

# How to Succeed in the Software Business While Giving Away the Source Code

## **The AdaCore Experience**

**Benjamin M. Brosgol, AdaCore**

©2019 IEEE

*IEEE Software*, Nov.-Dec. 2019, pp. 17-22

Special Issue: Twenty Years of Open Source Software

<https://ieeexplore.ieee.org/document/8792964>

Open-source software, or, more accurately, Freely Licensed Open-Source Software (“FLOSS”), at first appears to present a dilemma when adopted as part of a business model. If users are allowed to access, modify and/or redistribute the source code, how does a company protect its intellectual property, and more fundamentally, sell something that can be easily and legally reproduced?

AdaCore has faced this issue since the company's inception in 1994. Its major commercial product, GNAT Pro Ada, is an Ada development environment based on the Free Software Foundation's (FSF's) GNU Compiler Collection (GCC). AdaCore has implemented an Ada compiler front end and companion run-time libraries and tools and has contributed these components to the FSF. In turn, the GNAT Pro Ada compiler incorporates the GCC back end for a variety of target architectures. Leveraging the GCC back end has enabled AdaCore to make Ada available on a wide range of platforms, both native and embedded, at a significantly reduced effort – indeed, that was the technical rationale for choosing GCC and a design goal of GCC itself. But the challenge of this approach is how to generate a sustained and profitable business. AdaCore's 25 years of FLOSS experience offers an explanation and “lessons learned”.

## **Introduction**

A software company faces many challenges in realizing a profitable and growing business. Beyond the various technical, marketing and sales issues lies a fundamental question concerning the business model: Since software is relatively simple to replicate, how does a company protect its investment – i.e., prevent unauthorized uses of its product – without inconveniencing or penalizing customers who use the product legitimately? And a related question about the “bottom line”: How can a company realize a sufficient revenue stream to be profitable and fund ongoing product enhancements?

The answers are influenced by the company's choice between a proprietary and a FLOSS approach. AdaCore has adopted a FLOSS model, and this article describes the rationale for this decision and the company's experience with FLOSS licensing for its major product line, the GNAT Ada development toolset. The article describes the relationship between AdaCore and the FLOSS developer community (principally the FSF), identifies issues that have arisen

and explains how they have been addressed, and shows how the FLOSS approach has helped AdaCore sustain a growing and profitable revenue stream over its 25-year history.

## Early Decisions

In the early 1990s a team from New York University was awarded a government contract to develop a compiler for the language known as Ada 9X (later renamed Ada 95 [1]), a major revision to the original Ada 83 standard. The goal: a user-friendly compiler available to academia at no cost, on multiple platforms. Following discussions with Richard Stallman from the FSF, the project adopted the GCC technology [2] for the compiler back end, with GDB for debugging support, and added an Ada-specific front end and run-time libraries. The resulting toolset was somewhat whimsically named “GNAT”, which stood for “GNU NYU Ada Translator”. (The acronym expansion has long been abandoned, but the GNAT name has persisted.) Since the GNAT compiler contained GNU software and was intended for teaching and research, the standard GPL licensing was appropriate. This made the GNAT source code available and prevented any of the components from being used in proprietary software.

GNAT was not a production-quality compiler, but the project leaders recognized the commercial potential for a professional Ada 95 development environment and founded AdaCore (then Ada Core Technologies) in 1994 to productize the GNAT technology. Two years later ACT-Europe was founded in France, and the current AdaCore is a result of the subsequent unification of the two companies.

As explained by Jullien and Zimmermann [3], a company can take a variety of roles concerning its usage of FLOSS software, and the prospective customer base may exhibit a range of skillsets. For AdaCore, the FLOSS components have always been a core asset in GNAT, and the company staff includes experts in both the GCC/GDB technology and Ada language semantics / implementation. The customer base for the commercial GNAT offering initially comprised two sectors: hardware vendors who needed an Ada 95 compilation system, and software development teams who wanted to use Ada 95. Both sectors’ personnel were highly skilled, and they contributed to the technology not by generating new FLOSS software but by expressing requirements that would drive product enhancements to be implemented by the AdaCore team.

GPL licensing is appropriate for the compiler but not for the run-time libraries, since customers may need to develop and distribute proprietary or even classified applications whose source code has to stay hidden. As explained below, a variation of the GPL license was adopted for the run-time libraries.

## Why FLOSS?

FLOSS is sometimes touted as being more secure than proprietary software (“given enough eyeballs, all bugs are shallow” [4]) but the opposite claim has also been made (“security through obscurity” [5]). In fact, high-profile vulnerabilities have materialized in both types. AdaCore’s adoption of a FLOSS approach was motivated not by considerations of software quality – an effective software lifecycle management infrastructure is a much more important contributor – but rather for other reasons:

- AdaCore’s founders shared the FLOSS philosophy that openly available source code can help the software community advance.
- A production-quality Ada compiler technology targetable to a variety of processors was a major goal from the outset, and the GCC technology offered an effective solution.
- The company founders perceived that making product source code available to customers, and contributing components to the FSF, were not endangering its business. Compiler construction is a specialized field, the Ada market is a relatively narrow niche, and the expertise needed to package the FLOSS components into a commercial product (and then to provide the necessary technical support) is high. The risk of the source code being used by potential competitors would be low.

## Product Evolution

From the outset, AdaCore envisioned two user communities for its GNAT technology: individuals or academic institutions adopting Ada for FLOSS software development or teaching, and companies or other organizations using Ada to implement professional-grade software for commercial or government-sponsored projects. The version of the GNAT technology for FLOSS developers and academia has helped Ada gain traction in those communities; it was originally named the “public version”, later the “GPL version”, and most recently the “Community Edition” [6]. It consists of a no-cost downloadable executable (and source code) for the compiler and accompanying tools / libraries, available on a number of platforms, with licensing appropriate for academic usage and free software development.

The professional version, known as *GNAT Pro Ada* [7], shares a common code base with the Community Edition but has several major differences:

- Customers gain access to the toolset through an annual paid subscription, with licensing appropriate for software developers who do not want, or are not allowed, to distribute their source code along with the executable.
- It comes with product support that includes guaranteed rapid response to questions and defect reports, with access to wavefronts (specially generated interim releases) if

needed to correct critical problems. Support is provided by the product developers themselves.

- It undergoes a more extensive QA regimen, with nightly regression testing (millions of lines of test cases) on many platforms.
- Some specialized tools are only available with GNAT Pro.

Both the Community Edition and GNAT Pro have at least one major release each year, allowing all users to keep up to date with product enhancements, and, for GNAT Pro, serving as an incentive for customers to renew their subscriptions. FLOSS helps here, since the GNAT technology can take advantage of GCC improvements such as new code generators and back end optimizations. But recognizing that some customers need to standardize on a specific version of the toolset rather than upgrading each year (for example when their application requires safety certification), AdaCore supplies a specialized product edition known as *GNAT Pro Assurance*. This version provides support for a designated version of the toolset and includes maintenance of a branch of the GNAT Pro codebase, with defect analysis and correction.

## AdaCore and the FLOSS community

AdaCore's has always had a tightly coupled relationship with the FLOSS community, pulling periodic updates of the GCC and GDB software from the FSF and, in the other direction, pushing new versions of the Ada-specific components to the FSF. To properly control these interactions, and more generally to manage the production and release of multiple products on dozens of platforms, AdaCore's development and verification environment has always been highly structured, e.g., with regression tests run at check-in and with extensive QA before product release. The production process is much more like the "cathedral" than the "bazaar" [4], or, in Fitzgerald's terminology [8], more like the 2<sup>nd</sup> generation of open source software (OSS 2.0) than the original model (FOSS, or "Free and Open Source Software").

For AdaCore, the FLOSS components have always been a core asset in GNAT.

In addition to contributing its Ada technology to the FSF, AdaCore also makes a variety of tools and libraries available in github repositories where they are open to community enhancements and analysis. GNAT is by far the dominant technology in the Ada ecosystem – for example it is the only environment that fully supports the latest (Ada 2012) [9] version of the language – and as such is a *de facto* standard implementation. Free software developers working in Ada gravitate naturally to GNAT as the technology to contribute to.

## Business Model and Licensing

When AdaCore was founded, contracts from Silicon Graphics (to port GNAT to IRIX) and from DEC (to port GNAT to Alpha VMS) helped fund the initial development and productization, but these were “one-off” projects. A different source of revenue would be needed to support continued growth and technology enhancements. One approach, common for proprietary software, is to charge a large fee initially and perhaps additional fees for run-time licenses or support. However, this typically results in an uneven (and unpredictable) revenue stream, and charging for run-time licenses would put Ada at a competitive disadvantage since other languages do not impose such fees. Instead, AdaCore’s business model is based on an annual subscription that reflects the value that the company adds to its FLOSS software: an expert level of support coupled with assurance about the software’s licensing status. The subscription model has the benefit of revenue predictability and also incentivizes AdaCore to provide product improvements that encourage customers to renew. There are no run-time license fees.

The GNAT Pro product includes two kinds of software:

- a compiler and companion tools that users invoke to develop and verify their code and generate object modules; and
- run-time support libraries that are linked with the object modules to form the final executable.

Different licensing is appropriate for the two cases. The compiler and tools are covered by the GNU General Purpose License (GPL), more specifically GPLv3 [10]. Under this license, if a user builds the compiler or tool source code (either modified or unchanged), and then distributes a binary version of the result, they also have to make the resulting source code available under the same (GPLv3) terms. This scenario comes up often in practice (for example, Red Hat’s distribution of the GNAT Ada toolset from the FSF repository) and helps ensure that FLOSS software stays both freely licensed and open.

AdaCore selected the GPL over other FLOSS licenses because software licensed under the GPL cannot be made part of proprietary software and will always be freely available. In addition, software distributed under GPLv3 cannot be used in hardware that forbids other software from running on that hardware (e.g., by requiring the software to be cryptologically signed).

The run-time library situation, however, is different. Run-time libraries are linked with object modules from user code, and if the libraries are covered by a GPL license then a user distributing an executable will need to make the source available for their own components. But GNAT Pro is intended for users who may need to develop software that can only be

distributed in binary form. The run-time libraries for GNAT Pro (like those for GCC) therefore carry an exception to the GPL:

*[You] can freely distribute your programs built with the GNAT Pro compiler, including any required library run-time units, using any licensing terms of your choosing.*

This does not apply to the GNAT Community Edition, whose run-time libraries are covered by the GPL without any special exception. Since the Community Edition is intended for free software developers and academic users, the GPL licensing is appropriate.

## **Intellectual Property**

Although AdaCore is a software product company, it regards its intellectual property (“IP”) as being embodied not in its technology *per se* but rather in its development, verification and QA processes / infrastructure. AdaCore’s policy of making all of its product source code available with permissive licensing thus helps rather than threatens the company’s business: it is in AdaCore’s interest if users are familiar with how the technology works, and in fact having the run-time library code accessible and modifiable can be critical in some domains (for example real-time embedded systems). However, although the company’s source code is made openly available, its QA tests are not, since many of these tests are proprietary code submitted by AdaCore’s customers.

A basic question, noted in the Introduction, is how a company can protect its investment – i.e., prevent unauthorized uses of its product – without inconveniencing or penalizing customers who are using the product legitimately. AdaCore addresses this issue in several ways:

- A subscription is based on the number of users, and customers register the personnel who are authorized to download the software, send reports/questions to AdaCore support, etc.
- The product’s licensing guarantees are only assured during the duration of the customer’s subscription, incentivizing the customer to renew their subscription if they intend to continue using the product.
- There are no run-time license fees, product locks, or other intrusive mechanisms.

## **Other FLOSS Companies**

Other software companies besides AdaCore have adopted a FLOSS approach, with Red Hat serving as one of the best-known examples. They started in the mid-1990s with a Linux-based product offering, acquired the FLOSS-based company Cygnus Solutions in 2000 (which had

offered support and development services for the FSF software-development toolchain including GCC and GDB), and more recently have expanded to encompass a wide range of enterprise software [11].

FLOSS has brought different kinds of benefits to Red Hat and AdaCore. It has allowed Red Hat to offer a price-competitive alternative to the proprietary operating systems prevalent in the marketplace (most notably Windows), whereas for AdaCore it has provided a technology resource (GCC) for the company's compiler development. The markets for the companies' products are also rather different, with AdaCore's being a smaller and more specialized niche.

Notwithstanding these differences, the companies have succeeded with FLOSS through a similar subscription-based business model. They make money from free software by adding value through support and related services -- either for a broad set of tools (Red Hat) or in a market niche where users are comfortable paying for support (AdaCore) -- and each actively coordinates with its FLOSS developer community as both a contributor and a consumer.

## **Issues and “Lessons Learned”**

Coordination among different developers, and integration of components into a coherent product, are challenges for any software development project but are perhaps more acute when FLOSS components are involved. A specific issue for GNAT is when to synchronize with the FSF on incorporating new versions of GCC into the compiler, and, in the other direction, when to commit new versions of the Ada front end, tools and libraries. AdaCore has chosen to be selective with how often it incorporates new versions of GCC, because each upgrade requires significant QA work and often some minor changes to GNAT. On the other hand, the company has pushed its code to the FSF as time has allowed, as long as GCC was in a development stage where changes were permitted.

Another potential issue is how well FLOSS software satisfies customer requirements (reliability, performance, etc.), since, unlike typical projects that start with requirements analysis, FLOSS components are generally developed “bottom up” by contributors who do not necessarily know how the software will be reused. In the case of AdaCore's use of GCC, this has not been a problem: the GCC back end has served well as a retargetable optimizing compiler technology, with sufficient coverage of processors and operating systems to support AdaCore's needs. On the other hand, the standard FSF format for documentation, Texinfo [12], proved to be somewhat fragile and in general did not provide a modern “look and feel”. This issue has been largely ameliorated with the transition to Sphinx [13]. Also, the GCC technology dates back to an era when tools were invoked only through the command line, but modern development teams typically prefer a graphical Integrated Development Environment (IDE).



As a result, AdaCore has developed an IDE, originally known as the *GNAT Programming Studio* (GPS) and more recently rebranded as *GNAT Studio*, that has special support for the GNAT technology.

AdaCore's experience as a FLOSS user and contributor have produced several "lessons learned":

- The annual subscription model works well for a customer base that is developing critical software and therefore may need the added assurance that comes with expert and timely support.
- Since support has to be provided for the external FLOSS software (GCC and GDB, in the case of GNAT), the company needs to have internal expertise on these components.
- Synchronization with the FSF must be carefully coordinated (when to upgrade to a new version of GCC, when to commit a new version of the Ada components) to avoid disruptions.

In summary, freely licensed open source software can save development cost and be the basis of a profitable business. Its value is not strictly in the product IP but rather in the full complement of processes and the assurances they offer. The key is to provide a high level of support, as well as continued innovation, so that customers are encouraged to continue renewing their subscriptions. AdaCore has traditionally seen a renewal rate of more than 90%, so the company's experience is direct evidence that FLOSS can and does work.

## About the Author



Dr. Benjamin Brosgol is a member of the senior technical staff of AdaCore. He has been involved with programming language design and implementation throughout his career, concentrating on languages and technologies for high-assurance software. Dr. Brosgol was a member of the design team for Ada 95, and he has also served in the Expert Groups for several Java Specification Requests. Dr. Brosgol holds a BA in

Mathematics from Amherst College, and MS and PhD degrees in Applied Mathematics from Harvard University.

# Acknowledgments

The author wishes to thank Richard Kenner from AdaCore for his insightful contributions to this article, and also the anonymous reviewers for their many helpful comments.

## References

1. ISO/IEC, *Ada Reference Manual, ISO/IEC 8652:1995, Language and Standard Libraries*.  
<https://www.adaic.org/ada-resources/standards/ada-95-documents/lrm-original/>
2. Free Software Foundation, *GCC, the GNU Compiler Collection*.  
<https://gcc.gnu.org/>
3. N. Jullien and J.-B. Zimmermann, "FLOSS firms, users and communities: a viable match?" *Journal of Innovation Economics & Management*, 2011/1 (n°7), pp. 31-53.  
<https://www.cairn.info/revue-journal-of-innovation-economics-2011-1-page-31.htm#>
4. E. Raymond, *The Cathedral and the Bazaar*, O'Reilly Media, 2008.
5. A.D. Campbell, *Security vs Security Through Obscurity*, August 4, 2015.  
<https://ithemes.com/security-vs-security-through-obscurity/>
6. AdaCore, *GNAT Community Edition*.  
<https://www.adacore.com/community>
7. AdaCore, *GNAT Pro Edition*.  
<https://www.adacore.com/gnatpro>
8. B. Fitzgerald, "The Transformation of Open Source Software", *MIS Quarterly*, Vol. 30, No. 3, pp. 587-598, September 2006.  
[https://www.jstor.org/stable/25148740?seq=1#page\\_scan\\_tab\\_contents](https://www.jstor.org/stable/25148740?seq=1#page_scan_tab_contents)
9. ISO/IEC, *Ada Reference Manual, and Rationale*.  
<https://www.adaic.org/ada-resources/standards/ada12/>
10. Free Software Foundation, *GNU General Public License*.  
<https://www.gnu.org/licenses/gpl-3.0.en.html>
11. G. Satell, *How Red Hat Helped Make Open Source a Global Phenomenon*, January 6, 2019.  
<https://www.inc.com/greg-satell/how-red-hat-scaled-from-an-unlikely-startup-to-a-major-global-enterprise.html>
12. Free Software Foundation, *Texinfo – The GNU Documentation System*.  
<https://www.gnu.org/software/texinfo/>
13. G. Brandl, *Sphinx Python Documentation Generator*.  
<http://www.sphinx-doc.org/>