroughly tested in a wide range of production scenarios. Apache has a modular architecture consisting of a kernel that contains the basic functionalities of the service and numerous easy-to-install modules for specific applications, such as support for certain programming languages (PHP, Java, Perl, etc).

The migration of web projects[29] to an Apache web server requires a study of the individual features of each project, which can sometimes create incompatibilities. Apache offers trouble-free support of both static content (developed in HTML) and dynamic content (developed in languages such as PHP or Perl). The modifications needed to ensure compatibility of these projects in Apache are minimal or non-existent.

(29) The term web project refers to a website (for example, the organisation's website) and to web applications that can be accessed through a browser.

Projects developed in proprietary technologies such as ASP are a special case, since considerable effort is required for them to work on Apache. Wherever possible, it is preferable to implement the web project again in alternative technologies such as PHP to ensure technological independence in the future. This obviously involves more work but the opportunity can be used to optimise the web applications and contents of the organisation.

It is now increasingly common to see PHP used as a web programming language and LAMP platforms (Linux, Apache, MySQL and PHP) have become popular over recent years for offering web contents and applications.

Databases

There are many free software alternatives for the implementation of database management systems, but the most common are MySQL, PostgreSQL, Firebird and MaxDB. Choosing one solution over another will depend on the requirements of migration.

In all events, free databases are mature products that have been tried and tested in production environments and can in fact be regarded as one of the leading areas for free software and the GNU/Linux system in business environments. These solutions also have versions for proprietary operating systems so they could be used when migrating applications only.

Some proprietary databases such as Oracle[30] also have a GNU/Linux version, so in special cases where it is not advisable to migrate the database management system, we could migrate the operating system to GNU/Linux.

(30)Oracle is usually used in fairly complex environments with a series of requirements that cannot always be met by free solutions.

Most databases have fairly standard administration and query mechanisms, which theoretically encourages interoperability and the use of other solutions along with easy migration of data from one management system to another. This means that applications can continue to access the data transparently and without the need for further modifications.

Thus, database migration involves two operations:

- **Migration of data to the new database**. The effort required for this operation will depend on the initial status of the data. If the data can be accessed by SQL queries, an export or data transfer operation with subsequent importing to the new database should suffice. If the data are stored in a proprietary format or even as text files, we will need to implement a parser and subsequently import them to the new database.

- **Verification of data access from applications**. If the applications use a standard data reading mechanism (such as SQL queries), access will have to be set up in the same way, unless we are using commands that do not satisfy the standard. If the applications use standard drivers such as ODBC or JDBC, or a proprietary interface, it will be necessary to replace the original database driver with that of the new database or to implement a new interface. In both cases, this step could require considerable effort and cause problems with interoperability.

As a general rule, for easier database migration, we need to avoid as far as possible the use of predefined query procedures and specific manufacturer extensions for accessing the data from the applications. Instead, it is advisable to use standard drivers like ODBC and JDBC, which are easily interchangeable, and to implement SQL queries in the most modular way possible and separate from the rest of the programme.

Desktop environments and office automation applications

There are two main desktop environment options in GNU/Linux systems: GNOME and KDE.

Both GNOME and KDE provide an intuitive desktop environment with a user-friendly windows manager suitable for all users and a development platform for building applications integrated into the rest of the desktop and between one another.

The choice of one or the other is largely a matter of personal preference. KDE is generally more similar to the Windows interface and has more customisation possibilities, but this could introduce further difficulties for new users.

Although there are many office automation applications for GNU/Linux systems that offer good integration with the Gnome and KDE desktop systems, there are two solutions that stand out above the rest: OpenOffice.org and StarOffice.

OpenOffice.org is an open source, free software, office automation package that is freely distributed. It is available for many free and proprietary platforms, so it is often cited as an example of application migration. In most cases, it is compatible with Microsoft Office and supports the ISO OpenDocument data exchange standard, which can be used freely.

OpenOffice.org is actually based on the StarOffice project. StarOffice is the proprietary office automation package of Sun Microsystems, sold as such, with some additional functionalities[31] not available in OpenOffice.org, which Sun Microsystems continues to support.

(31) Similar *TrueType* fonts to those used by Microsoft, extra templates and picture galleries, additional updates and patches, among others.

OpenOffice.org and StarOffice contain different applications, each with specific function, but they integrate perfectly with one another:

- Word processor (Writer)
- Spreadsheets (Calc)
- Presentations (Impress)
- Mathematic formula editor (Math)
- Drawing (Draw)
- Databases (Database)

OpenOffice.org[32] uses an XML-based compressed file format for all of its applications, which differs from the binary formats used by proprietary office automation applications. With this format, we can easily separate the contents of the file from its data, styles, version control and the pictures included in the document. In addition, OpenOffice.org allows us to work with other formats also based on XML.

(32) Most of the OpenOffice.org features mentioned in this section are also applicable to StarOffice.

Migration of files in Microsoft Office format

OpenOffice.org has mechanisms to convert and import files in proprietary formats, such as those used by the Microsoft Office suite. It also allows us to save files created with OpenOffice.org in proprietary formats.

However, this compatibility is not perfect and while the quality is usually acceptable in most cases, there are sometimes differences in the format of the documents, particularly those containing complex elements, such as macros or other special features. In this case, it may be necessary to re-edit some of the documents if we require the format to be identical to the original.

Hence, before converting and migrating the documents, we need to study their features and classify them according to their use and technical complexity:

- Editable documents: these need to be converted to a new interoperable format such as ODT, so that they can be edited in the future.

- Read-only documents: these can be converted to PDF format, which simplifies the migration process considerably.

- Basic documents: these do not contain macros, proprietary graphics, formats or complex elements or styles such as footnotes, tables and indices. They can be easily migrated by batch processing.

- Complex documents: these contain macros, proprietary graphics and vector graphics, OLE objects, active objects, cross references, etc. They can be migrated but will most likely require individually processing.

OpenOffice.org offers the possibility of converting several documents by batch processing. All of the documents must be located in a source directory and we will need to specify a target directory to which all of the converted documents will be saved. In all events, we recommend checking the conversion with a representative sample taken from all the documents.

In addition, we have two possibilities when it comes to dealing with complex documents:

- Convert the documents one by one so that we can correct any differences with the original document before saving it in the new format.

- Revise the documents one by one to eliminate elements that could affect the conversion process and then batch process them all.

Corporate applications

The term corporate applications is used to refer to applications developed to meet the specific needs of the company or organisation in which migration is being performed.

For migration purposes, we can distinguish between the following types of corporate application:

- Applications that can be run problem-free on a free operating system, such as multi-platform applications (those written in Java, for example) or web-based applications (in PHP for instance, as we saw in the section on web servers).

- Applications requiring slight modifications to be able to run on a free operating system, such as those for correct database access for instance, as we saw in the section on this topic, or for configuring new environment variables.

- Applications that can run by emulation or virtualisation.

- Applications that cannot run on a free operating system, such as applications implemented in languages exclusively for the proprietary operating system.

Most corporate applications are proprietary so the company will not have access to the source code. If it is not possible to run the application by emulation or virtualisation, the best option is to implement the application again as free software based, if possible, on an existing free software project.

2.8. User training, communication and support

Thus far, we have looked mainly at the technical aspects of the implementation of free software systems. The importance of the technology should not detract from the fact that one of the factors in the success of any implementation project, particularly migration, is acceptance of the new system by its users.

This section will first describe the key elements of the free software training plan of an organisation and some good practices to introduce and encourage user acceptance. To conclude, we will take a look at the main channels of communication and key elements of a user support system.

2.8.1. Training

Correct user training plays a very important role in the success of the project. As a result, we should include it in our project plans from the very start.

To plan the training, we need to identify first of all which user groups will use which specific types of application. This will enable us to study the differences between the proprietary and free applications, and hence evaluate the difficulty in the adoption of the new applications for users. These elements will allow us to plan user training to meet their real and individual needs.

Some free applications, such as browsers, e-mail clients and office automation applications, are very similar to their proprietary equivalents. Obviously, less training will be needed in these cases.

Materials

There are many materials on the Internet that can be used to train users or to help us prepare our own materials.

Manuals and documentation are often only available in English, which can be a problem for some users, and the interfaces of certain applications are not translated or the translation is incomplete. In this case, we could think about publishing our own documentation to overcome language barriers.

The European SELF project has a platform for creating and sharing educational materials relating to free software and open standards.

Training manager

The training may be held in the organisation, in collaboration with an outside company or using an online learning platform.

In any case, it is important to facilitate access to training and materials as much as possible. Attendance of training activities may or may not be compulsory as this will depend on the organisation's policy.

Online learning platforms have the advantage that users can adapt their learning process and training access to their needs. Moodle is a free course management system used to create what are known as **online learning communities** in which students can follow the training and communicate with one another.

Another good option is to combine on-site training with an online learning system.

Finally, we should not rule out the possibility of offering users some sort of incentive to encourage them to attend the training, for example, awarding certificates of attendance and achievement.

Types of user

Not all users are alike. Firstly, there will always be some users who are more receptive to the new software than others. In most cases, however, once users overcome their initial misgivings about the use of free software, they find it very similar to proprietary software and are satisfied with its use, so we should not worry too much if their first experiences are negative.

Secondly, remember that technical and non-technical staff will need a different type of training and monitoring.

Technical staff will need more intensive training than normal users, particularly if they have no previous experience of free software and are used to working with a proprietary system that, in contrast, they are very knowledgeable of. The participation of technical staff is also key to ensuring the smooth running of the system after implementation. It is good practice to motivate them and get them involved in the implementation process so that they can get to know the system while it is being implemented.

2.8.2. Introduction to free software

In addition to training, another of the practices that can lead to a successful free software migration project is the gradual introduction of the new applications and services. This gives users time to get used to the new environment and means that they are not confronted by a completely unfamiliar system.

Installation of bridge applications

Many popular desktop applications are now available on GNU/Linux as well as proprietary systems, such as the OpenOffice.org office automation package, the Firefox browser or the Thunderbird e-mail client. Many server applications and services can also be run on both systems, such as the MySQL database manager or the Apache web server.

These types of application are called *bridge applications* and can be very useful in the early stages of migration[33] and for evaluating the response of users and estimating their training needs much more accurately.

(33)This would be a simple applications migration, like the one we saw in the section on types of migration.

Staggered migration of services

The primary aim of any migration is to produce a smooth transition from one system to another without the users noticing any major differences or, if possible, without them noticing any differences at all. One strategy for achieving this aim is to start migration on the servers, so that users can continue working as normal until the system is prepared for the migration of clients.

Services that can easily be migrated at the start include network services (DNS, DHCP, etc.), web servers and database servers. It may be necessary to use technological solutions that work well on heterogeneous systems, such as OpenLDAP in combination with Samba.

This gives us enough time to train technical staff too, whose support can be very useful for migrating clients and providing support to other users.

Both the introduction of bridge applications and the staggered migration of services should be taken into account in the project planning.

2.8.3. Project communication

As we have seen, the implementation of a free software system and migration to this system is a process that involves all of the organisation's users, not only the technical staff responsible for its complete introduction and maintenance.

The mechanisms must be put in place for effective communication between users and the technical and administrative managers of the organisation, and we must ensure transparency throughout the process. As of result, the communication activities must be set down in the project plan, which must include:

- **Initial group communication to all users**. We should use informative meetings, notes, internal e-mails and advertisements on the organisation's intranet to explain the reasons for the project and detail its general planning before it is implemented.

- **Regular communication of the project's progress**. We should explain which parts of the system will be migrated and when, along with any changes made to the project. Meetings with a small number of users should be organised at each phase of migration.

- **Regular meetings after conclusion of the project**. We should evaluate the success of the project and conduct general monitoring of its results and the experiences of system users.

2.8.4. User support system

A key element of the new system is the introduction of an issue management system for users, allowing them to find the answers to their questions and solve the technical problems caused by the situation. It is important to give a fast and efficient response to all these problems, particularly just after implementation.

When designing a user support system, we need to answer the following questions as they will define the main features of the system:

- Who are the users?
- How does the organisation operate?
- What type of support do the users need?
- What type of support is offered to each type of user?
- How much support will be offered?
- How will the support be offered?

Pilot tests can be used as a basis for characterising most of the problems encountered by users and for preparing a procedure for solving each one.

Similarly, we will need to identify the critical services and users of the system, as these will be given preferential support.

We also need to bear in mind that more support means higher costs. It is possible to provide more support in the weeks immediately after migration, when the number of queries and issues is higher. In all events, the key to an efficient user support system is fluid communication with users, to make them aware that their problem is being looked into.

Lastly, it may be the case that a user support system was in place before migration existed. If this system was based on proprietary software, we will need to evaluate the diverse free alternatives.

Website

There are many solutions. For a comparison of some, see http://en.wikipedia.org/wiki/Comparison_of_ticket-tracking_systems.

3. Free software companies

This third unit of the "Implementation of free software systems" subject outlines the main concepts and characteristics associated with the business side of free software.

Ever since it emerged, free software has always been present in information technologies and its evolution has been influenced by the structural changes that have taken place in technology, economics and society.

See also

To find out more about the history of free software, see point two of the materials for the "Introduction to free software" subject.

As time has gone by, a number of philosophies on the creation, production and diffusion of software have been developed. We can generally distinguish between two basic and antagonistic trends:

- On the one hand, the proprietary philosophy, which defends the protection of software by closing and privatising the source code, combined with the use of licences with heavy usage restrictions.

- And on the other, the free philosophy, which defends the freedom of the software and source code with licences guaranteeing user rights to run the programme, study and adapt the source code and redistribute and publish any improvements made to it.

These two philosophies have generated business models with conflicting ideologies, operation, development and economics:

- The privative software model normally establishes a financial value that has to be met by restricting the use of a binary format copy, which makes it impossible for people or organisations without copyright or the explicit authorisation of the copyright owners to adapt, correct or improve the source code. Many proprietary licences prohibit the transfer of user rights to third parties without the prior agreement of the copyright owners.

- The free software model tends to focus on the development and adaptation of free and qualitative solutions to meet the needs of users and organisations, and on complementary services for their implementation and day-to-day operation. Hence, the business model based on free software allows actions that are prohibited or restricted in the privative software model.

The unique philosophical understanding of free software does not only have a direct impact on the business model and business strategy, but also on the definition, management, organisation and operation of technology companies. Aspects such as the maturity of free software, the presence of a

world community of free software project collaborators and the viability of its business models seriously question the traditional concept of a business project.

The first section of this unit defines the different business models based on free software, which are valid and viable to be put into practice as a business strategy.

The second section focuses entirely on the drafting of the business project and details the main aspects of the creation, organisation, production and operation of a free software company.

The third section introduces free software production and describes the main features of the creation, organisation, communication and development of the source code.

Lastly, this unit has two appendices that briefly and systematically describe the main free licences and open standards directly related to the free software business.

3.1. Business models

This first section introduces the main free software business models, together with the characteristics and features setting them apart from business lines based on proprietary software.

By and large, the biggest difference between free and privative software from a business perspective is the licence. Broadly speaking, a licence is a contractual model by which the author of the product (or whoever owns the copyright) establishes the rights and duties of the users of the product and the scenario in which it can be used.

However, free licences[34] are based on four basic principles of freedom that relate to the software and its source code:

- The freedom to run the program for any purpose.
- The freedom to study the source code and adapt it to one's own needs. Hence, access to the source code is necessary.
- The freedom to redistribute copies of the software.
- The freedom to improve the software and release improvements. Hence, access to the source code is necessary.

(34)There is a debate between the Free Software Foundation (http://www.fsf.org) and the Open Source Initiative (http://www.opensource.org) over the implications of the terms *free* and *open*.

Internet resource

You will find the original definition of free software at http://www.gnu.org/philosophy/free-sw.html

The basic freedoms of free software conflict with the privative model focusing on the sales of licences for restricted use of the binary format[35], although free software does not necessarily have to be obtained without payment. However, much of the free software currently available can be obtained by direct, free download from the Internet site of the organisation that manages it.

(35) This is known as right-to-use licensing.

Direct or free downloads

Examples of direct or free downloads include:

- Debian GNU/Linux from http://www.debian.org/distrib/
- FreeBSD from http://www.freebsd.org/where.html
- KOffice from http://www.koffice.org/download/
- OpenOffice.org from http://download.openoffice.org/

The philosophical opening up of free software encourages business models based on human capital, knowledge, customisation and the adaptation of products, not to mention the constant evolution of the software. Hence, we need to highlight the role played by the community of free software users, which helps monitor the quality and evolution of free applications with a level of performance that would be difficult to match in proprietary models.

Over time, the free software model has managed to consolidate an offer that covers most sectors with a privative software presence, shaping a mature, qualitative and secure market on which to base a business strategy covering both software development and complementary services.

Internet resources

At http://freshmeat.net/ and http://sourceforge.net/, you will discover a wide range of free software projects in the key areas where the technology is used.

To an extent, the free software business strategy bases itself on the aspects that set it apart from the proprietary model, such as increased functionalities, tailored adaptation, numerous and continuous revisions, guaranteed product security and quality of operation, and a whole range of complementary services for its implementation and day-to-day operation.

The following sections introduce the main business models deriving from the conceptual philosophy of free software: development, consulting, installation and integration, migration, maintenance, and support and training.

The business models described should be seen as complementing one another rather than being independent. In other words, a combination of one or more business models may be necessary to cover the business strategy.

3.1.1. Development

The software development business model involves the total or partial production of a product based on free software in order to be marketed either directly or as part of a third-party implementation project, such as those described in the second unit of this subject.

The definition of free software makes no reference to the strategy of selling a free product at a price per copy sold but the characteristics of free licences make this a secondary option, albeit one used intensively by some organisations.

Packaged free products

Some organisations decide to offer their free products in packages (boxes, discs, manuals, documentation, etc.) in exchange for the payment of an amount that, while lower than the price of alternative solutions, is still greater than cost price. For example, the Ubuntu distribution can be purchased at http://www.ubuntu.com/getubuntu/purchase.

Free software production is mainly a response to the sale of complementary services with added value for clients, which also serve to extend the continuity of the software, such as customisation or adaptation to a specific environment.

The materials for the "Introduction to free software" subject include a classification of the possible alternatives to the development of free software, which we will summarise below:

See also

You will find more information and examples of this classification in Section 5.2 "Business models based on free software" in the materials of the "Introduction to free software" subject.

- **Better knowledge**. This is based on the idea of doing business with the knowledge of one or more free products, offering tailored developments, modifications or adaptations (among others that we will describe later). Active participation in the creation and development of free products is the added value introduced by the company in the eyes of its clients and the competition.

- **Better knowledge with limitations**. This is similar to the previous model (better knowledge) but with a mixed implementation of free and proprietary licences (or patents) to reduce competition. This model may prove not to be viable if the free product forks into a branch supported by the free community, causing the competitive edge to fade.

- **Source of a free product**. This is similar to the first model (better knowledge) but with the difference that the company produces almost all of the code. The client evaluates the positioning and competitive advantage over the competition. This model receives the support of the free community.

- **Source of a product with limitations**. This is based on the previous model (source of a free product) but with an implementation aimed at

reducing competition, such as beginning distribution with a privative licence and opening it up afterwards or limiting the initial distribution to company clients.

- **Special licences**. This involves the production of a single product distributed under different licences (free and proprietary). The proprietary product offers special implementations of the product, such as integration with other proprietary products.

- **Sale of a brand**. This involves the distribution of free products with the image of a corporate brand, offering quality and added value from the point of view of clients. These products are usually accompanied by numerous complementary services for clients.

The choice of the type of software development business must go hand in hand with the business strategy and be suited to the target market. Hence, an organisation may decide to use a customised typology for each of the products it plans to introduce on the market, based on the strategy and specific target market of each product.

The free software development business model also requires a careful selection of the licences of the source code it uses:

- The licence for the source code it modifies if the end product improves an existing application.

- The licence for the source code it links to if the end product needs to implement function calls to external libraries in order to operate.

- The licence for the source code it creates, i.e. when the source code is completely new.

- The licence for the source code of the end product, which encompasses the combined source codes of the end product.

The importance of carefully determining the licences associated with each part of the source code used lies in the differences between the various free licences. In other words, although all free licences guarantee the basic four freedoms, they differ in their policies for licensing the redistribution of modified code, which is precisely the object of the free software development business model.

Appendix I of this unit briefly describes the most important characteristics of the main free software licences and explains the redistribution policy and the compatibilities of linking and derived work.

The selection and correct combination of licences will have a direct effect on free software production and may have legal implications if it is not done properly. The last section of this unit on free software production looks in detail at how to select the relevant licence based on the product parameters.

Lastly, we should be aware that free software promotes and uses public specifications called open standards to promote the universality and interoperability of the formats handled. Appendix II of this unit includes a brief description of the main characteristics of open standards together with some examples.

3.1.2. Consulting

The consulting business model is based on the generation of professional services to complement free software for users and organisations.

To a certain extent, this business model is based on providing quality external professional technology services to organisations that are not fully in control of the creation, management, development and evaluation of their internal technology projects.

See also

For more information on the typology of production projects, see the "Classification by scope" section of the first unit.

Consultancies can offer a broad range of professional services, which will depend on the strategy and context of the business. Nonetheless, they will be closely related to the study, analysis, design and evaluation of the free software systems implementation project described in the second unit of this module.

There now follows a brief classification of the main services that consultancies can offer their clients:

- **Project management**: this involves the creation and functional management of the free software implementation project. The tasks carried out as part of this service cover the life cycle of the project, management of the teams of professionals involved at each stage of the project, control of the effective progress of the project and generally all tasks concerning the coordination, information, management and monitoring of the project.

More information

The "Project typology" section of the first unit and the "Life cycle" section of the second unit contain more information on the management and life cycle of free software projects.

- **Execution of the project analysis and design stages**: this requires carrying out one or more analysis and design stages of the free software implementation project. The tasks of this service are those that the organisation outsources to the consultancy, such as the study of the current system, study of the requirements of the new system, analysis of

free software solutions and/or design of the new system, in accordance with the stages indicated in the second unit of the module.

- **Evaluation and auditing**: this requires carrying out professional technological assessments of one or more characteristics of systems in operation. The tasks of this service may be typical of a systems implementation project, such as the execution of stages we saw earlier, but they can also be carried out independently and in isolation. The evaluation or audit evaluates one or more aspects of the system, such as security, performance, efficiency or efficacy, among others, and may be performed before and/or after implementation of the system.

- **Advice**: this service is geared towards offering support and professional help and advice for technological decision-making in the organisation. These tasks can be carried out prior to the start of any implementation project or during the study, analysis and design phases. In all events, they form part of the professional support provided for strategic decision-making on technological aspects affecting the future of the organisation.

See also

The "Study of the current situation", "Study of the implementation requirements", "Analysis of free software solutions" and "Development" sections of the second unit contain more information on the study of the current system, the study of system requirements, the analysis of free software solutions and system design, respectively.

This list is by no means exhaustive or exclusive because the business model can provide two or more services to cover its business strategy. In addition, the consulting business model can be combined with other business models to offer clients a comprehensive technology service.

Hence, the best knowledge of the technology platforms in place (or to be implemented), and excellence in the analysis and extraction of information and conclusions, or the scope and complexity of the project, are decisive characteristics that affect an organisation's decision to outsource management or stages of implementation.

Consulting work is normally formalised through open or closed contracts. In open contracts, the relationship begins when a specific service is commissioned and, depending on the result of this service, the contract may be extended with the commissioning of new services. For example, open contracts may be used for the execution of one or more stages of the project.

In contrast, closed contracts are awarded for the completion of a specific aim, task, result or assignment and there is no direct possibility of extending the contract in the same scenario. For instance, closed contracts may be used for the independent auditing of a system, given that these tasks are carried out in isolation and at specific points in time.

3.1.3. Installation and integration

The installation and integration business model is based on the direct implementation of free software systems for users and organisations, usually as part of free software projects.

In a way, this business model considers free software as the object of the production of its services, rather than a product in itself. This creates a market offering substantial benefits to clients:

- The organisation does not have to pay for licences for free software products that are freely distributed so it can cut the costs of technological implementation.

- The organisation does not need to practice product piracy and thus will not breach the applicable legislation.

- The organisation can directly adapt the free solutions, thus cutting the cost of the implementation of specialist systems.

- The organisation can adopt integrated direct implementation packages, and hence reduce the risk of technological implementation.

A number of services can be offered under this business model, the main ones of which are listed below, although this list is not exhaustive and does not exclude other services:

- **Configuration**: this model carries out the tasks of setting up and fine tuning[36] a system already in place in order to formalise the initial set-up, enhancing its performance or adapting it to new purposes not considered initially. In all events, the fine-tuning does not affect the source code of the application, only the configuration of the components that can be adapted to the specific features of the installation.

(36) Also known as *set-up* or *tune-up*.

- **Tests**: the aim of these is to provide benchmarking for systems, applications or free software solutions from a given perspective. We may need to conduct a comparative analysis of free solutions, testing of a new system design or testing of directly implemented software on specific hardware, either as part of an implementation project or as independent, one-off tests.

- **Integration**: this involves carrying out and/or checking integration between two or more free software solutions in order to provide a single package to resolve a specific operating function[37]. This integration can usually be resolved with a configuration to suit each element and possibly an additional component allowing more efficient integration.

- **Installation**: this involves the bulk installation of software for direct implementation on client machines or servers[38]. This service may require the configuration and fine-tuning of both the free software to be installed and the hardware on which it will be installed. It may also be necessary to integrate the diverse solutions we wish to install. When the same software needs to be implemented on a series of computers with identical hardware, it may be useful to use pre-configured image distribution and cloning software to save time and money and to improve the efficiency and efficacy of the process.

- **Distribution**: this involves redistributing free software to clients, either in the original format[39] or in customised configurations related to the scope of the business, such as tool integration, operation geared towards clients, servers or a work station, among others. The redistribution of integrated software is subject to the licences of the specific solutions. Appendix I describes the main free software licences and their compatibilities.

As we can see from the above classification, these services are closely related to the stages of free software solutions analysis and of implementation and migration of the implementation project described in the second unit. Hence, a better knowledge of the technology platforms and excellence and efficiency of services or the scope and complexity of the project are decisive characteristics affecting an organisation's decision to outsource the installation and integration of its system.

More information

The "Analysis of free software solutions", "Development" and "Implementation and migration" sections of the second unit contain more information on the analysis of free software solutions, system design and system implementation and migration, respectively.

(37)For example, LAMP (*Linux, Apache, MySQL, PHP*) is an integrated software package with diverse individual aims but which, as a whole, solves a specific problem efficiently and effectively.

(38)This is similar to the term *Installfest*, though applied here to a structured business.

(39)At http://freshmeat.net/ and http://sourceforge.net/, you will discover a wide range of free software projects in the key areas where the technology is used.

3.1.4. Systems migration

The systems migration business model is based on transferring the operating function from the system in place to the system to be implemented.

Migration is a complex process that must be carried out with accuracy and rigour, since the data and configurations we are dealing with are the organisation's capital.

The diversity of situations encountered by companies that migrate systems are the result of a combination of the source and target platforms of migration. A thorough knowledge of the platforms and experience in migration are the basis for offering added value to clients.

The following list indicates the main services offered under this business model, although it is not exhaustive and does not exclude other services:

- File system services, both for the server and clients.
- Printing services, between clients and between servers.
- Directory services and centralised authentication services.
- Network services, particularly automated management protocols for network control, communications and clients.
- System management and administration, for network and software management.
- Web services, for static and dynamic platforms.
- Databases, for data migration and access verification.
- Desktop environments and office automation applications, for applications and user data.
- Corporate applications, for applications that can be run directly and those that require tuning or virtualisation.

See also

The "Implementation and migration" section of the second unit details the characteristics of systems migration. In the "Migration of the services of a system" section, you will find more information about the services described here.

The complexity and scope of migration, the ability to carry out the process carefully, efficiently, effectively and as quickly as possible, and the best knowledge of the source and target technology platforms of the migration are decisive aspects that can persuade organisations to outsource migration of their systems.

As we can see from this classification of services, the migration process may need other services, such as installation, configuration, integration or testing to ensure that we meet our aims.

Free software uses and promotes open standards for interoperable data exchange and its role in systems migration is particularly important. For instance, starting off with a system that does not store data in open standards could complicate migration because of format conversion, especially if the original is privative. Appendix II of this module introduces the open standards, defining them and the organisations behind them, and offers some examples.

3.1.5. Systems administration and maintenance

The systems administration and maintenance business model involves carrying out management and monitoring tasks for a system already implemented and operating.

The main aim of the services offered under this business model is to keep the entire system up and running, adapting the configuration to changes, solving any problems that may arise and repairing malfunctions that affect the normal operation of the organisation.

The following list indicates the main services offered under this business model, although it is not exhaustive and does not exclude other services:

- **Administration**: consists of providing basic management of the system, the adjustments required as time goes by, supervision of its operation, the implementation of new functionalities and the evolution of the system. Many administrative system tasks can be carried out remotely[40].

(40) For instance, through the combined use of `ssh`, `cron` and `at`.

- **Maintenance and evolution**: consists of supervising, monitoring and redressing issues in the system that could affect its operation, together with the control and evolution of the obsolescence of its components. Examples include malfunctions and the deconfiguration of hardware or software, the control and updating of software versions, and the evolution plan for the hardware and software.

- **Security**: this consists of managing the security of the system, controlling risks, maintaining policies for prevention, contingencies, diagnosis and debugging. Examples include backups or the control and maintenance of keys and certificates.

Given the characteristics and features of these services, many organisations decide to keep staff on for these tasks, but some small and medium-sized organisations are unable to create such a position.

The outsourcing of some services, such as intranet and extranet servers, can also lead to the contracting of external administration and maintenance services. These services are normally contracted for a fixed monthly or yearly fee and cover a specific service level.

Intranet and extranet

Intranet and extranet web services are easily outsourced because of the proliferation of *data hotels* and *data centres*.

3.1.6. Support and training

The support and training business model involves providing professional technical assistance for the technological training of users and the resolution of issues and problems relating to use of the system.

The implementation of free software systems may initially require user support and training measures, particularly if the previous system was based on privative software. As we saw in the second module of this subject, the implementation project must take into account the need for user training in order to promote positive change management, whose features make it a service that can be easily outsourced to companies specialising in this sector.

The following list indicates the main services offered under this business model, although it is not exhaustive and does not exclude other services:

- **Training**: this service provides education and training on free software tools, including operating systems and office automation tools, to end users. The service can also include specialist software training as a result of the development of the systems implementation project, so it may be useful for change management purposes to coordinate this task with the implementation team.

- **Support**: this service provides technical assistance to users in order to solve everyday problems. Many of these services are provided from call centres, but it can be a good idea to provide e-mail addresses for resolving issues or instant messaging with professionals. It can also be convenient to combine these tasks with those of the systems implementation project in order to fix possible bugs in the implemented software.

See also

The "User training, communication and support" section of the second unit contains more information about user training and support.

This business model generally requires human, technological and material resources adapted to the aims of the training:

- Human resources with an deep knowledge of the issues and the ability to transfer knowledge and solve problems.

- Technological resources suited to training and support, such as technological platforms for learning or call centres offering technical assistance.

- Material resources adapted to the training, such as specific documentation and manuals on free software[41] with free licences.

Internet resource

Moddle is an example of a virtual learning platform based on free software. http://moodle.org/.

(41)The European SELF project has a platform for creating and sharing educational materials relating to free software and open standards (http://selfproject.eu/).

The quality of these parameters is essential if the organisation is to outsource its training and support services. Training services are usually contracted in the form of courses whose structure is agreed previously, while support services are contracted for a monthly or yearly fee, following agreement on the services covered.

See also

Section 10.2 "Licences of other free resources" of the "Introduction to free software" subject material contains two licences for documentation, materials and literary works that are widely used.

3.2. Business plan

A business plan or project is an instrument that identifies, describes and analyses a business opportunity, studies its viability and develops the procedures and strategies for creating the company to exploit this business opportunity.

Clarification

In this section, we will use the term business plan because our aim is to describe the elements required to set up a free software company, as in the case of Cometa Technologies, which we shall see in the second section.

Taking into account this definition, the aims of the business plan will be as follows:

- To conduct a market study to position the business plan and determine its technical, economic and financial viability.
- To develop the measures required to achieve the aims set out in the business plan.
- To monitor the evolution of the company and analyse deviations from the initial business plan.
- To serve as a calling card for the project and the business entrepreneurs in order to obtain the financing and support of third parties.

Although the first three aims are mainly internal, the last is external and visible by people who do not form part of the project, in theory at least. When drafting our business plan, we always need to consider this dual aim: to act both as a plan for the project and as the presentation of the project.

Naturally, we need to avoid falling into the trap of omitting the risks or negative parts of the project to make it look more attractive to investors. In fact, missing out these elements could be detrimental to our business project because it would be based on false suppositions. A true picture of the technical and economic aspects is thus one of the basic requirements for drawing up a business plan.

Any business plan needs to respond to a series of questions about the project we wish to introduce: Who?, What?, Why?, Where?, When? and How much?

- **Who?**

The name of the company, the brand of the products or services offered, the names and track record of the business developers.

- **What?**
  The description of the products or services offered, the markets at which they are aimed and the market share set as the target, among others.

- **Why?**
  Every business plan generally seeks to obtain and maximise profits. However, this is not incompatible with other aims, such as improved quality of life in society or the creation of jobs.

- **Where?**
  The geographical area where the products or services are to be marketed, which can be regional, national or international. The distribution channels that will be used, including possible agreements with other companies who will allow entry to other regions.

- **When?**
  The expected start of business and subsequent planning, including temporary conditions or limitations that could affect the company, such as procedures for obtaining licences, production time, obsolescence of certain technologies or seasonality.

- **How much?**
  The initial investment needed to launch the business project, the minimum and desired turnover, the threshold of profit and loss, the reinvestment of profits and the sharing of dividends, among others.

These issues are covered in the following aspects, which are found in almost any business plan:

- Executive summary
- Introduction
- Business description
- Organisation of production
- Internal organisation and human resources
- Market study
- Marketing plan
- Financial analysis
- Legal form
- Risk management
- Summary and evaluation

Depending on the nature of the company or business, these aspects will have a greater or lesser importance in the business plan and may be organised in different ways.

The following sections will discuss each of these aspects and study their relationship with the free software business models we saw in the previous section.

3.2.1. Executive summary

An executive summary is a short statement[42] that appears at the start of the business plan and summarises the main points of the document. It gives potential investors a comprehensive idea of the business plan without having to go through the various sections in detail.

(42)In all events, the executive summary should not be more than three pages long.

The executive summary should cover almost all of the points of the business plan, which are:

- Description of the business model, with a particular emphasis on the chain of value and source of income.
- Short description of the project developers, their training, knowledge and skills, professional track record and dedication to the new project.
- Concise description of the market, including size, clients, growth potential and barriers.
- Analysis of the functional areas of the project: production, quality and organisation of human resources.
- Summary of the financial analysis of the project and the investment required to set it up.
- Summary of the risks of the project and the plans to prevent them and remedy their consequences.

Obviously, the executive summary should highlight the strong points of the business plan, particularly for the business model we wish to adopt, the strategy that we will use to do so and the team developing the idea.

We recommend writing this part after completing the business plan and to do it from scratch, that is, without re-working texts that we have already written.

3.2.2. Introduction

After the executive summary and the index, the first part of the business plan should be an introduction indicating the name of the future company[43] and the team of developers, together with the other professionals involved in the drafting of the business plan.

(43)If the plan describes a new project or service for an existing company, it is a good idea to include a summary of the company's activity, history. evolution, size, etc.

As we have seen, the presentation of the team of developers should cover the professional career of each of its members and the knowledge they will bring to the business project. More often than not, the team of developers will have members with a profile specialising in business management and others specialising in specific technological areas, as is the case of companies that work with free software.

Lastly, the introduction should provide a brief description of the different sections of the business plan that will be developed later.

Mission and vision

The introduction is the place to describe the mission and vision of the new company because it allows the reader to see how these two concepts are developed in the business plan.

The mission and vision of an organisation are a concise definition of its main features and aims, and its strategies for meeting the latter.

The mission is a short phrase justifying the existence of the organisation, i.e. the basic aim of its activities and the values guiding the activity of its employees. The mission is closely linked to the internal values of the organisation and basically describes how to compete and generate value for customers.

The vision is also a concise phrase describing the medium- and long-term goals of the organisation. It is addressed to the market and should offer an expressive and visionary angle on how the organisation wishes to be seen by the world.

The main differences between the mission and vision can be summarised as follows:

- The mission describes the internal aspects of the organisation and its operation, while the vision describes the external aspects.

- The mission has a short- to long-term time horizon and highlights the aspects that should be put into practice immediately in the organisation,

while the vision is fixed in the medium- to long-term and gives the general lines of the future evolution of the organisation.

Corcaribe Tecnología and eZ Systems

The company Corcaribe Tecnología specialises in products and services based on free software and has the following mission:

"Corcaribe Tecnología provides technological solutions that generate added value with a business model that allows us to offer the best cost for results delivered to our clients, producing authentic tangible and intangible benefits for our members and collaborators."

And the following vision:

"To become a Latin-American reference of continued success in the implementation of integral technological solutions, applying the principles and values of free knowledge in a model of sustainable development."

Similarly, eZ Systems, which provides free content management software sets itself the following mission:

"To be the leading content management platform by 2012."

And the following vision:

"To help companies to manage, publish and share information."

It is not essential to define a mission and vision in the business plan, but it can help to summarise the short-, medium- and long-term aims of the business project and to convey them effectively to potential investors.

3.2.3. Description of the business

We recommend you should begin this section with a description of the company you wish to set up and a brief presentation of the project developers, even if you have already done so in the introduction.

The main aim of this section is to describe the products or services for which you are drawing up the business plan and the business model adopted to offer them, as we saw in the previous section.

Special care should be taken when explaining the features of business models based on free software since the reader of the business plan will not necessarily be familiar with them. This includes aspects of the protection of intellectual property and rights over products and services.

It is also a good idea to describe the needs that will be met by the products or services and point out the main differences with the existing offer, in order to show that the business project is well positioned in the market.

Lastly, we will need to indicate the capacity for the production and provision of services, which will serve as an introduction to the next section on the organisation of production.

3.2.4. Organisation of production

The section of the business plan on the organisation of production describes the technical tasks of the future company.

This far, we have looked at the business plan from the point of view of describing the marketing of new products and services, without distinguishing between them. Now, however, this section of the business plan will take one of the following forms, depending on whether it concerns products or services:

- If the company business is the development, production and subsequent marketing of a product, we need to detail the development and production phases.

- If the company provides a service that does not involve a production process, the procedures for provision of the service and the technical needs should be described in detail.

It goes without saying that the two options are not mutually exclusive and a business plan can contain both. For example, a company might specialise in the migration of systems to free software but also provide training in free software technologies to users and technical staff.

Generally speaking, a business encompassing the phases of research, development and production will be much more complex and involve greater risks:

- **Research and development phase**. When describing the research and development phase, we need to pay special attention to the estimated duration of the research and development phase and the necessary investment in human and material resources.
  In high technology sectors in particular, as is the case of free software, the business plan must evaluate the human resources skills and know-how needed for the successful completion of research and development tasks. This section should also detail the distribution of roles and duties, the risks inherent to all research and development activities, the potential synergies between projects, the process of innovation and continuous product improvement and how this process will be integrated into the production process.

- **Production phase**. The description of the production process should deal first of all with the operating cycle[44], the location of the production facilities, their cost and their accessibility. Secondly, we will need to describe the business premises, buildings and equipment needed for the production or provision of services. For each of these, we will need to indicate the modes of financing and purchasing[45], their features, availability, useful life and annual amortisation.

(44)This includes production capacity in number of units and expected production, along with the staff and number of hours or shifts needed for production.

(45)We can also present expansion plans for the facilities and the purchase of new equipment.

Special attention must be paid to quality management. Here, we should describe the quality standards and certifications that will be applied to both the processes and the results of the production process.

Lastly, we need to offer a strategic vision of the production process; for instance, if we are outsourcing production of certain components or part of the production process[46].

(46)For example, a free software publisher could outsource production of the distribution medium and packaging of its programs.

Again, the description of the free software production process reveals a number of differences with proprietary software development, which should be explained in detail in the business plan, particularly when the reader is not familiar with free software. We can also stress the added quality of free software when compared to proprietary software.

More information

The "Free software production" section of the third unit looks in detail at the special features of the production process for free software.

In all events, remember that you should always explain the advantages and disadvantages of the various alternatives and justify each of your decisions.

3.2.5. Internal organisation and human resources

This section of the business plan details the organisation of the team needed to develop the business project and the profiles required for it.

Firstly, we need to include a description of the key duties and management positions, along with the necessary profiles and even the name and professional background[47] of the people who will fill these positions if they have already been selected. We must then describe the categories of professionals required by the company, their duties, main tasks and the type of contract they will have. It is a good idea to indicate the salaries of each type of worker, regardless of whether or not they are management positions.

(47)This includes their professional experience, specialisation and main professional achievements. The purpose of this type of information is two-fold: on the one hand, it boosts the confidence of potential investors and, on the other, it allows us to pinpoint the strengths and weaknesses of the management team.

The internal structure of the company can easily be represented on an organisation chart by departments and business areas, naming the individuals occupying the management positions, if these have already been filled.

This section should conclude with a description of the company's general human resources policy and indicate whether the creation of a specific human resources department is necessary or, alternatively, this function can be split across the different departments.

The need for and availability of qualified staff in a given area and at an acceptable cost can sometimes be a major obstacle, which is the case of specialists in free software[48].

(48)For example, when it comes to publishing these materials, although free software is well known and technologies and solutions based on it are fairly popular, it is difficult to find professionals who have actively participated in free software projects, whether as employees of a company or on their own initiative.

Furthermore, a company that bases its business model on free software may require the creation of positions and responsibilities that suit its specific features. For example, besides the traditional positions of technical director or sales director, we can come across roles such as community director, the person who manages relations with software developers and users, or director of cooperation projects, who manages and coordinates projects developed in collaboration with other companies, research centres or universities.

3.2.6. Market study

Market studies are an essential part of business plans and hence one of the keys to its success.

A good market study will help us to assess the technical and financial feasibility of the business project correctly and to identify potential clients and competitors in order to come up with the right strategy for marketing the products or services detailed in the business plan.

When drafting a business plan, it is useful to conduct the market study first, at least as a rough version, because its results can affect various parts of the business plan.

Thus, the market analysis needs to provide information on the following aspects:

- **Current market situation**. We must first segment the market according to the most relevant features of the business plan and determine its size together with its past evolution. We will also need to determine the decision-making process of the market and the behaviour of its customers, particularly their reaction to the launch of new products or services.
  Secondly, we need to evaluate the needs that may arise as a result of the introduction of the products or services proposed in the business plan. This depends largely on whether the product or service offers something new and on our ability to influence customer habits.

- **Market growth forecasts**. Once we have established the current status of the market, we need to be able to predict its future evolution. Is this

market developing, stable or declining? How fragmented is the market? Is the market becoming concentrated?
Again, we need to take into account the potential influence of the new products or services on the market. For example, the introduction of solutions based on free software can effectively change the market because they encourage the formation of a new sector specialised in free software.

- **Identification and classification of clients**. One of the primary aims of market analysis is to discover who the potential clients of the proposed products or services are. The task of classifying the diverse types of client based on common features is also very important because it allows us to define different strategies for each. For example, a company that implements free software systems in companies will present itself differently to clients depending on whether it is a family business or a major corporation. We also need to remember that a product or service may be offered to clients who are theoretically different. The flexibility and interoperability of free software encourages this type of action.
  Moreover, we need to evaluate each type of client's reception of the product or service. Continuing with the previous example of a company that specialises in the implementation of free software systems, a major company with dedicated technical staff may be more reluctant to use free software, partly because of the fear of change, whereas a family business might be more receptive.
  Lastly, if the future company already has a client portfolio or has clients who have expressed an interest in its products or services, this can be included in the market study.

- **Analysis of the competition and its products**. A market study should reveal the competitors of the future company and identify their strengths and weaknesses as well as those of the products or services they offer.
  We must indicate the characteristics of their products and services, including price and quality, and their market share and sales strategy. It is very important to identify the market leaders for each of the products or services covered in the business plan.
  We should also be careful not to disregard potential competitors, i.e. companies that are not yet on the market but which could enter, or companies from other geographical regions. In our current climate, particularly in information and communication technology sectors, as is the case of free software, the competition tends to be global and many companies can offer their services directly or indirectly from any location.

- **Analysis of barriers to market entry**. Barriers to market entry are obstacles that companies come up against when they enter a new market, such as the need for major investments in the case of new companies or the lack of an established brand. An example of this is that free software tends to suffer from a perceived lack of quality in contrast to the proprietary software companies and solutions already on the market.

In the same way, however, we can also study which barriers to entry we should focus on once we are on the market in order to keep the competition at bay.

- **Influence of governments**. The market study should discuss the way in which local, regional, national and international governments can influence the market and, hence, the viability of the business plan. For example, governments can act both as market regulators and as providers and clients.

  This is particularly true in the case of free software, which, as we have seen throughout this subject, is a point of interest for many governments, including the regional government of Extremadura and the Federal Government of Brazil.

  Market studies need to be planned carefully because they involve a number of phases, which can be summarised as follows:
  - Collection of general information, by which we obtain a large volume of data on the market under study.
  - Analysis of the information obtained.
  - Selective search for information to obtain the missing information required to complete the market study, which will have been identified in the previous analysis.

  To conduct the market study, we will need a great deal of information and this is not always easy to obtain. There are countless bodies and sources of information, both general and specialised in specific regions and sectors: governments and national statistics institutes, local and regional governments, private bodies such as chambers of commerce and business associations, journals and specialist publications.

  A good market study should conclude with a strategic analysis[49] that relates the results of the study to the description of the business and planned resources, and that also highlights the potential of the business plan.

  (49)This analysis can also be supported by the use of strategic tools such as SWOT analysis (http://es.wikipedia.org/wiki/An%C3%A1lisis_DAFO) or Porter's Five Forces (http://es.wikipedia.org/wiki/An%C3%A1lisis_Porter_de_las_cinco_fuerzas).

3.2.7. Marketing plan

The purpose of the marketing plan is to define the commercial strategies that will enable us to reach the predicted turnover of the financial analysis, which we will look at in detail in the following section.

Hence, the marketing plan details the actions we need to take in order to apply the business model and opportunity described in the business plan and to exploit their competitive advantages.

Thus, the marketing plan needs to take into account the following:

- **Overall commercial strategy**. The overall strategy needs to define how the sales aspect is integrated into the business project. We need to explain how we will identify clients and how we will contact them, why will clients be interested in or decide on the products or services we offer, and hence, which features of our products or services we will emphasise to generate sales, such as price, quality, guarantee, technical support, etc.
  Free software is a prime example of this, since its main attraction for potential clients is the reduced costs it generates rather than its quality, which is often superior to that of proprietary software. In contrast, clients usually identify the high prices of proprietary software with superior quality, and free software, which is more economical, with inferior quality. Hence, when drawing up a business plan based on a free software business model, it is essential to emphasise the superior quality of[50] free software.

(50)This can be done by emphasising the interoperability and flexibility and the constant revisions and improvements undergone by free software.

- **Sales strategy**: defines the short- and long-term sales aims and the market sectors in which the products or services offered will be introduced initially and in the long run. In all events, decisions must be justified and backed up by the results of the market study.

- **Price strategy**: first of all, we must determine the prices at which the products or services will be sold, comparing them, if possible, with those of the competition, estimating a gross profit margin and evaluating whether this is sufficient to sustain the company's entire business activity[51].
  It is very important to justify the price policy, especially when comparing it to that of the competition. If the price of the products or services offered is higher than that of the competition, this should be explained in terms of innovation and quality, features and enhanced guarantees. If the price is lower, we will need to explain how we will make it profitable (for instance, greater efficiency and lower production costs). Again, it is very important to explain the reasons for the low cost of free software and the benefits that this brings.
  Lastly, the price strategy should be optimal, i.e. it should maximise the profit margin, and hence, profitability. A higher price can sometimes generate greater profits, even though it partially reduces sales.

(51)It is also a good idea to compare your own margins with those of the competition, if you have access to this information.

- **Sales policy**: this determines the composition, form of contract and profile of the sales team, including representatives, at the launch of the company and in its medium- to long-term evolution. It includes the sales margin policy and the promotional measures that will be offered to sales representatives and authorised distributors.
  The sales policy also includes: estimated sales for each sales representative, incentives, the collection periods agreed with clients, and special promotions such as discounts, advances, rebates, etc.

- **Promotion and advertising**: you should describe the measures that will be taken to attract the attention of potential clients to the products or services offered. These measures include mass e-mailing, participation in trade fairs and events, advertising on websites, etc.
  We will eventually need to quantify the cost of the promotion and its return through consultations with clients and the sales we have closed.

- **Aftersales service and guarantees**: you should describe the aftersales service and the guarantees of the products or services, where applicable. In other words, what type of service and guarantee are offered, their duration and price (if optional) and their cost for the company.
  With free software, part of the aftersales service is indirectly provided by the community of developers and users, which constantly improve successful products. The free software company plan should keep this in mind and explain it as an advantage, but never as the only form of additional support. Remember that the vast majority of clients would like the aftersales service to be included and guaranteed in the conditions of sale.
  Lastly, we also need to assess the impact of our aftersales service and guarantees on the client's final decision and compare our service with that of the competition.

- **Distribution policy**: the distribution policy should describe the distribution channels that will be used and the discount, commission and margin policies applied to each of these channels.
  In free software business models, we often come across programmes for partner companies[52] in different forms: system integrators, software vendors, etc., and those who are paid commission and offered dedicated services and assistance and access to promotional channels.
  As we explained above, free software products and services can easily be offered on the global market, so the marketing plan must study this possibility and its potential features, including the effect of international laws on the company's activity, overseas collection management, etc.

(52) One example is the Openbravo free software ERP partner programme, which you can visit at http://www.openbravo.com/partners/join-openbravo/details/.

3.2.8. Financial analysis

The financial analysis or study is also one of the essential parts of any business plan, since its aim is to evaluate the feasibility and financial potential of the business project, detect the investment needs for its launch, identify the resources available initially and describe the various possibilities of financing.

Contrary to what we may think, the financial analysis is one of the most creative parts of drafting a business plan.

The financial statements or main headings that need to be covered by the financial analysis are as follows:

- Cash position over the first year, broken down by months to reflect the effects of seasonality[53].
- Analysis of working capital, which allows us to determine the liquidity of the company.
- Calculation of the balance point and alternatives if target sales are not reached.
- Financing needs and alternatives, selecting those that are most profitable and including elements to explain the decision.
- Annual balance sheets with a five-year view and the first year broken down by months.
- Source and application of funds, allowing us to predict risk situations for the company and evaluate the source and long-term use of funds.

(53) Even highly technological business plans, such as those based on free software, can be affected by economic seasonality (the summer holiday period, for instance).

Working capital

Working capital measures an organisation's balance of assets and liabilities and confirms that there are more liquid assets than short-term debts. For more information, see http://www.innovaceei.com/es/knowledgebase/index.asp?faqsRecid=385&faqRecid=385&show=4460.

Combined analysis of these financial statements is recommended as we can draw conclusions about the business project as a whole: the amount of capital required and when it will be required, and the necessary debt and when it should be paid, among others.

We should also explain the expected return on our investment and indicate when this will be recovered.

As explained above, we need to avoid falling into the trap of presenting an overly optimistic financial analysis to investors in order to win them over, since this will go against the company sooner or later and question marks will be raised about its viability and credibility.

3.2.9. Legal form

If the last aim of the business plan is to create a new company[54], we need to choose the legal form of this new company, its tax system and its founding partners. We will also need to indicate the name of all partners and investors together with their participation in the new society.

(54)If the business plan is for an existing company, this section should describe its legal nature and any modifications that implementation of the business plan could bring about.

It is a good idea to detail the procedures required to set up the new company step by step, together with their costs and the time needed for them. We must also indicate whether we will be using the services of external advisory specialists and the cost of these.

3.2.10. Risk management

All business projects, whether to create a new company or a new line of business, involve numerous risks that are sometimes unavoidable. Hence, the business plan should offer a complete description of the risks and their consequences.

More information

You will find a general introduction to this topic in the "Risk management" section of the first unit.

Risks can be classified as internal (originating in the company) or external and by the functional area that they affect: technical, commercial, etc.

For example, internal risks can include delays in production or a lack of qualified staff, while external risks can be a new market regulation that partially reduces return or the emergence of new technologies that cause the products or services offered to become obsolete.

We need to define a contingency plan for each risk, which includes a series of preventive[55] actions, i.e. measures to try and prevent the risk from occurring, and a series of actions to mitigate or remedy[56] the risk, which should be adopted if it materialises.

(55)For instance, to prevent the appearance of new technologies that could leave the products or services in the business plan obsolete, we should practice active technological surveillance, possibly collaborating with companies or organisations that work in the same area.

Some risks can have negative effects, but they can also be positive. For example, changes in the legal or political framework that can affect the business model but which also provide new business opportunities.

(56)For example, human and material resources for other departments could be used to recover a delay in production.

The correct identification and assessment of risks in a business project and the drafting of suitable contingency plans for them, far from highlighting project weaknesses, actually emphasise the management skills and precaution of the business developers and enhance their credibility.

3.2.11. Summary and evaluation

The last section of the business plan should summarise the strengths and weaknesses of the business project, the advantages and opportunities it offers and its main risks and threats.

The summary is your last chance to persuade a potential investor so you need to be very convincing and seize the opportunity to emphasise the arguments in favour of the business project and those that its developers believe in.

However, after drafting the business plan, the project developers may find that it will not be as profitable as they had hoped or even discover that it is completely unfeasible. This shows how useful the business plan is as a tool to identify the best business opportunities.

3.2.12. Business plans and free software

Drafting a plan for a free software business is not that different to the procedure for drafting business plans in other sectors and we have already seen some of its features in the previous sections.

In general, we need to remember that a business plan may address different types of reader: advisors, investors, technicians, bankers, etc. Therefore, we need to use a language that they can all understand and avoid using highly technical vocabulary. When the use of technical terminology is unavoidable, you should explain each concept clearly in simple terms. Investors never invest in anything that they do not fully understand.

We also need to take the time to explain the features of free software, pointing out the differences between it and proprietary software, and highlighting its main advantages. Do not hesitate to use real-life examples and examples of success to back up the arguments put forward in the business plan.

Although free software is adopting an increasingly relevant role in the media and society thanks to the commitment of the free software community, companies and government bodies, its nature and financial implications are not so well known.

Again, we must take the time to explain the free software business models carefully and be prepared to answer and possibly even anticipate the most common questions, such as, How can you invest in and earn money from something that anybody can copy?

3.3. Production of free software

Many of the business models described in the first section depend to a greater or lesser extent on free software development.

One of the problems with free software projects is that only the successful projects are echoed in the community and only the very successful ones reach the non-specialist media.

However, before we turn to look at free software production, we should remember that the vast majority of free software projects are a failure for specific reasons. It may simply be that the project fails to produce quality and competitive software or that it does not manage to attract the attention of the community of developers and users.

Needless to say, as we saw in the subject materials, a free software project should be dealt with as a software project and only in the last instance, as a simple engineering project. Hence, a free software project will pose the same initial risks and problems as any other project.

However, given the free nature of this type of project, there are other strengths and weaknesses to take into account. Due to the seemingly non-professional nature of many free software development projects, their execution may appear easier than traditional software development projects, but there can be nothing further from the truth.

The aim of this section is to describe the features of free software development projects, contrasting them with proprietary software development and offering a series of good practices to encourage their success. These practices correspond to the key areas and elements required to set up and execute a free software project, namely:

- Creation and presentation of the project
- Necessary infrastructure
- Organisation of the community
- Development
- Releasing and packaging
- Choice of licences

Naturally, not all of these steps are compulsory. As we saw in the business models, a free software company might initiate a project or, as occurs in most cases, it might join an existing project.

This last option is often the most advisable and, given the nature of free software, it does not rule out the possibility that a new project could be created from an existing one under the identity of the company or organisation interested in leading the development.

3.3.1. Creation and presentation of the project

This section deals primarily with the steps required to create a new free software project and present it to the community.

Thus, the first step to take before creating a new project is to find out whether there is a project that already does what we propose, at least in part. If there is a similar free software project that we can contribute to or reuse to launch a new project, we can contact its leaders to explore the possibilities of collaboration and their future plans.

Generic search engines

Generic search engines are the first step to finding existing projects, along with news sites, directories and public forges, such as http://freshmeat.net, http://directory.fsf.org and http://www.sourceforge.net.

If we decide to create a new project, the first thing we should do is choose a name that will identify it in the community. Generally speaking, a good name will give an idea of what the software does or at least its field of application, and it should be easy to remember.

Whether we like it or not, English is the de facto official language of the Internet. So, if we want our project to have a global impact – and this should usually be the case – we should try to come up with a name that will have some meaning in English or that is neutral[57].

(57)In other words, a name common to several languages, such as Apache, or which is not associated with any major language, such as Ubuntu.

We should also pay attention to legal aspects, to ensure that the name does not conflict with brands and that the associated high-level Internet domains[58] are still available.

(58)In other words, .com, .net and .org.

As we saw in the section on the creation of business plans, all projects must have a clear definition of their mission that will attract the attention of users and developers and let them decide whether or not they are interested in the project.

Along with our mission, it is important to state clearly that the project concerns free software, which means making a clear reference to *free software* or open-source software).

Other key elements in the presentation of a free software project are:

- **List of planned functionalities[59] and current requirements**. This should be drawn up in simple terms without the use of technical vocabulary. It is a sort of detailed summary of what the software does and allows users to find out easily whether it has the functionalities that they are looking for.

(59)These can be indicated along with the words "in progress" or "in development", ideally with the date or version when they will be available.

The requirements should also be easy to understand so that users know whether the application can be installed and used on their system.

- **Development status**. In the free software community, users are usually very interested in knowing how the project is coming along, both if it is a new project and if it is an older one. Thus, we should explain the short- and long-term aims of the project, and the functionalities currently being developed and that will be available in future releases, etc.

- **Available downloads**. The source code should always be downloadable in standard formats, using a straightforward method that does not complicate the process for the user[60].
  The installation process should also be simple and, most importantly, comply with the standards from the very start. It is not initially necessary to provide binary packages or executables unless the compilation process is very complicated.

(60) For example, it is preferable to avoid user registration processes for access to the download area.

- **Development repository**. Unlike users, potential developers are more interested in accessing the working repository, where they can follow the day-to-day evolution of the project and participate in it, either by adding new functionalities or fixing bugs. Thus, it is a good idea to allow everybody anonymous read access to the repository.

- **Bug tracking**. As with the repository, the bug tracking[61] database should also be open to everybody. Paradoxically, the more bugs the project database contains the better because this means more users and more participation in the project.
  There won't be many bugs at the start of the project. It is good practice to log any bugs fixed internally by the project team in the database.

(61) We often come across the terms *bug tracker* and *bug database*.

- **Communication channels**. One of the aims of any free software project is to create a community around it and, in order for this community to organise itself, we need to facilitate the right communication channels. This includes mailing lists, IRC channels, forums, etc.
  In the first phase of the project, it is wise not to diversify or specialise the communication channels too much. A single forum or distribution list for users and developers may be sufficient to encourage interaction between them.

- **Documentation for users and developers**. Documentation is essential for any free software project, for both users and developers.
  Good user documentation needs to explain how to install the software and how to use its functions. You can also provide users with a basic tutorial, containing a step-by-step explanation of how to perform the most common tasks. Maintaining a section of frequently asked questions or FAQs is the perfect complement to the documentation.

Developer documentation should include the contact details of the main project developers, instructions for sending error reports and patches, and a presentation of the development organisation and the decision-making process used by the developers.

We will look at all of these elements in detail in the following sections.

To conclude this section, we would like to emphasise the importance of appearance – that is, how the free software community sees the project – for the success or failure of a free software project.

Many developers neglect this communicative and public relations task, but it forms an essential part of virtually any successful free software project.

For this purpose, we will need to clearly define the aims of the new software, which can usually be summarised as:

- **To explain clearly what the software does**: its main functionalities, the current state of development and future plans, and its positioning vis-à-vis existing solutions and projects.

- **To raise the profile of the software**: ensuring that it reaches the community or market of potentially interested users and developers.

- **To promote the use of the software**: ensuring that potential users and developers know how to use the new software and adopt it instead of the alternative solutions.

- **To involve new developers in the project**: allowing them to contribute to the development of the project through the implementation of new functionalities and to state their opinion on the future direction that the project should take.

These last two aims, obtaining lots of users and lots of developers, are often the most important ones. However, we need to implement one strategy for users and another for developers since, while they form part of the same free software community, they represent two very different audiences.

We need to clearly define the message we wish to convey to each and structure it with a gradual complexity to ensure that the level of detail corresponds to the effort required by the reader. For example, there is no sense in saturating the user with software architecture or explaining technical details to the developers without first giving them a reasonable overview of the architecture.

Finally, this message should be easily accessible, reaching its audience through advertisements on forums or related communities, on the project website or even in the documentation, among other options.

3.3.2. Infrastructure

All free software projects need a series of tools to manage the information generated daily by the project, from the developed code to communication among its members.

We introduced some of these tools in the previous section because they are needed to implement the project:

- **Website**
  This provides a centralised source of information about the project and offers access to other specialist management tools.

- **Mailing lists**
  This is one of the most common channels of communication in free software projects. Message exchanges are usually archived and used as reference and to form a knowledge base for the project.

- **Version control system**
  This allows developers to control the creation and management of the code, returning to previous versions and merging different versions. With the version control system, anybody can visualise the current status of the code and its evolution over time.

- **Bug tracking system**
  This allows developers to track the functionalities and bugs they are working on individually and to coordinate themselves and plan new releases. Although bug tracking is its main function, the database can also be used to track any project task, such as new functionalities.
  With the bug tracking system, anybody can find out if a bug has been fixed or if it is being worked on. In conjunction with the version control system, it tells us about the dynamism and logged activity of the project.

- ***Chats***
  These are a communication channel for solving queries and problems quickly. Conversations are not generally archived so it is better for complex discussions to take place on mailing lists.

Each of these tools responds to specific needs, mainly connected with communication and information management. The experience and characteristics of the community of users and developers associated with the

project will dictate the configuration and use of these tools. Nonetheless, it is worthwhile pointing out a few aspects that could be useful in most free software projects.

Mailing lists are an essential part of any free software project so we need to pay special attention to our management and use of them. It is virtually a must to have a management system for distribution lists, whose configuration and maintenance could be complicated in the early stages.

Internet resources

The most popular systems include Mailman (http://www.list.org), Smartlist (http://www.procmail.org), Ecartis (http://www.ecartisorg), Listproc (http://listproc.sourceforge.net) and Ezmlm (http://subversion.tigris.org/hacking.html).

The main options and functionalities of a distribution list management system are as follows:

- Subscription by e-mail or a web interface
- Subscription in *digest* or normal mode[62]
- Moderating
- Administrator interface
- Configuration of message headings
- File management and querying

(62) In *digest* mode, subscribers receive a regular compilation of all messages, usually once a week or once a month, while in normal mode, the messages are received immediately.

Mailing lists can also be integrated into other tools, such as the version control system or the bug tracking system, to inform of aspects such as source code changes or modifications to the error status or tasks in course.

The version control system is also essential for any free software project that hopes to create a developer community. Almost all version control systems operate through the existence of a remote copy, common to all developers, whose versions can all be consulted. Each developer has a local copy of this remote copy that he or she works on. Occasionally, the developer sends his or her modifications to the remote copy to share them with others.

The main functionalities of version control systems are:

- **Committing**: integrating the changes from the local copy to the remote copy, which will then be logged in the version control database.

- **Updating**: integrating the changes of the other developers in the local copy.

- **Checking out**: obtaining a local copy from the remote copy.

Any document or file edited in the project can and should be subject to version control, which should not be limited to source code files. The use of a version control system can be very practical for editing and sharing documentation and technical reports and generally any document created and maintained jointly.

As explained above, the bug tracking system has many other functions besides that suggested by its name. These include the tracking of all types of task, such as the implementation of new functionalities, the preparation of releases and user support.

The life cycle of a bug is usually as follows:

- **The bug is reported**: All bugs include at least a summary and initial description containing, where possible, the elements needed to reproduce it. Most bug tracking systems allow us to set up specific fields. Remember that bugs can come from both the community of users and the community of developers.
  Once archived, the bug status remains open and is not assigned to anybody. During this time, the individuals who access the database can read the bug description and ask the user or developer who reported it for more information.

- **The bug is reproduced**: based on the instructions in the bug description, somebody manages to reproduce the bug, thus validating it. In other words, we can now say that the bug is real.

- **Bug diagnosis**: in the previous phases, a developer takes responsibility for fixing the bug or somebody in an authoritative position in the project assigns it to the most suitable developer.

- **Bug assignation**:
  this must be entered in the database so that we do not have more than one developer working on fixing the same bug without realising it. It is also possible to report the expected fixing date or release in which the bug will have been fixed.

- **Bug fixing**: once the developer has fixed the bug, he or she will mark it as fixed or closed.

Bugs can sometimes be fixed quickly, so some of these phases may not be necessary. Sometimes, the bug is not really a bug and may simply be caused by incorrect use. In all events, no matter how easy the solution is, it is always a good idea to log the bug and report it correctly to users.

Another common situation is when several users report the same bug, referred to as duplicate bugs. In this case, it is best to group all of the reports into one so that we can concentrate our efforts and put all of the information in the same place.

Lastly, a bug may be reported as fixed when it has not actually been resolved, generally because the steps followed to reproduce it do not match those indicated by the user who reported the bug. In this case, the user can reopen the bug, adding all of the necessary information. There are numerous public forges offering these and other tools, ready for use in free software projects. These platforms come with a series of advantages and disadvantages.

Internet resources

The most popular public forges include SourceForge (http://www.sourceforge.net), Savannah (http://savannag.gnu.org and BerliOS.de (http://www.berlios.de). Some organisations also offer hosting for projects in their area of interest, such as Apache (http://www.apache.org) and Tigris (http://www.tigris.org).

Their advantages include their capacity and available bandwidth: the success of the project is irrelevant because the servers will always be in operation. Keeping a high-availability server running requires a lot of extra work. Moreover, the tools available on these forges have already been configured and are usually very easy to use. Obviously, the main disadvantage is the limited flexibility and configuration possibilities of the tools.

So, before starting our project, we may want to host it on a public forge but be open to the possibility of our own hosting in the future, starting by registering the name of the domain associated with the project. For example, while not the perfect solution, having a website informing about the project that redirects to a public forge for aspects of code development can be a good compromise.

3.3.3. Organisation of the community

One of the biggest differences between free software projects and proprietary software projects is the way in which the developer community is organised.

In proprietary software projects, the structure is normally that of a hierarchical organisation of the team or department in charge of development in the company. Although hierarchies can sometimes be seen in free software projects, partly due to the merits of the individual developers, the organisation of the developer community is more flexible and also stronger.

Paradoxically, one of the reasons why the developer community works as one and remains close-knit is the possibility of creating a new independent[63] project from the original project. The possibility of a free software project

(63) This is known as forkability, i.e. the possibility of *forking*.

forking is usually negative for both developers and users. It is precisely this threat that makes the community organise itself and strive to ensure that decisions are taken as a group.

In other words, the possibility of forking makes the community strive to achieve a more or less democratic consensus in major project decisions.

There are generally two forms of organisation for free software communities, although most projects end up adopting a position midway between the two. They are:

- **Organisation based on a "benevolent dictator".**
  A benevolent dictator is a figure of authority who makes final decisions with repercussions for the development of the project. Nonetheless, benevolent dictators often do not make decisions directly but act as moderators in discussions, attempting to conciliate the viewpoints of the developers and identify the most valuable contributions. Another of the actions of the benevolent dictator is to delegate experts to deal with the decisions or discussions underway. Benevolent dictators are usually developers with sufficient experience in the project and related technologies. However, they do not need to be the most expert developers; they must simply be capable of understanding the project as a whole and recognising the best contributions.

- **Organisation based on consensus.**
  The term consensus is used to refer to agreements accepted more or less tacitly by the entire community, i.e. where nobody opposes the decisions or the direction taken by the project. As a result, the process of consensus is not usually formal by any means. However, if a consensus cannot be reached on a given issue, a vote can be taken.
  Most discussions that take place during the development of a project are usually technical, so consensus is achieved when everybody agrees on an issue, such as the design or implementation of a functionality or the way to fix a bug. In these cases, a member also usually summarises the discussion at the end.

Generally all communities, particularly those based on consensus, have excellent support in the version control system, which means that they can go back and undo any decision that turns out to be incorrect.

Projects usually begin with an organisation based on a benevolent dictator and evolve towards an organisation based more on consensus as the community expands. This usually occurs at certain times in the development of the project, such as when a benevolent dictator gives up his or her position and the authority is distributed across the community, particularly among its most highly regarded members.

After a time, the conventions and agreements adopted through consensus by a community can become increasingly large, so the main points should be set down in a document that can be used as a guide and for future reference. This may include both the form of governance of the community and the conventions and recommendations for developers.

Internet resources

Take a look at the guides for the Subversion project (http://svn.collab.net/repos/svn/trunk/HACKING) or the Apache foundation (http://www.apache.org/foundation/how-it-works.html and http://www.apache.org/foundation/voting.html).

Lastly, we need to ask ourselves what role can companies play in free software communities.

On the one hand, we could have a company that wishes to start up a free software project and create a community of users and developers. And on the other, we could have a company that joins a free software project already underway. In both cases, the company must clearly define its aims with this free software project and know what its participation in the community will be.

There are many possibilities. For example, the company may seek a leadership position in the community and lead the project, or it may simply take part in discussions, participate actively in the implementation of new functionalities or have just a selection of its developers deal with the problems of its clients.

Bearing in mind the difficulties of constructing a successful free software project, it is clear that, at least in the case of projects begun by companies, the community already exists: it is formed by the developers in the company and its clients.

In these situations, the benevolent dictator model will probably be the most appropriate form of organisation, to start with at least, but we will still need to define the rules for participation in the community. The challenge lies in converting these clients into active users who will help to improve the project and in getting other developers involved.

The solution, albeit a difficult one, is to offer benefits or some form of added value to the users and developers who participate in the community.

It is good practice for the company's development team to be fully integrated in the community and to follow the development methodology for the free software project. This means that the developers must participate in the project over a long period of time in order to become familiar with the operation of the community and gain credibility in it.

3.3.4. Development

This section describes the development process for free software projects, not from a technical perspective, as this will depend on the nature of the individual projects, but from the point of view of project management and developer coordination.

With development, we need to remember that one of the differences between free software projects and proprietary software projects is the absence of a centralised organisation. For example, when the date of a new release draws near, a company can assign a certain number of resources to prepare for it. The voluntary developers forming the community, on the other hand, are not so easy to direct. Their individual reasons are different and while some may be interested in publishing a new release on time, others may only be interested in a specific functionality.

Thus, the distribution of tasks in a free software project is based mainly on the independence between them, with the general rule being that each developer works on what he or she wants when he or she wants.

Yet this approach is ideal in part, and a person or team is needed in most free software projects to coordinate all the voluntary developers. This team can be formed explicitly by the initiators of the project or the benevolent dictator or implicitly by members with more experience and influence over the community.

Some of the basic tasks of coordination, required to carry out the project successfully, include:

- **Delegating**: one of the main tasks of the project coordinators is to delegate tasks to other developers. When somebody delegates a task to another person and the latter accepts it, the benefits are two-fold: the coordinator finds somebody to do the work for him or her and this person's efforts are acknowledged in the sense that they have been entrusted with a task. Hence, the best way to delegate a task is through a channel of communication that can be seen by the entire community, always giving the option of turning down the offer.
  In this case, the coordinator must be aware of the skills and interests of the community members and direct the offers on this basis. For example, there is no sense in asking somebody to do something if they lack the necessary skills or if they already have several ongoing tasks.

- **Criticising and praising**: the correct evaluation of contributions from each project developer is very important for the creation of

a friendly atmosphere within the community. Moreover, evaluations from coordinators or higher members have greater repercussions on the community.

Hence, both criticism and praise should be used wisely. Continuous or unfounded criticisms will no doubt provoke negative reactions, as will the same sort of praise. However, in a technical discussion, detailed criticism can be constructive because it means that the person who is making it has taken the time to analyse the design or implementation being criticised.

- **Avoiding territoriality**: one situation to avoid is where some members of a community attempt to appropriate part of the project ("their part") and refuse to accept criticisms or contributions from other members. Although this attitude may appear positive at first, since these members are usually experts and spend a great deal of time on their part of the project, the long-term result is that no other developer revises the code, which can mean a loss of quality and lead to fragmentation of the community.

- **Automating tasks**: most developers generally work on one part of the code and do not know what the others are doing. Therefore, the coordinators must make it their task to obtain an outline of the project and be aware of what each member is doing. It is easy to identify a series of tasks inherent to code development that all developers must carry out and which it is often useful to automate and centralise.

  The clearest example of this is the automation of tests[64], which allows developers to make changes and experiment with parts of the code that they are unfamiliar with.

(64) Specifically, we can create a test package, a programme that runs the project software to reproduce all previously known and fixed bugs. This allows developers to make sure that they do not reintroduce old bugs that have now been fixed.

- **Treating users properly**: the existence of an active user community contributing valuable information to developers is key to the success of any free software project. However, developers and users often speak different languages, to put it one way. Many users are not familiar with software development or how free software communities operate. Developers need to be able to put themselves in the place of users and try to explain themselves as clearly as they can.

  Just remember that any user could be a contributor to the community and because the vast majority of users never address the developer community, we need to reserve special treatment for those who do. For example, when a user indicates that the documentation is incomplete, we can suggest that they complete it themselves, or when they report a bug, we can ask them to try and fix it. And of course, always thank them for their contribution, whatever it is.

- **Sharing management tasks**: besides code development, every project has a series of management tasks that become increasingly complex as the project expands. The coordinators or team that initiated the project usually take responsibility for them, but it is good to share them with other

members of the project, as we saw in the section on delegation. These tasks include:

- **Patch management**. Controlling which patches have been received and analysing them for acceptance or, as is usually the case, to detect their problems and report them to the author of the patch.

- **Translation management**. Coordinating the translation of the documentation and software.

- **Documentation management**. Keeping the documentation and the frequently asked questions or FAQs section up to date, introducing changes as they appear.

- **Bug management**. Managing the bug database, which includes ensuring its integrity and checking for duplicate bugs, among other functions.

- **Permissions management**. One of the more important management tasks is the management of permissions, i.e. deciding who has permission to commit and hence, integrate their code into the remote copy in the repository. Besides granting permissions, there is also the possibility that we will have to revoke them.
  Developers who do not have permission to commit can of course still contribute to the development of the project by producing patches to fix bugs or add new functionalities that will be analysed by the project developers and eventually incorporated. In fact, the most common mechanism for obtaining permission to commit is for a developer to contribute patches to the project until the team of developers considers that his or her contributions and knowledge of the project are valuable enough.
  To encourage the participation of new developers, it is useful to make the procedure for obtaining permission to commit and the procedure for withdrawal of this permission public and as transparent as possible.

3.3.5. Releasing and packaging

Preparing releases and packaging is, besides code development, one of the most important tasks of the entire free software project.

A new release involves changes, particularly for users. Firstly, all known bugs from the previous release will have been solved and it is highly likely that there will be new ones. There may also be new functionalities and configuration options. There may even be incompatibilities between the new version of the software and the previous ones, in the format of the data, for instance.

Since the jump from one release to the next can have important consequences – and not all of them good – one of the first aspects we need to decide on is how to identify each of the releases.

There are several conventions for this, some more creative than others, but the most common method is to number them with a series of digits separated by decimal points. For example:

- Release 3.4.1
- Release 3.4.2
- Release 3.5
- Release 4.0

The meaning of these digits can vary. Changes to the third digit usually indicate fixed bugs or minor improvements to some functionalities. Changes in the second digit usually indicate the introduction of new functionalities. And lastly, changes to the first digit indicate new groups of functionalities and no doubt important changes in version compatibility.

It is a good idea to indicate the meaning of the numbering of the releases on the project website.

Website

See the versioning of the APR project (http://apr.apache.org/versioning.html).

In addition, some releases are usually identified with the words *alpha* or *beta*, indicating their development status. For example:

- Release 3.4.1 (alpha 1)
- Release 3.4.1 (alpha 2)
- Release 3.4.1 (beta)

Generally, the word *alpha* is used to designate the first release, which allows users to access the software and all of its functionalities but for which a considerable number of bugs is expected. Users who install and run an *alpha* version usually do so to evaluate the software and report bugs to the team of developers. A *beta* version, on the other hand, has undergone far more debugging and, if it contains almost no bugs, will become the official version: this is what is known as a *candidate version*.

For developers, a free software project is in a constant releaseprocess and they always use the latest version available in the repository for development, so it can be difficult to specify the exact moment of the release.

The best practice in this case is to keep a branch in the repository containing the code that will be introduced in the next release, regardless of the trunk. This also ensures that the developers not involved in preparing the release can continue to work on the project.

Thus, one of the most important parts of the process for preparing a release is its stabilisation, i.e. deciding which changes and functionalities will be integrated into the branch of the next release. Here, the decision-making mechanism of the free software community should come into play, giving us two basic alternatives:

- Designating *a* release owner that will decide which changes to introduce in the future release.

- Voting for the changes to be introduced in the future release, for which we will need to define the voting rules. An intermediate solution is to establish the minimum number of developers needed to vote for a certain change in order for it to be included.

In addition, one or two release managers can be appointed to integrate and validate the changes to the branch of the release.

Free software is usually distributed as source code, adequately packaged and compressed in a standard format. The name of the package is usually formed by the name of the package, the version number and the appropriate suffix for the format. For example:

- myproject-3.4.1.tar.gz
- myproject-3.4.2.zip

The information that should accompany any new release includes the licence under which it is distributed, the instructions for installation and set up, and the changes and features added since the last release. This information is included in a series of files with more or less standard names: *LICENCE* or *COPYNG*, *README* or *INSTALL*, and *CHANGES*.

And finally, users need to compile the source code and install it on their system, which should always follow a standard procedure in order to reach as many users as possible. Another possibility, used especially with mature software, is the distribution of binary packages, either as executables or installables, which does away with the need for users to carry out the compilation process manually[65].

(65) For example, the RPM or DEB system with GNU/Linux systems and MSI or self-installing executables in Windows.

From the perspective of free software companies, the releases policy is one of the key tools for reaching potential users of the software. Correct planning of releases should meet the user needs of the moment, whether by adding new functionalities or by fixing bugs, so we must determine a suitable rate of publication for new releases.

For instance, publishing new releases too often can saturate users and they will probably not install all of the releases. In contrast, leaving too big a gap between releases could encourage users to look elsewhere for solutions. It is also a good idea to guarantee the quality of the new releasesby trying to fix as many bugs as possible before their publication. The effect of a release plagued by bugs gives a very poor image of the project and the company behind it, which can be difficult to remedy afterwards. It is therefore very important to focus on the open and cooperative nature of free software in order to improve the quality.

3.3.6. Choice of licences

The differences and advantages and disadvantages associated with the various free software licences make this one of the most hotly debated topics. One thing is certain though and that is the choice of a particular licence plays a minor role in the adoption and success of the project as long as it is a free software licence. The vast majority of users choose their solution based on the functionality and quality it offers, rather than its licence.

The most important thing is to be clear on the project aims and the free software company's aims for the project. On the basis of these, we should choose the most suitable licence or create a new licence based on existing ones[66].

(66)Appendix I includes a short list of the main licences used in the production of free software.

Many free software projects have their own licences, adapted to their needs and aims[67].

(67)Examples include the OpenBravo licence (http://www.openbravo.com/product/legal/license/) or the dual MySQL licence (http://www.mysql.com/about/legal/licensing/).

Free software licences and the relationships and potential incompatibilities between them can be very complex and we sometimes have to call on lawyers or specialist legal experts for help.

See also

The *Legal aspects and the features of exploitation of free software* subject of the official Master's Degree in Free Software looks in detail at these issues.

One of the main sources of incompatibility is the reuse of free components under restrictive licences. A typical example is the GPL licence, which requires any software that uses GPL components to be distributed under a GPL licence.

It is good practice to keep an inventory or chart of the external software and licences used in the project, describing the parts of the code that use them.

Summary

Although free software technology is tested, produced and run in a variety of scenarios and despite the fact that we can easily find news on products, events or figures related to free software in the general media, there are still many clichés about its real and effective implementation.

These clichés and misconceptions often have a negative impact on the implementation of free software systems, both in domestic situations and in organisations. Users often write off free software saying that it is for computer experts or *hackers* or that free software applications are unstable, unfinished or lack the necessary support. Business, on the other hand, takes the view that free software does not protect intellectual property sufficiently, that it represents a loss of competitiveness, or that, with certain exceptions, there are no feasible business models for free software.

We can summarise most of these ideas as being concerned with a lack of quality in free software processes and products, particularly in the quality perceived by users. To a certain extent, however, it is fair to say that the history, culture and nature of free software and the community of users and developers have encouraged these ideas.

The growing commitment of governments and major organisations to free software should encourage people to question these misconceptions. Moreover, many experts and analysts have pointed out the potential of free software in bolstering the development of local, European and world economies. For example, Gartner has said that "OSS is a catalyst that will restructure the industry, producing higher quality software at a lower cost".

The materials for this subject attempt to provide a response to some of these clichés, training professionals to carry out projects implementing free software systems through the detailed study – from a conceptual, methodological and practical point of view – of this type of project in a variety of scenarios and situations.

Generally, any free software project must be treated first and foremost as a software project and secondly, as an engineering project, so these materials cannot and should not be a substitute for the necessary knowledge of these topics.

Most of the materials cover the concepts, methodologies and tools required to carry out free software development and implementation projects. This is the reason for the general tone of some of the contents, which gradually introduce the reader into the methodological features of free software projects,

attempting to organise and structure – conceptually and functionally – the main stages and milestones of implementation. We have also explained how business activity linked to free software, whether software development, consulting, integration, implementation or support, can be the object of a profitable, valid and viable business model. Similarly, and although this was not the primary aim of the subject, we have introduced the basic aspects that a business plan should cover, specifically those dealing with free software.

The constant evolution of free software technologies and projects will no doubt make some of these materials obsolete, particularly those on specific projects and solutions. However, both the methodology and the bibliographical references should help us to find the right solution for each project or implementation scenario.

Moreover, it is hoped that the formalisation and structuring of the methodologies and procedures included here, together with the collection of good practices in free software projects will make a qualitative contribution to the community and to the development and expansion of free software in general.

Finally, we could not end without thanking the Fundació de la Universitat Oberta de Catalunya (http://www.uoc.edu) for its support in producing these educational materials. We also encourage all readers who wish to send in their comments or suggestions to contact the authors so that we can improve future editions of this material and the everyday practice of implementing free software systems.

Glossary

business model Business strategy that defines, plans, produces and markets one or more products or services aimed at generating profits for its producers.

direct implementation Process by which the system to be introduced does not require the components involved to undergo complex adaptations or configurations.

free software Series of computer programs and applications whose conditions of use are subject to a free licence.

free software community Group of free software users and developers.

free software licence Licence guaranteeing the four basic freedoms: running the software for any purpose, studying and adapting the source code, redistributing copies of the software, improving the source code and publishing improvements.

infrastructure Series of basic elements or components that are correctly structured, organised and coordinated to facilitate the operation of a system.

insourcing Strategic model consisting of the delegation or production of operations or jobs in an internal department of the organisation, usually specialised, instead of outsourcing them to a third party external to the organisation.

licence Contractual model by which the author of the product (or the copyright owner) sets down the rights and duties of the users of the product and the scenario in which they can use the product.

life cycle of a project Process that covers, structures, organises and coordinates all of the stages and phases guiding the execution of a project and allows us to deal with the complexity of the aims reducing the risk of failure.

market study Analysis conducted as part of a project for a business initiative with the aim of obtaining an idea of the commercial viability of a business activity, based on the general context, competition and consumers.

methodology Systematic analysis or study of the methods and procedures that are, have been or can be applied to a given discipline.

migration Process to substitute infrastructures based on proprietary software for others with equivalent functions based on free software.

open standard Format or protocol subject to public use and evaluation that does not depend on closed formats or protocols. It does not contain clauses limiting its use, is managed independently of specific interests and is available in various implementations (or in an implementation with fair access).

outsourcing Strategic model consisting of the delegation or production of operations or jobs in an organisation external to the organisation, usually specialised, instead of delegating them to an internal department of the organisation.

packaging Process to automate the installation, configuration and uninstalling of software packages on a computer. GNU/Linux systems in particular usually use thousands of different packages.

project Organised, structured and planned process of managing resources to achieve a specific aim, usually strategic.

project management Discipline that studies the best way to organise and administer the available resources to ensure that all tasks required by the project will be completed by the set deadline with the time and costs defined previously.

release or distribution Process to make available an initial version or update a software product for its potential users.

repository A repository, pool or archive is a centralised location for the storage and maintenance of digital information, usually databases or computer files. In free software development, repositories incorporate a version control system that maintains a log of all work and changes made to the archives (mainly source code) constituting a project and allows different developers (sometimes long distances apart) to collaborate with each other.

risk Likely event that could affect the progress of the project and possibly prevent the aims from being reached on time.

strategic plan of the organisation Series of proposals defining the future aims or directions of the organisation. This is normally developed afterwards in the different areas and functional departments of the organisation.

SWOT Acronym of strengths, weaknesses, opportunities and threats. A SWOT analysis is a strategic planning tool used to evaluate the strengths, weaknesses, opportunities and threats affecting a project.

system Series of independent physical or logical elements or components that interrelate to act as an integrated, functional unit.

system services Series of functions that can be run, with or without user intervention, and which are considered essential for the operation of the organisation.

systems implementation Process through which one or more technological innovations are introduced to an organisation as the result of an action deriving from its strategic plan.

user support Series of services that either integrally or through diverse means of communication offers the possibility of managing and solving all possible issues taking place during the use of a program or application.

version Number indicating the development level of a program or application.

Bibliography

Abella, A.; Sánchez, J.; Santos, R., et al. (2003). *Libro Blanco del software libre en España*. [Consulted on 1 March 2008]. http://www.libroblanco.com/document/1000-2003.pdf

Abella, A.; Segovia, M. A. (2005). *Libro Blanco del software libre en España (II)*. [Consulted on 1 March 2008].
http://www.libroblanco.com/document/II_libroblanco_del_software_libre.pdf

Abella, A.; Segovia, M. A. (2007). *Libro Blanco del software libre en España (III)*. [Consulted on 1 March 2008].
http://libroblanco.com/document/III_libro_blanco_del_software_libre.pdf

Clearly, D. W.; Fenn, J.; Plumer, D. C. (2005). "Gartner's Positions on the Five Hottest IT Topics and Trends in 2005". [Consulted on 20 May 2008]. http://www.gartner.com/DisplayDocument?doc_cd=125868

Díaz, R. M. (2007). *El arte de dirigir proyectos* (2nd ed.). Madrid: Ra-ma, 1995.

Guitérrez, J. D. (2007). *"Metodología para el análisis decisorio de la implantación de software libre"*. [Consulted on 1 March 2008].
http://www.informaticahabana.com/evento_virtual/files/SWL14.pdf

Hecker, F. (1998). "Setting Up Shop: The Business of Open-Source Software". [Consulted on 1 March 2008]. http://www.hecker.org/writings/setting-up-shop.html

Kerchmer, K. (2005). "The Meaning of Open Standards". Proceedings of 38th Annual Hawaii International Conference on System Sciences 2005. [Consulted on: 1 March 2008]. http://www.csrstds.com/openstds.pdf

Open Formats. [Consulted on 1 March 2008]. http://www.openformats.org/

Open Source Initiative. [Consulted on 1 March 2008]. http://www.opensource.org/

Open Standards. URL: http://www.openstandards.net/ [Consulted on 1 March 2008].

Perens, B. "Open Standards: Principles and Practice". [Consulted on 1 March 2008]. http://perens.com/OpenStandards/Definition.html

Qualipso Project. URL: http://www.qualipso.org [Consulted on 20 May 2008]

Sáez, D.; Peris, M.; Roca, R.; Anes, D. (2007). *Migración al software libre. Guía de buenas prácticas*. Instituto Tecnológico de Informática.

SELF Project. URL: http://selfproject.eu [Consulted on 1 March 2008]

Free Software Foundation. URL: http://www.fsf.org/ [Consulted on 1 March 2008]

Vega García Pastor, I. de la (2004). *El plan de negocio: una herramienta indispensable*. Instituto de Empresa.

Various authors (2005). *Migration guide. A guide to migrating the basic software components on server and workstation computers* (2005). Berlin: Bundesministerium des Innern. [Consulted on 20 May 2008].
http://www.kbst.bund.de/cln_012/nn_836802/SharedDocs/Anlagen-kbst/Migrationsleitfaden/migration-guide-2nd-edition__pdf,templateId=raw,

Appendix

Appendix I: Free software licences

There now follows a short list of the main licences used for the production of free software. Some of these are discussed in detail in the materials of the Introduction to free software and Legal aspects and the features of exploitation of free software (Part II) subjects.

GNU/GPL v3

GNU/GPL is an abbreviation that stands for the *General Public License* of the GNU project.

Internet resource

For more information about the GNU/GPL licence, see http://www.gnu.org/licenses/gpl.html.

It has a robust redistribution policy called copyleft, by which all derived works inherit the original licence, even if they have been combined with others. Linking from modules with different licences is not permitted. The protection policy for original copyright, among other rights, means that the GNU/GPL licence is incompatible with other licences, including the original BSD licence and proprietary licences.

Internet resource

You will find a full list of compatibilities at http://www.gnu.org/licenses/license-list.html.

Version 3 of GNU/GPL is not directly compatible with version 2. However, many programs licensed under the second version allow the use of subsequent versions of the licence under the same terms.

GNU/LGPL v3

The abbreviation GNU/LGPL stands for the *Lesser General Public License* of the GNU project, which is a licence derived from GNU/GPL.

Internet resource

For more information about the GNU/LGPL licence, see http://www.gnu.org/licenses/lgpl.html.

This licence was originally created to allow the use, linking and integration of free software libraries with other types of licence, sometimes proprietary, to work around the restrictions of GNU/GPL licences. The practice has led to the licensing of numerous programs, some of which are now widespread. The programs licensed under GNU/LGPL include the OpenOffice.org office automation package.

Version 3 of GNU/LGPL is not directly compatible with version 2. However, many programs licensed under the second version allow the use of subsequent versions of the licence under the same terms.

BSD licence

BSD is the abbreviation of the Berkeley Software Distribution licence of the University of Berkeley.

It forms part of a group of licences (*BSD-style or BSD-like licences*, including the *FreeBSD licence*) called permissive licences because their user rights policy is rather unrestrictive. This policy, called copycenter to distinguish it from the copyleft of GNU licences, allows the commercial use of the product, its conversion to proprietary code and linking from modules with a different licence, among other possibilities.

The original BSD licence incorporates an advertising clause making it incompatible with GNU/GPL. The clause was removed in later versions, producing what is known as the *3-clause BSD licence*, compatible with GNU/GPL.

Internet resources

For more information about the BSD licence, see http://www.debian.org/misc/bsd.license. You can also see an example of a licence derived from the original BSD at http://www.freebsd.org/copyright/freebsd-license.html, called a *2-clause BSD licence* due to the elimination of two clauses of the original licence.

MPL 1.1

The abbreviation MPL stands for the *Mozilla Public License* of the *Mozilla Foundation*).

It is a private initiative that is a hybrid of the BSD and GNU/GPL licences. It is considered a permissive semi-copyleft licence because it offers the possibility of establishing proprietary licences from derived works. Linking from modules with different licences is permitted. Article 13 allows the licensing of one or more parts of the code with a different licence, called an alternative licence. Only when the alternative licence is GNU/GPL – or any other compatible licence – will the part be compatible with GPL and be able to be linked with others if they are too.

Website

For more information about the MPL licence, see http://www.mozilla.org/MPL/MPL-1.1.html.

Apache 2.0 licence

The Apache licence is a licence of the *Apache Software Foundation*).

It is quite similar to the BSD licence and is considered permissive because it allows for the possibility of establishing proprietary licences from derived works and linking from modules with different licences. The use of patents of this licence and the provisions for compensation make it compatible only with version 3 of GNU/GPL, maintaining the incompatibility with the previous two versions.

Website

For more information about the Apache licence, see http://www.apache.org/licenses/LICENSE-2.0.

X11 licence

The X11 licence, incorrectly referred to as the *MIT licence*, is a licence of the *Massachussetts Institute of Technology, MIT*).

Website

For more information about the X11 licence, see http://www.opensource.org/licenses/mit-license.php.

It is quite similar to the 3-clause BSD licence and is considered permissive because it allows derived works to be licensed as proprietary software and can be linked to modules with different licences. It is compatible with GNU/GPL and related to the X.Org project. Hence, some older versions of XFree86 continue to use it while the more modern versions use the XFree86 1.1 licence, which is incompatible with GNU/GPL because of its requirements applying to all documentation containing acknowledgements.

Website

For more information about the X.Org project, see http://www.x.org/.

CDDL 1.0

The abbreviation CDDL stands for the *Common Development and Distribution License* of SUN Microsystems.

Website

For more information about the CDDL licence, see http://www.sun.com/cddl.

It is based on version 1.1 of the MPL licence. The main differences revolve around two aspects:

- The author (or copyright owner) can restrict the legal jurisdiction of the rights and duties of the software users.
- The licence establishes the requirement of identifying all authors who contribute to the modifications made to derived works.

It permits linking from modules with other licences and the licensing of derivatives with a different licence, which can be proprietary. Its intellectual property features make it incompatible with GNU/GPL.

CPL 1.0

The CPL abbreviation stands for the *Common Public License* of IBM.

Website

For more information about the CPL licence, see http://www-128.ibm.com/developerworks/library/os-cpl.html.

Its aim is to promote the development of open source, maintaining the possibility of combining the code with other licences, including proprietary licences, though it does not permit the licensing of derivatives with another type of licence. It also prohibits the derived code from infringing the patents of the original, requiring the payment of any royalties. It is incompatible with GNU/GPL due to its clauses on the legality of derived works.

EPL 1.0

The EPL abbreviation stands for the Eclipse Public License of the *Eclipse Foundation*).

Website

For more information about the EPL licence, see http://www.eclipse.org/org/documents/epl-v10.php.

This is based on the CPL licence and has a permissive policy with a business focus. The main difference between it and CPL lies in the treatment of patent infringement by software contributors. All code licensed under EPL maintains the licence in its derived works. However, it permits the separate licensing of addenda under other types of licence, including proprietary. Linking from modules with different licences is permitted. Its characteristics regarding the permissiveness of derived works and its treatment of copyright make it incompatible with GNU/GPL.

Appendix II: Open standards

Definition

The SELF project defines an open standard as a format or protocol that is:

- subject to full public assessment and use without constraints in a manner equally available to all parties;
- without any components or extensions that have dependencies on formats or protocols that do not meet the definition of an Open Standard themselves;
- free from legal or technical clauses that limit its utilisation by any party or in any business model;
- managed and further developed independently of any single vendor in a process open to the equal participation of competitors and third parties; and
- available in multiple complete implementations by competing vendors or as a complete implementation equally available to all parties.

Website

For more information about the open standards of the SELF project, see http://selfproject.eu/OSD.

However, there is no one definition of an open standard because every organisation establishes a series of characteristics or practices suited to its particular aims. These organisations can be organisations that develop standards, supra-national councils or state governments. Some definitions require publication under *reasonable and non-discriminatory* (RAND) conditions, i.e. not totally exempt from royalties.

Example

Examples of this are the definition of the European Union or the ITU-T (http://www.itu.int/ITU-T/othergroups/ipr-adhoc/openstandards.html).

Other definitions focus more on the characteristics of the process that a standard should follow in order to be considered open, such as the recommendations of the World Wide Web Consortium (W3C), Bruce Perens or Ken Krechmer.

Internet resources

For more information about these processes, see:
http://www.w3.org/Consortium/Process.
http://perens.com/OpenStandards/Definition.html.
http://www.csrstds.com/openstds.pdf.

Organisations

The leading organisations, associations, institutes and consortiums for information technology standards are:

ANSI: American National Standards Institute.

Website

For more information about ANSI, see:
http://www.ansi.org/.

ETSI: European Telecommunications Standards Institute
(http://www.etsi.org/).

FreeStandards (*The Free Standards Group*): independent organisation that promotes the use and acceptance of free technologies through standards.
(http://www.freestandards.org/.)

ICANN: Internet Corporation for Assigned Names and Numbers
(http://www.icann.org/).

IEC: International Electrotechnical Commission
(http://www.iec.ch/).

IEEE: Institute of Electrical and Electronics Engineers, Inc.
(http://www.ieee.org/).

IETF: Internet Engineering Task Force
(http://www.ietf.org/).

ISO: International Organization for Standardization
(http://www.iso.ch/).

ITU: *International Telecommunications Union*, which groups organisations from the private and public sectors to coordinate telecommunications and global services
(http://www.itu.int/).

JXTA (*JXTA Project*): combination of open *peer-to-peer* or P2P standards with open-source Java implementations.
(http://www.jxta.org/)

OASIS (*Organization for the Advancement of Structured Information Standards*): international non-profit consortium that guides the development, convergence and adoption of standards for *e-business*

(http://www.oasis-open.org/).

OpenGroup (*The Open Group*): international consortium of vendors for the neutral advance of technology
(http://www.opengroup.org/).

RossettaNet (*Open e-business process standards*): association to promote open standards in *e-business*
(http://www.rosettanet.org/).

VoiceXML (*Voice XML Forum*): organisation of industries to create and promote *Voice Extensible Markup Language* (VoiceXML)
(http://www.voicexml.org/).

W3C (*World Wide Web Consortium*): global consortium for the promotion of Internet standards
(http://www.voicexml.org/).

WS-I (*Web Services Interoperability Organization*):
(http://www.ws-i.org/).

Open standards

We will now list the main open standards identified in the SELF project (http://selfproject.eu/en/system/files/D1_WP2.pdf).

- **Unformatted text**
  ASCII, ISO8859 (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=28245) and UNICODE (http://www.unicode.org/).

- **Formatted text**
  ODT (*Open Document Text*) and DocBook (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office).

- **Scientific text**
  ODF (*Open Document Formulae*), MathML (*Mathematical Markup LanguAge*) (http://www.w3.org/Math/) and TeX/LaTeX (http://www.tug.org/) and (http://www.latex-project.org/).

- **Images (frames)**
  JPEG (*Joint Photographic Expert Group*) (http://www.jpeg.org/) and (http://www.w3.org/Graphics/JPEG/), PNG (*Portable Network Graphics*) (http://www.libpng.org/pub/png/) and (http://www.w3.org/Graphics/PNG/), PNM (*Portable Any Map*) (http://netpbm.sourceforge.net/doc/pnm.html), GIF (*Graphics Interchange*

Format) (http://www.w3.org/Graphics/GIF/spec-gif89a.txt), BMP (*Bitmap*) (http://atlc.sourceforge.net/bmp.html).

- **Images (vectors)**
  SVG (*Scalable Vector Graphics*) (http://svg.org/) and (http://www.w3.org/Graphics/SVG/), ODG (*Open Document Graphics*).

- **Video**
  OpenEXR (http://www.openexr.com/), Theora (http://theora.org/), RIFF (*Resource Interchange File Format*) (http://msdn2.microsoft.com/en-us/library/ms713231.aspx), AVI (*Audio Video Interleave*) (http://msdn2.microsoft.com/en-us/library/ms779636.aspx).

- **Printing**
  PDF (*Portable Document Format*) (http://www.adobe.com/devnet/pdf/), PS (*PostScript*) (http://partners.adobe.com/public/developer/ps/index_specs.html).

- **Hypertext**
  HTML (*Hyper Text Markup Language*), XHTML (*Extended Hyper Text Markup Language*) (http://www.w3.org/MarkUp/).

- **Presentations**
  ODP (*Open Document Presentation*).

- **Audio**
  Vorbis (*OGG Vorbis*) (http://www.vorbis.com/) and (http://xiph.org/), FLAC (*Free Lossless Audio Codec*) (http://flac.sourceforge.net/)) and (http://xiph.org/), RIFF, WAV (Wave) (http://www.borg.com/~jglatt/tech/wave.htm).

- **Education and learning**
  LOM (*Learning Object Metadata*) (http://zope.cetis.ac.uk/profiles/uklomcore), SCORM (*Sharable Content Object Reference Model*) (http://www.conform2scorm.com/), IMS (http://www.imsglobal.org/commoncartridge.html), LD (*Learning Design*) (http://www.imsglobal.org/learningdesign/index.html).

- **Business**
  XBRL (*Extensible Business Reporting Language*) (http://www.xbrl.org/).

Case studies

Marcelo D'Elia Branco
Mónica León Martínez
Alejandro Novo López
Alberto Otero García

PID_00148381

UOC
Universitat Oberta de Catalunya
www.uoc.edu

Index

Introduction

In this module we will look at various public and private institutions that have chosen free software, either as the basis of their business, as in the case of private enterprises, or as an institutional policy, in the case of the public institutions. These case studies do not attempt to give a comprehensive picture of free software implementation as this would be impossible in a subject of this limited duration. The basic idea of the module is to show that is possible to put free software into practice beyond the specific aspects you may have seen over the course of these studies. This module therefore attaches special relevance to all aspects concerning the integration of diverse elements, from technological and practical to financial and management-based, ethical and social.

To summarise, the main aim of this module is to provide an overview of the practical implementations of free software that is somewhat different to the more specific approach used in the other subjects of this course and even in the first module of this material.

After explaining why it is impossible to describe every possible case of free software systems implementation, the following units will look at four specific cases of implementation: two from the private and two from the public sector. The first case looks at a relatively small government body: the Junta (regional government) of Extremadura. The second describes another, much bigger government body: the Federal Government of Brazil. The third case looks at the multinational Sun Microsystems. And lastly, we describe a small business in Barcelona: Cometa Technologies. We will then offer a brief summary of the features of the four case studies.

The Junta of Extremadura

In 1999, the Junta of Extremadura launched a global information society project that included a series of actions concerning the information society. These included the spread of the Internet, the development of e-government, web learning, support to SMEs in practices relating to the new economy and incorporation into the information society, and the support and integration of disadvantaged areas in the region and socially marginalised groups. As part of an attempt to provide a Debian-based Linux distribution tailored to the needs of the education sector, the project created LinEx. Within a few months, LinEx became one of the most successful examples of GNU/Linux use in Spain and in Europe.

The Federal Government of Brazil

Twenty-two percent of the population of Brazil lives in poverty. This, along with the fact that Brazil is such a large country, means that communication between the different parts is difficult and some communities can be isolated. This combination of factors led the Brazilian government to launch a series of parallel projects for digital inclusion in a range of areas. The projects involved setting up telecentres to reduce digital exclusion by improving professional skills, spreading the use of free software and encouraging people to take part in new technologies. Free software played a key role in the creation of these telecentres because it enabled the digital inclusion of individuals through the use of legal software.

Brazil has been pioneering in the use of free technologies around the world and is a point of reference for other countries in South America.

Sun Microsystems

Sun is a large multinational with offices in over a hundred and seventy countries on all seven continents. The company has adopted a clear stance in favour of open standards: the promotion of free competition through the publication of protocols and interfaces is an intrinsic part of Sun's philosophy, which has led it to work closely and actively with the free software community, contributing both source code and human and financial resources. Through this collaboration, the free software community improves or adapts programs so that Sun Microsystems can market the products with additional services, such as support and training. Sun's biggest contributions include the release of the office productivity package or suite, OpenOffice.org.

Cometa Technologies

Cometa Technologies is a private sector company that provides information technology solutions based on free software tools and standards. This case study will thus describe an example of an SME a significant part of whose business involves the use of free software. The activity of Cometa Technologies revolves around two main lines of business: the development and integration of technology solutions, and consulting and training.

Given the differences between the four situations, the format in which each case is described is also unique. We have not set down common guidelines for the four case studies, preferring instead to give the authors free rein to describe each example using their own criteria.

We hope that these examples will give you an idea of the subtleties involved in creating free software solutions in the real world.

Objectives

The aims of this teaching module are:

1. to explain the various organisations and free software projects set up by the Junta of Extremadura;

2. to confirm free software's potential in popularising the Internet and developing e-government and web learning;

3. to describe the diverse organisations and free software projects set up by the Government of Brazil;

4. to promote understanding of the social impact of free software and its potential in bridging the digital divide in disadvantaged regions and among marginalised social groups;

5. to describe the development methodology for projects launched by Sun Microsystems;

6. to raise awareness of the benefits of company participation in free software projects; and

7. to confirm the viability of free software business models by studying the case of Cometa Technologies.

1. Development and implementation of free software in Extremadura: a proposal with firm backing from the regional government

1.1. Context. The socio-economic characteristics of Extremadura

Figure 1.

The geographical features of Extremadura make it a predominantly peripheral region. Located in the far south-west of the countries of the EU, in the west of Spain, it borders with Portugal and is the centre of the triangle formed by Madrid, Seville and Lisbon.

Extremadura spans 41,634 km^2 and has a population of 1,073,904 inhabitants (663,142 in the Badajoz province and 410,762 in the Cáceres province). The population is very disperse, with a density of 25.78 inhabitants per km^2. Extremadura accounts for 8.3% of Spain's total surface area while its population makes up 2.6% of the country's total.

Extremadura is split into 383 population nuclei and 57% of its population live in towns with less than 10,000 inhabitants. The main centres are: Mérida, the administrative capital, with a population of roughly 52,110 inhabitants; Badajoz, with 138,415 inhabitants, and Cáceres, with 87,088 inhabitants. Between the 1950s and 1980s, Extremadura witnessed the migration of its population to the north of Spain, the legacy of which is still visible today: between 1960 and 1975, the region's population dropped by 22%, while that of Spain as a whole increased by 18%. In recent years, Extremadura has observed an upturn in this trend, actually registering a positive migratory balance.

In financial terms, Extremadura is one of the least developed regions of the European Union (Objective 1). The region's income per capita is the equivalent of 54% of the average income of the European Union and the employment rate is 42%.

The economy of Extremadura has evolved favourably over recent years and was Spain's Autonomous Community with the greatest relative EU convergence in 1985-1999. Extremadura has used the EU cohesion funds to set up diverse projects in the fields of education, society and business. These projects, designed to introduce the region to the revolution in new technologies and knowledge, are fostering its development on the basis of freedom and equality and raising it to a level where it can adapt to the changes ushered in by the revolution of knowledge.

1.2. The framework of the gnuLinEx (free software) project.). The global information society project of Extremadura

1.2.1. The early days: the global information society project

Extremadura has been developing a strategy for the transition towards a knowledge-based economy since 1997. The strategic project is aimed at integrating the region into the revolution in information and communication technologies and has been the priority of the current regional government's political action.

Figure 2.

In 1998, on the occasion of the debate on the region's status, the President of the Junta (or regional government) of Extremadura issued a challenge for developing a strategy that would allow the region to reach the level of

development of its wealthier neighbours by focusing on information and communication technologies and the implementation of an information society strategy. The idea was thus to instigate political and technological action to position the region at the starting line for this new revolution, in contrast to what had historically been the case in the region, when it had either lagged behind or simply not formed part of the earlier revolutions of western modernity.

One year later in 1999, the global information society project was launched under the slogan *Inventors of our destiny in the new era*, set to inaugurate a whole series of actions relating to the information society. This project introduced a first generation of actions, which included: the spread of the Internet, the development of e-government, e-learning, support to SMEs in practices relating to the new economy and incorporation into the information society, and the support and integration of the most disadvantaged areas of the region and socially disadvantaged groups.

What was once a goal is now a reality in a continuous process of change and improvement, in a region that is moving towards an open, plural and egalitarian society of knowledge. In recent years, these early information society actions have evolved to shape a regional cross-sector policy in the field that embraces business, education, citizenship and government, acting equally in all sectors while adapting to their specific needs. With a marked Community focus that takes into account European trends in this field, it has positioned itself at the fore of Europe's networked regions.

Extremadura's strategic project for access to the information society was based from the outset on the fundamental principles of connectivity and technological literacy, in an attempt to improve quality of life for the people of Extremadura through equality and freedom.

Hence, the actions carried out in the region both then and now have equipped it with a powerful communications infrastructure, a regional intranet that can connect over 1,400 points in the 383 towns of the Autonomous Community by broadband; in addition, events have taken place and initiatives and programmes launched with both educational and socio-economic goals.

From the very start, it was clear that the most important pillar of the entire process would be education and that information and communication technologies could make a decisive contribution to improving teaching quality. This led to the design and introduction of the RTE (*Red Tecnológica Educativa*, Technological Learning Network) and the design of the PAT (*Plan de Alfabetización Tecnológica* or Technological Literacy Plan), which had to meet the needs of population sectors that had arrived late to the new technological revolution. Lastly, to be certain of covering all areas of society, the region launched Vivernet, a business incubator for companies of the new digital era, and set up the CFNI (*Centro de Fomento de Nuevas Iniciativas* or Centre for

the Promotion of New Initiatives), which would study and guide the region's information society strategy in line with the changing circumstances of each moment.

The cross-cutting project in this entire process is gnuLinEx (*free software*), which was launched as a response to the need to prevent success of the entire strategy from depending on external factors which, like proprietary software, elude any form of government regulation.

gnuLinEx was designed first and foremost to guarantee the connectivity of the education community, followed by that of the regional government and private business sectors, by offering an alternative to the current market.

1.2.2. Context and framework of development of the gnuLinEx project

The information and knowledge society has to be constructed as a group project, so the use of tools that were open and accessible to everybody was not merely a happy coincidence.

Transfer of competence

The competence for the information society, telecommunications and networks and research previously exercised by the Regional Department for Education, Science and Technology was transferred to the Regional Department for Infrastructures and Technological Development by Decree 2/2005 of 11 January (published in the Official Gazette of Extremadura on 18 January 2005).

The Junta of Extremadura, through its Regional Department for Education, Science and Technology (the department charged with guiding, coordinating and evaluating the Autonomous Community's actions in the information society), considered that the best way of encouraging freedom and equality among its citizens was through technological innovation, by using and putting within everybody's reach what is not owned by anybody, with particular reference to the knowledge accumulated over time by humanity.

The aim of the global information society project was to encourage citizens to use ICTs by taking advantage of their possibilities at all levels but particularly in education and business creation, which would improve quality of life for the people of Extremadura.

Consolidation of the diverse educational actions and support to the creation of new technology businesses or the fostering of an ambitious technological literacy plan led the regional government to a point where its successful maintenance depended on an external element: the computer programs and applications used. And this set the context for the creation of gnuLinEx: the need for free software programs to complete this task, which had to be fully controllable and this would only be possible with the use of free software programs.

Brief description

gnuLinEx is a GNU/Linux distribution geared towards end users of computer equipment whose needs are basically office automation and communication tools (e-mail and web browsing). It does not require extensive IT knowledge since GNU/Linux environments have now obtained excellent levels of quality and user friendliness.

Figure 3.

gnuLinEx is based on GNU/Debian, the most secure and robust GNU/Linux distribution available, the design of which facilitates the creation of other distributions to make the most of its advantages (integration of its packages, excellent variety and quantity of software, stability, speed, security, etc.) and eliminate its disadvantages (installation and setup). gnuLinEx modifies a series of these features to adapt the distribution to the needs of the Junta of Extremadura.

It is easier to upgrade gnuLinEx than other systems because a simple command can install the software and update the entire system. Diverse packages can be installed from a range of sources: gnuLinEx portal, CD-ROMs, floppy disks, remote ftp and Internet http sites, etc.

Be legal, LinEx copy: the GNU-GPL licence

gnuLinEx uses the GNU-GPL licence (GNU public licence) held by the Free Software Foundation; this makes the source code of a program available for modification by any user; in turn, this user must then make the new sources available to all other users. The GPL licence applies to all GNU programs (among others) and hence, to all programs in the gnuLinEx distribution, which guarantees users the freedom to distribute, copy and/or modify the software.

Early approach: achieving the social and educational aims

gnuLinEx is a unique experience in software distribution and use, an action that came about as a result of the regional government's commitment to its launch and development. For the first time, a government body promoted the development and installation of software tailored to the needs of the end user, pinpointing the needs of the education community in its initial approach.

Analysis of the available software for computers installed in the education centres revealed that the aim of providing one computer for every two students would be impossible without investment from the national government, as the regional budget would be unable to cover the costs. Moreover, as future upgrades could not be guaranteed, the initial effort would be cancelled out.

gnuLinEx did not therefore come about by casual coincidence or spontaneous generation. In fact, since its launch on 17 April 2002, when the first version (LinEx v. 2.0) was presented, it has covered a dual aim:

- The educational aim of contributing to the development of the Technological Learning Network with a ratio of one computer for every two students in every classroom in schools.

- The socio-economic aim of spreading free software throughout Extremadura by means of the PAT (Technological Literacy Plan), SMEs (small and medium-sized enterprises) and the regional government itself.

1.3. Events, initiatives and situations

Extremadura's strategy of access to the information society has given rise to a series of key projects allowing it to meet the aims of connectivity and technological literacy across the population, originally proposed to improve quality of life for citizens without geographical or socio-economic restrictions.

What was, four years ago, a statement of intent and future commitment is now a reality that has materialised into a series of complementary projects with a common reference: participation in the promotion, spread and use of gnuLinEx in the development of its activities.

1.3.1. Need for a telecommunications infrastructure: the intranet of Extremadura

With the deregulation of the telecommunications market in Europe, Extremadura found itself in a situation of risk because it was very unprofitable for telecommunications companies to extend broadband infrastructures to the small towns where much of the region's population live.

However, with the launch of the global information society project, Extremadura set up numerous cultural, social, educational, economic and administrative processes that could contribute to the progress of the region and its citizens, and which created the need for an advanced telecommunications network. The main aims of this were:

- To ensure the accessibility of all citizens to infrastructures and information society services.
- To promote the technological literacy of both the rural and urban population.

Firstly, the region ensured that all schools had broadband as this would ensure that it reached all Extremaduran towns (even the smallest ones have their own school). If this had not been the case and we had been entirely dependent on market interests, we would have to have waited a very long time for the infrastructure to reach these towns – indeed, it may never have arrived.

Secondly, given that the advantages of the intranet extended beyond institutions forming part of the regional government, an attempt was made to ensure that domestic users, business men and woman and other organisations could benefit from the infrastructures created, with these services being provided by the company awarded the network contract.

Spread and features of the regional intranet

As a direct result of the established aims and arising needs, the Junta of Extremadura held a call for tenders for the installation of a corporate network of advanced telecommunications services (Ruling of 24 January 2000), which was completed in December of that year (Ruling of 27 December 2000, when Retevisión I, S. A., was awarded the contract for the advanced telecommunications network service of the Junta of Extremadura, voice/data service, as published in the Official Gazette of Extremadura on 9 January 2001).

The successful bidder began to implement the network in 2001, which will include the following connection infrastructures:

- 2 Mgb/sec. data access for the 1,478 regional government buildings, spread across the 383 towns in the Autonomous Community of Extremadura, with the possibility of increasing the size of the infrastructure and equipment for greater bandwidth at points considered necessary by the regional government.

- Internet access for a high number of simultaneous users.

- Development of a numeration plan and implementation of the switchboards required for voice/data traffic consumption in government buildings, including schools and healthcare centres in the region.

All these features unify the telecommunications tariffs and services across the regional government, cutting costs and improving the services adapted to the latest technologies. Extremadura's intranet was the first in Europe to boast these attributes.

The intranet data service of Extremadura is supported by state-of-the-art equipment for connection of the existing local area networks and the possibility of supporting virtual networks between the various buildings or bodies so that independent security policies can be established for each network.

The voice service is supported by cutting-edge digital switchboards allowing digital connection between different points within the Extremadura intranet. The design of its scope and extension is open, so new organisations and bodies associated with the Junta of Extremadura can be incorporated as needs require.

A fully equipped management system is used for these services, which provides all the necessary information on the status of the network, allowing its real time management and configuration.

1.3.2. Educating critical citizens in the information society: the Technological Learning Network

The RTE (Technological Learning Network of Extremadura), regulated by Decree 177/2001 of 20 November (as published in the *Official Gazette of Extremadura* of 27 November 2001), represents the integration of the information and knowledge society into the education system of Extremadura by promoting improvements in teaching quality, which affects both the teaching/learning methodology and the training of teaching staff and management of schools themselves.

Figure 4.

The basic aims of the RTE are:

- to guarantee connectivity between all educational establishments (through the regional intranet);
- to equip secondary schools with one computer for every two students (adapting the architecture of the old schools and creating new ones);
- to facilitate access to free, quality software and applications for use in the classroom (gnuLinEx); and
- to train and advise teaching staff in computer tools and applications and to encourage teachers to create quality content (syllabus materials that can be used in class and shared with the rest of the teaching community through the website).

From the outset, development of the RTE (which affects approximately 580 schools) has been regarded as a strategy for the extension of public information society services because it guarantees a minimum of interconnectivity (2 Mb) in all Extremaduran towns.

The RTE is being developed on four different levels:

- Communications and IT equipment infrastructure: classrooms adapted to allow two students to share a single computer in secondary schools and five children per computer at pre-school and primary school ages, and implementation of the regional intranet, which connects all of the region's schools by broadband.

- Creation of content and methodological experimentation: allows progress to be made in the educational possibilities of ICTs, fostering the creation of inter-school task forces to encourage the development of joint projects.

- Teacher training: essential for optimising both the IT equipment in classrooms (tools and applications) and the services developed alongside the latter (web space, e-mail, etc.); training can be on-site or web-based.

- Digital tools: exclusive use of free software through adaptation to gnuLinEx and development of a range of programs to meet the specific needs of the education community (students, teachers and parents). These programs are constantly changing in line with the needs that arise, as determined by the education community.

Lines of action and project development strategies

Communications and equipment infrastructure

- **Development of the regional intranet.** The RTE is supported by the intranet of Extremadura, which provides all of the necessary tools for network access from all computers for education.
  The intranet guarantees access by all educational establishments in the region to the Internet (2 MB/s). This connectivity also promotes the development of tasks and projects between diverse centres, encouraging them to produce and share educational materials, bridging distances.
  The Extremaduran intranet connects 1,478 buildings of the regional government, including all non-university educational establishments.

- **Equipment of the centres**. Each educational establishment has its own server that can manage over 500 computers in each centre, and all classrooms have one computer for every two students in secondary schools and one for every five primary and pre-school pupils. The teacher also has a computer and printer. In total, the classroom computer equipment amounts to 66,289 computers.

This makes Extremadura an international reference for the incorporation of ICTs into the classroom, supported by recent data published by the Organization for Economic Cooperation and Development (OECD) from September 2003, which place Extremadura at the fore of Europe in the ratio of students to computers.

Figure 5.

In Extremadura, the ratio of students to computers is 2.25 students for every computer, while the EU average is one for every 11 students. Spain brings up the rear in the European Union with a ratio of 15 students per computer, while the top places are held by Denmark with 3 students per computer, Sweden with 4, Norway with 4 and Finland with 5.

- Services optimising the Technological Learning Network.
  - All educational establishments have 100 MB of web space to host their sites and 15,000 e-mail accounts have been set up for teachers.
  - The software installed on the computers is free software (gnuLinEx) and incorporates application packages developed specifically for teaching: gnuLinEx-Edu Primaria (letter arranging, handwriting tutor, etc.); gnuLinEx-Edu Secundaria (calculation of percentages, verb conjugation, periodic table, etc.) and gnuLinEx-Edu FP (technical vector drawing, printed circuit board designer, accounting package adapted to Spain's General Accounting Plan, billing and point-of-sale package, etc).
  - Virtual Network Computing program, which establishes a communication protocol between a network of computers allowing one computer to control the monitor, keyboard and mouse of another computer. This means that an image of the teacher's display can be sent to the screens of either all students or a selected group, and vice versa, which allows the teacher to display the screens of all of the computers in the classroom on his or her own monitor.
  - Remote control for switching equipment on and off, meaning that students and teachers do not need advanced IT knowledge.
  - IT administrator. Every secondary school has an IT technician to maintain the school's server and provide technical support to teaching staff.
  - Creation and maintenance of the RTE site, which includes news, competitions, links to teaching resources (grouped by level and area of knowledge), and syllabus materials produced by teachers in Extremadura.

Content generation and methodological experimentation. The generation of contents is a priority and constitutes a significant element in the operation of the RTE, since the development of new syllabus materials and classroom resources form the basis of the support given to teaching staff in scheduling

their subject for the academic year. As a result, a team of teachers is being created, distributed by level and area, and with a series of common parameters and educational principles, who will be advised and supported by IT technicians.

To aid in the generation of educational content, gnuLinEx includes packages containing an innovative new tool called Squeak that can generate multimedia content without a thorough knowledge of computers. We will discuss this tool later.

Note

We will return to this point in the "Squeak on LinEx Project" section.

- **Regional training plan**
  - Teacher training. Teachers need to learn how to use ICTs in education and acquire a knowledge of new teaching roles (with the teaching tools and materials now available, we can adapt to diversity, apply more active and less theoretical methodologies, apply new continuous assessment techniques, etc).

    The motivation of teachers to use these materials will no doubt increase in line with their instrumental and teacher training and as they discover good teaching practices relating to the use of these media, which they can easily reproduce in the context of their work. The introduction of computers into the classroom has been an important step forward in the use of the many multimedia teaching resources by teachers, who now have the support of a powerful teaching ally without the need to move the entire class to the IT room. Since the year 2000, there has been an intensive, coordinated effort to boost ICT training in schools in Extremadura and encourage content generation. This training has been carried out across the region, mainly by staff at the Centres for Teachers and Resources (new technology consultants), in two forms: on-site and web-based.

During the 2001-2002 and 2002-2003 academic years, 4,301 teachers attended web-based training in Extremadura.

Three web-based activities were scheduled for the 2003-2004 academic year (one per term), attended by almost 2,500 teachers. The high demand has necessitated large-scale action, both in the number of activities and in the resources required to launch these. The main aim of the courses has been to raise awareness of the free programs in gnuLinEx and the teaching applications that it includes (gnuLinEx-Edu).

Around 350 activities have been carried out through on-site training, attended by some 7,800 teachers. Web-based training has also been a great success with teachers of Extremadura because travel and timetable incompatibilities are no longer an issue with this type of training.

A web-based training platform has also been set up with the name Campus Abierto (campus.linex.org), under a GPL licence. This platform can incorporate any advances made in platforms because it can be freely modified and adapted.

In parallel to this training plan, each Centre for Teachers and Resources schedules training activities based on the needs detected at schools within its catchment area.

Obviously, training been based on gnuLinEx since the creation and presentation of the tool, both to provide basic knowledge of the system and to explain its possibilities in teaching: image processing, multimedia, etc.

- Training of Extra-Curricular Activity Monitors in ICTs. The school day for pupils of pre-school and primary age involves classroom activities in the morning and extra-curricular activities in the afternoon. The twenty-eight extra-curricular activities offered are grouped into six categories, one of which is new information technologies. The monitors who teach the activities in this category are given special training in order to set down the basic contents common to all centres.
  Meetings and conferences are also held to coordinate and provide support to monitors, taking advantage of the extra-curricular activities held at schools in Extremadura to integrate ICTs into all areas of the syllabus, extending them to related groups (such as parent-teacher associations) through training courses, project development, etc.
  Specific and advanced training on equipment, servers, programming and web design is also given to the new technology consultants of the eighteen *Centros de Profesores y Recursos* (Centres for Teachers and Resources) in Extremadura.
  Lastly, training is scheduled for students in higher education to give them sufficient knowledge of information and communication technologies to prepare them for Internet business management and to support and advise them on business creation in the new economy.

- **The decision to use free software and gnuLinEx in schools in Extremadura**. The choice of the RTE as the first step in the distribution and use of gnuLinEx was determined by the political goal of hitting the ratio of one computer for every two students in 2005, an aim reached in secondary schools in 2003 thanks to gnuLinEx and its features.
  The use of completely free software among students, developed over the Internet by people who are physically distant from one another but with a collaborative spirit, is a lesson in itself. The decision to use free software in education was based on the many advantages it offers for teaching:
  - It can be freely used, modified or distributed; copying is legal, which means that students and teachers can legally copy it.

  - The use of a free system largely eliminates the problems of piracy. It is natural to want to share your programs with others and, with free software, this is legal.

- It is open, so the program code can be used and modified. In subjects that so require, it can be studied and improved, and we can learn from real programs with millions of users.

- It is collaborative, participative and customisable.

- It is economical.
  In general, schools around the world work with virtually no budgets and obsolete computer equipment. Due to the limitations of technical and financial resources, the impossibility of purchasing up-to-date software to ensure that students do not become illiterate in the new information and communication era poses a further problem.
  Free software offers a solution to this shortage of resources since its advantages include the fact that it has very few technical requirements (basic architecture) and it is free, making it a very valuable technical and teaching resource in schools. The features of gnuLinEx are:

- system with applications and an operating system in a single distribution;

- office automation package that contains the most common applications and supports the standard formats on the market;

- multimedia and imaging editing programs;

- Internet browsing, communication and network management programs; and

- application for the remote control of computers.

The total cost of this equipment would amount to over € 1,800 per computer on the market but with the installation of gnuLinEx and the subsequent software licence savings for each of the 66,000 plus computers for education, around € 1,000 has been saved. The complete development of gnuLinEx (including servers, distribution of copies and promotion) has cost approximately € 300,000.

gnuLinEx in the education system of Extremadura

The need to boost the RTE on the one hand and to obtain absolute control over the high number of existing pieces of computer equipment (the result of efforts to bring IT to all classrooms) on the other, meant that a stable and powerful system had to be found for working as a network and that software was needed that could be updated without having to depend on third parties, with minimal costs and tools that could be used by teachers, students and the parents of students.

gnuLinEx is also used to manage the e-mail accounts of teaching staff and to house the websites of the teacher and resource centres and the educational establishments.

To install and run gnuLinEx in educational establishments in Extremadura, the results of previous partial experiences were taken into account. The main difficulties detected then concerned the lack of full support for migrating to the new system, although we should remember that it was the first time that a government institution in Europe launched a free software distribution.

The organisational structure of the teacher and resource centres, which divides the region into eighteen sub-regions, was used. All of the new technology teaching consultants at these centres have been trained in gnuLinEx, both in technical aspects and in its possibilities for teaching. These consultants offer training support to teaching staff at the educational establishments, in collaboration with the non-teaching IT administrators at each centre.

Execution consisted of three phases:

- Training of an "avant-garde" group, composed of around a hundred teaching staff with computing experience and a number of other professionals, the purpose of which was to reflect on and experiment with the possibilities of new technologies in teaching.

- The results of the previous phase stood the project in good stead for the second, more critical phase, which took place in 2002-2003, when nineteen secondary schools from the region were fitted with the complete infrastructure. The aim of this phase was to complete the technical model (hardware and software) for each centre and the teaching model (use of the infrastructure). Initially, the greatest difficulties were to overcome the typical inertia towards the use of computer applications and certain incompatibilities with existing contents. However, in the light of previous experiences, the use of free software meant that these contents could be adapted.

- The third phase began with a costly though necessary decision: to transfer the experience to all educational establishments, extending the training programme and informative activities to all teaching staff of secondary schools, based, naturally, on the experience acquired in the previous phases.

The main problems and challenges that needed to be addressed were: the lack of a model to follow, an overwhelming infrastructure and the incompatibility of contents with gnuLinEx. In the end, however, the features of the free software guaranteed the ultimate success of the project.

Coordination has been key. The success obtained in technical aspects stemmed from the adoption of a flexible free technology model that allowed for excellent control of all resources, thus minimising maintenance requirements. In organisation and teaching, success came about in the form of teaching practices now used to provide training in use of the new classrooms to all teaching staff of educational establishments.

Squeak project on LinEx

Squeak is a free software cross-platform program (it runs on over twenty platforms) for the creation of multimedia content. On the one hand, it is a multimedia development tool that allows new users and children aged 7 and up to programme and build complex systems while, on the other, it is a tool/environment/language based on Small-talk-80 that is geared primarily to multimedia and simulation environments for IT professionals and experts.

Given this wide array of possibilities, the tool is used not only in educational contexts but also for the development of web applications and research into user interfaces and operating systems.

> Its main architect, Dr Alan Kay, is regarded today as one of the fathers of modern computing: he created the windows environment for personal computers and was the first computer scientist to use the mouse as a screen cursor.

For its features and possibilities, the Regional Department for Education, Science and Technology has incorporated Squeak into the educational context of Extremadura with the support of Small-Land. For the RTE, the installation of gnuLinEx and Squeak on the region's 80,000 computers will enable teachers to generate their own active contents for subjects and transform students into key players of the construction of their knowledge. Moreover, because it is a flexible and customisable tool, any additional features required from an educational standpoint can be incorporated.

The educational community of Extremadura has welcomed the use of this tool, with working groups being formed immediately to develop Squeak projects. Beginners' courses for teaching staff are also being held (on-site classes and distance courses using the campus abierto tool) on learning with and the possibilities of everyday use of Squeak in the classroom.

1.3.3. Technological literacy for all citizens: new knowledge centres

The PAT (Technological Literacy Plan) initiative was launched by the Junta of Extremadura in collaboration with the Asociación Regional de Universidades Populares (regional association of popular universities), local governments and other collaborating organisations with the aim of providing IT training to the entire population of Extremadura and ensuring global access to the information and knowledge society. It was designed to meet the needs of

Figure 6.

sectors of the population that arrived late to the new technology era and to educate the adult population, making it one of the region's key technology actions.

The new knowledge centres

The PAT is carried out through NCCs (*Nuevos Centros del Conocimiento* or New Knowledge Centres), public spaces with an Internet connection and: an IT technician, a social catalyst, eight Internet-ready computers and the most common peripherals (scanner, printer, digicam, etc). This equipment creates an environment whereby users and social, economic and cultural institutions can discover and experiment with the opportunities arising in Extremadura as a result of information and communication technologies.

Figure 7.

The NCCs were launched as a pilot project in 1999 with six centres based in community centres. There are now thirty-four centres, located mainly in rural areas at a certain distance from big cities or in disadvantaged urban areas as a means of social and cultural integration. Local governments have launched many initiatives to introduce them into their respective areas, as is the case of the NCCs of Navalmoral de la Mata (Cáceres) and Los Santos de Maimona (Badajoz), the latter of which uses the available resources of the town's school.

The aim is to set up a centre in every town or city in Extremadura to guarantee equal access and connectivity.

The users of the centres, citizens and organisations interact on projects taking into account the interests and demand of the population of Extremadura. As a result, they act as à la carte technology training venues while also promoting social and cultural participation.

Another interesting contribution to the PAT is that made by the itinerant IntegraRed team, which has portable centres that visit rural areas with populations of 235-800 inhabitants. These centres remain in each area for one-week periods and meet the aims of the PAT by spreading and generalising the use of ICTs.

Aims of the PAT

The main aim of the PAT is to identify the attitudes and skills of the population of Extremadura, to find out which need to be maintained and which have to be changed. The specific aims of the PAT are:

- to promote free and democratic access to ICTs among the population, allowing them to take part in the structural changes taking place in the information society in Extremadura;
- to encourage the participation of social organisations and to train the population in enterprising skills;
- to promote the creation of social and virtual spaces, fostering collaboration between institutions, organisations and populations with common interests;
- to ensure the involvement of Extremaduran society in the spread of its local and regional culture and to reinforce its collective identity through the Internet;
- to transfer the knowledge generated as a catalyst, project and plan for social improvement;
- to guarantee the use and development of gnuLinEx for equality of opportunity among all citizens, social organisations, institutions, companies and local governments;
- to promote the use of IT tools as learning and communication management systems; and
- to act as a reference for the possibilities of technological literacy and free software.

Key points of the PAT

The most innovative aspect of the PAT is its teaching/learning model, which is carried out by management technicians in the different centres (IT catalyst and social catalyst) and a number of volunteers. The model

takes its inspiration from the educational participation of adult learning (motivation, skills acquisition, participation) and the activities are designed as methodological tools to encourage learning, with a particular focus on:

- Knowledge of the environment: familiarisation with the way of life and socio-economic features of the region.
- Focus on specific needs: specific needs and interests of the users.
- Technological initiation: familiarising the user with the technological environment (what computers are and how they are used).
- Development of basic technological skills to ensure the correct use of IT tools.

The use of free software for the acquisition of basic technological skills allows users to choose from the variety of available software options. Neither cost nor difficulty are obstacles to the general use of ICTs by the population.

Research and development through pilot projects geared towards sharing and improving knowledge with communities that develop free software, improved access to the Internet and communications, etc.

Technological literacy and gnuLinEx for Extremadura

gnuLinEx, designed for use in the educational environment but made available to the entire population for private or business use, is a key component of the PAT because its main aim is to ensure universal access to ICTs among the population, without discrimination for any reason.

The support for the entire technological literacy process of Extremadura to eliminate and/or bridge the digital divide is the use of gnuLinEx; hence, since its presentation, all NCCs have used and adopted it as their working philosophy. The incorporation of gnuLinEx into all computer equipment and its knowledge and use by users has had positive results and experiences in the design of innovative activities, with the aim of spreading and raising the profile of the alternatives. The numbers of elderly citizens who learn how to use a computer or surf the Internet at these centres reveals that fears about free software systems are completely unfounded.

The use of gnuLinEx has changed the vision and role of users: they are no longer mere consumers of technology, but rather participants of a community.

Actions with gnuLinEx in NCCs:

- Migration and adaptation of the previous infrastructure of all centres.

- Distribution conferences, with videoconferencing with key figures from the free software community.
- Creation of groups for learning skills in gnuLinEx.
- Creation of technological development teams among users with an interest in free software.
- Participation in promotional fairs and events.

The following conclusions can be drawn from the experience gained with gnuLinEx as a tool for everyday activity and the technological literacy of NCC users:

- For new users, free software training is as easy and affordable as training in any other software on the market.

- For advanced users who use or have used other operating systems, the transition to gnuLinEx represents no major difficulties, as has been the case when they switched from their usual system to more user-friendly or up-to-date systems.

Transferability of gnuLinEx in the PAT

gnuLinEx has emerged as a tool allowing us to create our own developments that can be shared with other organisations and institutions and vice versa. The process of continuing technological literacy continues under the "Be legal, copy LinEx Extremadura" slogan.

The participative technological methodology and gnuLinEx depend on the people who form the information society, making it easy to adapt and use in other rural communities. The governments of Autonomous Communities in Spain such as Castile-La Mancha, Valencia and Andalusia and countries such as Brazil, India, Colombia, Nicaragua, Peru, Chile and Uruguay have all visited Extremadura to learn about and study the PAT. The upshot of this is that they are using Extremadura's experience to develop similar plans in their respective regions and/or countries.

Castile-La Mancha is developing a technological literacy plan in collaboration with the Ínsula Barataria Foundation, with a hundred centres with similar features to the NCCs.

Andalusia has launched a project called Guadalinfo to set up technological literacy centres.

Colombia is developing the LinExCol (LinEx of Extremadura for Colombia) project, a distribution created by Fundehumano with the support of the Junta of Extremadura.

1.3.4. Support to enterprisers of ICT companies: business incubators of the new era

Vivernet (Centre for New Technology Businesses) is a programme of the Regional Department for Education, Science and Technology set up in collaboration with Fundación para el Desarrollo de la Ciencia y la Tecnología (FUNDECYT, the Foundation for the Development of Science and Technology). Its aims are: to aid in the generation of new businesses in the field of the information society by offering young enterprisers with creative skills the resources they need to develop their activities, to support the technological adaptation of SMEs by offering the businessmen and women of Extremadura a vision of the new opportunities for business and management of ICTs, and to promote the use of free software in the business sector.

Figure 8.

Vivernet was launched in mid-2000 and has since contributed to the consolidation of companies in the ICT sector already operating in Extremadura, offering support and encouraging collaboration and cooperation between them. As a business centre for new technology companies, Vivernet hopes to be a reference for these companies and has set up the following lines of action:

- Working with enterprisers who wish to launch projects in the ICT sector, whether in rural or urban areas.

- Offering consulting services to guide enterprisers to success in their projects and activities.

- Holding courses on technologies and business administration, including courses on tools and free software applications that specialise in business management.

- Promoting collaboration and the exchange of ideas and experiences to encourage the creation of business networks.

- Encouraging the use and development of technological tools to adapt SMEs in Extremadura, helping them to grow and increase their competitiveness.

- Supporting and spreading the use of free software in general and gnuLinEx in particular across businesses in Extremadura. This is done through training courses, either at basic level for an introduction to gnuLinEx or at a more detailed level for specific applications such as Zurbarán (Gimp), for digital image processing.

Conferences and forums are also held on the business opportunities of free software in this sector in Extremadura for businessmen and women, enterprisers and students about to complete their studies.

Business centres: Cáceres and Badajoz

Figure 9.

Vivernet has strengthened its business support structure with two business centres for enterprisers, located in Cáceres and in Badajoz, respectively, with a virtual space (http://www.vivernet.com) and an itinerant team that takes its work to the rural parts of Extremadura. These business incubators offer the following resources and services:

- Logistical resources and technological equipment. The centres provide offices equipped with sufficient technological infrastructure for the optimal development of business activities. These offices are offered to enterprisers who opt for a business activity related to the information and knowledge society. Candidates for this option must submit a business plan and a project feasibility study that a panel of experts will study to decide who these areas will be offered to. The business centres also have other areas and common resources (fax, copying, security, cleaning, etc.) that can be used by those in charge of business initiatives, including classrooms and meeting rooms.

- Information and documentation service. This service offers a combination of information and guidance on the diverse facets of business. They also manage the vivernet.com virtual space, which produces specialised

informative newsletters and provides selective information services for the initiatives and companies based at the centres.

- Legal advice and consulting service. This service helps businessmen and women to draw up a business plan and study its economic viability. It also offers advice on the administrative procedures involved in setting up a business and on tax requirements and financial issues.

- IT and technology support service. This manages the IT systems of the centres and offers technology advice to enterprisers.

In order to achieve the aims of Vivernet with potential enterprisers in the rural parts of the region, an itinerant team of experts has been set up that travels across Extremadura to help with the drafting of new business plans and offer legal, business and technological advice.

Vivernet inaugurated the Cáceres business centre in April 2000 and in July of that year, another centre was opened in Badajoz. The itinerant team began its work in September of the same year. The results obtained since then have been: 148 business plans analysed and 70 companies given support; 3,370 hours of courses attended by 2,703 students; 4,327 registered Vivernet users and 68,934 visits from businessmen and women and enterprisers requiring its services; 12 companies hosted under the vivernet.net domain, 287,451 vivernet.com users and 4,115,208 visits to the Vivernet site.

Technological adaptation and promotion of free software in companies in Extremadura

gnuLinEx is stimulating the creation of new companies and businesses in the development of content and applications based on free software.

The Vivernet programme, whose main aims include the promotion of free software in the region's business sector, has led to the creation of seventy firms, of which 80% use gnuLinEx and 20% base their business on activities related directly to it.

Two companies from Extremadura (.DEV and Astron) recently created two programs based on gnuLinEx for the business community, Facturlinex and Contalinex, under the general name of Gestionlinex.

Along these lines, Vivernet has launched what it has called LinEx-Empresa, a space designed to promote actions to encourage the use of gnuLinEx in companies, which can be found at the gnulinex.net business site.

Vivernet has been working on two new activities since October 2003 that will extend the initial aims of the project and create new fields in which to apply the working methodology of the Vivernet team of experts.

The first activity involves the technological adaptation of SMEs in Extremadura while the second deals with the migration of companies in the region from proprietary software to the use of free software. This activity is carried out through the **business portal**, which offers a series of migration tools and utilities, besides spaces for collaboration, queries, news and repositories of free programs for business management.

The second activity is called **Linex-Empresa** and is a combination of actions designed to extend the use of gnuLinEx and free applications and programs in companies across Extremadura. It is funded by the Regional Department of Labour and the Economy of the Junta of Extremadura as part of its SME consolidation and competitiveness plan.

The purpose of Linex-Empresa is to establish and/or consolidate itself as a public reference of business unity, promoting synergies between companies and projects and actions undertaken in the field of free software through the exchange of information, the development of training actions and the fostering of business cooperation.

Moreover, Linex-Empresa seeks to become a tool of reference for the traditional businesses in Extremadura that wish to update their commercial and administrative processes or migrate their software licences to free software.

In short, Linex-Empresa is an action devised as an introduction to the use of free software in the world's business sector. It is an excellent opportunity to spread and support the regional market of software creation and development, which bases its business model on the development of applications and the provision of services, competing with the business model used by major corporations that base their profits on licence sales. Access to the source code of the applications and programs allows these to be modified and adapted to suit the specific needs of a particular company.

Through its Linex-Empresa project, Vivernet has released the full Gestionlinex management package (Facturlinex v.1.3 + integrated accounting) and the Contalinex accounting tool under a GPL licence.

Facturlinex, developed by .DEV, is a client-server application for invoicing that can be run by multiple users, which is useful for both small companies and large corporations with branches.

> It can be used in shops, whose main operations are the management of information on sales, warehouses and cash registers, or in administrative offices that need to carry out invoicing, ordering, accounting or strategic management with real-time connections between all of these activities.

When used as a point-of-sale terminal, the application speeds up ordering and invoice requests, reducing waiting time and improving customer care and the competitive efficiency of the company. The application can control the movement of the entire warehouse, cash register and invoicing, all at a competitive price because of the savings made with free software licences. Version 1.3 of the application offers improved database access and modules for monitoring size and colour; it comes with an integrated accounting application the result of which is the Gestionlinex package, the ideal solution for the commercial management and daily accounting of SMEs, all under the gnuLinEx distribution.

Gestionlinex has become the alternative and ideal solution for account management and daily business management on the gnuLinEx and other Linux distributions.

ContaLinEx, developed by Astron and released under the GPL licence for gnuLinEx, is an accounting tool providing commercial and accounting management solutions for any form of company that runs on any GNU/Debian 3.0 (or higher) environment. The possibilities of this program include editing of the general ledger, journal, profit and loss accounts, balance sheets, etc.

Extremadura is home to companies that have developed and continue to develop free software or which offer services related to them. Other companies from the region have migrated their systems and adopted this type of program for their daily management and the training of their employees.

The following companies either develop free software or provide related services:

- PuntoDev GNU S.L. This company located in the city of Badajoz develops free software for SMEs.

- Ilkebenson S.L.L. A company with head offices in Cáceres and Badajoz that produces free administrative software with a customised distribution.

- Adaptia. Company located in Cáceres that specialises in free software such as GNU/Linux and gnuLinEx.

- Silex Consultores. New company based in Badajoz that offers solutions for gnuLinEx.

The following companies have adopted free applications created in Extremadura, specifically the Gestionlinex suite, for the administration/management of shops and points of sale:

- Perfumery-Drug Store (Cáceres)

- Nuovapelle (Madrid, Badajoz, Cáceres, Don Benito, etc.)
- Granja El Cruce (chain of stores selling poultry products in Extremadura)
- Cava de Puros Ángel (Badajoz)
- Sexy Cats (Badajoz)
- Alfonso, Bebé y Modas (Almendralejo)
- Amaya, Centro de Modas (Almendralejo)

1.3.5. Strategy for transforming the information society into a society of knowledge: Centro de Fomento de Nuevas Tecnologías

The early days of the strategy: the Infodex project

Infodex (Estrategia Regional de Sociedad de la Información, Regional Strategy of the Information Society) is a project launched in 1997 that formed the strategic framework and starting point in the history and development of all actions that have guided Extremadura along the way to the information society.

Infodex was an inter-regional cooperation project funded by the European Union and the Junta of Extremadura through FUNDECYT and set in the context of the RISI (Regional Information Society Initiative).

In its first phase (1997-2000), Infodex conducted a study of the situation, potential and challenges faced by the region in its use and spread of information and communication technologies.

The conditions of diverse sectors were analysed in order to learn about Extremadura's ability to deal with the impact of the information society. This diagnosis resulted in a master plan, which included the main programmes that Extremadura needed to carry out in education, telecommunications, government, health and business.

In its second phase (2000-2001), Infodex identified and performed specific actions showing that it was possible and profitable for Extremadura to form part of the ICT revolution through the proposal and development of specific pilot projects.

Its third and final phase (2001-2002) was geared towards support for consolidation of the previous phase, which included the following actions:

- The telecommunications network, materialised through its intranet.
- The extension of the intranet of Extremadura to the diverse schools of the regional government in all towns and cities of the region and the technological development of the Technological Learning Network.
- Human resources training in ICTs through the NCCs, Vivernet and the Technological Learning Network.

- The generation of contents for the network.

At the end of the Infodex project, the actions, projects, training activities, relationships and collaborations set up and built on throughout the latter required new actions if we were to reach the aims set at the project launch. This led to the creation of the CFNI (*Centro de Fomento de Nuevas Iniciativas*, Centre for the Promotion of New Initiatives).

New strategic programming: the CFNI

This programme of the Junta of Extremadura was launched as the logical evolution of the Infodex project, designed to establish and introduce a new strategic schedule and continue the activities identifying the information and knowledge society, guaranteeing coherence with the actions carried out in the framework of the Infodex project, which was important for the development and implementation of gnuLinEx.

Figure 10.

In the first stage (2003-2003), the CFNI developed the following programmes:

- Consolidation of a regional centre for the promotion of new activities in the information society that would act as a hub for networks of information and knowledge society activities in Extremadura.

- Support to the Technological Learning Network and generation of contents for it.

- Execution of the e-Extremadura programme (a regional ERDF 2000-2006 programme for innovative actions).

- Development of gnuLinEx.

The results of the centre's activities have all been very positive:

- The e-Extremadura programme has mobilised all of the region's public and private agents (regional, local and provincial governments, universities, non-profit organisations and SMEs), which have launched more than eighty innovative projects in the field of ICTs.

- Support for the Technological Learning Network has led to its materialisation and provided schools in Extremadura with an up-to-date technological infrastructure offering the world's highest student to computer ratio. A major breakthrough has also been made in the development of educational content.

- The use of free software and the creation of gnuLinEx have been crucial to the region's progress, the universalisation of connectivity and technological literacy.

The direct and active participation of the CFNI in all of these processes has encouraged its consolidation and allowed it to carry out new activities as well as improving those carried out to date. At its current stage (2003-2004), the CFNI acts in the following capacities:

- Centre for the support and development of gnuLinEx, the main aim of which is to provide technical support to the new versions of gnuLinEx and the linex.org website, besides promoting, developing and innovating through free software.

- Regional observatory of the information society, whose main purpose will be to study, document and raise the profile of the information society process taking place in Extremadura, in order to obtain a global vision of it.

- Coordinator of the e-Extremadura programme as a continuation of the previous stage and implementer of the new programme of innovative actions.

In its new role, the CFNI will obviously continue to support the Technological Learning Network and collaborate on the generation of contents. It also carries out an important task of documentation and exposure for free software and the information society, creating networks of collaboration and cooperation either in the framework of European Union programmes and initiatives or in the national and international development and implementation of gnuLinEx.

The CFNI as a centre for the support and development of free software

The boom in free software experienced in all areas and the national and international importance and repercussions of this for the creation and development of gnuLinEx have made it necessary to bolster actions for this distribution and to continue its technical and strategic improvement.

The CFNI is responsible for the technical coordination and development of the new versions of gnuLinEx, working at the core of the system and on the programming and release of the various versions. In this task, it has the support of leading national and international figures in free software and the team working on the development of the other information society programmes of the region and the regional government.

The specific tasks carried out by the CFNI for gnuLinEx development and support are:

- Technical development of gnuLinEx and the new versions of the operating system, with the creation and integration of new applications tailored to the needs of the regional government, and the promotion of the Squeak tool in the design of educational content for the Technological Learning Network.

- Management of the linex.org site and technical support to users through it.

- Promotion of free software and systems migration in government bodies. The CFNI is carrying out training actions and offering advice to the regional government of Extremadura in order to raise awareness and encourage the migration of its systems to free software.

- Free software training for all sectors of the population.

- Support and collaboration with Vivernet to support companies that offer services related to free software and incentivise the incorporation of this software into their business activities.

Since the creation of gnuLinEx and in accordance with the aims set by the regional government, the CFNI carries out a task of promotion and diffusion in collaboration with the other programmes, resulting in the following actions:

- General meetings, conferences, courses and panels of experts designed to familiarise the population with gnuLinEx, free software and ICTs, and reveal the benefits and possibilities that they offer.

- Organisation of events specialising in the demonstration of experiences and examples of good practices in free software.

- Training in the use of ICTs, free software and gnuLinEx.

- Presence and participation in events specialising in the information society and/or free software, whether locally, nationally or internationally.

- Participation in task forces, particularly those to promote free software in the government of Extremadura.

- Activities promoting the regional information society strategy of Extremadura, particularly gnuLinEx, in diverse media.

As part of its contribution to technological research, it promotes innovation, quality and competitiveness in the region in order to improve the development of free software. The CFNI supports research groups from the region through FUNDECYT and in collaboration with the public and private organisations that develop R&D programmes.

The e-Extremadura innovative programme of actions that we will discuss later has been crucial for establishing contact with research groups and for attracting and carrying out projects with gnuLinEx. The project assessment criteria include the project's use of or compatibility with free software.

To sum up, the CFNI fosters access to greater knowledge of IT systems and supports technological research and development by attempting to halt the exodus of intelligence as a guarantee for the region's technological growth and greater likelihood that problems will be solved more quickly.

The CFNI and the Regional Observatory of the Information Society

Since the launch of the Infodex project and its evolution through the CFNI, its activities have always included a crucial task: analysis and documentation of the strategies, processes and evolution of the information society in Extremadura and in Spain and the world in general. It was therefore necessary to adopt a new method to continue the tasks of observation and monitoring of the evolution and changes affecting the latter: the Observatorio Regional de Sociedad de la Información or Regional Observatory of the Information Society.

The observatory is the instrument allowing us to obtain and analyse information on the level of development and use of ICTs in Extremaduran society and to spread this information across the existing regional and European observatory networks in order to collaborate with similar bodies and set indicators and comparative studies for introducing actions to improve and reach the level of other European regions.

Obviously, the observatory plays a vital role in the study of the status and evolution of free software around the world and in collaboration with the centre for the support and development of free software.

The CFNI and the regional innovative actions programme: e-Extremadura

The activities of the CFNI include helping to coordinate the regional innovative actions programme of the ERDF 2000-2006, presented by the Regional Department for Education, Science and Technology.

Figure 11.

The e-Extremadura programme, regulated by Decree 64/2002 of 28 May (published in the *Official Gazette of Extremadura* on 11 June 2002), has, on the one hand, ensured the collaboration of all regional government departments

in the programme and, on the other, introduced a new method for defining initiatives, since it is the first time that an action has been designed with the cross-sector participation of regional agents.

As explained earlier, the e-Extremadura programme played a significant role in the promotion and use of gnuLinEx, which was fundamental for the latter as a candidate innovative project in the framework of the programme. Moreover, the design of gnuLinEx took into account the general aims of this programme as the definition and design of the two were almost parallel.

Since the launch of the two calls for projects (June 2002 and January 2003), which saw the presentation of over seven hundred projects and the co-funding of over seventy, private organisations (companies and the tertiary sector), the University of Extremadura and the local and provincial governments of the region have established the following horizontal objectives with the contribution of gnuLinEx:

- Differentiated technological literacy. With this aim, the use of gnuLinEx, which can be freely copied and distributed, can reduce the digital divide by offering the population free access to ICTs.

- Generation and adaptation of digital content. The creation and adaptation of digital content with free software means that it can be used, improved and adapted to the specific needs of individual users. Content created in applications that can be used generally without influence from the program, operating system or brand used, thus allowing the user to obtain the software free of charge.

- Promotion of a new business and labour culture. gnuLinEx contributes to the promotion and increase of new business models for the generation and adaptation of contents and applications in free software.

gnuLinEx was taken into account in the project assessment and selection criteria because many of these considered it as a development platform or for literacy and learning content.

1.4. Spread and transferability of gnuLinEx (*free software*)

Free software in general and gnuLinEx in particular represent an increasingly popular alternative, not only in Extremadura but also in other regions of Spain and other parts of the world.

The *Be legal, copy LinEx* slogan is, in effect, a declaration of principles. gnuLinEx came about as a response to a series of needs that had arisen in Extremadura at a specific time in order to solve a problem associated with

achieving a set of proposed aims. gnuLinEx was initially planned to reach the whole of Extremadura and to be extended to all those who requested at a later date.

1.4.1. Actions for the spread of gnuLinEx

With the creation of gnuLinEx, a massive promotional, profile-raising campaign was launched, which resulted firstly in collaboration offers from free software professionals and user groups, including GULEX (Extremadura's Linux user group) and HISPALINUX (the association of Linux users of Spain).

Since its official presentation, the spread and distribution of gnuLinEx has taken a number of forms:

- CD-ROM copies at diverse events and on request through the linex.org site (over 200,000 copies).

- Downloads of the program from www.linex.org, or existing *mirror sites* (over 100,000 downloads from the website).

- The regional press of Extremadura, which distributed a free gnuLinEx CD-ROM in June 2002 with each copy of the newspaper.

- Specialist national magazines, such as TodoLinux and MundoLinux, which distributed a free gnuLinEx CD-ROM with their magazines in December 2002.

- Spread of gnuLinEx through its presence at various technology fairs and conferences, congresses, etc., (approximately 87 events to date).

- Courses and workshops on gnuLinEx (around 50) held for teaching staff, IT system administrators, other government employees, businessmen and women and enterprisers through Vivernet and for the general population through the NCCs.

Around 400,000 copies of gnuLinEx have been distributed in total, including those downloaded directly from the site and those on CD-ROM. Of these, the gnuLinEx Live v. 4.0 version, first distributed in SIMO TCI-03, was widely acclaimed because it could run directly from the user's computer disc drive without the need to install the program on the hard drive while offering the same number of applications and features. The purpose of this action was to give users an introduction to free software and an invitation to test out its many possibilities.

The appearance of gnuLinEx-related news in diverse national and international media also helped to raise the profile of the distribution, affording it considerable visibility, all revealed by the growing interest of technological companies and government institutions in learning about it and in the increased demand for gnuLinEx copies.

This action attempted to reach all users regardless of their age. Hence, since the start of the year, over 25,000 copies of a paper comic about gnuLinEx have been distributed among primary-school children from the region. Its cartoons explain how and why the project was launched, drawing the reader into an adventure of knowledge about the possibilities of free software and gnuLinEx applications. Linextremix, the main character, is the link between gnuLinEx and its younger users.

Figure 12.

To complement all this promotional and profile-raising activity, not only of gnuLinEx but of free software in general, the Junta of Extremadura has also encouraged reflection on the latter. Thus, diverse personalities and groups of international repute took part in various events hosted by the regional government, including Jesús González Barahona, Richard Stallman, Miguel de Icaza, José M.ª Olmo, various free software associations, etc.

1.4.2. Collaboration, cooperation and the transferability of free software and gnuLinEx technology

As explained earlier, the diffusion of gnuLinEx has led to calls for collaboration and cooperation from prestigious associations and figures, basically for the transfer of knowledge and collaboration on the technical development of free software to avoid going over ground already covered. It has also served to establish communication with other regions and countries for sharing experiences and good practices.

The alternative represented by free software is becoming increasingly stronger and popular, which is probably due to the constitution of two associations of free software users in Extremadura: GULEX (Linux/Unix user group of Extremadura) and Sinuh (LinEx user association), and the first gnuLinEx business association in Extremadura. LinEx Debs was also set up in the same way. The members of this gnuLinEx user group are regular participants of forums on the linex.org site, who decided to form a group to detect bugs and provide solutions to improve subsequent versions of gnuLinEx.

National collaboration agreements

One example of gnuLinEx transferability and collaboration agreements for research and development in free software between the Junta of Extremadura and other regions – that is also a model demonstrating the advantages of free software in the development of e-government – is the cooperation protocol signed between the regional governments of Extremadura and Andalusia

in April 2003 for the use and diffusion of free software and gnuLinEx in particular. By this agreement, the government of Extremadura facilitates the use of gnuLinEx to the government of Andalusia and establishes mechanisms of cooperation for the development of new applications and free software support and diffusion activities. This agreement resulted in the launch of the GUADALINEX project in June 2003.

Other regional governments of Spain have also shown an interest in implementing free software in their respective Autonomous Communities, following the example of Extremadura. These include the Basque Country, Aragon, the Canary Islands, Barcelona, Madrid and Valencia.

International collaboration agreements

On an international scale, though still within Europe, the French region of Nord Pas de Calais is preparing to introduce gnuLinEx in its institutions after confirming in situ the results of a two-year experience in the region and seeing from the example of Extremadura that free software is a perfectly viable alternative.

Although gnuLinEx has become a reference for the entire world as an example of good free software implementation in a government organisation, there has been a great deal of interest in the distribution from South America, a region that is keen to collaborate with Extremadura. As a result, the Junta of Extremadura has signed diverse collaboration agreements and protocols for the diffusion of free software and gnuLinEx in particular with South-American governments and institutions, including:

- Protocol for collaboration with Porto Alegre (Rio Grande do Sul, Brazil) in December 2003.

- Collaboration agreement with ILAEDES (Latin-American Institute of Education for Development) of San Salvador (El Salvador) in December 2003, with the AUGM (Association of Universities of the Montevideo Group) joining the agreement later.

- Protocol for collaboration with the Prefecture of the Department of Santa Cruz (Bolivia) in February 2004.

- Framework collaboration agreement with FACS (Fundación Augusto C. Sandino). Managua (Nicaragua) in August 2004.

- Protocol for collaboration with the Huancavelina region of Peru in September 2004.

- Protocol for collaboration with the Rectory of the Technology Institute of Sonora (Ciudad de Obregón, Sonora, Mexico) in November 2004.

Private collaborations

Actions in the private sphere include the collaboration agreement signed in May 2003 between the Junta of Extremadura and Lambdaux, at the request of this company, which was also extended to the Juan Carlos I University and the firm Infinity System. The purpose of the agreement was to share the technology that they were integrating in order to provide technical support to leading national hardware firms. This technical solution is based on the same standards as gnuLinEx, so its progress had an immediate effect on this distribution and vice versa: gnuLinEx benefitted from the incorporations required for these firms to run their products with free software. Users can now choose from two types of free software: gnuLinEx or Lux, a free operating system that is 100% compatible with gnuLinEx, created by Lambdaux under the GPL licence and pre-installed on Airis laptops. The product costs € 15 per year, which includes support and maintenance services.

1.4.3. Social and economic repercussions of the spread and use of gnuLinEx

From a financial point of view, the existence of a full software that can be legally copied, modified and distributed without legal restrictions helps to break down economic barriers such as the high cost of software licences.

Government organisations around the world invest a substantial proportion of their budgets in the purchase of commercial software and on improving and maintaining their IT systems. These expenses do not generate profits, so free software is both an alternative and a major opportunity as it allows savings to be made on licence fees. Moreover, new lines of research and development can be generated with this technology through collaboration and participation, which means that efforts and resources are not wasted.

In the experience of Extremadura, the independence and savings generated by the gnuLinEx system in the regional government has also been significant.

> It is estimated that almost € 48,000 have been saved with the migration of the systems in its various bodies to free software. This is the cost of licences for the use of the proprietary software programs that would need to be installed in a reference implementation of twenty-two computers. Hence, the use of free software for the development of e-government offers clear benefits.
>
> This expense, applicable not only to the regional government offices but to all of the IT equipment of the Technological Learning Network, would be impossible to meet and would hinder the development and success of a project of this size in Extremadura.

Moreover, the spread of gnuLinEx, largely responsible for breaking down these barriers in the region, has already brought benefits to companies in Extremadura related to ICTs, companies that have uncovered new business opportunities in free software and are taking advantage of the opportunities offered by gnuLinEx, developing their business along two lines:

- Companies that sell IT equipment have seen their profits increase with a default installation of gnuLinEx on their computers, passing on savings in licence fees to consumers.

  Megasoft System, an Extremaduran wholesale company from the IT sector is now leading the market in the sale of computers with a pre-installation of gnuLinEx instead of proprietary software. The company passed on the licence fee savings to the retail price of the product to obtain a more competitive market edge and saw its sales increase by 37.16%. The growing popularity of this type of software has recently encouraged firms such as OKI to develop it for its printer drivers.

- Companies offering traditional IT services have reoriented their business by developing improvements and adapting and migrating applications to free software.

  Projects offering added value and support to diffusion and aid to companies wishing to migrate their systems to free software include the Vivernet Free Software Business Platform (LinEx-Empresa), launched by the Regional Department of Labour and Economy of the Junta of Extremadura through its funding of the PCCP (SME Competitiveness and Consolidation Plan), which also received funding from the State Ministry of Economy. The aim of this project was to facilitate migration from proprietary software in companies. The project managed to consolidate itself as a public point of reference, uniting companies and encouraging synergies between them for the launch of free software projects and actions.

1.5. Linex.org services website

1.5.1. The functions of linex.org: support, information, social aid

The chief aim of the linex.org site is to define what gnuLinEx contains in order to control its evolution. The existing free software is organised rationally because otherwise it would be virtually impossible to provide organised technical support for it all as there is so much of it. As a result, users can find technical support with what they find on the site being what they see on their computer screens.

Technical support is fast for technical queries concerning gnuLinEx, most of which concern the installation process or how to install specific devices. However, the site offers added value in one of its top features, which allows users the possibility of easily upgrading gnuLinEx through the site, an essential service. This ensures that the latest versions and/or innovations in applications of the distribution are used.

The site also has an informative function and acts as a point of reference for topical issues, not only concerning free software and gnuLinEx, but also the evolution of the information and knowledge society in Extremadura and beyond.

linex.org has an important social function in that it has become a virtual meeting space and point of reference for gnuLinEx users.

1.5.2. What does linex.org offer?

Data recorded on the evolution of interest in the implementation of free software in diverse social and economic contexts are interesting because they reveal a significant increase in users who opt to use free software or who have already migrated their systems.

This growing interest stirred by the possibilities of free software and gnuLinEx in particular can be seen in various signs obtained through the site:

- The number of downloads and requests received for the distribution. Many gnuLinEx downloads are carried out through the site: around 100,000 downloads to date of all released distributions.

- The number of users currently registered, approximately 6,976 users.

- The number of visits to the site. Since its creation, it has received more than twelve million visits, over half of these in the last year (according to data from July 2004), which works out at an average of 700,000 visits per month. Interestingly, 72% of these visits were made through free browsers.

1.6. gnuLinEx from a technical perspective

Since the creation and launch of gnuLinEx, several versions of the distribution have been released, including Live 4.0. The new versions and the revisions of these have enabled us to improve the software, allowing it to work with hardware that previous versions had difficulties with.

gnuLinEx is a GNU/Linux distribution, a series of free applications that make up a complete operating system aimed at end users, based on the GNU/Debian distribution in the GNOME graphical environment, which is a very simple user interface that offers users a whole range of common applications and programs allowing them to get to work immediately: word processor, spreadsheets, image editor, photo retouching, website editing, browser, music and video player, e-mail, etc., and the series of educational applications included in LinEx-Edu.

gnuLinEx uses a very simplified graphical installation process in which almost all tasks are carried out automatically. As regards compatibility, the files generated by gnuLinEx applications are fully compatible with typical file formats. gnuLinEx is also compatible with virtually all available devices, such as printers, scanners, digital cameras and other peripherals.

1.6.1. Versions of gnuLinEx

The first version of gnuLinEx was v.2.0, launched in April 2002. This version was developed by a company called Ándago and based on the last stable distribution of Debian Potato 2.2, GNOME 1.4 and OpenOffice 1.0. The graphical installation took six clicks and the applications icons appeared with representative names of Extremadura. This first version marked the culmination of a project that had been in preparation for over a year, studying its possible areas of application, evaluating its advantages and disadvantages and using it as a pilot project in a number of offices of the Regional Department for Education, Science and Technology.

However, it was version 3.0 of gnuLinEx that proved the most popular. In the summer of 2002, two lecturers from the University of Extremadura, José Luis Redrejo and Antonio Ullán, GNU/Linux users for more than six years and considered to be the architects of gnuLinEx, began to create an improved version of the distribution, the result of which was an excellent distribution with over 200 different applications and some specialist applications requested by users.

Version 3.0 of gnuLinEx is based on Debian Woody 3 and GNOME 2.0, with numerous backports to adapt it to the graphical environment. The installation process was improved (based on progeny) and included educational programs for primary and secondary schools and the packages containing the Squeak authoring tool. A number of revisions were made of this version: 3.0r1, 3.0r1+ g2.2, 3.0r2, the latter of which was also presented at SIMO 2003 along with the Live version, developed by José Ángel Díaz, bootable from a CD-ROM, with improved visual themes and based on metadistros.

Version 3.0 of gnuLinEx also had a version called Free, with 100% free software (no Flash or Java).

The last version to appear was gnuLinEx 2004, based on Debian Sarge, GNOME 2.4, kernel 2.6 and OpenOffice 1.1.

The first revision of this last version was gnuLinEx 2004 rc2, which contains kernel 2.6.7. and GNOME 2.6. There have been many improvements both to the applications and utilities and to the graphical appearance. They include the following:

Figure 13.

- First stable distribution using component technology, allowing the production of packet blocks with their own logic and conditioners and facilitating the mixing of versions and distributions.

- Improved installation process (based on the Anaconda port). The combination of Anaconda and component technology allows for customised installations.

- Based entirely on GNOME; the KDE configuration tools have disappeared.

- All applications have been translated, although the help for some programs is still only available in English.

- The names and icons of the applications are the same in Brazil, Andalusia and Extremadura. Their appearance can be changed back to their original status.

- Includes the series of Linex-Edu educational applications, classified by subject.

There is a second revision, gnuLinEx 2004rl, whose improvements include:

- The possibility of upgrading from Menu applications (fingerprint).

- The official repository was upgraded from gnuLinEx 2004 to gnuLinEx'2004 rl with an image of GNU/Debian SARGE, which means that the 10,000 plus applications of the latter can also be installed.

- Others, such as:
  - Genome 2.8.2
  - New Bootsplash
  - Evolution 2.0
  - Mozilla Firefox 1.0
  - K3b 0.11.17
  - Numerous bugs fixed
  - New drivers for USB modems and winmodems
  - etc.

1.6.2. gnuLinEx 2004 applications

A list of the applications contained in each of the gnuLinEx utilities now follows, with the customised name of the application, the original name and a brief description of what it does.

Accessories

- Almazara (File Roller 2.6.1). Application for creating, viewing, modifying or unpacking files, folders or compressed subfolders. Offers a unique GUI and uses commands such as .tar, .gzip and .bzipg2 for filing operations.

- Diana (Gedit 2.6.0). Application for creating text files. With additional Gedit modules, diverse text editing tasks can be carried out inside the application window.

- Mapuche. Unicode character map.

- Nebrija (Online dictionary 2.6.0). Online dictionary for searching definitions of words in a database.

- San Salvador (Gcalctool 4.3.51). Powerful calculator with different modes (basic, scientific and financial) for solving a wide range of mathematical problems.

Images

- Alhambra (gthumb 2.3.2). GNOME Image viewer and browser. For browsing the hard drive displaying files with images and viewing images in a variety of formats.

- Aliseda (Sodipodi 0.33). Application for processing vector images. Works with SVG (Scalable Vector Graphics) formats but can also export illustrations to PNG (Portable Network Graphics) for editing with any retouching program such as Gimp.

- Azteca (Ghostview Gnome 2.6.1). Previews documents in PostScript and Portable Document Format (PDF) format.

- Cuzco (Gnome 0.112 PDF viewer). Viewer for PDF files based on Xpdf.

- Dehesa (XSane 0.92). Application for scanning documents.

- Hervás (Camorama 0.17). Program for viewing, editing and saving images from a web camera.

- Morales (DIA 0.93). Program for drawing structured diagrams. Application for basic diagram editing. It has a tool panel and a range of specific pre-designed and classified symbols that can be incorporated directly into the diagram at a single click.

- Picasso (Eye of Gnome 2.6.0 in the new 2.4.1). Program for viewing and cataloguing graphics files in various formats (.bmp, .gif, .ico, .jpeg, .png, .pnm, .ras, .svg, .tga, .tiff, .xbm and .xpm). It offers various levels of magnification and full-screen viewing using little memory and Bonobo technology for embedding graphics in other GNOME applications.

- Zurbarán (Gimp 2.0.1). Cross-platform graphics processing tool (drawing, composition and photo retouching).

System tools

- 112. Creates an emergency disk for rebooting the computer.

- Aljibe (system registry viewer 2.6.0). Application for monitoring and viewing system registry files. To view system registry files, it may be necessary to start the session as a primary user.

- Almenara. Displays an icon indicating when documents are being printed.

- Amizada (Gnome System Tools 0.33.0). Configuration of network devices and connections).

- Arco (floppy disk formatter 7.10.4).

- Azarías-Internet (Foomatic-gui 0.7.1). Installs printer drivers from http://www.linuxprinting.org.

- Boto (root terminal). Opens a terminal as an administrator using gksu to prompt for the password.

- Caipora (Samba Server Configuration Tool 1.2.2). Creates, modifies and deletes Samba partitions. Allows Microsoft and Linux networks to exist simultaneously.

- Candil (system monitor 2.6.0). Application for viewing current processes and monitoring system status.

- Cazorla (Gnome System Tools 0.33.0). Tool for configuring system start-up. Used to specify the operating systems that users can choose on start-up if the hard drive is partitioned.

- Cortázar (NFS server configuration tool). Creates, modifies and deletes NFS (Network File System) partitions.

- Hurdes (VNC viewer). Tool for connecting to another computer running VNC and controlling the latter.

- Marwan. Application for running the computer as another user. Opens a dialog window in which to enter a command to be executed as indicated by the user.

- Peropalo. Starts the session as another user without quitting the current session.

- Picota (Gnome System Tools 0.33.0). Management of system users and groups. Allows us to view the available users and to create new users.

- Potosí (Synaptic 0.48.2). Tool for installing, deleting and upgrading software packages.

- Puchero (Gconf 2.6.1). Configurations database editor. GConf is a CORBA-based system for storing configuration information, known as "key-value pairs". It forms part of the GNOME project.

- Séneca (Gnome System Tools 0.33.0). Configures the services to be run on system start-up.

- Tentudía (Gnome System Tools 0.33.0). Time and date settings.

- Terminal (GNOME terminal 2.6.1). GNOME terminal emulator application for the following actions: access to UNIX shell in the GNOME environment and/or to run any application designed to be run on the VT102, VT220 and xterm terminals.

- Trajano (GDM, Gnome Display Manager 2.4.4.4). Graphical application that configures the Gnome display manager.

gnuLinEx tools

- Upgrade gnuLinEx. Application for automatically upgrading gnuLinEx.

- Menu configuration tool. Configures certain menu tricks.

- Nautilus configuration tool. Configures certain Nautilus tricks.

- Wine configuration tool. Application that lets you configure Wine, the Windows emulator.

- Gnome session configuration tool. Configures certain Gnome tricks.

- Original icons. Application for changing the customised names and icons in gnuLinEx back to the program's original ones and vice versa.

Internet applications

- Algar (gFTP 2.0.17). Multithreaded FTP client.

- Corniche (PAN 0.14.2.91). Application for reading Usenet news and managing news groups.

- Giralda (Chestnut Dialer 0.0.6). Configuration of telephone network access.

- Grulla (Mozilla Firefox 0.8): Netscape-based web browser.

- Guaraní (GnomeMeeting 1.0.2). Audio/videoconferencing application for calls to remote users over the Internet.

- Hurdes (VNC viewer). Tool for connecting to another computer running VNC and controlling the latter.

- Iguazú (WebDownloader for X 2.5 Orc3). Application for downloading files from the Internet.

- Mágina (Firestarter 0.9.3). Tool for configuring firewalls.

- Medellín (Nvu Mozilla 0.2). Application for designing and editing web pages.

- Terrona (GAIM 0.77). Instant messaging program.

Multimedia applications

- Amazonia (Volume Control 2.6.1). Mixer for audio devices.

- Brasero (CD ECLiPt Roaster 2.2.0-0.8). Tool for burning audio and data CDs.

- Camarón (CD player 1.547.0). Application for playing audio CDs.

- Ceres (Rhythmbox 0.8.3). Application for organising and playing music files in mp3, FLAC or Ogg/Vorbis.

- Doñana (Audio Recorder 2.6.1.). Application for saving and playing sound files in .flac, .ogg and .wav format.

- Fluxus (Mplayer 0.90). Application for playing audio and video files.

- Monfragüe (X Multimedia System 1.2.10). Cross-platform multimedia player for audio files.

Office productivity applications

- Alcántara (Open Office Impress 1.1.1). Open Office presentations editor for creating slides, transparencies, etc.

- Brocense (AbiWord 2.0.1). Word processor.

- Cáparra (Gnumeric 1.2.11). GNOME spreadsheet.

- Espronceda (Open Office Writer 1.1.1). Word processor. For designing and producing text documents with graphics, tables and diagrams. It also allows documents to be saved in a variety of formats, including Microsoft Word, HTML and Adobe PDF.

- Glifo. OpenOffice.org printer administration.

- Guadalupe (Ximian Evolution 1.4.6). E-mail manager.

- Guaiba (Gnome MDB Viewer 0.6pre1). Graphical interface for MDB tools. For viewing and exporting databases created with MS Access 97/2000/XP.

- Iulipa (MySQLControlCenter 0.9.3-Beta). Professional tool for MySQL administration.

- Macondo (OpenOffice 1.1.1. HTML). HTML editor based on an OpenOffice template.

- Ovando (OpenOffice 1.1.1. templates). OpenOffice tool for creating documents from a template.

- Porto Alegre (Open Office Calc 1.1.1). OpenOffice spreadsheet application that can be used to calculate, analyse and manage data. It can also import and modify Microsoft Excel spreadsheets.

- Quipú (Openoffice.org Math 1.1.1). Application for creating mathematical formulas, with numerous operators, functions and format help.

Programming tools

- Chivato. Programming tool (Bug Buddy 2.6.1.). Graphical Gnome utility for bug reporting.

Other applications

- Desktop preferences. For the configuration of devices such as audio, keyboard, mouse, desktop background, screensaver, etc.

- Games. Comecocos, Frozen-Bubble, GCompris, etc.

- Almuzaffar (Nautilus 2.6.3). Graphical Gnome shell for viewing and administrating system files and applications and for viewing available network servers.

- Mochilero (Yelp 2.6.1). Gnome help viewer.

- Startup folder. Displays the start-up folder in the Nautilus file manager.

linex-edu: educational applications

- Astronomy: kstars (desktop planetarium).

- Geography: Kgeografía (geography learning program), Kworldclock (map of time zones), Sunclock (sun clock and time zones).

- Languages: Khangman (game based on the familiar hangman), Klettres (application for learning the alphabet in a different language and reading simple syllables), Kmessedwords (game based on letter and word puzzles).

- Children: memory game, Mr. Potato, Gcompris educational suite, Tux Paint, Tux Type.

- Language: kverbos (application for learning verb forms in Spanish).

- Maths: Kig (interactive geometry for KDE), kBruch (small program for generating fraction exercises), kmplot (application for creating mathematical functions), kpercentage (program for learning how to calculate percentages), Xabacus (simulation of the age-old calculator).

- Music: Audacity (advanced audio editor), Grip (audio track reader/ripper), Rosegarden (musical composition and MIDI sequencer), Solfege (musical note identification program).

- Chemistry: gperiodic (periodic table of the elements and information on them), kalzium (periodic table of the elements and information on them, in Spanish).

Others: FlashCard (application based on the traditional method of learning with flashcards, KEduca (application interactive tests based on forms), KTouch (application for learning to touch-type).

2. Free Software in the Brazilian Government

2.1. Preface. Tentative steps on a long road

Brazil is on its way to entering the information society. However, as in other areas of the planet, the different segments of its society are not being incorporated at the same speed. The Brazilian elites have been quick to connect to the global computer network. Virtually all leading industrial and trading companies and financial capital use computer communication as an additional way of stimulating, updating and expanding their business. The problem lies with the poorer levels of society, which make up the bulk of Brazil's population, whose inclusion in the digital era is slow and comes up against the obstacles posed by the traditional concentration of income into the hands of the few in Brazilian society.

Sergio Amadeu da Silveira

Chairman of the ITI (Information Technology Institute) and coordinator of the Technical Committee for the Implementation of Free Software of the Government of Brazil. Sociologist with a Master's Degree in Political Sciences from the University of São Paulo with the dissertation "Poder en el Ciberespacio: Estado-Nación, control y reglamentación de Internet" ("Power in Cyberspace: Nation State, Control and Internet Regulation"). A doctoral candidate at the University of São Paulo, he is studying democratic theory in the information era. He is the author of *Exclusión Digital: la miseria en la era de la Información* (Digital exclusion: destitution in the information era), published by Perseu Abramo. He supervised the *Software Libre e Inclusión Digital* (Free software and digital inclusion) selection of the publishing house Conrad and wrote one of its texts.

Marcelo Branco has attempted to show that information technology is not neutral. Here, we are presented with an important description of the efforts being made by a series of government institutions and Brazilian militants to reshape models of technology by focusing knowledge on a new paradigm that will create a more balanced information society.

In the information society, sharing knowledge is the equivalent of sharing wealth. It is the fundamental basis for the development of a more democratic information society with a less concentrated power. Hence, when we talk of free and open source software we are referring to a new technology policy. The facts, ideas and expectations detailed here by Marcelo Branco are the equivalent of a search for a government policy for the knowledge and mastery of a technology that will allow for a wider distribution of the positive options that can be guaranteed by a knowledge-based society.

Free software is moving forward both in Brazil and the rest of the world. This progress is not uniform and has its dangers. We need to be aware of the powerful interests being stirred up and pushed to one side when we adopt a stance in which we wish to share the essence of a piece of software: its

source code. We will see here that protocols and software are essential for the networked communication of society. Whether it be a language, a mediator of human intelligence or an invention or commodity of considerable financial value, software must be free. Freedom is only possible with autonomy. This is what Branco attempts to show in this text.

Sergio Amadeu da Silveira

2.2. Introduction

This text was written at the request of the Universitat Oberta de Catalunya (UOC), which required material on free software in Brazil to use as part of its International Master's Degree in Free Software.

The aim here is to describe something of what is happening in Brazil to promote free software in the government, using explanations and statements from top government officials. I am wholly responsible for the structure and content of this text, which does not necessarily coincide with the official position adopted by the governments involved.

Due to deadline constraints and the sheer size of the country, many stories about Brazil will be left untold in this first edition and there will probably be some errors or inaccuracies in the text itself.

Our aim with this text is to create a starting point for a dynamic process of construction in which new cases will continue to be added and efforts will be made to update the ones described in this first edition.

Thus, we are open to criticisms and suggestions from the free software community.

We would like to thank the Universitat Oberta de Catalunya for the opportunity to recover part of the recent history of our country in the knowledge that this could help to construct a new information society.

2.3. Free software in the Brazilian Government

"We must satiate the thirst for knowledge. We must promote digital inclusion as a matter of urgency."

"I consider the debate on the potential and challenges of new information and communication technologies to be of great importance. These technologies are opportunities for improving communication, dialogue and progress among our countries. [...] It all comes down to solidarity and our collective

will. All peoples have the right to advances in human intelligence and creativity to stimulate their own progress and well-being. [...] We will make digital inclusion a powerful weapon for social inclusion.

The Brazilian Government's dialogue with civil society is decisive. [...] We must satiate the thirst for knowledge. Access to technological progress must be a universal right – not a privilege of the few.

We must promote digital inclusion as a matter of urgency.

The speed of technological transformations can cause us to miss out on opportunities.

So I have taken the initiative of making digital inclusion a state policy [...].

Free software meets these requirements. Its main merit lies in promoting the transfer of technology among individuals and nations, helping everybody to form part of the information society."

Address of Luiz Inácio Lula da Silva, President of Brazil, in Africa
(available at: http://www.softwarelivre.org/news/2794).

2.3.1. The Government's reasons for the implementation of free software

During the first few months of Lula's term in office, important changes were made to the Brazilian Government's e-programme, coordinated by the Chief of Staff, José Dirceu. Two technical chambers were set up: the Technical Chamber of Free Software Implementation and the Technical Chamber of Digital Inclusion. The ITI (Information Technology Institute), an organisation reporting directly to the *Casa Civil* (Cabinet Office) of the Brazilian Government, was charged with coordinating the Federal Government's migration to free software. This operation was not delegated to a government agency but carried out by the President's Office itself, which reveals just how much of a priority the government initiative was in the plans for the information society.

The sociologist and free software activist, Sergio Amadeu da Silveira, was the administrator and coordinator of the successful telecentre programme of São Paulo, based entirely on free software. Da Silveira was chosen to chair the ITI and hence guide the implementation of free software in the government.

Website

ITI: www.iti.gov.br

The Brazilian Government's main reasons for introducing a free software implementation project were macroeconomic factors, the guarantee of increased security for government information, the enhanced technological

autonomy and capacity of the country, greater independence from suppliers and support for the socialisation of technological knowledge as an alternative for developing countries.

Macroeconomics

Each year, over € 752 million leave Brazil destined for the payment of software licences on a domestic market that moves € 2,255 million a year. This means that a third of the funds managed by the software industry in Brazil are passed on in the form of *income* to the monopolistic software giants.

Besides increasing the deficit of the balance of services, in this situation, the development of IT companies in Brazil is unfeasible. According to Sergio Amadeu, "to computerise the population, small businesses and town halls, the use of a proprietary platform would require sending more *income* abroad. And the more computerised the population (mainly with the use of basic software), the greater this sum would be. And yet, there are stable, secure and very cheap alternatives to be found in free software".

Less than 8.6% of Brazil's population is connected to the Internet and official sources state that over 53% of these users use illegal software without the authorisation of their owners. Hence, under intellectual property laws, they are considered criminals. The straightforward decriminalisation of this segment of Brazil's population with proprietary software would mean sending more than double the current amount for *income* abroad. Considering that we need to increase the number of digitally included individuals, the proprietary software option is clearly unfeasible from a macroeconomic perspective.

Secure information

Security and privacy were two other key factors in the Government's decision. The Government must guarantee the security of public information systems and the privacy of the data of its citizens. To achieve this, access to the source code of the programs used is essential. Without the source code, it is impossible to audit programs to check that they only do what the manufacturer says they do or whether there is a backdoor allowing breach of data privacy. Moreover, it is impossible to correct program errors without the source code, other than by contacting the proprietary manufacturer.

"By definition, closed software cannot be used by a government and considered to be secure software because the government does not have access to its source code. The development model of proprietary software in security terms is very outdated. The most obvious proof of this is that Microsoft partially opens its source code to governments so that it can remain on the market. However, these governments are always at the mercy of this foreign developer and when a new version is released, they must go through the new code again. In addition, the government technician allowed access

to this code is subject to draconian confidentiality agreements. As a result, governments are not interested in using software over which they do not have *full auditability*. The watchword in logical security is full auditability and this is yet another reason for our decision to opt for free software", says Sergio Amadeu.

Technological autonomy and independence

The basic capital of the digital revolution and information society is digital knowledge itself. In other words, "digital outsiders" are those who do not have access to digital knowledge.

How can we include Brazil in the knowledge society if its universities, businesses, research centres, governments and society are not fully aware of the technology spreading across the country? Hence, any programme for the digital inclusion or integration into the information society will only be consistent if it uses free software. Digital inclusion programmes that use secret, proprietary software are effectively programmes for "exclusion" from digital knowledge.

According to Sergio Amadeu, this was the Brazilian Government's main motivation: "In my opinion, the main reason is technological autonomy. The more a person uses free software, the more one can progress from user to developer. This is where the great potential of free software lies. Technicians from the Government, businesses and the country in general can master the essence of the software, and this is the main reason for using free software. "It is for mastery of the essence of the software, not simply for macroeconomic reasons."

Brazil does not want to be a mere consumer of proprietary technologies and products; it has the right to be an active subject in the information society.

Supplier independence

We know that due to the logics of public law governments must purchase with utmost transparency and make these purchases public using the public bidding model. Moreover, governments have the right to know what they are purchasing. Proprietary platforms create a technological dependence that keeps the competition in check, make understanding the purchased product impossible and create a market niche for the company that sold the product to the government. As we know, this contravenes the public principles of government.

"Free software allows greater independence from the distributor of the solution. When the Government purchases a free software solution, it has access to the source code and its four basic freedoms. Hence, because it has the source code, it is not tied to the developer, which guarantees interoperability in the future", explains Sergio Amadeu.

Think of it this way: the purchase of proprietary software by governments is like us buying medicines without the right to know their chemical formula, or buying processed food without being entitled to know what it is made of.

Socialisation of knowledge

To allow underdeveloped or developing countries to rise out of this historical era of dependence and subordination to the rest of the world, the current international treaties and laws on patents and *copyright*, trade names protected by the ideology of intellectual property, need to be changed.

Intellectual property protection was originally created to encourage freedom of creation by stimulating inventors and to generate benefits for society. Nowadays, it is a market niche for developed countries and their monopolies, as suggested by the Chairman of the ITI when he says, "Brazil has seen that the peoples of the world are interested in the socialisation of technological knowledge. And this is the clear stance taken by Brazilian diplomacy, which is consistent with the notion of free software. Brazil has come across many contentious issues similar to that of free software. The crux of the matter is knowledge and socialisation. Free software is born by saying: Can I be free software? To which the answer is: Only if you socialise", says Sergio Amadeu.

This is the stance that Brazilian diplomacy and the President of the country have adopted in their dealings with the rest of the world.

2.3.2. Migration planning and the free software community

One of the earliest initiatives of the ITI was to establish relations between the Brazilian Government and the free software community. This initiative began to take shape during the first "strategic planning" of the Technical Chamber of Free Software Implementation in 2003, in which activists from Brazil's free software community were invited to take part alongside government technicians.

More than a hundred and forty people were involved in this strategic planning process, which laid down recommendations, objectives and actions for the implementation of free programs in the Government. In all, eighteen recommendations, twelve objectives and twenty-nine priority actions make up the set of guidelines for migration. At an official ceremony at the Palácio do Planalto, Minister Dirceu presented the *Projeto Software livre Brasil* (Free Software Project of Brazil) with a copy of the strategic planning, which indicated the first steps taken by the Government in its adoption of free software and its relationship with the community.

Website

The strategic planning document is available at the following address:

http://www.softwarelivre.gov.br/

The Projeto Software livre Brasil is available at the following address:

http://www.softwarelivre.org

2.3.3. Training of Government technicians

The free software community was called on for a second time by the Brazilian Government. Over one week in April 2004, more than 2,000 government technicians began a free software training programme. Hackers, members of the community, were responsible for training the government employees on the 150 courses held.

In his inaugural address, Minister Dirceu stressed the importance of training civil servants in free software programs and the impact that this would have on society: "This event is the result of a group effort whose origins lie in the Executive e-Government Committee, which I am proud to chair. Its aim is to train technical staff of the Federal Government, thus spreading the free software culture across the country, specifically in the civil service, in order to make it more efficient and inclusive. We believe that the Federal Government's decision to opt for a system allowing the free modification and redistribution of software programs will enable us to eventually cast off the technological shackles imposed by the monopoly of a handful of companies and develop our own software that will meet our needs more adequately", added the Minister.[5]

Website

Minister Dirceu's address at the inauguration of the civil servant training week. Available at:

http://portal.softwarelivre.org/news/2047

"What was meant to be a straightforward training exercise turned out to be a huge event that brought together numerous professionals from across the country, giving us all the opportunity to share knowledge with the civil servants", pointed out Marlon Dutra, hacker and activist of the Free Software of Brazil project. "I am here to teach the 'Comprehensive Open LDAP Training' course. The whole community is very happy to be taking part in this event, which will no doubt go down in the history of Brazil, and we hope that the experience will be repeated. We know that this is helping the government to finally change the future of free software for all of us. It is a great honour for us all to be a part of this history", concluded Dutra.

2.3.4. Migration plan and strategy

The aim of the migration strategy of the Brazilian Government was to "free" workstations. Instead of starting with the big information systems and migration of the large databases which, as one can imagine, would take years

to produce a visible result, the ITI chose to migrate the personal computers of the ministerial employees. Migration of the larger systems would require a consistent, long-term plan that would take years to complete and which would not alter the logics of the technological dependence of government purchases.

Sergio Amadeu explains: "In the migration plan of a private sector company under the total control of a group of shareholders, migration begins with the company's structural systems and works its way down to the workstations. If the Federal Government of Brazil had done this, we would have been as surprised as the German troops were at the Battle of Stalingrad. We use extremely complex structuring systems that would take a long time to migrate and re-write. But while we are writing the new system, the legacy is growing, whether it be in a databank or as basic software for workstations".

The Government's plan turns the logic on its head by defining a strategy with three basic guidelines:

- Release of all workstations.
- New system developments use free software.
- Migration is initially only carried out with systems for which the above two operations are not possible.

The alternative to using applications that cannot run on stations with GNU/Linux is to build a web interface through which the user can obtain browser access to them to avoid having to rewrite the application initially. "We are working with cultural change so it is important to have a very clear vision of who is going to be migrated, given that we are dealing with technological difficulties and the constraints of the proprietary model. At the same time, there are cultural changes taking place for thousands of government employees who are the system users", explains Sergio Amadeu. "We found migration to be slow because there is a daily battle to break with the proprietary software culture and the lobby of proprietary companies, which is no mean feat", he adds.

The ITI decided to concentrate its efforts on five ministries that had already begun to migrate their workstations. The aim was to create a network effect that debunked certain myths about free software. In addition to the ITI, all of whose workstations run free software, the ministries of Energy and Mines, Cities, Culture, Science and Technology and Education, are already working with free desktops and full migration of these ministries is planned for the end of the Government's term.

This does not mean that the migration plan only extends to these ministries. For instance, Radiobrás (Brazil's public broadcasting company) already has more than eighty desktops running free software. There are also several government bodies and public undertakings such as SERPRO (the state company for Federal Government data)[6] and DATAPREV (social security data company)[7], that are introducing their own migration strategies.

Website

SERPRO:
http://www.serpro.gov.br
DATAPREV:
http://www.dataprev.gov.br

We know that we are only just getting started and that there is a lot of ground to be covered to guarantee the success of our current programmes. Many battles, inside and outside the Government, have yet to be fought and they will be decisive in the spread and consolidation of this alternative. However, we can say for certain that never before has a national government made this such a public issue, helping to step up the debate and increase the understanding of free software.

The Government's stance is also helping on an international level to construct a new information society and to defend free software in the context of the United Nations, as occurred at the World Summit on the Information Society. These initiatives took on a perspective of solidarity in the official speech by President Lula on a visit to Africa in defence of a new information society and free software as a basic option for developing countries. There can be no doubt as to the information technology label defining President Lula's Government: free software.

2.3.5. *Free Guide*. The Brazilian Government's guide for migration to free software

In the migration of the Brazilian Government's proprietary platforms to free software, we have possibly one of the most experience-packed processes in the free software movement. The difficulties encountered, the mistakes made, the search for alternatives and the results obtained will contribute greatly to stimulating and consolidating free software around the world. The record of these experiences and the sharing of this knowledge with other governments and the international free software community is proving essential for the construction and fine-tuning of the "collective memory".

Website

Guia Livre. Referência de Migração para Software Livre do Governo Federal (The Free Guide. Federal Government Reference for Free Software Migration). Available at:

http://www.softwarelivre.gov.br/publicacoes/guia-livre-referencia-de-migracao-para-software-livre/?searchterm=guia%20livre

With this aim in mind, through its e-Government technical committees for the implementation of free software and legacy systems and software licences, the Brazilian Government drew up its *Free Guide*, a reference work on

Website

The *IDA guide* is available at:
http://ec.europa.eu/idabc/

migration available to all government agencies and to the general public. This important tool took its initial inspiration from the *IDA guide* (Interchange of Data between Administrations) guide of the European Community.

However, the work was restructured and received significant contributions from Brazil's free software community and the technicians of the governments involved in the various migration tasks. "At the end of December, while we were debating the actions to introduce in 2004, we agreed that it was important to create a migration guide to help with the migration of free software", explains Corinto Meffe, project manager of technological innovations of the Ministry of Planning and coordinator of the Free Guide project.

The contribution of the Free Guide and the difference between it and the EC's IDA Guide is that we added the Brazilian experience of the community and governments. We changed some things, such as the fact that they use the term 'open-source software' while we preferred to use 'free software', which is clearer for the aims of Brazil. This indicates a change in conceptual focus.

The main focus of the EC's guide is costs, which contradicts a survey conducted by the IDA in 2002, which indicated that the cost factor of the use of free software in the evaluation of public administrations of the EC was sixth in importance. The Brazilian guide, however, focuses on the four freedoms of free software. We highlight the advantages of technological independence, access to the code and collaboration with development. This marked an unusual difference in aim.

Another point is that the European guide focused on the analysis of successful cases in private companies, because there were very few cases of government authorities. We removed this and introduced the successful cases of the Brazilian Government, which are "much more important", adds Maffe.

However, as you can see, the most important change is that the EC guide says that 'the views expressed in this document are purely those of the authors and may not, in any circumstances, be interpreted as stating an official position of the European Commission'. In its *Guia livre*, the Brazilian Government accepts liability and subscribes to the collective work, attaching institutional significance to it and validating an important technical reference for migrations within and beyond the country's borders. Brazil is the first country to have an institutional document in this framework.

Collective construction and launch of the *Migration Guide*

The beta version of the guide was launched at the 5th International Free Software Forum of Porto Alegre in June 2004. The Brazilian Government presented the guide to the 4,800 plus visitors to the event and publicised it in the media and Brazilian free software community channels (websites

and lists). It underwent a process of collective creation at this event, in which the community was encouraged to contribute to the improvement, validation and creation of new topics relating to free software migration. Contributions were subsequently incorporated based on the experiences of the Federal Government and on public meetings held in various Brazilian cities. On 7 September, Brazil's Day of Independence, launch of the "Ipiranga" version was made public.

Following the incorporation of hundreds of new contributions, the initial 151-page version increased to 221 pages, freely licensed under the CC-GNU GPL (Creative Commons-GNU General Public License) and was given the official approval of the Brazilian Government.

Website

3Creative Commons - GNU General Public License:
http://creativecommons.org/license/cc-gpl?lang=en

New topics and chapters were incorporated, such as the government's institutional political relationship with free software topics and the legal reasons for migration, besides the inclusion of free tools for geoprocessing, webmail, GNOME/ KDE graphical interfaces, fax servers, *groupware*. Brazilian distributions such as Conectiva and Kurumim were also included.

Aims of the *Free Guide*

The aims of the Free Guide are:

1) To help administrators to define a strategy for a planned and managed migration process.

2) To describe how this migration can be carried out, using general technical terms.

The guidelines were drawn up to have a practical use for administrators and should therefore be relevant and precise as well as accessible and understandable. This is not a handbook of detailed technical references. The structure attempts to encourage and allow changes to be made as administrators gain in experience and confidence and as the products become available to meet their needs.

Website

For more information on the Brazilian Government's relationship with free software, visit:
http://www.softwarelivre.gov.br

3) To focus the guidelines and definitions in this guide on the models of interoperability of the Brazilian Government.

4) To create the conditions to allow these migrations to be explained in greater technical detail on the free software site of the Federal Government.

Tropicalisation and internationalisation

This wonderful process of the "tropicalisation" of free software migration experiences must continue by introducing the international community to this Brazilian contribution. Breaking down language barriers, particularly that

of the hegemonic language of the Internet, English, will be crucial to the fine-tuning of this collective and continuous effort. And Brazil is waiting for the contribution of our international community.

2.3.6. *João de Barro*: Secure GNU/Linux

The João de Barro programme is a Brazilian Government initiative to provide greater security and to guarantee the authenticity of transactions made over the Internet, as part of its quest to popularise digital certification and stimulate the growth of e-commerce. The current platform electronically certifies all clearing system operations for national financial system transactions, amounting to € 19,770 million per day. The ultimate aim of this ITI project is to guarantee the technological independence of the entire digital certification process by the end of 2005 and to replace the entire proprietary encryption platform of the Root Certification Authority with free national hard and software by the end of 2006.

Note

João de Barro is the popular name of the Red Ovenbird (*Furnarius Rufus*).

The government's chief collaborator on this programme is CASNAV (the Shipping Systems Analysis Centre of the Brazilian Navy). CASNAV is the specifier for the entire solution, since it has been using free software for encrypting solutions for over ten years.

The project, developed through a network of collaborators from government security agencies, research centres and universities, is stimulating the production and exchange of national knowledge in the field. Besides being a strategic project for the technological development and future of the Internet in Brazil, João de Barro collaborates on enhancing the security of international free software projects.

In parallel to this, the MERCOSUR countries are working to come up with a common regulation for the recognition of certificates between member countries to speed up e-commerce in the region.

Digital certification

The ICP-Brasil (*Infra Estrutura de Chaves Públicas Brasileira*, Infrastructure of Government Keys of Brazil) was set up in 2001 by the Brazilian President's Office to guarantee the technical means (hardware and software) and regulations for enabling government institutions and private sector organisations to act on the basis of the legal validation of documents produced, transmitted or obtained electronically. Digital certification guarantees the security and authenticity of these transactions.

Through its infrastructure, ICP Brasil is the only legally valid and recognised certifying authority in Brazil. In other words, only electronic transactions and documents validated by this authority are legally recognised.

Most of the messages we exchange over the Internet are not encrypted, which means that the security and privacy of these messages can be easily breached. As they are not digitally signed, it is not possible to confirm their authenticity – and whether the person sending the message really is the sender or whether the contents sent really are the ones we see.

The ICP-Brasil identification and security process uses an asymmetric key with two passwords. A private password for encrypting and signing messages, used only by the sender of the message or document, and the root, a password that will be in the public domain. With full knowledge of the root construction, it is possible to "breach" or improve the security and privacy of information.

The proprietary technology currently used by ICP-Brasil does not allow the certifying authorities full auditing or knowledge of this technology. At most, this absolute knowledge is in the hands of employees of foreign companies. Building and sharing knowledge on this subject, from hardware to web-based business strategies is essential for any nation that does not want to be left on the sidelines of the world stage.

The hierarchy of this complex structure, coordinated by a Management Committee, consists at the top level of a policy management authority called a "Root Certification Authority" or "Root CA", which reports to the ITI and is responsible for signing the certificates from the certification authorities (CA).

At the next level, we have the chain of Certification Authorities (CAs), such as Serasa, Certsing, Caixa Econômica Federal, Receita Federal (tax agency), Serpro and the Brazilian President's Office, which issues the certificate for services authorities, called Registry Authorities (RAs) found at the far end of the next level.

RAs act as digital registries, dealing with clients, individuals or companies who wish to obtain legal recognition of their digital signature. The clients visit the registry unit and request digital certification for their company. The RA completes the registration and the clients receive digital certificates with digital signatures that can be used with all electronic transactions they wish to make with these certificates, which will guarantee the authenticity of the operation.

The quest for a more secure GNU/Linux

The João de Barro programme carries out a series of parallel projects and has created a number of different products that can be integrated into the free Root CA encryption platform. One of these involves the customisation and improvement of the security of the GNU/Linux operating system, which will be specifically applied to the João de Barro programme and other sensitive government areas.

This task is coordinated by ABIN – the Brazilian Information Agency – and will involve several Brazilian universities who will work on auditing, *bug* detection, security holes and customisation of the GNU/Linux operating system. "ABIN is locating and identifying competent individuals from academia to collaborate on the secure GNU/Linux project. Individuals with the right profile discovered in universities are being invited to take part in the programme. Around forty-two people are estimated to be working on the auditing of the operating system code", says Ricardo Valle, coordinator of the João de Barro programme.

"The universities taking part in the agreement will audit the kernel and other features of the operating system to pinpoint vulnerabilities and validate the system so that the institutions can produce a distribution of the operating system for sensitive areas of the Federal Government, such as the *Receita Federal* (Brazil's tax agency) and the federal police force.

These collaborators will attempt to locate vulnerabilities and contact the individuals who maintain the various packages in order to obtain fixes or improvements. I believe that these collaborators will very often be able to find a solution to fix a bug that has been found or come up with something else that could improve functionality", adds Ricardo Valle.

This project will have two by-products: the first is a finely-tuned and debugged operating system that will only contain the functionalities required for the HSM (*hardware security module*), which is the cryptographic security hardware. "This version will be able to fit on a floppy disk, because anything superfluous will be cut out. The software will only contain what is needed for the hardware", says Ricardo Valle.

The other by-product is the operating system that will be run on the workstation that controls and contains the Root CA applications. This version of GNU/Linux will be more complete and have more features than the HSM system.

The secure GNU/Linux project will thus help to improve the operating system overall and be able to offer its results to the development community. The project involves a specific form of collaboration between the government, universities and the free software community. It has also enabled performance of a scientific and technological development project in strategic areas of interest for the State, namely security and the use of free software, and it has extended the command of technology to diverse parts of Brazil and to academia.

The secure GNU/Linux project and the João de Barro programme as a whole have been designed to promote knowledge of cryptography and digital certification in Brazil. The budget for this project amounts to € 750,000.

A new cryptographic hardware

The heart of the structure is housed in a server cabinet in the Palácio do Planalto, where it deals with an infinite number of national security requirements. Besides the HSM (*hardware security module*), this cabinet houses an off-line workstation with the Root CA applications and the secret private key.

The HSM (*hardware security module*) that safeguards the private key of the Root CA – the starting key for the entire public key infrastructure of Brazil – is imported and proprietary. As a result, the Brazilian Government has neither the autonomy nor the knowledge to improve or maintain this equipment, which form the core of Brazil's digital certification system. This introduces an element of insecurity to the country's certification model because it depends on the supplier even for access to this private key.

The private key may be vulnerable to attack and the Brazilian Government can do nothing to improve this. In the light of this, a strategy is being adopted to develop a nationalised hardware solution. As part of a joint collaboration, the ITA (Technological Aeronautical Institute), with its vast experience and skills in hardware and electronics, will develop the cryptographic hardware to house the private Root CA key.

Free software for the Root CA

Another product under development is the free software application for the Root CA. This application is being designed for the signing of new certificates from a certifying authority (CA) enabled as part of the infrastructure and for issuing lists of revoked certificates.

The collaborator will be LABESC (Security Laboratory of the Federal University of Santa Catarina), which will lead a security task force from the RNP (National Research Network), which also includes the UNICAMP (University of Campinas) and the UFMG (Federal University of Minas Gerais). Both the ITA and the University of Santa Catarina will receive funding from the Ministry of Science and Technology. The three universities are working together on this project.

2.3.7. Casa Brasil

Digital technology to promote the integration of governmental action and create a point of reference and introduction for citizens with their government.

Like many governments have experienced, when Brazil's current government came to power, the country was suffering from fragmented actions and its traditional affliction of the separation of policies into "fiefs" in each ministry.

Social programmes and public services are often badly implemented with very little dialogue between the diverse ministries; there is little cross-cutting between programmes, which is often negative for much of the population. Moreover, in many parts of the country, the population find it hard to imagine where the government actually is. This fragmentation and the problems with connecting government programmes tends to superpose policies and action programmes, which are simply repeated instead of being complemented. The government also seeks to create synergies so that actions from different government agencies complement rather than compete with one another.

On the initiative of the Minister of the Department for Government Communication and Strategic Management, Luiz Gushiken, the Brazilian Government is beginning to take important steps to change this picture. The Casa Brasil project is a reference for the Brazilian population aimed at unifying the defining label, the presence of the government and improving the deployment of ministerial programmes by integrating state policies into a single physical space.

The gauntlet thrown down by Minister Gushiken, the architect of Casa Brasil, is to give a more consistent visibility to the Federal Government's action across the country. The unifying core of this space was information technology and the digital inclusion programmes of the Federal Government, which, though numerous, were disjointed and disconnected.

Although Casa Brasil is not just a telecentre or digital inclusion programme, all of these spaces, whether public or set up in conjunction with the general population, will be prepared for an intensive use of digital technology, particularly free software.

According to Antonio Lasance, Minister Gushiken's Cabinet Chief, the forecast is for a thousand centres to be up and running by June 2005 and for this number to increase to six thousand by the end of 2007, all with free software.

The Casa Brasil project is a sort of Lego construction that the Government is getting ready to create a series of government services with separate distributions, separate labels, etc. around a basic telecentre structure. Another of the aims of Casa Brasil is to act as an adhesion project, encouraging ministries to share these spaces. It is a point of reference that will have its own defining label and be implemented in conjunction with the community, local authorities and regional governments. Casa Brasil will use only free software and could have a major impact, particularly on poorer classes in the outer limits and remote areas of the country", says Rogerio Santanna, National Secretary for Logistics and Information Technology of the Ministry of Planning.

The Ministry of Communications GESAC satellite

One of the foundations of Casa Brasil is the GESAC (e-Government and Citizen Support) programme, which is connecting 3,200 satellite points of presence with VSAT antennas and modems. This provides high-speed Internet connections in some poor, remote communities (such as inland indigenous regions in very poor parts of the country, located at long distances from towns or cities), some of which do not even have electricity (in which case solar cells are used) or premises that are absolutely impossible to reach through the traditional corporate networks of Brazil.

Website

For more information on the GESAC programme, visit:
www.idbrasil.gov.br

The Ministry of Communications estimates that over four million individuals are currently being served by GESAC satellite antennas. "We now have more than four million people using GESAC and 800 computers running free software implementation, mainly used as servers and printers in these spaces, a free call centre (0800) for the whole of Brazil (for providing information and receiving requests for support and complaints) and a total of twelve services, entirely in free software, running at the programme's *data centre*", says Antonio Albuquerque, coordinator of the Communications Ministry's digital inclusion programme.

The free software community is taking part in the GESAC programme through CIPSGA (Committee for Incentivising GNU and Alternative Programmes), whose main tasks are providing the *call centre* service and training multipliers. "Multiplier training covers the GNU/Linux operating system and explains how to use all network applications, share bandwidth, support people at the various points in their region and state and, in short, to make the most out of the programme. The free software community also helped to prepare the GNU/Linux version with the GESAC satellite connection and completed the entire technical demarcation of the programme with the help of the Federal University of Minas Gerais", adds Albuquerque.

Website

CIPSGA: www.cipsga.org.br

Topawa Ka'a: digital inclusion with free software in the Amazon Basin

In the heart of the Amazon Rainforest of Brazil, we have witnessed one of the most wonderful experiences of contact between remote villages and state-of-the-art technologies: Topawa Ka'a, the Rainforest Digital Inclusion Network.

Over the last 500 years, the first inhabitants of South America, the indigenous people, have seen their rights denied, their culture stifled and they have been virtually eradicated from the land that they ruled with freedom and sovereignty. In the administrative region of the Amazon Basin, which covers 58% of Brazil's surface, we find the richest and most valuable part of the

original culture of this continent and many of the survivors of the massacre. We owe it to humanity to preserve this culture, restore harmonious relations with nature and guarantee the continuity of these ethnic groups.

As part of the aim of rescuing and restoring the prestige of the peoples of the Amazon Basin, the name of Topawa Ka'a was chosen, which comes from the Akawawa language, a dialect of Parakanã, from the Tupí-Guaraní family, from the linguistic branch of Tupí, which includes the Asurini and Surui languages. In Parakanã, "Topawa" simply means "network" and Ka'á means "wood, forest".

ELETRONORTE is a state company that supplies electricity to the Amazon Basin and has been charged with the project. It is now accepting social responsibility for its past liabilities of serious environmental and social damage caused by large hydroelectric works and an uncoordinated occupation of the land. The Director of Planning and Engineering and architect of the programme, Ismael Bayma, affirms that this sort of attitude is now considered unacceptable right across the world. For him, "the civilisation of Brazilians affected by these works, which did not always benefit them, deserved respect now and then. As does the environment, which belongs to us all".

Bayma explains that this was the background to creating the Forest Digital Inclusion Network. "Beneath the high-voltage cables that often pass over their heads without their benefiting from the electricity they carry, these citizens must have had their own ideas about the purpose of progress: How could the white men from the city build such works and confine the great rivers of the Amazon Basin while they, the native inhabitants or heroic forerunners, obtained no benefits from them? All these problems and all of this potential were there within arm's reach. So we decided to prioritise their social inclusion and, as a part of this, their digital inclusion", he adds.

The Director explains that, in view of the nature of the project with its vast geographical scope and technological implications, thought was immediately given to collaboration between government bodies and the necessary synergies. Bayma reveals that the first ELETRONORTE collaborator was ITI, which is linked to the Cabinet Office of the Government and had its own experience with the highly successful implementation of telecentres in São Paulo, a winning model with extensive network spread and popular support that was quickly assimilated by users in the poorest regions of the capital of São Paulo.

"Other government bodies quickly enlisted their help. We decided on a corporate optical fibre network that travels along the high-voltage lines and capitalises on this method; we also used satellite data transfer systems made available free of charge by GESAC, as well as other solutions", enthuses the leader of the Brazilian state company.

Social data

The social indicators and HDI (Human Development Index) of the Amazon Basin region are the worst in the country. The literacy rate is 9.7%, also one of the country's worst, and this is directly reflected in the level of digital inclusion. Less than 6.7% of the population have computers in their homes and only 4% are connected to the Internet. In Maranhão, for example, less than 2% of the population have access to the global computer network, making it the Brazilian state with the least digital inclusion. Another example is Pará, which is the fourth most excluded.

Website

For more information about the HDI, see:
http://pt.wikipedia.org/wiki/IDH

"The social issue of the Amazon Basin is one of the commitments of the Federal Government. When President Luiz Inácio Lula da Silva came to power, he decreed that state companies would incorporate social responsibility into the regions in which they operate", recalls Evandro Nonato de Souza Filho, coordinator of the project. This caused a change in the understanding of electricity distribution, which ceased to be simply a "business", as it was previously seen, to be considered a public service that had been turned on its head to provide a service to the citizens of the country. In the areas in which it operates, ELETRONORTE has set up regional integration programmes, classified into educational and technological development, and which include digital inclusion.

Performance of the programme

The initial idea of ELETRONORTE with this programme was to make use of the existing technological structure, with thousands of kilometres of optical fibre installed beneath electricity lines. The company's network and knowledge in the Amazon region also helped to shape the initiative.

It would be wrong of us to assume that technology is the main challenge to the implementation of a digital inclusion programme in the north of Brazil. The logistics for maintaining the operational context and set up socio-educational programmes is a difficult task. The equipment for some of the parts covered by the programme takes two weeks to arrive at its destination by boat. Awareness of these special features is proving vital to the success of a wide-ranging and bold project such as this.

Technical teams

In all states of the Amazon Basin, ELETRONORTE has teams of staff that manage the company's internal network and are being trained in free software. The Rainforest Digital Inclusion Network uses the logistics structure and knowledge of the company technicians who already know what a computer network is and have the skills to carry out maintenance. These telecentres are manned by the staff who cover the region.

Implementation of the telecentres

The Topawa Ka'a project already has three telecentres up and running: Tupiranga, Vitória del Xingú and Altamira, and ten more units are planned for the end of 2004. The initial investment for the programme is € 263,000 and the goal is to set up twenty telecentres in the northern region and in the administrative region of the Amazon Basin. Each unit has a start-up cost of approximately € 30,000, which is used to purchase computers, servers, furniture, air conditioning and to fund refurbishment of the centre. It also includes a monthly maintenance cost of € 2,650, carried out by ELETRONORTE technicians.

The project is coordinated in such a way as to optimise costs and it is seeking to increase its list of original collaborators, which are: the ITI, Ministry of Communications (which set up the satellite link), the Bank of Brazil (which has donated used computers) and the municipal authorities, which generally provide the premises with the refurbishment already complete. In these cases, ELETRONORTE's calculations indicate a 50% reduction in the sum of investments made for the programme. The project also has a private collaborator, the Internet provider IG (www.ig.com.br), which maintains the domains and e-mail accounts provided for users of the telecentres.

Infrastructure

Each Topawa Ka'a telecentre has 10-20 computers (depending on available space and number of inhabitants), all of which have free software installed on them. "The free software decision was made on the basis of the four freedoms it offers; distribution, copying, modification and distribution of the modified version", says Ana Carina Gomes de Andrade, Head of the Topawa Ka'a's Social Programmes. "We can modify programs to suit our specific needs and characteristics. It would be impossible to carry out this project with proprietary software because we would have to pay licence fees for each computer and we would not have the freedom to alter, modify and distribute the programs", she explains.

The decision for free software

The decision to use free software is in line with the guidelines of the Federal Government, a decisive factor in making the project feasible, since it needed to spread and hence the costs had to be as low as possible. This decision was also based on other concepts, such as freedom of knowledge, the ability to expand the contents of the programs – currently only available in Portuguese – "but we are studying the possibility of building a version in an indigenous dialect", says Carina.

"The possibility of creating an interface in an indigenous dialect is only possible with free software, which is easy to use and allows us to develop the distribution in the community now that many experiences have already been customised in our language, using expressions rather than letters. If we don't strengthen this relationship with our end users – the traditional communities of indigenous ethnic groups – we will fail in our aim of making technology an everyday object. If we can move forward – and this is the idea behind the project – we can create distributions that these communities can control and know what they are working with", she tells us. ELETRONORTE also took the experiences of the São Paulo telecentres and adapted them to the situation of the indigenous communities of the north.

Management council

The Topawa Ka'a telecentres have a Management Council made up of individuals from the local community, social movements, representatives of the municipal parliament (which debates their priorities) and the municipal authority of the region (which implements the actions on the premises where the telecentres have been set up). The Management Council draws up proposals for digital inclusion workshops for the community and for regional development activities, in order to provide state policies with a way in. An example of this is a request from the Ministry of the Environment to host a workshop on *quemadas* (fires lit by Brazilians to "clean" the land for growing crops), because of the number of accidents that occur in the northern region.

Strategy

The project adopts the strategy of training monitors and technical agents in all telecentres so that they can act as multipliers of the action. Training is given to a wide number of individuals and then, following an assessment, three candidates are chosen to perform paid professional tasks for the project – an administrator and two monitors – who are relieved of their positions during the centre's operation. These multipliers are inhabitants of the community, which means that Topawa Ka'a is also stimulating job creation and generating income.

Target

Each telecentre is set the initial target of benefiting four thousand people a month in the region through basic courses in computing and free computer use. The target public is the general community (indigenous and rural), whose lives are generally unstable. The interest of the region's inhabitants in the project has been overwhelming. Less than a week after the inauguration of the Altamira telecentre, for example, over three thousand people had signed up for the IT training courses and the timetables for free computer and Internet use

were fully booked up. On 13 August, the region's first 1.6 thousand inhabitants from the first four groups of the Altamira and Vitória del Xingú telecentres received their basic computing diploma.

The training includes basic computing courses (basic GNU/Linux commands), OpenOffice classes, web browsing and e-mail communication. The project also offers e-mail to all those who drop by at the telecentres.

Social concern

A new social concern that is beginning to rear its head now needs to be tackled. The arrival of technology, the participation of communities in the global computer network and their inclusion in the information society are not solving the problem of job creation. Where can we place the young people who have already been trained on the programme? There are no vacancies for office work or other typical city jobs in the region. The main purpose of a telecentre is to become a useful tool for the everyday activities of the population. They cannot create "functional digital illiterates", people who were trained to use computer tools but cannot find a way for these resources to improve their quality of life and that of the region in which they live.

Telecentres must be run from a multidisciplinary angle rather than as a basic academy of computing. Evandro explains that the second step, once students have completed the basic computing course, is for them to continue their process of digital and social inclusion. "The idea is that we convert the telecentres into digital community centres. The Altamira telecentre, which has been operating for two months, is used as a cultural centre and is proving successful", adds the coordinator with satisfaction.

This project will certainly not solve the problems of the region's digital exclusion but it is emblematic in the contradictions it generates between the digital revolution and the perspectives of this evolutionary transformation of technology, used to improve quality of life for these ancestral peoples. This project, through its many collaborations – public and private, national and international – can be regarded as one of the most significant cases of social and digital inclusion of Brazil's information society programme.

Technical features of the project

- The project uses the Debian-based GNU/Linux (www.debian.org) distribution of the telecentres of São Paulo: SACIX.

- Local area network "client-server" model using LTSP – Linux Terminal Server Project (www.ltsp.org). The twenty computers access programs run on a local server.

Website

SACIX:
www.sacix.org/

- 128 Kbps Internet connection, via satellite or fibre optic cable.
- Proxy, DNS server and local DHCP running on a GNU/Linux server.
- The project web pages are hosted by the ITI on an Apache server.
- The POP e-mail uses the structure of the Internet provider, IG, which manages the accounts and domains. IG has set up 300,000 e-mail accounts with the name of the project xx@topawa-kaa.com.br for ELETRONORTE to distribute among users of the telecentres.
- Main programs: Debian-GNU/Linux operating system, Desktop GNOME 2.6 and KDE, OpenOffice.org office applications, Ximian Evolution e-mail client, Mozilla browser, Gpaint and Gimp.
- Server: Pentium 4 or similar with 2 HDs 40 GB, 2 GB RAM, CD burner, floppy drive.
- Workstations: Celeron 800 or similar with 128 Mb RAM, floppy drive, laser printer.

Bibliographical references

For more information on the Marea project, see the website of the Special Department for Fishing and Fish Farming of the Brazilian President's Office:

http://www.presidencia.gov.br/seap/

You may also find it interesting to look at the following work, which is the source of the quotations that you will find in this section:

Adriane Lobo Costa, Gilson Ribeiro de la Silva, Josémilton Florêncio Lima, Karin Bacha, Luiz Eduardo Bonilha, Maria Luiza Gonçalves Ramos "Project for the Digital Inclusion of Traditional Fisherman and women"

2.3.8. Fisherman and women discover a new "Net". The Marea programme or fishing telecentre

In Morpará, in the state of Bahía near Xique-Xique and Bon Jesús de la Lapa, as in most of Brazil's fishing communities, its inhabitants live in a situation of isolation. They have problems receiving information and are socially marginalised. Morpará, like many communities like it, has no landline telephone and there is no mobile coverage either.

However, this situation began to change with the *Programa Maré de Inclusão Digital* (Tide Digital Inclusion Programme), launched by the Special Department for Fishing and Fish Farming of the Brazilian President's Office. The arrival of the first computers donated by the Bank of Brazil to the association of fishermen, with a broadband connection through a Ministry of

Communications satellite antenna, is enabling this community to participate in the new information society and changing the lives of many people for the better.

João da Silva, aged 60, has been a traditional fisherman for over 40 years and masters the tool of his trade, the fishing net with a skill known to few. João's greatest joy this year was to discover the new possibilities of another type of Net: the Internet. He has been separated from his son for more than six years since, upon seeing the lack of opportunity and bleak outlook of the region, his son left home to study and seek out new opportunities in the big city of São Paulo. João had kept his son's e-mail address at home, written on one of the few letters that had reached him in all these years. On the day of the inauguration of the telecentre of Morpará, once the use of e-mail had been explained to him, João wasted no time and sent his first e-mail to his son. "It was one of the most exciting things that have happened at the centre", says Gilson Ribeiro da Silva, cabinet official of the Ministry of Fishing Project Management and one of the organisers of the digital inclusion programme. It was our happiest moment because João's son replied immediately and he sobbed his heart out and said: "I want to talk to my son every day". The fishing telecentre is open 24 hours a day, so the fishermen drop by in the early morning when they return from fishing.

According to Adriana Lobo Costa, manager of the digital inclusion project of the Special Department for Fishing and Fish Farming, "some staff of the Department who know about the Federal Government's Digital Inclusion Programme realised that fishing communities could benefit from it because of the isolation in which they live, the problems they have in receiving information and because of their marginalisation. After lots of preparatory discussions, we launched the digital inclusion project, realising that it would be a tool for the social organisation of communities. This was our main motivation in carrying out the project. We think that telecentres can intensify the interaction of communities in the global computer network through a critical interpretation of the information received".

Collaboration

The Maré programme was made possible through agreements signed with other Federal Government bodies – the Bank of Brazil provided the used computers and the Ministry of Communications installed the GESAC antennas – and the organised community which, through its organisations, settlements, associations, rural communities and social movements, provided the furniture and infrastructure for the centre where the telecentre was installed.

By the end of 2004, the country will have 27 telecentres and the plan is to double this figure in 2005 and in 2006. The computers will only use free software, based on guidelines issued by the Federal Government. "We believe

that all of the features of free software are in line with our observations on the organisation and independence of the communities and the autonomy that we wish to enjoy", explains Maria Luiza Ramos, technical advisor to the Project Management.

The arrival of the telecentres to the communities has interrupted traditional practices and free software is among these newcomers. "We visited some IT academies with proprietary software and saw that they put limits on the users, restricting the learning of how to handle the technological tool. In the telecentres running free software in fishing communities, users have greater freedom to investigate. They have just discovered computers and the Internet and they are now discovering themselves, seeing that they are capable of so much more – and this is very important", concludes Gilson.

The Special Department for Fishing trains two monitors and one technician for each telecentre, who are chosen by the community for the local knowledge they can add to the project. Age and sex are also taken into account in the selection criteria and there must be at least one female and one more mature candidate. With young people, the programme gives priority to the selection of those who are familiar with the community context so that they can recover its history, gain self-confidence through the profession and tell their story and describe their experience to other communities.

One of the main aims of this digital inclusion programme is to stimulate and develop the organisation and structuring of the communities. In some cases, the theoretical contents are applied in collaboration with ANCA, the National Association for Farming Cooperation, which is linked to the MST (No Land Movement). "Some individuals are being trained in collaboration with MST because this increases the potential of the community organisation contents", adds Maria Luiza.

The selection criteria used for the towns where the telecentres will be set up take into account a number of issues, such as the insecurity of the premises, the physical isolation caused by the lack of information and communication and, most importantly, the level of organisation for the survival needs of these communities. Some telecentres were set up in bigger cities on the coast to help people defend themselves from real-estate speculation, predatory tourism and the lack of attention from some governments to fishing communities.

A financial issue too

This programme is already seeing that the inclusion of communities in the paradigm of the digital revolution can also have financial benefits.

"Traditional fishing has a significant socio-economic relevance, particularly if we take into account the activity's social role as provider of food and quality animal proteins for the population, particularly in the north-eastern states

of Brazil" (Galdino, 1997). Fonteles Filho (1997) explains that the traditional fishing system survives in Brazil because of the diversity of tropical species, as it is difficult to create industrial companies that can concentrate production and specialise the catching technology, consolidated through socio-economic conditions that are more favourable to the creation of job opportunities and the generation of income.

The biggest problems affecting traditional fishing include precarious living conditions, poor organisation within the sector, the traditional lack of support and incentives, the high rate of illiteracy, inefficient storage and preservation infrastructures, occupation of the coasts by real-estate speculation, the competition of industrial fishing, pollution and environmental degradation.

The techniques and tools used in some cases are also primitive and inefficient. In addition to all this, fishermen find it hard to obtain loans. The long-term result of all this has been to generate a process of social exclusion that has persisted over time through social reproduction in these communities, severely undermining the self-confidence of traditional fishermen and their culture.

To confront the challenge of digital inclusion, this project hopes to trigger an educational process based essentially on fostering participative citizenship, which will no doubt generate specific results in the social organisation of production and in restoring the self-confidence of traditional fishermen and women, creating social inclusion in the political and economic spheres too.

To reach this aim, technical advisory instruments are being created to educate local workers with literacy processes and access and information to loans. According to the planning, a specialised professional will visit these telecentres once or twice a week to provide technical advice and help draw up projects for production and sale.

The organisation of the producers through the network of connected fishing communities, the exchange of experiences and the possibility of contact and selling their products all over the world, have all been successful.

According to Maria Luiza, "in the most recent telecentre to be set up in Cabo Frío (RJ), the women are producing flower craft with fish scales. They found out about the experiences of Belén and Río Grande do Sul with fish scales through the network. The women began to exchange information, perfect their techniques and spread the word about their work over the Internet, which sparked the idea of selling it – even to countries abroad. It is a huge umbrella that covers not only computing but also the organisation and recovery of culture".

"The introduction of telecentres in traditional fishing communities is bringing digital inclusion to this population segment, enabling access to new technologies, wider relations, Internet access, the democratisation of communication, the use of new technologies for education, increased access to knowledge and incentives for investigation, the speed and skills to deal with demand, and the possibility of exchanging experiences and results with other communities connected to the network. All this generates a context that strengthens the organisation of the sector. In short, a new world is being discovered.

Specific aims of the project

1) To qualify the profession of traditional fishermen and women.

2) To set up a national communication network using computers.

3) To give the population access to basic digital inclusion.

4) To improve the technical profile of communities and increase their job prospects.

5) To educate young people from fishing communities in technical IT support.

6) To train educators/multipliers in digital inclusion.

7) To provide a catalyst for the social organisation of the traditional fishing sector and its social integration.

8) To promote the development of free software for the fishing activities of associations, cooperatives and communities.

9) To encourage the habit of recording fishing data and storing it in free databases.

Points of culture. The digital inclusion programme of the Ministry of Culture

"What we are seeing in digital computing in the world today started out in the libertarian counterculture movement. So what could be more natural, from our politico-cultural perspective, than a movement promoting free software for the pragmatic implementation of another of our realistic Utopia projects.

It is a strategic stance. Free software will be basic and essential if we are to have freedom and autonomy in the digital world of the twenty-first century. It is a *sine qua non* condition of any truly democratic digital inclusion project.

We cannot settle for paying *income* for all eternity to the owners of closed languages and models. Free software is at odds with all this. It will permit the mass inclusion of the people. It will enable the development of small companies in Brazil, of our future *soft houses* and it could create jobs for thousands upon thousands of technicians.

This is why the Ministry of Culture of Lula's Government believes that Brazil must be prepared to become a world free software leader. This is the path to the absolute mastery of digital culture. This is the road to inclusion of each and every Brazilian in the contemporary cultural universe."

Gilberto Gil, 19 August 2003

The words of Minister Gil leave no doubt as to the position and plans of the Ministry of Culture for Brazil's digital inclusion. The first steps were taken with the launch of a public bid for the creation of "Points of Culture". A total of € 4,510,000 will be invested in the project, which plans to create a thousand points of cultural irradiation in *favelas* and among indigenous tribes. One hundred "points" will be installed by the end of 2004, a further 500 in 2005, and another 1,000 by the end of 2006.

The "Points of Culture" consist of the distribution of complete kits of computers, microphones, scanners, video cameras, digital cameras and funds for hiring monitors and training multipliers, all through a series of public bids. The aim is for each point to act as a laboratory of digital culture, making full use of the benefits brought about by the digital revolution. The aim of digital literacy, in this case, is to train cultural producers and artists, who will then be able to experience the new possibilities of the converging digital language for creation, production, advertising and distribution of their work.

The challenge of breaking down the barriers posed by the traditional forms of cultural diffusion and appropriating this new paradigm for the communities involved is the crux of the project. The use of free software is the natural option, but the need to perfect free tools for image and audio production does not worry Claudio Prado, digital policies organiser of the Ministry of Culture and architect of the project. "We believe that there is a possibility of literacy in languages based on free software because we will have full knowledge of what we are doing, why we are doing it and how we are doing it", he explains.

This challenge is supported by Minister Gil, who underlines the Brazilian Government's commitment to fostering what he has termed "agrarian reform" in cultural property, comparing free software to "a clear example of confiscation of intellectual estates, essential for opening up the field to future new creations."

The "Points of Culture" project will be a permanent bridge between technology producers, *hackers* and digital art, and forms the substantive part of the needs of digital inclusion and for sharing knowledge of art. It is the road to mastery of digital culture.

But Prado goes further. For him, "the Ministry of Culture considers that revolutions in digital technologies are essentially cultural and have translated into a paradigm shift. Technological convergence is generating an extraordinary possibility of rethinking all issues. Digital culture is an iceberg with three visible tips. The biggest visible tip of digital culture is the Internet, the second is free software and the third is the digital distribution of cultural and intellectual goods. These are the main areas of action of the Ministry of Culture.

Our digital culture programme is a digital inclusion programme because we understand digital inclusion to be social and cultural inclusion through digital technologies. This is the idea that technology can be transformed for social policies. The innovation we are bringing about for digital inclusion is based on the premise that the foundation stone of digital inclusion is an interactive multimedia studio, which requires broadband and, at the far end of this broadband, interaction in all transportable languages: text, 2D image, audio-visual and software. For us, the focus of digital inclusion is a space where it is possible to become literate in these languages."

Alexandre Freire and Dalton Martins, project consultants and postgraduate students of Computer Science at the University of São Paulo, explain: "we want to work on two focuses of production. The first is multimedia production, which follows the line of audio-visual development and includes a small technical infrastructure for working with video, audio, image, text development, Internet, editing, etc. And the other is technological production, the aim of which is to work with hardware, programming, GNU/Linux customisation, server listing, *firewalls*, writing *scripts* – the "hacklab" part.

We are customising the distribution (the *script* testing and customising the software that will be used and creating "survival kits" to keep these laboratories running. The infrastructure will be connected to a national network and provide a collaboration system with an effective distribution, where each point will publish everything that it is producing and all of the points will communicate with one another".

Instead of outsourcing maintenance, the project will train people from the communities involved to receive *online support*). The idea is to provide complete autonomy to the points of culture and encourage the multiplication of these points across the region. "We need to create solid cores at these initial

points that can be transformed into replicators for the other points that will be set up next year. Thus, the project will multiply the number of people required to host the workshops", says Prado.

The first "point of culture" to open is based on an agreement with the local authority of São Paulo. The space will consist of a telecentre and a large adjoining area for the recycling of hardware and digital technology. "It is a very interesting project and we will be spreading it to other areas. Recycling is a big step for us in the task of breaking down two digital exclusion barriers: one is that of people who think that only those with studies can be included, because everything is very difficult and complicated, when we see boys learning to build a computer in two days in the recycling office. This simple fact of assembling and disassembling a computer, switching it on and off, and doing something with this material gives them a power that they did not know they had and makes them feel included.

They can then start asking questions and discovering means and possibilities. And those with "technical DNA" in their blood begin their technological literacy process when they discover that there is something that can be understood in two days inside that machine, even if they have never stepped foot inside a school. The second exclusion barrier is cost, because people often think that technology is only for those who can afford it. And what actually happens with technology is that prices are falling dramatically. Technology access is then increasingly democratised and open; many more people can have a recording studio, for example. Thirty or forty years ago, a studio cost € 1,500,000 while nowadays, you can set up your own studio with a computer and another € 150", explains Claudio Prado.

"We believe that this other vision, that in which we begin to transform computers that were tossed on the rubbish heap and convert them into state-of-the-art technology, adapting them, making *clusters*, adding computers to make small-scale servers, etc., when we see that it is possible to transform rubbish into cutting-edge technology, that we are doing the things we want to do. Free software is a great help with all this because this alchemic possibility of transforming garbage into things that work – not into an old computer for poor people but into working technology – is another element that cancels out the exclusion factor", explains Prado.

Intellectual property and copyright

A critical view of intellectual property and the recovery of the true meaning of copyright is a hot topic that has come to the fore over the course of this project. We are seeing a situation in which the authors of intellectual works, be they musicians, film-makers or software developers, currently transfer their works and rights to intermediaries. These intermediaries, protected by intellectual property laws, take over the works of their authors. This intermediation that simply "copies and distributes" the works of the authors

is being questioned in the digital revolution, since the Internet does this alone; in other words, digital technology and the global network of computers can restore a direct relationship between the producers and their public without the need for intermediaries. The Ministry of Culture has supported and discussed alternative licensing options for cultural works in the digital context, such as the Creative Commons.

"The idea of intellectual property is one of the paradigms of the three tips of the iceberg I mentioned earlier. I will use the example of music to explain how all this works and how it is integrated into the Points of Culture. For example: When a young boy comes into a Point of Culture studio to record a CD or a song, he is sent to a workshop where he is informed about what a licence is and how it works in this new Creative Commons model. This means that when the boy records his song, he will have already been introduced into the logic of a system that is exactly the same logic of the musician. The musician and the boy can understand the logic of the system, of having the song recorded and having it distributed on the Internet so that it can be made available and possibly marketed in the future. This is how we see all this relating to the situations that arise at the Points of Culture.

Nowadays, there is an exclusion pyramid based on the closed concept of copyright. All reserved rights exclude anybody who does not have a commercial perspective for their music. This is true of 100% of the boys who record their music for the first time; nobody is looking to market it, they just want it to circulate. The logic of circulation is flawed because this circulatory movement is excluded from the system while, in the Creative Commons model, it can be included in the system. He authorises his music to be made available under the conditions he chooses, and this is included in a logic of music management in the twenty-first century. We believe that doing this in the multimedia studios we set up in the Points of Culture will give the musicians that come out of these studios a direction. At the moment, hopefuls who want to enter the world of music have to leave wherever they are for Río/São Paulo and prostitute themselves artistically – and possibly physically – to get a break on Fantástico*. It is a flawed pyramid", adds Prado.

Note

Fantástico is a very popular open television programme that showcases new talent, among other features.

The Ministry's policy is to bring attention to the fact that the digital revolutions have introduced an aspect of non-viability into the traditional system of distribution. "The Creative Commons model flexibilises the rights of authors and establishes a possibility of circulation forbidden by the current copyright model. The Ministry of Culture considers that the progress made in digital distribution is inexorable and there is no point in fighting it. Indeed, digital distribution has been accused of being responsible for piracy. We prefer to look at the possibilities of digital distribution as offering great potential, an excellent possibility of democratising knowledge and information. This new form of distribution must be regulated in ways other than those used in the nineteenth and twentieth centuries, using a model devised and generated

on the basis of rights of the authors, on reserved rights. We understand this flexibilisation of the rights of authorship to be essential to an understanding of the new management model we need for cultural matters.

In music, the record labels will logically fight digital distribution because it excludes them from the wonderful bounties of which they were usufructuaries in the twentieth century. In other words, those who fight against making the system more flexible will be fighting to maintain their profits. We, the Government, are obliged to see this as an excellent opportunity of giving access to people who were always excluded. And this cannot be achieved without a change in attitude towards the rights of authors", says Prado.

Combating functional digital illiteracy

We know that "functional digital illiteracy" exists today. It refers to people who can use e-mail and browse the Internet but do not know what to use it for because it has no place in their everyday lives and context. Digital technology "did not penetrate fully enough to be transformational for that person, as the implementer of what they produce. He or she stopped at the use of programs and tools and did not delve further into the appropriation of the technology or the possibilities that it offered. This represents an enormous gap and functional illiteracy exists both among excluded people and among those considered to be included from a technical point of view" says Prado.

"Digital culture presupposes an understanding that commercial sites are a sort of screen that hides the reality of the Internet, its wealth and the possibility of making technology one's own. On a commercial site, we are in the same situation as when we go shopping. We go in and find a myriad of things to buy, but most people do not have access to these things, so they are excluded." Professor Nelson Pretto of the Federal University of Bahía, goes a step further and compares the sites to "pens" that restrict the possibilities of Internet users, herding them along a single path.

This project acknowledges the need for citizens to appropriate this digital revolution, not as consumers of technology or content, but as producers, as active and restless subjects of the digital revolution.

2.4. Porto Alegre, a free thought factory

The city of Porto Alegre in the far south of Brazil has become a world reference for its practices in the construction of a new citizenship based on participative democracy. Every year, tens of thousands of people meet at popular assemblies to decide how to allocate the municipal budget funds and set down the priorities for its public policies. This experience of direct democracy known as *orçamento participativo* (participative budget), in use since 1989, is responsible for an increase in popular participation in all areas and has acted as a launch

pad for an exchange of experiences never before seen in an international community seeking a new point of reference for social development that was different to the neoliberal thinking dominating the planet at that time.

The relationship between the state authorities and the city's inhabitants, and between the latter and avant-garde sectors of new international thinking, culminated in the 1st World Social Forum of Porto Alegre in January 2001, in contraposition to the World Economic Forum of Davos.

"The World Social Forum is an open meeting place for reflective thinking, democratic debate of ideas, formulation of proposals, free exchange of experiences and interlinking for effective action, by groups and movements of civil society that are opposed to neoliberalism and to domination of the world by capital and any form of imperialism, and are committed to building a planetary society directed towards fruitful relationships among mankind. The WSF proposes the debate of alternatives for the construction of a globalisation in solidarity that will respect universal human rights, and those of all citizens of all nations and the environment and will rest on democratic international systems and institutions at the service of social justice, equality and the sovereignty of peoples."

Website

Charter of Principles of the World Social Forum:

http://www.forumsocialmundial.org.br/main.php?id_menu=4&cd_language=2

Against this backdrop, the municipal authorities of the city of Porto Alegre began to draw up policies for a new information society. Full access to digital knowledge as a "human right" and a vision integrated with state policies of social inclusion were the guiding principles of the municipal actions in information technology.

2.4.1. Municipal digital inclusion policies

Since 1977, the municipal authorities of Porto Alegre have had their own public ICT company, PROCEMPA, which develops the municipal government's systems for health, taxes and finance, education, transport and traffic, sewage system, cleaning, planning, building and administrative management.

Almost coinciding with the launch of commercial Internet in Brazil, in 1995, the municipal authorities of Porto Alegre, through PROCEMPA, took the pioneering step of creating the Internet service provider (ISP) PortoWeb. This measure sought to encourage the population's participation in the global network of computers and help to modernise and de-bureaucratise the civil

service by making it accessible from the Internet. This, the first government action for integration into the information society, was a great success and registered thousands of Internet users in the first few months of its operation.

Another important milestone was reached in 1996, with the creation of the "Infovía", a fibre optic network connecting the main public buildings, which have over six thousand computers installed and connected to the Internet between them, and the poles of economic development with modern multi-service broadband technologies. This created a backbone of over 210 km of fibre optic cable illuminated by ATM/IP technologies and alternative broadband access using radio-linking. During this period, the authorities began introducing computers in all municipal schools.

Later, in 1999, the municipal authorities took their first steps in the use of free software and took part in the creation of the *Software Livre RS* (RS Free Software) project and the "International Free Software Forum". In 2003, the authorities officially assumed the use of free software as a government policy and began to break free of proprietary environments.

2.4.2. Free software in schools

The municipal schools of Porto Alegre are mainly located in areas with a high rate of social exclusion. The introduction of computers to the *Rede Municipal de Ensino* (Municipal Education Network) was defined as a politico-educational principle of the Municipal Department for Education of Porto Alegre and has resulted in progress from educators and in the comprehension of students as regards access to new technologies and the furthering of knowledge construction possibilities. Access to the global computer network paves the way for the educational task of using its countless resources for communication, research and interaction, all available in the digital context.

The "IT in education: a network of digital inclusion" project, incorporated into Porto Alegre's Municipal Education Network, uses free software. The preference for free software was down to the need to make available more suitable tools that were coherent with the municipality's particular educational policy: *Escuela Ciudadana* (Citizen School). The latter organises the World Forum on Education, also in Porto Alegre.

According to Professor Sofía Cavedon, a municipal member of parliament who was Municipal Secretary for Education and one of the architects of this policy, "the educational project should be viewed as an educational Utopia. This is because it offers the challenge of being a school for all and of being a generator of citizenship, with emancipating assessments and a relevant syllabus that

welcomes the diverse cultures of the community. And a Utopia because it promotes the critical and creative construction of knowledge and educates students as the subjects of their learning".

The idea of Citizen School has been taking shape in the daily activities of the Municipal Education Network over the last 14 years and is the result of an enormous group effort in conformity with the decisions of the city's inhabitants, aided by the participative budget.

"It is in these spaces where, with the deeper involvement of the community, through new practices, reflection on results, systematisation and theoretical consideration, we search incessantly to build solutions that will guarantee learning with social quality for those who were unable to continue at school", says Sofía.

The classrooms with computers running free software are already benefiting 91.3% of the students of a system that interconnects 51 local school networks in diverse parts of the city, which amounts to 56,533 students and 3,762 teachers. All schools have free Internet access through a variety of technologies, such as private 64 kbps data communication lines, the PROCEMPA infovía, *wireless* and ADSL.

Implemented as part of the educational policy, the free software computer rooms are used to develop the content for various disciplines. The project, which has won national awards, attempts to break with the logic of social exclusion, creating means for the appropriation of IT at schools.

2.4.3. Community telecentres in Porto Alegre

The city of Porto Alegre has the most telecentres per inhabitant in Brazil, which it began to install in 2001. "The idea of community telecentres goes hand in hand with other initiatives adopted by the municipal authorities of Porto Alegre to reduce digital exclusion and promote e-citizenship, installing computer equipment and Internet connections in community spaces on the edge of the city, where access to these resources is socially and economically difficult for most of its inhabitants", explains Joel Raymundo, CEO of PROCEMPA (public IT company of Porto Alegre).

The telecentres of Porto Alegre are not "municipal" spaces; in other words, the centres where they operate are community spaces such as parent associations, churches, community associations, business associations and public buildings. The community also looks after administration of the centre through a Management Council and their priorities are defined at the participative budget assemblies.

The telecentres are the subject of many collaborations, as Joel Raymundo points out: "telecentres can be set up on the joint initiative of the public authorities and a variety of organisations, governmental or non-governmental, in which the Internet is positioned as a tool within the reach of the more disadvantaged members of the population, with specific applications to make their lives easier and place them in the new context of the information society. Social inclusion policies are strategic objectives of public action, given the serious shortcomings affecting large sectors of the population. In this sense, community telecentres form part of a policy to reduce social *apartheid* through digital inclusion".

Ilton Freitas, executive coordinator and one of the architects of the programme explains that "a digitally excluded citizen will be unable to access the flow of information, services and symbolic wealth on the Internet and will see his or her right to information and expression threatened by the new communication system brought in by information technologies.

The inclusion of individuals in the digital world is a new topic on the public agenda. The right to freedom of expression and information must be complemented by the right to digital knowledge. This new social right must be instrumented by a government policy that can universalise it. The State needs to democratise access to computers and the Internet and to foster the training of citizens in the use of these tools".

Porto Alegre now has 30 telecentres up and running, which serve 25,000 users.

Through PROCEMPA, the municipal authorities supply and install the hardware equipment, free software and logical network, carry out maintenance and meet the costs of the data transmission line.

Maintenance of the physical space, the supply of energy and office materials, and municipal security and conservation are all paid for by the community organisation or association. The initiative has collaborators from the private sector and civil society, including SEPRORGS (Trade Union of IT Companies of RS), the Pensamiento Digital Foundation, UFRGS (the Federal University of Río Grande do Sul), the organisation Pessoas.Info, UNESCO, SENAC, the Free Software RS Project and the Federal Government, through the Bank of Brazil, which donated many of the computers used in the telecentres.

2.4.4. Government migration plan

In accordance with a municipal law giving priority to the use of free software in the government and a political ruling by the executive branch of the municipal authorities, in 2003, technicians of PROCEMPA launched a bold

plan to migrate the entire technological structure of the municipality of Porto Alegre – a metropolitan network with over six thousand computers in 370 subnetworks.

A new systems development environment with 44 system analysts, 50 programmers and 25 trainees is now ready to develop all of the information systems of the Porto Alegre municipality on a free platform.

2.4.5. Some technical details

The choices made were: PHP programming language for websites, Apache web server, MySQL database and the GNU/Linux operating system. To develop the initial pages, the municipal company created "Proweb Libre", a tool used to include content and manage the websites of all the organisations and departments of the municipal authorities. "We have a working system that can be connected with a password anywhere with Internet access", explains Volney Alves, Internet and e-Government supervisor at PROCEMPA. By the end of 2004, the sites of all municipal agencies will be remodelled using this tool.

For critical missions systems, the chosen option was the J2EE platform (Java 2 Enterprise), which avoided dependence on proprietary libraries and compilers. The productivity tool that will be used to develop these systems will be Eclipse.org.

These development environments are using version control (CVS – Concurrent Versions System).

In the municipality's network infrastructure, proxies using Microsoft software were replaced by a GNU/Linux proxy with the following services: HTTP/HTTPS/FTP support (Squid); SOCKS support (Dante); FTP uploading/downloading support (jftpgw); Site/URL blocking (SquidGuard); NTLM authentication (Samba); access querying (MALA).

The network services servers that used Microsoft Windows NT are being migrated to GNU/Linux to guarantee a heterogeneous environment in the transition, with support for authentication, printing, DNS, DHCP, NTP, Samba directory and LDAP.

Related links

Municipal authorities of Porto Alegre: www.portoalegre.rs.gov.br

World Social Forum: http://www.forumsocialmundial.org.br

Charter of Principles of the World Social Forum:
http://www.forosocialmundial.org.br/main.asp?id_menu=4&cd_language=4

PROCEMPA: www.procempa.com.br

Municipal Department for Education: www.portoalegre.rs.gov.br/smed/

World Forum on Education: www.portoalegre.rs.gov.br/fme/

Software livre project: www.softwarelivre.org

International Free Software Forum: www.softwarelivre.org/forum2004/

Telecentres of Porto Alegre: www.Telecentros.com.br/

Eclipse Foundation: www.eclipse.org/

2.5. Telecentres in São Paulo

The city of São Paulo was founded in 1554 by Jesuit priests at a time when the Portuguese were occupying and exploiting the land in South America, around the sixteenth century.

It is now the financial, commercial and industrial capital of Brazil and one of the world's biggest metropolises with almost 16 million inhabitants in its metropolitan region. Of all Brazil's cities, São Paulo has the most marked social contrasts and cultural diversity.

High crime rates, a lack of basic infrastructure and major social problems caused by erratic growth are some of the traits of this typically Latin-American capital.

From all this, we can imagine that launching a digital inclusion programme in a city with these characteristics is no mean feat. In 2002, however, the e-Government Coordination Department of the municipal authorities of São Paulo began to respond to this challenge with the creation of its telecentre project.

This project is now one of the biggest digital inclusion projects of Latin America and has served almost 500 thousand people to date. And of course, it has all been done with free software.

2.5.1. Telecentres and the digital inclusion plan

The newest type of social exclusion and the inequalities left behind by the digital revolution can be seen in this great metropolis in the digital exclusion that denies its citizens access to information and technological knowledge, widening the existing gap between rich and poor.

The Digital Inclusion Plan, also known as e-citizenship, is setting up telecentres in the most deprived and peripheral areas of the city. It is based on the idea that it is only possible to combat digital exclusion if governments approach the task from the angle of government policy.

The premises for the telecentres were selected on the results of an analysis of the city's Human Development Index (HDI) and priority was given to regions with lower quality of life and less state presence. There are now 120 units up and running which, in addition to actions for teaching IT skills and using Internet tools, are linked to popular participation and citizenship programmes.

The main aims of the programme are to reduce the rates of social and digital exclusion, retrain professionals, requalify the space around the unit by increasing the flow of people through the streets in the area; spread free software, stimulate popular participation through management councils and generate community journalism.

2.5.2. Structure and operation of the telecentres

Each telecentre has 10 to 20 computers with a broadband Internet connection. Users can use the computers in a variety of ways: free use of the equipment, basic IT courses and special workshops. Free use of the equipment, as one might guess, is where users learn how to use the technology for their specific needs. Individuals can freely browse the Internet, conduct research, read the news, visit chat rooms, play on-line, scan documents and CVs, send e-mails and use the resources of the Internet to the full. The only content that is strictly prohibited is pornography.

Monitors are selected from the community to teach basic IT, basic GNU/Linux, OpenOffice and the GNOME GUI. The programme's biggest success is its special units, whose results suggest social inclusion. The programme teaches *workshops* on Community Communication, Connection of Knowledge, Website Creation, Environmental Education, Professional Training, Digital Art and Access to Public Services over the Internet.

Digital inclusion does not only involve guaranteeing access to computer programs and the Internet or training individuals to enter the labour market. This would be a very limited vision and could lead to failure because it suggests that knowledge of an office automation tool is enough to secure a job. The construction of alternatives to ensure improved quality of life must involve the community and immerse it in the digital world, applying technology across the board to deal with individual needs and the area's organisational requirements. If we can comprehend this, we will have tapped into the secret to the success of any digital inclusion programme.

The special units were set up to structure the community through cultural and educational activities that use technology as a means for the exercise of citizenship and reaching out to individuals. Hence, the collaboration of the participants has been vital to the construction of a collective project in the telecentres of São Paulo.

The telecentre employee is a further ally because he or she acts as an educator, catalyst and community agent all in one. Each unit has a set number of regular meetings but the group activities invite interested participants to work together to develop projects, which might be a presentation, show, website, *fanzine*, etc.

Each telecentre has a Management Council formed and selected by members of the community. This Council assists municipal employees with tax aspects and management of the space. Some telecentres were set up on pre-used or abandoned public premises as part of a process to regenerate the surrounding streets. Many were refurbished and adapted to receive the equipment and community. They open from Sunday to Sunday, except for public holidays.

There is another type of telecentre called the convened or community telecentre, installed in spaces provided by civil agencies or non-government organisations. They are set up in cooperation with the municipal authorities of São Paulo, which also provides the equipment, municipal employees and resources for maintenance. These spaces open from Monday to Saturday, except for public holidays.

2.5.3. Free software used in the telecentres: SACIX

The needs and experience of setting up telecentres led the municipal authorities to develop their own distribution called SACIX. Based on the Debian distribution (one of the most common distributions among government authorities, such as Porto Alegre and Extremadura), SACIX includes a free software package adapted for use in the telecentres that can be freely copied and distributed. There are two versions of SACIX, one for government agencies and civil organisations seeking to develop their own telecentres, and one for domestic users, who can have a diverse package of computer programs on their PC in GNU/Linux.

Website

SACIX:
www.sacix.org/

The free operating system used in the telecentres is GNU/Linux with the GNOME graphical user interface, chosen for its user-friendliness. The main applications available on SACIX are the OpenOffice.org package, which allows users to carry out basic office tasks, and G-Paint, which is a common design application.

2.5.4. Required investment

In order to offer users a suitable environment, the e-Government coordination department adopted a strategy of obtaining quality equipment at low prices.

Data obtained from the programme coordination department reveal that the cost of setting up a new telecentre, including refurbishment of the premises provided by the municipal authorities, is around € 34,570, or € 73,000 if the telecentre is to be built in a brand new space.

If the programs used on the equipment at the centres had not been free software, this cost would have increased by at least 50%. Besides economising on licence fees, the installation of free software allowed the use of less powerful computers with less sophisticated hardware and, hence, cheaper equipment, with a similar performance to superior and more expensive hardware and equipment running Microsoft Windows.

In most cases, to allow network administration and optimise processing, LTSP (*Linux terminal server project*) is used with a server with greater processing capacity to host the applications, which are then run by more lightweight clients.

2.6. Municipal authorities of Río das Ostras

2.6.1. Sand, oil and free software

The city of Río das Ostras is located between Macaé and Cabo Frio on the Costa do Sol, part of the beautiful coastline of Rio de Janeiro. It is a young municipality undergoing a major economic boom. Besides tourism, which triples its population of 45 thousand inhabitants in the summer months, it is also rich in oil and the *income* generated by this activity is sufficient to stimulate the local economy and fill the coffers of the municipal authorities.

Website

Municipal authorities of Río das Ostras: www.pmro.rj.gov.br

But besides its oil and the natural beauty of its beaches, this coastal city has become a reference across the country for its free software implementation initiatives.

2.6.2. Public & free

The municipal authority's free software project, entitled "Public & Free", was launched in 2001 on the initiative of the recently created IT Advisory Office, reporting directly to the Mayor's Office.

The initial concern was to reduce the costs of software licences for operating systems and office productivity suites for the whole of the local government. Replacing Microsoft Windows with GNU/Linux and Microsoft Office with a "Free Office" package was a great initial idea. But it did not stop there.

This initiative also resulted in benefits from the improved allocation of funds. What had previously been spent on software licences was used to purchase more computers, develop new solutions such as print centres, set

up an Internet services provider (ISP) for the municipal authorities and invest in employee health at the municipal government with the purchase of equipment to prevent RSI (repetitive strain injury) and WROD (work-related operational disorders).

It was later seen that the programme had many technical advantages too, particularly in the area of maintenance. "We now encourage the use of free software on all government computers, even by trying to incentivise civil servants who collaborate with us on this – they are first to benefit from equipment improvements, such as new computers and liquid crystal display monitors. Our aim is to have 100% of our computers using free software", adds Marcos Vinicius Pecly Marini, the man behind the project and head of the municipal authority's IT Advisory Office.

2.6.3. Savings and technological freedom

The savings obtained by the use of free software were approximately € 370,000, since the municipal authorities did not purchase any more licences for operating systems (Microsoft Windows), office automation packages (Microsoft Office), databanks, development tools or graphics programs. "Besides an improved use of public funds and technical advantages, it was vital to adopt a stance whereby we could not fall victim to "technological slavery" using an expensive product that did not fully meet our needs and over which we had no control", explains Vinicius.

Free software has now been installed on more than 130 computers in the municipal authority of Río das Ostras, the municipal departments of Social Welfare, Administration, Planning, Sport and Leisure, Tourism, Social Communication, Attorney General's Office and Internal Control, as well as the Mayor's Office. It is also being used in special government projects, such as the "Un Bien Mayor" and "Curumim" youth centres, and in the activities of the "IT for all" programme.

Since the project launch, these activities have recorded a continuous rate of growth and the municipal authorities of Río das Ostras are discovering new possibilities every day for the use of free software. This year, the main focus is "social and digital inclusion" programmes. Free software has grown from the internal context of the municipal authorities and now has a constant presence in diverse activities connecting it to the population, such as the creation of telecentres and basic IT courses, by agreement with neighbourhood associations. All entirely in free software.

2.6.4. Tatuí, the Río das Ostras distribution

The municipal authorities of Río das Ostras created their own GNU/Linux distribution called Tatuí (working environment technology for IT users). This distribution is based on the work of the young Carlos Eduardo Morimoto, creator and developer of the most popular distribution in Brazil today, "Kurumim", which is even used by the Federal Government. Morimoto also runs the "Hardware Guide" site.

Website

Hardware Guide:
www.guiadohardware.net

Kurumim is a live CD (it runs directly from the CD without installing the programs on the computer hard drive) based on Knoppix/Debian and has a relatively easy installation process combined with an excellent recognition of hardware components. Another advantage is that its basic version only takes up 200 MB. It is now the easiest way to install the Debian distribution on workstations. Afterwards, we simply need to upgrade and install the new programs required by each user with the Debian "apt-get" commands.

The developers of Tatuí are in the process of completing a version of the program for IT laboratories in municipal schools. It must be distributed freely to students, which will allow them to use the programs and environments of their schools in their own homes. And because it is a live CD, the children will be able to use their parents' computers without this affecting the other software and files on them, since any changes made are saved to Tatuí. At the same time, the parents and guardians of the students will be able to discover free solutions in a less direct way.

The beta version of Tatuí for Education was presented to the teachers and coordinators of the municipal education department and a study group was set up to select other tools to incorporate into the system. There will be different versions for first and second (before and after what was previously the fourth year of the first level) segments.

The municipal authority of Río das Ostras has also developed and is now using SALI (Free Administrative System) to integrate the diverse sectors and procedures of the municipal authority, along with other systems in PHP and PostGreSQL. For Internet sites, the authorities decided on PHP-Nuke, which offers greater interaction and speed of content management.

All of these initiatives show that, besides oil and beautiful beaches, Río das Ostras has a great deal more to contribute to Brazil's information society programme.

2.7. The Brazilian Parliament and the information society

2.7.1. Parliamentary association for free software and digital inclusion

Brazil is now one of the few countries in the world to have a parliamentary association for free software and digital inclusion in the national congress (Senate and House of Representatives) and is one of the biggest parliamentary associations in the congress with 135 representatives and 26 senators. The association is chaired by Senator Serys Slhsarenko and its honorary chairman is no less a man than the former President of Brazil and current President of the National Congress, Senator José Sarney.

2.7.2. A story that began in the Basque Country

It is an interesting historical fact that the management of this association began in Bilbao, Spain, in February 2003, during the IT4ALL [1] event "Opportunities and Challenges for Regions in the New Information Society". It was a preliminary meeting prior to the first phase of the Information Society Summit and its Brazilian speakers were the President of the Congress, Senator José Sarney, and myself. On the last day of the event, following the brilliant addresses of Diego Saravia and Manuel Castells, I had the pleasure of talking in my speech about the main ideas and philosophies of free software and the status of our movement in Brazil, in the presence of Senator Sarney, who was waiting his turn to give the closing speech for the event. In the evening, at the hotel, we had the chance to exchange impressions on the idea of the information society and free software. This meeting sparked a keen interest in the Senator that would have important consequences for the future of free software in Brazil".

Website

"Opportunities and Challenges for Regions in the New Information Society" and preliminary meeting prior to the first phase of the Information Society Summit, organised by the Basque Government and the Diputación Foral de Bizkaia - Spain, with the backing of the EU's Directorate-General for the Information Society and by the UN/UIT through the Executive Secretariat of the "World Summit on the Information Society"

http://www.it4all-bilbao.org

2.7.3. A week of free software in the Brazilian Parliament

On 2 April, we met Senator Sarney in Brasilia to discuss the initiatives of the National Congress to promote free software. His interest in the issue led to the organisation of the "1st Week of Free Software of the Legislative Body" in August of that year"*. Under the slogan "Free Software and the Development

Note

http://www2.camara.gov.br y
http://www.senado.gov.br

of Brazil", the relationship between free software and digital inclusion and the country's development were discussed in the Brazilian Parliament throughout the week.

Richard Stallman and Miguel de Icaza were speakers at the event and took part in its inaugural act. The Brazil free software community also participated in the event on various panels.

The speakers at the prestigious inaugural act, besides Richard Stallman, included the presidents of Brazil's two legislative chambers, Senator José Sarney, the MP João Paulo Cunha and three of Lula's cabinet ministers: the Chief of Staff, José Dirceu, the Minister for Culture, Gilberto Gil, and the Minister for Science and Technology, Roberto Amaral. The addresses of these government officials were the important political statements needed to consolidate free software as a relevant political issue for Brazil.

Website

Opening address of the President of the National Congress, Senator Sarney
http://portal.softwarelivre.org/news/1191
Minister Gil's speech
http://portal.softwarelivre.org/news/1186

"This event organised by the national congress is a milestone showing that free software is more than a mere possibility for our growth; indeed, it is here to stay", affirmed the Chief of Staff, José Dirceu.

Website

Address of Minister Dirceu
http://portal.softwarelivre.org/news/1188

"Today, we are closing the Week of Free Software and the 'Free Software and the Development of Brazil' seminar. This week's success is a strong indication of the vitality of IT in Brazil and has brought me great personal satisfaction; thus, I feel truly rewarded for being a part of its organisation.

This event is of strategic importance because it marks the adoption of a position vis-à-vis free software by many areas of the State. Brazil decided that the public sector would consider open programs as an alternative to be fully exploited and stimulated, both in its economic aspects and, more importantly, in its conceptual aspects. It has been adopted in order to discover and carve out paths of cultural independence, of creation, of national identity.

From now on, with the aim of disseminating IT, we will always keep in mind the idea that computer languages must be public, developed by users, and not subject to international monopolies", stated Senator José Sarney in his closing address of the event.

Website

Closing address of the President of the National Congress, Senator Sarney
http://portal.softwarelivre.org/news/1202

"Education is an essential stage of this road. I believe that, as Professor Stallman suggests, our children should learn the basics of programming in open languages so that they can play a part in their evolution and retain their independence", concluded Sarney.

2.7.4. Creation of FRENSOFT

Crowning the efforts of the ministers Walter Pinheiro and Sergio Miranda, pioneering in their historic defence of free software in the Brazilian Parliament, the end result of this event was the formation of FRENSOFT (the Mixed Parliamentary Association for Free Software and Digital Inclusion, whose President is Senator Serys Slhsarenko.

"The result of these debates is the Parliamentary Association for Free Software, which is proving itself to be an influential group that will bring to the agenda of our legislative chamber the concern for supporting open systems in order to instrumentalise our independence in the IT sector", concluded Senator Sarney.

The Parliamentary Association for Free Software and Digital Inclusion is strengthening institutional actions and extending the possibilities of alliances needed to consolidate free software in our country, revealing itself as one of the most important political initiatives for the construction of a Brazilian alternative to the information society.

2.8. Brazil at the Summit on the Information Society

The first phase of the World Summit on the Information Society, an official UN event held from 11 to 13 December 2003 in Geneva, Switzerland, was marked by profound differences in the interests of representatives of the governments of the wealthier countries and the bloc of developing and poor countries spearheaded by Brazil, India, South Africa, Egypt and Argentina.

The government delegations of the United States and the European Union almost always worked as a united front, leading the conservative bloc, and did not beat about the bush when it came to defending the sole interests of the big US monopolies.

2.8.1. Socialisation of knowledge at the Summit on the Information Society

One of the major controversies at the Geneva Summit revolved around the free software alternative and the socialisation of knowledge as instruments of digital inclusion and stimulation for innovation and technological development. Brazil and India led the bloc that saw focusing on the exchange of technological knowledge between peoples as being more suited to the development of a democratic society of information and inclusion, being the only chance for developing countries to make up for their technological backwardness.

The Brazilian thesis was taken up by the bloc led by the United States, which championed the alternative of tightening intellectual property laws on digital works, increasing penalties and criminalising users who seek to

copy and freely share works on the Internet. Most of the governments of the wealthy countries, led by the United States, revealed that they wished to maintain a tight and selfish control over technology, protecting themselves by intensifying the ideology of intellectual property.

Besides being a clearly protectionist policy, this stance proposes an information society "without information" and without shared knowledge. In effect, a disinformation society. Clearly, poor and developing countries would be left to play the role of consumers of technology and "packaged" products manufactured in the northern hemisphere, preventing universities, research centres, private sector companies, governments and the population in general from mastering and obtaining knowledge of the technology being – or which ought to be – disseminated.

2.8.2. International governance of the Internet

Another relevant topic was the debate on the "democratisation of Internet governance". The bloc, also led by Brazil, argued that control of the addresses, names and management of the Internet should be tripartite (governments, civil society and the private sector) and be subject to an international body. ICANN (Internet Corporation for Assigned Names and Numbers) is the body currently in charge of establishing the rules of Internet use around the world and it is unilaterally dependent on the US Government.

The Brazilian Government does not argue that governments or even large companies should govern the Internet. However, it does favour the increased participation of users and civil society in Internet definitions. The Brazilian government also argues that in government forums debating Internet governance, all countries should be represented multilaterally – not only the US, as is currently the case.

2.8.3. Digital solidarity fund

The African countries and a ruling of the World Summit of Cities and Local Authorities, held in Lyon (France) a week before the Geneva Summit, supported the creation of an International Solidarity Fund for Digital Inclusion. The proposal is also backed by Brazil and the bloc of developing countries. The resources for this fund could come from a tax on a small part of the profits obtained from the international transactions of IT companies, for example.

The representatives of the countries led by the United States wished to have no talk of this fund, even if it were a non-governmental voluntary fund. They argued that the "market" should lay down the rules for digital inclusion; in other words, whoever has the resources to pay for and purchase from the

monopolistic giants of the north will be able to take part in the information society. The others will need to wait their turn in the long line of digitally excluded individuals.

2.8.4. In Tunisia, for a more inclusive information society

In Geneva, all these points had a questionable and contradictory outcome, owing to the tough diplomatic negotiations. Brazil was a key player on the world stage and this was important, but the result of the Geneva Summit was light years away from reflecting and revealing new ideas for the information society or any type of innovative thinking. It was a summit dominated by "conservative reaction" to the new possibilities opened up by Internet and the digital revolution. The debate must continue and we must try to spread these ideas among civil society until the second phase, which will take place in Tunisia in 2005. There is a great deal to be done. We need to lay this debate on the table and show society the positions adopted by its governments.

Representatives of civil society at the Geneva Summit approved an alternative statement in line with the position adopted by the Brazilian Government and its international bloc.

I consider it vital to organise the support of international public opinion immediately so that the world's governments begin to comply with the wishes of peoples across the planet in the search for a new, more democratic and more inclusive information society. This will allow the benefits and results of the digital revolution to be considered as human rights and not as a simple tool for the accumulation and concentration of wealth.

The digital revolution is on our side.

3. Sun Microsystems

3.1. Business analysis

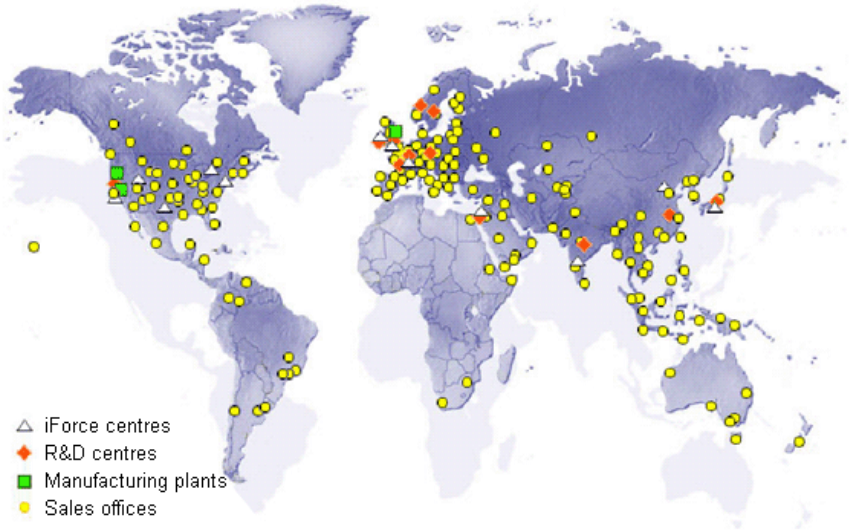
Sun Microsystems Inc. is a US company that was founded in 1982 by three students of Stanford University (Scott McNealy, Andy Bechtolsheim and Vinod Khosla) and one student from the University of Berkeley (Bill Joy).

The main purpose of the company was to design microprocessors and workstations for the University of Stanford, which is why the company was named Sun (after the Stanford University Network). From the outset, the founders were aware of the importance of using open standards to develop their products, which is why they chose to design their microprocessors using the SPARC standard. They also chose an operating system based on an open standard, Unix, to act as the driving force behind the systems they designed. In doing so, they had a great advantage because Bill Joy had developed Berkeley's Unix operating system (BSD) some years earlier.

Since 1982, the company has continued to grow and it markets its systems in virtually every country of the world. Sun's designs have also been adapted to market needs, which has seen the sale of workstations evolve into a wide range of servers and multiprocessor systems.

Sun Microsystems was and still is an innovative company that, as we will show, has always maintained a close relationship with communities that support open standards. Proof of this are the many technological contributions made to these communities by Sun.

Figure 14.

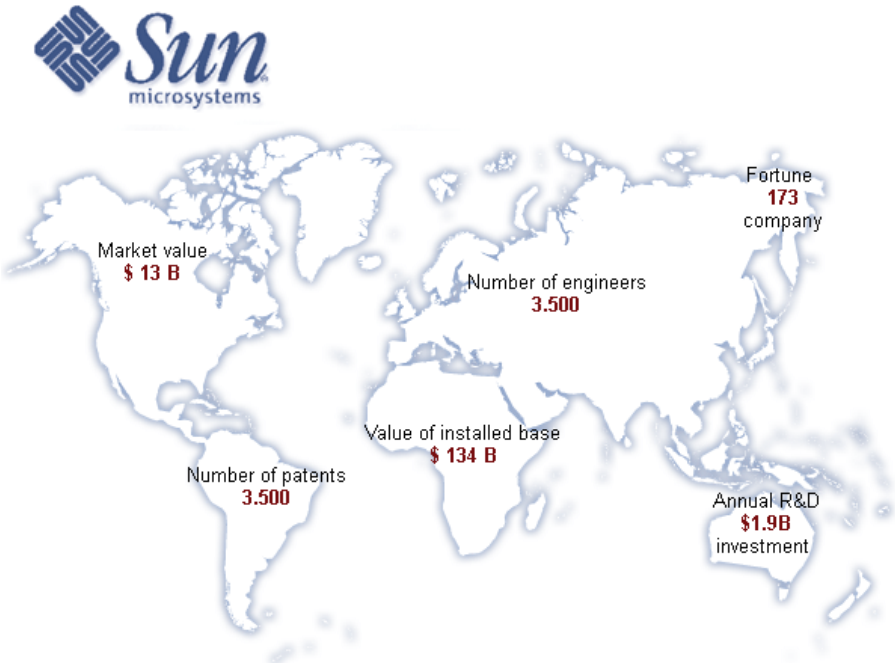


Sun Microsystems has commercial offices in over 170 countries around the world, which means that it needs a wide area network to offer top communications and services to its 35,000 plus employees. This network is referred to as SWAN (Sun Wide Area Network) and consists of over 6,800 subnetworks with 6 data processing centres (DPCs) that house the over 1,700 servers and 500 TB of data. The number of subnetworks, servers and storage systems increases as needs grow. To cite some key figures, Sun servers currently host over 400 network applications and 18,000 websites of over 4 million pages that receive 600,000 visits and move over 6 million e-mail messages every day.

Sun has over 3,500 engineers working on the research and development of innovative new technologies that could help businesses and end users to cut the cost of implementing new network services. The free software community is one of the groups that most benefits from the investments in research and development made year upon year by Sun, since most of these technologies are donated to the community to ensure its continuous evolution.

A prime example of the sizeable R&D investments made each year by Sun are the USD 1.9 million allocated to this heading, which represents approximately 18% of the company's turnover. The more than 3,500 registered patents are the result of these efforts.

Figure 15.



3.2. The private sector and the role of the free software community

The private sector's main concern when it comes to software is how to innovate efficiently. Innovation is straightforward and relatively common and yet, predicting which of the products being developed in R&D laboratories will have an impact on the market and generate profits for the company is no easy task.

Innovation expenditure can be very high for a company if it cannot obtain an efficient return on its products. This is one of the main reasons why the private sector has been paying more attention to the free software community in recent years. The amount of money that a company can afford to pay for top engineers is irrelevant; it needs to realise that no matter how brilliant the company's development team is, there will always be others who can innovate more efficiently. So the old expression "innovation is all around us" rings true here; we just need to find the mechanisms to profit from it.

There are a number of reasons why a private sector company might decide to launch a free software project. Here are a few of the main ones:

- It can obtain quality software at low costs.
- It can obtain a group of users who will test the software in a very wide range of environments.
- Quicker TTM (*time to market*).
- Good positioning of the company, which is seen as an ally rather than the enemy.
- Creation and positioning of standards.
- Lower maintenance costs.

- Fewer risks.
- Greater customer satisfaction.
- Improved integration with third-party products.
- Does away with being tied to companies that create proprietary software products that do not use the available open standards.
- Changes the rules of the games vis-à-vis market prices.
- Encourages free competition.

When a multinational launches a free software project, it contributes experts and a series of procedures and processes that do not generally match the way the developers of this community work. Its main contributions are:

- Project management.
- Management of quality and metrics.
- Development methodology specifications.
- Documentation writers.

The company must still use the project launch and management paradigm that it usually uses because this is the only way to control expenditure and project time, thus avoiding delays and unforeseen expenses that could lead to failure.

On many occasions, the costs of a project based on free software are higher than those that use a proprietary software approach.

One of the key points to bear in mind is the model adopted for licensing the free software applications. One of the myths surrounding free software is that it is not owned by anybody and that when a company develops a free software product, it must forego its rights and control over ownership. There can be nothing further from the truth: the free software model recognises ownership and its concomitant rights.

Companies need to think carefully and consider whether there is any point to launching a project linked to the free software community; in other words, it will need to study whether the code adapts to the needs of the company, whether the licences suit the deployment model that the company wishes to adopt, etc. Following this preliminary study, the company will no doubt need to adapt the source code, choose the licence model, establish a development model, draw up a cost study for launch, etc. We will deal with all of these points later.

Earlier, we listed some of the reasons why a company might opt for a free software project. We will now look at some of these in detail:

- Visibility. Launching a project in free software gives it visibility in the developer community outside the company walls. It is possible to use the standard communication channels of the free software community to

exchange opinions, ask questions and even, in the phases prior to product launch, conduct a series of validation tests.

- Enhanced development of standards. Some projects are designed to develop a standard. When we open this process up to a community of users and developers, we are increasing our possibilities of obtaining outside help and of the effective adoption of this standard.

- Creation of proprietary products. Some free software licences allow proprietary products to be based on free software products through the addition of new features or simply by improving the existing product and offering after-sale support.

- Creation of a market for a proprietary product. When a company has a product built from another free software product, it can gain customers and increase its potential market. This is because the free version of the product encourages users familiar with its use to purchase the proprietary version to obtain support, training and extra consulting services. Moreover, the fact that there is already a free version makes things difficult for the competition, which will need to penetrate the market with another product with better features.

- Better quality. Because the different versions of the applications are made available to a community of developers and/or users before they hit the market, it is easier to detect and fix any minor development bugs in a short space of time. In addition, developers from the community (who are not employed by the company) can add new features that the R&D development team may not have added to the product due to lack of time.

- *Time to market*. By using the code available in the free software community, a company can guarantee a product's quick release on to the market, simply because it does not have to start from scratch. Most of the features it is looking for in the product have already been designed, so it only needs to focus on the additional features without compromising on quality.

- Fewer risks. One of the most common problems faced by companies is the discontinuity of software. Products very often rely on other ones, production of which may be stopped for a number of reasons. With free software applications, development of the product can continue even if the original developers have abandoned it.

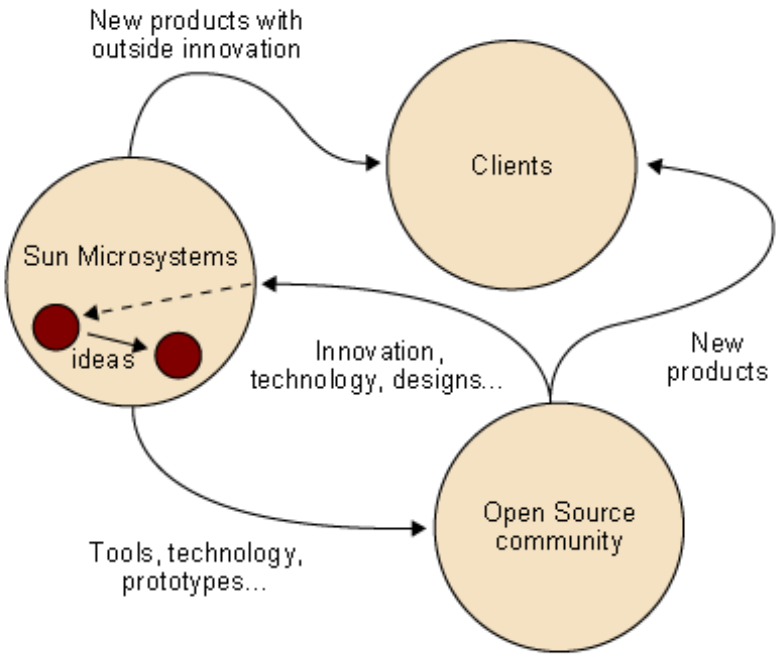
Most customers have misgivings about purchasing products from small companies. The risk of the company going bankrupt and no new versions or support being available for the application they have bought means that small companies very often have problems selling their products, even though they might be excellent.

By adopting a free software strategy, small businesses can convince their customers that, even if the business winds up, continuity of the product is guaranteed because there will always be an up-to-date version available.

3.2.1. Feedback

Private businesses can use the free software model as a basis for sharing tools, technology and prototypes. However, it is more important to create an ecosystem around the free software in which the company and the community join forces and share common aims in order to strike a perfect balance.

Figure 16.

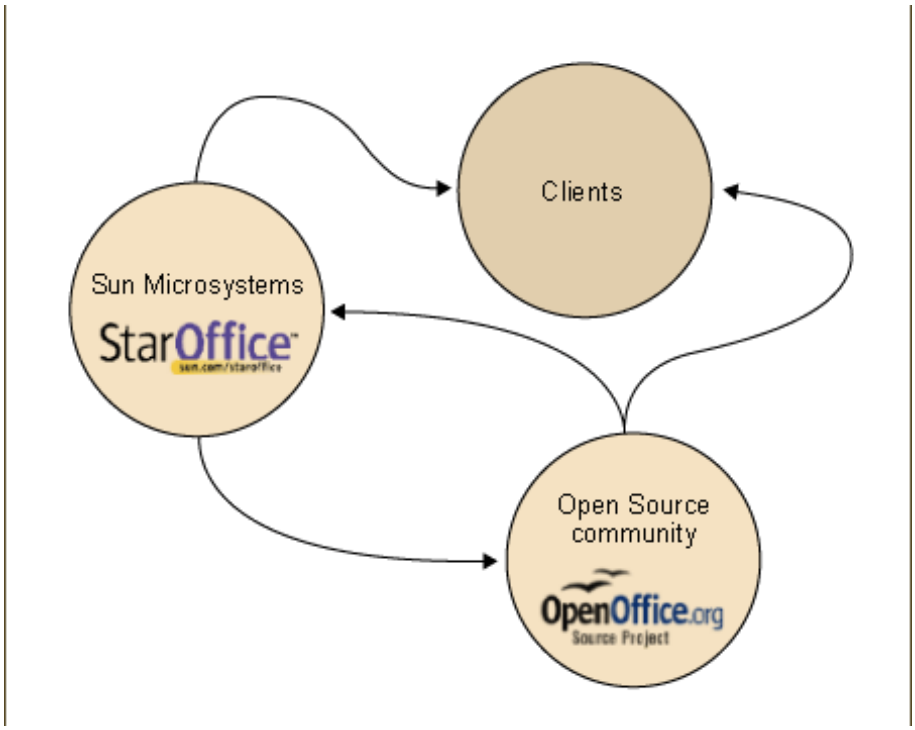


All of the participants in this model reap benefits from it:
The customer obtains a top product with minimal risks and a guarantee of continuity.
The private company helps and supports the developer community in exchange for improved products that it can subsequently market.

A prime example of this type of ecosystem is that created by Sun Microsystems with its StarOffice office suite. In version 6 of StarOffice, Sun Microsystems released the source code, which led to the creation of openoffice.org. Developers from the free software community have now evolved this product, adding new features and allowing the ecosystem to operate just as we have described.

The community developers add new features and Sun subsequently markets the new version of the product after extensive quality testing, with an additional user support service.

Figure 17.



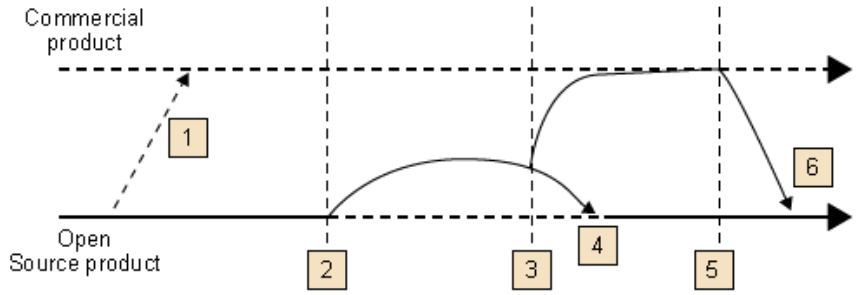
3.2.2. Creation of an appropriate development plan

When a private sector company embarks on a free software project, regardless of the purpose of the end product, it is going to need a coherent development plan that adapts to the company's standards and allows deadlines to be met.

It is important to consider that a project developed entirely by the private sector will be subject to a series of working methodologies and procedures that products developed by the free software community do not have. Therefore, both parties (the company and the free software developer group or community) must be clear on the parameters and paradigm used in the development cycle of the end product, since the two products need to be connected throughout the cycle in order to avoid incompatibilities. There are a number of models for linking the two versions:

- One involves basing the brand version on a stable version of the free software application. In this model, any changes to the commercial product flow in the direction of the code of the free software application. This process ensures that the two codes are compatible and that they have the greatest possible number of similarities in order to minimise version problems. The development plan requires the development cycle of the commercial product to be in step with the cycle of the free software application.

Figure 18.

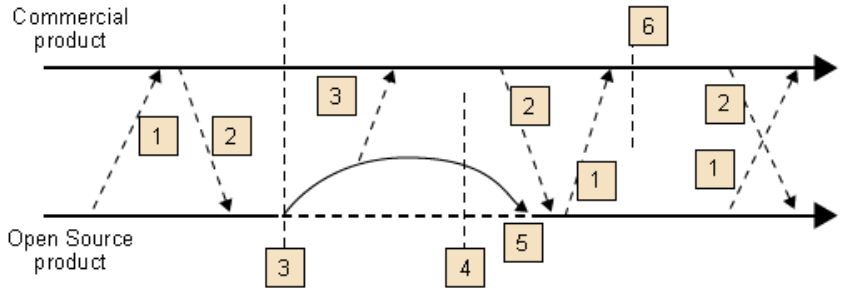


1) Incorporation of changes from the free software version to the commercial product code.
2) Start of the development of the free software code. The solid lines represent developments in the next stable version while broken lines reflect the current development (new or experimental).
3) End of the cycle of the free software project. Stable version available. Start of the development of new versions of the commercial product, based on the stable version of the free software platform.
4) Merger of the versions.
5) End of the development cycle of the commercial product. Stable version available.
6) Merger of the commercial product code with that of the free software application.

As we can see, there will not be many differences between the free software version and the commercial version. The latter may have the odd new feature or functionality but most of the parts it has in common with the free software version will be a subset of the official free software version.

- Another, less desirable, development process involves having two code bases and leaving it up to the developers of the private company to decide when to merge theirs with the free software application. This model has an advantage over the previous one in that it gives greater flexibility to the company developers. However, it also has the disadvantage of complicating source code management due to the existence of two bases evolving independently.

Figure 19.



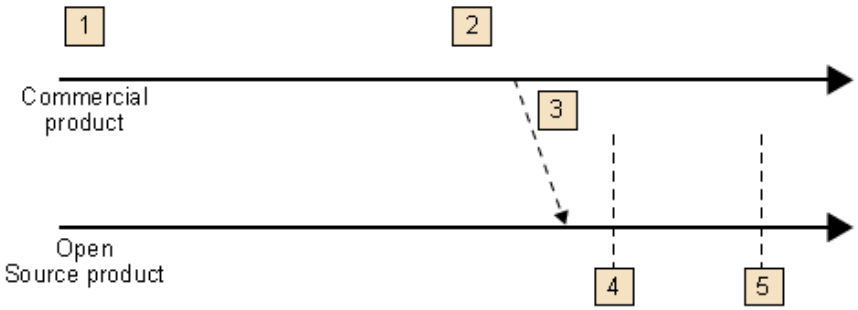
1) Incorporation of changes in the free software version to the commercial version.
2) Incorporation of changes in the commercial version to the code of the free software version.
3) Start of the free software application versions.
4) End of the cycle of the free software application. Stable version available.
5) Merger of the free software version with the commercial version.
6) End of the development cycle of the commercial version. Stable version available.

Development of the two products is parallel and there is no relationship between their delivery cycles. The code from one of the projects is simply injected into the other when the developers of the company consider it convenient to do so.

This model has the advantage that the two versions will have different features, since they are essentially two different products with very similar appearances.

- A third development model is as follows:

Figure 20.



1) Start of the development cycle of the commercial product.
2) End of the development cycle of the commercial product. Stable version available.
3) Incorporation of the changes in the commercial product to the free software application.
4) Start of the free software application development cycle.
5) End of the cycle of the free software application. Stable version available.

In this model, development of the commercial product is internal and not visible to the developer community. When finished, it is shared with the free software developer community. The private sector companies that use this model generally have their own developers working exclusively on the commercial product, leaving the development of the free software to outside volunteers.

This model requires the least contact between the company and the free software developer community. Many companies opt for this model to add their name to the list of companies that promote free software and collaborate with its community. However, these projects usually fade away after the first version because the company no longer collaborates to allow the project to evolve satisfactorily.

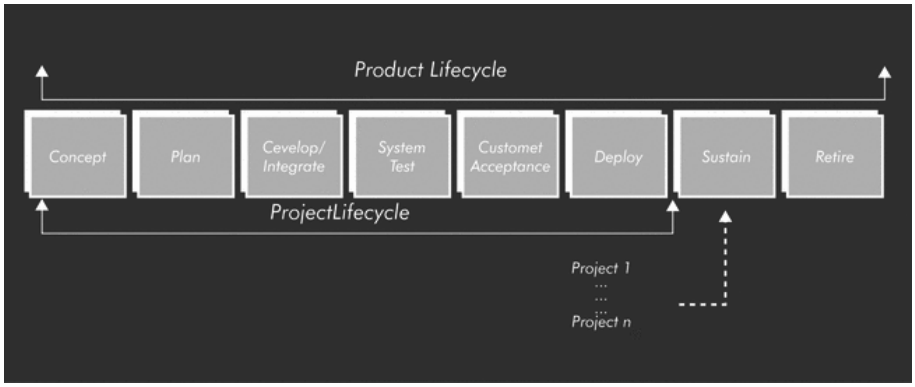
Sun Microsystems began the openoffice.org in this way but adopted a different strategy to prevent the free software project from petering out. Its strategy was to combine this development model with the first one we described: after creating the commercial product, the code is released and handed over to the free software developer community (Model 3), after which Model 1 is adopted, allowing both communities to release new versions.

3.2.3. Software life cycle models: SBS PLC

SBS PLC (Sun Business Systems Product Life Cycle) is the standard life cycle model for information technology infrastructure projects and their developments.

Any project developed by Sun Microsystems is subject to the procedures and methodologies set down in this standard. The model may vary according to the type of project, its priority, critical status, etc. Nonetheless, the life cycle of a project is always divided into the same stages:

Figure 21.

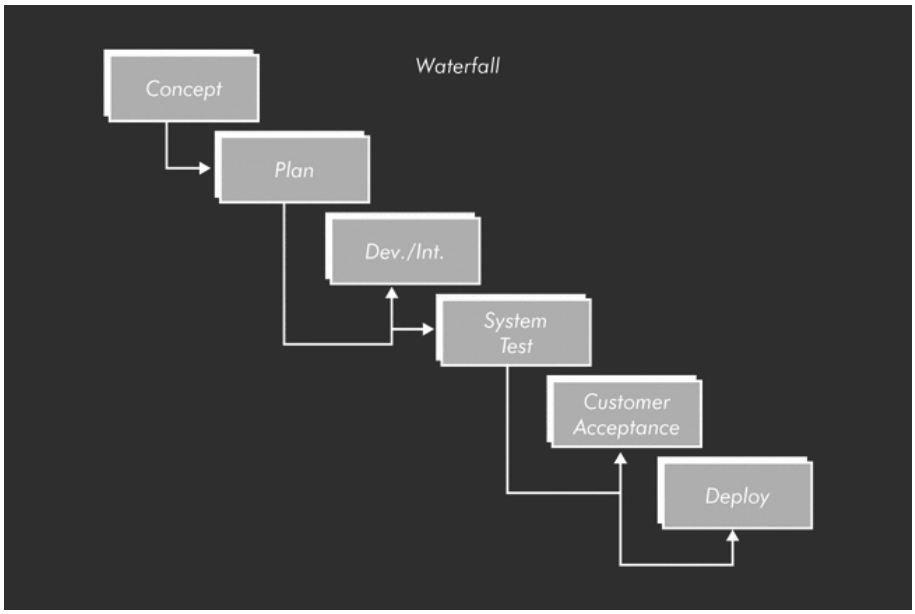


The life cycle of a product extends beyond the life cycle of the project because the product maintenance phase generally falls outside the scope of the project.

Sun Microsystems carries out projects to develop applications for use in its own departments, while others are designed for global use. Hence, it classifies project models according to type, which include:

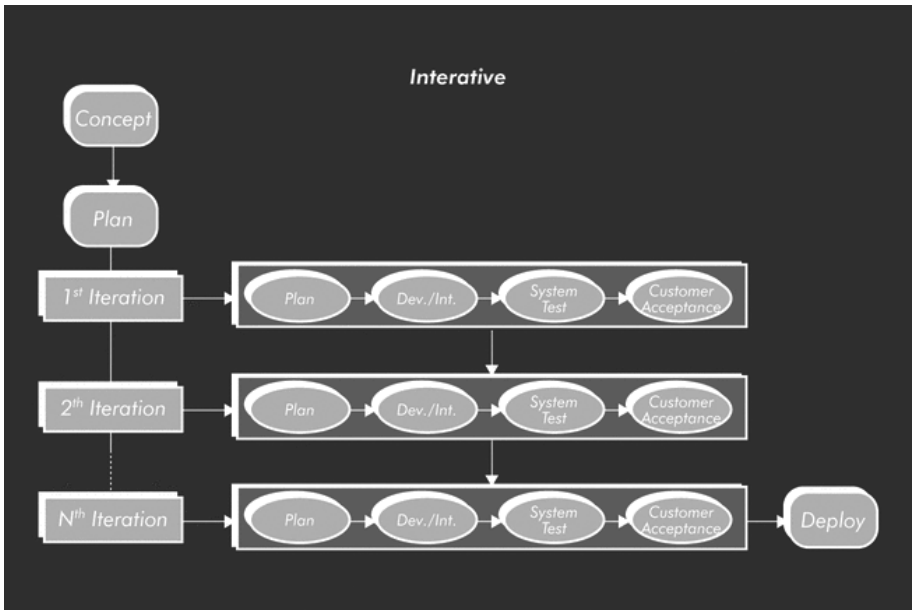
- Waterfall

Figure 22.



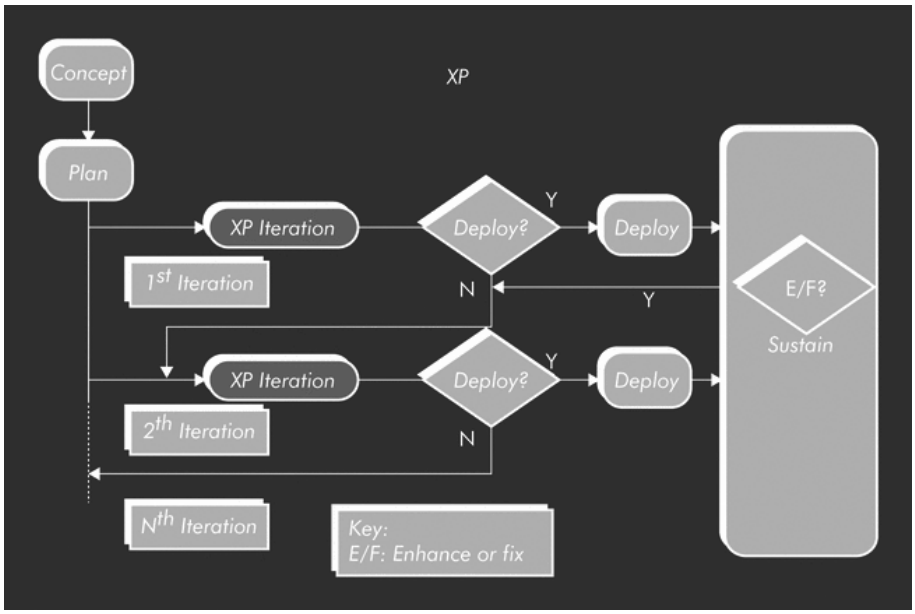
- Iterative

Figure 23.



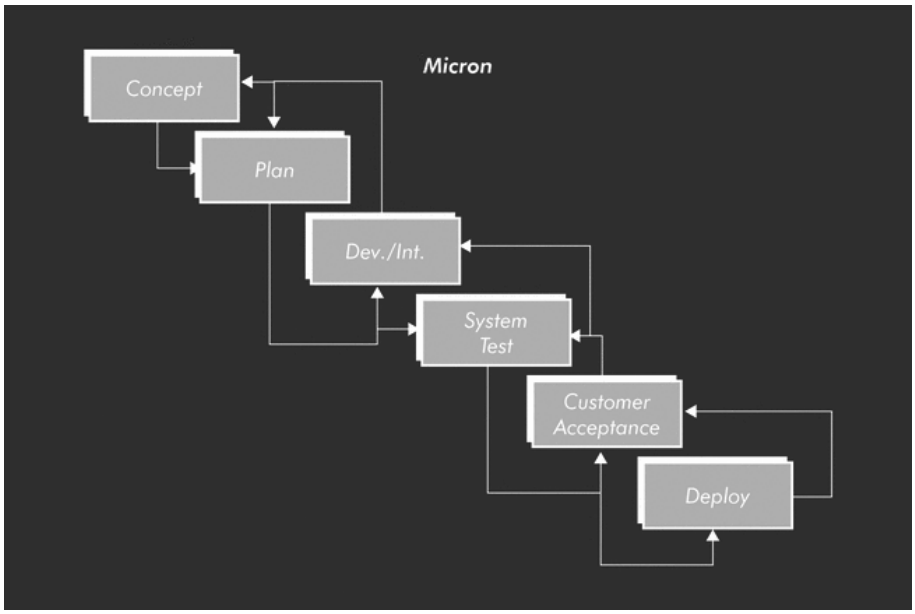
- XP

Figure 24.



- Micron

Figure 25.



Depending on the duration of the project and the time and people required for its development and subsequent deployment, we can draw up the following table to identify the type of model that needs to be used in each case:

Figure 26.

	Development/Deployment (ASP, integration, etc.)			
Major Project	Waterfall	Iterative		
Minor Project			XP	
Micro Project				Micron

Major: over 90 months/employee
Minor: 9 months/employee to 90 months/employee
Micro: Less than 9 months/employee

We will now look at the aims and potential risks of each of the phases of a project in this life cycle model:

- Diagram phase (pre-concept phase). By creating, reviewing and subsequently approving diagrams, the company can guarantee that the project will have a business case coherent with its strategy and priorities. The most obvious business risks in this phase are the possible lack of justification for the project or discrepancies with the company's strategy and priorities.

- Concept phase. This guarantees that the concepts have been reviewed and that there is no better alternative than the one proposed. Approval from the management committee is required to ensure that the project meets the business aims.
  Potential risks may stem from a poor assessment of the alternative products, which could lead to a system being implemented that does not meet the needs of end customers and is hence doomed to failure.

- Planning phase. This ensures that the project meets the architectural requirements and complies with the security conditions, standards and policies of the company. All of these parameters need to be validated before we can move on to the next phase.
  The most obvious risks usually concern a functional, technical and deployment design that is not in tune with company policy and would result in a product veto in the next phase.

- Development and integration phase. This ensures that the product has been correctly developed and integrated and ready to move on to the next phase.
  The only risk associated with this phase is that the product is not ready for the validation phase.

- Validation and product testing phase. This phase ensures that the product has undergone rigorous revision and validation with the sole aim of moving it on to the next phase (customer acceptance). The product validation and testing reports must be submitted to the Management Committee.
  The only risk associated with the product in this phase is that it does not qualify to move on to the next phase.

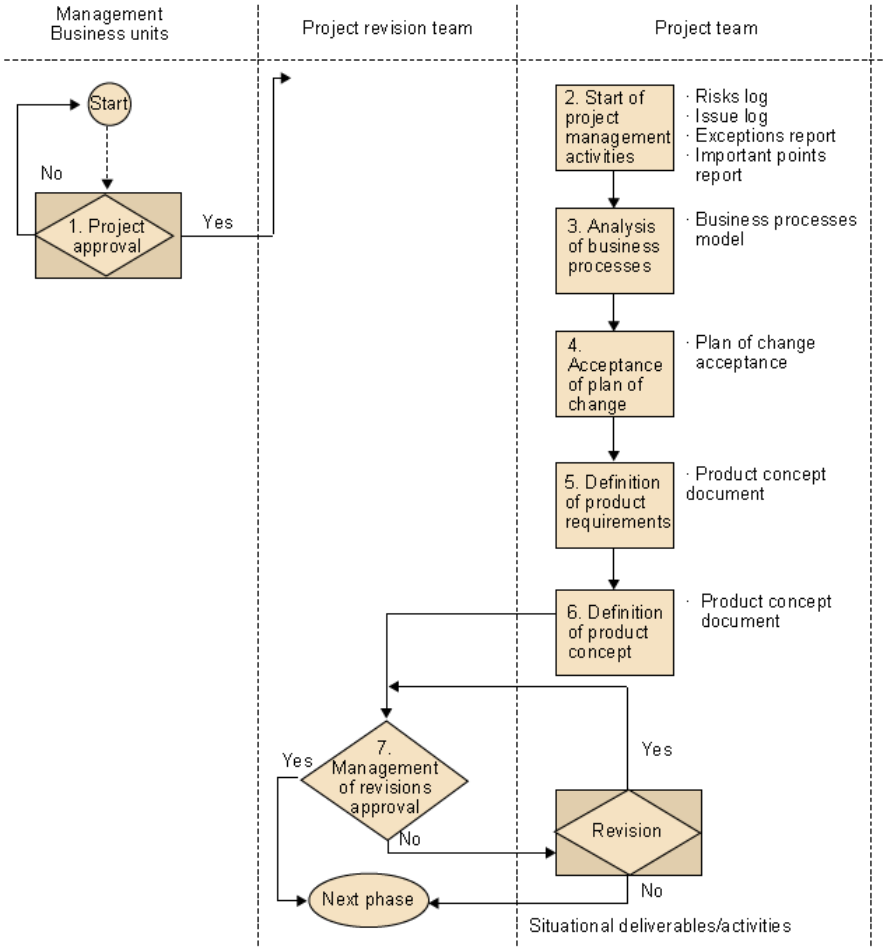
- Customer acceptance phase. This phase can be seen as a condensed version of the previous phases, since it allows the company to review all of the deliverables obtained in the previous phases and approve the project for launch.
  There are no risks other than the possibility of the product not being ready for the launch phase.

- Launch phase. This ensures that controls and measures are in place for the solution in order to ensure that the product meets the client's needs for the duration of its life cycle.
  The risk associated with the product at this stage is that it may be unsuccessful if any of the previous phases were carried out inadequately.

3.2.4. Process flows for the implementation of a project at Sun Microsystems

Concept phase

Figure 27.



1) The flow diagram is used as a starting point for the project conceptualisation phase.
2) The key activities at this stage are risk management, issue management and the progress report.
3) The business process is only analysed when the project team considers that the result can be used as a starting point for refining the business requirements.
4) The change approval plan is implemented to ensure that the changes are accepted by those in charge of decision-making.
5) The business requirements are converted into product requirements. Before this, a document indicating the product's requirements is required in order to define them correctly.
6) A high-level product concept is defined, which must be approved by those in charge in order to develop the solution.
7) The resources needed to carry out the project are approved (staff, financial and technologies).

Planning phase

Figure 28.

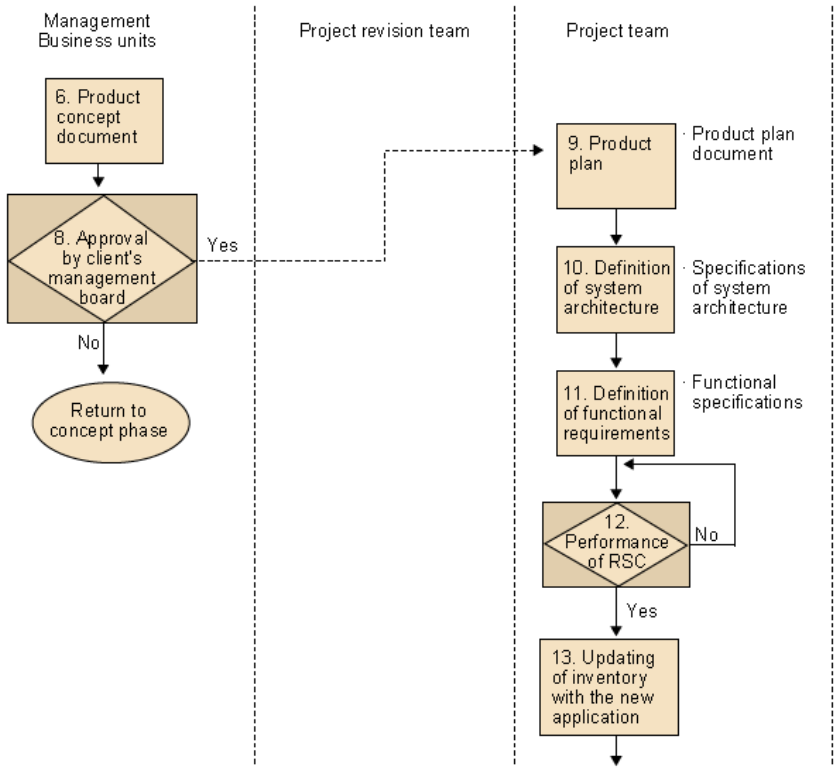
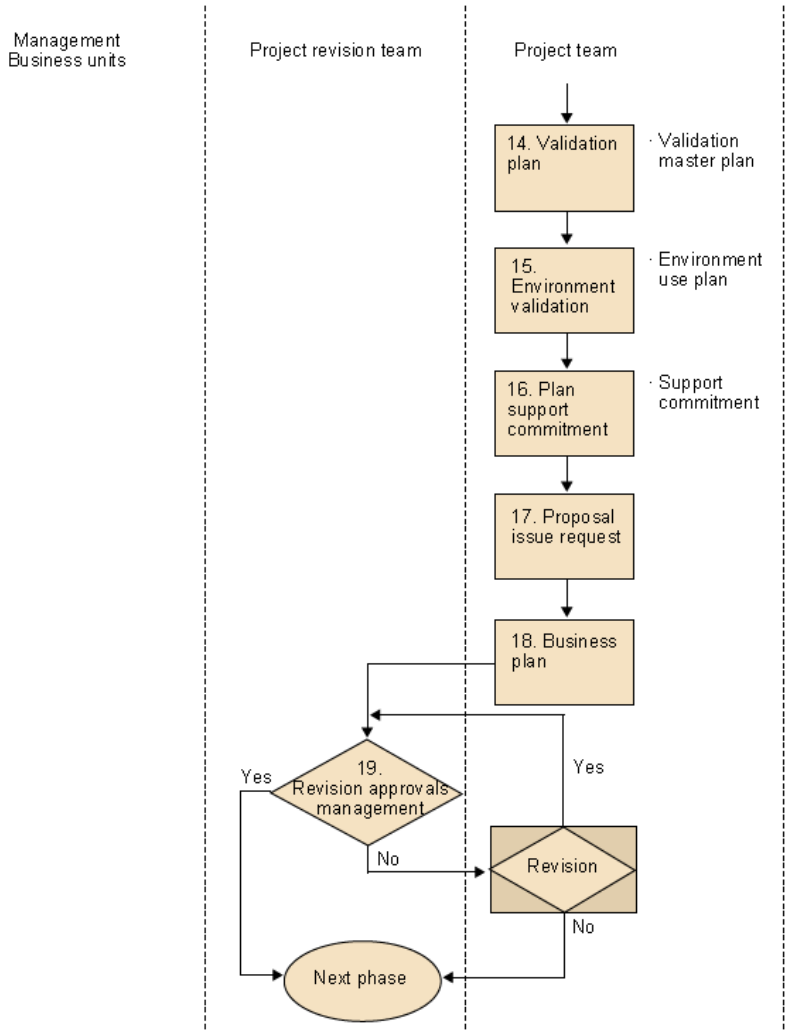


Figure 29.



8) If the solution is approved but requires acceptance of some sort from the end client or department (for which the product is being developed), the approval of a higher-level body will be needed.
9) At this point, the project can be given the go-ahead as it already has a detailed implementation plan.
10) The system architecture specifications are defined and approved based on the documents generated up to this point: product requirements, product concept document, etc.
11) The specifications that define all the features of the end product, are created.
12) (RSC: review of security compliance). The solution is studied and checked to ensure that it does not breach any company or third-party security standards (if it is a project that uses a free software licence).
13) The product is registered in the company inventory.
14) The validation master plan, which is used for high-level testing, is created.
15) The environment in which the product will be tested is checked to ensure that it is correct.
16) The quality of the subsequent product support is verified.
17) The version management processes are launched.
18) The business preparation plan is defined for the effective launch of the solution.
19) Guarantees are put in place to ensure that the project complies with the quality requirements, architecture constants, security considerations, standards and policies.

Development and integration phase

Figure 30.

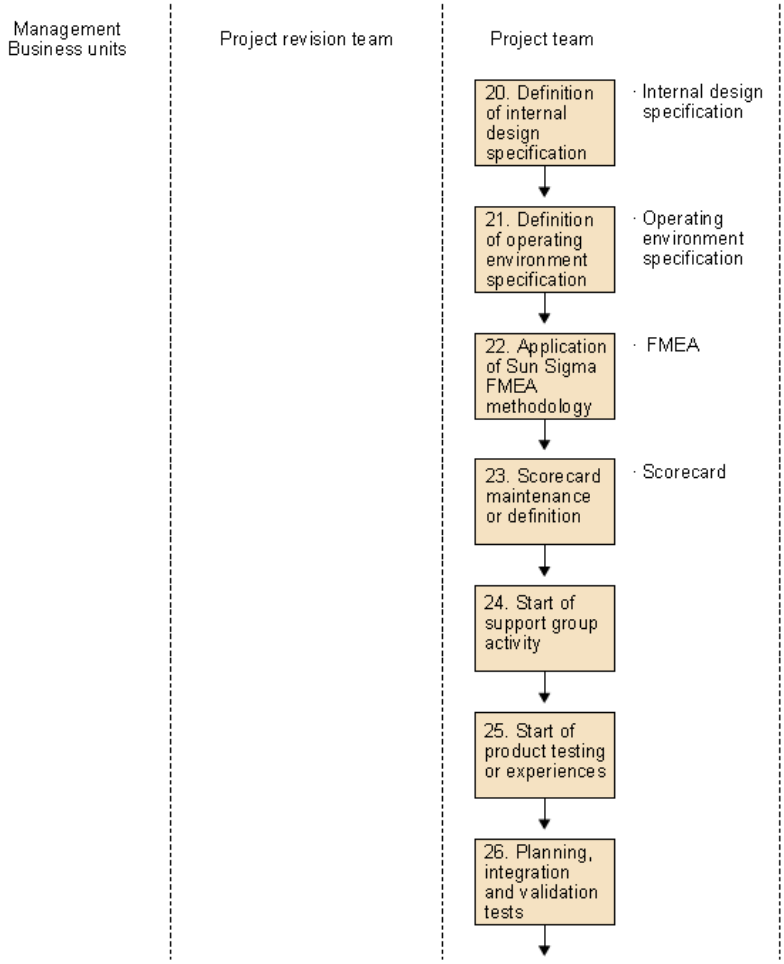
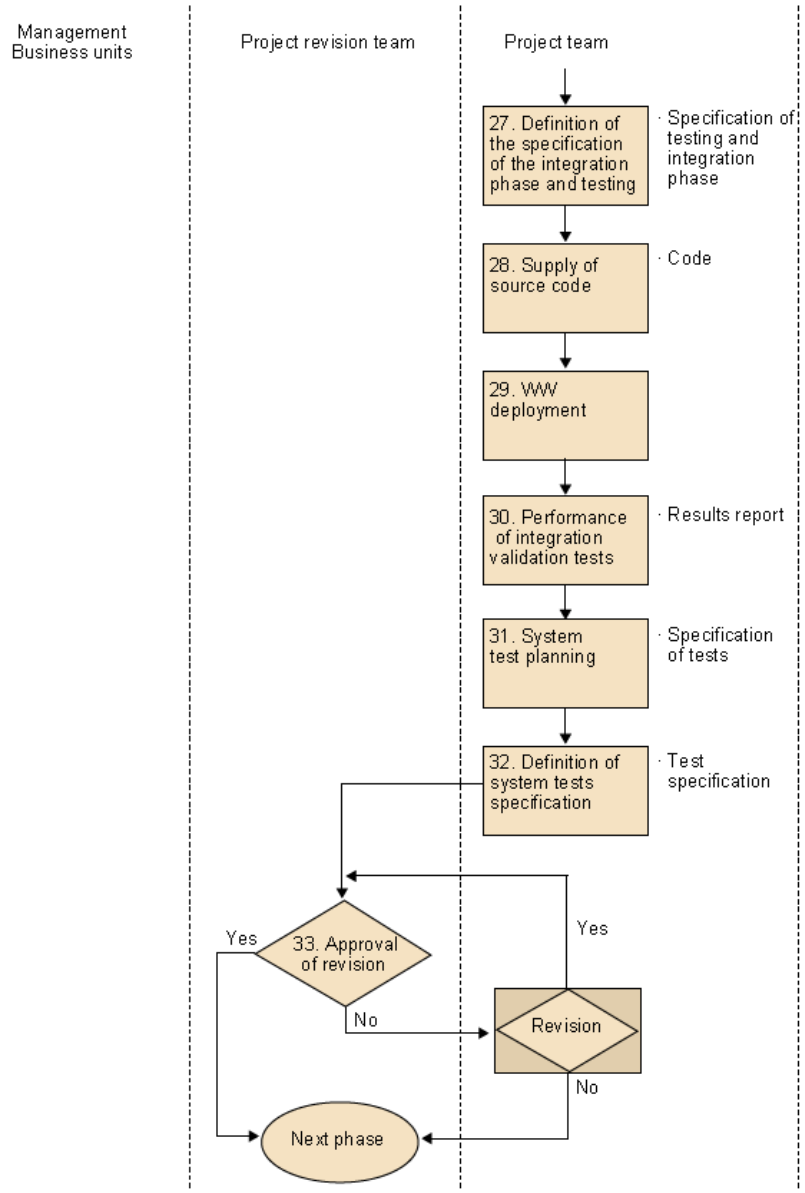


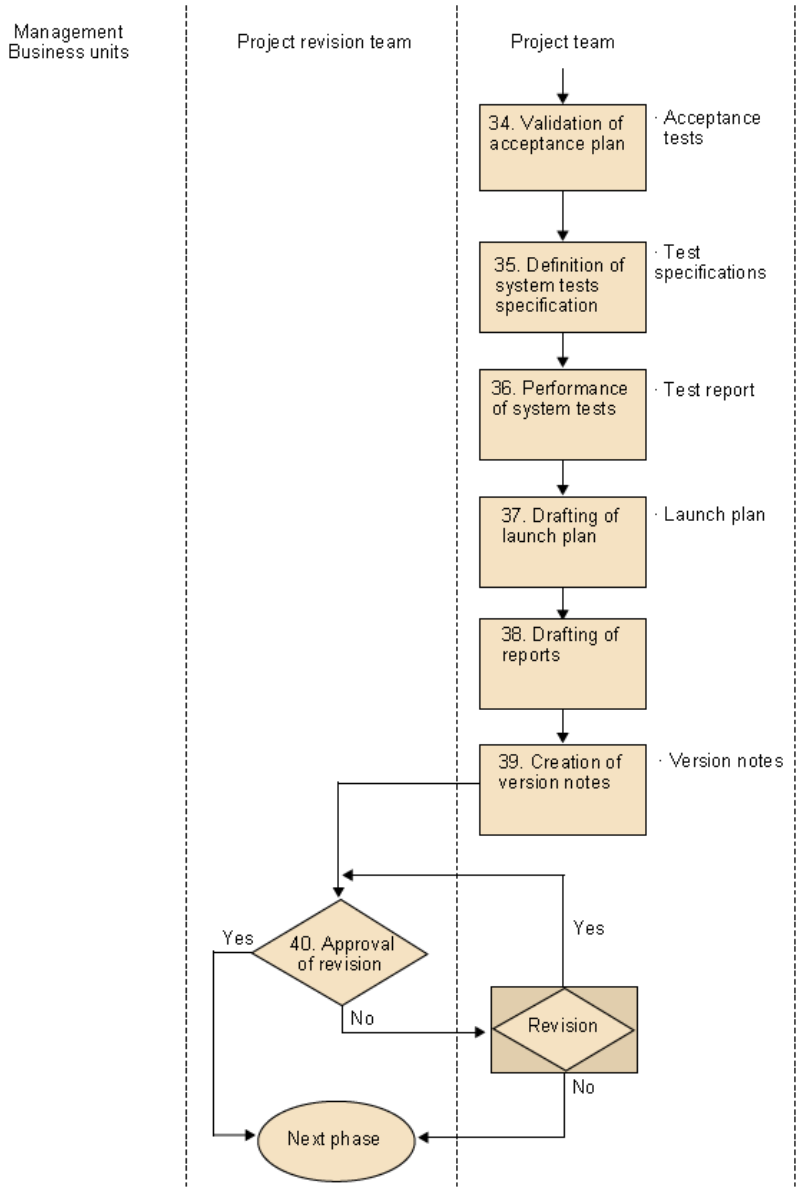
Figure 31.



20) The technical design of the solution is obtained through the internal specification design.
21) The operating procedures are defined through the operating environment specification.
22) The product is guaranteed to be "fault tolerant" using Sun Sigma techniques.
23) A scorecard is created to measure the critical parts of the design.
24) New contact is established with the product support groups.
25) Start of the experimentation phase for testing optimum product performance.
26) Planning of the integration phase and final product testing.
27) Definition of the specification of the integration phase and testing.
28) Registration of the product's source code.
29) The first global deployment is conducted in a controlled environment with a small group of end users.
30) The deployment integration and testing phase is carried out.
31) System verification testing is planned.
32) Definition of the system verification specifications.
33) Validation with the Management Committee to ensure that the solution meets the technical and functional requirements and that it is ready for the validation phase.

Validation and product testing phase

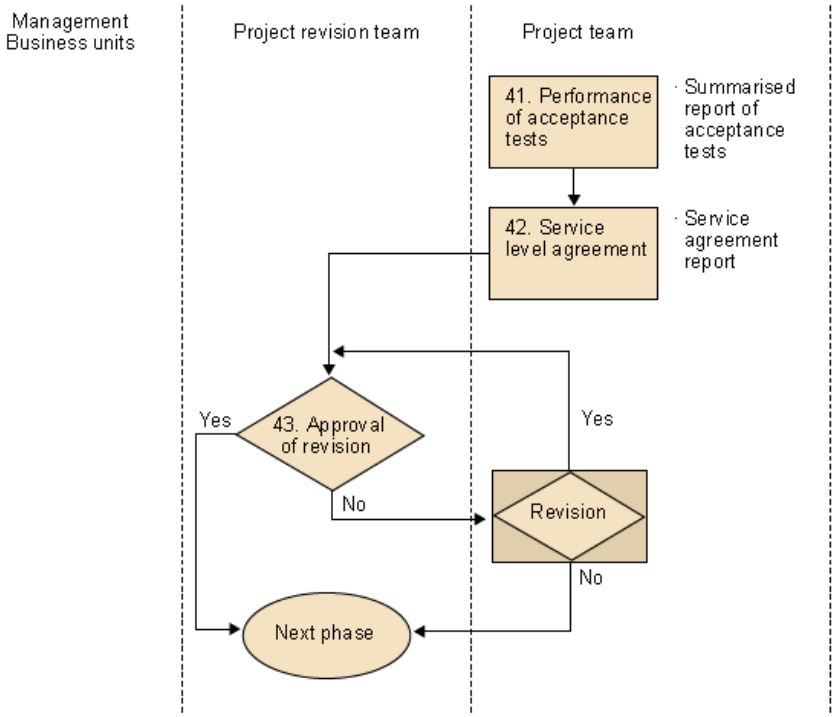
Figure 32.



34) Validation of the acceptance plan.
35) Definition of the testing acceptance specification.
36) Execution of the validation plan. The validation report is drafted.
37) End of the validation plan.
38) Drafting of reports.
39) Creation of version notes.
40) Validation in conjunction with the Management Committee to ensure that the solution meets the technical and functional requirements and that it is ready for the acceptance phase.

Client acceptance phase

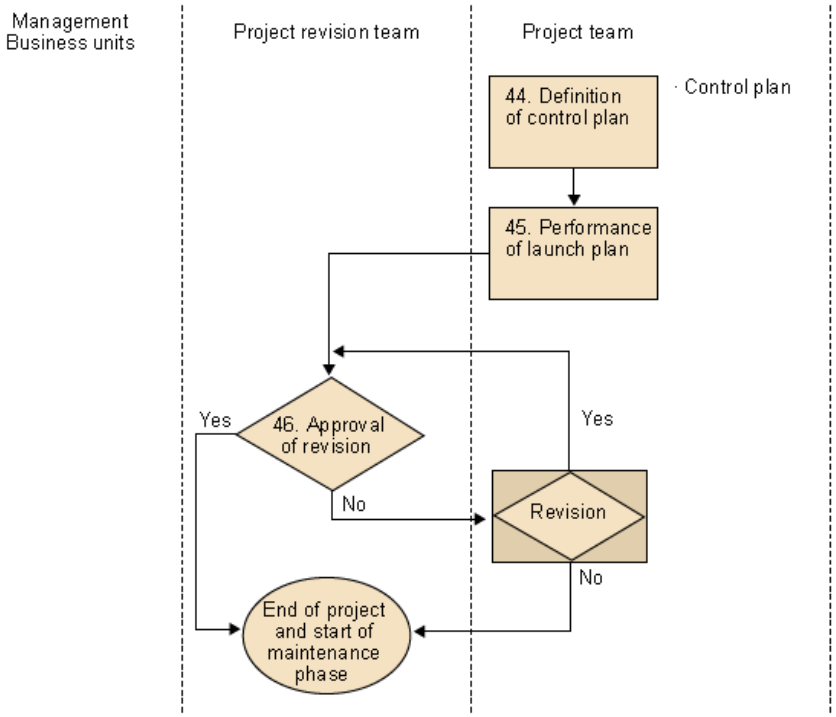
Figure 33.



41) Acceptance testing is carried out.
42) Creation of a service type agreement.
43) Review and acceptance, where applicable, of the management of the client acceptance phase.

Launch phase

Figure 34.



44) Completion and approval of the control plan.
45) Execution of the previously approved launch plan.

46) Final validation with the management committee to guarantee that the solution meets the technical and functional requirements and that it is ready for the maintenance phase.

3.3. Sun's free software positioning

Sun Microsystems has adopted a clear stance with regard to the creation and evolution of software. From the outset, the company has always encouraged the development of hardware and software products based on open standards, with a view to fostering free competition through the publication of protocols and interfaces so that other companies working in software creation can compete freely.

Sun considers it necessary to support the free software community by contributing source code and human and financial resources to allow the community to improve or adapt these programs and for Sun Microsystems to subsequently launch them on the market with a series of added services, such as support, training, etc.

Figure 35.

After releasing the code of Solaris 10, Sun Microsystems became the number one organisation for lines donated to the free software community, ahead of the University of Berkeley.

3.4. Study of free software applications implemented inside the company

When Sun Microsystems decided to launch a workstation system based on free software technology, it did so with a series of conditions:

- To create a workstation for all types of user with a true business focus, as an alternative to Microsoft Windows.

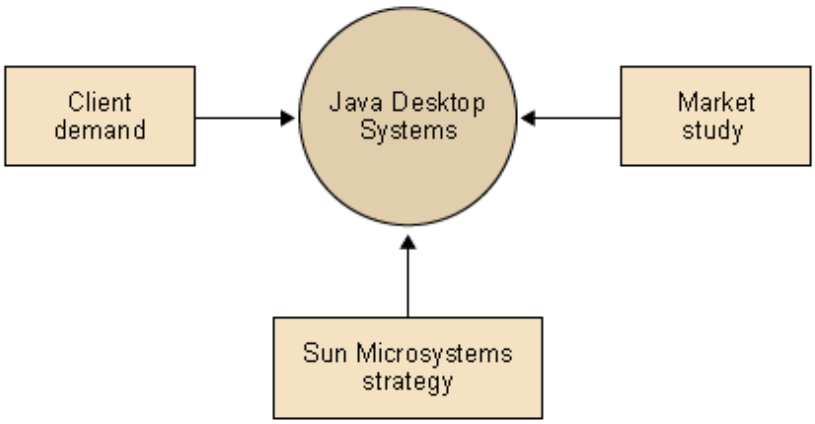
- To adapt and use this workstation to make it compatible with Sun technology, and to allow the company's 30,000 plus employees around the world to benefit from use of the product.

The project, known internally as "Mad Hatter", would thus have two types of client: on the one hand, the employees of the company and, on the other, users and companies. Although this case study will deal primarily with the first case, i.e. the launch of a desktop system known externally as the *Java Desktop System* within Sun Microsystems, the phases prior to distribution of the product were exactly the same, regardless of the end client of the product.

3.4.1. Market needs and study

The first step before deciding on the applications that would form part of the project was to pinpoint the needs of the market and the associated risks and benefits. After identifying these, they had to be guaranteed to fit in with Sun's strategy.

Figure 36.



The product had to meet the following aims in order to meet market needs:

- To provide a series of applications based on open standards that were integrated and would provide an interoperable, secure and well-defined working environment.

- To guarantee a working environment for users that would make them feel comfortable and be familiar to them.

- To use pieces of free software code to avoid being bound to a single technology.

The desired workstation features for clients were:

- An open and cheap workstation allowing them to be set free from the bonds of proprietary technologies.
- An elegant design that was easy to handle and manage.
- A working environment that could let them rest easy about computer viruses.

After analysing the market needs, Sun had to determine which applications would form part of the new workstation:

> Given that Sun's main aim was to gear the product towards professionals, it had to be clear on the fact that employee needs and uses of workstations are very different to those of users in non-professional environments.

- An environment in which to run applications.
- An e-mail client, directory and calendar.
- A browser for visiting Intranet/Internet sites.
- An office automation suite for giving presentations and producing text documents, spreadsheets, etc.
- An intuitive graphical environment to run the above applications and other less important ones that had to be included nonetheless.
- All applications had to run on any of the Sun Microsystems operating systems:
  - Solaris (SPARC architecture)
  - Solaris (x86 architecture)
  - Linux (x86 architecture)

The first version of the Java Desktop System (JDS) workstation had the following components:

Figure 37.

Web Browser	Mozilla 1.4 (Browser)
Mail, Calendar, Contacts	Evolution 1.4 (e-mail client, calendar and contacts)
Office Applications	StarOffice 7 (office applications suite)
Instant Messaging	Java 1.4 environment (platform for the applications)
Java	GNOME 2.2 (desktop environment)
Desktop Environment	Operating system:
Operating Environment	Solaris (SPARC, x86)
Java Desktop System	SuSE Linux (x86)

3.4.2. Study of the Java Desktop System (JDS) applications

Before describing each of the applications in detail, we ought to think about the value of Sun's idea of offering a workstation such as Java Desktop System based on open software:

- The appearance of the graphical environment had to be as similar as possible to Windows operating environments because this is the most commonly used PC platform in the world.
  The GNOME desktop environment has a series of menus and icons to allow as smooth a transition as possible.

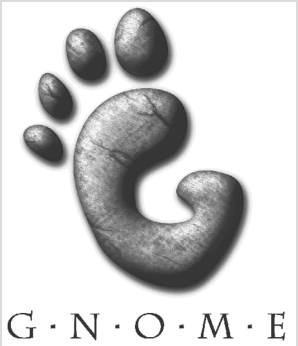
  Many business users are not advanced users of IT systems, so the main aim was to avoid rejection by this group.

- Device management had to be straightforward. It is often necessary to connect the workstation to an external device (scanner, printer, etc.) and this operation should not cause problems for users.

- The workstation had to be integrable and interoperable and not limited to a specific type of application. Therefore, it had to be based on open standards, as this would allow its integration with existing technologies in any business environment (directory servers, e-mail, databases, applications based on the Windows operating system, etc).

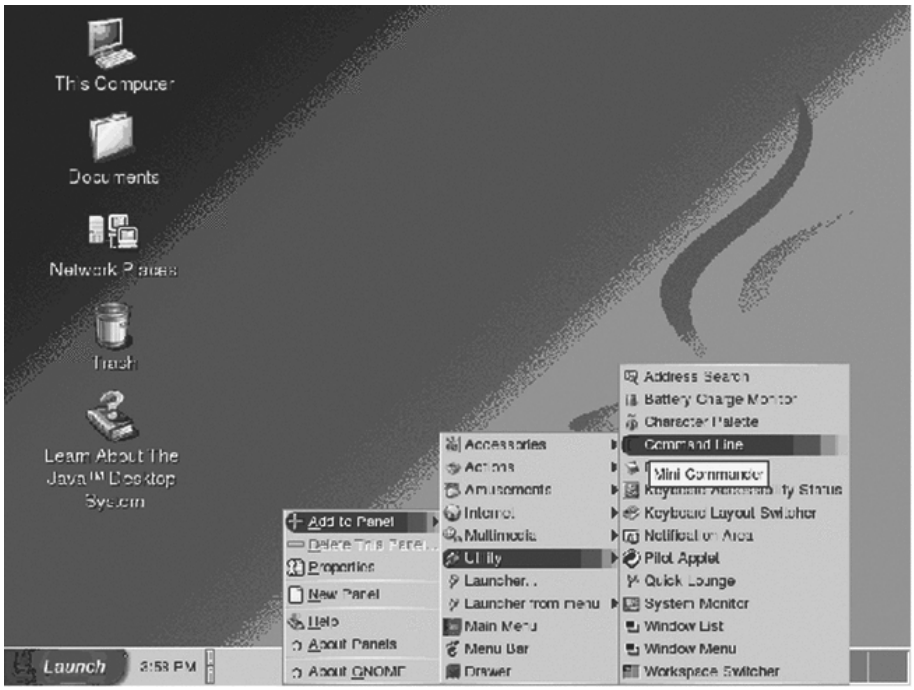
GNOME graphical environment	Description and features
Figure 1[1]	GNOME is most popular desktop environment in the free software community. It has a familiar operating system management appearance with menus, icons, file management, accessories, system tools and a range of customisable screens that provide a comfortable environment for the end user to move around in.

(1)



The appearance of the working environment with GNOME looks like this:

Figure 38.



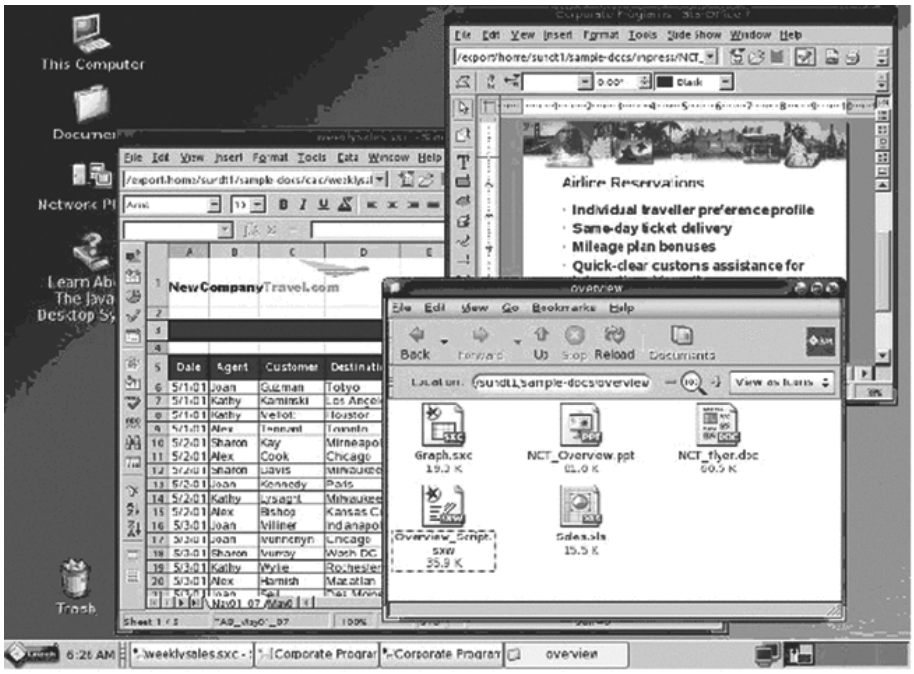
One tool that is a must in professional working environments is an office suite. The tool with the biggest market impact at the moment is Microsoft Office, which is why Sun Microsystems has invested considerable human and financial resources in producing a market alternative that matches its features and guarantees 99% compatibility.

StarOffice office productivity suite	Description and features
Figure 2[2]	Ease of use. Anybody who has used MS Office is able to work with the StarOffice tool with no further training. Interoperable. It can export any type of document to a variety of formats, including PDF, Flash XML, .doc, .xls, .ppt, .rtf, .psw., etc. Open format. It works with files with an internal format based on XML.

(2)

The tool looks like this:

Figure 39.

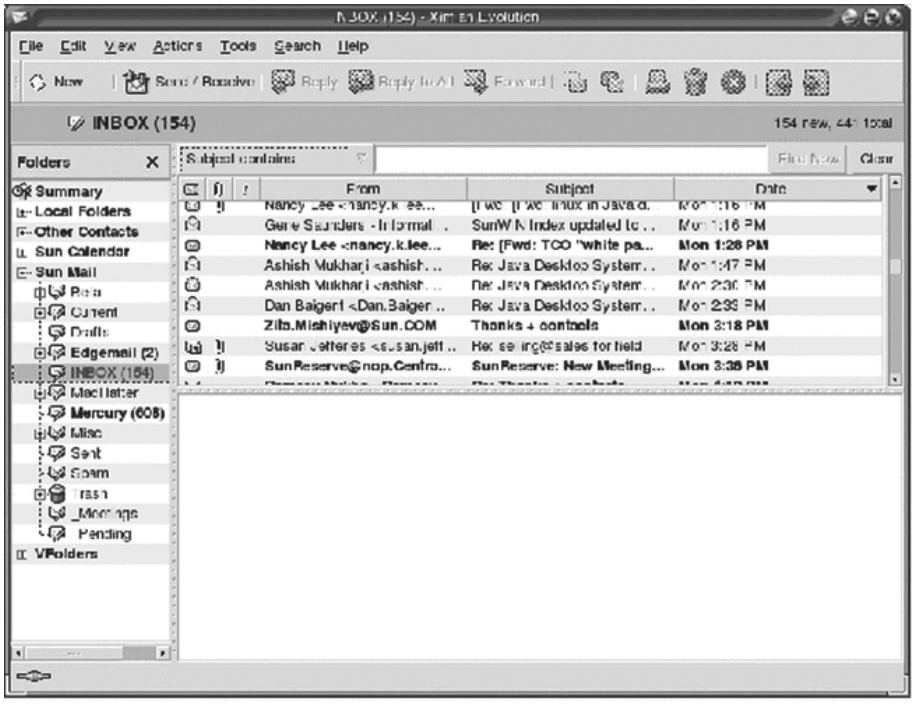


Ximian e-mail client	Description and features
Figure 3[3]	E-mail client, directory, calendar, etc. It supports the most common e-mail protocols and can be integrated with a calendar server. It is very similar in appearance to the MS Outlook client.

(3)

You can see what Ximian Evolution looks like in this screenshot:

Figure 40.



Mozilla browser	Description and features
Figure 4[4]	Browser that supports the HTML, XML and CCS standards, among others. It includes an HTML editor, download manager and a wide range of advanced features. It is installed with pre-configured plug-ins for Java, Macromedia Flash, Adobe Acrobat Reader, RealPlayer, etc.

(4)

The appearance of this browser is as follows:

Figure 41.

There are many other applications that come with the ones described here but they are not included in this study because, while useful, they are not essential to the day-to-day running of a business.

3.4.3. Control metrics used by the IT operations department

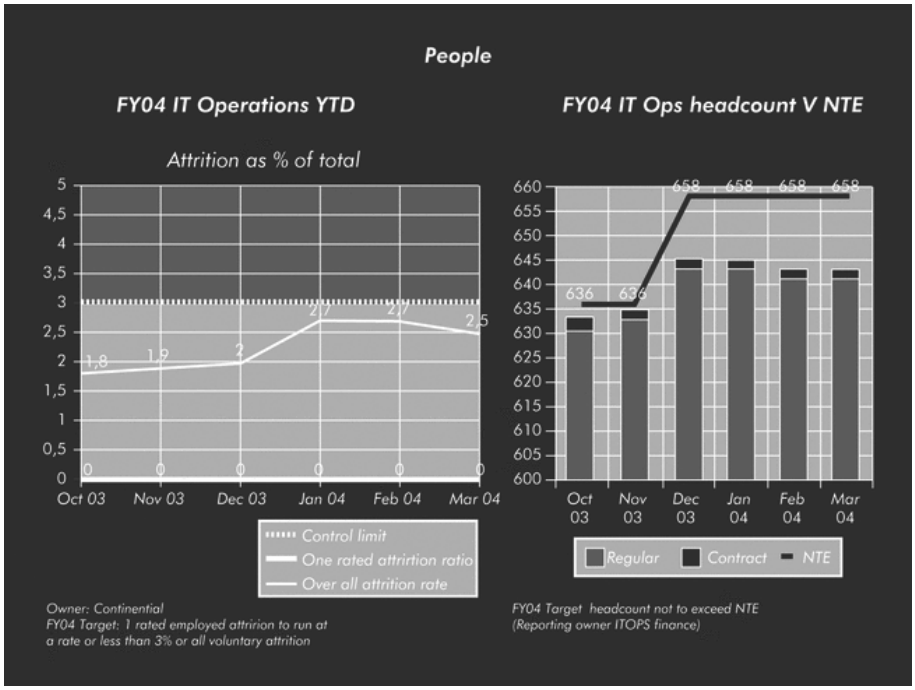
One of the most significant processes of the design and subsequent implementation of a free software system at Sun are metrics, which help to predict the guarantees of success of a product.

There are many metrics for estimating costs, risks, quality, efficiency, etc., studied by the Head of Operations for the geographical region in which the system is to be implemented.

The key data managed in these metrics are those concerning the staff available to implement the applications correctly. They include the ratio of end users to the number of engineers occupied with these tasks.

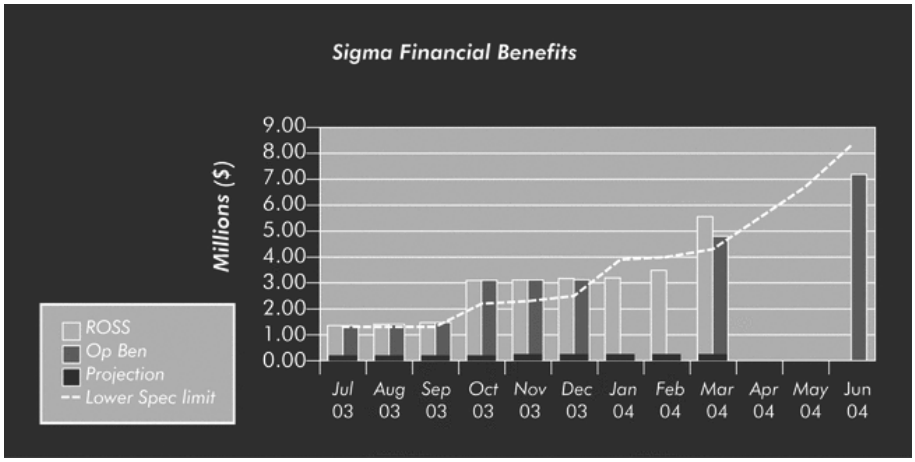
The Sun Microsystems IT Operations Department has a total of 265 staff administrating and installing corporate applications on over 1,810 servers, which serve the 35,000 plus employees. To put it another way, each system's administrator is responsible for managing and monitoring more than 135 workstations.

Figure 42.



Another interesting metric is that concerning quality and productivity: Sun Sigma methodology. Sun Sigma is an adaptation of Sun's Six Sigma methodology, which is basically a scientific method for improving processes, products and services. All decisions on the introduction of improvements are based on data collected from the various departments.

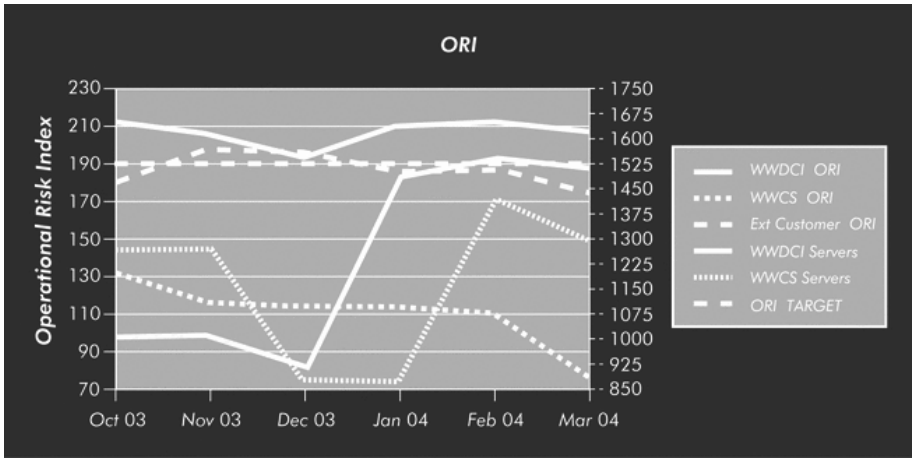
Figure 43.



These metrics can be used to pinpoint the areas in which processes can be improved in order to help to cut costs.

ORI (*Operational Risk Index*) is an index obtained by analysing the environment of Sun's systems in production. It analyses the systems and studies their risks, classifying them according to status (low, moderate, high and critical). Once we have the values for each system, we can then obtain their risk of failure.

Figure 44.



The above diagram illustrates the ORI of the systems in some departments of Sun Microsystems in the EMEA (Europe, the Middle East and Africa) region. As this index concerns the number of servers in production, this number is indicated to the right of the diagram.

The formula for obtaining a department's ORI is:

$$(10*\text{critical}) + (5*\text{high}) + (3*\text{moderate}) + (1*\text{low}) = \text{ORI}$$

After analysing a system in production with a tool called Explorer, the results are compared to the values that Sun Microsystems believes the system should have in an ideal environment. Risks are classified by criticalness using the above formula.

A system has been evaluated and the following results have been obtained:

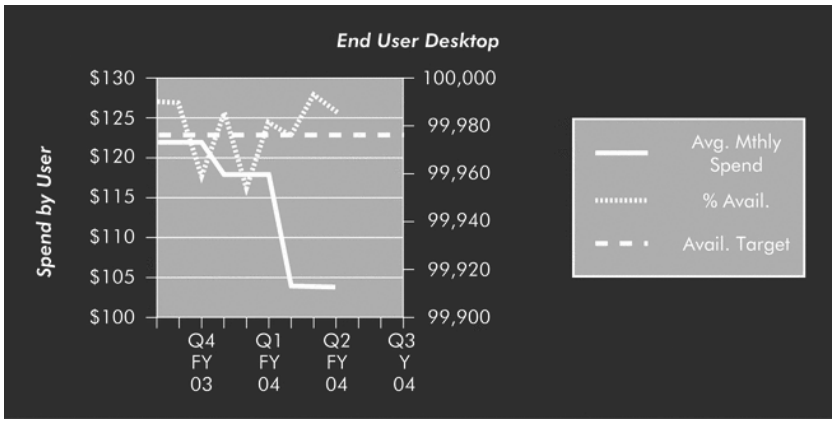
Risk level	Number of potential problems
Critical (critical)	2
High (high)	5
Moderate (moderate)	8
Low (low)	10

The ORI of this system would be 79.

$$(10*2) + (5*5) + (3*8) + (1*10)=79$$

Finally, it is important to obtain metrics for the efficiency of the workstation, local network services and wide area network services.

The figure below illustrates an example of the cost savings obtained by Sun through workstation efficiency:



3.5. Development and worldwide launch of the platform

When the product is ready to be installed on all file servers and SWAN network applications, the company needs to adopt a global approach to its *deployment* that will allow the software currently installed on the systems to be replaced with the new product. Deployment management allows the diverse groups responsible for hardware or software to coordinate implementation of the new system in an environment of centralised servers located in different countries around the world.

The responsibilities associated with this management can be summarised as follows:

- Planning and supervision of the effective deployment of the new software (and hardware if applicable), removing any programs classified as obsolete.

- Coordination with change management if any part of the software requires a simple upgrade.

- Guarantee that all the elements to be installed are secure and can be located in the corresponding database.

- Management of end-user expectations.

On occasion, installation affects a specific group of company applications that can be treated as a single group while at other times deployment only involves upgrading the version of an application. Hence, there are several types of launch, each depending on the nature of the components to be installed:

- Complete launch: all launch modules and components have been created, interconnected and tested as a single unit.

- Differential launch: only the components that have changed since the last launch are included.

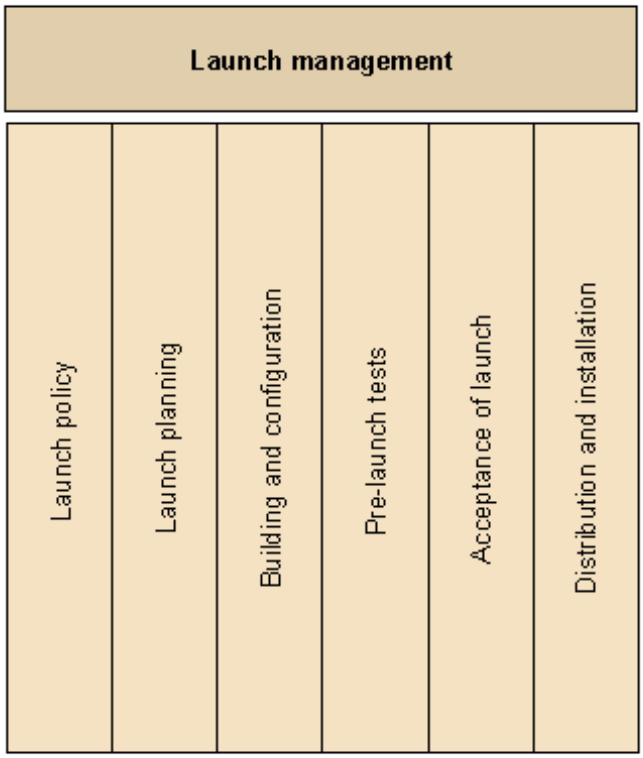
The corporate software database contains the master copies of all software controlled in the company (both purchased programs and software developed in the R&D laboratories). The exact configuration of this database must be defined and revised before any launch.

The software components included in a new installation should be assembled under controlled conditions to ensure a reproducible process. It is quite common to automate this process to reduce dependence on human involvement, thus increasing reliability. Sun Microsystems has developed an entire systems and processes architecture called *N1*, by which this process can be carried out with minimal human intervention.

The launch must undergo rigorous testing and obtain user acceptance. Due to the high number of end users in a multinational like Sun, a small group of users (with different technical profiles) is selected to pass and validate the tests for global launch. Sometimes, despite completing each of the steps described in the launch procedures, a software version can generate a run error or prevent the installation from working correctly. In this case, we need a backtrack plan that documents the steps to take after a launch fault.

3.5.1. Launch management

Figure 45.



A centralised launch management like the one described has countless advantages over a non-centralised one:

- The installations are carried out on a limited number of servers, which means that the database of registered equipment is not too large.

- The time spent on the last phase of the launch is reduced to a few minutes.

- The performance of backtracking procedures and reinstallation of the previous software is also a matter of minutes.

  This is very useful when an error occurs or a fault is generated in one of the installed applications.

- Improved quality of service due to more successful installations and a considerable reduction in business downtime, which is zero in most cases.

- It ensures that the installation remains in the hands of professionals who specialise in this type of management: the end user will never have to perform operations with the installed software.

- Improved use of resources.

3.5.2. Distribution tools

The IT Operations Department of Sun Microsystems uses a procedure called *SoftDist* to distribute the software once it has been launched. The aim of this process is to ensure that the central servers in each of the regions have the latest version of the software that the company wishes its employees to use.

Before moving on to the steps for distribution of this software, we need to be familiar with the following terms:

- *Submitter*: Sun Microsystems employee in charge of preparing packages for distribution.
- Package: software application that can be used by any Sun employee inside the company.

Requirements for software distribution

- Identification. All packages for installation must be assigned to a Sun employee before distribution can begin. This employee is usually the Head of the Operations Department for the region. The information required is:
  - Name of the individual or department group.

  - Identification and supply of the resources needed to deal with any problem with the application, regardless of whether the questions concern the end of the product's life, support, bugs or maintenance.

  - Identification of the contact person for any product support issues.

- Approval. All packages for installation must be approved by the Head of the Operations Department for the corresponding region, which guarantees functionality of the application and coordination of compatibility with other products, databases and applications relevant to the business aims. The Head of Operations must be perfectly aware of the company's aims, the risk of a fault in the product launch and the product requirements in order to determine departmental resources needed to make the distribution a success.

Product distribution requirements

- Use of `preptool`. This tool, exclusively for Sun Microsystems' internal use, allows us to combine and organise the data needed to prepare a package for distribution. All of the relevant information must be entered to ensure operability of the package across the company.

- Export control. Since Sun Microsystems is a US company with offices in different countries, the product must meet the import/export regulations of each country. These regulations are described in Sun's export control department (*Sun International Trade Services*).
  The *SoftDist* distribution hierarchy and the network topology allow software packages to be distributed to all Sun offices around the world, even in countries where trade with the US is restricted, thanks to agreements with their governments.
  The products developed at Sun Microsystems R&D centres in countries other than the United States must also abide by the regulations of the International Trade Services department.

- Contact with the Product Support Centre. The Head of the Operations Department must contact the Head of the Support Centre as part of the pre-distribution preparations for the package in order to define the product support mechanisms and the support resources and contact person.

Verification of package functionality during distribution

- Functional testing. The Head of Operations must ensure that the packages for installation will not compromise the productivity or functionality of other applications and that they will not affect business operation from the point of view of systems performance.

The packages for distribution must run correctly on the two operating systems supported by Sun: Solaris and GNU/Linux.

As explained earlier, the packages must be validated in a simulated environment before global distribution. The package proprietor is responsible for setting the criteria to determine whether the package is suitable for distribution; in most cases, this figure is the person responsible for development of the product or the Head of IT Operations.

- Management of application faults. The Sun IT Operations Department, acting under the head of this department, must provide global network users with a guarantee period and respond to any issues raised during this period, which begins on distribution of the package. Risk situations are processed differently from the standard procedure.

  If a bug has not been detected in the application for distribution and it appears within the first week of distribution, the Operations Department must assume any expenses arising from this moment on. This same procedure is applied when the distributed package has a severe impact on Sun systems performance, causing delays that affect normal activity or prevent users from opening other applications from their workstations. The criteria for determining whether a package has been distributed correctly are as follows:

  – Users must be able to launch the application from their workstations without administrator intervention.

  – The application cannot access areas that the user launching it does not have permission to access.

  After distribution of the software, the Head of IT Operations for each region must obtain metrics of its results by auditing the installations and reporting any unsuccessful cases, detailing the reasons for them and the plan of action to be put in place.

Figure 46.

	Q1 FY04			Q2 FY04			Q3 FY04			Q4 FY04			Q1 FY05			
	Pass	Fail	Exempt	Pass	Fail	Exempt	Pass	Fail	Exempt	Pass	Fail	Exempt	Pass	Fail	Exempt	
Printer Alias	100.00%	0.00%	0.00%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Printer Alias
Printer Firmware	100.00%	0.00%	0.00%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Printer Firmware
Console Server	98.36%	0.00%	1.64%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Console Server
DHCP	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	DHCP
ENSMail	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	ENSMail
IAS Audit Client	97.84%	0.00%	2.16%	95.59%	0.00%	4.41%	97.95%	0.00%	2.05%	99.49%	0.00%	0.51%	98.35%	0.00%	1.65%	IAS Audit Client
Infrastructural Aliases	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Infrastructural Aliases
Samba	97.73%	0.00%	2.27%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Samba
SOMS	0.00%	0.00%	100.00%	100.00%	0.00%	0.00%	87.50%	0.00%	12.50%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	SOMS
Sun Cluster	95.83%	0.00%	4.17%	100.00%	0.00%	0.00%	70.97%	0.00%	29.03%	78.57%	0.00%	21.43%	100.00%	0.00%	0.00%	Sun Cluster
SUNMC 3.0	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	98.97%	0.00%	1.03%	99.45%	0.00%	0.55%	97.84%	0.00%	2.16%	SUNMC 3.0
Sunray	95.24%	0.00%	4.76%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Sunray
Sunray Naming	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Sunray Naming
Timetone	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Timetone
Softdist	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Softdist
Java Desktop System	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	98.89%	0.00%	1.11%	Java Desktop System
ITNetGen	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	ITNetGen
NAMM	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	NAMM
Printer	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Printer
Patches	N/A	N/A	N/A	100.00%	0.00%	0.00%	94.59%	0.00%	5.51%	79.31%	0.00%	20.69%	96.76%	0.00%	3.24%	Patches
Webcache	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Webcache
Puppet	N/A	N/A	N/A	80.00%	0.00%	20.00%	97.56%	0.00%	2.44%	97.30%	0.00%	2.70%	97.44%	0.00%	2.56%	Puppet
Regtool	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Regtool
Backup	N/A	N/A	N/A	N/A	N/A	N/A	97.30%	0.00%	2.70%	96.97%	0.00%	3.03%	96.97%	0.00%	3.03%	Backup
GSM	N/A	N/A	N/A	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	GSM
UAM	N/A	N/A	N/A	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	UAM
Printer Centralization	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%	Printer Centralization
Total Results:	**92.33%**	**0.00%**	**7.67%**	**98.84%**	**0.00%**	**1.16%**	**97.70%**	**0.00%**	**2.30%**	**98.04%**	**0.00%**	**1.84%**	**99.45%**	**0.00%**	**0.55%**	**Total Results:**

Installations have a quarterly review, at which the support group covers issues detected after the first week of installation.

Other metrics that need to be taken into account by Sun's Operations Department are DPMO (Defects Per Million Opportunities), which attempts to reflect the final result of the quarterly software distributions.

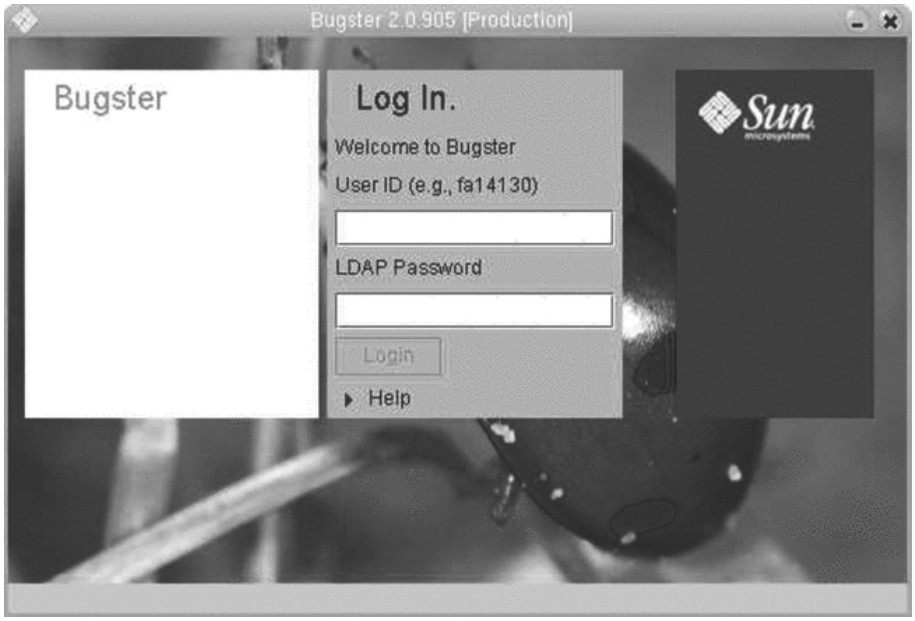
3.6. Product support

We have already explained that the IT Operations Department is responsible for the correct performance of applications (both free and proprietary software) at Sun. However, any problems detected in distributed packages, whether in design or development, are the responsibility of the product development group.

Any detected bugs can be fixed during the product maintenance phase but it is often the case that a bug found in a package can have a direct impact on a systems performance, and hence on the day-to-day activity of the company. To avoid these problems, there is a standard procedure for assigning a higher priority to the JDS code review, allowing the engineers in charge of product maintenance to solve it as soon as possible.

When a bug is found in a package or application, it is usually reported to the Operations Department, which must then contact the product support team. To do so, Sun has created a bug reporting tool for users to enter information on the type of bug, the part of the application where it occurs, etc.

Figure 47.



The tool can be accessed from any workstation. Users simply need to enter "bugster" and a Java application will ask users to identify themselves and enter the corresponding information.

3.7. Corporate training

The Sun Microsystems Education Department draws up tailored training plans for departments that require special JDS workstation training, offering users intensive courses on the product (user level). This training is only provided if formally requested by the department that needs it, which must also indicate the staff that will attend the courses.

There are different course formats depending on the availability of the employees:

- Users with basic training needs or who cannot leave their workstations can attend on-line training with Internet courses.
- More advanced groups or employees who are able to travel can receive on-site training with an instructor.

In all events, each department must assume the costs of training its staff.

4. Cometa Technologies

4.1. Introduction

This chapter will look at open source software from a business angle, using the real case of an SME, Cometa Technologies, that uses it intensively.

As we will explain, when a company chooses to use free software in the technology solutions it adopts, it opens itself up to a series of benefits that would not otherwise exist or be so obvious. For example, the use of free source software results in cost savings, supplier independence, shorter development times, etc. With open source solutions, the suppliers of related services also reap the benefits.

Free software offers a myriad of possibilities to companies wishing to provide related solutions. This chapter will describe the basic services that Cometa Technologies offers to its clients in order to help them rise to their challenges: development and integration of projects based on the use of free software tools, and training and consulting on the subject.

The cases described in the different sections show how Cometa Technologies uses free software in projects for very diverse companies. They show how this type of solution is now a real alternative to those based on proprietary software.

At the end of the chapter, the student should:

- understand the basics of the business model of Cometa Technologies, as an example of an SME that bases a considerable proportion of its business on the use of free software;

- understand the business benefits of an intensive use of open source tools, from the point of view of a technology solutions provider and from that of clients;

- be able to reflect on the main services associated with free software that companies can offer; and

- be familiar with real examples of free software in diverse business sectors.

4.2. Company presentation

Cometa Technologies is a private sector company that provides information technology solutions based on free software tools and standards.

Figure 48.

Website

Visit Cometa Technologies' website at: http://www.cometatech.com.

Broadly speaking, Cometa Technologies offers two lines of services to clients:

Note

We will look at these services in detail later on.

- Development and integration of technology solutions. This line of solutions is designed to fully adapt to the functional needs of clients and to make the most of the products available in order to minimise development and implementation time.

- Training and consulting. This line of solutions has been designed to equip clients with the knowledge to choose the right technologies, know their possibilities, and use them.

The staff of Cometa Technologies are experts in information technologies and in using them to solve real business problems. With its knowledge of the free software movement (philosophy, aims, available tools, etc.), Cometa Technologies can help its clients to tackle their concerns by applying and integrating the most appropriate technologies into long-term reliable solutions at economical rates.

Cometa Technologies was founded to bridge the gap between the free software community (their knowledge of the tools that they have produced, how they relate to each other, which ones are the most interesting, etc.) and companies (which need turnkey solutions with subsequent maintenance).

This knowledge and its application constitute the *core competency* of Cometa Technologies, a skill that sets it apart from its competitors and gives it an edge, since:

- it offers potential access to a wide variety of markets;
- it increases the benefits for clients; and
- it is difficult to imitate.

Given that Cometa Technologies offers solutions through the use of technology, we will now describe the main technical features of the projects that it carries out:

- Reliability and scalability. The solutions provided by Cometa Technologies are stable, always function as planned and can grow with the client company.

- Application of the most appropriate technologies. The Cometa Technologies team is aware of the possibilities of the various technologies available at any given time and is thus able to apply the most appropriate one to each case.

- Use of free software tools. Cometa Technologies specialises in the use of this type of tool, which means that it can take full responsibility for their correct operation and incorporate their advantages into the projects in which it participates.

Note

We will return to this point in the "Use of free software at Cometa Technologies" section.

- Use of standards. Cometa Technologies uses technology standards wherever possible, both in the solutions it develops internally (tailored development) and in those incorporating third-party products (open source project integration).

4.2.1. Working methodology of Cometa Technologies

The life cycle of a Cometa Technologies project generally encompasses the following phases:

- Initial study in conjunction with the client of the new needs and possible ways of meeting them.

- Drafting of a proposal that sets down the general lines of the suggested solution and the time/staff required to carry it out, together with a financial appraisal.

- Design of the solution, including a detailed analysis of free software tools and standards that could be used to add value to the project (shorter development time, enhanced security, etc). In this phase, constant communication with the client is critical for the validation of partial results.

- Acceptance and implementation of the solution with a strong focus on the training of the users or technicians who will be involved in the everyday operation of the solution so that it can be used to its best advantage.

- Maintenance of the solution to guarantee correct operation (corrective maintenance) and evolution (upgrade maintenance), depending on the needs of the client.

Cometa Technologies also markets its knowledge and expertise in certain areas of IT in a variety of ways: consulting, training, etc.

Note

The "Solutions offered" section contains a detailed description of the solution development phase of the project life cycle and the different approaches taken.

To sum up, we could say that the aim of Cometa Technologies is to offer a quality service to its clients, which means emphasising the following points in particular:

- Accessibility and communication. Clients are able to contact Cometa Technologies easily for the hire of its services. Communication with clients is fluid and constant and uses a language that both parties can understand (which is not always easy when talking about technology).

- Quick response. Clear willingness to respond and provide a quick service based on the description of the client's needs.

- Understanding of clients. Efforts are made to assimilate and deal with the needs of clients.

- Professionalism. Cometa Technologies has and uses the knowledge to propose the best solutions and implement them effectively.

- Credibility and reliability. The company portrays an honest image based on getting the job done well with total transparency and on providing the promised solutions (which work as planned) by the agreed deadline.

- Security. With the service provided by Cometa Technologies, clients do not feel that they are taking risks of any sort and there are no pitfalls or concerns about the service provided.

Note

Understanding of clients and professionalism are key to the activities of Cometa Technologies: the aim is to generate benefits for the client.

Its use of open source and standards allows the company to offer some of the service features described above, particularly those concerning factors such as reliability, professionalism and security, which can help to enhance the overall sense of quality perceived by the client.

4.3. Use of free software at Cometa Technologies

4.3.1. Information technology challenges for companies

In this climate of economic instability and constant technological change, all firms, whether small businesses or big corporations, face challenges in this area:

- Reducing the time to project launch from its conception. Information technology projects have to produce short-term results: we need to see the benefits of our actions and investments in a short space of time.

- Improving integration with other systems. Companies increasingly find themselves with inherited components in information technologies (rarely do they start from scratch), so project success is not simply about obtaining the expected results in isolation but rather that the project integrates seamlessly with the other technology used in the various business processes of the company.

- Increasing security. Companies need to prevent the theft of confidential information, keep their technology systems running around the clock, guarantee the privacy of the data they handle, etc. In our increasingly complex technological environments, security has become one of the chief concerns of many companies.

- Maintaining independence from suppliers. Companies are becoming increasingly interested in remaining independent from their technology suppliers in order to avoid situations (as has occurred in the past) in which suppliers, seeing that their clients are in no position to make demands of them, tend not to meet their needs as they once did (for instance, putting up their prices for no apparent reason).

- Improving the use of technology solutions, taking into account that they are used in different countries and cultures. It is increasingly common for companies to have systems used by international teams. Technology projects, like all other company actions, must take into account the need to adapt to this variety of circumstances.

- Cutting costs. The pressure is on to cut the TCO (*total cost of ownership*) and increase the ROI (*return on investment*) of projects carried out in our uncertain economic climate.

Note

* TCO: cost of hardware and software purchasing, staff costs (technical and user training, engineering, etc.), cost of support (installation, maintenance, etc.) and costs incurred through malfunctions (downtime, lost business opportunities, etc).

4.3.2. Role of Cometa Technologies in meeting business challenges

Since Cometa Technologies provides solutions through the use of information technologies, it needs to respond to the challenges faced by its clients. As explained earlier, the use of free software tools and standards allows Cometa Technologies the possibility of enhancing the quality of the services it offers and equips it with the know-how to help deal with these challenges. The following sections will look at each of these points in turn.

Use of free software

Free software provides us with the source code, which can be freely redistributed and modified. Thousands of people around the world create and maintain free software and millions more use it on a daily basis.

The advantages of basing solutions on free software for Cometa Technologies and its clients include:

- Increased reliability and security. The use of free software improves security, particularly because:
  - It offers greater transparency: anybody can detect and fix bugs; *a peer review process is used for this purpose*.
  - There are more privacy guarantees, since code audits can be carried out (this is impossible or difficult to do with proprietary software.
  - It allows independence from suppliers: a client company can freely decide to terminate relations with a supplier if dissatisfied with its services. The company will not find itself trapped again because it has everything it needs to continue the project alone or with a new supplier.
- Enhanced performance and scalability. In some cases, we can use free software tools to take full advantage of hardware that would otherwise be regarded as obsolete or which would be under-used. Scalability is guaranteed through the various free software applications (ones that normally perform critical functions in a solution) equipped to grow with the problem we are dealing with (all without having to pay for new licences, extend existing ones, upgrade professional versions to enterprise editions, etc).
- Absolute flexibility and freedom. The use of free software tools helps to reduce the time between the conception of a project and its launch because software components – and their source code – that have already been developed can be reused (with no purchase fees). Hence, this code can be adapted to the specific needs of individual clients. The ability to modify existing tools also means that they can be adapted for different countries and cultures.
- Savings in purchase fees and maintenance. These savings are made because there are no purchase fees for the various software tools used (libraries, database engines, content managers, web servers, etc). Hardware purchase costs are slightly lower and maintenance costs could also be reduced because of increased security, the absence of recurrent licence expenses (which add no real value to the solution), etc.

The following is a list of some of the open source tools that Cometa Technologies regularly uses in the solutions it designs:

- Databases: MySQL (http://www.mysql.com/), PostgreSQL (http://www.postgresql.org/).

- Operating systems: different GNU/Linux distributions, such as RedHat, (http://www.redhat.com/) and Debian (http://www.debian.org/).

- Web servers and applications: Apache (http://httpd.apache.org/), JBoss (http://www.jboss.org/), Tomcat (http://jakarta.apache.org/tomcat/), Cocoon (http://xml.apache.org/cocoon/).

- Office productivity: OpenOffice (http://www.openoffice.org/), DocBook (http://www.docbook.org/).

Use of standards

As with the use of free software applications, intensive use of standards can help improve the quality of the solutions offered and meet the challenges of Cometa Technologies' clients.

Free software and standards are clearly complementary. We understand *standards* in the broadest sense of the word to mean:

- de jure standards. Standards approved by an organisation set up for their formal definition, such as ISO (http://www.iso.org/).

- Open standards. Standards approved by consensus and public acts for use by any company, such as those defined by OASIS, Organization for the Advancement of Structured Information Standards (http://www.oasis-open.org/).

- de facto standards. Standards adopted as a result of their extended use and general acceptance, such as the PDF format created by Adobe (http://www.adobe.com).

- Proprietary standards. Standards controlled by a particular company that are not made public for use by other companies, such as some Microsoft file formats (http://www.microsoft.com).

As far as possible, free software projects will generally use existing standards because they coincide with many of its aims, such as reuse and interoperability. Indeed, it is increasingly common to come across organisations that create standards developing reference implementations for these, which they release under one of the existing free software licences.

By using standards in the fields in which it implements solutions, Cometa Technologies can:

- Make the most of the experience and know-how of others in their respective fields of business or technology. This results in lower project costs because better solutions are reached in less time.

- Offer interoperability and ease of integration with other systems. The use of standards allows improved and easier connection of new systems to existing ones through the use of standards common to both.

- Reuse existing tools or create new ones that can be reused in the future, with the associated cost savings. Many standards have reference implementations or associated free software projects, meaning that when we use them, not only do we benefit from their inherent advantages, but we also obtain access to a series of tools that have already been developed and tested.

- Maximise the longevity and growth capacity of developments. By using standards, we are encouraging the future evolution of solutions because they are supported by a series of elements (specifications, free and open source implementations, etc.) that improve by themselves, allowing integration with new elements that did not exist or were unknown at the time.

The technology standards regularly used by Cometa Technologies include:

- Presentation of information: XHTML (http://www.w3.org/TR/xhtml1/), PDF (http://www.adobe.com/), SVG (http://www.w3.org/Graphics/SVG/), VoiceXML (http://www.voicexml.org/), WML (http://www.openmobilealliance.org/).

- XML and web services: XML (http://www.w3c.org/XML/), XSL (http://www.w3.org/Style/XSL/), SOAP (http://www.w3.org/2000/xp/Group/), WSDL (http://www.w3.org/2002/ws/), UDDI (http://www.w3.org/2002/ws/).

- Programming languages: Java (http://java.sun.com/), Perl (http://www.perl.com/), PHP (http://www.php.net/), JavaScript (http://www.ecma-international.org/), C (http://std.dkuug.dk/JTC1/SC22/WG14/), C++ (http://std.dkuug.dk/JTC1/SC22/WG21/).

- Methodologies and support tools: Métrica 3 (http://www.csi.map.es/csi/metrica3/index.html), UML (http://www.uml.org/), Extreme programming (http://www.extremeprogramming.org/).

4.4. Solutions offered by Cometa Technologies

The solutions offered by Cometa Technologies can be grouped into the following broad areas:

- Tailored project development based on the use of free software programs and standards.
- Integration of existing free software applications into new or existing projects.
- Consulting on the use of different technologies, standards and free software tools.
- Training in different technologies, standards and free software tools.

Note

The following sections will describe these areas in more detail, analysing the advantages that they can offer to clients using the example of a real case for each.

The services offered by Cometa Technologies relate to the company's knowledge of information technologies and the specific development (in the broadest sense of the word) of projects in the framework of free software tools and standards. Hence, its services are merely different ways of utilising this knowledge, i.e. of maximising its use of the skills that constitute competitive advantages for the company.

This knowledge has become even more valuable in the light of the current use of free software solutions in companies, which are facing the following problems:

- Lack of support, whether because they lack qualified staff or because there are no stable organisations that can provide support to projects. Cometa Technologies offers the possibility of outsourcing projects and carrying out their subsequent maintenance (hence assuming responsibility for them) or of passing on the necessary know-how so that companies can carry out this task internally. Besides its training and consulting services, Cometa Technologies affords help to companies for selecting and discovering which free software projects will obtain greater success now and in the future, in order to minimise the risks involved in their adoption.

- Immature business models. The projects carried out by Cometa Technologies are guaranteed by the experience it has accumulated over the years in information technologies, specifically in free software, for which it can cite many success stories.

- Lack of applications or immature applications, supported by de *facto standards*. With its consulting service (whether in isolation or as an integral part of a large-scale project), Cometa Technologies helps its clients to find and use the most mature and stable free software applications, complementing them where necessary – hence contributing to the

evolution of the tool through its development and integration services – and taking responsibility for its correct operation.

- A lack of organisations that can systematise free software diffusion and training. Cometa Technologies offers a training service tailored to client needs that provides specific information on given areas (mainly standards and free software tools). In addition, the UOC offers the possibility of obtaining a generic but very comprehensive background in free software with its International Master's Degree in Free Software.

4.4.1. Development of tailored projects

The tailored development services offered by Cometa Technologies consist of the design and implementation of solutions adapted to the specific needs of its clients, with the most suitable technologies for meeting these needs and an intensive use of free software tools and standards.

The main tasks carried out by Cometa Technologies for its tailored projects are:

- Project management: Cometa Technologies plans, controls and monitors the project outsourced by the client.

- Feasibility study: Cometa Technologies carries out a study prior to the project launch to confirm its financial and technical feasibility, among other aspects.

- Analysis: in conjunction with the client, Cometa Technologies conducts a functional analysis of the project, i.e. it specifies the business requirements that need to be met.

- Design: Cometa Technologies obtains the models and specifications defining the technology project based on the functional analysis performed with the client. The various free software solutions to be used and standards to be followed will be introduced at this stage.

- Construction: Cometa Technologies takes care of the technical development of the project, for which it uses free software tools such as frameworks, libraries, services, etc. Each of the modules required to meet the business requirements are developed and the necessary unit and integration tests are conducted.

- Implementation: together with the client, Cometa Technologies takes the developed systems to the production stage, carrying out any migrations and the appropriate implementation tests. It is during this phase that training is carried out for the end users and technicians of the client

company who will be using or carrying out maintenance of the developed solution.

- Maintenance: once the developed project has been accepted, Cometa Technologies takes over its maintenance, which may be corrective or upgrade, depending on the level of service agreed with the client.

To carry out these tasks, Cometa Technologies uses free software programs, either with the express knowledge of the client when these tools form part of the solution provided (such as use of a database management system like MySQL or web service programming libraries like Apache Axis), or without it, reporting only the results obtained (such as project planning using Planner or writing up of project documentation with DocBook).

Website

Apache Axis: http://ws.apache.org/axis

Planner: http://live.gnome.org/Planner

Once the system developed to meet the needs of the client has been accepted, Cometa Technologies provides the client with the full source code for the solution. This code can then have one of the following destinations:

- Release via the projects of the free software tools used in the solution. Cometa Technologies maintains a strong commitment to free software so any development produced for a client will adhere strictly to the licences for the tools used. The changes made to the free software tools used in the projects (bug fixing, new features, etc.) are released under the corresponding licence and sent for incorporation into the aforementioned tools (with the corresponding benefits for the client of the solution).

- Non-release, if the client so wishes. The source code developed specifically by Cometa Technologies for a given solution, which is not subject to compulsory release under any licence, will remain in the hands of the client. The client is thus the proprietor of the code and may decide not to release it for a variety of reasons.

- Release, if the client so wishes. Cometa Technologies always encourages its clients to release the source code of the developed projects that they own exclusively, explaining the advantages of the free software development model over proprietary alternatives.

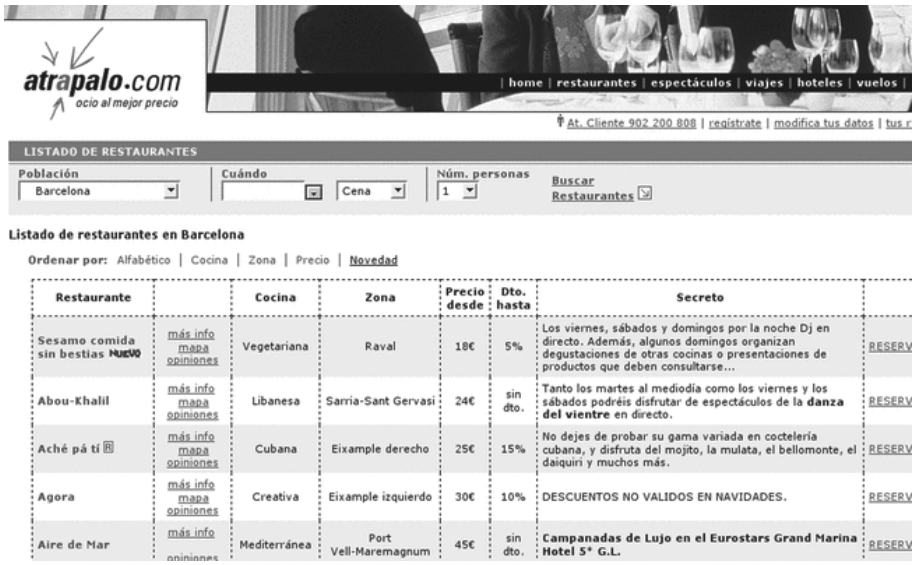
Atrapalo.com

The private sector company Atrapalo.com (http://www.atrapalo.com) bases its business on e-commerce with the promotion and placement of surpluses from the leisure sector (flights, hotels, car rental, shows, etc.), transforming them into offers on the Internet.

Cometa Technologies developed the restaurant table booking system of the Atrapalo.com website using the site's technology platform and diverse free software tools.

The table booking system was approached from the outset as a tailored project, given that it had to be fully integrated into the Atrapalo.com system, using the existing technological infrastructure and maintaining its technical approach. The end result was a series of applications based on the use of open source tools (including PHP and MySQL) that met the needs of the client.

Figure 49.



4.4.2. Integration of tools

The tool integration services offered by Cometa Technologies use existing free software solutions that are parameterised to adapt them to the features of client businesses. The basic tasks carried out by Cometa Technologies in its tool integration projects are the same as those for a tailored development.

The integration of free software tools, in contrast to tailored development services, is not limited to the use of libraries, *middleware*, and so on, as it uses projects that can cover entire functional areas simply by parameterisation or with very specific changes.

> Cometa Technologies offers complete solutions based on the use, parameterisation and customisation of open source content managers, such as ezPublish (http://www.ez.no/) and monitoring tools, as Nagios (http://www.nagios.org/).

By starting out with mature final solutions that have been tested by teams the world over and whose source code is available under a licence allowing its modification at no extra cost, Cometa Technologies can offer solutions tailored to the needs of its clients while minimising the time and cost of development.

BJC Diálogo

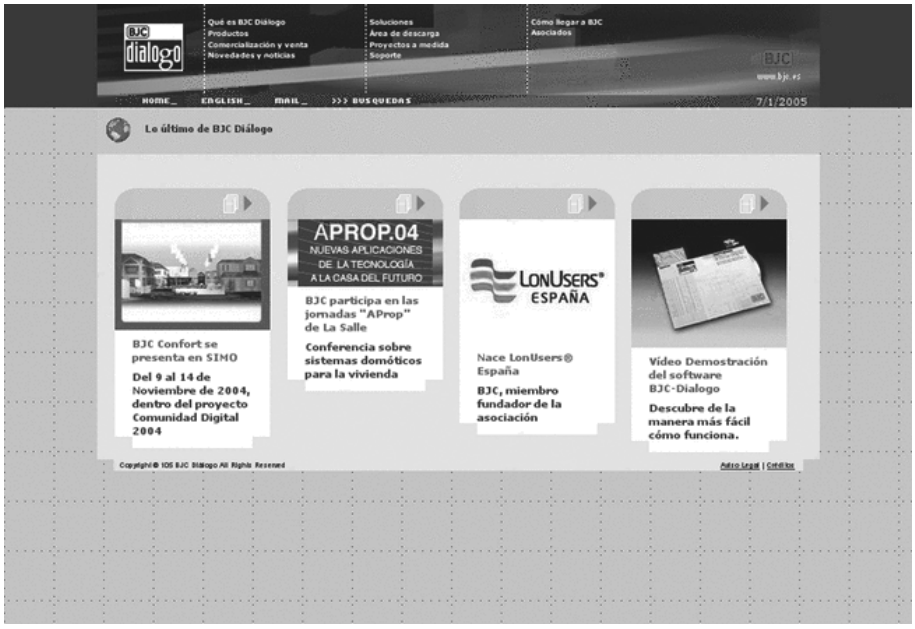
BJC Diálogo (http://www.bjc-dialogo.com/) is a subsidiary of BJC that markets intelligent technical home management systems to automate and manage power in small or medium-sized homes and to control the facilities on sales premises and in hotels and offices.

Together with the Grasa Studio company (http://www.grasa.net/), Cometa Technologies developed the corporate website of BJC Diálogo, where users can request information on tailored projects, find out news and discover innovations, access the company product brochure, read about domotics, etc.

The integration services offered by Cometa Technologies were the perfect partner for the BJC Diálogo corporate website, since the client's needs could largely be met by the use of free software applications developed essentially for content management.

Based on the graphic design provided by Grasa Studio, a series of templates were made; these templates were used with ezPublish, a free content manager. ezPublish needed some modifications and the addition of new modules to fully meet the specifications agreed with the client but a complete solution was provided in less time than would have been needed if development had been carried out from scratch, which also reduced the financial cost of the project.

Figure 50.



4.4.3. Consulting

The consulting service offered to clients by Cometa Technologies consists of providing professional support in decision-making processes within its specialist areas.

Note

Cometa Technologies mainly provides advice on the selection, use and adaptation of free software solutions and standards, and on certain technologies in which it has expert knowledge.

Through its consulting service, Cometa Technologies furthers the knowledge of its clients in specific areas allowing them to take more informed decisions and make the most of certain technologies, tools and standards, which they can use to their full advantage in their business.

Example

Cometa Technologies has helped clients in areas such as security in environments that make intensive use of free software solutions, particularly GNU/LINUX, *e-learning standards* such as ADL SCORM (http://www.adlnet.org/), security improvements and

optimised performance of specific tools such as Apache web server or the Tomcat applications server.

Contex Scanning Technology A/S

Contex Scanning Technology A/S (http://www.contex.com/) is a multinational with head offices in Denmark. The company produces and markets plotters and scanners for professionals.

Cometa Technologies provides technological security consulting services to Contex Scanning Technology A/S and audits some of its products in this field.

Cometa Technologies conducts two types of security audit for Contex Scanning Technology A/S: monthly, designed to check the security of the system, taking into account its various points (operating system, services offered, etc.), and one-off, of a wider scope, before *release* of the audited product (generally every nine months).

The audits generate documents and sets of tests as output that inform the client of possible security issues with its products, if applicable: how a potential attacker might take advantage of them and what risks they would pose. The audits require the performance of security tests for the various free software tools used by Contex Scanning Technology A/S in its products and the use of free software applications used to verify system security, such as Nessus (http://www.nessus.org/).

4.4.4. Training

With its training service, Cometa Technologies transfers its knowledge of specific technological areas (free software tools, standards, etc.) to the staff of its clients.

The purpose of this training is to equip staff at the client company with the skills for understanding and using certain technologies, free software tools and standards that could generate competitive advantages.

Cometa Technologies provides systematic and rigorous training based on the programme agreed with the client and tailored to the specific needs of the latter. In addition to the knowledge gained by the company staff who participated in the training, this service also includes a complete set of documentation describing the issues dealt, which can be used as reference at a later date. Where the client gives its express consent – always with the encouragement of Cometa Technologies – this documentation is released under an open licence, such as GFDL, GNU Free Documentation License (http://www.gnu.org/copyleft/fdl.html) or Creative Commons (http://creativecommons.org/).

Universitat Oberta de Catalunya (UOC)

The Universitat Oberta de Catalunya (Open University of Catalonia) is an institution that offers ongoing training to individuals. The subjects offered by the UOC include the International Master's Degree in Free Software, one of whose courses includes this material.

Cometa Technologies assists the UOC and the education of its students through its consulting services and by auditing the materials of various courses that form part of the university's International Master's Degree in Free Software.

The task covers the design of materials for courses that make up the International Master's Degree in Free Software, which include: *Databases*., *Advanced software development concepts*. These are released with the UOC's consent under the GNU Free Documentation License. Some of the staff of Cometa Technologies also act as consultants on a number of courses for the degree.

4.5. Use of free software within Cometa Technologies

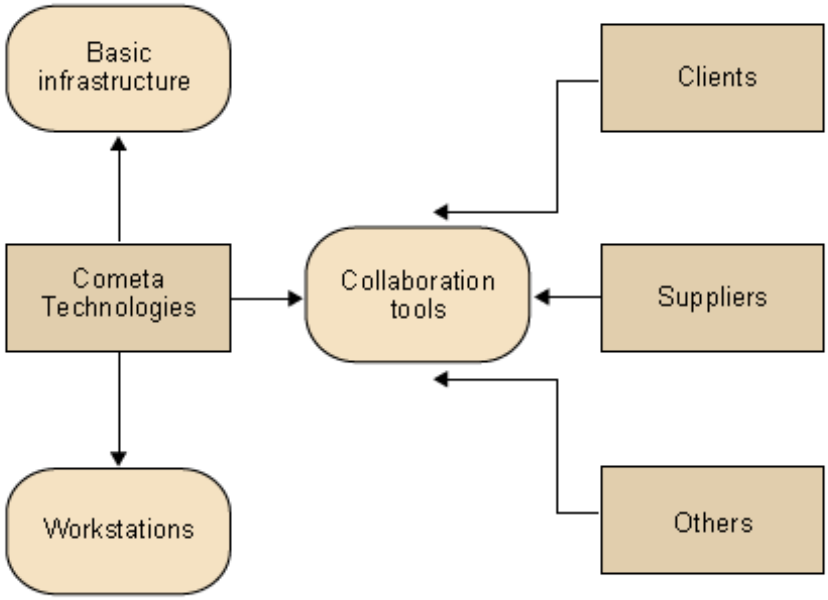
Given Cometa Technologies' commitment to free software and all that it entails, as one might imagine, the company's entire computer infrastructure is based on this type of solution.

The advantages of the use of free software applications for Cometa Technologies over proprietary alternatives are exactly the same as the ones it offers to its clients: greater flexibility, increased security, enhanced performance, cost savings, compliance with standards, etc.

The IT infrastructure of Cometa Technologies can be divided into the following broad areas:

- Basic infrastructure: tools for the everyday, basic operation of the company.
- Collaboration: tools for communication between company staff and between the latter and others (clients, for instance).
- Workstations: tools on the computers used by those who work at the company.

Figure 51.



The following sections will describe each of these areas in detail and analyse the free software tools used in each case.

4.5.1. Basic infrastructure

The basic IT infrastructure of Cometa Technologies allows it to carry out the basic operations necessary for the everyday running of the company, which include correct connection of the computer network to the Internet, sharing of files stored on central servers (which require backups to be made), etc.

The table below summarises the main software components used in the basic IT infrastructure of Cometa Technologies:

Type of application	Solution used
Operating system	GNU/Linux
Print server	Common UNIX Printing System
Routing	route (GNU/Linux OS kernel)
Firewall	iptables (GNU/Linux OS kernel)
DNS server	ISC BIND
DHCP server	ISC DHCP Server
Proxy cache	Squid
File server	Samba
Backups	Amanda

Type of application	Solution used
Directory service	OpenLDAP

4.5.2. Collaboration

The collaboration aspect of the IT infrastructure of Cometa Technologies uses tools for setting up channels of communication between company staff or between company staff and external agents (clients, suppliers, etc). These include the usual applications for Internet communication such as e-mail, as well as more specific applications to cover the particular needs of Cometa Technologies, perhaps not as common in other types of company.

Project monitoring tool

This application has a web interface and is used to manage aspects of the projects of Cometa Technologies' clients, including functional requirements, detected bugs, related documents, etc. Each user of this tool is assigned a profile (project manager, analyst, programmer, client, etc.) allowing him or her to carry out a series of actions in the areas mentioned. This tool is used to conduct meticulous project status controls and to keep clients informed of the status of their project at all times.

The table below summarises the main software components used in this area:

Type of application	Solution used
E-mail server	Postfix, IMAP
Fax server	HylaFAX
Project monitoring tool	Mantis
Web server	Apache
Database management system	MySQL
Content management	ezPublish
Version control system	Subversion

4.5.3. Workstations

Workstations are the computers, both laptops and desktop computers, used by the staff of Cometa Technologies. These individuals are assigned different profiles depending on the tasks that they perform (project management, marketing, technical development, etc.), but all of the workstations have the same basic setup, which only varies in the applications that they normally use.

Since all of the applications are free software and there are no licence fees payable for each workstation, user, etc., all applications are installed on all computers for convenience (any user can use any company computer and have access to all of the applications he or she needs).

The table below summarises the main software components installed on a basic workstation at Cometa Technologies:

Type of application	Solution used
Operating system	GNU/Linux
Office automation suite	OpenOffice
E-mail client	Thunderbird
Web browser	Firefox
Image editor	Gimp
Audio and video player	Kaffeine
Instant messaging client	Kopete
Personal information manager	Evolution
CD and DVD burning tool	K3b
Integrated development environment	Eclipse
Version control system	Subversion
Project planning	Planner
Diagram creation tool	Dia, Umbrello

4.6. Case studies

4.6.1. Tailored project

The following section describes the tailored project carried out by Cometa Technologies for the Derecho.com legal site (http://www.derecho.com/).

Presentation of the client

Derecho.com is a company providing quality legal services and content over the Internet to allow its clients to solve their legal issues, offering the possibility of consulting databases, asking questions about legal issues, looking up legislation and purchasing contract templates and case forms, among other services.

Presentation of the project

Derecho.com is a vertical legal website. Since its creation, Derecho.com has entrusted Cometa Technologies with the development and maintenance of the technology platform on which its business is based.

Figure 52.

Project aims

When the site was set up in 1999, the following aims were identified as essential:

- Ability to display the information managed on the site on different platforms (PCs, mobile telephones, PDAs, etc).
- Possibility of looking up content in different views for their transparent integration into third-party websites (*co-branding*).
- High availability: the platform had to be operative all day, every day.
- High performance: the platform had to support the concurrent access of hundreds of users and maintain reasonable response times.
- Minimal development and maintenance costs.

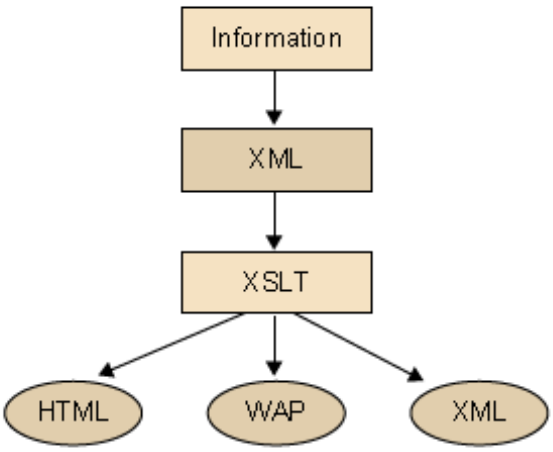
Figure 53.

Description of the solution

In order to meet the chief aims for content management, the decision was made to base the entire solution on XML for representing the information.

The contents are extracted from a database and expressed in XML. By transforming it using style sheets (XSL), the content is represented in diverse languages (XHTML, WML, XML, etc.), allowing it to be displayed on a range of platforms.

Figure 54.



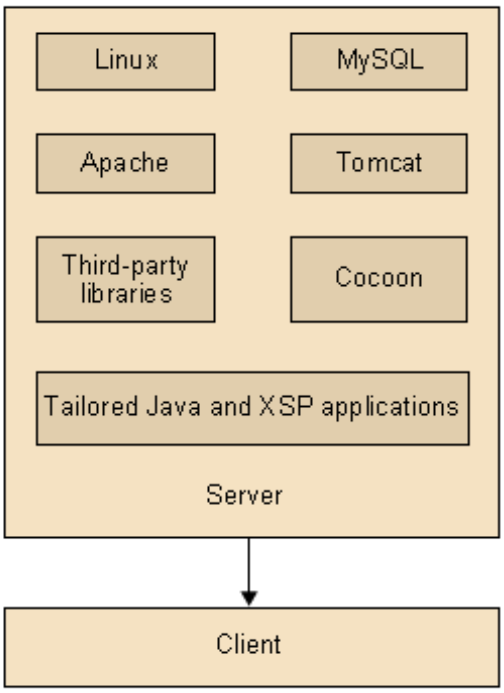
This process also adapts the graphic representation of the information to its destination. For example, a web page might be displayed differently depending on whether it is aimed at users of the Derecho.com site or users who consult it through other sites with which Derecho.com has set up agreements.

Tools used

To maximise the stability and performance of the platform and to save on development and maintenance costs, it was decided to use free software solutions, which included:

- GNU/Linux: operating system of the diverse servers housing the solution.
- MySQL: relational database management system storing all of the information.
- Apache and Tomcat: web server and servlet container on which the various applications are run.
- Cocoon: *framework* that facilitates and optimises the job of working with XML documents and XSLT operations.
- Third-party libraries.

Figure 55.



4.6.2. Integration

The following sections describe the integration project carried out by Cometa Technologies and 0 y 1 Factoría for I+C2 magazine (http://www.fsjd.org/cat/ic2.php), published by the Sant Joan de Déu Foundation (http://www.fsjd.org/).

Presentation of the client

The Sant Joan de Déu (FSJD) Foundation is a non-profit organisation created by the monastic order of San Juan de Dios (Province of San Rafael) to set up an area of action for the convergence of synergies in teaching and research

between the diverse centres and professionals that form part of the Order, particularly in Catalonia but without excluding those from other parts of Spain or even abroad.

The professional services firm 0 y 1 Factoría with which Cometa Technologies developed the project specialises in detecting innovations, stimulating creativity and coming up with distinctive digital spaces.

Presentation of the project

The online I+C2 (Investigación, Innovación, Ciencia y Compromiso or Research, Innovation, Science and Commitment) newsletter is an initiative of the Sant Joan de Déu Foundation designed to stimulate awareness and raise the profile of different innovative initiatives being carried out in research, teaching and social development within this institution.

Cometa Technologies was given the task of assisting the Sant Joan de Déu Foundation in the development and maintenance of the technology platform on which publication of the I+C2 magazine is based, working in collaboration with 0 y 1 Factoría, which was responsible for the creation, editing and publication of the newsletter contents.

Figure 56.

Project aims

The aims of the project were defined by the end client, the Sant Joan de Déu Foundation, and by the two companies that came up with the final solution, 0 y 1 Factoría and Cometa Technologies. They were:

- To create a technology platform with basic functionality for the management of an online magazine, including regular e-mails containing online news, the creation of editions grouping these news items, etc.

- To incorporate certain features that would add value to the newsletter, making the most of its use of the Internet as the channel of publication, such as search engines, customised subscriptions, multiple language support, etc.

- To allow easy and intuitive maintenance of the information generated by 0 y 1 Factoría or by end client of the solution, the Sant Joan de Déu Foundation.

- To prepare the solution for upgrade maintenance in a way that would allow the easy addition of new functional features and add value to existing ones.

- To minimise the financial cost of the solution and the time from the initial proposal to production.

Figure 57.

Description of the solution

To meet the required aims, it was decided to base the technological solution on the use of a content manager with a free software licence. This would give the team a large base to work with while minimising financial costs and the turnaround time for having the I+C2 newsletter site fully up and running.

The chosen content manager was version 3.3 of ezPublish, the most recent version available when the project was launched. The use of ezPublish necessitated the following tasks:

- Installation of the ezPublish content manager, integrating it with other elements such as the database engine, web server, etc.

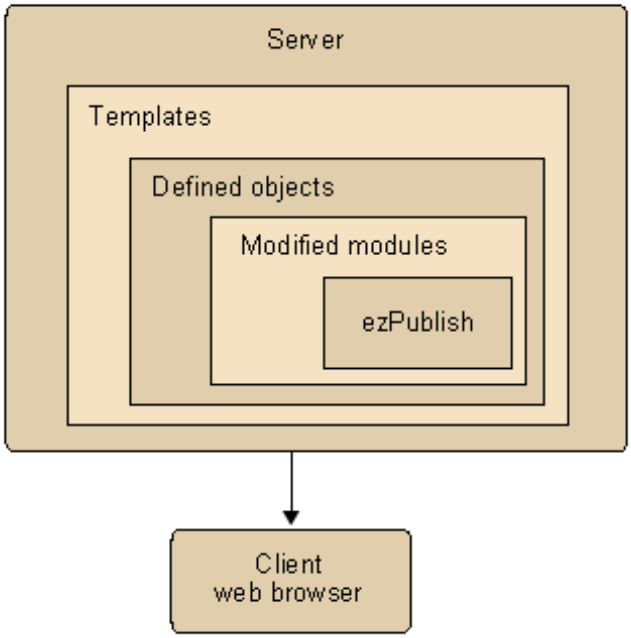
- Creation of the objects for housing the contents of the website, such as article, edition, news item, subscriber, etc., indicating their attributes and views and taking into account multiple language support.

- Creation and modification of a series of templates used by the content manager to generate the HTML needed to publish the contents of the website; in this case, the I+C2 magazine.

- Creation and modification of a series of additional templates that the content manager uses for website content services, such as the search engine, important news, etc.

- Modification of certain content manager modules to add features that did not originally exist and are typical of an online magazine (grouping of articles in editions, management of user subscriptions and personal data, regular e-mailing of the newsletter, etc).

Figure 58.



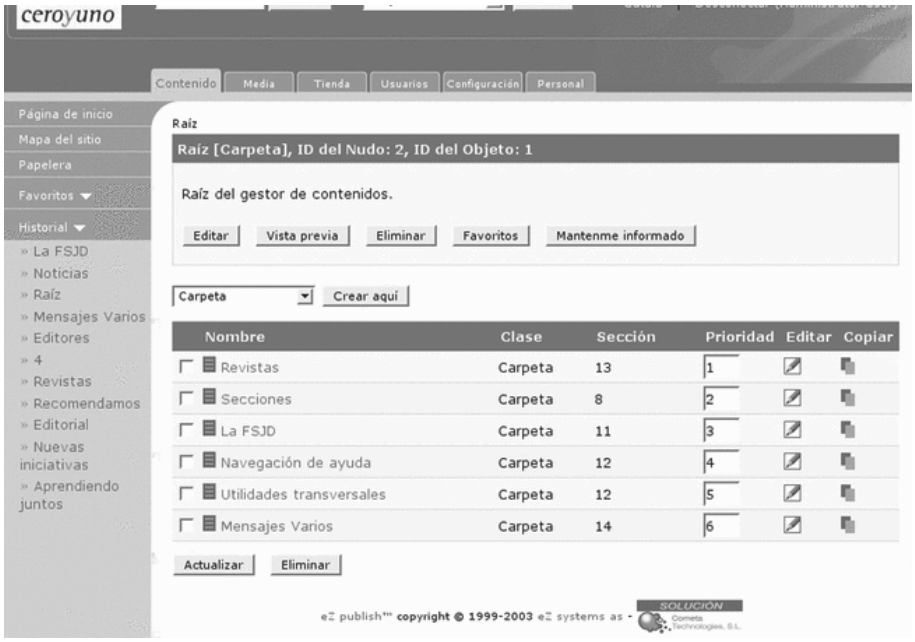
Tools used

As explained above, the solution developed was based on the use and modification of ezPublish as a content manager of the I+C2 newsletter site. This application and those created specifically for the project were based on other free software applications, generating a stack of coordinated software that integrated a platform with the characteristics mentioned earlier in this module (reliability, security, low cost, etc).

Some of the tools used were:

- GNU/Linux: operating system of the server housing the solution.
- MySQL: relational database management system for storing all of the information used with the content manager.
- PHP: interpreter for the *script* language used by the content manager and the modifications that Cometa Technologies made to it.
- Apache: web server.
- ezPublish: content manager using the GPL licence.
- Third-party libraries.

Figure 59.



4.7. Summary

This chapter has looked at the case of Cometa Technologies, a private sector company that provides information technology solutions based on free software tools and standards.

Cometa Technologies offers its clients services in areas in which the company has an excellent knowledge:

- Development and integration of technology solutions based on the use of free software tools and standards.
- Training and consulting in specific technologies, free software tools and standards.

With these services, Cometa Technologies attempts to tackle the challenges of information technology projects faced by organisations today:

- Reducing the time to project launch from its conception.
- Improving integration with the company's other systems.
- Increasing system security.
- Maintaining independence from suppliers.
- Improving the use of technology solutions while taking into account different countries and cultures.
- Cutting the final costs of solution ownership.

The use of free software solutions is an important step in this direction because they offer the following advantages:

- Reliability and security.
- Enhanced performance and scalability.
- Flexibility and freedom.
- Savings on purchase and maintenance costs.

We have looked at Cometa Technologies as an example of a company that bases its entire computing infrastructure on the use of free software solutions. With the same aim, we described two case studies of the application of this type of solution (Derecho.com and the Sant Joan de Déu Foundation), which used the tailored project development and integration services, respectively.

Summary

The growing commitment of government authorities and major organisations to free software has worked wonders to improve its visibility and how it is perceived, proof of which can often be seen in the media.

Moreover, many experts and analysts consider free software to have the potential to drive economic development worldwide and in Europe in particular. For example, Gartner has said: "Free software is a catalyst that will restructure the industry, producing higher quality software at a lower cost".

By presenting a series of implementation cases, this module attempts to confirm the validity of the free software model as an alternative to proprietary software in diverse regions, contexts and economic sectors.

The impetus given by the Junta of Extremadura to free software in diverse areas of the public sector, particularly in education (with the LinEx distribution as its standard), was the first of these cases. Similarly, though on a much larger scale, we looked at the efforts of the Federal Government of Brazil to bridge the digital gap, a policy in which free software has played a crucial role. The cases of free software implementation in Extremadura and Brazil, despite their obvious geographical differences, have many points in common, including their motivation, and have served and continue to serve as an example to other governments and administrative agencies wishing to promote the development of free software in their respective areas of influence.

In all events, the policies introduced in both Extremadura and Brazil reveal a long-term commitment to stimulating the information society on the basis of free software. The existence of a project with mid- to long-term aims is a basic feature that sets apart sound free software initiatives with guarantees of success.

Turning to the private sector, the Sun Microsystems case study shows how a large company with a consolidated position and all that comes with it has managed to uncover new opportunities for technology and business development in free software. Nonetheless, it should be remembered here that the vast majority of free software development projects fail, even with business backing. One of the main reasons for the success of Sun and its projects lies in its ability to define a software development methodology that is compatible with the business and technological strategy of the organisation.

Lastly, Cometa Technologies shows us how free software can form the basis of a valid business model and an alternative to proprietary software, offering considerable growth potential for an SME with a local base, with the benefits of participating in the free software community and the use of free software tools and solutions whose quality has been confirmed across the world.

Bibliography

Acero, F.; Hontoria, L. (2000). *Manual de LinEx (GNU/Linux): Guía de referencia para trabajar con este sistema operativo*. Madrid: EDIT-IN.

Asociación de Usuarios Linux/Unix de Extremadura (GULEX) [online]: http://www.gulex.org

Asociación de Usuarios de LinEx (Sinuh) online]: http://www.sinuh.org/web/index.php

Asociación Regional de Universidades Populares de Extremadura (coord.) (2001). *Alfabetización Tecnológica: Manual de Buenas Prácticas de Alfabetización Tecnológica en Extremadura*. Mérida: Consejería de Educación, Ciencia y Tecnología.

Asociación Regional de Universidades Populares de Extremadura (coord.) (2003). *Metodología de Alfabetización Tecnológica: II Manual de Buenas Prácticas*. Mérida: Consejería de Educación, Ciencia y Tecnología.

Consejería de Educación, Ciencia y Tecnología (2002). **Vivernet: Emprender en la Economía del Conocimiento.** Mérida: Consejería de Educación, Ciencia y Tecnología.

Consejería de Educación, Ciencia y Tecnología (2004). *El viaje virtual de Linextrmix: Todo lo que deseabas saber en cómic sobre gnuLinEx GNU/Linux software libre*. Mérida: Consejería de Educación, Ciencia y Tecnología.

Cosgrove, Lorraine (2002). "Open Source Gains Momentum". *CIO Research Reports*. [online]: http://www2.cio.com/research/surveyreport.cfm?id=48

Danish Board of Technology (2002). *Open source software in e-government*.

Fueyo Díaz, J. et al. *Manual en español de Squeak* [online]: http://www.small-land.org

Fundación Insula Barataria [online]: http://www.ibarataria.org

GNU Operating System-Free Software Foundation [online]: http://www.gnu.org/

Grimaldos Parra, J.J. (2003). *Guadalinex-edu: Manual básico de usuario*. Seville: Consejería de Educación y Ciencia.

Gustafson, Paul; Koff, William (2004). *Open Source: Open for Business*. Computer Sciences Corporation.

Hamel, Gary; Prahalad, C. K. "The Core Competence of the Corporation". *Harvard Business Review*.

Hickman, Amanda (2004). *Choosing and Using Open Source Software: A Primer for Non Profits*. [online]: http://www.nosi.net/node/24

Instituto Nacional de Estadística [online]: http://www.ine.es/

Linextremix [online]: http://www.linextremix.com/

Margulius, Dave (2003). *The real cost of switching to Linux*. [online]: http://www.infoworld.com/t/platforms/real-cost-switching-linux-544

Nuevos Centros del Conocimiento de la comunidad autónoma de Extremadura [online]: http://www.nccextremadura.org

gnulinex.net business portal [online]: http://www.gnulinex.net

www.linex.org freedom portal [online]: http://www.linex.org

Guadalinex project [online]: http://www.guadalinex.org

Guadalinfo project [online]: http://www.guadalinfo.net

LinExCol project (LinEx of Extremadura for Colombia) [online]: http://www.linexcol.org/

Vivernet project [online]: http://www.vivernet.es

Technological Learning Network of the Autonomous Community of Extremadura [online]: http://www.rte-extremadura.org

Robert Frances Group Inc. (2002). Total Cost of Ownership for Linux in the Enterprise.

Department for Planning, Statistics and Financial Analysis of the Autonomous Community of Extremadura [online]: http://www.estadisticaextremadura.com

Small-Land [online]: http://www.small-land.org

Soreon Research (2004). Saving Cash: *A Comparison of Open Source and Proprietary Software*.

Squeak [online]: http://www.squeakland.org

Stoller, Jacob (2004). *Open source: assessing the TCO picture*. [online]: http://www.managementmag.com/index.cfm/ci_id/2230/la_id/1

The Open Group.*Open Source and Standards*.

Wheeler, David A. (2004). *Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!*. [online]: http://www.dwheeler.com/oss_fs_why.html

Ullán de Celis, A.; Redrejo Rodríguez, J.L. (2004). *Linex*. Madrid: Anaya Multimedia.

Further reading

Abella, A. et al (2004). *El libro blanco sobre el software libre en España 2003: la informática libre en nuestro país*. Mérida: Consejería de Educación, Ciencia y Tecnología.

Álvarez Franco, P.M.; Martínez Montero, F. (2004). *Teseo Linex*. Mérida: Consejería de Educación, Ciencia y Tecnología.

Ball Satate University (2002). *Squeakers* [DVD]. [s.l.].

Centro de Fomento de Nuevas Iniciativas (coord.) (2004). *Aprender con gnuLinEx: experiencias docentes curso 2002-2003*. Mérida: Consejería de Educación, Ciencia y Tecnología.

Consejería de Educación, Ciencia y Tecnología (2004). *5 años de Alfabetización Tecnológica en Extremadura: 1999-2004* [DVD]. [s.l]: Consejería de Educación, Ciencia y Tecnología.

Gómez Deck, D.; Redrejo Rodríguez, J.L. "Squeak en España como parte del proyecto LinEx" [online, in *Small-Land's Swiki*) http://swiki.agro.uba.ar/small_land

González Sánchez, J.L.; Gazo Cervero, A. (coord.) (2004). *Sociedad y software libre: Curso Internacional de Verano de la UEX sobre Software Libre*. Mérida: Consejería de Educación, Ciencia y Tecnología.

Guerrero Alonso, D.A.; Ortiz Durán, J.S. (2003). *Camino hacia Itaca*. Mérida: [s.n.].

gnuLinEx v.2.0 manual [online]: http://guia.linex.org/pdf/manual.pdf

WITH SUPPORT FROM THE

Education and Culture DG

Lifelong Learning Programme

THIS COURSE BOOK EXAMINES THE CONCEPTS RELATED TO FREE SOFTWARE BUSINESS AND ALLOWS US TO KNOW AND IDENTIFY CONCEPTS RELATED TO ITS PRODUCTION.

THE STUDENT SHOULD BECOME FAMILIAR WITH AGENCIES AND PROJECTS RELATED TO THE IMPLEMENTATION OF FREE SOFTWARE IN THE PUBLIC SECTOR, PRIVATE SECTOR AND KNOW, IDENTIFY AND UNDERSTAND THE CONSEQUENCES OF THE USE AND EXPLOITATION OF FREE SOFTWARE IN DIFFERENT AREAS OF IMPLEMENTATION.

THIS BOOK ALSO STUDIES HOW TO APPROACH AND IMPLEMENT PROJECTS BASED ON FREE SOFTWARE, PROJECTS TO MIGRATE, PRODUCTION AND DEVELOPMENT METHODOLOGIES AND BUSINESS PROJECTS RELATED WITH FREE SOFTWARE

freeknowledgeinstitute

UOC Universitat Oberta de Catalunya www.uoc.edu

www.ou.nl

UNIVERSITETET I AGDER