The Long Road To Free Software in Quebec.

by Cyrille Béraud, Savoir-faire Linux CEO
cyrille.beraud@savoirfairelinux.com

During the course of the year 2000, I was contacted by the IT Manager from the Valleyfield plant of one the industry's largest tire manufacturers.

He informed me that he had decided to migrate several applications that still ran on old systems to GNU/Linux. Our first meeting included a tour of the factory. It was the first time I had ever been inside a tire manufacturing plant. This world reminded me of Charlie Chaplin's *Modern Times*: a gigantic factory, several buildings high, where, amidst a deafening racket and a wreaking odour, huge steel machines engaged in a clinking frenzy as they exchanged production components by means of long conveyor belts that criss-crossed through space in every direction.

This dance, so often rehearsed and corrected, like sheet music, did not suffer from a single false note. Electronic sensors by the hundreds, control systems, computers of all sizes and hundreds of kilometres of cable silently orchestrated this strange symphony.

At the center of this surrealist world reigned the presses: the plant's proud and majestic queens. Objects of all the attention, cherished above all others, they accomplished the most noble and crucial task. Using components prepared by their little sisters, large steel arms grasped pre-cut pieces of rubber in a deadly embrace and suddenly brought these every day objects to life with a violent hiss of steam that shook me to the core. All that was left was a quick clean up and the tire would be off on a final journey to the warehouse, from where it would be shipped a few hours later, in order to land in a garage or at a car manufacturer's.

This visit made a lasting impression on me. At last I had discovered the secret of our industrial economy! Or so I thought. Some questions still plagued me, presenting new mysteries to be pierced and I seized on a break in the din to question my host: "Where on earth did this well established and noble company make these gigantic presses? What amazing minds had created them? Could we meet with them?" His answer left me in a profound state of perplexity: "We don't build these machines. We have the same ones as all of our competitors. There are two components manufacturers in the world: one is German, the other is Japanese and they supply all of the world's tire manufacturers!" I was speechless. What then, was the difference between these companies? Where lied the secret to competitiveness if everyone had the same equipment?

Wanting to prove that I was a good student, I anticipated the answer: wages and payroll taxes! That, as so many politicians proclaim, is the secret that will make all the difference.

But my new teacher hardly bothered to answer me. He gave a shrug and grumbled: "Our competitors relocate to all the same places we do. In the end, it doesn't widen the gap." After several long minutes of silence he added: "You see, this year we're expecting a particularly cold and early winter here in Quebec. The first among us to produce the new next generation winter tire will have the advantage. We know how to make the tire and we can do it well and inexpensively at that. The problem is making it fast enough to stay as close to market demand as possible. In order to do

so, we have to reprogram all the computers that guide these machines and that; we don't know how to do. We don't know how because the software doesn't belong to us. We only have a right of use. We can't make any changes to the programs we use since we don't have their sources. We have to ask our suppliers who generally don't know anything about our specific constraints, they don't have the same priorities as us and besides, even in the best case scenario, we would end up having to pay big bucks."

He continued: "It was still possible a few years ago when there were a lot less computers and they didn't communicate with each other. Today, all computer systems must exchange information and not only within the plant, but with our research centers, our suppliers, our clients, accounting, marketing agencies, etc. Everything is interconnected... Now,when we want to make a change, all of our software providers (that's hundreds!) are concerned and it's become impossible." The report was overwhelming: " We have no control over our own information systems! And yet that is the one and only area in which we can achieve the necessary gains in productivity."

That day, I came to understand the many needs that are fulfilled by free software and how it is of crucial importance to our country's economy.

Free Software

What is free software exactly? In software engineering terms, free software is the consequence of a new legal framework, which controls its trade and use. To fully understand the concept, we must go back a few decades: it is most likely at the end of the Second World War that the first legal elements regarding regulation of the software business were put into place.

The budding post-war computer industry began to commercialize software programs. Legislation was forced to determine the rules that would regulate this new trade. It was decided, and such was the case in most countries, that since dealing with a series of typographical characters, copyright was the most appropriate legal framework for this situation. Software is therefore not considered to be a good and that only its use, its right of use, can be sold. The software's owner remains the publisher; this is the *proprietary model.*

The consequences of this error of appreciation will weigh heavily on the industry's future and on today's businesses, particularly for the two following reasons:

On the one hand, the legal framework resulted in a situation of *de facto* monopoly with all the inconveniences that come with such a situation. When you buy a book, for example, you enjoy the book then place it in your bookcase. You don't need that book to read the others books. Your bookcase is filled with a multitude of books, all independent of one and other. Such is not the case with software. Every program uses components from other software programs. Every program will need to communicate the data it has processed to other programs. It's as if choosing the work of one publisher at first obligates you to buy all of his books and only his books.

In the field of computer science, the consequence was felt immediately and it wasn't long before one software publisher, whose only merit was to have been in the right place at the right time, found itself in a monopoly situation in several market sectors, dictating its prices and products to the detriment of productivity and free competition.

A second consequence is deduced from the fact that computer science, like mathematics for example, is a cumulative science. Every progress uses past progresses in successive layers. The more computer science advances, the more expensive it becomes due to the price that has been placed on each one of these algorithmic layers. This framework is therefore the source of a boundless waste of resources and intelligence as it imposes the constant re-writing of all these layers.

We have Richard Stallman, a researcher from the Massachusetts Institute of Technology (M.I.T.), to thank for taking a closer look at the existing legal framework back at the beginning of the 1980's, when he put forth a model that would meet the new economic and political requirements that are tied to the ever growing emergence of information technologies in our societies. In legal terms, this new framework is called the GPL (General Public License), in software engineering terms, it is called free software. This framework is based on four liberties, which Richard Stallman sees as fundamental :

- Freedom 0: The freedom to run the program for any purpose.
- Freedom 1: The freedom to study how the program works, and adapt it to your needs.
* *Access to the source code is a precondition for this*
- Freedom 2: The freedom to redistribute the program so you can help your neighbor.
* *This includes the freedom to sell copies*
- Freedom 3: The freedom to improve the program and release your improvements to the public, so that the whole community benefits.
* *Access to the source code is a precondition for this*
This encourages the creation of a community of developers all working to improve the software.

Based on his new framework, Richard Stallman invited this community of developers, which was mainly supported by the simultaneous emergence of the Internet, to re-write all operating system software. In a few years the GNU/Linux operation system was operational and of a technical quality that was unprecedented in the history of computer science. The entire spectrum of applications used by both companies and private users was henceforth freely placed at everyone's disposal. Thanks to the collaborative development model, these programs are getting better every day and being enriched with countless new functionalities.

Towards An Economy Based on Expertise and Innovation

I will only touch on the economic aspects of this issue. The political and democratic stakes are certainly fascinating, but I will limit my comments to that which I have ascertained as certain based on my knowledge of the business world. From an economic standpoint, the consequence for a software industry that subscribes to this new framework is relatively simple. When you sell software, you're selling the program's sources and giving the client the four-abovementioned freedoms. Yes, software publishers are loosing a certain amount of power, but when you think about it, they are gaining even more: the power to use all software developed under GPL licence freely and without charge.

As for the businesses that use these programs, such as our tire plant and in the case of most industries, the ability to control constantly mutating information systems will allow them make important gains in productivity.

Free software not only gives businesses the power to control their system's software components, it also allows them to shape these components according to market

requirements and the ever changing constraints of their production processes. It is important to note that the larger organization, the more complex and heterogeneous the information system and therefore the more free components will be needed to give it structure and allow it to evolve.

Using free software is also beneficial to businesses for other reasons that are more difficult to evaluate but no less important. The first being the perpetuity of human investments. Indeed, a free software economy is eminently an expertise-based economy. The organization uses the expertise from the free software in order to create an information system that's tailored to its needs and by using software components that, like tiny bricks, come together until the desired result is obtained. Once trained on that tiny brick-assembling machine, the employee is trained for life and with every passing year his or her knowledge of this or that brick, or of the organization's specific computer needs increases.

This of course, is in strong contrast to the proprietary model in which the organization will regularly spend considerable time and money to train its personnel on which button to press without understanding the system's intrinsic mode of operation. Companies have only learned to compensate for these repetitive non-value added expenses that only yield fleeting results with mind-boggling personnel turnover rates that generate monumental waste.

By minimizing unproductive investments (licenses) and maximizing productive investments (service), a free software economy is a sustainable, highly value added economy that creates lasting highly qualified jobs.

The State and Free Software

The roles and positions of the State, public administrations and territorial collectivities are, on this subject, exemplary for at least two reasons. Firstly because the State is above all, a huge information processing machine that is complex, heterogeneous and in a constant state of mutation. Every law that is voted by our assembly raises an army of computer specialists charged with the task of transforming these texts into long and complex algorithmic processes. In terms of State modernization, the ability to control this information system is at the heart of the issue.

Secondly, as with traffic laws, it is up to the State to provide a common framework that will allow different social and economic players to circulate on these information highways and communicate with each other. Imagine if, on certain sections of road, red lights suddenly meant green and if we were forced to drive on the left side just because the private owner decided so? Imagine if, in order to communicate with its citizens, the State imposed the use of a single supplier and forced everyone to buy the same operating system, the same e-mail provider, the same Internet browser, etc.?
This is why free software has become the norm in many governments around the world where the success and confirmation of our hypotheses are being demonstrated every day. In France for example, the entire Ministry of Economy and Finance is gradually shifting towards the use of free software. With over 170,000 workers, this ministry alone employs two times more civil servants than the entire Quebec public service. Still in France, the National Police, the National Assembly *and* the Ministry of Culture are but a few other examples. Most European countries have already taken decisive steps (Germany, Spain, Italy). Among other large countries, Brazil and India (renown for its software industry) are very advanced in the process. Several American states have already defined some very ambitious voluntarist strategies.

What is the free software situation in Quebec? Let's not beat around the bush: it is clearly catastrophic and in certain respects, even down right scandalous.

It is characterized by a total absence of political will, archaic regulations and an apparent hostility from those who are in charge of these issues. In Quebec City, a questionable connivance with the old proprietary economy rules.

When free software is used within the Quebec public administration, it is at best, anecdotal. With the exception of a few rare cases (that deserve to be applauded) where real strategies have been put into place, only a few non-connected services spread out in different areas use it sporadically. In most ministries, the use of free software is bluntly prohibited or systematically denounced.

In its issue of April 10th 2007, *Le Soleil* published comments made by an executive civil servant from the *Ministère du centre des services partagés* in which he unabashedly and publicly denigrated the free software industry in Quebec, which, in his opinion, "is more akin to a fringe industry". With a composure worthy of the Soviet era and without a trace of the impartiality or fairness that is expected of him, he shamelessly declared: "We sign umbrella contracts with companies so that ministries that need to update the versions of their software can automatically buy more products from Microsoft, Novell or IBM."

Quebec's taxpayers and citizens know it: a great majority of all software license purchases, estimated at tens of millions of dollars per year, are made directly without bids, through mutual agreements, without comparisons and without giving other suppliers a chance to present alternative solutions.

All of this is supported by archaic bidding regulations that seem to permit this type of procedure. It will be up to the courts to decide. Whatever the case may be the fact remains that public markets are closed to free software and no one seems to be in a hurry to move things along.

The situation is absurd and grotesque. After decades of discussion with the same suppliers within the scope of protected markets, it comes as no surprise that we have developed these bad habits. After the embarrassment and scandal surrounding the GIRES[*] project, nothing will ever shock us again.

Quebec's taxpayers and citizens know it: every dollar that would be spent by the government on free software would mean the creation of sustainable employment in Quebec, more powerful on-line governmental services, expertise acquired for the future; it would mean money that would stay in the country.

The modernization of our Public Administration is imperative and because this modernization will only come to be if we can control our own information system, the need for a resolute and informed policy on the use of Free Software within Quebec's public administration is long overdue.

Live Free or Die

The tire plant closed last winter. Eight hundred employees lost their jobs. The multinational blames the closure on its inability to adapt to new production processes. Only a small division of the plant will remain operational, saving two hundred jobs. In fact, for reasons that no one seems to be asking, that division, of all the company's plants around the world, is the most successful and productive.

Strangely, it was seven years ago in this same division, that the IT Manager went against the better judgment of all his superiors and proprietary software suppliers and decided to use free software and trust new and emerging Quebec companies that specialized in GNU/Linux.

* The GIRES project, acronym for *Gestion Intégrée des Ressources*, a vast governmental computer project aimed at the global standardization of the State has, according to Radio Canada, cost taxpayers over a Billion dollars without yielding any results. The saddest and most chocking demonstration of our theses in terms of the need to control the components of an information system as soon as it becomes too complex or in a constant state of mutation.

Translated from French by Kelly L'Archevêque.