

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/FISI (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 Team

Chief Editor: Llorenç Pagés-Casas

Deputy Chief Editor: Francisco-Javier Cantais-Sánchez

Associate Editors: Fiona Fanning, Rafael Fernández Calvo

#### Editorial Board

Prof. Wolfried Stucky, CEPIS Former President

Prof. Nello Scarabottolo, CEPIS Vice President

Fernando Piera Gómez and Llorenç Pagés-Casas, ATI (Spain)

François Louis Nicolet, SI (Switzerland)

Roberto Carniel, ALSI – Tecnoteca (Italy)

#### UPENET Advisory Board

Majjaz Gams (Informatica, Slovenia)

Hermann Engesser (Informatik-Spektrum, Germany and Switzerland)

Brian Runciman (ITNOW, United Kingdom)

Franco Filippazzi (Mondo Digitale, Italy)

Llorenç Pagés-Casas (Novática, Spain)

Veith Risak (OCG Journal, Austria)

Panicos Masouras (Pliroforiki, Cyprus)

Thorvaldur Kári Ólafsson (Tölvumál, Iceland)

Rafael Fernández Calvo (Coordination)

**English Language Editors:** Mike Andersson, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green, Roger Harris, Jim Holder, Pat Moody.

Cover page designed by Concha Arias-Pérez

"Falling Upwards" / © CEPIS 2009

Layout Design: François Louis Nicolet

Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç Pagés-Casas <[pages@ati.es](mailto:pages@ati.es)>

Advertising correspondence: <[novatica@ati.es](mailto:novatica@ati.es)>

UPGRADE Newslist available at

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

#### Copyright

© Novática 2009 (for the monograph)

© CEPIS 2009 (for the sections UPENET and CEPIS News)

All rights reserved under otherwise stated. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, contact the Editorial Team

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (August 2009)

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

(The full schedule of UPGRADE is available at our website)



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*

- 2 Presentation. Libre Software for Enterprises: Create your Product, Feed your Community, Eat your Cake! — *Jesús-M. González-Barahona, Teófilo Romera-Otero, and Björn Lundell*
- 5 Libre Software and the Corporate World — *Jesús-M. González-Barahona, Teófilo Romera-Otero, and Björn Lundell*
- 11 Best Practices for FLOSS Adoption — *Carlo Daffara*
- 17 Build and Sustain a Community of Practice: Method Applied to FLOSS Projects — *Stéphane Ribas and Michel Cezon*
- 22 Community Management in Open Source Projects — *Martin Michlmayr*
- 27 The Morfeo Project: an Open Source Approach towards Open Innovation — *Cristina Breña and Andrés-Leonardo Martínez*
- 32 Applying Open Source Software Principles in Product Lines — *Frank van der Linden*
- 41 Addressing Industry Needs in OSS — *Jan-Henrik Ziesing*
- 44 SpagoWorld, the Open Source Initiative by Engineering — *Gabriele Ruffatti*
- 51 An Opportunity for Libre Software Companies: Emerging Market in Developing Countries — *Susana Muñoz-Hernández and Jesús Martínez-Mateo*

### UPENET (UPGRADE European NETWORK)

#### 54 From **Novática** (ATI, Spain)

Mobile Technologies

A Case Study of the Global System of Mobile Communication (GSM) in Nigeria — *Adeyinka Tella, 'Niran Adetoro, and Paul Adesola Adekunle*

### CEPIS NEWS

#### 60 Promoting Skills Development in Challenging Times — *ECDL Foundation*

#### 62 Selected CEPIS News — *Fiona Fanning*

\* 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/>>.

# Libre Software and the Corporate World

Jesús-M. González-Barahona, Teófilo Romera-Otero, and Björn Lundell

*Libre software is a new world in itself, for both companies and professionals. As the use of libre software products becomes more widespread in the software industry and in other sectors which are heavily dependent on software for their activities, it is becoming increasingly important to understand how to make the most of this new world. In this article we explore some of the new issues that emerge when approaching libre software, and how companies are addressing them.*

**Keywords:** Communities, Companies, FLOSS, Libre Software, OSS, Strategy.

## 1 The Libre (Free, Open Source) Software Concept

Libre software is a term we use throughout this paper to refer to both "free software", as defined by the Free Software Foundation<sup>1</sup>, and "open source software", as defined by the Open Source Initiative<sup>2</sup>. Although the two definitions are different, they cover almost the same collection of software. However, the terms are not interchangeable. For some, the word "free" refers to the freedoms that free software brings and therefore should never be omitted. For others, the ambiguous meaning of "free", which can also be understood as "*gratis*", is an obstacle to the understanding of the concept, especially in corporate environments. To avoid this possible confusion we will use the term "libre software" throughout this text.

To help define libre software it is useful to mention the "four freedoms". Libre software allows those who have it to: run and use it; study and adapt it to specific needs; redistribute it to others; and improve and add functionalities to it. All these freedoms can be exercised once the software is obtained, without the need for any more permissions from the copyright owners than that provided by the libre software licence covering the software.

From this point of view, libre software is mainly a legal concept: it defines some basic permissions that copyright holders grant to those receiving the software from them. In fact, copyright holders do not relinquish all their rights: they continue to reserve some rights, in the form of clauses and conditions specified in the libre software licences.

From an economic and business point of view, it is important to stress that there is no mention of how the software is obtained. It may be downloaded free of charge from a public repository on the Internet, or purchased for a princely sum, nicely wrapped, from a shop.

From a technical point of view, libre software is not in itself a technology. In fact, the only technological feature that all libre software has in common is that the source code must be available (as the term "open source" implies). Nevertheless, all libre software shares a number of common features in the way it is developed: open development models with more information publicly available; the existence of support communities; a mixture of volunteers and hired de-

## Authors

**Jesús-M. González-Barahona** teaches and researches at the *Universidad Rey Juan Carlos*, Mostoles (Spain). He started to be involved in libre software in 1991. Since then, he has collaborated in several working groups, has developed some research lines, and has started training programmes on the matter. He also collaborates in several libre software projects and associations, writes in several media about topics related to libre software, and consults for companies and public administrations on issues related to their strategy on these topics, within the framework of the GSyC/LibreSoft research group, <<http://libresoft.es>>. <[jjgb@gsync.es](mailto:jjgb@gsync.es)>.

**Teófilo Romera-Otero** earned a degree in computer engineering at the *Universidad Rey Juan Carlos*. His academic interests are libre software and how corporations relate to it. He is also interested in global software development in libre software environments and enabling technology through libre software. He works at the GSyC/LibreSoft group as an Research and Development (R&D) Project Manager on national and European projects, such as Calibre, Edukalibre, FLOSSWorld, Morfeo, Vulcano, Qualipso, FLOSSInclude and Tree. As part of his duties in the group he also participates in technical consultancy on libre software and coordinates teaching in the Master on Libre Software organized and run by GSyC/LibreSoft. He is a member of the Qualipso Network Board that manages the Qualipso Network of competence centres and of the NESSI OSS Working Group. He was a guest researcher at the University of Leeds (UK) and at the Irish Software Engineering Research Centre (Lero) at Limerick University in Ireland. <[teo@gsync.es](mailto:teo@gsync.es)>.

**Björn Lundell** is a researcher at the University of Skövde's Informatics Research Centre (Sweden). His research interests include open source software, evaluation, and method support. He was the technical manager of the COSI (Codevelopment Using Inner & Open Source in Software Intensive Products) research project. He is a founding member of the IFIP Working Group 2.13 on Open Source Software and the founding chair of Open Source Sweden, an industry association. Lundell has a PhD in computer science from the University of Exeter (UK). <[bjorn.lundell@his.se](mailto:bjorn.lundell@his.se)>.

<sup>1</sup> Free Software Foundation <<http://www.fsf.org/>>.

<sup>2</sup> Open Source Initiative <<http://www.opensource.org/>>.

velopers; meritocracy and community-based governance rules, etc.

From an ethical point of view, the hacker ethic may be strong among many of the participants in libre software projects, but this is not necessarily the case: many others will probably not recognize or even be aware of it.

This short introduction to the idea of libre software may be enough to show how complex this phenomenon is, with all its legal, businesses, technical, and ethical implications. Because of this multi-faceted nature, and the interrelationships between the various interests of the people involved, it tends to be a little baffling to newcomers. This is probably the root cause of many of the problems that companies face when they want to enter this new world of libre software.

### 2 Common Problems Encountered by Companies

At some point a company, whether it is focused on Information Technologies (IT) or just using IT for other purposes, will consider producing, integrating, using, or providing libre software-based services. There are many reasons why a company may consider taking this step, reasons which are widely discussed elsewhere and which we will not go into here. Rather than focusing on those reasons, this article will take the decision to "go with libre software", an increasingly common occurrence in corporate environments, as its starting point.

The hard part comes once the decision is taken. From that point on companies have to deal with a series of issues that, if properly addressed, can produce considerable benefits, but which if overlooked or ignored may also be the cause of a number of risks and problems. Several of these issues are briefly discussed in the remainder of this section.

#### 2.1 Dealing with Communities

Libre software development is usually carried out in communities which include not only the core developers themselves but also casual contributors or developers working on related products (maybe forks<sup>3</sup> for in-company use, for example), and different kinds of users (from end users who know almost nothing about the development itself to experienced integrators who know the product almost as well as its core developers). The individuals in these communities may be merely self-motivated volunteers, or professionals hired by companies to contribute in pursuit of corporate interests, or a combination of the two.

Any company approaching (or trying to create) one of these communities will encounter a new environment, quite different from the environment to which it is accustomed,

with its contractual relationships with other companies or employees. For instance, there are usually no contracts regulating relationships between participants in a community which can then be enforced once signed; motivation and the creation of win-win relationships is usually the name of the libre software game. Adapting to these new rules is not always easy.

Corporate image also plays an important role in these environments. When a company becomes a member of a libre software community (usually by means of their employees), it starts to acquire an image in the minds of other developers, which usually overrides the "traditional" corporate image. Others start to consider it in terms of how much it contributes to the community, benefits from it, collaborates with other parties within it, or even to what extent it is perceived as trying to control it. This image can produce all sorts of side effects.

#### 2.2 Legal Issues

Libre software licences, although based on the same intellectual property laws that IT companies are familiar with, are used in very different ways. Unfortunately, legal experts with a full understanding and experience of libre software licensing models are still not easy to find. But the proper analysis of licensing schemes, and the full understanding of not only their legal implications, but also their economical and technical implications, is vitally important to enterprises. Development schemes, ways of disseminating new technologies, or business models are usually constrained or empowered depending on the licences involved.

Libre software licences are to some extent the legal fabric that binds together all the players involved in a libre software product. Depending on the licence used and on how it is explained and enforced, other players will be more or less interested in collaborating. Licences also bring a sort of legal certainty to the companies interested in a product, and impact on many different issues, from the re-use of components to strategic dependence. Of course, they are also strongly related to other legal aspects that concern companies, such as laws regulating industrial property (patents).

Despite the importance of licences in the libre software environment, it is not always easy to get to the bottom of them. In fact, this is a field which the corporate world usually views as potentially troublesome.

#### 2.3 New Development Processes

Libre software can be developed using traditional processes, common to software development companies, but this is not usually the case. Libre software communities tend to use their own processes, which take into account, and even try to take advantage of, the specific characteristics of the development team. One of the main differences usually encountered by companies is that developers are not bound by contracts or hired by a single company. Instead, libre software development communities are usually composed of a mixture of volunteers and professionals hired by different companies. Because of this mixture, relation-

---

<sup>3</sup> In software engineering, a project fork happens when developers take a copy of source code from one software package and start independent development on it, creating a distinct piece of software. <[http://en.wikipedia.org/wiki/Fork\\_\(software\\_development\)](http://en.wikipedia.org/wiki/Fork_(software_development))>.

ships between developers, and between companies and developers, are quite different from traditional projects, which are normally characterized by a well-defined management hierarchy and clear contractual constraints if several companies are involved.

In most libre software projects, motivation is the watchword. Developers cannot be "managed" in the traditional sense of the word. Rather they work together, with decision and planning processes usually based on the notion of meritocracy and mutual trust. This does not of course mean that decisions are not taken, or that everyone's opinions and proposals are considered in the same way. Libre software projects usually follow certain rules, which in some cases may even be partially in writing, but they can be quite different from those in corporate environments.

This new environment tends to be difficult to understand for some developers in companies, who may have many years of experience but may be inexperienced in how libre software projects work. For instance, they may find the fact that their contributions may be accepted or rejected difficult to understand, and they will need to learn how best to submit a contribution in order to get it accepted.

When a company embraces a libre software product, or decides to launch its own, this kind of problems have to be considered. However, there are several strategies to deal with them, from hiring new personnel trained in the libre software environment to providing the in-house team with training to make them aware of and experience the kind of situations they will encounter.

### 2.4 Driving a Product

When a company launches a new libre software product, it is usually interested in maintaining some degree of control over it and its future evolution. In the case of proprietary software, the strict licensing terms of the product guarantees this control. But in the case of libre software, others can also work on it, improve it (maybe along lines not desired by the company) and redistribute new versions of it, provide services based on it, and maybe, in time, take over the product (in the sense of distributing the most used and well-known version).

To avoid this, companies may use several strategies, but this is a new world for them. These strategies are based on maintaining their status as the primary source for the product, which usually depends on how much effort and investment the company is prepared to put into the product once it has been distributed. Therefore, they need to plan for this in advance, or else risk losing their privileged status.

This situation is even more problematic when the company is not the sole producer but has joined a development community. Usually, several companies compete within the community for the position of "project leader". Finding win-win situations which involve other players while, at the same time, understanding how to contribute to the creation of a healthy community which will ensure the viability of the project, but ensure a certain degree of control over certain decisions, is not any easy task.

## 3 New Strategies to Address New Situations

Libre software is a new world for companies. This is uncharted territory not only for those producing software as their main line of activity, but also for others which need software for their products or services. This latter category accounts for a large percentage of all companies, since companies are more and more strategically dependent on software.

Therefore, new strategies are needed and many companies are actively exploring such strategies to benefit from libre software, to use libre software as a means to further other ends, or even as an integral part of the company's general business policy. In this section, we look at some of these strategies, which in many cases amount to new ways of collaborating with other players.

### 3.1 Communities of Companies

Communities of companies based on libre software represent a new form of corporate collaboration, not based on contractual relationships. The core around which the community forms is usually a set of libre software projects, which may or may not be interrelated. Companies participate in the community because they have some interest in the software (they either wish to produce it, use it, provide services for it, etc.), or even in the community itself, for the business opportunities that may emerge from it.

In some ways these communities are modelled on traditional libre software communities, the main difference being that they are composed of companies rather than individuals. They are not usually subject to formal institutional alliances, although in some cases there may be an explicit concept of membership. To some extent, the same principles of meritocracy and "those who contribute the most make the decisions" found in other libre software communities also apply here.

They are also similar to informal interest groups, but libre software communities are especially bound together by the code they produce. Within this framework, companies interested in producing a certain product simply start developing it, maybe after reaching collaboration agreements with other players. Business actions may be coordinated by several members, and in some cases there may be a certain degree of pooling of resources to help launch new ideas. Usually communities comprise a mixture of large companies and small and medium enterprises (SMEs), together with research institutions and public bodies. This special mix facilitates both the transfer of knowledge from academia to industry and the creation of rich ecosystems around specific libre software projects.

The fact that the software produced is libre software ensures that no one has complete control and that communities strive to create situations with which all contributing players are comfortable. In some way this levels the playing field and sets some rules which, under certain circumstances, allow even competing companies to collaborate on specific projects in their mutual interest, without the burden and complexity of drafting detailed legal agreements.

Some examples of these communities are OW2<sup>4</sup> (formerly Object Web), built around Web technologies, and Morfeo<sup>5</sup>, which deals with various technologies.

### 3.2 Corporate Alliances

Alliances are a specific case of groups of companies in which a certain goal, common to all the participating companies, acts as the glue to hold the alliance together. This goal is usually the production or promotion of a specific technology or standard based on libre software. Participating companies usually sign a common agreement and commit resources to that end. In broad terms they are not so different from other kinds of corporate alliances, the main difference being that they are built around libre software. This fact usually helps the alliance to establish itself as a neutral point of control, open to new players. Since nobody controls the technology (because it is libre software), in principle the playing field is levelled for all participants to compete according to their capabilities.

Some examples of libre software based alliances are LiMo Foundation<sup>6</sup>, Open Handset Alliance<sup>7</sup>, and Symbian Foundation<sup>8</sup> (all working in the field of operating systems for mobile devices), and Genivi Alliance<sup>9</sup> (aimed at producing a software platform for cars).

### 3.3 Promotion of Traditional Communities

Another strategy which is commonly used by companies is the creation and promotion of a community of developers and users, mimicking traditional libre software communities. In this case, a company leading a libre software project tries to interest other companies and individuals in it by using a structure based on meritocracy, contribution and trust, similar to what happens in communities driven by volunteers.

The company promoting the community plays the role of benevolent promoter, to some extent renouncing some

of the benefits of being the primary source of the product. The fact that the product is libre software also provides some guarantees to other players as to the promoter's lack of total control, which encourages them to participate. Usually, win-win relationships are established, in which the promoter benefits from substantial amounts of effort and innovation put into the product, while the other players benefit either from the resources contributed by the promoter, or from business opportunities arising from the scale of the project.

Some companies producing Linux-based software distributions, such as RedHat<sup>10</sup> with its Fedora community<sup>11</sup>, or Canonical<sup>12</sup> with its Ubuntu<sup>13</sup> community, were among the first to explore this strategy. More recently, we have the well known case of Nokia<sup>14</sup> with its Maemo<sup>15</sup> community (although there are many more cases).

### 3.4 Involvement in Traditional Communities

In some cases, companies consider that the best strategy is to join an existing, traditional libre software community. In many cases, these communities started out as a group of individuals, probably volunteers, with the common goal of pooling efforts to develop and maintain some specific software product. But over time it is common for developers hired by companies interested in the product to join the community, or for companies to hire former developers directly, so that in some way they can join (and, in some cases, try to influence) the development effort.

This was the case of one of the better known libre software communities: Apache<sup>16</sup>. Since its beginnings, several key developers have been working for companies while participating in Apache projects in their working hours. The Apache community recognized this fact, but tried to maintain its status as a community of individuals, in which members are considered as individuals and not as representatives of the company they work for. However, tensions due to corporate interests can arise from time to time.

Sometimes communities formalize the participation of companies. The community agrees to receive donations (either in the form of efforts from employees, cash, or in other resources, such as computing infrastructure), and specific rules are established to strike a balance between the interests of the companies and those of the project as a whole. Usually companies try to drive the project not by asking for more influence due to their donations, but by producing code for the specific functionalities they want. In some cases, this leads to the creation of new products within the project. This can distort the relationship between volunteers and employees in the project, but it also produces a number of benefits: there is usually an increase in the volume of human resources actively working in the project, and fixes may be found for specific problems which volunteers are seldom interested in dealing with.

There are many examples of this case. Probably the best known case is the GNOME<sup>17</sup> project, in which, during the late 1990s and early 2000s in particular, many companies collaborated (including large ones, such as Sun<sup>18</sup> or Novell<sup>19</sup>, or small start-ups, such as Eazel<sup>20</sup> or Ximian<sup>21</sup>).

---

<sup>4</sup> OW2 Consortium <<http://www.ow2.org/>>.

<sup>5</sup> Morfeo Project <<http://morfeo-project.org/>>.

<sup>6</sup> LiMo Foundation <<http://www.limofoundation.org/>>.

<sup>7</sup> Open Handset Alliance <<http://www.openhandsetalliance.com/>>.

<sup>8</sup> Symbian Foundation <<http://www.symbian.org/about/index.php>>.

<sup>9</sup> Genivi Alliance <<http://www.genivi.org/>>.

<sup>10</sup> RedHat <<http://www.redhat.com/>>.

<sup>11</sup> Fedora <<http://fedoraproject.org/>>.

<sup>12</sup> Canonical <<http://www.canonical.com/>>.

<sup>13</sup> Ubuntu <<http://www.ubuntu.com/>>.

<sup>14</sup> Nokia <<http://www.nokia.com/>>.

<sup>15</sup> Maemo.org <<http://maemo.org>>.

<sup>16</sup> The Apache Software Foundation <<http://www.apache.org/>>.

<sup>17</sup> GNOME: The Free Software Desktop Project <<http://www.gnome.org/>>.

<sup>18</sup> Sun Microsystems <<http://www.sun.com/>>.

<sup>19</sup> Novell <<http://www.novell.com/>>.

<sup>20</sup> Eazel <<http://en.wikipedia.org/wiki/Eazel>>.

<sup>21</sup> Ximian <<http://en.wikipedia.org/wiki/Ximian>>.

A more recent case is the relationship of the Mozilla foundation<sup>22</sup> with several companies, especially Google<sup>23</sup>, from which it receives a considerable amount of funding.

### 4 The Role of SMEs

Some small and medium enterprises tend to be very active in terms of innovation and the creation of new technology. Usually they are considered to be more agile and dynamic than large corporations. It is therefore no surprise that many of them were pioneers in the field of libre software.

Some of the first cases arose as a result of libre software developers funding an SME to monetize their investment in the creation of a successful libre software product. In this case, the development was started by a community of volunteers but, at some point, part or all of them decided to create a company, which usually took over a large part (if not all) of the development, while at the same time they launched services targeted at obtaining profits. One canonical example of this case was MySQL<sup>24</sup>, in which the main developers founded MySQL AB, the company that drove most of the development. At the same time, they offered business services with which they obtained income to fund that development.

This model is found in many other cases before and after MySQL. In fact, it is so popular that in many cases the company is established even before the development actually starts, with the specific aim of driving it. To some extent, Red Hat or Ximian followed this model, and there are a host of other examples. In time, such companies may be bought by large corporations, as was the case of MySQL AB (acquired by Sun) or Ximian (acquired by Novell), thereby incorporating innovation into their corporate strategies.

Libre software developers may also find it easier to get a job in some innovative SMEs. In fact, hiring libre software developers is a way for companies to demonstrate their involvement in the products or services they offer (when they are based on libre software), and a way of differentiating themselves from the competition. The relationship between SMEs and volunteers in some projects is so close that developers may switch from one SME to another while they continue to work on the same project, or vice versa. From many points of view, a large number of projects would simply not be possible without the participation of SMEs and the resources they commit to their development, while at the same time those SMEs would not be viable without the successful libre software products with which they work.

The relationship between SMEs and large corporations in development communities is not always easy. SMEs find

it difficult to create rich ecosystems by themselves, which means that they normally welcome the entry of large companies. But the entry of large companies can create a situation of imbalance for the smaller players. Although the smaller companies are usually more active and dynamic, the sheer volume of the large companies, and the resources they have available, can drive the project in unforeseen directions. However, when their different capabilities are used to find synergies, the relationship between SMEs and large companies can provide an enormous boost to the project and benefit all the players.

### 5 Libre Software and IT Professionals

IT professionals are at the heart of libre software development. Despite many uninformed opinions, most libre software projects are led and driven by experienced software developers, usually with many years of experience. In some cases, they have always been involved in libre software, but in others they come from the traditional software industry. Maybe they started to work in libre software projects as a hobby in their spare time, or maybe they became involved because of a decision made by their company. But in any case, they, as individuals, are generally speaking the glue that keeps the projects together.

As we have mentioned several times in this article, in many cases companies are involved in projects but are not "first-class citizens" in them. Most libre software communities are based on relationships between individual people, not on corporate agreements. Therefore, the role of IT professionals participating in libre software communities is usually far more important than the role of the companies for which they work. To some extent, most libre software projects are actually driven by IT professionals, who are recognized on the basis of their merit and effort.

In fact, this recognition of merit and effort leads to a new role for the professional developers. Instead of being constrained by non-disclosure agreements and company policies which prevent them from showing their peers their capabilities and achievements, the game in the libre software world is exactly the opposite. All work is public, and all actions are traceable. Professionals doing a good job are recognized by their peers, and they can carry their reputation with them as they move from company to company.

From the point of view of life-long training, which is so fundamental for IT professionals, libre software is also a world of opportunities. By reading the code and understanding the architecture of libre software programs, new technologies and paradigms can be more easily understood. The professional is no longer limited to the kind of technologies used in his or her company, but can join and collaborate with other peers in state-of-the-art endeavours, thereby keeping up-to-date with new trends. In some cases this can be achieved with the support of the company for which the professional works, but it can also be achieved in his or her spare time, perhaps paving the way for a move to a more innovative job. However, certification programmes to certify the specific capabilities of a developer in the field of

---

<sup>22</sup> Mozilla Foundation <<http://www.mozilla.org/>>.

<sup>23</sup> Google <<http://www.google.com/intl/en/corporate/>>.

<sup>24</sup> MySQL <<http://www.mysql.com/>>.

libre software are not as well developed as in the field of proprietary software. This can also be a problem for capable, experienced professionals who do not have a clear and simple way of demonstrating their competence.

In summary, libre software brings new opportunities for professionals. Of course, there are also some risks, such as the problems arising from publicly exposing improper coding practices, for example. But all in all, professionals have a great deal to benefit from by working in libre software projects.

### **6 Conclusions**

Libre software is a new field in the IT landscape, not because it is a new technology, but because it brings new procedures and possibilities. Both companies and professionals can obtain benefits if they find how to take advantage of their capabilities and can align their goals with those of the libre software community. However, as in any new world, to reap the full benefit we need to understand the new rules and explore the new opportunities. And of course, there are also a number of risks, especially for those who fail to realize that, in many senses, the environment has changed.