From Open Source to long-term sustainability: Review of Business Models and Case studies

Victor Chang, Hugo Mills, Steven Newhouse

OMII-UK, University of Southampton

Abstract

This paper presents several case studies to demonstrate how open source software can achieve long-term sustainability by adopting the relevant business models. The objectives of this paper are to study the different models, processes, and legal/licence requirements that have been successful for such transformations. We classify the business models used in the open source area into five types: (a) Support Contracts; (b) Split Licensing; (c) Community; (d) Valued-added closed source; (e) Macro R&D Infrastructure. Each model's strengths and weaknesses are discussed. The five business models detailed in this paper are the most common and arguably the most successful methods of generating revenues from open source software. Those in the e-Science community are encouraged to consider these methods for longer term sustainability.

1. Introduction

1.1 Open Source definition

The term "Open Source" describes the principles, and methodologies of promoting open access to the production and design process for various goods, products, resources and technical conclusions or advice [16]. The term is most commonly applied to the source code of software that is made available to the general public with either relaxed or non-existent intellectual property restrictions.

Therefore, open source software (OSS) is computer software whose source code is available under a licence or arrangement in the public domain that permits users to study, change, and improve the software, and to redistribute it in modified or unmodified form [17]. OSS projects consist of people working together to create a particular piece of software, for which (1) user support and (2) development activities are the essential criteria [4]. However, the emphasis of this paper is to study various business models and propose those ones achieving long-term sustainability for open source software projects.

1.2 Proprietary Software

In contrast to open source software, proprietary software normally requires payment for licences or services, and disallows examination of the source code and restricts or prohibits modification and distribution of the code. It is a popular model adopted by commercial organisations such as Microsoft, Adobe and MATLAB, and has generated revenues and maintained momentum of software sales. Apart from high cost as a likely issue, commercial software usually has very strict licence schemes, and users are subject to legal requirements if installed, copied or modified inappropriately outside the licence or intellectual property protections. In contrast, open source software (OSS) allows users to obtain the source code and install, copy, modify and redistribute the source code with few, if any, restrictions.

1.3 Licences for OSS

Currently there are more than 50 open source licences certified by the Open Source Initiative (OSI). Below are the most commonly-used OSS licences:

- The GNU General Public Licence (GPL)

- The GNU Lesser General Public Licence (LGPL)

- Modified BSD (Berkeley Software Distribution) Licence (new BSD)

- Apache Licence

- Mozilla Public Licence (MPL)

The main difference between these licences is the extent of code control – how it can be combined with other software. Taking the BSD and GPL licences as examples, the BSD licence allows integrations between OSS and closed-source code which may then be sold under a conventional "closed source" or proprietary licence. On the other hand, the GPL only accepts integrations with GPL-licenced software [6]. Licencing issues play an influential role to a new project, as the decision to which open source licence it use may express and shape the development goals of the project [9].

2. Software Business Models

2.1 Substainability

Organisational sustainability refers to the long-term maintenance of an organisation, particularly if securing funding, resources, operations and clients. In order to maintain sustainability, OSS organisations must adopt a model for its long-term existence, which is dependent on the organisational goals, operational requirements, sources and types of funding and influence of their stakeholders or clients.

2.2 Open Source Models

The JISC [8] classifies OSS organisations into four sustainability models: (a) community model; (b) subscription model; (c) commercial model; and (d) central support model. The community model is one where the costs of sustaining the product or service are covered by building a community of users and industry partners who agree to cooperate on development work and maintenance. Examples of this model are Apache and the Globus Alliance. The subscription model requires users to pay subscription costs to an external body in order to obtain central maintenance and support. SAKAI and Red Hat are examples of this model. In a commercial model, users choose to adopt and pay for a 'commercialised' version of a piece of software, normally to gain guaranteed support, maintenance and service models. The central support model refers to a central body that provides robust releases and support for open source products that are of strategic importance to its community, and OMII-UK is an example of this model.

2.3 Commercial Models

Forfas and the International Data Corporation [5, 7] define a set of models more relevant to proprietary software, or OSS organisations planning to move into the commercial field. Owners or prospective organisations of software intellectual property may use one or more of the following methods for generating income:

(1) Require a subscription fee for using the product. This is a conventional proprietary software model where the right to use the application should be paid. This is referred to as "Product" in the IDC commercial model [7].

(2) Sell paid-for services. Services include basic support, on-site support and premium support, the latter of which includes troubleshooting, repair, debug and maintenance of the systems or the applications. This is referred to as "Services" in the IDC commercial model.

(3) Make a margin for reselling other companies' intellectual property. Some organisations sell customers a commercial licence that allows them to use the product without being covered by GPL. This is known as "Resale" in the IDC commercial model, or Split-Licencing model in this paper, and further details will be described in the Section 3.2.
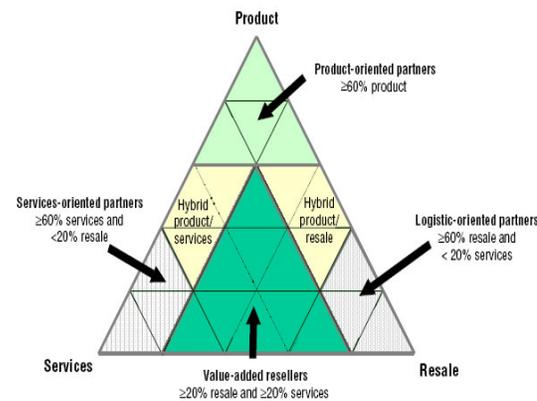
Figure 1: Commercialisation model, IDC 2001 [7]

Figure 1 shows the range of options for commercial activities in the IDC model. The triangles indicate the best business model that an organisation can best fit into. OSS organisations can be interpreted as "Product-oriented", "Service-oriented" and "Logistic-oriented" if they fit into "Product", "Service" and "Resale" respectively. OSS organisations that operate between Product and Services, are categorised as "Hybrid Product Services" and those operate between Product and Resale are categorised as "Hybrid Product Resale". The model recommends OSS organisations to integrate these three hybrids of models to achieve sustainability and this new hybrid is known as "Valued-Added Resellers", shown in the green region of the model where a minimum of 20% of business activities focus on Resale and a minimum of 20% focus on Services with the remaining percentages focus on the Product.

2.4 Model Classifications

Each OSS organisation requires a community – typically substantially unpaid – in order to provide support, maintenance and growth. The main exception being the *Split-Licencing* model described in Section 2.3.3. Managed by a variety of governance procedures, a community of users and developers normally work together to either report bugs, investigate problems, fix errors, share knowledge or improve functionality of the software. Such a community-based organisation which does not have a specific funding body but instead relies on donations and enthusiasm, is known as *Community* model. The Apache Software Foundation is the best example of such a model and will be discussed in Section 3.3 of this paper.

Referring to JISC subscription and central support models [8], both can be categorised as a *Support Contracts* business model. The levels of support can be generically divided into three levels: basic support (subscription), middle-class support and premium support (on-site and 24/7). Red Hat is the best example of this model and is described further in Section 3.1.

OSS organisations exploring how to commercialise their work and to operate like a small and medium business fall into a model called *Valued-added closed source*. In such a case their source code is not released and users are required to purchase the software or licence.

JISC points out that the central support model is often an interim solution while an organisation's business model is still being developed. However, such organisations are mainly research and development-based, and involved in high-level complex technical challenges, with collaborations and partnerships between local and global partner institutes. Such a model is classified as a *Macro R&D Infrastructure* model, where the funding initially comes from a government's research grant, and sources of funding will come from research grants of local or international partner institutes.

3

3. Case Studies

Based on previous discussions, we classify all OSS organisations into five models: (a) Support Contracts; (b) Split Licensing; (c) Community; (d) Valued-added closed source; (e) Macro R&D Infrastructure. A case study for each model is then described as below.

3.1 Support Contracts: Red Hat

Red Hat [19] adopts a support-based subscription model for its open software business. This means customers pay for Red Hat Enterprise Linux, which is a tested, certified and stable version of its free and community-based Fedora Linux, thus ensuring a high level of deployment, scalability and security. Apart from this, support subscription allows users to download and install security patches, and provides 24/7 online and phone customer support. Users can get technical account management, development support, premium developer packages, discounted commercial software (JBoss), as well as bug fixes and troubleshooting for users' local nodes. This premium service is provided at an additional cost to the basic service fees. In addition, Red Hat Linux Certification is one of the best well-known certification programmes in the open source arena. In conclusion, Red Hat obtains revenues from:

- Subscriptions from Red Hat Enterprise Linux (RHEL) per system or server basis;
- Subscriptions from commercial open source applications per system or server basis;
- System/Architecture management services;
- Support services;
- Red Hat Certification and Training.

3.2 Split Licensing: MySQL

MySQL [10] is a Swedish-based organisation specialising in database development, which comprises a free, community edition and a commercial, certified "server edition". MySQL server is a popular database in the open source field, and it has been deployed in many websites and database applications. MySQL Community Edition is available under the open source GPL license and has both stable and beta software releases.

Apart from receiving profits from premium customer support, MySQL primarily obtains its revenues from selling customers a commercial license that allows them to use the product without being covered by GPL. Consequently, these customers can include MySQL in their own products for resale. This licence is designed for organisations that do not want to release the source code for their applications or those who do not wish to comply with the GNU GPL. Examples of these include:

- Selling software that includes MySQL to customers who install the software on their own machines;
- Selling software that requires customers to install on their own machines;
- Building a system that includes MySQL and selling that system to customers.

3.3 Community: Apache Software Foundation

The Apache Software Foundation (ASF) [1, 2] is a non-profit corporation to support Apache software projects, including the Apache HTTP Server, which was started in 1994 and was the first software developed from Apache Group. The ASF was formed from the Apache Group and incorporated in Delaware, USA, in June 1999.

The Apache Software Foundation is a decentralised community of developers. All their produced software and all software contributions to ASF, are distributed under the Apache Licence, which requires preservation of the copyright notice and disclaimer. Unlike the GPL, the Apache Licence allows the use and distribution of the source code in both free/open source and proprietary/closed source software. In this way, the Apache license is similar to the modified BSD license.

Along with Red Hat/Fedora Linux, ASF is one of the largest OSS organisations, as evidenced by the 66.9 million sites using Apache as the web server [12]. Backed up by a large and active community, ASF has vast resources of OSS projects and developers – those who contribute and get accepted can become members. Although this business model best fits to the original open source philosophy, its sole but critical weakness is that it relies on the community donation and this sustainability model is applicable to large OSS organisations.

3.4 Value-added close source: XandrOS

Succeeded from its pioneer, Corel Linux, XandrOS [20] was founded in 2001, with the organisation goal to make easy-to-use Desktop Linux. This strategy earns them revenues mainly from its business and educational customers, particularly those based in North America. At the beginning, XandrOS operated a Split Licencing model similar to MySQL's,

where the Open Circulation product had a GPL Licence and the Commercial product came with its own licence that does not allow software redistribution without having legal permissions. From 2006 onwards XandrOS has stopped releasing the open source version and now only distributes the commercial product, which contains proprietary software and some GPL software. In their commercial business model, XandrOS adopts "pay for software product" and "pay for services" and runs the two operations in parallel. They have launched partner and investor programmes to attract further investment funds

XandrOS is therefore considered as a Value-added closed-source, because (1) they are providing "pay for software", "pay for services" and "attract investors or venture capitalists" for their business model; (2) they have added new proprietary software and improved on their functionality based on customer requirement, making themselves differing from most Linux products.

3.5 Macro R&D Infrastructure: OMII-UK

Founded in January 2006, OMII-UK [14] is funded by EPSRC through the UK e-Science Core programme. It is a collaboration between the School of Electronics and Computer Science at the University of Southampton, the OGSA-DAI project at the National e-Science Centre and EPCC, and the ᵐʸGrid project at the School of Computer Science at the University of Manchester. This partnership aims to be a leading provider of reliable interoperable and open-source Grid middleware components services and tools to support advanced Grid enabled solutions in academia and industry.

OSS development is achieved by investing in community developers to produce the functionality required by our user community. Releases from the community, alongside the products from Edinburgh and Manchester, undergo integration and testing at Southampton to produce a software release. OMII-UK also promotes community growth and knowledge transfers with international partners in the US, EU and China, and jointly develops OSS software in global collaboration.

OMII-UK is therefore presented as a Macro R&D model, as it:

- presents engineering challenges, integrating 12 different software components in a single container and provides solutions to meet demands of such challenges.
- offers a secure, robust and fully integrated

Service Oriented Architecture for academia and industry in the UK and globally.
- provides interoperable solutions and is involved in international partnership, community expansion, research and development.

4. Special case studies

4.1 XenSource: Move between business models

There are organisations that have switched business models. They are normally either in the process of business model transformation or in the process of high-level organisational changes. One such organisation is Xensource [21], which was set up in January 2005 and raised £23.5 million in the first two rounds of venture capital funding. Xensource's open source software, Xen, is a hypervisor. Xen allows a single machine, typically a server, to simultaneously host multiple different operating systems and to share resources between them, providing resource guarantees to each virtual server – a process known as virtualisation.

Before January 2005, most work was done in the Computer Laboratory, Cambridge University, where 'Community' was the best term describing their OSS project. Currently, Xensource provides two licensing models, the first one through the GPL licence, which allows users to download, install, build from source and customise for personal or organisational uses. The second licensing model is through an Enterprise Linux (mainly Red Hat and SuSE) Licence, where clients can use this software if purchasing or subscribing to these Linux distributions. Xen can be purchased independently – their first commercial software package, Xen Enterprise, was introduced in April 2006, and was based on development and improvement of Xen 3.0.

Although it is too early to say if XenSource will in the future become a "Support Contracts"-type business model, this case study illustrates that an OSS organisation should be responsive to changes and ready to evolve if such changes can benefit organisations in the long term.

4.2 National Computer Systems, Singapore: Dual business models

Achieving a dual business model requires a long-term establishment of customer relationship, and a strong reputation in product and services sustained over a significant period of time. This is applicable even if an OSS

organisation can generate improved revenues, a large number of clients and investors in the regional or global context.

National Computer Systems, Singapore [11], started in 1981 with a Macro R&D business model, as a subsidiary unit of the National Computer Board, Singapore. After becoming privatised in 1996, it first started with a valued-added closed source business model, with the Singapore government as its major client. Its services mainly include (a) computerisation and digital transformations for client organisations; (b) software outsourcing; (c) telecommunications network support and (d) application service provider. Their clients include local and global organisations in telecommunications, IT, education, energy and infrastructure.

Its business has evolved to be a dual-business model: running in parallel a support-contract model and valued-add closed source model depending on the client needs and contracts. This organisation has its overseas office in eight countries and its highest turnover net profit was S$4.9 billion (£1.623 billion) for 1997/1998 period.

4.3 Sun Microsystems and OpenJDK: Commercial organisations starting open source projects

There are more commercial organisations starting their own open source projects. The main advantages are perceived to be (1) to consolidate a stronger community; and (2) to build up more robust, reliable and user-oriented software by having more developers and testers involved. This is a different business model to OSS organisations but it is worthwhile to briefly discuss this strategy.

Java development was originally a closed-source project started in 1991. As a mainstream in Web Service and SOA, it now has a huge number of developers and a strong community. Their decision to move to a GPL licence and start up a new OpenJDK project [15] in 2006, directly benefits the OSS community – not just to test and understand Java Development Framework but also to become part of the software development and decision-making process to determine the future directions of Java.

5. Business Model Comparisons

The major advantages and disadvantages for the five OSS business models are summarised in the tabular form below:

5.1 Support Contracts

Advantages	Disadvantages
- Large organisations often require vendor support for their software and services, thus ensuring long-term sales and profits. - It provides a predictable and dependable revenue stream; - Subscription renewal rates can be very high, thus ensuring a large number of clients and contracts. - It provides different levels of support for different organizational needs. This also provides users more options.	- A lot of customers feel there is no need to pay for support since the product is open source and plenty of free information is available. - It requires an existing base of customers to support, or it needs to ensure a large number of users already available. - It is easy for some orgasnisations to clone an entire support architecture and services from an existing one, such as Oracle Unbreakable Linux.

5.2 Split Licencing

Advantages	Disadvantages
- Provides a high level of flexibility for users and organisation, which can retain both as an open source and commercialised operation. - It allows clients to use and customise the software for further sales without licensing restrictions ; - If clients' software sales include the software (such as MySQL), it increases the number of users and might increase potential sales.	- Some clients are confused with the boundary between commercial or GPL licence under the same product, particularly if they switch from using commercial support to OS support. - Any product or organisation in the entire sales chain, might be subject to licence and legal requirements if it is not guided or reviewed thoroughly.

5.3 Community

Advantages	Disadvantages
- Backed up by a large community, community effort and product can become a mainstream such as Apache. - Being portable and functional on many products or platforms and widespread of world-of-mouth, it is presented and appealed to a wider range of users and organisations. - Can become a core component in a widely used product or platform, such as Apache HTTP.	- The leading developers or donators/investors may influence its development cycle and direction. - A lot of such organisations find it difficult to sustain and often request community donations.

5.4 Value-added close source

Advantages	Disadvantages
- This is equivalent to commercialisation model where companies receive additional funds from share, investors' funds, sales commission, retailers and so on. - May generate much higher revenues if targeting the right market or products.	- If failing to impress users, clients and investors for some time, companies might fail to sustain themselves. - Certainly not OSS developers' favorites.

5.5 Macro R&D Infrastructure

Advantages	Disadvantages
- Can easily attract funds from government, global partners or commercial organisations if they meet a specialised area where there are high demands for both R&D and investment. - Promote collaboration and partnership, and organisations may merge together to form a power-house in a specialized area to attract more expertise and funding. - Can create spin-offs and generate more revenues and useful research results, particularly for bioscience or medical or e-Science R&D projects.	- Sustainability model is under development and is influenced by investors (which might in conflict with initial roadmaps). - Need to seek funding with regular intervals, and can create a sense of instability and insecurity at those periods. - Might be difficult to integrate academic theories and industrial perspective in some organisations.

6 Further Discussions

6.1 Mergers & Acquisitions: SuSE Novell

Mergers and acquisitions (M&A) are a useful business strategy and not explicitly an OSS business model, however, they may have a direct impacts on OSS organisations. A good example is SuSE, which was acquired by Novell [13] with US$210 million (£105 million) in November 2003. The acquisition helps Novell's ability to provide enterprise-class services and support on the Linux platform, and expand its business strategy to influence and generate revenues from open source community. To improve the business ecosystem, Novell SuSE launches partnership with its major vendors (AMD and IBM) and clients (ITV) and it is the first Linux vendor to join strategic alliance with Microsoft to ensure not only interoperability but also profit-making.

6.2 Licence revisit: Modified BSD Licence

We have discussed licencing issues in different part of this paper, and now revisit this subject to discuss the Modified BSD licence (new BSD) [3, 18], which is currently adopted by OMII-UK. In general, a licensee of Modified BSD software can: (a) use, copy and distribute the unmodified source or binary forms of the licenced program and (b) use, copy and distribute modified source or binary forms of the licenced program. This has to satisfy two conditions: (1) all distributed copies are accompanied by the licence and (2) the names of the previous contributors are not used to promote any modified versions without their written consent.

The simplicity of the BSD Licence can be seen as a great strength, but can also be seen as a weakness. For example a licensee can take software under the BSD licence and incorporate it into their closed source work. Another feature is that code licenced under new BSD can be relicenced under the GPL software. The original intension is seen as simple, customised and convenient for developers and OSS organisations, however this does not prevent competitors from borrowing, reusing and re-modifying codes for their own use, sales and re-branding. In the worst case, this could result in vicious circles in competitions, law suits or legal responsibilities.

Before any OSS organisations stepping into any of these business models, licence issues need to be clearly announced, reviewed and reinforced

through the governance structure and with legal advisors.

7. Conclusions

This paper has categorised several open source software (OSS) organisations into five business models: (a) Support Contracts; (b) Split Licensing; (c) Community; (d) Valued-added closed source; (e) Macro R&D Infrastructure. Case studies for each model have been discussed, explained and presented, each with a number of advantages and disadvantages. Based on our analysis, the long-term sustainability depends on (1) adopting the relevant business models, (2) securing funding or revenues and (3) reviewing the needs to move from one model to another or to use multiple business model. The business model that will lead to best long-term sustainability is also dependent on organisational needs, long-term goals, customer requirements and primary funding sources.

The initial phases of the UK e-Science Core Programme helped set up many e-Science organisations which are now facing the challenge of long-term sustainability. The Macro R&D Infrastructure and other business models presented in this paper are worth considering if setting up spin-offs from research projects, or setting up long-term sustained entities within the e-Science community.

8. References

[1] Apache Software Foundation Wikipedia http://en.wikipedia.org/wiki/Apache_Software_Foundation

[2] Apache Website http://www.apache.org

[3] BSD Licence Wikipedia http://en.wikipedia.org/wiki/BSD_license

[4] Edwards, K., "Epistemic Communities, Situated Learning and Open Source Software Development" Working Paper, Department of Manufacturing Engineering and Management, Technical University of Denmark.

[5] Forfas Business Paper, "Open Source Trends and Business Models: Guidelines for Key Irish Stakeholders", Forfas, July 2006.

[6] GNU General Public Licence (GPL) http://www.gnu.org/copyleft/gpl.html

[7] Graham S., "IDC Partner's Segmentation Model", IDC paper.

[8] Joint Information System Committee (JISC), "Open Source Briefing Paper", http://www.jisc.ac.uk/publications/publications/pub_ossbp.aspx

[9] Libre Software Paper, "Free Software / Open Source: Information Society Opportunities for Europe?", April 2000, http://eu.conecta.it/paper/

[10] MySQL Website http://www.mysql.com

[11] National Computer Systems (Singapore) website http://www.ncs.com.sg

[12] Netcraft Web Survey Report, "Open Source Parking Spoofing Headers to Benefit Apache", http://news.netcraft.com/archives/web_server_survey.html

[13] Novell Website http://www.novell.com

[14] OMII-UK Website http://www.omii.ac.uk

[15] OpenJDK Project https://openjdk.dev.java.net/

[16] Open Source, Wikipedia, http://en.wikipedia.org/wiki/Open-source

[17] Open Source Software, Wikipedia, http://en.wikipedia.org/wiki/Open-source_software

[18] OSS Watch, "The Modified BSD License - An Overview", http://www.oss-watch.ac.uk/resources/modbsd.xml

[19] Red Hat Website http://www.redhat.com

[20] XandrOS Website, https://www.xandros.com

[21] Xensource Website http://www.xensource.com