

Open-source software business models that create value

Stephen Turner

Known-Quantity.com, part of Turner & Associates, Inc.

ABSTRACT

The late management theorist Peter Drucker described a business model as the answer to three questions: “Who is your customer, what does the customer value, and how do you deliver value at an appropriate cost?” (Casadesus-Masanell & Ricart, 2011). Harvard Business Review reported that 70 percent of businesses in a variety of industries surveyed in 2009 were engaged in business model innovation (Casadesus-Masanell & Ricart, 2011). This includes users of software who want to drive costs down and have more control over their technology infrastructure. It also includes vendors who find new ways to compete.

Business models revolve around all expenses of the enterprise in order to make a designated price possible. Revenue models are customer-driven and price-centric (Popp, 2014). A revenue model is one dimension of a business model, and some vendors have multiple revenue models. The most successful proprietary software vendors have revenue models that generate cash flow with high margins, typically based on licensing fees and services. Those margins are used to develop upgrades, correct bugs, and make the software more secure.

A variety of open-source software (OSS) models are challenging the status quo. Some purists envision a dichotomy between good (i.e. open-source) and evil (i.e. proprietary) software. This paper provides a marketing and business context and connects value creation with new software programming paradigms. It describes twenty-two emerging OSS models that aspire to generate sufficient revenue to finance research and development (R&D) and marketing initiatives that will identify and address customers’ business problems.

Keywords: Business model, value creation, R&D, intellectual property, open-source software

INTRODUCTION

The second half of the Twentieth Century ushered in the “Information Age.” Five primary enablers are data, software, hardware, telecommunications and networking. Of course, computer networking has been essential for the development of the Internet, which has facilitated the free-flow of data, information, and knowledge. Networks require hosts, such as laptops, PCs or mobile devices. Networks are connected with routers, switches, links, protocols, applications, and agents, which are a mix of hardware, software and people (Park, 2014). Cisco is a company that has built a virtual monopoly in networking, which is troubling to critics. Concerns of these critics revolve around less innovation, few options, price manipulation, as well as security.

Many programmers, scholars and even chief information officers (CIOs) are working against monopolization in the software industry. These communities agree that too much power and capital concentrated among vendors has significant risks—particularly those whose business models are structured around intellectual property (IP) protections. Cornell University School of Law defines IP as “any product of the human intellect that the law protects from unauthorized use by others. The ownership of IP inherently creates a limited monopoly on the protected property. IP is traditionally comprised of four categories: patent, copyright, trademark, and trade secrets” (Wex, 2014).

The first submission of a computer program for registration in the United States was in November 1961. It was a tape submitted to the Copyright Office by North American Aviation. Shortly thereafter, two additional applications were received: a Columbia University law student was investigating how software might be protected, so he submitted a printout as well as tape. All three were finally protected in 1964 (Hollaar, 2002). This established a long tradition of software as IP with the help of the Federal Government.

Software vendors argue that it can cost millions of dollars to conceptualize a roadmap to address customer problems, build a marketing team, conduct market research, identify use cases, develop applications, test, resolve bugs, and obtain IP protection. Consequently, they need to charge end-users license fees that generate sufficient revenue to cover ongoing costs. This business model could be described as ‘build it once, license it a thousand times, and then reinvest in upgrades.’

Licensing then maintaining proprietary software on-site can be very expensive for enterprise users. Unreal Engine is a platform for gaming developers. A single license for Unreal costs \$750,000 (Salonek, 2014). Many proprietary software vendors do not make it that simple for example Oracle offers a menu of options. In-Memory Cost Management for Discrete Industries from Oracle costs \$25,000. Agile Food and Drug Administration Validation Pack is \$50,000. Agile Engineering Collaboration prices in at \$75,000. Agile ERP Adapter is \$100,000. Enterprise Data Quality Profile and Audit is \$150,000. Watchlist Screening is \$200,000. Moreover, Enterprise Data Quality Standardization and Match is \$275,000 (Oracle, 2014). Keep in mind, those fees do not include software upgrades or support. Oracle charges for licenses for each user in an organization known as “per seat.”

As a result, CIOs have become very sensitive to the total cost of ownership (TCO) when engaging with a vendor. Many companies do not use commercial-off-the-shelf (COTS) software “as is” because their operations have idiosyncrasies that require customization, integration and professional services. PricewaterhouseCoopers (PWC) calculated that the total annual expenditures for information technology are approaching \$1.5 trillion (PWC, 2007). CIOs are seeking ways to spend less and have more control. Cloud computing is an increasingly popular

way to cut costs on hardware purchases and software upgrades. These are capital expenditures that would otherwise appear on the balance sheet. Amazon has also jumped on this trend with its web service offering. Overall, these expenses significantly increase the TCO.

Software-as-a-service (SaaS) is a variation on that model, often including seamless cloud-based upgrades, on shared servers. Salesforce is the most well-known proponent of SaaS, offering a web-based customer relationship management (CRM) application, with related tools and data. OSS is another option that allows enterprises to download free software. They can control what they do with the software that processes their data. Most free software licenses, such as the GNU General Public License mandates distribution of a copy of the license with both binary and source forms of the application or framework (Stallman, 2010). This enables enterprises to avoid being 'locked into' a relationship with a single vendor. It sounds like the perfect way to contain costs, but does not encourage consistent innovation to address enterprise requirements. As a result, new revenue models are emerging to offer value to customers while also generating cash flow to support ongoing development.

MIT researchers investigated business models across publicly traded enterprises. Malone, Weill, Lai, D'Urso, Herman, Apel and Woerner (Malone, et al.) identified four types: Brokers, Landlords, Distributors and Creators (Malone et al., 2006). Dealers sell the right to be with prospective buyers or sellers of assets (Ferreira & Tanev, 2009). Landlords license or sell the right to use – but not own – an asset. Distributors license or sell rights without significantly transforming the asset. Creators license or sell right of ownership by significantly changing the asset.

Most companies generating revenue through OSS are Distributors and Creators. Software developers are Creators. Some build a new framework or application. Others may build upon an ecosystem or introduce new features and functions. Others sell what already exists. Distributors understand that the value they add lies not in the software itself, but in identifying a problem that can be solved or mitigated with a specific application.

OSS REVENUE MODELS

There are twenty-two models described in this paper. They fall into five categories: innovation in licensing, value-based, 'impure' business solutions, side businesses, and alternative funding. Open core, time-delayed open-sourcing, proprietary extensions, exclusive re-licensing and licensing content or insights are five models that each play with licensing. Value-based models include freemium and business source. Purists (i.e., advocates in favor of free software) envision open-source to follow certain conventions, but selling hardware optimized for OSS, open source as a service, and obfuscation of source code violate these agreements in various ways. While there are alternative funding sources, such as donations and sponsorship, advertising-supported software, partnerships with funding organizations, bounties, and crowd funding, some advocates feel it is better to develop side businesses, such as selling documentation, professional services, technical support, training & certification, branded merchandise, offer job placement or throw conferences.

These OSS revenue models vary widely in their ability to create value for customers. To create value, quantified user benefits need to exceed the adoption costs at the price of the software. Free beats the price of proprietary software every time, but adoption costs can be unpredictable. Some open-source developers even make their code harder to use, which creates a need for their services. Customers want to reduce the total cost of ownership (TCO), but need to

be able to achieve what they want as efficiently as possible. Free does not always have the lowest TCO.

Many vendors see each model as an opportunity to develop cash flow, but they can result in limited value if the technology is so complex that it needs endless documentation or technical support. Some vendors even combine multiple revenue models to create multiple streams of cash flow. This cash flow can support a broader business model that enables an OSS vendor to conduct market research, hire marketing and sales staff or distributors, pay commissions, build strategic partnerships, fix bugs, and introduce new versions with additional features.

A consumer of technology should read each of these to determine whether they add value to the open-source software and evaluate whether a vendor is actually committed to open-source. A seller should assess whether one or more of the models offer utility in building a sustainable business. This paper does not make the assumption that proprietary software models are inappropriate, or free software is superior. Each has their place in a free market.

Model One: Open Core

This model is the most fundamental form of innovation around licensing. Users are able to evaluate a free version: ‘if you try and like the open-source version, you will love the proprietary version.’ For example, SugarCRM is an open-source alternative to Salesforce. End-users can download a free version of the software with source code and access upgrades and patches. The company also offers four increasingly complete commercial versions for the price. Added-value features include support for Oracle databases or MS SQL, systems integration with other applications, such as MS Outlook as well as many other characteristics (CRM, 2014).

Gartner analyst Mark Driver found the Open Core model, also known as Dual-License, is the most common among open-source initiatives (Brandel, 2010). A dual license is an example of a transformation (Ferreira & Tanev, 2009). However, Driver is concerned “the open-sourceness of the product comes into question” because the commercial versions often have terms and conditions that limit what users can do with the application. As an analyst who comes in contact with many buyers, Driver frequently hears from customers who want to avoid commitments to vendors of proprietary software. Others don’t like paying for software if there are free options that are acceptable. (Brandel, 2010).

Given that this is the most common approach in OSS, an entrepreneur may want to consider to starting with this model. The challenge is to have a powerful core that people will value while also adding enough features to the proprietary version to create incentives for the customer to pay for licensing.

Model Two: Freemium

Jared Lukin coined the term Freemium in 2006 (AVC, 2006). Model Two is not quite the same as a dual-license in that it is more about usage and less about licenses or rights. Dropbox, Skype, and Actuate all offer this model. Actuate is an open-source business intelligence vendor, with code downloadable from the Eclipse ecosystem. Actuate’s BIRT iHub F-type’s freemium model is subscription-based. Actuate’s value proposition gives developers 50MB of data output each day at no cost, but charges additional fees for visualizations and tables over a limit. Data input is not restricted (Actuate, 2014).

Venture capital investors (VCs) commonly bet on businesses that can scale—generating a significant return on their investment if they succeed. Freemium gives lots of prospective customers a chance to use the product at no cost upfront. This is very attractive for VCs as it

demonstrates the need for the software and fuels the viral effect. It also enables many technical users to try it without selling through the CIO. The Managing Director of Bain Capital Ventures notes that OSS tends to be complex components that could become integrated into other systems. Salil Deshpande observes, “engineers can set it up and combine it with existing code, and it has the possibility of eventually becoming mission critical” (Forrest, 2014). This approach provides an easier path to upsell, because the technical team has already been convinced of the software’s technical merits.

This is a value-based model, because the vendor can capture value based on either usage or benefits the customer was able to realize.

Model Three: Time-Delayed Open-Sourcing

Much like the pharmaceutical industry, which has a limited period to make money while a drug is under patent, some proprietary software has a limited period of commercial viability. At the end of that timeframe, a vendor can choose to make some or all of the code open. This may not always be altruistic, but it can benefit the community of users, and expand the popularity of a product. An example of this is in 1998 when Netscape proclaimed it was focusing on enterprise clients and away from websites, so it could make code free for Netscape Communicator and Navigator free for users (Netscape, 1998). The open-source code ended up being repurposed in the development of Mozilla Firefox (Bell, 2006). Another similar instance was with OpenOffice, which is now an Apache project, and has reached more than 100 million downloads of free software. Apache’s website states “OpenOffice.org is an open-source project through which Sun Microsystems has released the technology for the popular StarOffice Productivity Suite. All of the source code is available under the GNU Lesser General Public License (LGPL). Sun is participating as a member of the OpenOffice.org community. OpenOffice.org is being hosted by CollabNet” (Open Office, 2002).

Red Hat’s Travis Kepley admires the approach of Id Software. He feels that Id’s framework is extremely powerful, which has allowed many open-source games to be built using their engine. This includes OpenArena, Xonotic, and World of Padman. Id makes its last version of proprietary software available and open once it has developed a new, more stable version. Kepley feels that games “would be near impossible to recreate without Id’s continuous support of the community.” He goes on to note that this has risks: “if Id were sued once, the hope of ever opening the source of any new Id project would be near impossible” (Kepley, 2011).

This model provides vendors an opportunity to position themselves as players in the open-source community by exploring how to license software at different time intervals. It could be considered the ‘after-market’ model, providing opportunities for software to be useful for years to come. However, vendors should be cognizant of whether an earlier version it is being released with a core code in its current release. Software such as Black Duck can be used to assess what code is already in the public domain, but vendors have to be careful to distinguish has not yet been made open-source.

Model Four: Business Source

Michael “Monty” Widenius developed a variation on the Time-Delay model. This offers ample opportunity to innovate with licensing, but revolves around adding value for a segment of customers. Widenius is the original developer of MySQL, which was sold to Sun Microsystems

for \$1 billion (Open Ocean, 2014). In this model, “all source code is available from day one and that most (but not all) users can use it anyway for free.” Widenius developed this model because he felt that a 30 percent margin on services (see Models Nine, Ten, and Twelve) is not sufficient to fund ongoing upgrades and R&D (Monty Says, 2013). Widenius suggests a three-year period before the software becomes open-source (Monty Says, 2013). End-user programmers are satisfied because they can see, test, manipulate and fork source code while also being able to re-distribute for free. CIOs are happy because a single vendor is not locked-in.

Craig Mundie is Senior Vice President at Microsoft. He is critical of a pure open-source model: “A common trait of many of the companies that failed is that they gave away for free or at a loss the very thing they produced that was of greatest value in the hope that somehow they’d make money selling something else.” (Software pluralism, 2014). Mundie argued that it was essential to retain the value of IP so that the investment in R&D can generate a return attractive to investors, as well as the broader community. (Software pluralism, 2014).

It is important to note that Microsoft Office is designed for broad usage on a massive scale. Office is a suite of applications that people are expected to know in order to do their jobs/ A user can purchase Office 365 Personal edition for \$59.99 (MS-Office, 2014). Microsoft sells 650,000 Office licenses per day (Velazco, 2011). Microsoft had gross margins that ranged from 70 to 76 percent between 2012 and 2014 (Yahoo finance, 2014a).

It is important to compare the 30 percent margin in service-based OSS model to the proprietary software model because the real problem for an open-source vendor may be the focus on niche markets, in that there is simply not enough demand. Niche markets often include specific programmers and IT professionals.

Widenius feels that the viral effect must be fostered, which then leads to revenue. Business Source pricing should be relative to the prices of vendors of proprietary software. His advice is to price Business Source between 10 and 33 percent of those competitors. “The entrepreneur needs to ensure sufficient income for both the staff and the entrepreneur to be able to work full-time on the product without having to do consulting or training on the,” (Widenius & Nyman, 2013).

The producer of Business Source software must decide which segment pays for access and usage. As a rule of thumb, one customer pays – based on specific criteria – for every hundred who uses it for free (Widenius & Nyman, 2013). Companies with a large number of users are more likely to be able to afford to pay for Business Source (Widenius & Nyman, 2013). Another criterion for payment is the value derived by using the software. Value-based pricing (VBP) charges different users different fees, based on the value generated by using a product. This can be complicated to administer, so it is not common with lean technology start-ups. However, it does provide a basis for differentiated pricing and cash flow for open-source vendors.

Model Five: Proprietary Extensions

Open-source tools are not always ‘enterprise ready.’ In fact, developers may intentionally produce a core that is of limited value to enterprises (see Model One). Proprietary but optional add-ons, extensions, plug-ins, libraries, and modules, are often part of the roadmap and business model canvas for OSS products or ecosystems. The approach is ‘license conform,’ but requires technical care. For example, statically-linked libraries can mix open-source code such as Apache

Hadoop with proprietary code, such as Impala by Cloudera. (Simms, 2009) As a result, this requires innovation with licensing.

Apache Hadoop's MapReduce (YARN is the next generation of MapReduce in the 2.0 ecosystem) and Hadoop Distributed File System (HDFS) are not designed to be 'enterprise ready.' Vendors such as MapR, WANdisco, and Cloudera add solutions that complete the offering with their proprietary extensions (Woods, 2013). MapR has its file system, which is faster than the Hadoop Distributed File System (HDFS). WANdisco adds value to HBase, which is more like a traditional database than HDFS, but scalable. Cloudera has built Impala, which is an analytic database more familiar to those who use SQL (Cloudera, 2014b).

This model is similar to Dual Core (see Model One). It can be leveraged to create a specialization in a particular industry, such as financial services, energy, transportation, or retail. The main difference: Open Core assumes that the vendor developed the core, while Proprietary Extensions can be developed by a different vendor—one who did not develop the core. This is an excellent model for vendors with expertise and existing relationships in a particular industry vertical.

Model Six: Proprietary Re-licensing

A vendor can re-license an application under a proprietary license and sell the product without the source code if a software product uses OSS under a permissive free software license. Berkeley Software Distribution (BSD) is one form. Apple has used this model for a range of open-source projects licensed under BSD. The Unix operating system kernel was sold in proprietary Macintosh PCs (Oram, 2011). See Model Twenty-One, which is often coupled with this model. Re-licensing is a form of innovation with licensing, but the vendor should be adding value in some way to the original code. They should disclose to customers what is new, better or different.

Model Seven: Obfuscation of Source Code

Machine code is not abstracted for use by programmers. Instead, it is written to provide exact instructions for the way central processing units operate. Source or machine code can be distorted so that a competent programmer cannot easily follow the instructions. Princeton scholar Allison Chaney explains "Programmers may deliberately obfuscate code to conceal its purpose or its logic to prevent tampering, [or] deter reverse engineering." Perl, C, and C++ are easiest to obfuscate (Princeton, 2014).

Obfuscated open-source is considered by some to be deceptive and misleading when it is being released as 'open-source.' It could be described as the 'have your cake and eat it too' model because it allows a vendor to protect its IP, while claiming to offer access to source code. The Free Software Foundation has taken a position against obfuscation because the source code needs to be understood, usable and enable further development. "Obfuscated 'source code' is not real source code and does not count as source code" (GNU, 2014b). Additionally, the draft for version three of GNU General Public License – which is the standard for OSS – specified that 'preferred' source code must be released, not in an obfuscated form. In other words, programmers outside of the tight-knit circle of developers who worked on the project need to be able to understand the source.

Revolution Analytics makes source code open, but provides executable binaries to revenue-generating customers. Their service includes packaging and compiling the software (Rickert, 2014). They produce a version of R, a statistical modeling software used by analysts and data scientists. Nevertheless, obfuscation is not the way to build customer value because it can initiate a need for technical support that would otherwise not be necessary without obfuscation. It is one of the three models that purists shun. There is good reason for this type. It does not solve a business problem.

Model Eight: Sell Documentation

Selling documentation, offering professional services, providing technical support, training & certification, branded merchandise, offer job placement or presenting conferences are seven of the models that are forms of side businesses. They do rely directly on the OSS, and it is possible for third-parties to create a niche for themselves offering one or more of these, without ever writing code.

Marc Achteilig includes user manuals, online help files, screencasts, demos, and tutorials in the list of documentation needed to support software. For manuals, he estimates that it takes two hours to write a single page and another hour per revision (Achteilig, 2014). If a writer followed GNU guidelines, an extensive manual would be required, taking hundreds of hours. GNU wants each variable, function and command to have a set of instructions, known as a 'documentation string.' These are expected to be written in narrative form, with sentences that use subjects and verb predicates. GNU also specified that exhaustive documentation should have a two sentence summary for each string. (GNU, 2014a). Efficient documentation enables the user to take advantage of the software as quickly as possible—reducing adoption costs.

Lean development is becoming more common in open-source projects. This trend impacts documentation. Traditionalists are likely to see documentation as a risk reduction strategy, compared to proponents of agile development who want to be as efficient as possible as they development documentation to support the project (Agile Modeling, 2014). Nevertheless, easy-to-use documentation does add value, so there are some vendors who recognize that they can charge for the service. SYSPRO is a value-added reseller that produces documentation for other vendors' software. It charges for demos, e-Learning modules, feature guides and some webinars (Syspro, 2014). Moreover, Wordpress makes its manual available for free to download as a low-resolution electronic version, but charges for versions with more features. It costs \$6 for a higher resolution version with printable images. It charges \$60 for a version that can be rebranded and sold by a reseller to an end user. That version can be edited and customized (Wordpress, 2014a).

Many open-source programmers may assume that users have a certain level of knowledge when they produce documentation. As a result, there may be an opportunity to produce additional documentation that addresses different target markets that vary from the assumptions. Additionally, formats of the documentation can add value for end users who need more graphics, screen shots and examples. Some vendors also provide libraries or samples for users to leverage when practicing new software. All of these options focus on the user, rather than the technology. Each option can expand the market of users.

Model Nine: Selling Professional Services

As noted at the start of this paper, Malone et al. (2006) identified four types of business models across industries. They also categorized four types of offerings that provide the basis for the exchange of value: Physical, Financial, Intangible, and Human. The last two are most closely associated with software. OSS is intangible because it is a non-physical asset. It is also often coupled with human capital – professional services – that require the time, expertise, and skills of people (Malone et al., 2006).

Professional services often involve customization to suit the needs of a particular customer. A simple integration of two different pieces of software can be one service. For example, the end user may want to use HAMR, a real-time big data analytics engine from HAMR Analytic Technologies, along with Apache Hadoop file systems, such as HBase or HDFS. The customization would require minimal time because an existing application programming interface (API) would be leveraged to make them work together. Professional services can also involve forking or tuning (forking can be executed by the author of proprietary software) of a piece of software. OSS (with non-obfuscated code) thus enables a third-party to create new features that were not part of the source code. In some cases, enterprises hire third-party developers to fork software to make it more secure or add functionality.

As previously noted, gross margins are typically in the range of 30 percent for services, which is not sufficient to fund ongoing R&D, but can enable service businesses to hire staff and grow around a particular ecosystem. Independent businesses can offer professional services, such as value-added resellers (VAR), distributors or consulting practices. PWC, Deloitte, and Accenture are all consulting firms that offer professional services. Without the need to generate profits to reinvest in software development, these consultancies have additional flexibility.

Model Ten: Selling Technical Support

Some might argue that technical support is one of the many professional services offered by companies with open-source business models. Technical support often has tiered levels of assistance and can be outsourced abroad or even performed by a bot. Algorithms are developed based on the range of problems submitted or expressed by users. A decision-tree allows the support person or bot to ask a series of questions to identify the exact problem. The same problems often re-occur with different users, enabling tier one support person or a bot to rectify those problem easily with a proven solution. Many websites provide a live forum or library of common questions and answers. StackOverflow.com can be searched by keywords or tags. Some developers or service providers offer helpful suggestions on this site, which leads to engagement and job offers. It has build a business model around the traffic generated from crowd-sourced technical support.

Adobe is a vendor of proprietary software for the creative community. It directs people to its website to get answers to common questions – that is equivalent to tier one customer care. The site provides many answers but also upsells tier two support. Services can be charged on a one-time basis. Individuals pay from \$29 to \$249 per incident (Adobe, 2014b). Adobe also offers support programs for enterprises that aid in users of Creative Cloud for enterprise, Acrobat (i.e., PDF), Digital Publishing Suite and other software (Adobe, 2014a). Adobe's Enterprise Term License Agreement, which includes support and requires a three-year commitment (Foxen, 2014).

In contrast to Adobe's software, Drupal is an open-source project with a community of more than 35,000 programmers and 1.1 million website developers around the world (Drupal, 2014b). In addition to a support web forum (for tier one support), it offers a free Internet Relay Chat (IRC) with other members of the community. They describe it as a lounge, thus experts with time volunteer to help people with questions and concerns (Drupal, 2014a). This is equivalent to Adobe's tier two, but does not use algorithms or decision trees. In addition, there are at least 70 vendors who offer support services to the community for a fee. Many embed support with other services. Acquia, for example, provides web hosting, staging environments, and support staff with technical expertise in Drupal (Acquia, 2009).

ExpressionEngine (EE) Core is an open-source website content management system (CMS) built by EllisLab (EllisLab, 2014b). They charge \$299 for the proprietary version of EE, a web development platform that includes three months of support, commerce and other add-ons not included in the free version (EllisLab, 2014a). In addition, they have three levels of support for those who need more help. EllisLab charges \$1,999 per month for Platinum support, guaranteeing first response within one business hour and unlimited support tickets. A Silver plan costs \$49 per month and users can expect a within a day while a Gold plan costs \$299 per month, which guarantees a response within four hours.

LivePerson is a business that creates value by offering the infrastructure and people needed to provide training. There are opportunities for businesses to succeed within ecosystems, such as Drupal and Hadoop, without having to develop code. A side business can be run by anyone who can address the needs of users.

Model Eleven: Conferences

Blogger and developer Joe Eames does not work on conferences for the money: "I wish I could say that putting on conferences is lucrative, but sadly, no matter how it may appear on the outside, conferences are just not a profitable venture." He feels that it is a great way to build community and enhance one's resume (Eames, 2014). In contrast, David Saef has found that successful conference organizers can generate 30 percent profit margins. Thus, it is a big business and creates value for those who know how to leverage potential synergies and tap into growing markets. Saef is executive vice president of marketing for Global Experience Specialists—a conference organizer (Ferrel, 2013).

U.S. Bureau of Labor Statistics projects that conventions will grow by 44 percent between 2010 to 2020 (Ferrel, 2013). "Black Hat" is a conference that began during the dot-com era to address the curiosity of hackers. With an attendance rate of over 6,000 participants (Ferrel, 2013) at a fee of \$2,595 for the full event pass, the conference could generate as much as \$15 million (Black Hat, 2014b). Moreover, sponsors such as Cisco, Juniper Networks, and IBM generate additional revenue. Diamond sponsorships can cost up to \$145,000 (Black Hat, 2014a). As a force behind Disrupt conferences, the chief operating officer of TechCrunch is positive about the opportunities for those who want to tap into growing markets.

"Everyone is getting into the game," said Ned Desmond, "because it is a real business, and it has better margins, when done right" (Kaufman, 2013).

M. Scott Havens as the president of The Atlantic tells, "The best part is that once you have the infrastructure in place it can be very scalable." Moreover, "There is very little incremental cost to adding a fixed sponsor." (Kaufman, 2013) The Atlantic is a magazine and digital media property. It can also be noted that The Atlantic likes live events because they

generate 20 percent of the company's revenue. However, Julie Hansen has concerns about who is sponsoring and who is speaking or sitting on panels at conferences. She is president and chief operating officer of Business Insider and claims that a large sponsorship check cannot buy a spot on a panel or keynote speech (Kaufman, 2013). Many other conferences are perceived as vehicles for sponsors to get their stories in front of potential customers.

VARs and distributors might develop conventions to create demand for a range of products from various vendors they represent. Conferences also generate lists of attendees, which can be monetized or leveraged in a number of ways.

Model Twelve: Training & Certification

Many trade associations offer training and certification, which are considered to be vendor-agnostic. Certified Business Intelligence Professional (CBIP) is the culmination of training by The Data Warehouse Institute. A number of open source vendors offer a limited curriculum, revolving around their products. For example, Cloudera charges \$2,295 for Data Analyst training and \$200 to sit for the Data Science certification test (Cloudera, 2014a; Pearson, 2014).

Then Zipfian Academy appears to be an independent training organization, but is closely aligned with GraphLab, a machine learning abstraction developed at Carnegie Mellon University. Zipfian offers a 12-week immersive boot camp for data science and data engineering, charging \$16,000 tuition. It issues a refund of \$4,000 if one of their hiring partners successfully recruits the graduate (Zipfian Academy, 2014c). The Academy claims a 93 percent placement rate, with an average salary of \$115K within six months of graduation (Zipfian Academy, 2014b). Graduates have been hired by Uber, Square, Airbnb, Coursera, Skymind, Tesla and Facebook (Zipfian Academy, 2014a).

Zipfian co-founder Jonathan Dinu made a presentation at the GraphLab Conference in 2014. "If we are to push the field of Data Science sustainably forward," Dinu asserted, "we can most effectively teach complex topics and modern tools like GraphLab" (Dinu, 2014). This is not to say that Zipfian follows the same model as Cloudera, but students participating in Zipfian should expect to learn about GraphLab. Presumably, Zipfian Academy is self-sustaining but reduces the cost of marketing for GraphLab by penetrating the world's leading organizations with people trained in GraphLab. Thus, it is argued that that Zipfian is designed to train and place enterprise users for the GraphLab open-source framework for machine learning (GraphLab, 2014).

Model Thirteen: Job Placement

Although ExpressionEngine Core is an open-source CMS, its community is not large, in relation to Drupal. It and has a job board that costs \$49 for 60 days, and showcases developers in search of work (Director-EE, 2014a, 2014b). For example, EllisLab has a job board, online user forum and conferences—multiple revenue models that reinforce demand creation.

One of the most popular websites for the open-source community is GitHub. (GitHub, 2014a). GitHub has hundreds of jobs posted on a given day (GitHub, 2014c). It charges \$450 for a 30 day posting (GitHub, 2014d). Any business that wants to build a community needs to help its members find ways to monetize their skills, code or content. What better way to promote the

use of an open-source framework than to create a community that has employees, contractors and employees who use the technology?

Ruby on Rails is also an open-source web development framework introduced by 37 Signals (Basu, 2013). In 2014, the company was restructured and created at least three new brands: Basecamp, Highrise, and WeWorkRemotely.com. By analyzing who was submitting and responding to job postings, 37 Signals determined that 75% of those hired worked remotely. Hence, the rebranding focuses on telecommuters and supports employers who want to hire within a day. (Signal v. Noise, 2013).

Model Fourteen: License Content or Insights

Creative Commons (CC) has seven types of licenses to use cultural content at no cost – a form of open-source for the public domain (Stieben, 2014). Each is a form of licensing innovation that provides flexibility to entrepreneurs. CC requires attribution and has exemptions, limitations, and specified rights. Users may copy, redistribute, remix, transform, or build upon the content at no cost (Creative Commons, 2014). However, YouTube has concerns about people making money from videos that use content from the public domain. “For your videos to be eligible for monetization, you must own all the necessary rights to use all visuals and audio elements commercially, whether they belong to you or to a third party. These factors include (but are not limited to) logos, thumbnails, intro/outro/background music, software interfaces, and video games. If you decide to incorporate third-party content in the video, you must clear the rights to use and monetize this content on YouTube” (Google support, 2014a). When content is eligible, YouTube allows account-holders to generate revenue from ads and other fees on the Google platform (Google support, 2014b).

Media companies are now exploring ways to monetize added-value content. Bloomberg charges for videos, photos, news feeds and data (Bloomberg, 2014a). Bloomberg Valuation Service accesses data from markets around the world. Bloomberg aggregates and transforms mostly free data to enable subscribers to price their securities in a more transparent way (Bloomberg, 2014b). It also licenses non-real-time data for \$11 million that can be used by risk managers (Waters Technology, 2014) The process of aggregation and packaging data to address the needs of users—based on market research—is what creates value.

Data or other content can be licensed for use in financial services, marketing, and entertainment, however the underlying software is open-source. Video game Steel Storm has an open-source engine that is free, but the artwork is licensed for use – at a cost. Free software with non-free artwork (content) conforms with CC Attribution-NonCommercial-ShareAlike 3.0 Generic license (Steel Storm, 2014a). The producer of Kot-in-Action charges \$15 for a license for content to play the game “Burning Retribution” (Kot-in-action, 2014). Steel Storm 2 has changed the model, asking for pledges ranging from \$1 to \$400, depending on the content that will be used (Steel Storm, 2014b).

Model Fifteen: Donations and Sponsorship

Five models that provide alternative funding sources are not so much about adding value for customers, but doing whatever is needed to finance software development. These are advertising-supported software, partnerships with funding organizations, bounties, crowd funding, and donations and sponsorship. Eran Galperin is the entrepreneur behind BinPress, a

platform that adds value to GitHub by introducing licensing and pricing options. He is critical of the simple donation model, regardless of whether it utilizes PayPal, Flattr, Gittip or Bitcoin Grant. In his experience, open-source projects either generate revenue through some kind of license model, are handed over to the community, or slowly die when volunteer developers are not able to focus on adding features, because they are inundated by bug reports. His point is that volunteers and donations are not enough to make open-source a viable model (Galperin, 2013).

Donations are commonly given on a one-time basis, as a way of saying 'thank you' to a non-profit organization or team of developers. Open-source foundations struggle to apply fundraising practices to leverage these contributions. Basic data management disciplines are rarely applied by these 501(c)(3) organizations. Donors who receive news about projects feel more connected to the organization and see they are making a difference. In turn, this leads to higher levels of funding in the future from the same donors. Bo Crader and Kristen Fulks recommend a single donor database, data enrichment with wealth screening and biographical updates (Crader & Fulks, 2011). They work with Blackbaud, a vendor of software and databases designed for foundations.

A sponsorship model frequently enables recurring revenue over many years from larger, more stable organizations. Those brands often want to demonstrate a commitment to open-source. The Apache Software Foundation is the organization behind more than 150 open-source projects, such as Hadoop, Spark, Flume, Kafka and Tez (Project-Apache, 2014). In addition to asking for donations, Apache offers four levels of sponsorship, ranging from \$5,000 to \$100,000 per year. The foundation spends the funds on administrative and secretarial duties, network bandwidth, hardware, marketing and professional services, such as legal and accounting (Apache, 2014b). Six-figure platinum sponsors include Cloudera, Hortonworks, Facebook, Google, IBM, HP and Matt Mullenweg, the man behind Wordpress (Apache, 2014a).

The open-source foundation is directly involved in Wordpress for bloggers, as well as bbPress for forums and BuddyPress for social media. More than 434 million Wordpress plug-ins have been downloaded (Wordpress, 2014c). Thus, The Wordpress Foundation keeps it much simpler by asking for donations to pay for educational initiatives as well as trademark protections (Wordpress, 2014b)

As another variation, The Linux Foundation offers renewable memberships. This model offers more predictable revenue than pure donations but requires someone to administer the benefits. The individual membership fee included discounts and hardware, peripherals and books. Memberships are probably the best balance for both donors and foundations, but tend to leave vendors out of the equation unless a vendor is the force behind an organization. R&D is expensive, and a significant fundraising effort is required to finance ongoing development.

Model Sixteen: Branded Merchandise

Non-profit organizations with memberships and online stores can make money and build brand awareness. Online retail stores typically have a 14 percent profit margin (Peters, 2006). Some for-profit organizations that giveaway branded swag at tradeshow also sell USB drives, apparel, pens, drinkware, backpacks, mouse pads, and electronics online. Red Hat sells gift certificates to their online store and red fidoras (Redhat, 2014).

Model Seventeen: Advertising-supported Software

Spacebug.com enable developers to download new and updated open-source projects, access documentation and sample uses as well as blogger articles and news about the community (Spacebug, 2014). Amir Shevat, the entrepreneur behind Spacebug, calculated that 60,000 unique visitors go to the website each year, and advertising revenues are roughly \$100 to 200 per year. Understandably, Shevat feels advertising is not a sustainable model for open-source web sites. He assumes that developers are simply looking for free code when they visit his site. He also suspects that ads become transparent to some developers, while others use ad blockers. Advertisers also do not seem to provide content that address the needs and interests of developers, so they don't click-through them (Shevat, 2010).

An open-source project must have a high-level of popularity, and the website must have significant traffic to generate substantial ad revenue. To generate \$100,000 per year from a site using Google's AdSense, a site would need 36,500,000 visits per year, a far cry from Spacebug revenue (Mohan, 2014).

Vendors need to be realistic about what it will take to earn money with this model. Seven-figure investments are often required to build and update the website with content that will attract nearly 40 million visits annually. This is the least viable model for generating revenue to finance R&D.

Model Eighteen: Partnership with Funding Organizations

The Federal government makes Small Business Innovation Research (SBIR), and Small Business Technology Transfer (STTR) grants available to entrepreneurs. Their ideas must have the potential for commercial viability, which is assessed before the end of the second phase of the SBIR or STTR. Departments of Defense, Energy, Health & Human Services, Agriculture, and National Science Foundation and several other agencies publish opportunities for grants that encourage innovation and entrepreneurship (SBIR, 2014a). Through funding of \$21 billion by the federal government, 15,000 firms are alive today (SBIR, 2014b). Open Source Solutions, Inc. received an SBIR grant, and they were able to develop OSCINT, an open-source tool for use by the intelligence community (Steele, 1996). On the other hand, software such as Apache Stanbol, Create.js, and VIE.js were funded by The Interactive Knowledge Stack (IKS) project. In part, its funding came from the European Union to build an open-source platform for content management systems (Fago, 2012). Funding can be found in the private sector as well.

AT&T worked with Continuity (renamed Cask) to combine and improve IP from each organization. The resulting data processing framework was made available for free in GitHub. The open-source project was released in 2014 as Tigon.io (Tigon, 2014).

Larger enterprises grow stagnant and often look to fast moving start-ups for new ideas. In turn, entrepreneurs seek capital and distribution networks. This may be one of the best models for financing ongoing innovation. Joint ventures are excellent ways to structure agreements of this kind. Some involve cash investments, while other are focused on generating revenue through sales.

Model Nineteen: Bounties

One small monetary offer by one person can be combined with many others who use the same software and have the same idea. This encourages development and also helps to identify the most common problems and show what the market wants. Some people have a wish-list of features and make a bid. Others encounter bugs, crashes or unexplained errors and set the value to fixing them.

Larger open-source projects, which have more than a million lines of code, tend to have more bugs than smaller open-source projects—70 percent to be precise. According to security analyst Coverity, small projects have .44 errors per thousand lines of code, compare to large projects with .75 (Koetsier, 2013). Apache Spark, for example, has 33 committers and has been plagued with crashes and other problems. Large projects of this kind tend to have less formal testing processes. Smaller open-source projects are more tightly-controlled. Bounties represent one way to place control on a project.

Several websites enable bounties, including and members who place bounties fix the bugs or add feature. Most bounties on Bountysource range from \$5 to \$100 (Bountysource, 2014). If 100 people each placed a \$20 bid or bounty for the same bug or feature, the bounty would add up to \$2,000. Some developers are able to make a living by chasing after bounties. Other developers can build a reputation as an expert in a particular ecosystem or project when they win a bounty for a challenging problem.

Model Twenty: Crowd Funding

Bountysource can also be used by the developer to fund an open-source project in advance. The entrepreneur posts what they want to do, and individuals decide whether it is something that they would like to encourage financially. A supporter typically expects to be able to get access to a pre-beta version and provide feedback. Other platforms such as Indiegogo, Kickstarter, and Catincan can also be used for what is described as a “reverse-bounty” or “pre-order.”

The process of this can be understood better through examining The Codebender project, which is a small project with set a goal of raising \$5,000 on Indiegogo (Scherschel, 2013). Codebender describes itself as “online development & collaboration platform for Arduino users, makers and engineers” (Codebender, 2014). The Arduino community makes interactive projects on a hardware-software electronics platform (Arduino, 2014). Codebender raised \$6,881 from 98 people to pay for the server and operating costs for Arduino IDE, which is web-based (Scherschel, 2013).

JIBO is Indiegogo’s most successful campaign. It is a companion robot for the home that can see, hear, speak, relate, learn, and help its family. They were able to pre-sell 4,800 on Indiegogo and raise \$2,287,609 from 5,550 people (IndieGoGo, 2014). The campaign was so successful that its web site stated that JIBO was no longer accepting pre-orders.

Codebender is a good example of an open-source project, which is one or more pieces of code—its components were integrated into a larger system—Arduino. The adoption costs are higher to integrate open-source into an application or framework, which makes it less valued in the marketplace. This is in compared to JIBO, which is considered to be a whole product with lower adoption costs. Nevertheless, Crowdfunding provides a good way to test the perceptions of value of a project or product.

Model Twenty-One: Sell Hardware Optimized for OSS

Tivo is a hardware device that was designed to pause live content on television, giving users a chance to take a break or come back later to pick-up where they left off. Series 1 Tivo was based on the Linux operating system and enabled sophisticated users, developers, and hackers to modify the open-source code—licensed by the GPL. Series 2 was also Linux-based, licensed by the GPL, but denied user access to the code. This became known as “Tivoization” and it infuriated the open-source community (Coding Horror, 2008). It does not solve customers’ problem though free software.

Cray is a creator in the high -performance computing (HPC) hardware space. It recognizes that customers want to do more with software than is optimized to run on specific HPC hardware. In 2014, Cray announced that their new Urika-XA hardware system include a software stack with Hadoop, Spark as well as Cray Adaptive Runtime for Hadoop and the Urika-XA management system for big data analytics. These are open-source projects managed under the Apache Software Foundation. Cray customers will pay for hardware nodes which have been pre-integrated with storage and free software. This approach allows a hardware vendor to target software users who may be dissatisfied with the functionality of open-source code on commodity machines. It addresses the problems of existing users who already know the issues with certain software, and can be sold as a capital expenditure in which the end user makes a purchase. It can also be bundled into an Open-Source as a Service model, in which a third-party owns the equipment and sells access.

Model Twenty-Two: Open Source As A Service (OSaaS)

Pioneer Richard Stallman advocated for the concept of “free software” beginning in the 1980s. Stallman declared the need for four freedoms: “the freedom to run a piece of software ; for any purpose; the freedom to understand and change it; the freedom to share copies of the software; and the freedom to share changed or modified copies. A piece of software must have all of those properties to be free software” (Gordon-McKeon, 2014).

For those who do not wish to spend the time to configure and distribute data across their own cluster of nodes, companies such as Qubole Data Service (QDS), Amazon Web Services’ Elastic MapReduce (EMR), and Altiscale offer ways to analyze data through a terminal or web-based user interface. Customers pay for access to hardware on a shared basis, much like a utility—following the Landlord model (Malone et al., 2006). Altiscale’s value proposition is “Pay as you go on monthly fixed fee plans. We don’t charge by the node, but by usage.” They focus on eliminating capital investment while bring in operating expenses within a given budget. (Altiscale, 2014). This alternative accounting method is particularly attractive to publicly-held companies that must report financial results to equity markets.

While OSaaS does not satisfy the purists who follow Stallman, it does offer a compelling value proposition to CIOs who are tasked with minimizing capital expenditures and posturing their IT organization for flexibility. Moreover, as TechCrunch’s Peter Levine continues to be optimistic about this model because it enables much faster time-to-use than on-site solutions he also notes “staggering results across the software landscape” (Levine, 2014).

CREATING VALUE

GitHub was founded in 2008 and has grown to over seven million members of the open-source community to collaborate on software development. It's the go-to site for open-source projects. Not everyone is engaged in commits and pushes—the most common tasks of open-source programmers (GitHub, 2014a). Lauren Orsini of ReadWrite.com observes “one of the main misconceptions about GitHub is that it's a development tool, as much a part of coding as computer languages and compilers. However, GitHub itself isn't much more than a social network, like Facebook or Flickr. You build a profile, upload projects to share and connect with other users by 'following' their accounts. And while many users store programs and code projects, there's nothing preventing you from keeping text documents or other file types in your project folders to show off” (Readwrite, 2014).

Contrary to misconceptions, GitHub does not interfere with contributors' IP. Their terms of service enable users to keep their IP, if they wish as indicated in their online terms and conditions, “We claim no intellectual property rights over the material you provide to the Service. Your profile and materials uploaded remain yours” (GitHub, 2014b). However, the Github community encourages collaboration, transparency and discourages proprietary ownership.

After the dot-com crash, VCs began vetting software vendors more thoroughly, with particular attention to their business models. Alternative sources of funding—such as advertising—fell out of favor because the math just did not add up. However, models that offer business solutions, add-value, and innovate with licenses have been seen in the most favorable light. Between 2004-05, nearly \$400 million was invested by VCs into OSS models (Bloomberg BusinessWeek, 2005). This trend has continued as major Silicon Valley brands also invest in vendors with the OSS model. For example, HP invested \$50 million in Hortonworks and Intel invested \$740 million in Cloudera (Henschen, 2014; Hesseldohl, 2014). IP is a huge point of contention between proprietary vendors and their open-source counterparts—mostly because there is value in retaining IP rights. It appears that IP is becoming less important than finding the right business model.

By 2012, the Linux operating system company Red Hat reported earnings of one billion dollars. That ushered in the perceived legitimacy of free software, but some analysts note that the “overwhelming success of Linux is disproportionate to the performance of Red Hat” (Brandel, 2010). The OSS operating system is given away at no cost by Red Hat. They provide maintenance, support, and installation for a fee. When this worked, new ventures were formed in the image and likeness of RedHat, such as XenSource, MySQL, Revolution Analytics, Ubuntu, Cloudera, MapR and SugarCRM (Brandel, 2010).

Mary Brandel of InfoWorld feels that the “key point of failure” with Red Hat's business model does not generate enough cash flow for ongoing investments. She says that the “consequence of the model is minimal product differentiation resulting in limited pricing power and corresponding lack of revenue.” She continues stating, “product roadmaps and requirements are often left to a distributed group of developers. Unless a company employs a majority of the inventors of a particular open-source project, there is a high likelihood that the project never gains traction or another company decides to create a fork of the technology” (Brandel, 2010).

VCs invest in companies that have large addressable markets and can move quickly to solve a problem that others have not. Solving the problem typically results in greater productivity or cash flow for the customer – ideally with a lower TCO and lower adoption costs. A fair

exchange of value between vendor and customer is required to enable sustainable software businesses to emerge and grow. Scaling of a business requires capital, which should be generated from cash flow, not just external investment. Vendors have a responsibility not simply to see the product, but the long-term vision that the customer can also invest in.

An IT professional who downloads free software typically tests it and finds bugs. CIOs often look to those technical specialists for an assessment at the most concrete level and open-source is becoming more main stream. Computerworld conducted a survey of 143 IT professionals. Cost savings was cited as the top benefit of OSS, among 80 percent of respondents. 61 percent said OSS is seen as less of a risk in recent years (Brandel, 2010). To take advantage of this increasing level of comfort with OSS, marketers are innovating around ways to make money.

Bill Miller, CIO of Broadcom, stated, “you can afford a little more innovation, a little more risk, so to speak.” Miller continued, “Because you’re constantly innovating, you’re constantly pushing the envelope and testing products and testing you against those limits. Engineers are more willing to do that, take corrective action, iterate through and move forward” (Lebeaux, 2014). All this effort is making some technology decision-makers more concerned about the value offered by vendors of proprietary software—especially when the customer is locked into a relationship.

Profit margins of these vendors are significantly higher than those with open-source models. Oracle had \$38 billion in revenues for its 2014 fiscal year, with gross profits of \$31 billion – or 81.0 percent. In the last three years its gross margins ranged from 78.4 to 81.0 percent, and its net income ranged from 26.8 to 29.3 percent (Yahoo finance, 2014b). SAP and Microsoft also had impressive revenue and net income. SAP’s revenue was \$23 billion with 19.7 percent net income, while Microsoft has \$86 billion in revenue with 25.4 percent net income (Yahoo finance, 2014a, 2014d). This is in comparison to the performance of open-source leader Red Hat where their total revenue was \$1.5 billion in 2014, with net income of 11.6 percent (Yahoo finance, 2014c). Red Hat developed the original open-source business model around the Linux operating system.

Levine was CEO of XenSource, which offered enterprise-grade virtual infrastructure solutions and was one of the more successful open-source companies. Citrix acquired Xen and entered the datacenter and desktop virtualization markets (Citrix, 2014). One of his biggest concerns is that product roadmaps are not easy to manage with distributed developers and successful open-source projects have code that other companies want to leverage to generate revenue. “Every major software and hardware company leveraged our code base with nearly zero revenue coming back to us. We had made the product so easy to use and so important that we had out-engineered ourselves. Great for the open-source community, not so great for us” (Levine, 2014).

Red Hat charges for maintenance, support, and installation, while making Linux available at no cost. When their model appeared to be viable, other companies emerged with similar models – XenSource, MySQL, Revolution Analytics, SugarCRM, and Ubuntu – to name a few. Andreesen Horowitz, a partner of Peter Levine and a venture capital firm in Silicon Valley, observed “the overwhelming success of Linux is disproportionate to the performance of Red Hat. Great for open-source, a little disappointing for Red Hat.” Red Hat has no significant competition in the Linux market. If there were more money to be made, other competitors would have jumped in to compete for their share. As a result, Red Hat has nobody to differentiate itself from. With an open-source model, it also has few options when it comes to pricing of its

licenses. Levine concludes “it’s nearly impossible to invest properly in product development, support or sales the way that companies like Microsoft or Oracle or Amazon can.” (Levine, 2014). This is the primary challenge with open-source revenue models—they cannot generate enough cash flow finance a rapid growth trajectory. Scaling requires a sizable available market, as well as the resources, to invest in the development, and hiring marketing and sales staff. A product roadmap must address users’ evolving needs, while marketers take the time to understand those needs.

Customers naturally seek the cheapest alternative, but marketing and sales people are required to educate buyers. By differentiating their offering, a vendor can also imply that their product mitigates the risks associated with competitors. Many open-source entrepreneurs do not have the capital or marketing team to identify, understand and address business problems. Therein lies the value gap between proprietary software and open-source business models.

CONCLUSION

Proprietary software vendors would be smart to seek a safe middle-ground by lowering their target net income to around 12-18 percent and focusing on rethinking contracts and customer service, so customers do not feel locked-in. This may not be good news for investors, but it would likely impede the progress of open-source as a threat. Those companies wishing to succeed with an OSS model should consider a combination of revenue models that enable them to capture value from the software itself, as well as drive in additional revenue streams. This is more of a hybrid approach, but addresses concerns of IT decision-makers.

Open-source will continue to be a force in the marketplace, but revenue and business model innovation are key to making it more viable as a way to create long-term customer value.

REFERENCES

- Achtelig, M. (2014). Costs. Retrieved 06.12.2014, from <http://www.indoition.com/en/services/costs-prices-software-documentation.htm>
- Acquia. (2009). Acquia to Provide High Availability Managed Hosting for Enterprise Drupal Sites. Retrieved 06.12.2014, from <http://www.acquia.com/about-us/newsroom/press-releases/acquia-provide-high-availability-managed-hosting-enterprise-drupal>
- Actuate. (2014). Actuate Launches BIRT iHub F-Type – Free Server for Over 3.5 Million BIRT Developers Building Data-driven Reports, Visualizations and Applications. Retrieved 05.12.2014, from <http://www.actuate.com/company/buzz/press-release/?articleid=33003>
- Adobe. (2014a). Enterprise Support Program. Retrieved 06.12.2014, from <http://helpx.adobe.com/support/programs/enterprise-support/program-details.html>
- Adobe. (2014b). Stay productive with help from Adobe support specialists. Retrieved 06.12.2014, from <https://www.adobe.com/support/programs/individuals/>
- Agile Modeling. (2014). Best Practices for Agile/Lean Documentation. Retrieved 06.12.2014, from <http://www.agilemodeling.com/essays/agileDocumentationBestPractices.htm>
- AlternativeTo. (2014). BountySource alternatives for all platforms. Retrieved 06.12.2014, from <http://alternativeto.net/software/bountysource/>
- Altiscale. (2014). The First Purpose-Built Hadoop as a Service. Retrieved 06.12.2014, from <https://www.altiscale.com/hadoop-as-a-service/>

- Apache. (2014a). The Apache Software Foundation. Retrieved 06.12.2014, from <http://www.apache.org/foundation/thanks.html>
- Apache. (2014b). The Apache Software Foundation Sponsorship Program. Retrieved 06.12.2014, from <http://www.apache.org/foundation/sponsorship.html>
- Arduino. (2014). What is arduino? Retrieved 06.12.2014, from www.arduino.cc
- AVC. (2006). My Favorite Business Model. Retrieved 06.12.2014, from http://avc.com/2006/03/my_favorite_bus/
- Basu, S. (2013). Ruby on Rails Study Guide: The History of Rails. Retrieved 06.12.2014, from <http://code.tutsplus.com/articles/ruby-on-rails-study-guide-the-history-of-rails--net-29439>
- Bell, S. (2006). Where Did Firefox Come From? Retrieved 05.12.2014, from <https://web.archive.org/web/20110623034401/http://weblogs.mozillazine.org/ben/archive/s/009698.html>
- Blackhat. (2014a). Blackhat: 2013 Delegate Snapshot. Retrieved 06.12.2014, from http://blackhat.com/docs/us-14/sponsors/Black_Hat_USA_2014_Prospectus.pdf
- Blackhat. (2014b). Welcome to Blackhat USA 2014. Retrieved 06.12.2014, from <https://www.blackhat.com/us-14/registration.html>
- Bloomberg. (2014a). Bloomberg video, news and photos. Retrieved 06.12.2014, from <http://www.bloomberg.com/content-service/>
- Bloomberg. (2014b). Pricing Services Ensure Transparency and Consistency. Retrieved 06.12.2014, from <http://www.bloomberg.com/enterprise/data/pricing-services/>
- Bloomberg businessweek. (2005). Open Source: Now It's an Ecosystem. Retrieved 06.12.2014, from <http://www.businessweek.com/stories/2005-10-02/open-source-now-its-an-ecosystem>
- Bountysource. (2014). Get features added and bugs fixed. Retrieved 06.12.2014, from <https://www.bountysource.com/>
- Brandel, M. (2010). Open source software's hidden snags: users say savings and other benefits are easy to reel in, but there are hitches. *InfoWorld*. Retrieved 07.12.2014, from <http://www.infoworld.com/article/2626903/open-source-software/open-source-software-s-hidden-snags.html>
- Casadesus-Masanell, R., & Ricart, J. E. (2011). How to design a winning business model. *Harvard Business Review*, 89(1/2), 100-107.
- Citrix. (2014). Citrix Unveils End-to-End Virtualization Strategy. Retrieved 07.12.2014, from <http://www.citrix.com/go/news/xensource.html>
- Cloudera. (2014a). Fins a class near you. Retrieved 06.12.2014, from <http://cloudera.com/content/cloudera/en/training/course-listing.html?course=data-science&loc=all>
- Cloudera. (2014b). The Leading Open Source Analytic Database for Apache Hadoop Retrieved 06.12.2014, from <http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html>
- Codebender. (2014). Codebender: online development & collaboration platform for Arduino users, makers and engineers Retrieved 06.12.2014, from <https://codebender.cc/>
- Coding horror. (2008). Tivoization and the GPL. Retrieved 06.12.2014, from <http://blog.codinghorror.com/tivoization-and-the-gpl/>
- Crade, B., & Fulks, K. (2011). Top Trends in Technology. Retrieved 06.12.2014, from https://www.blackbaud.com/files/resources/downloads/WhitePaper_TopTrendsInTechnology.pdf

- Cray. (2014). RSS Content: Cray adds New Advanced Analytics Solution to its Big Data Product Portfolio. Retrieved 06.12.2014, from <http://investors.cray.com/phoenix.zhtml?c=98390&p=RssLanding&cat=news&id=1978291>
- Creative commons. (2014). Attribution 2.5 Generic. Retrieved 06.12.2014, from <http://creativecommons.org/licenses/by/2.5/>
- CRM. (2014). SugarCRM Software Reviews and Market Coverage. Retrieved 06.12.2014, from <http://www.crmsearch.com/sugar-crm.php>
- Dinu, J. (2014). Modern Education in a Post-Modern World. Retrieved 06.12.2014, from <http://www.slideshare.net/jonathandinu/modern-education-in-a-postmodern-world>
- Director-EE. (2014a). Featured Developers. Retrieved 06.12.2014, from <http://director-ee.com/developers/featured>
- Director-EE. (2014b). The Official ExpressionEngine and CodeIgniter Job Board. Retrieved 06.12.2014, from <http://director-ee.com/jobs>
- Drupal. (2014a). Chat with the Drupal Community on IRC. Retrieved 06.12.2014, from <https://www.drupal.org/irc>
- Drupal. (2014b). Why Choose Drupal? Retrieved 06.12.2014, from <https://www.drupal.org/>
- Eames, J. (2014). How to run a tech conference. Retrieved 06.12.2014, from <http://blog.pluralsight.com/running-a-tech-conference>
- EllisLab. (2014a). ExpressionEngine Core 2.9.2. Retrieved 06.12.2014, from <https://store.ellislab.com/#ee-core>
- EllisLab. (2014b). Why ExpressionEngine? Retrieved 06.12.2014, from <https://ellislab.com/expressionengine/features>
- Fago. (2012). Semantic content enhancements with Drupal, Apache Stanbol and VIE.js Retrieved 06.12.2014, from <http://wolfgangziegler.net/semantic-content-enhancements-drupal-apache-stanbol-viejs>
- Ferreira, E., & Tanev, S. (2009). How companies make money through involvement in open source hardware projects. *Open Source Business Resource*(February 2009).
- Ferrel, D. (2013). The Conference Industry is Booming, and It Is Only Getting Bigger Retrieved 06.12.2014, from <http://skift.com/2013/08/04/the-conference-industry-is-booming-and-it-is-only-getting-bigger/>
- Forrest, C. (2014). Salil Deshpande: Software Engineer. Venture Capitalist. Open Source Investor. Retrieved 05.12.2014, from <http://www.techrepublic.com/article/salil-deshpande-software-engineer-venture-capitalist-open-source-investor/>
- Foxen, D. (2014). Adobe Licensing Quick Guide 2014. Retrieved 06.12.2014, from <http://www.itassetmanagement.net/2014/07/08/adobe-licensing-quick-guide-2/>
- Galperin, E. (2013). Open-source cannot live on donations alone. Retrieved 06.12.2014, from <http://www.binpress.com/blog/2013/04/14/open-source-cannot-live-on-donations-alone/>
- GitHub. (2014a). GitHub is the best place to share code with friends, co-workers, classmates, and complete strangers. Retrieved 06.12.2014, from <https://github.com/about>
- GitHub. (2014b). GitHub Terms of Service. Retrieved 06.12.2014, from <https://help.github.com/articles/github-terms-of-service/>
- GitHub. (2014c). Job description. Retrieved 06.12.2014, from <https://jobs.github.com/positions>
- GitHub. (2014d). Job Posting Retrieved 06.12.2014, from <https://jobs.github.com/post#create>
- GNU. (2014a). Tips for Documentation Strings. Retrieved 06.12.2014, from <https://www.gnu.org/software/octave/doc/interpreter/Documentation-Tips.html>

- GNU. (2014b). What is free software? Retrieved 06.12.2014, from <https://www.gnu.org/philosophy/free-sw.html>
- Google support. (2014a). Getting started as a YouTube partner: What kind of content can I monetize? Retrieved 06.12.2014, from <https://support.google.com/youtube/answer/2490020?hl=en>
- Google support. (2014b). What is the YouTube Partner Program? Retrieved 06.12.2014, from <https://support.google.com/youtube/answer/72851?hl=en>
- Gordon-McKeon, S. (2014). Open Source Comes to Campus/Curriculum/Tools/transcript. Retrieved 06.12.2014, from <https://openhatch.org/wiki/OSCTC/Tools/transcript>
- GraphLab. (2013). GraphLab Announces \$6.75 million in Funding from Madrona Venture Group and NEA to Fuel the Fastest Graph Analysis for Modern Datasets Retrieved 06.12.2014, from <http://graphlab.com/company/press/release-2014-05-14.html>
- GraphLab. (2014). GraphLab Open Source Retrieved 06.12.2014, from <http://graphlab.org/projects/index.html>
- Henschen, D. (2014). HP Invests \$50M In Hortonworks: Hadoop Bet. Retrieved 06.12.2014, from [http://www.informationweek.com/big-data/software-platforms/hp-invests-\\$50m-in-hortonworks-hadoop-bet/d/d-id/1297542](http://www.informationweek.com/big-data/software-platforms/hp-invests-$50m-in-hortonworks-hadoop-bet/d/d-id/1297542)
- Hesseldohl, A. (2014). Intel Takes Massive \$740 Million Stake in Cloudera. Retrieved 06.12.2014, from <http://recode.net/2014/03/31/intel-takes-massive-740-million-stake-in-cloudera/>
- Hollaar, L. A. (2002). Copyright of Computer Programs. Retrieved 05.12.2014, from <http://digital-law-online.info/lpdi1.0/treatise17.html>
- IndieGoGo. (2014). JIBO, World's First Family Robot. 4,800 pre-sold! Retrieved 06.12.2014, from <https://www.indiegogo.com/projects/jibo-world-s-first-family-robot-4-800-pre-sold>
- Jibo. (2014). Meet JIBO, The World's First Family Robot. Retrieved 06.12.2014, from <http://www.myjibo.com/>
- Kaufman, L. (2013). Media Outlets Embrace Conferences as Profits Rise. Retrieved 06.12.2014, from http://www.nytimes.com/2013/10/28/business/media/media-outlets-embrace-conferences-as-profits-rise.html?pagewanted=all&_r=1&
- Kepley, T. (2011). How we almost lost Doom 3 and id for good. Retrieved 06.12.2014, from <http://opensource.com/life/11/11/we-almost-lost-doom-3-id-for-good>
- Koetsier, J. (2013). 450M lines of code say large open source and small closed source software projects are worst quality. Retrieved 06.12.2014, from <http://venturebeat.com/2013/05/09/450-million-lines-of-code-say-large-open-source-and-small-closed-source-software-projects-are-worst-quality/>
- Kot-in-action. (2014). Steel Storm: Burning Retribution (with DLC). Retrieved 06.12.2014, from https://shop.kot-in-action.com/product_info.php?products_id=30&osCsid=pldkadgcl98is4gonou1fvhr47
- Lebeaux, R. (2014). Technology leadership quotes from the October 'CIO Decisions' e-zine. Retrieved 07.12.2014, from <http://searchcio.techtarget.com/opinion/Technology-leadership-quotes-from-the-October-CIO-Decisions-e-zine>
- Levine, P. (2014). Why There Will Never Be Another RedHat: The Economics Of Open Source. Retrieved 06.12.2014, from <http://techcrunch.com/2014/02/13/please-dont-tell-me-you-want-to-be-the-next-red-hat/>
- Linux foundation. (2014). Individual Membership. Retrieved 06.12.2014, from <https://www.linuxfoundation.org/about/join/individual>

- Malone, T., Weill, P., Lai, R., D'Urso, V., Herman, G., Apel, T., & Woerner, S. (2006). Do some business models perform better than others? *MIT*. from <http://seeit.mit.edu/Publications/BusinessModelsPerformance12July2006.pdf>
- Mohan, M. (2014). How Much Traffic Do You Need To Make \$100,000 With Google AdSense. Retrieved 06.12.2014, from <http://www.minterest.org/how-much-traffic-do-you-need-to-make-money/>
- Monty Says. (2013). Business Source (A software license with some Open Source aspects) Retrieved 06.12.2014, from <http://monty-says.blogspot.de/2013/06/business-source-software-license-with.html>
- MS-Office. (2014). Buy Office 365. Retrieved 06.12.2014, from <http://office.microsoft.com/en-001/buy-microsoft-office-2013-suites-and-office-365-subscriptions-FX102886268.aspx>
- Netscape. (1998). Netscape Announces Plans to Make Next-Generation Communicator Source Code Available Free on the Net. Retrieved 05.12.2014, from <http://web.archive.org/web/20070401072854/http://wp.netscape.com/newsref/pr/newsrele ase558.html>
- Open Ocean. (2014). Open Ocean Helping European Entrepreneurs build Disruptive Software Companies Retrieved 06.12.2014, from http://www.openoceancapital.com/about_us
- Open Office. (2002). What is OpenOffice.org? Retrieved 06.12.2014, from <http://www.openoffice.org/FAQs/faq-overview.html#1>
- Oracle. (2014). Oracle E-Business Suite Applications Global Price List Software Investment Guide. Retrieved 06.12.2014, from <http://www.oracle.com/us/corporate/pricing/applications-price-list-070574.pdf>
- Oram, A. (2011). How Free Software Contributed to the Success of Steve Jobs and Apple. Retrieved 06.12.2014, from <http://radar.oreilly.com/2011/08/how-free-software-contributed.html>
- Park, K. (2014). Data Communication and Computer Networks Retrieved 05.12.2014, from <https://www.cs.purdue.edu/homes/park/>
- Pearson. (2014). Purchase Test Vouchers - Cloudera Retrieved 06.12.2014, from <http://www.pearsonvue.com/vouchers/pricelist/cloudera.asp>
- Peters, K. (2006). E-retailing's high margins are getting pinched by marketing costs. Retrieved 06.12.2014, from <http://ecommerce-news.internetretailer.com/retailing/Average-Profit-Margin>
- Popp, K. (2014). Revenue Models in the software industry. Retrieved 05.12.2014, from <http://drkarlpoppp.com/revenuemodelssoftwareindustry.html#.VIIfZMknmSp>
- Princeton. (2014). Obfuscated code. Retrieved 06.12.2014, from http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Obfuscated_code.html
- Project-Apache. (2014). Web-framework. Retrieved 06.12.2014, from <http://projects.apache.org/indexes/category.html#web-framework>
- PWC. (2007). Software Pricing Trends: How Vendors Can Capitalize on the Shift to New Revenue Models. Retrieved 07.12.2014, from http://www.pwc.com/en_us/us/technology-innovation-center/assets/softwarepricing_x.pdf
- Readwrite. (2014). GitHub For Beginners: Don't Get Scared, Get Started. Retrieved 06.12.2014, from <http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1>
- Redhat. (2014). Associate fedoras Retrieved 06.12.2014, from <https://redhat.corpmerchandise.com/staticpage.aspx?spid=4810>

- Rickert, J. (2014). Quick History 2: GLMs, R and large data sets. Retrieved 06.12.2014, from <http://blog.revolutionanalytics.com/big-data/page/2/>
- Salonek, T. (2014). 13 Crazy Expensive Software Products. Retrieved 05.12.2014, from <http://www.intertech.com/Blog/13-expensive-software-products/>
- SBIR. (2014a). Open/Future Solicitations. Retrieved 06.12.2014, from <http://www.sbir.gov/solicitations>
- SBIR. (2014b). The SBIR Program. Retrieved 06.12.2014, from <http://www.sbir.gov/about/about-sbir>
- Scherschel, F. A. (2013). Free's a crowd-funding. Retrieved 06.12.2014, from <http://www.h-online.com/open/features/Free-s-a-crowd-funding-1847399.html>
- Shevat, A. (2010). Ad Supported Open Source Site/Blogs – Poor Revenue Model. Retrieved 06.12.2014, from <http://spacebug.com/ad-supported-open-source-siteblogs-poor-revenue-model/>
- Signal v. Noise. (2013). Our new job board: WeWorkRemotely.com. Retrieved 06.12.2014, from <https://signalvnoise.com/posts/3671-our-new-job-board-weworkremotelycom>
- Simms, M. (2009). Our new way to meet the LGPL. Retrieved 06.12.2014, from <http://web.archive.org/web/20090220234542/http://blog.linuxgamepublishing.com/2009/02/08/our-new-way-to-meet-the-lgpl/>
- Software pluralism. (2014). Business Models. Retrieved 06.12.2014, from <http://www.law.washington.edu/Ita/swp/Institutions/businessmodels.html>
- Spacebug. (2014). Get Involved. Retrieved 06.12.2014, from <http://spacebug.com/get-involved/>
- Stallman, R. (2010). The JavaScript Trap. Retrieved 05.12.2014, from <https://www.gnu.org/philosophy/javascript-trap.html>
- Steel storm. (2014a). End User License Agreement Retrieved 06.12.2014, from http://www.steel-storm.com/ss_license.html
- Steel Storm. (2014b). Steel Storm 2 crowdfunding campaign. Retrieved 06.12.2014, from <http://two.steel-storm.com/>
- Steele, R. D. (1996). Open source intelligence: What is it? why is it important to the military. *American Intelligence Journal*, 17(1), 35-41.
- Stieben, D. (2014). What is Creative Commons, and Should You Use it? Retrieved 06.12.2014, from <http://www.makeuseof.com/tag/creative-commons-use/>
- Syspro. (2014). SYSPRO Education (Training and certification) Retrieved 06.12.2014, from <http://www.syspro.com/services/education>
- Tigon. (2014). Tigon: 100% open source low latency, high-throughput streaming technology. Retrieved 06.12.2014, from <http://tigon.io/>
- Velazco, C. (2011). Microsoft Sold 450 Million Copies Of Windows 7. Retrieved 06.12.2014, from <http://techcrunch.com/2011/09/13/microsoft-sold-450-million-copies-of-windows-7/>
- Waterstechnology. (2014). News: Free trial or subscription. Retrieved 06.12.2014, from <http://www.waterstechnology.com/buy-side-technology/news/1606799/bloomberg-opens-master-database-sets-usd11m-price-tag>
- Wex. (2014). Intellectual property. Retrieved 05.12.2014, from http://www.law.cornell.edu/wex/intellectual_property
- Widenius, M. M., & Nyman, L. (2013). Introducing “Business Source”: The Future of Corporate Open Source Licensing? *Technology Innovation Management Review*.

- Woods, D. (2013). A Quick Guide To Choosing The Right Way To Use Hadoop. Retrieved 06.12.2014, from <http://www.forbes.com/sites/danwoods/2013/12/16/a-quick-guide-to-choosing-the-right-way-to-use-hadoop/>
- Wordpress. (2014a). Do you sometimes struggle to understand all the features that WordPress 4.0 has to offer? Retrieved 06.12.2014, from <https://interconnectit.com/products/wordpress-user-guide/>
- Wordpress. (2014b). Donate. Retrieved 06.12.2014, from <http://wordpressfoundation.org/donate/>
- Wordpress. (2014c). Projects. Retrieved 06.12.2014, from <http://wordpressfoundation.org/projects/>
- Yahoo finance. (2014a). Microsoft Corporation (MSFT)-Income Statement. Retrieved 06.12.2014, from <http://finance.yahoo.com/q/is?s=MSFT+Income+Statement&annual>
- Yahoo finance. (2014b). Oracle Corporation (ORCL)- Income Statement. Retrieved 07.12.2014, from <https://finance.yahoo.com/q/is?s=ORCL+Income+Statement&annual>
- Yahoo finance. (2014c). Red Hat, Inc. (RHT)- Income Statement. Retrieved 07.12.2014, from <http://finance.yahoo.com/q/is?s=RHT+Income+Statement&annual>
- Yahoo finance. (2014d). SAP SE(SAP)- Income Statement. Retrieved 07.12.2014, from <https://finance.yahoo.com/q/is?s=SAP+Income+Statement&annual>
- Zipfian academy. (2014a). Alumni of Zipfian Academy. Retrieved 06.12.2014, from <http://www.zipfianacademy.com/alumni/>
- Zipfian academy. (2014b). Become a data scientist. Retrieved 06.12.2014, from <http://www.zipfianacademy.com/>
- Zipfian academy. (2014c). Data Science 12-Week Immersive Retrieved 06.12.2014, from <http://www.zipfianacademy.com/programs/#datascienceprogram>