



**HAL**  
open science

## FLOSS, COTS, and Safety: A Business Perspective

F Gasperoni

► **To cite this version:**

F Gasperoni. FLOSS, COTS, and Safety: A Business Perspective. Conference ERTS'06, Jan 2006, Toulouse, France. hal-02270495

**HAL Id: hal-02270495**

**<https://hal.science/hal-02270495>**

Submitted on 27 Aug 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# FLOSS, COTS, and Safety: A Business Perspective

F. Gasperoni

gasperoni@adacore.com  
AdaCore, 8 rue de Milan, 75009 Paris, France

**Abstract:** This paper discusses the relationship between COTS software (Commercial-off-the-shelf) and FLOSS (Freely Licensed Open Source Software) from a purely business perspective. The emphasis of this work is on safety-centric industries such as aerospace, automotive, and railways.

**Keywords:** COTS, Free Software, Open Source, FLOSS, safety-critical software, certification.

## 1. Introduction

After explaining the nuance between Free Software and Open Source Software, this paper shows that, contrary to popular belief, COTS software and FLOSS are two orthogonal and non-conflicting attributes of a software product since the first deals with commercial off-the-shelf availability while the second deals with licensing terms. COTS vendors have a continuum of choices between providing only restrictively licensed COTS, a mixture of restrictively licensed and FLOSS licensed COTS, or entirely FLOSS licensed COTS.

Furthermore, this paper demonstrates that the notion of software quality is a third orthogonal dimension to COTS and FLOSS. We also dispel the myth that all FLOSS is developed in bazaar-style.

This work shows that the availability of source code in FLOSS-licensed software is not its key advantage when it comes to safety-focused industries. Rather, it is the set of freedoms guaranteed by FLOSS that makes it an interesting alternative over restrictively licensed COTS.

More specifically, copyright grants a temporary monopoly on a software product. This monopoly may result in vendor lock-in for software changes, support, and certification material. This drawback has usually been overshadowed by COTS advantages. Our work shows that COTS software sold with a FLOSS license (COTS FLOSS) has all the advantages of restrictively licensed COTS without the vendor lock-in disadvantages.

This paper does not claim that COTS FLOSS is systematically superior to restrictively licensed COTS since the choice between two COTS products involves commercial and technical aspects in addition to legal ones. Our objective is to show that all other things being equal FLOSS-licensed COTS is always better than restrictively licensed COTS

because the absence of lock-in aligns the COTS vendor interests with the customer's.

This paper draws on over 10 years of working experience at AdaCore the commercial company behind GNAT Pro, the Free Software Ada 83, Ada 95, and now Ada 2005 technology used by the primary actors in safety focused industries such as aeronautics, military, railways, and space.

## 2. FLOSS

FLOSS (Freely Licensed Open Source Software) [1] is a recent term used to denote Free Software [2] and Open Source Software [3]. In Europe FLOSS is also used as the acronym of Free/Libre Open Source Software. The French and Spanish word "Libre" means free (as in freedom) and emphasizes the notion of liberty that is at the root of the FLOSS movement. For most people FLOSS means:

- Access to the sources;
- Existence of a community dedicated to the evolution of the software;
- Free-of-charge software availability.

While these attributes are sometimes true, this is not what FLOSS is all about. FLOSS is all about the terms and conditions of the software license.

### 2.1 Software License

Software, like literature, music, and cinematography, is protected by copyright and governed by the Berne convention [4]. Copyright controls the right to use, copy, distribute, modify, and make derived works of the software (whether in source or binary form). To undertake anyone of these activities, one must obtain a license from the copyright holder. Broadly speaking a software license is the legal document listing the things the software recipient is and is not allowed to do with the software.

Unless explicitly placed in the public domain, the use, copy, distribution, modification, etc. of all software whether FLOSS or otherwise, is governed by the laws of copyright. From this standpoint there is absolutely no difference between Microsoft Windows and GNU Linux. These are both copyrighted works. To be able to use them one must obtain a license from its copyright holders. This software license grants users certain rights and may

impose certain restrictions in the use, modification, and/or redistribution of the software.

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 along with its license.

## 2.2 Free Software and Open Source Software

At the root of FLOSS is the Free Software movement started by Richard Stallman in the mid 80s [2]. In 1998, with the emergence of the open and community-based GNU Linux project, which was licensed as Free Software, some members in this community started using the term Open Source Software instead because of the openness of development, the existence of an open community of developers that Free Software licenses enabled. Today Free Software and Open Source Software are separate movements.

The fundamental difference between the two movements is philosophical: Open Source Software is a development methodology (the bazaar) [5] while Free Software is all about freedom including, and this is the key point from a business perspective, freedom to market [6, 7].

## 2.3 FLOSS Licenses

Even though Free Software and Open Source Software are associated with different criteria for acceptable software licenses [8, 9], in practice, nearly all software meeting one definition also meets the other. More specifically, all Free Software (FS) licenses qualify as Open Source Software (OSS) licenses and most OSS licenses are also FS licenses, the one notable exception being the Reciprocal Public License [10].

The Reciprocal Public License is not a Free Software license because (a) it limits the price that can be charged for copies of the software, (b) it requires notification of the original copyright holder for publication of a modified version, and (c) requires publication of any modified version that an organization uses, even privately [11].

Because this paper focuses on the business aspects of FLOSS as applied to safety-related industries, we concentrate on the key freedoms promoted by Free Software and embodied in the vast majority of FLOSS licenses. These freedoms are:

- The freedom to run the software for any purpose;
- The freedom to redistribute the software;
- The freedom to create and distribute derived works;

- The freedom to market the original or derived versions of the software.

Access to the source code and build procedures is a consequence of the freedom to create derived works since without these FLOSS recipients are not free to create and market derived works.

The ability to create and redistribute derived works allows the creation of an open community of developers collaborating in the evolution of the software. This is a possible side-effect of FLOSS licenses, not a requirement and not a guarantee.

FLOSS licenses put no restrictions on distribution or redistribution fees. The recipient of FLOSS can sell/resell copies of the software at any price.

## 3. COTS and FLOSS

### 3.1 What is COTS?

COTS (Commercial Off-The-Shelf) [12], is the term used for systems which are manufactured commercially and then tailored for specific uses. A COTS product is sold to the general public in the course of normal business operations at prices based on list or market prices. COTS are in contrast to bespoke or custom-made systems that are produced solely for a specific customer.

In the software area the significance of COTS varies greatly. In office automation, for instance, tools such as word processors and spreadsheets require no customization prior to their use. In embedded devices, on the other hand, COTS operating systems or kernels require a great degree of customization: from creating or adapting the BSP (board support package) for the target hardware, to configuring certain OS modules such as the scheduler.

### 3.2 Advantages of COTS

Advantages of COTS include:

- Existence of a commercial organization that can be held liable for all the IPR (Intellectual Property Rights) issues regarding the software such as copyright, patents, etc.;
- Existence of a commercial support organization responsible for answering questions and fixing problems encountered in the use of the software;
- Availability of complete, easy-to-install software package;
- Sharing of the cost for the creation, industrialization, maintenance, fixing of, and evolution of the software among users having similar needs.

### 3.3 Evolution of COTS Software Licenses

As we explained in section 2.1 software falls under the laws of copyright. Copyright grants copyright holders a temporary monopoly on their work. The duration of this temporary monopoly varies from life plus 70 years for individuals to 70 year after the first publication in the case of corporations (95 years in the USA). After this duration the software work falls in the public domain.

Coming back to COTS, up until recently COTS software was marketed under very restrictive licenses which severely limited the user's ability to use and copy COTS software let alone make modifications or redistribute the software. These restrictive licenses created vendor lock-in, making the temporary monopoly on a software work very apparent.

Because waiting 70 or 95 years for the software work to fall in the public domain was not a viable alternative and in any case the potential lack of source code would have made this event irrelevant, Richard Stallman launched the notion of Free Software in the mid 80s. This started as an ideological movement with the creation in 1985 of the Free Software Foundation. Thus, initially, COTS software was opposed to Free Software in people's minds since the first had to be purchased and had a restrictive license while the latter was available free-of-charge and came with a freer license.

This changed with the creation of corporations such as Red Hat in 1993 [13] and AdaCore in 1994 [14]. Whether it's GNU Linux or the GNAT Pro Ada 2005 development environment these companies market COTS software licensed under a FLOSS license. Thus, when looking for software two orthogonal aspects must be distinguished:

- (a) Whether the software is COTS, i.e. commercially available off-the-shelf;
- (b) The terms and conditions of the software license (restrictive or FLOSS).

### 3.4 A Remark on COTS and Licenses

It is important to understand that a COTS vendor owning the copyright of its software is under no obligation to:

- Give the same software license to all its customers: higher paying customers may obtain more favorable terms (e.g. a FLOSS license) while lower paying ones may receive a more restrictive license.
- A COTS vendor can provide certain of its products under a restrictive license, while marketing other products under a FLOSS license. This is the case, for instance of Wind

River [15], which markets various versions of VxWorks (restrictive license) and GNU Linux (FLOSS).

- The COTS vendor is under no obligation to make the sources/binaries available to all (e.g. by internet download) when the software is licensed as FLOSS. The source code must be provided only to recipients of the COTS software licensed as FLOSS.
- It is perfectly possible for a company to market patches for FLOSS packages that are not themselves under a FLOSS license. For example, a company may distribute GNU Linux and separately sell a patch adding industry-specific functionalities for the GNU Linux kernel under a restrictive software license. The entity that receives the restrictively licensed patch can insert it into GNU Linux. This entity can use the resulting program internally but cannot, however, redistribute it.

The point of the above remarks is to show that COTS vendors have a continuum of choices between providing only restrictively licensed COTS, a mixture of restrictively licensed and FLOSS licensed COTS, or entirely FLOSS licensed COTS.

## 4. COTS FLOSS and Safety

We come to the core question of this paper: what are the advantages, if any, of COTS software licensed under a FLOSS license in the area of safety-related industries such as the aeronautics, automotive, military, railway, or space industries. The statements made below, except for issues dealing with certification, apply to other industries as well.

### 4.1 Software Quality and Software Licenses

Software tools, libraries, and components used in safety-related systems benefit from careful planning and quality assurance procedures. They often must be part of an auditable and repeatable process with stringent quality requirements and a clear set of software development artifacts. In the avionics industry, for instance, the DO-178B certification protocol [17] stipulates that the contribution of software to potential failure conditions is used to define 5 software levels. Each software level imposes specific requirements on software planning, development, verification, configuration management and quality assurance. The stringency of these requirements increases with higher safety levels.

What are the quality guarantees of FLOSS-licensed COTS and restrictively licensed COTS? Answer: absolutely none (unless explicit evidence, such as certification material, is produced to the contrary).

With restrictively licensed COTS customers may not even have access to the sources. Even when customers have access to source code and build procedures no license (FLOSS or otherwise) requires the availability or redistribution of design documents, development plans, quality assurance procedures, test suites, or other software development artifacts.

The quality and quality-oriented procedures vary greatly from one COTS to another and the software license gives no guarantees in this matter. Some FLOSS proponents claim that because FLOSS is developed by a community of dedicated developers this yields better software. There is no systematic evidence of this. This really depends on the project. Furthermore, not every FLOSS is developed in a bazaar style. GNAT Pro [14], for instance, is developed mainly following a cathedral model, and although the sources of all of its components are available to all, AdaCore is very careful from both an IPR and software quality standpoint when accepting contributions from customers or other users.

There is nothing that prevents FLOSS-licensed COTS from achieving high-quality and being used in safety-critical systems. For the GNAT Pro Ada compiler alone, for instance, AdaCore has accumulated more than 10,000 tests in 12 million lines of Ada code. Furthermore, GNAT Pro in its High-Integrity Edition is used in several safety-critical projects in the avionics and railway industries (Airbus A380, 767 Tanker, C130AMP, Boeing 787 ...). Finally, GNAT Pro comes with an Ada runtime certifiable to DO-178B Level A.

#### 4.2 Drawbacks of Restrictively Licensed COTS

COTS software that comes with a restrictive license has a number of drawbacks. Up until recently, because the industry's only choices were between do-it-yourself and restrictively licensed COTS, COTS downsides were considered as inevitable.

These downsides stemmed from the fact that the customer did not own the copyright of the COTS software and had to abide by the restrictive rules set by the COTS owner. Once a restrictively licensed COTS was chosen and deployed the monopoly conferred by copyright to its owner was used to create:

- Vendor lock-in for software changes and evolution;
- Vendor lock in for support;
- Vendor lock-in for certification material (in safety-related industries).

The above drawbacks do not include absence of source code. In fact, thanks to the pressure exerted by the appearance of FLOSS, it is now possible to

obtain the sources of certain restrictively licensed COTS against a modest fee. The difference of this source code with that of COTS FLOSS resides in the restrictions attached to the former. In particular, and unlike COTS FLOSS, customers of restrictively licensed COTS cannot use its sources to free themselves of vendor lock-in.

#### 4.3 COTS FLOSS: No Vendor Lock-in for Software Evolution

Because COTS software is designed to meet the needs of a majority of disparate users there is a possible compromise on features, performance, and usability. In safety-related industries, for instance, a customer may have the choice between a vendor offering a POSIX-only API from one embedded operating system vendor and another offering only an ARINC 653 one [16].

Restrictively licensed COTS forces customers to go to their original COTS vendor for software evolution. Because the COTS vendor has a monopoly over changes to its software this can be arbitrarily expensive. This is not much of an issue in office automation and similar software, it is, however, an issue in safety related industries where the COTS software is used in very disparate hardware environments and the features required can vary greatly between integrated modular avionics and railway systems.

With FLOSS the customer is not hostage of its COTS vendor. Because FLOSS licenses guarantee a free market, FLOSS customers can solicit competitive bids when software modifications, additional tools, or extra libraries are needed. This is attractive for business and particularly so in safety-focused industries where the vendor lock-in is significantly more pronounced because of the heavy engineering investments required.

Taking GNU Linux as an example there are today several reliable COTS vendors for it, some making available specialized versions for the embedded market. Because of the lack of vendor lock-in, a customer that has committed significant resources to the use of the GNU Linux kernel in its embedded software infrastructure can switch COTS vendors or can occasionally use additional vendors to add industry-specific features. In the avionics industry a customer could obtain GNU Linux from one COTS vendor and have a separate company produce an ARINC 653 conformant API layer on top of it.

In the case of GNAT Pro, one of AdaCore customers (EUROCONTROL) needed an additional tool. To be economical this tool had to leverage on the guts of the GNAT Pro compiler to avoid redoing a parser and a semantic analyzer for Ada. To show upper management that COTS FLOSS has no vendor lock-

in the customer did an open bid for the additional tool which was won by a small and very capable French firm.

As another example, there was a segment in our industry that we (AdaCore) were not serving. An independent company used the GNAT Ada technology to build and market a COTS product for that market segment. We have since entered that market segment and are now competing with our own technology.

#### 4.4 COTS FLOSS: No Vendor Lock-in for Support

The source code, scripts, and build procedures of restrictively licensed COTS software cannot be given by a customer to third-party service providers to obtain higher levels of support whether in the form of training, consulting, or professional services.

Conventional wisdom tells us the COTS vendor knows best because it has developed the technology. More often than not, however, the development team of the COTS vendor is segregated from its services organization and is often shielded from the final customer.

The possibility to have support organizations separate from the original COTS vendor allows the creation of a cottage industry which may be in a better position to provide localized services to medium and smaller size customers.

Note that for strategic reasons a number of COTS vendors selling restrictively licensed software have decided to unbundle or outsource part of the service business inherently linked with their products. Some (e.g. Microsoft) have done this to create and foster an ecosystem of partners around their products.

Again, the difference with COTS software licensed as FLOSS, is market freedom. With restrictively licensed COTS it is the COTS vendor that solely decides its support strategy. With COTS FLOSS it is the market.

The key point here is not that production of software and its support should be systematically unbundled, rather it is the customer that should have the freedom to decide so when a business case exists.

#### 4.5 COTS FLOSS: No Vendor Lock-in for Certification Material

COTS software is designed to meet the needs of a majority of disparate users. In safety-related industries, certification material may be available only for certain certification protocols such as DO-178B [17] or not available at all.

In the case of restrictively licensed COTS it is again the vendor that has a monopoly in what to do. It is possible for a customer to provide certification material for restrictively licensed COTS. This has happened for instance at organizations such as Honeywell Aerospace who internally certified their VxWorks-based Global Star 2100 Flight Management System [18], or at Rockwell Collins with LynxOS for the adaptive flight display architecture on the Bombardier Challenger 300 [19].

Without special arrangements with the original COTS vendor this certification material may not be shared with partners or subcontractors.

With COTS licensed as FLOSS, the customer can contract with an independent provider for certification materials and audit trails. Incidentally, in this case the customer may decide to own the copyright for the certification material which he may provide to its partners, subcontractors or perhaps resell. Alternatively, an industry segment may decide to pool its resources to create certification evidence for kernels such as GNU Linux.

The key point here is that quality, when present, can be reversed engineered. Indeed, Verocel [20], the certification company AdaCore has partnered with to certify its Ada runtime for VxWorks 653 [21], starts from the final code to produce audit trails and certification evidence.

Once more the difference between restrictively licensed COTS and COTS licensed as FLOSS is market freedom. With the former the COTS vendor is the only one deciding who can produce certification evidence for its software. With COTS FLOSS it is the market.

## 5. Conclusion

In the previous sections we have shown how COTS FLOSS is unaffected by the downsides (for customers) of restrictively licensed COTS. COTS vendors have a continuum of choices between providing only restrictively licensed COTS, a mixture of restrictively licensed and FLOSS licensed COTS, or entirely FLOSS licensed COTS.

Our objective is not to show that COTS FLOSS is systematically superior to restrictively licensed COTS since this choice involves commercial and technical aspects in addition to legal ones.

Our objective has been to show that all other things being equal a FLOSS-licensed COTS is always better than a restrictively licensed one because the lack of vendor lock-in aligns the COTS vendor interests with the customer's. Put it another way, to the question: should an industry segment lobby with its favorite vendor(s) so that, when it is of value, they

make their COTS software available to them with a FLOSS license the answer is a definite yes.

## 6. References

- [1] <http://www.infonomics.nl/FLOSS/>
- [2] <http://www.fsf.org/>
- [3] <http://www.opensource.org/>
- [4] Berne Convention for the Protection of Literary and Artistic Works, available at [http://www.wipo.int/treaties/en/ip/berne/trtdocs\\_wo001.html](http://www.wipo.int/treaties/en/ip/berne/trtdocs_wo001.html)
- [5] The Cathedral & the Bazaar, by Eric S. Raymond, published by O'Reilly in 1999.
- [6] <http://www.fsf.org/licensing/essays/selling.html>
- [7] Panelists consider the 'business case' for open source, Infoworld, October 1, 2002, <http://www.infoworld.com/articles/hn/xml/02/10/01/021001hnbizopen.html?s=IDGNS>
- [8] <http://www.fsf.org/licensing/essays/free-sw.html>
- [9] <http://www.opensource.org/docs/definition.php>
- [10] <http://www.opensource.org/licenses/rpl.php>
- [11] [http://www.fsf.org/licensing/licenses/index\\_html#NonFreeSoftwareLicense](http://www.fsf.org/licensing/licenses/index_html#NonFreeSoftwareLicense)
- [12] <http://www.google.com/search?hl=en&lr=&oi=define&q=define:COTS>
- [13] <http://www.redhat.com/>
- [14] <http://www.adacore.com/>
- [15] <http://www.windriver.com/>
- [16] Avionics Application Software Standard Interface, ARINC Specification 653, <http://www.arinc.com/>
- [17] DO-178B, Software Considerations in Airborne Systems and Equipment Certification, by RTCA. See <http://www.rtca.org/>.
- [18] <http://europe.windriver.com/literature/articles/Safety%20Critical%20Avionics%20Software%20Development.pdf>
- [19] <http://www.businesswire.com/webbox/bw.092203/232655485.htm>
- [20] <http://www.verocel.com/>
- [21] <http://www.windriver.com/products/product-overviews/Platform-Safety-Critical.pdf>.