Open-Source Software in Business and its Advantages & Disadvantages

Carlos Rodriguez and Onur Gürcay

February 7, 2020

Abstract Open-Source Software has established its popularity in the last years and it's growing with the interest of individuals and companies day by day. This paper will shed light on how open-source software became the trend in the enterprise area and reliable as a business model before going into details regarding the advantages and disadvantages of using open-source software in business and companies. This section will cover different aspects of open-source software and how they affect their users. Moreover, some examples from history will be shown as proof of adopting open-source software over proprietary software products to emphasize the conversion throughout the years. When looked back, it will be clear that most people's expectation from Red Hat to remain as the sole exception for open-source company got disproved crushingly. Additionally, the paper will cover some of the known companies on the market that use open-source software now to showcase how much popular open-source software has become.

1 Motivation

Using open-source software has become a long way during the last two decades and its transition always left us in awe. If you tell 20-25 years ago that open-source software is going to be the go-to trend for companies, people would probably laugh at you but here we are. It was always that "why?" question that needed to be answered. Why would giant companies in the software world that are making millions and billions of dollars of profit embrace open-source software? Microsoft who opposed open-source software almost 20 years ago bought the pre-eminent open source development platform GitHub in 2018. Figure 1 shows the growth of GitHub contribution from 2014 to 2018.

Steve Fisher, a User Experience Designer, explains in his TED talk that people can learn a lot from open source. [5] Open source provides new perspectives to look at the problems and come up with different solutions using collaboration, innovation, inspiration, and sharing. These elements together make the products overall superior in many ways compared to non-open-source software. However, this is not always the case since open source is not perfect and comes with some disadvantages. Getting additional knowledge regarding these elements and more motivated us to write this paper.

2 Introduction

> "Open source software is software with source code that anyone can inspect, modify, and enhance." [14]

It is important to mention the difference between open-source software and other types of software. Some software has source code that is only available to the person, team or organization controls it. Anyone else has no right to see or make modifications to

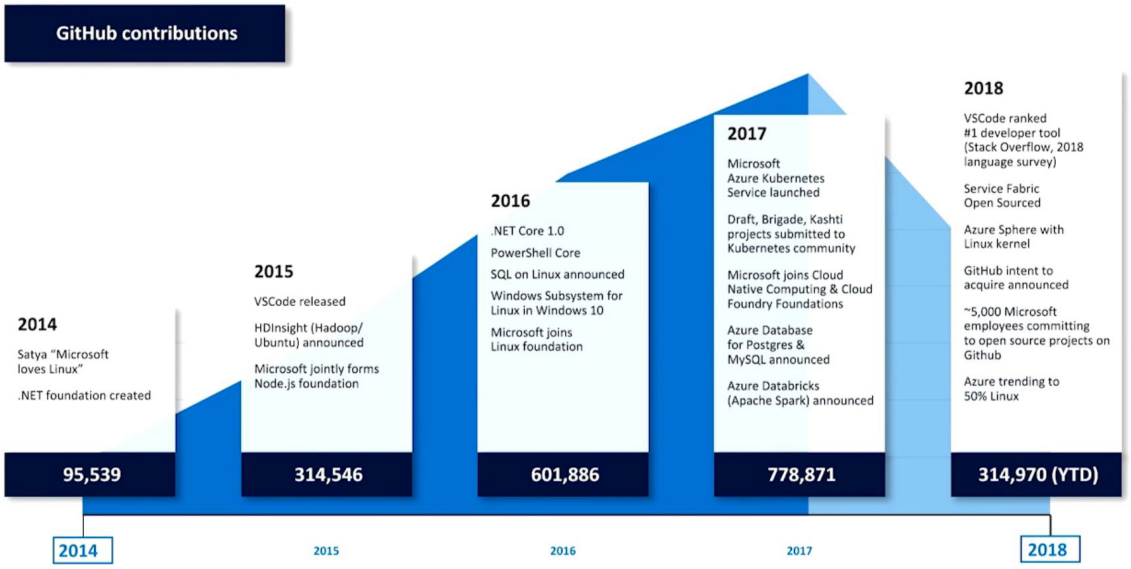

Figure 1: Microsoft's GitHub open source code contributions since 2014 [10]

it. This kind of software is called "proprietary" or "closed source" software. [14] These kinds of software owners allow other people to use their work by making users sign a license during their first use. With this agreement, the user is obliged to not do anything with the claimed software that the software owners have specifically permitted. Microsoft Windows and iTunes are some examples of proprietary software. On the other hand, open-source software authors allow users to view, alter and learn from it by making the source code public to everyone. Of course, open-source software also has licenses, however, these licenses generally give users permission to use the provided code for any purpose they wish as long as they do the same when they share their work. This generosity makes open-source software a great authority to promote collaboration and sharing, Collaboration and sharing will be mentioned in detail later as one of the advantages of open-source software in a later section.

The idea of making software accessible to everyone did not make everyone happy. Former Microsoft chief Steve Ballmer said during a commercial spot masquerading as an interview with the Chicago Sun-Times on June 1, 2001: [7]

> "Linux is a cancer that attaches itself in an intellectual property sense to everything it touches."

In his saying, Ballmer used Linux just as an example that uses the Free Software Foundation's GNU General Public Licence (GPL). The main "problem" for him was that GPL is forcing the software author to make their code open as well if open-source software code is used at any point in the coding process of the new software. It is also worth mentioning that GPL came into force in 2007, so Ballmer's concerns were clarified with the new version. In the 2010s Microsoft's opposition to open-source software did a U-turn and started to embrace this new approach. This is only an example, but a crucial one since it is about Microsoft's view. The next section of the paper will give more information about the growth of open-source software in popularity. This will lead us to various advantages and disadvantages of using open-source software on a company-level which will be explained in detail. The remaining sections will be about companies participating in the use of open-source software and when it makes sense to use open-source software.

Chart: Competitive Advantages of the Open Source Model

Business Attribute	Gen 3 Open Source (2019)	Gen 1 Open Source (2009)	Closed Source
Market Size	Large	Small	Large
Innovation Velocity	High	High	Low
Customer Acquisition Costs	Low	Low	High
Average Selling Price (ASP)	Low	Very Low	High
Market Elasticity	High	High	Low
Customer Lock-in	Low	None	High
Cost of Development	Low	Very Low	High
Lifetime Value	High	Low	High
Cloud First	High	Low	Low
Business Value	**High**	**Mid**	**Mid**

Figure 2: Competitive Advantages of the Open Source Model [10]

3 Growth of Open Source Software in Popularity

The first open-source projects were not really business-related, but more of a revolutionary movement as in taking a stand against closed-source software companies that were profiting 'unfairly'. Operating systems and databases became the first targets for open-source developers since these components were broadly used. First open-source software codes were originally created by and for developers. This means that they were not user-friendly at the early stage of this revolution. However, it was safe to say that these pieces of code were functioning well, flexible and robust. These properties evolved over the years and over a decade, Linux became the second most popular operating system for servers, after Microsoft. MySQL, on the other hand, became a serious contender for Oracle and started to steal from its market value.

The first entrepreneurial ventures attempted to capitalize on this adoption by offering "enterprise-grade" support subscriptions for these software distributions. [17] Red Hat and MySQL stood forward in this matter, however, this new approach of using the software was hard to monetize with only support services. These first attempts formed the first generation of open-source companies.

The second generation open-source companies had two main differences compares to the first generation: First, the software got developed within an existing company and not by a broad unaffiliated community. Second, these businesses were not making the whole source code public, but only license part of it for free. The rest of the code was under a commercial license, which made it easier to monetize the full product. With these changes in the second generation, companies were able to capture more revenue even if the market for their product did not have quite as much appeal as operating systems and databases. [17]

The latest generation (3rd Gen) companies developed the open-source software within the limitation of businesses. Often, more than 90% of the lines of code in these projects are written by the employees of the company that commercialized the software. Additionally, these businesses offer their own software as a cloud service from very early on which gives them multiple ways of monetizing their products. In Figure 2 you can see the differences between 3rd generation open-source, 1st generation open-source, and closed-source business models. [17]

Although the products of these Gen 3 companies are deeply under control of the host companies, the open-source community still plays a crucial role in producing the most innovative and relevant projects. GitHub is a perfect platform for this purpose where millions of developers can showcase their code, come together and work on various projects. The community is also its own product manager and QA department when it comes to improving these projects. The community itself can ask for improvements, adjustments or report bugs and at the same time the community itself again will be the care-taker to these requests.

Figure 3: GitHub Octoverse 2016 Top 5 open-source contributers. [6]

The improvement of open-source over decades get the attention of many known companies and made them embrace its business model. The next section will look into some of these known companies.

4 Known companies using and contributing to Open Source Software

The companies listed under this section are playing a major role in developing and maintaining the open-source software that powers today's business.[8] (See Figure 3)

4.1 Microsoft

As mentioned earlier in the paper, Microsoft was maybe the company that opposed open-source software most, but since then they reversed the course and started to embrace it. In 2016, Microsoft had the largest number of employees contributing to GitHub projects than any other company. It now has a partnership with other leading open-source companies such as Red Hat. It also open-sourced some of its most popular software, including .NET development tools, Visual Studio Code, PowerShell Code, the CNTK deep learning toolkit, TypeScript, and Redis. It also supports Linux on its cloud computing service and takes a cross-platform approach to development. [8]

4.2 IBM

IBM was one of the biggest contributors to the Linux kernel. Recently, it released WebSphere Liberty project under the Eclipse Public Licence, and it also has created and contributed to many other open-source projects such as OpenWhisk, Project Intu, and LoopBack. Moreover, IBM is also a member or sponsor of many leading open-source foundations, including the Linux Foundation, the Eclipse Foundation, the Apache Software Foundation, and the OpenStack Foundation. [8]

4.3 Intel

Intel was the most active company contributing to the Linux kernel in 2016 with a contribution percentage of 12.9. Moreover, like IBM, it is a member or sponsor of several

open-source foundations, including the Linux Foundation, the Eclipse Foundation, and the OpenStack Foundation. [8]

4.4 Google

Google released and contributed to more than 2000 open-source projects. It was the fifth on the most GitHub contributors list in 2016. Google also owns Angular which was the fourth on the same list. Some of the well-known Google open-source projects are Android, Chromium, Dart, Go, Kubernetes, and TensorFlow. [8]

4.5 Facebook

Facebook became one of the leading companies for open-source software and hardware in 2016 with the second highest number of contributors in GitHub. Its most popular open-source projects are React and React-native JavaScript development tools, Flow, HHVM, and Relay. [8]

4.6 Docker

The Docker containerization technology has emerged one of the most influential open-source projects for enterprise users and became one of the most downloaded repositories in GitHub with over 8 billion times. The Docker software is very popular with companies using agile and DevOps approaches, and the company claims,

> "On average companies using Docker experience a 7X improvement in how frequently they are able to ship software." [8]

4.7 Adobe

Adobe has a large commitment to open-source with its more than 250 public repositories on GitHub site. Some of its best known open-source projects are developer tools like the PhoneGap web development framework, the Brackets text editor and the Topcoat CSS library. Moreover, Adobe staff also contribute regularly to other open-source projects like Gecko, Blink, WebKit, Apache Cordova, Flex, Felix, and many others. [8]

To show the importance of open-source software for organizations we can look at Figure 4.

5 Advantages and Disadvantages of Open Source Software

This section will focus on the advantages and drawbacks of open-source software from many different points of view.

5.1 Cost

Open-source software itself does not cost anything. It can be deployed again and again on multiple machines without the need for tracking the license compliance and terms of use. Moreover, these software pieces are created to work with almost any type of computer which does not limit the user to use a specific type of hardware or operating system. The provided open-source software can be plugins, interfaces or engines for back ends. Based on company needs, these systems can be adapted and used for further development. Each and every feature can be integrated to make the project unique in its own way.

It is safe to say that open-source software help companies save money by providing ready-to-use software as a whole. Saving money on software allows organizations to invest

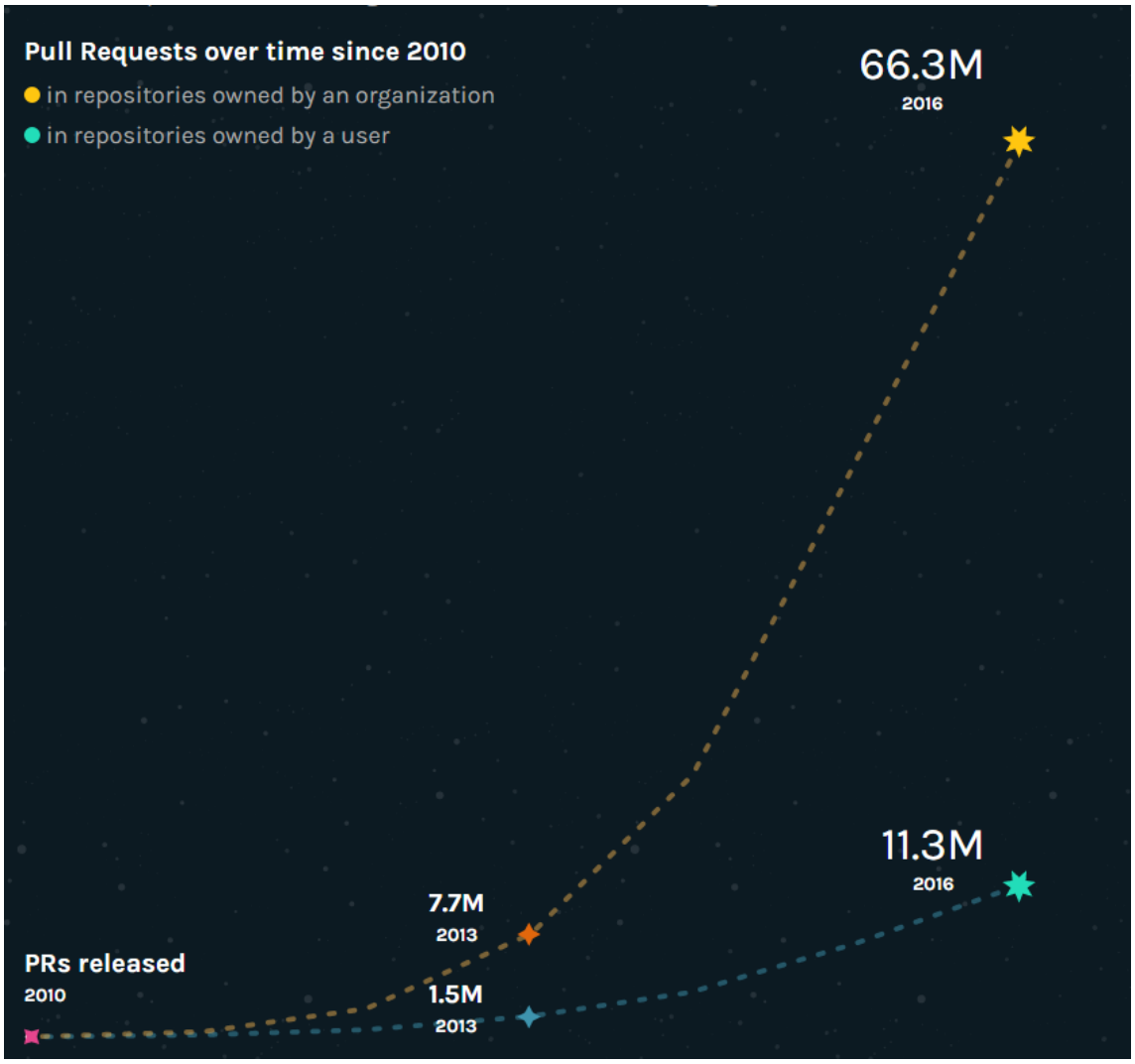

Figure 4: Pull requests on GitHub since 2010 according to GitHub Octoverse 2016. [6]

money elsewhere such as for high-speed networks or faster storage arrays, and also allows them to pay better wages to their developers who will work with the open-source code.

Time is another pivotal element open-source software saves for companies. In the software development life cycle, testing, debugging and integration consume valuable time during the development stages. Open-source software is good at reducing substantial time of development planning and these stages. [9]

In addition to these advantages, open-source software has some drawbacks in terms of cost. Even "free" software, in fact, comes with a cost. [2] Although the software is free, organizations have to train their software development teams to use this open-source software. Since the acquired open-source software will probably not match your needs perfectly, it will require some adjustments which will cost money and time. Besides, some open-source software providers also publish ready-to-go plug-ins for a fee that is not included in the "free" package. Despite all of these additional costs that might come with using the free open-source software, using open-source software saves money and time in total.

5.2 Service and Support

Open-source software support mostly consists of communities that are out there which might be very responsive or not. This is the main issue when it comes to providing support for open-source software. However, most open-source software applications do offer support and maintenance services for a fee. There are many qualified agencies and freelancers that can take care of problems. It is worth to mention that paid support options on most open source packages still fall far below what most proprietary vendors will charge. Providers of commercial support for open-source software tend to be more responsive, too, since support is where their revenue is focused. [12]

A perfect example of free support for open-source software is Linux distribution. Most Linux distributions have an online community with excellent documentation, forums, mailing lists, wikis and even live support chat. This is, however, not always the case. In a scenario where free support is the way to go, there is always the risk that the developers can receive a late response. This is not really the community's job to provide an answer to questions or problems in the shortest time possible. None of the community members is a part of any dedicated support unit. Also, since the software is developed by numerous people, users exactly do not have a specific person or company they can ask for support. Therefore, free support is not always reliable for open-source software.

5.3 Innovation

Open-source software provides a platform for exploring new opportunities and trying out components for software. Vice president principal analyst at Forrester points out the experimentation aspect of open-source software with the following quote:

> "Experimenting with open source is easier than with commercial products. Developers can download and try free existing open-source and then can decide if the technology is appropriate for their experimentation, if not they can change and experiment quickly with other components." [15]

Director of Solutions Engineering, EMEA at Github, Kai Hilton-Jones emphasizes the importance of OSS's innovation aspect with these words:

> "Open-source is an enabler for innovation, enterprises now understand that the same culture and working methodology that drives innovation at speed in the open-source community, also contributes to faster innovation inside the organization." [15]

5.4 Vendor Lock-in

With open-source companies or users are not forced to be in a relationship with a particular software provider. [11] Since the code is public, support will be available as long as the community exists. In comparison to open-source software, proprietary software customers are at the mercy of the vendor's requirements and dictates. Vendors can reduce the support and updates for commercial software that no longer generates enough revenue to justify their investments. The customer can not do anything about it and just have to accept the consequences. Open-source software reduces this risk by removing the financial motivations of a regular business. Gerald Pfeifer, CTO at the open-source vendor SUSE, underlines the advantage of open-source software regarding vendor lock-in with the following saying:

> "The idea of openness today is far-reaching: We all know Wikipedia, which makes knowledge available publicly and for free. Open source in its original

> form applies to software, made available for free and as source code. This became particularly important for the tech community after vendors started using "closed" code as a mechanism to create lock-in and capitalize on that. With open-source, on the other hand, everybody is able to examine the software, use and modify it at will, ..." [15]

However, vendor lock-in is still a possibility for open-source software: The community that provides the software might move off the project and officially orphan the software which would create the same negativity that happens with commercial products.

5.5 Flexibility & Customizability

Open-source software offers a high degree of customization possibilities for the raw software provided by being flexible in adjusting the code. [18] Developers can customize the software by adding extra functionalities or removing unnecessary features to match the requirements for the business. Flexibility and customizability of software is a requirement at the enterprise level because the open-source software in its purest form is often basic. Many additional layers are needed to meet the IT functionality level. Not only the in-house developers of enterprises but the whole community can benefit from such changes which would even provide more customization opportunities to develop the software even further.

5.6 Security

Open-Source software code is open to everyone, so all can view it whereas the situation can be described as a double-edged sword. On one hand, the security holes can be discovered and fixed more quickly because of all the contributors to the software. Anyone can fix bugs or upgrade the code, without relying on a proprietary vendor. On the other hand, it also means that the code is open to malicious users that can find out ways to exploit potential vulnerabilities and keep this knowledge to themselves.

Rob Whitely, Vice President of Marketing in NGINX assures that open-source software is more secure with the following quote:

> "OSS is more secure. That may seem counterintuitive, but it goes back to one of the core tenets Linus Torvalds espoused when he created Linux: the wisdom of crowds. Having more eyes on software means more testing, bug fixing, and hardening. OSS solutions benefit from a degree of security rigor that most companies can't match - either with software developed in-house or purchased proprietary offerings." [15]

Security level also can be a measurement for companies to adopt the software or not. They can easily review the code and determine the actual level of security, then make their decisions based on their needs.

5.7 Quality

The more people contribute to the code and the better the code will become since that piece of code will also include the experience of developers from different countries and different technologies, industries, and projects. This way, open-source software also gets closest to what users want because those users can have a hand in making it so. [12]

Ben Balter, Senior Product Manager at GitHub, voices his trust in the quality of open-source software with this quote:

> "...Blogging, content management, and operating systems are all problems with established (and mainstream) open-source solutions, to name a few. While your developers could spend their time reinventing wheels that the open-source community has already perfected, it's far preferable to use the world's best wheel, especially when that wheel comes at no cost to you. This frees developers up to work on yet-unsolved challenges, the types of challenges that are unique to and add value to your organization's mission..." [3]

A study in 2010 also has shown that technical superiority is the main reason enterprises choose open-source-software over proprietary software. [13]

5.8 Auditability

The importance of the audibility of open-source software can be understood better when the opposite direction - proprietary software - is observed. With closed-source software, the user does not have anything but the vendor's claims ensuring you that the software is kept up to standards and it is secure. This is more of a leap-of-faith situation where the user has nothing else but trust the vendor without being able to confirm the claims.

The transparency open-source software, on the other hand, assures the availability of viewing the code at all times.

5.9 Reliability

If a software serves its required specifications efficiently, then it is reliable. Reliability consists of three major factors:[4] fault detection, fault prevention and improvement of the software. Keeping these factors at the highest level maximizes reliability. In an open-source platform, since anyone can access the code and fix any bug, the software will receive continuous improvement such as new features and versions every now and then. The accessibility of open-source software indirectly fulfills the requirements of a reliable product alone.

5.10 Collaboration

Open Source developers are primarily motivated by pride in their work and the opinions of their peers. They tend to focus more on the quality of their product than everything else. All collaborators complete each other in some way and fix each others' mistakes to produce the best possible code.

Ben Balter mentions in his blog the importance of collaboration with these words:

> "Open source provides three advantages: first, you have the opportunity to tap the knowledge of the world's best developers, not just those on one organization's payroll. Second, the number of potentially contributing developers and thus the potential knowledge pool is orders of magnitude larger. Finally, open source software gets adapted to a variety of use cases, not just the one the publisher originally intended, surfacing bugs and edge cases much more rapidly than traditional, predictive QA processes. " [3]

5.11 License Terms

As mentioned many times before, source code is open and available to everyone when it comes to open-source software. One of the advantages of license terms with open-source software is that license terms tend to be equal to both sides, licensor, and licensee. Another plus worth mentioning is the conciseness and straightforwardness of license terms without

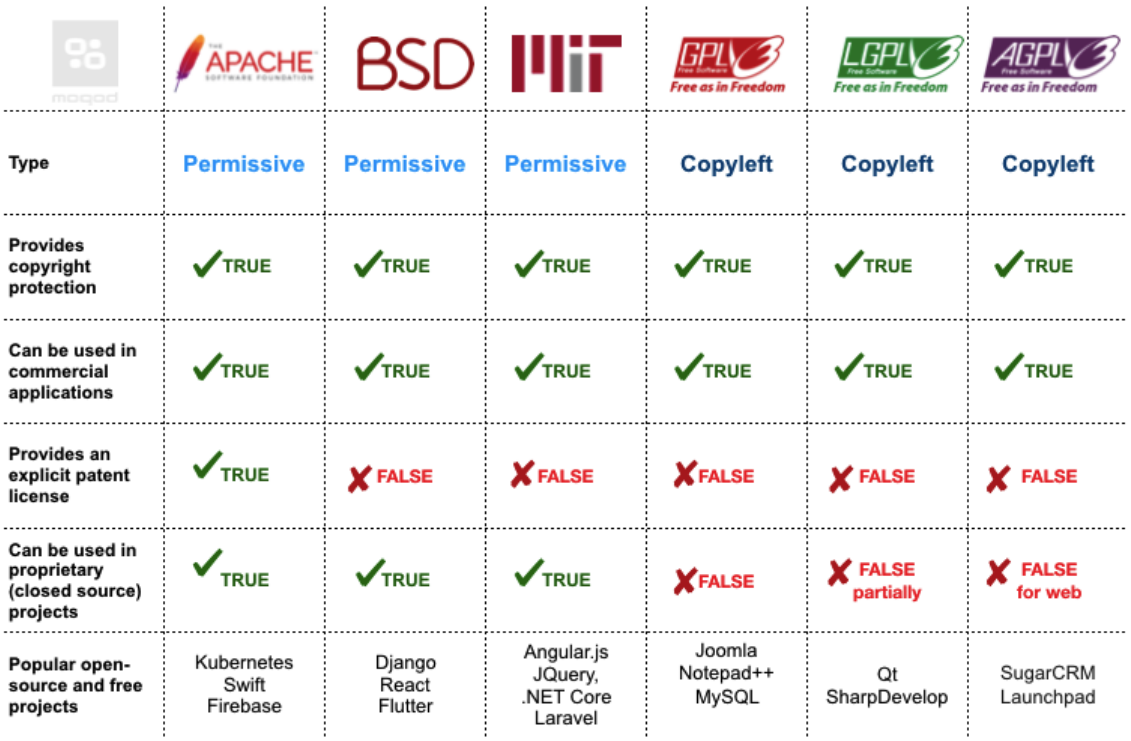

Figure 5: The most popular open-source software licenses compared with examples. [1]

leaving any grey areas to take advantage of situations favoring the licensor. There is no requirement to track license use in relation to licenses purchased.

The only drawback is the incompatibility possibilities between different versions of BSD (Berkeley Source Distribution) and GPL which limits the ability to use some open-source products with others.

This drawback is not a great issue when compared to proprietary software. [16] In this case, first of all, license terms usually tend to be favoring the vendor's benefit than the user. This is usually done by including confusing, hard to understand license terms that would make the compliance more difficult. The license terms can also show huge differences amongst each other which would make tracking the license requirements painful for the user.

Figure 5 shows a detailed list of open-source software licenses with similarities and differences.

6 Conclusion

The growth of open-source software is an undeniable fact. Even companies that were strictly against open-source software ended up supporting and even buying them over the years. As mentioned in the advantages and disadvantages section of the paper, it is clear that there are a lot more reasons to invest in open-source software and embrace it than shying away from it. According to Octoverse GitHub 2016[6], the number of open-source software contributors is almost doubling every year in several countries and keeps growing all over the world. There is no perfect code, but this growth makes the code get closer to perfection. As Linux Law asserts,

> "Given enough eyeballs, all bugs are shallow."

This improvement in code produces better software which encourages companies to use

open-source software. “Infecting” big companies with this “virus” is actually spreading the “disease” wider and faster so more and more users can benefit from open-source software. However, there is still room to improve such as the points mentioned as disadvantages in a previous section. The advancements regarding these problems will eventually happen which is going to make open-source software even more desirable.

References

[1] Understanding open-source and free software licensing. *Medium*, Sep 24, 2019.

[2] Neetu Jha Amandeep Singh, R.K. Bansal. Open source software vs proprietary software. *International Journal of Computer Applications*, 114(18), March 2015.

[3] Ben Balter. Why open source. *Ben Balter's blog*, Nov 23, 2015.

[4] CodeCreators. 8 advantages and disadvantages of open source software for business. *CodeCreators*.

[5] Steve Fisher. What can we learn from open source, Nov 29, 2011.

[6] GitHub. The state of the octoverse 2016. *GitHub Octoverse 2016*, 2016.

[7] Thomas C Greene. Ballmer: 'linux is a cancer'. *The Register*, 2 June 2001.

[8] Cynthia Harvey. 35 top open source companies. *Datamation*, Sep 21, 2017.

[9] Khalil Khalaf. The pros and cons of open source software. *Medium, 4Thought Studios*, Jul 11, 2917.

[10] Kurt Mackie. Microsoft's open source journey: From calling it a 'cancer' to buying github. *Visual Studio Magazine*, Sep 17, 2018.

[11] NiBusiness. Advantages of open source software. *nibusiness info*.

[12] Katherine Noyes. 10 reasons open source is good for business. *PCWorld Software*, Nov 5, 2010.

[13] Katherine Noyes. Linux is on the rise for business. *PCWorld*, Oct 11, 2010.

[14] opensource.com. What is open source?

[15] Tom Macaulay Scott Carey. what are the advantages of open source software in the enterprise? *Computerworld*, pages 1–10, Jul 2, 2019.

[16] Thomas Trappler. Is there such a thing as free software? the pros and cons of open-source software. *Educause Review*, Jul 30, 2009.

[17] Mike Volpi. How open-source software took over the world. *Tech Crunch*, Jan 12, 2019.

[18] Yaniv Yehuda. Open-source software: Minimizing disadvantages and maximizing advantages. *DZone, Devada Media Property*, Jun 07, 2018.

About this document. This seminar paper was written as part of the lecture *Free and Open Technologies*, held by Christoph Derndorfer and Lukas F. Lang at TU Wien, Austria, during the winter term 2019/2020. The authors A. N. Other and A. N. Other can be contacted by email at `onur.guercay@tuwien.ac.at` and at `carlos.rodriguez.lopez@student.tuwien.ac.at`. All selected papers can be found online.[1]

[1] `https://free-and-open-technologies.github.io`