An Exploration of the Benefits and Drawbacks of Open-Source Projects

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Julia Friedman

Spring 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Richard D. Jacques, Department of Engineering and Society

Introduction

Open-source projects are particularly important and beneficial for developers and organizations alike. Open-source projects provide source code that is free and may be copied or altered by anyone (Corrado, 2009). All open-source projects can thus undergo perpetual enhancement by developers anywhere and at any time. Von Krogh and Spaeth (2007) present the advantages of the accessibility provided by open-source projects, including its value in promoting research. Essentially, it opens the door to an abundance of data for researchers and opportunities for developer improvement of code. Open-source also allows for continuous improvement of products and services, and for an open conversation about the functionality and enhancement of these software. A well-known example of an open-source product is the programming language, Python. Python is available for all users for free, allowing it to be easily distributed and universally accessible. A user can download any source code within Python and further modify and distribute their version of that code. Overall, open-source encourages collaboration, community, and transparency among developers.

Although open-source software has many benefits to its users, there can be drawbacks of implementing such software. Open-source code can compromise security of applications, lacks the level of service and support that comes with commercial software, and introduces a risk of orphan software (*Pros & Cons of Open Source in business*, 2021), among others. The motivation for my overall sociotechnical research is to uncover the benefits and drawbacks of open-source in order to determine how companies may decide what kind of software best fits their needs. Taking it a step further, my research will employ an ethical lens to examine the ethos of using open-source vs. proprietary (commercial) software.

Importance of This Open-Source Technical Project

My interest in open-source projects was piqued during an internship at IBM where I completed a project that involved an enhancement of the type checker for one of Styra's open-source products, Open Policy Agent (OPA). OPA is an open-source engine that unifies policy enforcement across the cloud native stack, providing a declarative language that allows for specification of policy as code. However, one of the limitations of its Rego type checker was the absence of the keyword "allOf." My technical work involved adding support for the implementation of this keyword, which enhanced OPA's type checker, as users now receive detailed error messages regarding the use of this keyword. The results of my technical project included a contribution into the OPA open-source project that enhanced its type checker by adding support for an additional keyword. The completion of this project helps OPA as an organization because it is an instance of the continuous improvement that is possible because OPA is open-source. The creators of OPA did not have to do any work (aside from approving the Pull Request in GitHub) to improve their product.

Research on the tradeoffs of using open-source is important and worthy of attention because open-source can be a great decision for a company that uses OPA for example, but it is not always the right decision for every company's needs. Organizations need ample information to make business decisions, and this research can contribute to their understanding of both the positive and negative implications of using open-source technologies. How should an organization evaluate the benefits and drawbacks of utilizing open-source products to fit their business needs? My research focuses on what business circumstances may lead a company to want or need distinct aspects of the tradeoffs of open-source, as well as the ethics of open-source itself.

Methodology

The goal of my STS research project was to discover the positive and negative aspects of using open-source products, as well as how to make an appropriate and effective business decision using that data. The nature of this research is mainly qualitative and relies heavily on information from scholarly articles about the adoption of different open-source software. The specific data explores how different business scenarios relate to aspects of open-source and how that would either help or hinder a company. For example, a business that wants customization in their software might choose to use open-source because of the ability to add or remove features freely without relying on a commercial vendor. This research was focused on the business circumstances involved in making this decision and the ethics of open-source as a whole.

The methodology behind the technical portion of this research is a bit more complex. The three main components of the design of my technical project included comprehension of OPA's code base and its limitations, using JSON schemas for code structure validation, and understanding the relationship between OPA's Rego type checker and its error messages.

Completion of this project required a strong understanding of OPA's code base that is publicly available on GitHub. The first steps of this project were reading and determining what each part of the code does. The limitations of the code were already defined in the Issues tab within the OPA GitHub repository, and one of these issues was the absence of implementation for several keywords in the type checker. The goal of this project then became adding support for the "allOf" keyword.

To implement a new keyword, it was necessary to understand how the other keywords are defined and evaluated, and further, to understand which location in the code itself keyword definition and testing was implemented. Achieving this understanding relied on coupling the use of JSON schemas and OPA's type checker. Schemas are a helpful feature of policy management systems to perform type checking. A JSON (JavaScript Object Notation) schema is a tool for validating the structure of data, outlining what input should look like. Schemas are typically passed in as input when evaluating a piece of policy code, and the associated type checker will use that schema as a blueprint to authenticate the structure of the code and inputs. In the case of OPA, the JSON schemas passed in as input during evaluation can interact with code written in Rego, which is a declarative language utilized by OPA for policy writing. When these schemas are included in the evaluation input, the Rego type checker uses them as a guideline for the intended code structure.

With a JSON schema as a guideline, the type checker can subsequently give more detailed and helpful error messages, as it can compare the usage in the actual policy code to that defined in the schema. As implementation is added for additional keywords in the Rego type checker, the type checker gains the ability to give specific error messages that indicate if the user has made a mistake involving *those particular* keywords. For example, if a developer made a typo involving the use of "allOf" in their input schema, the type checker could tell them that the root of that error relates to "allOf," instead of generally telling them that there was an error. Developers leveraging OPA can still use keywords that are not yet implemented in the type checker, but any type errors that they make in using such keywords would result in difficult-to-debug output.

The final steps of this technical project were to combine the knowledge of the code base, JSON schemas, and OPA's type checker to create a solution and add code for implementation of "allOf." After testing, this code was submitted as a Pull Request into the main branch of the OPA GitHub repository; this Pull Request was approved by the maintainers of OPA and was thus incorporated into the open-source, publicly available code base.

Analysis of Open-Source Tradeoffs in Business

The popularity of open-source software has been rising in recent years; in the last year, 60% of organizations have increased their usage of open-source software (Pimcore, 2023). Proprietary software has decreased in popularity and demand recently as well, with the main reason being the slow pace of innovation that is employed by enterprise proprietary software. The usage of proprietary software is expected to fall from 42% to 32% over the next two years, while the usage of enterprise open-source software is expected to rise from 36% to 44% usage in that same time frame (Pimcore, 2023). Open-source seems to be the trend of the future, which underscores the importance of the question that all innovation-driven businesses should be asking themselves: “Should I transition to open-source software?”

For businesses, the answer may be yes if they are looking for specific benefits from their choice. Some of the advantages of open-source software include cost savings, customization, flexibility, and quality (Kimachia, 2022). If a business is looking to reduce software development and/or operational costs, open-source software can be an attractive option since it is most often free to use. Cost savings can be especially important to a smaller company that may not have the necessary resources and funds to allocate solely to software development. In this case, open-source can significantly reduce their budget.

If a business wants to customize software to meet specific needs, meanwhile, open-source software provides access to the source code, which makes it easier to modify and tailor to precise requirements. For example, a growing business may want to use customized software so that they can flexibly scale their infrastructure as their business grows. Open-source also helps businesses quickly adjust to market changes in their industry in that they can add or remove features as needed without having to rely on the software vendor to make alterations (Kimachia, 2022). The ability to swiftly adapt to a changing market is helpful for a business that exists in a volatile industry such as finance, social media, or energy.

Another important reason a business may choose to use open-source is the potential for a heightened quality of software. Because open-source can be modified by anyone, there is a much larger pool of talent that could be working on a particular software, as opposed to a smaller team of developers that would work on a proprietary software. More contributors to a technology allow for increased collaboration, improvement of code, and more sharing of ideas (Kimachia, 2022). The value of this greater quality of software is notable because a company can get a better product for no cost instead of paying face value for commercial software. Although most companies presumably would appreciate a fundamentally better-quality software, perhaps a technology company (or a company that is looking to become more of a technology-focused company) would value this higher-quality software and thus want to pursue an open-source solution.

Despite these diverse benefits, a company may ultimately decide that open-source is not in its best interest due to factors such as dependability, time-to-market, and complexity. In the context of dependability, there is a risk of orphan software when using open-source products. Orphan software is software for which the developers have lost interest in the maintenance of the code. The software therefore becomes outdated or faulty due to a lack of upkeep (*Pros & Cons of Open Source in business,* 2021). Using abandoned software could be catastrophic for users of a product and subsequently for the stakeholders of a company, especially for larger businesses. To prevent this, a business would be well-advised to research the vendor of the software they are considering and to identify who is developing and maintaining the product. Does this vendor have a good history for keeping open-source projects functioning and operational? If the vendor uses the software themselves, this is an indicator that it is well-maintained. However, if there has been limited development in recent years, it is probably not regularly managed (Reichert, 2017). For a large business with many stakeholders, it might be smarter in this case to use commercial software that will most likely be more dependable.

In the case of a non-technical business that does not have employees who specialize in technology development, the complexity of open-source software might be reason enough to avoid it. If a software used by a company is especially sophisticated, it may require specialized expertise. While commercial vendors offer continuous technical user support for their software, open-source products do not provide such services (Gilani, 2017). As a result, organizations may have trouble managing their use of open-source software, especially if they lack the necessary technical skills. This issue may cause a company to have to spend more money and resources on learning the necessary skills for software servicing; for a less technically-experienced company, commercial software may be the best choice. If a business needs software that has guaranteed uptime and reliability, commercial software may be a better alternative than open-source because commercial software typically has a support team and service-level agreements (SLAs) (Gilani, 2017). Moreover, the support behind proprietary software typically extends to include pre-built solutions and deployment tools that can speed up the time-to-market for a product, which is helpful for a business that needs to deploy software quickly.

The issue of security of applications and products is understandably at the forefront of many business agendas. However, the evaluation of security when comparing open-source and proprietary software is somewhat of a gray area. On the one hand, because open-source is not developed in a controlled environment and anyone can alter the code, there is a chance that a developer could have malicious intentions and compromise the security of an organization's data. It is also very possible for a well-intentioned developer to introduce a vulnerability into the code base by mistake. It is important for organizations that decide to utilize open-source software to be conscious of the cybersecurity risks involved. On the other hand, open-source software can provide greater transparency and scrutiny given that the code is publicly available. Regarding open-source vulnerabilities, there are often dedicated communities of developers working to identify and patch vulnerabilities, while proprietary software is typically developed and maintained by a smaller team. Also, the speed at which vendors of open-source software can patch vulnerabilities is generally faster than that of commercial software (Kimachia, 2022). Regardless, it is difficult to decide one way or the other whether open-source or commercial software is more secure. Consequently, a business should carefully assess the security of their technologies on a case-by-case basis.

Ethics Associated with Open-Source Software

It is critically important for businesses to consider the ethical issues involved with open-source technologies, and subsequently to take the necessary steps to ensure that their software is transparent, accessible, secure, and ethical. For example, open-source software often relies on intellectual property, such as code libraries, which may be protected by patents or copyrights. If this intellectual property is used without proper permission or attribution, ethical issues, including patent infringement, may arise.

Another ethical gray area of open-source technologies is privacy. Open-source software could theoretically be used to collect and distribute user data, and without appropriate consent, this decreases the credibility and transparency of the software and that of any company using the software. Once again, the potential for compromised security of open-source software can cause ethical issues if developers do not take reasonable steps to protect their users' data and privacy.

Accessibility should also be on companies' radars if they choose to use open-source. This software is often developed and maintained by a community of volunteers who may disregard the needs of users with disabilities. To avoid this discrimination, businesses and vendors should be guaranteeing that the software they choose to employ takes accessibility into account (or modifying the software so that it does so).

How is the Open-Source Decision Ultimately Made?

There is no generalized formula for companies to use in their decision of whether to utilize open-source software. However, with this information, companies' decisions can be *informed* about the tradeoffs as well as the ethics surrounding their decisions. To summarize, a team that needs a low-cost, high-quality, and flexible solution may opt to use open-source products; conversely, a team that tends to value vendor support, speed-to-market, and simplicity may want to choose a commercial software to meet their objectives. Furthermore, a business needs to guarantee that they can address any ethical issues that may arise if they choose to pursue open-source. In general, a company should take this compiled information and decide which aspects of software they value the most. A company also should not make a blanket decision for all their endeavors in terms of open-source—given that one project may be suited this kind of product, while another project may not be. This decision should be a fluid one that is constantly reevaluated to ensure that the best-suited software is being used for any particular project.

Conclusion

The main goal of my STS research was to uncover the benefits and drawbacks of utilizing open-source products, specifically in a business setting. The findings of my research will help businesses make informed decisions about their software usage based on their own personal needs and goals. A company should first identify what they value most for their particular project at hand in terms of factors that include cost, customization, time-to-market, security, dependability, among others. A company should rigorously research each software in question, and the quality and reputation of the vendor. These findings need to be taken into consideration to ensure the reliability of the software. From an ethical standpoint, businesses should carefully consider how their choice of software will impact their integrity in terms of ensuring that their products are secure, honest, and accessible.

My technical work in the open-source space during my IBM internship made meaningful contributions to the open-source community as well as the potential users of open-source products such as OPA. This work will clearly benefit companies that utilize OPA for their policy management needs. Furthermore, users will have a more efficient debugging process, as implementation was added for an enhancement to the type checker that improves the quality of error messages. Improvements to OPA are made possible because of OPA's open-source nature. It is important to recognize the value in developers being able to freely contribute relevant enhancements to any open-source code base and associated product or service.

Future technical work could involve adding support for additional keywords in the OPA type checker. For example, the "with" keyword is still unimplemented. Additionally, there are many more limitations of the code base listed under the Issues tab in the OPA GitHub repository. Because open-source code is freely available and editable, any developer can make a contribution (pending Pull Request approval by code maintainers) to eliminate other existing limitations of OPA.

Future sociotechnical work could involve providing individualized aid to businesses who are deciding whether to move forward in using an open-source product for their needs. This research could even be helpful for a consultant who works for such a company, or for a project manager him/herself. In the future, there may be additional factors to consider that differentiate open-source software from commercial software that may influence a business’s decision about which technologies best meet their needs and requirements.

Word count: 3122

References

Corrado, E. M. (2009, June 16). *The importance of open access, open source, and open standards*. Retrieved September 26, 2022, from http://codabox.org/15/1/istl.pdf

Gilani, A. (2017, May 2). *Why Some Enterprises Don't Do Open Source*. EnterpriseAI. Retrieved April 4, 2023, from https://www.enterpriseai.news/2017/05/02/enterprises-dont-open-source/#:~:text=Some%20open%20source%20services%20contain,open%20source%20provides%20no%20accountability.

Kimachia, K. (2022, July 6). *10 Reasons Why Businesses Choose Open Source*. Enterprise Networking Planet. Retrieved April 4, 2023, from https://www.enterprisenetworkingplanet.com/data-center/why-businesses-choose-open-source/#:~:text=One%20of%20the%20most%20important,what%20a%20particular%20vendor%20offers.

Krogh, G. von, & Spaeth, S. (2007, August 13). *The open source software phenomenon: Characteristics that promote research*. The Journal of Strategic Information Systems. Retrieved September 26, 2022, from https://www.sciencedirect.com/science/article/abs/pii/S096386870700025X

Open-Policy-Agent. (2022, August 11). *OPA/adopters.md*. GitHub. Retrieved September 26, 2022, from https://github.com/open-policy-agent/opa/blob/main/ADOPTERS.md

Peters, S. (2018, July 18). *Do you know who your stakeholders are? - o'reilly open source convention in Portland 2018*. Open Source Conference | O'Reilly OSCON. Retrieved October 31, 2022, from https://conferences.oreilly.com/oscon/oscon-or-

2018/public/schedule/detail/66945.html#:~:text=Stakeholders%20are%20the%20people%20that,a%20project's%20contributors%20and%20users.

Pimcore. (2023). *Why Businesses Need Open Source Software*. Pimcore.com. Retrieved April 4, 2023, from https://pimcore.com/en/why-open-source

Pros & Cons of Open Source in business. Investintech. (2021, February 26). Retrieved October 31, 2022, from https://www.investintech.com/resources/blog/archives/7975-pros-cons-open-source-business.html

Reichert, J. (2017, September 22). *Open Source: To Use Or Not To Use (And How To Choose)*. Forbes. Retrieved April 5, 2023, from https://www.forbes.com/sites/forbestechcouncil/2017/07/20/open-source-to-use-or-not-to-use-and-how-to-choose/?sh=29c59787138d