SSC19-WKVI-06

How, What, Where, and Why to Open Source Small Satellite Software

Ryan Melton
Ball Aerospace
1600 Commerce St, Boulder CO, 80301; 303-939-6771
rmelton@ball.com

Jason Thomas
Ball Aerospace
1600 Commerce St, Boulder CO, 80301; 303-533-5254
jmthomas@ball.com

ABSTRACT

How do you open source small satellite software? What are the expected costs, obstacles, and benefits? Why would you even want to release your software to the world? Is it safe to use open source software? Open source software like Ball Aerospace COSMOS fills an important need for SmallSat missions where spending the budget for a traditional ground system would be more than the full cost of the satellite. However, nothing is ever truly free and there are a lot of misunderstandings about open source software in the Aerospace industry. This presentation will answer those questions while discussing the specific choices that were made in open sourcing the Ball Aerospace COSMOS Command and Control System. Additionally, this presentation will discuss the adoption of COSMOS since it was open sourced 4 years ago and where it is going in the future.

INTRODUCTION

There are four words that you rarely hear together: "Aerospace Open Source Software". This is a shame, especially with the rapidly growing SmallSat ecosystem. There is a large potential for collaboration across aerospace companies to create best in class products that don't have to reinvent the wheel.

One of the best ways to enable this collaboration is by using open source software. Besides collaboration, there are numerous other reasons why open sourcing your aerospace software can benefit your company. This paper discusses those reasons and how to make it happen.

WHY OPEN SOURCE YOUR SOFTWARE

The first question asked to creators of open source software is usually "Why did you open source it?". There are many reasons for open sourcing software. It is especially important to know the answer to this question when convincing stakeholders within your company that open sourcing a potentially valuable piece of proprietary software is the right path forward.

Giving Back

The first, and most altruistic answer is to give back to the open source community from which you already benefit greatly from.

At this point in time, it is extremely hard to justify writing any piece of software of moderate complexity without making use of open source. Popular operating systems (Linux, FreeBSD), programming languages (Ruby, Python), databases (PostgreSQL), and support libraries (OpenGL, OpenCV) are all open source.

Open sourcing your own software allows it to reach the maximum audience, achieve maximum utility, and help others who have open sourced their software. If your management is not feeling as altruistic as you, there are also numerous financial opportunities to open sourcing.

Commercial Licensing

One of the potential methods to monetize open source software is through a dual licensing strategy that offers a commercial licensed version of the software on top of the open source license.

One of the common features of all open source licenses is the fact that they are all offered as-is, with no warranty. Customers can buy commercial licenses that may offer some form of warranty or offer different terms than the underlying open source license. For example, a piece of code offered under the open source GPL license could also be sold under different terms that did not require the source code to be shared if distributed to customers.

Training

Even with the best documentation, customers will always value real life training for software that they are going to be using every day. Offering training classes that teach the proper use of your open source software can be a great way to make money and build relationships with customers.

Support Contracts

Open source repositories such as Github offer ticketing systems that allow users to post bug reports and feature requests. However, these requests may sit for a while, or fall behind higher priority tasks for the projects. By offering paid support contracts, you can give open source software users the opportunity for direct access to developers, quick turnarounds for changes (either bug fixes or enhancements), and the ability to get quick answers to questions.

Community Contributions

One of the great features of open source software is it allows the community of users a way to contribute back to the project. Users can submit changes that include bug fixes, enhancements, and completely new features.

With a large enough community, a significant portion of the maintenance of the software can conceptually be provided from outside of your organization. This provides your company cost savings and a higher quality product.

Add-On Products

Another monetization opportunity is to sell add-on products that work alongside your open source product. Conceptionally, you can offer plugins, complementary software products, and even hardware that works directly with your open source software.

Follow-On Contracts

One of the hidden costs of open source software is that you still must learn how to use it, configure it, and set it up. These are costs of all software, open source or not, but often they are not considered.

Companies can take advantage of this by offering white glove services to fully configure the open source software for customers. You can also offer to build more featured solutions on top of the open source software.

HOW TO OPEN SOURCE YOUR SOFTWARE

The actual act of open sourcing your software is remarkably simple, but there are a lot of considerations that need to be made before you do. The following sections describe a basic process that can be followed to successfully open source your software.

Approval and Buy-in

Before open sourcing any of your company's software, you need to get approval and buy-in from someone with the authority to approve open sourcing. For most companies this will be someone at the vice president level, or C-level executive level. Be prepared to explain the benefits of open sourcing and to show financial projections. This is rarely a fast process, so also be prepared to be patient.

Export Control

For larger companies, you will likely have company export control staff that you can work with directly to make sure you are following the correct process to handle export control rules. For smaller companies, you should consult an export control lawyer and/or work directly with the state department (ITAR) and/or commerce department (EAR).

Understand and Choose a License

A software license describes how software may be used by others who don't own the copyright to the software. It is important to choose the right license to meet your goals. There are many open source licenses available, and you can even write your own!

Common open source licenses include the GPL, LGPL, MIT, BSD, and Apache licenses. The GPL licenses ensure that the software stays free by requiring any who distributes the code, or anything built on top of it, to also distribute the rights to do anything they want with the code. The LGPL license is very similar, except it does allow building on top of the open source code without requiring giving away rights to the final product.

The MIT and BSD licenses are the most permissive and can lead to the most adoption of software because they allow all uses of the software, including building proprietary software on top of it.

The Apache license is like the MIT and BSD licenses, but is more modern and also addresses the issue of software patents.

Contributor License Agreements

If your goal is to maintain complete control of an open source project, then it is very important to make contributors sign a contributor license agreement. These agreements typical require the contributor to give full rights to their contributions to the project, such that

the project is not hindered by mixed ownership of copyright.

Publish to the Internet

Once you have approval from management, have determined how you will license your code, made sure it is approved for export, and have a contributor license agreement put together you are finally ready to publish to the internet. Make sure that your code is clean and ready for public consumption, then post to an online repository. At the time of this writing, Github is the most popular online repository for open source software and is free for open source.

COSTS OF OPEN SOURCING

Other the costs of getting the code published, what are the ongoing costs of open sourcing software? Honestly, this can be as much, or as little as you want, but is mainly confined to the following activities.

Code Maintenance

Maintained code is successful code. On Github, one of the first statistics users see is how long has it been since the last change. Keeping your code maintained and up to date is crucial for its long-term success. Fortunately, this is typically a cost you would have had even if the code was not open sourced. Once open sourced, you now have the opportunity of having a community provide some of the work.

Building Awareness

Let's imagine you've put together an amazing piece of code that can solve everyone's problems and change the world. Great! Unfortunately, just posting it to the internet and assuming it will achieve mass adoption is a pipe dream. Even free products like open source software require marketing.

Recommended channels to get the word out include attending conferences, posting on relevant online forums, and even paid marketing. This can easily be the most expensive cost of open sourcing code, depending on your goals.

Tickets/Support

Once others start using your software, users will start submitting questions, bug reports, and new feature requests. This is a good thing! The more software is used, the more it gets rung out, and the higher quality it becomes. Being responsive to users provides a great positive for your product, but you must allocate time and staffing to respond to these requests.

Documentation Maintenance

Alongside the actual source code of any good open source product is also the documentation that goes along with it. This documentation includes user guides, frequently asked questions, examples, and configuration guides. Maintaining this documentation is a huge effort that should not be overlooked.

IS USING OPEN SOURCE SAFE?

The second most common question asked about open source, is "is it safe?". This question is typically asked due to the common but misplaced fear of malicious code being added to a code base because it is open source.

Trusted Sources

The most important aspect in using open source code is that it is maintained by a trusted company or group. Just because a software is open source does not mean that it is open for anyone to commit to it. Even when changes are proposed from the community, the maintainers of the software still get to review and approve all changes. This review process prevents malicious code from entering a codebase.

Risks

A greater risk with open source code is using older versions with known vulnerabilities, and the potential risks from an open source project's dependencies. To mitigate these risks, open source code should be kept up to date, and only open source with trusted dependencies should be used.

Summary

Just because code is open source does not mean it is open to changes from the world. Using trusted open source code is a safe solution with great benefits in cost savings and code quality.

ACTUALS FROM OPEN SOURCING COSMOS

Ball Aerospace COSMOS was first open sourced on December 29th, 2014. At the time of this writing, it has been downloaded 95,039 times according to rubygems.org.

Community Contributions

16 Github accounts have contributed to COSMOS. Of those accounts, 8 belong to Ball Aerospace employees, and the remaining 8 are from outside of the organization. 567 tickets have been written against COSMOS with, 517 closed and 50 currently open. Of those tickets 193 were written by Jason Thomas, and 194 were written by Ryan Melton, the two primary authors of COSMOS. The remaining 180 tickets have

been primarily written by the COSMOS user community outside of Ball.

Adoption

COSMOS has been used on over 60 programs within Ball Aerospace and at more than 50 known organizations outside of Ball. Users range from universities, government organizations, the large aerospace primes, Smallsat developers, and individuals. Many companies are using COSMOS for integration and test activities, and several are now operating satellites with it.

Contracts

Ball has sold numerous commercial licenses of COSMOS, held training for several companies, and won one large development contract to add functionality to COSMOS. Overall the open sourcing effort has been a success, and the company is building additional products around the open source version of COSMOS.

THE FUTURE OF BALL AEROSPACE COSMOS

The future of Ball Aerospace COSMOS is bright with wide adoption, and an increasingly good heritage story with use across the industry. Going forward, the next big change will be greatly improved cloud support and a web frontend to supplement and possibly replace the current desktop-based implementation.

CONCLUSIONS

Open sourcing your Smallsat software can have numerous benefits, both financially and for the entire community. The costs are not too high, and the rewards can be great. By following a simple process, any company large or small can reap the benefits of open sourcing software. Additionally, open source software is a safe and cost-effective option to build your systems upon.

References

1. *Melton, Ryan, "Ball Aerospace COSMOS" Retrieved from http://cosmosrb.com June 4th, 2019*
2. *Melton, Ryan, "COSMOS Source Code", Retrieved from https://github.com/BallAerospace/COSMOS June 4th,, 2019*
3. *"GNU General Public License" Retrieved from https://www.gnu.org/licenses/gpl-3.0.en.html June 4th, 2019*
4. *"GNU Lesser General Public License" Retrieved from https://www.gnu.org/copyleft/lesser.html June 4th, 2019*
5. *MIT License - Retrieved from https://opensource.org/licenses/MIT June 4th, 2019*
6. *BSD License - Retrieved from https://opensource.org/licenses/BSD-3-Clause June 4th, 2019*
7. *Apache License - Retrieved from http://www.apache.org/licenses/LICENSE-2.0 June 4th, 2019*