UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by Novática <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by Novática

UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <novatica@ati.es>

UPGRADE Newslist available at
<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>

ISSN 1684-5285

Monograph of next issue (August 2009)

"20 years of CEPIS: Informatics in Europe today and tomorrow"

(The full schedule of **UP**GRADE is available at our website)

CEPIS UPGRADE
The European Journal for the Informatics Professional
http://www.upgrade-cepis.org

Vol. X, issue No. 3, June 2009

Monograph: Libre Software for Enterprises
(published jointly with Novática*)
Guest Editors: *Jesús-M. González-Barahona, Teófilo Romera-Otero, and Björn Lundell*

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Best Practices for FLOSS Adoption

Carlo Daffara

Many companies and public organisations are considering free software (Free/Libre and Open Source Software, FLOSS) adoption or migration, since they potentially give lower costs for implementing new systems or maintaining current Information Technologies (IT) infrastructures. The migration or adoption processes are however always complex multidisciplinary efforts that touch several areas, requiring a complete understanding of the composition and execution of individual workflows and of how people interact with IT systems. This paper will propose a set of guidelines for maximizing the success of a FLOSS adoption or migration in a company, covering three groups: management guidelines (oriented to upper management), technical guidelines (oriented to a lower, technical oriented level) and community guidelines (oriented to several features of Open Source Software (OSS) which companies may want to take into consideration, such as the relationship with the OSS community, or to improve internal support through internal experimentation and training).

Keywords: Companies, FLOSS Adoption, Libre Software, Migration, OSS Communities.

1 Introduction

Within a company, the decision to start adopting OSS solutions or to integrate OSS into the company service offering is usually based on one of the following business cases:

- Basic substitution/migration: the use of FLOSS in the IT infrastructure, frequently in substitution of proprietary software.
- New deployment: the introduction of FLOSS for a new project internal to the company (adoption).
- Selling services based on FLOSS.
- Selling products that contain FLOSS as a significant component.

In this sense, a company may find FLOSS useful from a tactical point of view (FLOSS is cheaper to implement, with fewer constraints than from a traditional vendor, or may help in introducing products with a reduced time to market) or from a strategic point of view (creation of new markets, adoption of different business models). Internal adoption processes are commonly modelled using an incremental model, originally developed by Carbone et al [1], that models adoption as a process of successive steps, where the first one (initial adoption) is usually the most complex one (see Figure 1).

In fact, migration and adoption process are complex, with multidisciplinary efforts that touch many different areas and require a complete understanding of how individual workflows are composed and executed as well as understanding how people interact with IT systems in their daily work. In this sense, a FLOSS migration is a major endeavour and, as with most complex undertakings, can easily go wrong. There are several obstacles to the execution of a migration, and some of these can be avoided easily by using simple procedures. Most of the difficulties are not really technical in nature, but are rather organizational ones, and will require the most effort from upper management.

Author

Carlo Daffara is the Italian member of the European Working Group on libre software, and chairs several other working groups like the Open Source Middleware Group of the IEEE technical committee on scalable computing and the Small and Medium Enterprises (SME) Working Group of the EU Competitiveness Task Force. His current research activity is centred on the sustainability of OSS-based business models. He is currently the head of Research and Development (R&D) at Conecta Research Ltd, an open source consulting company <http://www.conecta.it/>. <cdaffara@conecta.it>.

Another important aspect is the social impact of the migration (like user acceptance) which may require special attention.

2 Management Guidelines

The main drive for a successful migration to FLOSS always starts with a clear assessment of the IT landscape, a clear vision of the needs and benefits of the transition and continual support. The differences in open source development models and support (compared to traditional vendors) may require a significant change in the way software and services are accounted for and procured, and in general a shift of responsibility from outside contractors to in-house personnel.

Be sure of management commitment to the transition

Management support and commitment have repeatedly been found to be one of the most important factors in the success of complex IT efforts, and FLOSS migrations are no exception. This commitment must be guaranteed for a time period sufficient to cover the complete migration. This means that in organizations where IT directors are frequently changed, or where management changes in fixed periods of times (for example, in public organisations where changes happens frequently) there must be a process in place to hand over management of the migration. The commitment should

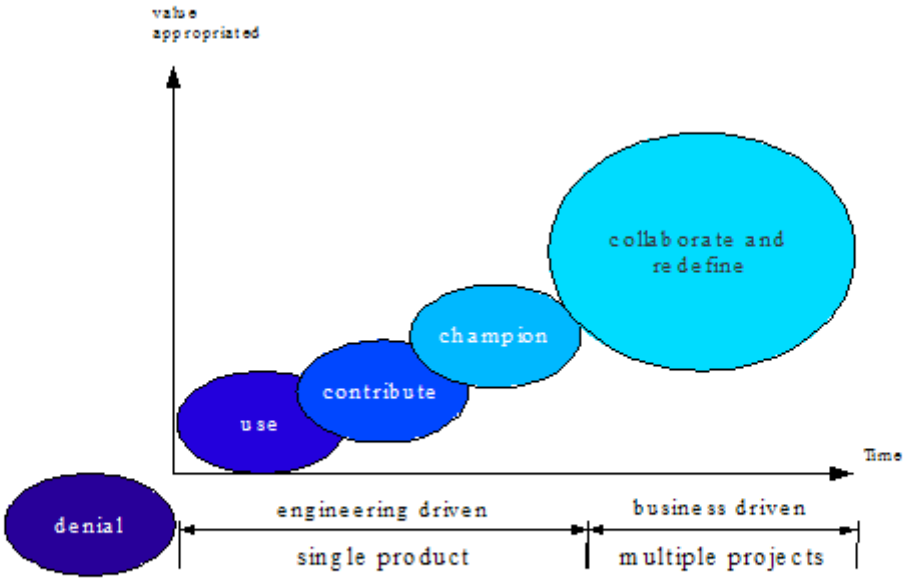

Figure 1: Successive Steps Model for FLOSS Adoption.

also extend to resourcing (as transitions and training will require both funding and in-house personnel). The best way to ensure continued coordination is to appoint a team of mixed roles (management and technical) to provide continuous feedback and day-to-day management.

Troubleshooting point: if the only people working on planning the migration are from IT, there may be insufficient information being provided to upper management and financial planning in order to continue the migration after the initial step.

Prepare a clear vision of what is expected from the migration or adoption, including measurable benchmarks

The transition can be started for several reasons, including better control of IT costs, greater flexibility, independence from suppliers or support of open data standards. To be sure that the migration is effectively producing benefits or is going according to the migration plan, it is fundamental to know beforehand what indicators will be used to evaluate progress.

These requirements must be realistic; in particular expectations of total cost of ownership (TCO) reductions must be compared with publicly available data.

Troubleshooting point: if the only perceived advantage is that "the software comes from the net for free", there may be a set of false assumptions that will probably lead to a negative assessment of the migration in the end.

Make sure that the timetable is realistic

The introduction of a new IT platform will always require a significant amount of time. As a rule of thumb the time to perform a full transition to FLOSS may be considered to be comparable to that of the introduction of a new company-wide enterprise resource planning (ERP) application; for smaller transitions, time and effort should be scaled accordingly.

Troubleshooting point: when migration time is measured in days, and no post-migration effort is planned, the process may be forced to a stop after the planned resources are expended.

Review the current software/IT procurement and development procedure

As implementation effort is shifted from commercial to open source software, the procurement and development process needs to be updated accordingly. In particular, the focus may change from acquisition to services, as less software is bought "shrink-wrapped" (commercially bought), and this change may require changes in how the internal IT budget is allocated.

Internally developed software will require a porting or a rolling transition to new software that is either multi-platform or accessible using standard interfaces (for example, Web applications), and this should be taken into account in the overall IT plan.

Troubleshooting point: When no change of procurement and development is planned, this may indicate that managers have not understood the scope of changed required for the adoption of FLOSS.

Seek out advice or search for information on similar transitions

As the number of companies and organisations that have already performed a migration is now considerable, it is easy to find information on what to expect and how to proceed. In this sense, the COSPA project has developed an online knowledge base that is accessible through the main COSPA site [2]. Public organisations can also contact their local Open Source Competence centre, which will provide information and support in the migration process.

Avoid "big switch" transition, and favour incremental migrations

Most large scale migrations that are performed in a single, large step[1] (involving the abrupt change from one IT environment to the other) are usually marred by extremely high support and technical costs. While the need to support more than one environment does increase support and management cost, "gentle" or incremental migrations usually bring a better overall experience for the users and result in minimal disruption of business processes.

An example of gentle migration can begin with the migration of server side applications, that are usually standards or network-based and thus easier to replace, leaving desktop and user-facing applications to last. Such a scheme was depicted in [3] (see Figures 2 and 3).

Assign at least one person to interact with the OSS community or the OSS vendor, and try to find online information sources

A significant advantage of OSS is the availability of online free resources, in the form of knowledge bases, mailing lists, wikis (collaborative sites) that may provide a substantial support in many cases comparable to commercial offerings. The biggest problem is the identification of such knowledge sources; in this sense assigning a resource to find, categorize and interact with such sources is a way to reduce the cost of support; a common way to provide a unified source of information is by setting up a small intranet Web page with links to online resources.

Troubleshooting point: Difficulties may arise when no one knows where to find information on the tools that are in use, or when everyone has to search on Web sites on their own for finding usage tips.

3 Technical Guidelines

A significant difference in FLOSS adoptions is the different development model adopted by most open source projects, and the difference in delivery of updates and support. This requires a change in the way adoption and updates are handled, to reduce as much as possible interoperability problems.

Understand the way OSS is developed

Most projects are based on a cooperative development model, with a core set of developers providing most of the code (usually working for a commercial firm) and a large number of non-core contributors. This development model does provide great code quality and a fast development cycle, but also requires significant effort in tracking changes and updates. The adoption of an OSS package should be proposed when:

- The project itself is "alive", i.e. it does have an active development community.
- There is a clear distinction between "stable" and "unstable" software. In many projects, there are two distinct streams of development, one devoted to integrating the latest changes and additions, and another focused on improving stability and bug fixes. Periodically, the developers will "freeze" development to turn the unstable version into the stable one, and create a new development, which becomes the latest version. This distinction allows the developers to satisfy both the users willing to experiment with the latest functionality, and those using the software for day-to-day operations, but requires extra effort in collecting information and new versions.

If new functionality or fixes are necessary, it may be easier to ask for a commercially supported version of the software. In many cases, the commercial vendor will also contribute financially to the open source project.

Troubleshooting point: Be cautious when the IT manager or the developers think that OSS is some kind of commercial software that someone has put for free on the net, and that it "simply works".

Create a complete inventory of software and hardware that will be affected by the migration, and what functionality the company is looking for

There can be no successful migration when the starting point is not a known quantity. Most companies and administrations have no process in place for auditing software and hardware platforms, and are thus unable to quantify the number of tools and software that needs to be replaced or integrated in an OSS migration. The audit process must also take into account the number of concurrent users, average use across the organization, and whether the software uses open or closed communication protocols and data formats. This audit will be the basis for the decision on what users will be migrated first and for taking into account the cost of software re-development or migration to a different data format. Automated software inventory tools are readily available and may reduce the cost of performing the inventory and allow for a stricter control on installed software (thus reducing maintenance costs).

Some of the aspects that should be surveyed are:

- Data formats in use, at the document exchange, database and network protocol level.
- List of applications in use, including those internally developed, macros and active documents.
- Available functionality.
- Shortcomings and problems with the current infrastructure.

It is fundamental that the migrated software can meet the same functional requirements as the current IT infra-

[1] Note from the Editor: This is also known as a "big bang approach" or "big bang adoption" <http://en.wikipedia.org/wiki/Product_Software_Adoption:_Big_Bang_Adoption>.

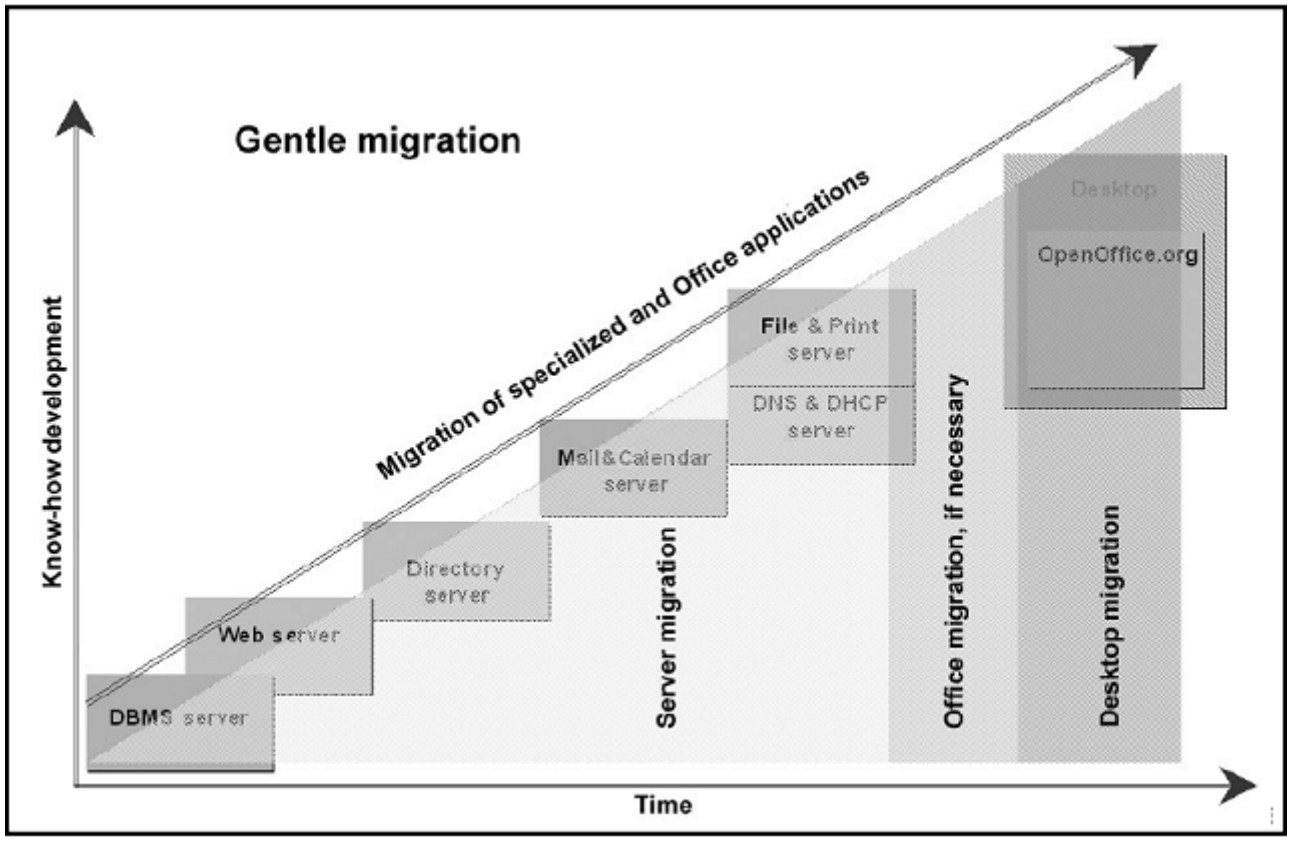

Figure 2: "Soft" Migration Scheme, Beginning with the Migration of Server Side Applications.

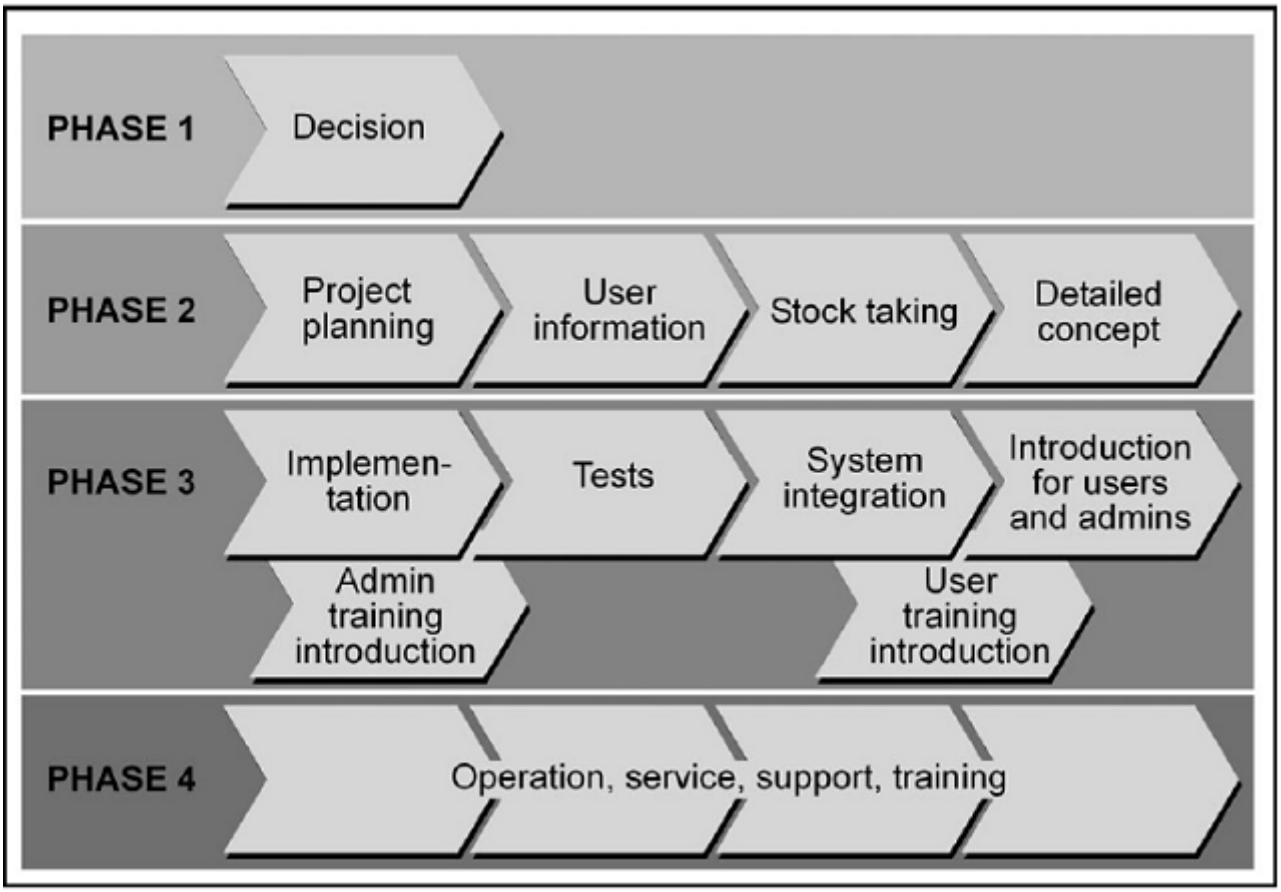

Figure 3: Scheme of a "Soft" Migration.

structure, and usually also improve on that in functional terms or in non-functional measures like availability, reliability or performance.

Use the flexibility of OSS to create local adaptations

The differentiating property of OSS is the flexibility and freedom that it gives to users and developers in creating new versions or adapted versions of any package. This flexibility can greatly enhance the perceived value of OSS, for example it is possible to create customized packages that contain local configurations, special fonts and other supplemental material like preset macros and templates commonly used in the company. Also, custom look and feel may significantly improve user acceptance, both by presenting a nicer looking desktop, and by maintaining common links and menu entries.

These customizations can be integrated in a simple way in the most commonly used Linux distributions, or by creating a local repository of software. Note that in many cases, it is not necessary to produce software or code, as most adaptations involve selecting the appropriate package, changing the graphical appearance, or providing templates and defaults.

There is much more software available than what is installed by default

Licensing or design issues limit substantially the amount of software that is usually included in the default installation of the most commonly used Linux distributions. For example, only a few include playback capability for the most common audio and video formats, due to licensing and patent issues. Some packages that may be of interest to only a minority of users are also not included by default.

For this reason, it is important to research and include additional packages in the default installation that may help in the transition period. Such packages include additional fonts, multimedia tools, and other packages that may be useful in a mixed environment.

In selecting packages, always favour stability over functionality

Among the many potential packages available for every function, there is always a balance between functionality and stability. In general, among the potential candidate packages that satisfy the functional requirements for the migration, preference should be given to the one that is more stable, thus having a longer real-world usage (and thus more information available for the administrator) and least change between different releases.

Troubleshooting point: When the IT administrator wants the latest version of everything on user's desktop it may be difficult to find adequate support or documentation, or some unreported bugs may arise.

Design the workflow support infrastructure to reduce the number of "impedance mismatches"

Every transition from an Information and Communications Technologies (ICT) infrastructure to another leads to some "impedance mismatch", that is to small differences and incompatibilities. For example, this can be observed when translating documents from one data format to another. The overall infrastructure should reduce the number of such transition points, for example by redesigning the document templates in the OpenDocument Text (ODT) open format instead of reusing previously developed versions made using proprietary tools. This reduces greatly the formatting and style differences that arise when one format is translated into another.

Introduce a trouble ticket system

A challenge for every new IT deployment is how to assess user satisfaction and the degree of acceptance of the new solution, especially in medium sized companies when user feedback is more difficult to collect. An online trouble ticket system may provide an easy and simple way to discover weak points in the deployment, and can help to identify users that may need additional training by analyzing the per-user submission statistics. It may also point to weaknesses in the deployment, for example when several trouble tickets are related to the same specific area.

Compile and update a detailed migration workbook

A large scale migration effort requires a coordinated plan, and clear and up to date information. The best way to provide this information is through a "migration workbook", a single information point that allows for the collection of documentation prepared for the migration (including the rationale, the detailed plan and the technical documentation) and the timetable, updated according to the project progress. This also simplifies project management when there is a change in the team performing the migration.

4 Social Guidelines

Provide background information on OSS

A significant obstacle to OSS adoption is acceptance by users, who usually have a very limited knowledge of open source and open data standards. In many cases, OSS is perceived as lower quality as it is "free" and downloadable from the internet like many shareware packages or like amateur projects. It is important to change this perception by providing information on how OSS is developed and its underlying rationale and business model.

Don't force the change on the users, but provide explanations

The change of IT infrastructure will force a significant change in how users work and how they use internal resources.

This change may cause resistance from the users. Such change may be simplified by explaining clearly why and how the change will happen, and what benefits will be introduced in the long term both internally (like lower cost, better flexibility and security) and externally (openness, adherence to international standards, less burden on external users).

Troubleshooting point: Be careful when internal users

believe that the migration is done to pay less for software.

Use the migration as an occasion to improve users' skills

As training for the new infrastructure will be required, it may be used as a way to improve overall ICT skills. For example, in many companies and public organisations users usually receive little in the way of formal training. Providing training helps not only in increasing confidence, but can also used to harmonize skills among groups and in general improve performance.

This may give rise to some resistance from the so called "local gurus", who may perceive this overall improvement as reducing their role as technical leaders. The best way to counter such resistance is to identify those users and suggest that they access higher-level training material (that may be placed in a publicly accessible Web site, for example).

It may also be useful to identify local "champions", i.e. local FLOSS enthusiasts, who can provide peer support to other users and offer them additional training opportunities or management recognition. In general, it is useful to create an internal intranet accessible page that provides links to all the different training packages.

Make it easy to experiment and learn

The licensing freedom that is the main point of OSS allows for free redistribution of software and training material. Therefore providing users with Linux live-CDs (that require no hard disk for installation) or printed material that can be taken home may help in overall acceptance.

Acknowledgements

The article is based on material prepared for the EU projects COSPA, OpenTTT and FLOSSMETRICS; more information is available in the FLOSSMETRICS guide for SMEs, <http://guide.conecta.it>.

References

[1] P. Carbone. "Value Derived from Open Source is a Function of Maturity Levels". OCRI conference "Alchemy of open source businesses", 2007. <http://www.ocri.ca/email_broadcasts/041907partnership.html>.

[2] EU COSPA Project. "D6.1 Report evaluating the costs/benefits of a transition towards ODS/OS". <http://www.cospa-project.org/>.

[3] KBSt, Germany. Migration guide, 2006.

Other Useful References

C. Daffara, J.B. Barahona. "Free Software/Open Source: Information Society Opportunities for Europe?". Working paper. <http://eu.conecta.it>.

C. Daffara. "Sustainability of FLOSS-based business models". II Open Source World Conference, Malaga (Spain), 2006.

C. Daffara. "Introducing open source in industrial environments". 3rd CALIBRE workshop.

IBM. "Linux Client Migration Cookbook, Version 2: A Practical Planning and Implementation Guide for Migrating to Desktop Linux". Available online at <http://www.redbooks.ibm.com/abstracts/sg246380.html?Open>.

