The Business Model of Free Software: Legal Myths and Real Benefits

Franco Gasperoni

Session 3: Business Models

Title: The Business Model of Free Software: Legal Myths and Real Benefits

Keywords: Free Software, Business Models, GNAT Pro.

Author: Dr. Franco GASPERONI
Position: Managing Director
Company: ACT Europe
Address: 8, rue de Milan, 75009, Paris, FRANCE
Telephone: +33.1.49.70.67.16
Fax: +33.1.49.70.05.52
e-mail: gasperon@act-europe.com

Bio:
Franco Gasperoni has an engineering degree from the Ecole des Mines de Paris, France and a PhD in Computer Science from New York University, USA. While at the Ecole des Mines, Franco did a master thesis with Maurice Allais, the French Economics Nobel prize, on the fundamental instability of the international banking system. For his PhD, Franco worked at IBM Research on compilation techniques for VLIW architectures.

Franco then taught Programming Languages and worked on the Free Software GNAT Ada project at New York University. Later on he joined the Ecole des Telecommunications (ENST), in Paris, where he taught Programming Languages, Software Engineering, Computer Architectures, and Operating Systems. While at the ENST, Franco conducted research in advanced compilation techniques, software development, the Ada programming language, and participated to the MPEG-4 standardization effort.

With Cyrille Comar, Franco is co-founder and Managing Director of ACT Europe, the company that develops, maintains, and offers commercial support for the GNAT Pro Ada technology in Europe. ACT Europe is a 100% Free Software Company basing its business model on high-quality support and subscription-based pricing. Franco, who is a member of IEEE and the ACM has published over 20 papers.

The Business Model of Free Software: Legal Myths and Real Benefits

Abstract

Free Software is the term coined by Richard Stallman in 1983 to denote programs whose sources are available to whoever receives a copy of the software and come with the freedom to run, copy, distribute, study, change and improve the software.

As Richard Stallman's concept grew in popularity, and with the subsequent advent of GNU/Linux, Free Software has received a great deal of attention and media publicity. With attention and publicity came expectations as well as a number of legal myths and confusions.

The objective of this article is to clarify some of these legal misunderstandings while explaining how the legal fundamentals on which Free Software is based allow for a long-lasting business model based on a special kind of expertise-based support that benefits customers and guarantees the creation of a local pool of expertise.

This article is based on our experience with GNAT Pro. GNAT Pro is the Free Software development environment for the Ada 95 programming language. It comprises a compiler that is part of GCC (the GNU Compiler Collection), a toolset and graphical Integrated Development Environment, and a set of supporting libraries.

Developing, maintaining, and marketing GNAT Pro for ten years have provided significant experience with both technical and non-technical aspects of Free Software. This article summarizes the principal legal and business lessons learned.

Disclaimer

Copyright and intellectual property rights (IPR) law differs from country to country. The following distills some copyright and IPR aspects as they apply to software in several western countries. Please seek legal advice for specifics in your own country.

Principles of Copyright

Copyright is the exclusive legal right to copy, distribute, modify, display, perform, rent or more generally exploit a work by its copyright holder. The creator of the work automatically owns the copyright unless one of these holds:

- Work-made-for-hire implied by employment;
- A specific work-made-for-hire agreement was put in place;
- A specific contractual assignment occurs.

Not everything can be copyrighted. An idea, for instance, cannot be copyrighted. Copyright protects the way the idea is expressed in a piece of work, such as a book, but it does not protect the idea itself.

In the 19th and first half of the 20th century copyright concerned mostly literary, musical, or artistic works. Today copyright equally concerns software works. In the context of software both source code and binaries are protected by copyright. Loading a copy of the software onto a computer is considered copying.

Only the copyright holder of a software program is allowed to copy, modify, make derivative works, and distribute the software. Anybody else needs permission from the copyright holder. In the software industry this permission is called a software license.

Nothing requires the copyright holder of a software work to give a license, or to give the same license to everyone. Copyright holders can, at their discretion, charge a fee in exchange for a copy of the software work and its attached license. In this case what the recipient is buying is not the software work per se but a copy of it. Virtually all software products are sold this way.

The software license specifies the conditions under which the recipient can (and most often cannot) copy, distribute, modify, the copy of the software received. Software licenses have a great degree of freedom in the conditions or restrictions that they can contain. Not every condition is legal. In the European Union, for instance, software users are allowed to decompile a software work distributed in binary form for the purpose of interoperability with other systems.

Free Software

In the 60s and 70s software was bundled with the computer hardware sold to a customer and came with full source code. In those days the computer itself was the added value and software a mere technological necessity to operate valuable hardware. Twenty years later the situation reversed. Computer hardware had become a commodity and the added or differentiating value was the software application. In the course of this transition software went from being a give-away to being ferociously protected with copyright (and patents). In the process software developers went from being an open community where collaboration was the rule to a multitude of closed shops protecting their software works in hope to become the next millionaire.

To rekindle the spirit of cooperation and openness that existed in the software community Richard Stallman created the GNU project in 1983 to create a free software system upwardly compatible with Unix ("GNU is Not Unix" goes the recursive acronym). In doing so Richard Stallman created the Free Software movement and the Free Software Foundation (FSF). In addition to being the organizational sponsor of the GNU project, the FSF mission is to "*preserve, protect and promote the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free Software users*" [1].

But what is Free Software exactly? The *free* in Free Software refers to the following four types of freedom:

1. The freedom to run the software for any purpose;
2. The freedom to redistribute copies of the software;
3. The freedom to study how the software functions and adapt it;
4. The freedom to improve the software and release the improvements.

Access to the software source code is a necessary pre-condition for the last two items.

Free Software and Public Domain

When Richard Stallman created the GNU project and the notion of Free Software he created a special software license to protect Free Software. Why? Isn't public domain what Free Software is all about?

Public domain software is software that is not copyrighted. This can happen if the copyright holder has explicitly put the software work in the public domain or if "*enough time*" has passed

from the creation of the software work. Under the American Copyright Term Extension Act of 1998, for instance, "*enough time*" means life of the author plus seventy years.

It is perfectly possible that the binaries of a software work be in the public domain without availability of the corresponding source code. If the source code is in the public domain as well it is possible for anyone to take those sources modify them, assert the copyright on the new derived work and redistribute the modified software under very restrictive conditions violating the four basic freedoms of Free Software.

Thus, had Richard Stallman decided to release the GNU project as public domain software with sources, anyone could have taken that work, modified it and redistributed it under non Free Software terms, thereby defeating the very nature of Stallman's objective.

The GNU GPL (General Public License)

To protect the four freedoms of Free Software, Richard Stallman created the GPL (General Public License) [2], the main, but not the only software license under which Free Software is distributed.

In this respect Free Software is no different from proprietary software that closed software companies like Microsoft sell. A Free Software program comes with a license (the GPL), just like the software work from closed software vendors. If a user of Free Software violates the GPL the copyright holder can take legal action, just like closed software vendors can if their software license is violated.

The big difference between the GPL and closed software licenses is that the GPL is written to favor the users of Free Software, while the license of closed software vendors is written to allow the bare minimum and to favor the closed software vendor. Specifically, the GPL allows:

- Copying with no restrictions of any kind;
- Redistribution of original software with sources;
- Redistribution of derived works with full sources.

When a user redistributes a Free Software program that person must make available to the recipients all the program sources. If the user modifies a Free Software program and distributes it, the user must make available to the recipients the sources of these modifications.

Free Software distribution is a transitive process: you must confer to others the same rights that were conferred to you. The idea behind Free Software is to encourage secondary distribution, derived works, general use, and create the notion of ownership of ideas by the community interested in the software work.

Free Software and Internet Downloads

It is not because something can be downloaded with sources from the Internet that it is necessarily Free Software. In fact it is the user's responsibility to check the license of the software downloaded. If the originator of the software is not willing to make any license claims the user cannot make any assumptions on whether what he is doing is legal or is infringing somebody's copyright. This is typically not a problem if students, academics or hobbyists use the software. It can, however, be a major liability for somebody using the downloaded software to implement a commercial product.

When a user receives a software product from its authors, Free Software or otherwise, by paying a fee or gratis, the user receives the software license allowing to load the software on a computer under certain conditions. This license, explicitly obtained from the authors or included in a shrink-wrapped box, protects the user against third party copyright infringement claims. It is the safe way that a software product can be used, Free Software or otherwise, without having to worry about legal implications.

In the case of Free Software, strong of the rights that the GPL confers recipients, users of a Free Software program can in turn redistribute the program and make the legal claim that the software is Free Software. There are two things that re-distributors of Free Software cannot do and that is (a) change its license (b) take legal action against somebody that violates that license. Both (a) and (b) can be done only by the original software copyright holder(s).

Free Software and Public Availability

The only obligation that Free Software licenses, such as the GPL, impose on its users is that any work derived from a Free Software program must itself be Free Software. This means that if one distributes a Free Software program, then that person must make its sources available and allow its recipients to do likewise. However, there is nothing that requires that anyone distributes or redistributes a Free Software program.

Free as in Freedom

There is some confusion over what can be charged for a Free Software program. This confusion has arisen from the sentence in the GPL that says

"...for a charge no more than your cost of physically performing source distribution"

This sentence only applies to the charge of distributing the sources a posteriori if the sources are not distributed with the product itself. This statement does not apply to the distribution of the product. As Richard Stallman puts it:

> *"Free Software" is a matter of liberty, not price. [...]*
> *you should think of "free speech", not "free beer."*

Thus, like any software product, there are no specific restrictions on the price that can be charged for software that is licensed as Free Software.

Business Models of Free Software

Several business models can be built around the notion of Free Software. We enumerate below the key ones which can be mixed in any number of ways to give rise to hybrid business models.

1. **Pure Free Software Product.** In this model the software vendor limits itself to selling a stand-alone, self-contained set of Free Software programs with perhaps some minimal installation help. Because recipients of Free Software have the freedom to redistribute copies of the software product (and charge for it) this limits the amount that the initial vendor can charge to pretty much commodity costs, unless the Free Software work is a one-of sold in low volume to address specialized business needs. Most commercial GNU/Linux distributions up until recently were based on the pure product commodity price model.

2. **Dual License Free Software Product.** In this model the Free Software work is distributed at no cost under the terms of the GPL while it is sold under a different license to customers who do not want to be bound by the terms of the GPL. This model is available only to the copyright holders of the software. In addition it only makes sense when the software sold is included, in whole or in part, in another application that is sold by the customer under terms different from the GPL. If the customers were to include in their own work Free Software licensed according to the GPL, the customer application would be considered a derivative

work of the Free Software thereby forbidding redistribution or requiring that the application be licensed under the GPL. MySQL, a database, and Cygwin a UNIX-like environment for Windows are sold this way. This is an interesting illustration of the fact that the copyright holder can distribute the same version of a software work with different licensing conditions.

3. **Pure Service.** In this model a service company builds expertise around a number of Free Software works and helps its clients in the installation, use, maintenance, upgrade and any adjustments or customizations that the clients require.

4. **Leveraged Service.** A leveraged service is an expertise-based service that leverages on a Free Software product. This product/service combination is typically sold as a yearly subscription consisting of the Free Software product with upgrades and high-quality product support and online consulting from the product specialists. GNAT Pro, the Free Software development environment for the Ada programming language is one such example. This business model will be described in more detail in the following section.

5. **Free Software Co-op.** This business model stimulated by the work of the Network for Dependability Engineering [3] is described in a forthcoming paper [4].

GNAT Pro

In the early 1990s work was underway to revise the original Ada 1983 standard (known as Ada 83). Ada 83 is the programming language designed for safety and mission critical applications, especially embedded applications. Over the years it has built an impressive track record in this field. The revision of Ada 83, known as the Ada 9X project, gave rise in 1995 to Ada 95, the upwardly compatible version of Ada 83 that incorporates object-oriented and real-time programming features for native and embedded systems.

As a result of the Ada 9X effort the Ada group founded in 1980 at New York University received a grant to develop an Ada 95 compiler to be made available as Free Software. This effort led to the birth of the GNAT technology and subsequently gave rise to two sister companies: Ada Core Technologies in the United States and ACT Europe in Europe which will be referred to as AdaCore in the remainder of this paper.

Today AdaCore provides native, embedded, and safety-critical Ada and mixed-language software development solutions for a large variety of systems. Our offering is based on GNAT Pro, which has become over the last decade the leading Ada development environment.

Developing, maintaining, and marketing GNAT Pro for ten years have provided significant experience with both technical and non-technical aspects of Free Software. In the previous sections we have reviewed several key legal aspects. We summarize below some of the business lessons learned.

- On Free Software and university projects. When the GNAT project was funded at New York University there was a specific clause that required New York University to make the resulting software work available as Free Software. Had that provision not being there, GNAT would probably be sitting on some university shelf in New York.

- On public releases. Although it does not have to, AdaCore makes public releases of its GNAT technology. Making the GNAT technology publicly available was important in gaining mind-share and has fostered a user community around it. Furthermore this has allowed AdaCore to attract a pool of talented engineers on both sides of the Atlantic, the vast majority of whom knew GNAT before joining.

- On the GPL. Free Software compilers represent an interesting situation for developers of non Free Software. On the one hand a compiler is a standard development tool that is not meant to be included in the final application, while on the other hand a compiler comes with libraries and runtimes that are linked, and hence included in the final executable. To ensure that programs generated by GNAT Pro can be made proprietary, or classified as secret, or be subject to any other restrictions that customers deem appropriate, AdaCore is licensing the GNAT Pro libraries and runtimes with a special version of the GPL, called the GNAT Modified GPL (GMGPL) that explicitly allows links with non Free Software code.

- On selling GNAT Pro. GNAT Pro is sold as a yearly subscription. A subscription provides customers with:
    1. The GPL license for GNAT Pro and the GMGPL license for the runtime and libraries to guarantee that programs generated by GNAT Pro can be distributed under customer-specific terms and conditions;
    2. The GNAT Pro development environment for the selected application domains and platforms;
    3. GNAT Pro tool support directly from the GNAT Pro developers;

4. Online consulting directly from the GNAT Pro and Ada experts.

- On aligning with customer interests. The subscription-based model of GNAT Pro aligns the customer interests with our own. The customer is interested in receiving a high-quality product with a high quality service to develop its application, often a safety, mission, or business critical one. We are interested in making sure that our customer succeeds. Not only does this bring us engineering pride it also ensures that the customer renews its subscription year after year and keeps using our technology and services on the projects to come. As a result the engineering and financial interests of our customers are aligned with our own.

- On subscriptions and recurrent revenue. It is gratifying to see that many players in the software industry are discovering the benefits of subscription-based models (as opposed to pay-up-front). When properly implemented subscription-based models generate a predictable and recurrent revenue stream. For companies traded on the stock exchange this means predictable earnings per share, quarter after quarter.

- On the benefits of a leveraged service. GNAT Pro customers receive a high-quality Free Software product with a high-quality service from the Ada experts and the GNAT Pro developers. This allows customers to develop sophisticated systems where they can leverage on GNAT Pro and its software components as if they had developed it themselves since they have direct access to the sources and the GNAT Pro developers. The GNAT Pro experts serve as partners to the customer development team. They work closely with the customer team to help them make optimal use of our products and to assist them with all aspects of software development. As part of that dialogue our engineers regularly address issues such as code optimization, programming language semantics, multi-language systems, and code organization. The net result for the project is reduced risk, higher productivity, and shorter time to delivery.

- On trusting customers. The price of the yearly GNAT Pro subscriptions is based on the number of developers in the teams using GNAT Pro. To select the appropriate GNAT Pro subscription level AdaCore offers several predefined team sizes based on the maximum number of developers covered by the GNAT Pro subscription: up to 5, 15, 25, 35, 50, 100, 200, and 500 developers overall. These different team sizes allow for easy personnel adjustments during the course of a GNAT Pro subscription year. How do we check that users do not exceed the subscription level they purchased? We don't, we trust our customers. Trust builds trust.

Conclusion: Free Software and Sustainable Development

Beyond the legal and business aspects of this paper, I would like to conclude on the benefits that the freedom of Free Software licensing provides in the realm of sustainable development.

Firstly, support and maintenance of Free Software can be bone by organizations other than its copyright holder. This means that if, for some reason, the latter no longer provides satisfactory service the customer can select a different provider or train an internal team to achieve this task. This is especially relevant in industries where long-term projects may outlive the maintenance lifetime or even the provider of the target operating system, libraries, and development tools.

Finally, Free Software allows the creation of a local pool of software expertise and enables every region or country whether in Africa, North or South America, Asia, Australia, and Europe to forgo electronic colonization.

References

1. Visit http://www.fsf.org for more information on the Free Software Foundation and the GNU project.

2. http://www.gnu.org/copyleft/gpl.html

3. http://www.ris.prd.fr Open-Source Software and Dependability working group. The final study is available in the book: "*Logiciel libre et sûreté de fonctionnement*", by Philippe David and Hélène Waeselynck. Lavoisier Publisher.

4. *Free Software Co-ops*, by Cyrille Comar, Robert Dewar, and Franco Gasperoni. Forthcoming.