

## IDC PERSPECTIVE

# Open Source Software: Choosing Between Community-Supported and Commercially Supported Products

Al Gillen

Susan G. Middleton

## EXECUTIVE SNAPSHOT

---

### FIGURE 1

---

#### Executive Snapshot: Choosing Between Community-Supported and Commercially Supported Software Products

For as long as there has been commercially supported open source software (OSS), there has been a misconception about the relative costs and benefits associated with OSS supported by a subscription. Users deploying commercially supported OSS products, which receive patches, fixes, and maintenance backed by subject matter experts, generally outperform community OSS in production deployments. However, for some scenarios, community OSS serves an important need and delivers value.

#### Key Takeaways

- There is a use case that justifies both nonpaid and paid, commercial open source software.
- The workload type, the expectations surrounding that workload, and the resources of a given organization to support an OSS portfolio over the long term must be considered.
- Free software is not free; realistically, free software requires substantial opex investment of human resources that quickly offsets any savings from avoiding subscription fees.
- It is important to think about the longer-term goals your organization has, along with the anticipated life cycle of the OSS products you deploy.

#### Recommended Actions

- Focus on where your IT organization can add value to your company's business goals. Supporting a custom OSS infrastructure stack probably delivers less business value than a custom application stack.
- Use nonpaid OSS to support prototyping and proof-of-concept activities, but plan to substitute paid, commercially supported OSS for production deployment.
- Train your IT staff to help them understand the true costs — and potential incremental costs — associated with using software that is not supported commercially in production scenarios so that the right decisions can be made to deploy commercial or community-supported products.

Source: IDC, 2019

## SITUATION OVERVIEW

---

### Evolution of Software Licensing

For professionals that entered the industry before the 2000s, there was little choice on software: You bought access to your infrastructure software and your applications through either a perpetual use rights license, or in a small minority of circumstances, you subscribed to that same software on an ongoing basis. Even in the case of perpetual use rights licensing, that license, in most cases, did not include ongoing software maintenance. Usually, there was no professional support offered – that is, unless customers signed up for a maintenance contract at an incremental cost or paid a high fee, that was charged by the event and the hours to resolution.

In the late 1990s and early 2000s, the growing maturity of Linux operating system software initiated what would ultimately become an industrywide transformation of software licensing as we had known it. Because Linux is open source software (OSS), commercial vendors selling products based on Linux created a model where the operating system software was effectively made available for free, but ongoing support for the operating system was sold on a subscription basis.

Widely known Linux brands such as Red Hat and SUSE popularized the use of Linux as an operating system solution as an alternative to Windows on x86 hardware. After Linux broke down those barriers, other OSS technologies followed, and today we have middleware, database software, management, orchestration, virtualization, and container orchestration software – all available in both community-supported and commercially supported versions. In the developer world, the vast majority of tools, languages, and related products are OSS or are based on OSS technologies.

There are some fine points to this notion of commercially supported "open" source software, particularly with regard to the free nature of OSS. Most OSS licenses mandate that the source code is made available for free. But source code still needs to be compiled into executables that can subsequently be run on the hardware. The mere act of compiling that source code ensures that the binaries customers produce themselves are, by definition, not identical to those that are supported by the vendor. The better way to understand this is that a customer compiling the Linux source code does not produce a branded executable set of bits. Only the vendor itself can produce a branded version of Linux that is able to be placed under long-term subscription support.

Some vendors have tried to circumvent this issue in a way to increase adoption of their products and, in the process, capture share from competitors. For example, both Oracle and Canonical offer Linux distributions in both source code and executable form, making it possible to run the same bits whether covered by a support contract or not covered by commercial support. This approach has seen some successes, particularly for Canonical, which has seen Ubuntu become widely used among the developer community; and for many private cloud pilot projects built, Ubuntu is the distribution used for the proof-of-concept stage – but often is replaced by another commercial distribution at formal deployment.

This approach of offering executable bits (in addition to source code) applies to far more than just some Linux distributions. Executable code is available for some OpenStack distributions, Apache software services, Kubernetes services and other widely embraced projects.

IDC has seen the industry attempt to rationalize the different benefits of these two options: to pay for commercial support or go with the alternative, using a nonpaid/community-supported version of the technology.

## Choosing Fee Versus Free: Not So Simple

IDC research has found that most customers use a mix of commercially supported and nonpaid OSS in their organizations, with selective deployments based on the workloads being supported, the critical nature of the task, and the potential impacts should an outage occur, contrasted against the cost of the subscription fees. There are benefits and challenges associated with both forms of software. Figure 2 provides a summary of the challenges and benefits of commercially supported OSS.

FIGURE 2

### Benefits and Challenges of Commercially Supported Software

Benefits and Challenges of Commercially Supported Software	
Benefits	Challenges
<ul style="list-style-type: none"><li>• Feature curation</li><li>• Professionally hardened</li><li>• Better security</li><li>• Ongoing support</li><li>• Predictable support life cycle</li><li>• Actively patched</li><li>• Certification and compliance tested</li></ul>	<ul style="list-style-type: none"><li>• Cost</li><li>• Broad subscription commitment</li><li>• May require universal subscription coverage</li><li>• Updates slower than community versions</li><li>• May have hybrid mobility limitations</li></ul>

Source: IDC, 2019

The benefits of commercially supported software all lean toward improving the experience of the organization and of individual employees inside the organization and reducing potential risks associated with outages through proactive maintenance. A vendor that builds a commercial product based on OSS will select a stable community release as the basis for a commercial release that will be constructed. The commercial entity is likely to consider aspects such as the maturity of new or emerging features and the compatibility of those new features with existing commercial versions the vendor supports, along with the ease or difficulty of migrating from a current commercial release to a new release that might incorporate emerging features.

In addition, the company doing the commercializing and hardening of the software will take responsibility to audit the source code to ensure it contains no malware or security vulnerabilities. The commercial organization then will take steps to test and harden the code to ensure it will offer the level of performance and reliability/stability required by commercial customers. It is worth nothing that any

found vulnerabilities or other improvements made during the process of preparing the commercial version generally are returned to the upstream community.

One of the most important benefits for users of a commercially supported product is the predictable life cycle of a product, which can extend up to as much as a 13-year period after the initial release. Part of that life cycle is the continuous delivery of patches and fixes that keeps the software product current with all known and newly found vulnerabilities and bugs. In other words, the vendor takes responsibility on behalf of the end-user organization to ensure that the solution the vendor provides is as safe, secure, and reliable as possible.

In addition, a support subscription with a commercial vendor means that customers have a pool of deep expertise to fall back should problems arise after deployment. This means that an organization's IT team can focus their expertise on the core technologies that are differentiating for their business, rather than supporting low-level system infrastructure software, where little meaningful differentiation can be made for most businesses.

Figure 3 provides a summary of the challenges and benefits of community-supported OSS.

### FIGURE 3

#### Benefits and Challenges of Community-Supported Software

Benefits and Challenges of Community-Supported Software	
Benefits	Challenges
<ul style="list-style-type: none"><li>• No cost for use</li><li>• Latest features available immediately</li><li>• Users can access and tweak code, if desired</li><li>• Ability to influence community</li><li>• No limitations on use or use location</li></ul>	<ul style="list-style-type: none"><li>• Requires in-house expertise</li><li>• Requires time spent on maintenance</li><li>• Applications won't be certified</li><li>• Difficulty in meeting regulatory compliance</li><li>• Frequent updates required</li></ul>

Source: IDC, 2019

Conversely, community-supported software can trade off the best business benefits that commercially supported products offer (including the long life cycle and steady stream of patches and fixes) for far lower acquisition costs. Community-supported technologies also offer a high level of agility to deliver

the most current, cutting-edge (and bleeding-edge) features and to be able to quickly embrace the changing surrounding ecosystem. With most community OSS products, there is no out-of-pocket purchase costs to acquire, deploy, and use these products. Further, there are no limitations on how customers use these technologies, where they choose to deploy, or how many individual instances they may choose to spin up.

In exchange, customers give up the external testing, validation, and delivery of patches and fixes that commercial products enjoy – and the availability of technology experts a phone call away to resolve issues that may crop up. Further, while the community typically addresses a newfound vulnerability quickly, the way that security flaw or bug made available frequently is through the release of a new version of the mainline project. This resolves the issue quickly – but means that a user that has built a business operation around the community solution must replace and reinstall the software product to gain access to the improved code.

This fast-upgrade cycle can have potentially negative impacts on layered software further up the stack including regulatory compliance certification, or it may have disruptive effects on the servers on which the software product is installed. Ultimately, there is no guarantee that new community releases will offer backward compatibility, and a customer's application might no longer run with the latest release of OSS. This is not a rule, but a risk that comes with free community OSS technologies.

In addition, organizations that use community technologies need to maintain a baseline of technical expertise on staff to support the OSS technologies in use. The level of expertise, the number of professionals trained in a given technology, and the degree of their coverage (meaning 8 x 5, 8 x 7, or up to 24 x 7) will depend on how critical these OSS systems are and how impactful an outage of one of these community-supported OSS solutions might possibly be.

For companies with sufficient scale, it might make good business sense to keep such subject matter experts on staff, including having multiple individuals covering a 24 x 7 x 365 window, and avoid subscription fees. For example, major hyperscale cloud providers can easily justify having such expertise on staff, as doing so supports their core business models. We believe that this approach may be practical for only a very limited number of very large enterprise accounts. Justifying such an approach quickly becomes difficult to justify as the size of the operation is smaller, and the IT budgets and IT staff all scale down.

Typically, what IDC finds is that customers triage their installations and use commercially supported software where there are regulatory or application certification requirements – and may use community-supported OSS selectively in installations that are less critical, where an outage won't hobble critical business operations or in preproduction deployments. Most enterprise customers IDC spoke with use a mix of commercially supported OSS and community-supported/nonpaid OSS technologies, with the majority of their deployments for production use built on top of commercially supported OSS solutions.

## ADVICE FOR THE TECHNOLOGY BUYER

---

### Capex Versus Opex Considerations

For most organizations, it's far easier to measure capex because those costs usually show up in form of a purchase order or a requisition. By comparison, staffing costs are buried in an opex journal entry. What makes it even harder to segment out is that typically IT professionals have a multitude of

responsibilities, and it can be challenging to segment off the portion of that individual's time that was associated with community OSS technology support versus supporting commercial products (either OSS or traditional proprietary products). Aboard many systems, a problem may span both OSS and proprietary software because of the interaction between different software layers.

IDC has conducted many return-on-investment (ROI) studies over the past two decades. One thing that stands out from these studies consistently is that the primary cost of ownership over a three- to five-year period is not the cost of acquiring software (through license or subscription) but rather the costs of IT professionals who install and configure systems, and maintain those solutions over the course of their life cycle, followed by the retirement of those systems at the end of their life cycle.

IT professionals who are supporting commercialized OSS products typically have a smaller set of specializations that they need to cover. Instead of needing to be a Linux kernel expert, an OpenStack maven, or a Kubernetes guru, IT professionals need to be familiar with how to work with these technologies, but they can rely on the commercializing vendor for deeper problems that may develop. This is essentially outsourcing for the deepest levels of technical support for technologies and frees up IT staff so they can create a competitive differentiation for enterprises.

**Benefits and Challenges of Software Licensing and Subscriptions**

Some of the biggest challenges businesses face are the need to transform their IT infrastructure to keep pace with the rapid changes within their industry and stay ahead of the competition while dealing with constant budget pressures. Further complicating these issues was the 2016 introduction of new FASB accounting rules that changed how leasing obligations were reported on company balance sheets and adding additional scrutiny to the evaluation of capex versus opex. In 2018, IDC surveyed 300 U.S. businesses and asked whether the new FASB requirements will impact their future procurement spending (see Table 1).

**TABLE 1**

**Impact of FASB on Procurement**

*Q. Will new FASB/IFRS regulations around IT purchases and revenue recognition rules impact your company's procurement strategies going forward?*

	Respondents (%)
Yes	86.3
No	11.0
Don't know	2.7

n = 300

Source: IDC's *IT Procurement Trends and Consumption Models*, August 2018

Overwhelmingly, respondents acknowledged that FASB would change their future procurement strategies, and most recognized that shifting toward opex models would be a solution.

CFOs are embracing these subscription models because opex-centric models are services based and closely align business usage with budgets. The advantages of the opex model versus a capex solution are there is no significant capital outlay up front, cash resources can be preserved for other business initiatives, and these models are customer centric: focused on providing detailed metrics about all aspects of usage. Because subscription models offer operational metrics about consumption and services all bundled into one monthly fee, they provide the information that businesses need to invest and manage product life cycles.

IDC research indicates that adopters of these models have found the following:

- Provide strong services and support options, which frees IT staff to focus on projects that generate revenue for the company.
- Align usage with budgets, which improves spending predictability.
- Focus on opex not capex, which improves flexibility and planning.

The challenges of these models also revolve around budgets. With growing investments for on-premise and off-premise clouds, opex budgets can be strained, and businesses that have significant capital reserves prefer capex options because of the depreciation options for assets that have longer life cycles.

IDC believes that the advantages of subscription models is that they align costs with usage and provide a predictable pricing model: a great solution for IT departments that are overburdened and facing budget challenges. Ultimately, CFOs will make budget decisions based on an analysis of many factors, but the use of a subscription-based model should be part of the evaluation.

## Skills Requirements

Today, CIOs are facing pressure from two separate, but interrelated drivers:

- **Digital transformation:** Business models are changing. Businesses with a long legacy must modernize, automate, and use software as an enabler to increase efficiency and competitiveness and lower costs. In many cases, these businesses are at risk of being disrupted by nonconventional competitors that have no legacy IT investments to slow them down, are highly agile, and use cutting-edge software for building, deploying, and supporting life cycle of their solutions.
- **IT skills:** Acquiring, retaining, and evolving IT skills is typically one of the top challenges that most CIOs face in running their operation. Today, the skills required to keep pace with a rapidly changing technology deployment environment demands organizations either invest heavily on existing employees or aggressively hire new talent into their organizations – or in many cases, do both.

Most organizations are challenged to keep up with the speed of technology changes taking place in the industry. The best benefit for CIOs comes from focusing their IT administrators and developers where those individuals deliver the most business value to the overall organization. This means not getting distracted by constructing a deep community-supported OSS stack with community edition products at multiple levels. Instead, organizations should selectively use noncommercial OSS products where there is a match between the cost of the related support burden, the potential for outages, and the comparative cost of commercially supported OSS alternatives.

## Developer Tools: The Exception to the Rule

In contrast to most other OSS technologies, tools used for development, deployment, life-cycle management, and related developer and DevOps functions are far more likely to be community-supported OSS. This is not to say that in every case, community OSS is the right option to consume, but often, there is a business benefit associated with using community OSS by developer and DevOps professionals. Some of the justifications include:

- **Developers are power users.** If there is a problem with their systems, typically developers have the skills to reimagine their systems and restore their work environment.
- **Development is not mission critical.** Developers' work is typically not mission critical, in the sense that an hour of downtime for developers does not cost the company millions of dollars. While no company can afford to have its developers sidelined for an extended period of time, when a developer's work environment is broken, corporate applications continue to operate, and rank-and-file employees continue to be productive.
- **Developer tools are changing very quickly.** The notion of a long-term commitment to developer tools is contradictory to the very nature of doing development work, which often relies on newly emerging technologies and experimental actions to develop breakthrough application functionality.

Accordingly, a substantial portion of developer tools are only available as community-supported technologies. While it is important to accommodate developers' needs during the development life cycle, organizations should have a plan in place that supports the use of OSS technologies for the proof-of-concept and prototyping stages but allows for a natural and nondisruptive plan to transition to commercial software stacks during late stage prototyping, so production deployments can be fully supported.

## Third-Party Support

One interesting aspect to the benefit of commercial support to open source software is the availability of third-party support. One example is IBM and Rogue Wave Software (acquired by Perforce in early 2019), which together offers support for community-edition OSS technologies. Rogue Wave extends Perforce's DevOps (application life-cycle management and code analysis tools) with Rogue Wave's API management, analytics, and static and dynamic code analysis tools.

A component of Rogue Wave's business is focused on providing support to community OSS projects. The company supports a Kubernetes distribution: CentOS Linux, OpenJDK; Apache ActiveMQ, Apache Kafka, Apache Spark; Jenkins CI/CD; and several additional technologies.

The company has been partnering with IBM for approximately three years now, with Rogue Wave leveraging IBM's expansive sales force to reach clients. The company employs deeply experienced enterprise architects that understand not only the products but the scenarios for where and how these technologies may be deployed in enterprises.

Like many other companies that work with OSS communities, Rogue Wave pushes patches and fixes it crafts for technologies it supports back up into the upstream community. How this relationship between IBM and Rogue Wave Software might change in the future remains to be seen after IBM's acquisition of Red Hat completes.

## Where the Industry Is Going

The general trends in the industry strongly favors the use of OSS. Few new software start-ups dare produce proprietary code, with the exception of companies that exclusively deliver a software-as-a-service offering. However, we see more start-ups building a core/enhanced model where the core software is open source, but the commercially supported version blends the core OSS together with proprietary extensions or management tools that address more of the needs of enterprise users.

In public cloud infrastructure, OSS is essential to construction of the basic services that clouds offer and is the enabling layer for an increasing number of value-added services including artificial intelligence (AI), Internet of things (IoT), and blockchain. However, at the same time, hyperscale cloud providers represent a threat to this delicate balance as cloud vendors commercialize – and support – OSS solutions themselves. Of course, these hyperscale providers have the critical mass that justifies supporting community OSS technologies for their own use (and for the use by their customers). At the same time, we are seeing some companies that commercialize OSS – particularly those that are also the primary contributors to the community project – moving to make it more difficult or impossible for hyperscale cloud providers to develop a version of that community code and offer it for commercial consumption in their cloud.

What is clear is the value proposition for disruptive open source software continues to move up the overall software stack. Relying on commercially supported open source software continues to make sense for most enterprise shops.

## LEARN MORE

---

### Related Research

- *Supporting Maturing Open Source: Why Ongoing Support for Infrastructure Software Is Critical* (IDC #US44640319, March 2019)
- *IDC PeerScape: Software Practices to Accelerate Adoption of Open Source Software* (IDC #US44877019, March 2019)
- *Five Open Source Software Projects to Watch Carefully* (IDC #US44639719, March 2019)
- *Microsoft Continues to Break New Ground with Open Source Software* (IDC #ICUS44532018, December 2018)
- *Microsoft Stuns Industry by Opening Up Patent Portfolio* (IDC #ICUS43392418, October 2018)
- *Hackathons Focused on a Greater Cause* (IDC #ICUS44288418, September 2018)
- *Market Analysis Perspective: Worldwide Open Source Ecosystem, 2018* (IDC #US43392618, September 2018)

### Synopsis

This IDC Perspective discusses the pros and cons of using community-supported open source software versus using commercially supported, paid-for subscriptions of commercial implementations of OSS.

"The common belief is that community-supported, nonpaid OSS is less expensive to use than the commercial alternatives. While this might be true for some classes of workloads – developer and DevOps environments in particular – most application deployments that support revenue generation in some capacity are usually better served by a commercially supported platform because of the longer life cycle, predictable upgrade paths, and support for problems that may crop immediately or years after the

product is put into production," said Al Gillen, GVP, Software Development and Open Source at IDC. "In the end, pairing OSS form factors will very much be driven by the workload and the importance of that workload."

## About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

## Global Headquarters

5 Speen Street  
Framingham, MA 01701  
USA  
508.872.8200  
Twitter: @IDC  
idc-community.com  
www.idc.com

---

### Copyright Notice

This IDC research document was published as part of an IDC continuous intelligence service, providing written research, analyst interactions, telebriefings, and conferences. Visit [www.idc.com](http://www.idc.com) to learn more about IDC subscription and consulting services. To view a list of IDC offices worldwide, visit [www.idc.com/offices](http://www.idc.com/offices). Please contact the IDC Hotline at 800.343.4952, ext. 7988 (or +1.508.988.7988) or [sales@idc.com](mailto:sales@idc.com) for information on applying the price of this document toward the purchase of an IDC service or for information on additional copies or web rights.

Copyright 2019 IDC. Reproduction is forbidden unless authorized. All rights reserved.

