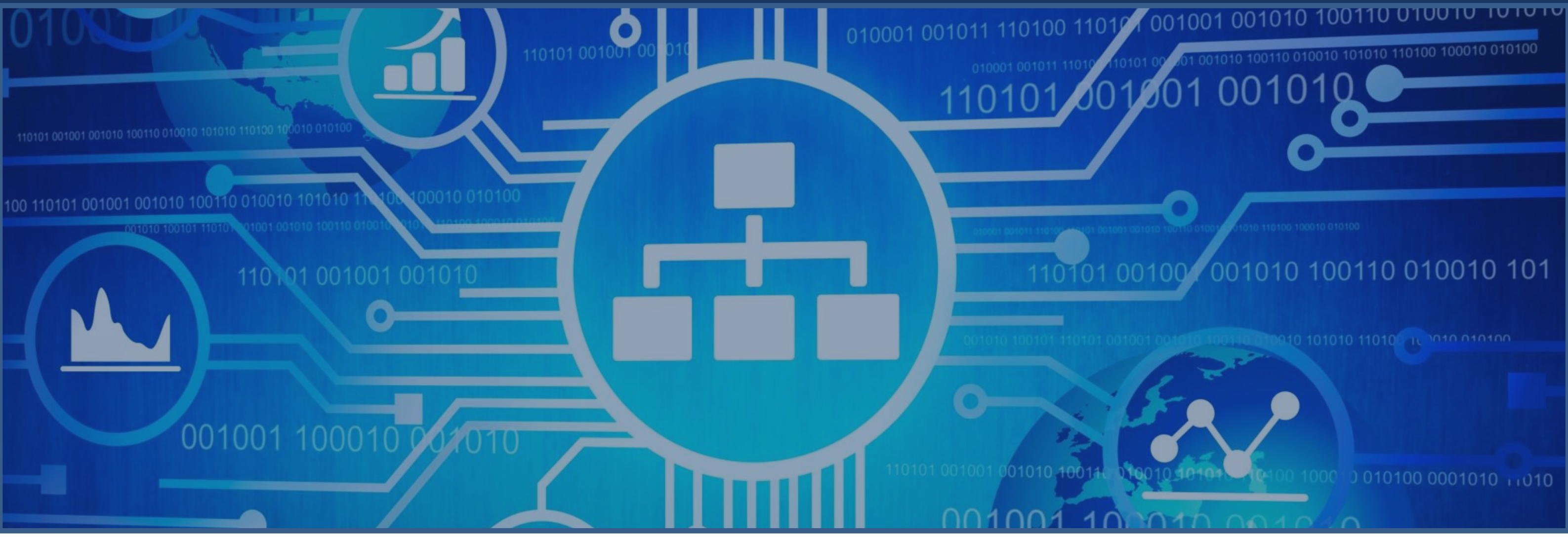
Technology **Insights**

FTI CONSULTING



A Perspective on Open Source Software Business Models:

How to Develop a Value-Driven Business Model

Evolution of Open Source Software

Open source software, where users have access to source code, has grown exponentially in the past decade and is driving a transformational shift in the software industry. With its origins rooted in the hacker community and free software principles, open source has become mainstream — disrupting existing business models for multi-billion dollar technology companies. Today, technology companies with open source business models are proving they can monetize their products and ancillary services, creating value through cheaper and improved technologies. Open source is here to stay and continues to create myriad opportunities for nontraditional players.

Open source changes the way enterprises perceive, purchase, and utilize software. In its early years, enterprises were attracted to open source software ("OSS") because it was a lower cost alternative to closed source proprietary software. By 2011, however, the price tag was still a key consideration but the primary reason for adopting OSS was to avoid being locked into one software vendor (Exhibit 1). More recently, quality of software solutions ranked as the primary reason for OSS adoption.[1]

EXHIBIT 1

Drivers for OSS Adoption[1]

Rank	2011	2014	2016
1	Freedom from Vendor Lock-in	Quality of Solutions	Quality of Solutions
2	Lower Costs	Consumer Awareness	Competitive Features & Technical Capabilities
3	Flexibility	Ease of Development	Ability to Customize and Fix

OSS Continues to Accelerate

Driven by consumer demand, OSS use increased in 65% of enterprises. "Hot" development and investment areas[1] include:

- 3D Printing
- Social Media Platforms
- Drones
- Enterprise Resource Planning ("ERP")/Customer Relationship Management ("CRM")

[1] Black Duck Open Source Software Survey;
2016: http://www.slideshare.net/blackducksoftware/2016-future-of-open-source-survey-results;
2014: http://www.slideshare.net/blackducksoftware/2014-future-of-open-source-survey-results.

As more enterprises moved to OSS, the entire culture and methodology around software development was disrupted. Traditional business models were threatened, triggering many merger and acquisition ("M&A") events. Vendors and service providers who survived and flourished did so because they evolved the ways they developed, licensed, supported and monetized software.

Today, open source is foremost a development methodology; not a single product, technology, licensing scheme or business model. It is the strategic foundation on which software is developed, delivered and deployed.

For enterprises, open source is an engine of innovation, which allows faster, more agile product development, accelerated time-to-market and superior interoperability.[1]

Can Free Code Generate Revenue?

The fundamental question often asked is, "How does one make money if the technology is free?" This is what discourages many technology companies from open sourcing their products. But sticking to traditional business models can give the advantage to competitors who make the move to OSS. To generate revenue with OSS, it is important to understand the dynamics among price, user adoption and revenue generation.

When software is created, it is worth something to an end user — it solves a user problem or creates a new opportunity — and therefore comes with a price tag. As users buy the software and adopt it, revenue is generated. Software companies that are price-focused tend to think that higher prices will generate more revenue. However, it is **user adoption** that is the driver for creating value. User adoption levels increase, as do revenues, when software prices drop — but only up to a point. Beyond that point, dropping price will cause revenues to stagnate (Exhibit 2).

However when a software company chooses to offer its base software for **free** and shifts its focus to monetizing user adoption, it can uncover opportunities for additional revenue models such as support, hosting, consulting, customization or add-on software.

Companies getting the OSS business model right understand how to monetize adoption and leverage community collaboration to create even more value with their software offerings.

EXHIBIT 2

Both price and adoption impact revenue

Price	Adoption	Revenue
$$$	Medium	$$
$$	High	$$$
$	Higher	$$$
FREE	Highest	New Revenue Models

Offer OSS or Someone Else Will

If your company currently offers software under a traditional proprietary licensing model, it is at risk of a competitor coming along and giving away the functionality for which you receive fees. When that happens, you have two choices: stick with your current business model or adopt the open source model.

The mobile phone space provides a cautionary tale. When Google entered the smartphone market with Android, its open source mobile operating system, Apple chose to go its own way and stick with its iOS platform as a closed ecosystem that they charged for, rather than adopting Android. Because Apple focused on customer experience and had the design experience, they have been able to withstand the Google business model.

Nokia, on the other hand, was not as fortunate. Rather than joining the Open Handset Alliance, a technology alliance focused on the developing and monetizing Android, the company chose in 2011 to partner with Microsoft on a Windows phone. This contributed to Nokia's fall from the top of the smartphone pyramid and the company has struggled since then. Blackberry went a similar route as Nokia and in 2015 came around to finally adopting the Android and getting on the OSS bandwagon.

OSS Community and Life Cycle

The concept of companies making money around OSS is not new – Red Hat has been doing it for years – but how it is done continues to evolve. Two factors, community and the open source software lifecycle, are crucial components of successful OSS business models today. Here's how they come into play.

Community plays a role in the two types of open software:

- **Community, or Project, Open Source** is developed and managed by a distributed community of stakeholders and developers who cooperatively improve and support the source code without remuneration.
- **Commercial Open Source** has open source software for which the full copyright, patents and trademarks are controlled by one stakeholder with the purpose of monetization. Code contributions can be made by community members but the stakeholder gets full rights to them.

Experience has shown that a having a strong community is a critical success factor for OSS companies, and the most successful companies produce OSS from a mix of community and commercial open source.

Software companies who want to enter the OSS space must develop an optimal way to engage their community without controlling, diluting or losing community members. The key challenge is typically around maintaining control and influence on product strategy, quality, documentation, user experience, features and functionality without alienating the community or losing agility and innovation.

Many companies resolve this by having an open innovation platform where care has been taken to ensure return on investment ("ROI") from the time, money and resources invested. Red Hat, for example, sponsors the Fedora community project, while selling Red Hat Enterprise Linux ("RHEL") to enterprise customers who need a more robust software package. Fedora code often serves as a testing ground for the enterprise features later found in RHEL.

EXHIBIT 3

The Open Source Life Cycle

Stage 1 – Communities and individuals start developing code.

Stage 2 – Vendors engage existing developer communities, provide structure and invest resources.

Stage 3 – Vendors dominate development and distribution; deploy evangelists to influence community and users.

Stage 4 – Vendor-dominated development communities and monetization plans in place. Time, resources and funds are committed.

Stage 5 – Commercial open source strategies in place and monetization plans executed.

Stage 6 – Financial community invests in scaling and improvement. Initial public offerings ("IPOs") and M&As occur.

Within the open source lifecycle, commercialization and monetization only begins to make sense by Stage 3, in which vendors begin to take the lead role in development and distribution (Exhibit 3). Commercialization does not always mean a company makes money directly from the "product halo;" it can also be ingrained in competitive strategy. For instance, IBM adopted Linux and accelerated its scale as an enterprise grade operating system, even though it was competing directly with its own version of Unix called AIX. By promoting Linux on IBM servers, it was able to launch a massive attack against Sun's UNIX-based operating system Solaris, which was shipped on its own hardware.

Direct and Indirect Business Models

Open source business models come in two varieties: direct and indirect. The direct business model creates earnings directly from software development, while indirect business models generate revenue from activities not directly related to coding or producing software.

For a software vendor to succeed commercially with either an indirect or direct model, it must have a strategy to drive rapid adoption through its installed base and then must retain users. Experience has shown that if a vendor expects to scale an open source product quickly, it cannot be obsessed with fine tuning product features and functionality. The vendor must invest in the revenue elements that drive adoption and retention (Exhibit 4) and deliver value to justify its price premium.

Given the differences in business models and revenue elements, where a software vendor focuses its investment can vary widely. Those like Cloudera, with dual licensing and an annual subscription for proprietary features, would focus on sales, marketing, customer experience and add-on pricing. The fundamental value creation occurs at the product innovation level. Vendors with indirect models need to focus sales, marketing and customer experience functions around training and educating users and providing support. M&A exits are an option for either type of business model, with successful deals including XenSource ($500 million), Jboss ($350 million) and Zimbra ($350 million).

As mentioned, customer retention is also a key strategic element, especially when competing vendors are commercializing and monetizing the same open source code framework. For instance, Hadoop vendors such as Cloudera, Hortonworks and MapR, monetize this Big Data framework very differently. These vendors not only have to retain customers for their adopted version but also ensure they stick around for ancillary services like training and support.

EXHIBIT 4

Comparison of Direct and Indirect Business Models

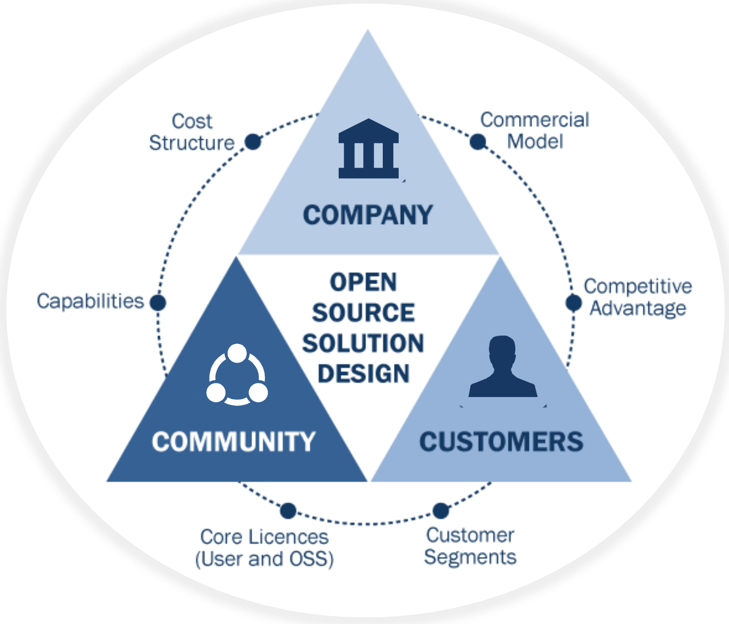
Business Models	Revenue Elements	Industry Examples
Direct	• Donations by individuals, non-profit organizations and commercial companies • Advertisements (e.g., website or merchandise) • Sponsors • Fees for add-on or proprietary software features • Dual licensing • IPO or buyout by a bigger company	**Cloudera, MapR**
Indirect	• Implementation • Training and education • Support • Books / manuals • Seminar and conference fees • Advertising • Hosting	**Hortonworks, RedHat, Google AdWords, WordPress**

Creating an OSS Ecosystem

The OSS ecosystem is built around the solution offering – the product, service and/or hybrid solution being marketed –which drives revenue generation and profitability. Company, community, and customers are the three primary components of the OSS ecosystem. The open source business model integrates these components and is built around six critical elements – the "6 Cs" – which are customer segments, commercial model, cost structure, capabilities, competitive advantage and core licenses (end user and open source).

EXHIBIT 5

OSS Ecosystem Components



COMPANY

The company developing and commercializing the solution; it plays the primary role in determining how the 6Cs are managed to foster a healthy ecosystem and drive profitable growth.

COMMUNITY

The community of users supporting the open source software and/or the solution offering. Community adoption is largely driven by the core license, capabilities, competitive advantage and the commercial model.

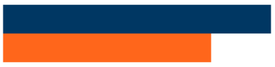
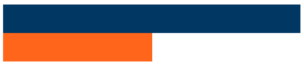
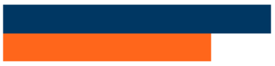
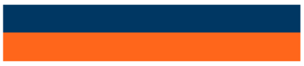
CUSTOMER

The end customer using and deriving value from the solution offering. The company must balance key factors including capabilities, cost structure and the core license to drive customer satisfaction and build deeper relationships with them.

There is considerable diversity in how companies manage the 6Cs to deliver value to their customers, the community and their ecosystem of partners. The four input elements (commercial model, customer segments, core licenses and capabilities) need to be managed for optimal outcomes of cost structure and competitive advantage. Each configuration of elements has different risks, opportunities and scale factors (see Exhibit 6).

EXHIBIT 6

Diverse OSS Ecosystems

Component	Hadoop Distributor ("HD") 1	Hadoop Distributor 2	HD 1 & HD 2 Comparison
Commercial Model	Provides the complete software solution for free and sells three levels of support subscriptions, as well as professional services	Provides a downloadable free, unsupported software version and sells premium, supported versions and add-on components as well as professional services	
Core Licenses	Download software with license directly for free	Provides different types of software licenses, including free, trial and premium paid	
Capabilities	Provides an entirely open source Apache Hadoop distribution without additional proprietary software; differentiated features come from broad community innovation as well a free to download add-on that enables data management and analysis in real-time (on-premises or cloud)	Provides an open source platform core in conjunction with a proprietary management solution, including enabling automated deployment and enhanced user convenience and providing a complete data governance solution	
Customer Segments	Focuses on enterprise customers who need to analyze, store and manage Big Data	Focuses on enterprise customers who need to analyze, store and manage Big Data	

■ Hadoop Distributor 1 ■ Hadoop Distributor 2

OSS Business Model: Key Considerations

Commercial Model	Industry Examples	Scale Factors	Risks	Opportunities
Product	**RedHat**	Software license sales determine revenue scale	Competition with proprietary software providers	Recurring annual revenue stream
Add-on Software	**Cloudera, MapR, Talend**	Depends on high community adoption of core software	Complex IP governance due to coexistence of free and paid versions	Lock-in revenue from software sales
Support	**Cloudera, MapR, RedHat, Talend**	Depends on product sales to determine volume of support contract	Renewal rate of support contract	Additional (recurring) revenue stream based on software sales
Services	**Hortonworks, Cloudera, MapR, Talend, RedHat**	Software license sales and community adoption for core software determine need for services	Service offerings to support multiple versions	Enable additional revenue stream and future product and support sales
Training	**Hortonworks, Cloudera, MapR, Talend, RedHat**	Software license sales and community adoption for core software determine training needs	Training to support different types of licenses	Increase community adoption and revenue opportunities
Customization	**RedHat, Talend, Cloudera, MapR**	Software license sales and service contracts determine customization requirements	Version proliferation from core based on type of license	Enable revenue lock-in from customers due to enhanced functionality

Insights from the Industry

Understand Scale Rates – Each commercial model has a different scale factor; earnings before interest, tax, depreciation and amortization ("EBITDA") profile; go-to-market strategy and value creation method. Understand each of these before developing an overall business model.

Focus on Commercial Model – Commercial models have multiple external factors, such as customers, competitor moves, pricing and adoption rates driving their success, speed and value creation.

Design "Lock-in"– Lock-in via drivers for recurring revenue is critical; these can range from add-on proprietary software to support and maintenance. Models without lock-in, such as services, training, etc., need higher investment in customer retention.

Look for Symbiotic Relationships – Mutually beneficial relationships can accelerate value creation through the network effect. Think of Android needing a phone and Samsung wanting an operating system. This was a key factor in driving higher Android adoption.

Executing the Open Source Business Model

Selling the Offering

To deliver and derive the most value from an open source solution offering, companies need to “get in front of it” – they must set a clear plan for what they are selling and how they will make money before taking the offering to the broader market. Given the “free” aspects of OSS products, companies must offer a competitive advantage and a compelling value proposition if they expect prospective customers to pay for anything. There are many different approaches to selling the offering around the 6 Cs, depending on what a company is trying to accomplish.

Cost structures can take multiple alternatives, even when the end solution offering is the same. For instance, Cloudera and Hortonworks, two companies besides MapR offering Hadoop-based solutions, have differing business models. While Hortonworks monetizes product support, Cloudera looks to monetize value-added solutions around the open source platform, including security, performance and analytics.

These differing strategies are in-line with common go-to-market and monetization approaches, including selling support as an insurance policy on deployments (e.g., real-time help); providing professional services including customization, consulting, install and set-up; offering both a free version and paid versions; mixing open source with proprietary code and selling it as a service; selling proprietary software apps/solutions or hardware solutions that add value; or combining any of these strategies into a hybrid strategy (e.g., Red Hat provides free and paid “enterprise” versions of their software as well as charging fees for support, install, maintenance and certification).

It should be noted that while Hortonworks and Cloudera employ different monetization strategies, neither is trying to make money from the open source element – rather they are innovating around the open source element and capitalizing on their innovations.

In addition, a key rule of thumb in determining revenue elements is that you need to choose an overall approach that will generate enough free cash flow to plow back into research and development (“R&D”) to continually improve your solution offerings and stay relevant.

Establishing Boundaries

Core licenses and the commercial model together determine the level of freedom provided with the solution offering. It can be restrictive or non-restrictive (e.g., the MIT license) regarding what companies and individuals can do, including modifying, distributing, re-packaging and selling. Licensing and copyright control need to be carefully considered – if they are too restrictive, you won’t be able to grow a community, and if they are too open, you risk competing with your community.

Growing the Community

Programmer communities are a crucial part of the OSS development model, enabling companies to expand R&D resources in a cost effective manner. When that happens, the endeavor becomes more of an open source community collaboration rather than solely a company-driven offering.

Companies need to engage and support their communities directly and should work to be viewed as a valued collaborator rather than a controlling figurehead. Achieving community validation is critical to establishing a long-term open source solution that will maintain interest and achieve profitable growth.

Nitin Kumar
Senior Managing Director
nitin.kumar@fticonsulting.com
+1 646 202 3166

Brian Hoard
Managing Director
brian.hoard@fticonsulting.com
+1 512 585 3080

Renjit Lal
Senior Director
renjit.lal@fticonsulting.com
+1 214 576 8110

About FTI Consulting's Technology Sector Team

Technology company executives do not need to face today's challenges alone. FTI Consulting is a leading advisor to tech companies across the software, hardware, services, Internet and semiconductor sectors. Our experts provide guidance to companies, their executives, equity holders and/or creditors throughout all stages of a company's life cycle. We provide strategic, financial and operational advice with hands-on implementation for business model innovation and transformation, mergers acquisitions, divestitures, due diligence, integration and separation.

The views expressed herein are those of the authors and not necessarily the views of FTI Consulting, Inc., its management, subsidiaries, affiliates, or other professionals.

About FTI Consulting

FTI Consulting, Inc., is a global business advisory firm dedicated to helping organizations protect and enhance enterprise value in an increasingly complex legal, regulatory, and economic environment. FTI Consulting professionals, who are located in all major business centers throughout the world, work closely with clients to anticipate, illuminate, and overcome complex business challenges in areas such as investigations, litigation, mergers and acquisitions, regulatory issues, reputation management, and restructuring.

www.fticonsulting.com