FOSS Governance And Collaboration: From A Good Idea To Coherent Market Approach

By Shane Coughlan[72]

Free and Open Source Software (sometimes called Open Source or FLOSS, and referred to in this paper by the commonly used term FOSS) is an approach to software that emphasises the freedoms provided to end users. Originally formulated in 1983 by a computer scientist concerned with access to technology, it has become a central component of mainstream IT. The popularity of FOSS has produced a wealth of related terminology and perspectives which occasionally lead to confusion about what it actually is and what are the best ways to engage with the field. This paper will address such confusion by providing a clear overview of FOSS, how it works, and why it is successful. It will go back to first principles in defining FOSS, explaining the concept of licensing that underpins it, and examining how this paradigm facilitates multiple development and business models. The key assertion is that the productive application of FOSS relies on good governance and active collaboration. While it is difficult to determine which precise governance model (or models) may be best suited to the long-term sustenance of FOSS as an approach to developing knowledge products, the indicators provided by the previous two decades

[72] Reprinted from Coughlan, S. (2011) Journal of Economics, No. 29 (Special Issue), The Facility of Law and Literature, Shimane University, Japan.

suggest that FOSS governance will continue to be effectively refined by its stakeholders.

Defining And Understanding FOSS

Free and Open Source Software (FOSS) is an approach to software that facilitates multiple development and business models. It is best characterised as a software paradigm. A software paradigm (also referred to as a software model) helps contextualise how stakeholders will create, distribute and/or use the software on computers. There are different software paradigms that compete for attention, investment and market-share in the modern business environment. The two predominant software paradigms are termed proprietary and FOSS, with the criteria for differentiation being based on the level of control over software that each facilitates. With proprietary software, control tends to lie primarily with the vendor, while with FOSS control tends to be weighted towards the end user.

The Origin Of FOSS

FOSS originated in the USA during the early 1980s. While in the early years of computer science it was common for people to share software relatively freely, the concept of selling software untied to physical hardware had begun to change this practice. What is termed the 'Software Industry' started in the early 1960s, and by the late 1970s it had grown significantly, due in no small part to the development of the personal computer in the mid-1970s and the rise of companies such as Microsoft.[73] The

[73] http://en.wikipedia.org/wiki/Software_industry

tension between those who wanted to share software technology and those who wanted to charge for access to software is illustrated by a letter Bill Gates wrote to the Homebrew Computer Club in 1976.[74] Entitled 'An Open Letter to Hobbyists', it charged that the practice of sharing code damaged the ability of people to produce good software.[75]

In 1983 Richard Stallman, an employee at MIT's Artificial Intelligence laboratory, decided to formalise the concepts behind the sharing of software technology. He founded a project to create a complete FOSS operating system that was compatible with Unix called the GNU Project.[76] This project also necessitated the development of terminology to describe how and why the FOSS paradigm worked.[77] In 1985 this emerging 'Free Software Movement' consolidated with Mr Stallman's establishment of the Free Software Foundation, the formal publisher and maintainer of the first and the most popular FOSS licences.[78]

The Definition Of FOSS

FOSS is not simply an aspiration to share software. It is a formally defined set of attributes applied to compliant software. The full definition of FOSS is hosted on the GNU Project website.[79] A concise overview is provided by Richard Stallman in his 2002 book, 'Free Software, Free Society':

[74] http://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists

[75] Ibid.

[76] http://www.gnu.org/

[77] http://en.wikipedia.org/wiki/Free_software

[78] http://www.fsf.org/

[79] http://www.gnu.org/philosophy/free-sw.html

> "The term “Free Software” is sometimes misunderstood—it has nothing to do with price. It is about freedom. Here, therefore, is the definition of Free Software: a program is Free Software, for you, a particular user, if:
>
> • You have the freedom to run the program, for any purpose.
>
> • You have the freedom to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)
>
> • You have the freedom to redistribute copies, either gratis or for a fee.
>
> • You have the freedom to distribute modified versions of the program, so that the community can benefit from your improvements." [80]

These four freedoms have been simplified in certain ways to illustrated the benefits of the approach. On the front page of the GNU Project website it suggests that “To understand the concept, you should think of “free” as in “free speech”, not as in “free beer”.”[81] Another is to shorten the four freedoms themselves into the form of 'use, study, share and improve.’[82]

[80] http://www.gnu.org/philosophy/fsfs/rms-essays.pdf (page 26)

[81] http://www.gnu.org/

[82] http://lwn.net/Articles/308594/

Challenges To Foss From Incumbent Market Interests

In 2000, Steve Ballmer, Chief Executive Office of Microsoft, famously likened FOSS to Communism.[83] Its advocates would counter that the FOSS movement is not and has never been a movement against the principles of financial gain nor is it inherently anti-corporate. Rather the opposite, in the sense that FOSS explicitly and purposefully allows commercial exploitation.[84]

This being said, Richard Stallman contends that key stakeholders in early software production were acting a way that he found unethical. He felt they were abusing their position and by doing so abusing the users of computers. But this assertion is less of an anti-market stance than an observation regarding inefficiency and control (given, of course, that we assume markets are intended to serve the majority participating rather than a narrow group who control supply and demand):

> "The modern computers of the era, such as the VAX or the 68020, had their own operating systems, but none of them were Free Software: you had to sign a non-disclosure agreement even to get an executable copy.
>
> This meant that the first step in using a computer was to promise not to help your neighbour. A cooperating community was forbidden. The rule made by the owners of proprietary software was, “If you share with your

[83] http://www.theregister.co.uk/2000/07/31/ms_ballmer_linux_is_communism/

[84] http://www.fsf.org/licensing/essays/selling.html

> neighbour, you are a pirate. If you want any changes, beg us to make them."

The idea that the proprietary-software social system—the system that says you are not allowed to share or change software —is antisocial, that it is unethical, that it is simply wrong, may come as a surprise to some readers. But what else could we say about a system based on dividing the public and keeping users helpless? Readers who find the idea surprising may have taken this proprietary-software social system as given, or judged it on the terms suggested by proprietary software businesses. Software publishers have worked long and hard to convince people that there is only one way to look at the issue." [85]

Stallman's issue could be described as what people now may term 'lock-in' and 'market distortion.' His perspective has since been validated in two critical ways, one being the recent spate of anti-trust cases and the other being the wholesale commercial adoption of FOSS precisely because it facilitates competition, market growth and the maximisation of investment.

Those involved in FOSS did not historically perceive it to be an extreme movement but rather to be a different to software from what an incumbent group of self-interested parties wished. Professor Laurence Lessig sums it up well with his introduction to 'FOSS, Free Society':

> "there are those who call Stallman's message too extreme. But extreme it is not. Indeed, in an obvious sense, Stallman's work is a simple translation of the freedoms that our tradition crafted in the world before code. "Free Software" would assure that the world governed by code is

[85] http://www.gnu.org/philosophy/fsfs/rms-essays.pdf, page 24.

> as “free” as our tradition that built the world before code."[86]

It is reasonable to suggest that some parties who were extremely worried about FOSS invested a lot of money and time trying to challenge its rise in the technology market. One reason for this is that FOSS as a paradigm presents a significant challenge to proprietary software as a paradigm. Proprietary software depends on charging per-copy licence fees to derive the majority of its profit while FOSS imposes no per-copy licence fees. The difference between the models can be worth millions of dollars in upfront fees.

Those working to challenge FOSS's credibility during its ascendancy to a market-leading position ultimately failed for a simple reason. FOSS is an approach to software that allows people to do a great deal with code. Some people - usually computer scientists like Richard Stallman - understood that FOSS was a good idea in its early days. Some people - perhaps those from portfolio management or sales backgrounds - took longer to understand the benefit. Nowadays all types of parties in all types of segments tend to see and derive some value from FOSS.

Understanding FOSS Means Understanding FOSS Licenses

The concept of FOSS describes a way to use, study, share and improve software, though this alone does not equate to providing the formal structure required for its potential to be realised. Stakeholders need to derive and maintain value regardless of their status of collaborators or competitors, and this leads us inevitably

[86] http://www.gnu.org/philosophy/fsfs/rms-essays.pdf, page 18.

to the common rules - rather than general concept - by which FOSS transactions are managed. These rules provide a framework that underpins the realisation of expectations in the field.

The goals of FOSS are realised through licences governed by copyright law. These licences take a different form compared to traditional proprietary documents. Instead of providing a narrow grant of use with a long list of exceptions and restrictions, they tend to provide a broad grant of use with few restrictions. But each license differers slightly in the grants it provides, and a common challenge for adopters of FOSS relates to what licence is beneficial for their situation.

FOSS licences are often divided into three categories by its advocates and users; non-Copyleft, weak-Copyleft and strong-Copyleft. Therefore Copyleft - while not inherent to Free Software - is perhaps the most important distinguishing features to categorise FOSS, and is one of the best places to start when one seeks to understand how such licenses work.

As with the definition of Free Software, Copyleft was first defined by Richard Stallman. He wanted to ensure that the GNU Operating System would be available to people with the four freedoms he had identified as being important, and he wanted to ensure this availability would continue in the mid-to-long-term.

> "The goal of GNU was to give users freedom, not just to be popular. So we needed to use distribution terms that would prevent GNU software from being turned into proprietary software. The method we use is called copyleft.
>
> Copyleft uses copyright law, but flips it over to serve the opposite of its usual purpose: instead of a means of privatising software, it becomes a means of keeping software free." [87]

[87] http://www.gnu.org/philosophy/fsfs/rms-essays.pdf, page 28.

Copyleft says that the freedoms provided with the software apply to all subsequent users of the software as well. Copyleft is not an inherent characteristic of FOSS, but rather a way of maintaining a set of grants applied to the software in question. This is a distinction sometimes overlooked by people new to FOSS, leading to confusion when encountering FOSS licences that provide the ability to use, study, share and improve code according to the formal definition of the Free Software Foundation, yet not containing Copyleft provisions.

Some would suggest that non-Copyleft licences are best because the cooperative model does not require formal statements of subsequent sharing.[88] Some maintain that they want an explicit Copyleft requirement applied to their code.[89] Some parties like the Free Software Foundation advocate the use of strong-Copyleft whenever possible.[90] Perhaps the most useful guide for adopters with a pragmatic perspective is popularity. The form of licence is used by over 50% of FOSS are strong-Copyleft licences such as the GNU GPL.[91] It is most notably used on the Linux kernel,[92] most of the GNU Project,[93] and other critical technologies like SAMBA.[94] This is probably because strong-Copyleft provides a very predictable and stable grant for the technology, allowing

[88] http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html

[89] http://www.freesoftwaremagazine.com/columns/why_i_choose_copyleft_for_my_projects

[90] http://www.fsf.org/licensing/licenses/why-not-lgpl.html

[91] http://www.BlackDucksoftware.com/oss

[92] http://en.wikipedia.org/wiki/Linux

[93] http://en.wikipedia.org/wiki/GNU

[94] http://us6.samba.org/samba/docs/GPL.html

multiple parties to cooperate in using and developing it over prolonged periods.

Understanding The Most Popular Foss License

The GNU GPL is a very popular FOSS licence, accounting for over half of the total FOSS licence use according to BlackDuck Software research.[95] The most widely used variant of the GPL is version 2 of the licence, though version 3 – released in 2007 – is becoming increasingly popular and has been adopted by major code projects like SAMBA.[96] It was created to encapsulate the four freedoms applied to FOSS as effectively as possible for current and future users, and for this reason it is also a strong-Copyleft FOSS licence. Its purpose has never been otherwise, as Stallman's description of its origin attests:

> "The specific implementation of copyleft that we use for most GNU software is the GNU General Public License, or GNU GPL for short." [97]

Some parties have taken issue with the way that the GPL contains a preamble that explains its originals and purpose, and that this makes it a political manifesto as well as a legal document.[98] But one could equally argue the preamble is measured and makes clear what the document is, as evidenced by - for instance – its first paragraph in version two of the licence:

[95] http://www.BlackDucksoftware.com/oss

[96] http://news.samba.org/announcements/samba_gplv3/

[97] http://www.gnu.org/philosophy/fsfs/rms-essays.pdf, page 29.

[98] http://www.netc.org/openoptions/background/history.html

"The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change FOSS--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too."[99]

While there is little doubt that organisations such as the Free Software Foundation have a political agenda, FOSS licences such as the GPL are no more impacted by this then the licences of proprietary companies are impacted by those parties having a financial interest in the market. The aims of issuing entities and the inherent validity of the licenses they issue are two different matters.

As FOSS grew into a mainstream approach in IT, questions were raised about whether the primary licence used, the GNU GPL, was actually valid.[100] These questions suggested that the model applied by FOSS was not something that worked in copyright law, and were immediately contested by essays produced by figures central to FOSS development.[101] Later they were contested more substantially through court cases against parties infringing the GPL licence in Europe.[102] These cases resulted in court victories, and were followed by events in the

[99] http://www.gnu.org/licenses/gpl-2.0.html

[100] See for example Andrés Guadamuz (2004) '*Viral Contracts or Unenforceable Documents? Contractual Validity of Copyleft Licenses*', E.I.P.R. Vol. 26, Issue 8, pp.331-339. Also online at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=569101

[101] http://www.gnu.org/philosophy/enforcing-gpl.html

[102] http://gpl-violations.org/news/20040519-iptables-sitecom.html

USA that further validated the licensing approach[103] and its effectiveness in being applied to commercial transactions.[104]

Today there is little doubt the GPL is a valid legal document. Version 2 is well-entrenched in the market, and the growing use of version 3 has occurred despite some criticism of the document while it was being drafted.[105] This may be indicative that such criticism, as with criticism directed at earlier versions of the GPL or at FOSS itself, was largely unfounded. It is also possible to suggest that criticism of the GPL provoked responses, elaboration and clarification that contributed to maturing the licence, and perhaps the paradigm as a whole.

The Governance Of FOSS

The Internet has allowed people to communicate and to work together across great distances at a lower cost and at a higher speed than ever before. It has been a powerful driver in reducing barriers to working with partners and customers to accomplish goals, what is sometimes referred to as co-innovation.[106] In the software field it is difficult for a single vendor to meet all the requirements of multiple customers, and it is more effective for several parties to cooperate on developing and enhancing a shared platform. This is what increasingly happens, and it has lead to the commercial sustainability of FOSS projects such as the Linux

[103] http://www.fsf.org/news/wallace-vs-fsf

[104] http://www.softwarefreedom.org/news/2007/oct/30/busybox-monsoon-settlement/

[105] http://www.eweek.com/c/a/Linux-and-Open-Source/Latest-Draft-of-GPL-3-Comes-Under-Fire/

[106] http://theotherthomasotter.wordpress.com/2007/05/03/co-innovation-is-a-strength-not-a-weakness/

kernel.[107] This is because FOSS, a software paradigm built on the inherent assumption of cooperation and sharing, is a natural beneficiary of the global trend towards increased cooperation.

One good example is the Linux kernel, which started as a student project,[108] and has grown into the core of an operating system used in a wide variety of fields with financial backing from companies like Fujitsu, Hitachi, HP, IBM, Intel, NEC, Novell, and Oracle.[109] Linux is GPL software designed to run on many types of computer, and it is developed through a world-wide cooperative project on the Internet.[110] Given its scale and success, it provides an excellent example of co-innovative development inside the FOSS paradigm. It is structured into teams with leaders who consolidate work, and a handful of key developers that then combine the components into the final product. There is a relatively low barrier to entry regarding participation in development, and each individual stakeholder will have their own reasons for investing in the project. What is noticeable is that the collective output of the parties collaborating is stable, reliable and widely used in critical industries.

Cooperation As A Lasting Mechanism For Change

Cooperation in creating software has profound implications for development models and the management of processes, and

[107] http://linuxfoundation.org/en/Members

[108] http://www.linux.org/people/linus_post.html

[109] http://linuxfoundation.org/en/FAQ#Who_are_members_of_the_Linux_Foundation.3F

[110] http://www.kernel.org/

has expanded far beyond the concept of working with a small, select group of similar companies. That was a template of interaction tied to the Industrial Revolution, and appears archaic in a world where instant communication allows an individual in Shenyang, China to work as effectively as one in London. Modern cooperation requires the broad sharing of information and tools without delay between multiple parties and even legal entities, with an emphasis on reducing access time further to optimise the benefit of cooperation. An increasing number of formal models have been emerging to facilitate this, with one example being 'Agile software development,' which places emphasis on the feedback provided by creative participants to guide further development.[111]

Such cooperative development is arguably permanent for two reasons, one systemic and one market-based. From the systemic perspective, the reduction of barriers and costs to cooperation have lead to a self-sustaining cycle where new development models have emerged that increase the efficiency of cooperation, and in turn foster further optimisation and investment in such activity. From a market perspective, users are requiring more complex and interconnected software, and without unlimited engineering resources, the most efficient way to produce such software is through building shared platforms with other market participants.

The dynamics of the software industry have altered in the last two decades. Twenty years ago the dominant proprietary paradigm resulted in a small number of providers controlling innovation and serving a large number of users in a fairly static relationship. However, the emerging FOSS paradigm encouraged new development models and new software development processes that moved the decision-making emphasis to users.

[111] http://en.wikipedia.org/wiki/Agile_software_development

Since the FOSS paradigm gained mainstream traction this has had a profound effect on the market as a whole. Increased user involvement in consultation, design, testing and improvement is noticeable in every approach to software today. One consequence of this has been to blur the distinction between what constitutes a user and what constitutes a provider. FOSS notably empowers all users to become providers at any time of their choosing.

The Many Development And Business Models Of FOSS

The proprietary software and FOSS paradigms facilitate the establishment and improvement of various software development models and processes. These development models may be hierarchical, loosely managed or unstructured depending on the given software paradigm and the requirements of the individuals or organisations working on a project. It would be incorrect to associate FOSS exclusively with one development or business model, though new observers or entrants to the FOSS market occasionally do so. This is perhaps a result of limiting their research to a narrow range of case-studies or usage models.

Such misconceptions are partly attributable to an essay by Eric Raymond circulated in 1997 entitled “The Cathedral and the Bazaar,”[112] and extended into a book published by O'Reilly Media in 1999.[113] The proposition that "given enough eyeballs, all bugs are shallow" appeared to suggest that the limited, hierarchical and restricted world of proprietary commercial software ultimately could not compete with the broad, dynamic and more bazaar-like

[112] http://www.catb.org/~esr/writings/cathedral-bazaar/

[113] http://oreilly.com/catalog/9780596001315/

world of FOSS. However, it should be understood that Mr. Raymond's paper was not originally a comparison of the FOSS development methodology versus a proprietary development methodology. It was a criticism of hierarchical structures applied by the GNU Project (a FOSS project) versus the more flat management structure of the Linux Project (a FOSS project).[114]

Misunderstandings regarding the organisation and management of FOSS are not isolated to development models. From the perspective of the traditional proprietary software world it can be difficult to understand the approach taken with FOSS, and some parties have questioned its validity as a commercial approach.[115] However, concern with regards viable business models and FOSS tend to arise when parties have a preconception that per-unit licence costs are an inherent requirement to qualify as commercial software. While FOSS allows a wealth of business models to be applied, per-unit licensing costs is not one of them.

Per-unit revenue models would either have to prevent sharing of code to maximise their market and thus undermine one of the four freedoms defined by the Free Software Foundation, or they would be circumvented by users who would have a choice of paying the originator for a copy of the software or getting one from a third-party without cost.

There are many business models applicable to FOSS for the same reason that FOSS facilitates multiple development models; this paradigm provides a broad range of parameters that participants operate inside. Examples of FOSS business models include:

[114] http://www.alamut.com/subj/economics/misc/cathedral.html

[115] http://business.timesonline.co.uk/tol/business/industry_sectors/technology/article733264.ece

- Development-related services to produce specialised products, such as bespoke product customisation for enterprises.
- Integration-related services to ensure that products work with existing systems, such as in Enterprise intranets, SME office networks and banking communication systems.
- Support-related services to maintain deployed solutions, particularly in the SME, governmental and enterprises sphere.
- Software as a Service to deliver application functionality over a network, such as in Web 2.0 companies or search companies like Google.
- Cloud computing to deliver processing functionality over a network, such as those provided by companies like Sun Microsystems.
- Mixed-models combining FOSS and proprietary software, such as the product offerings from Oracle with GNU/Linux and their proprietary enterprise database running on top.
- Dual-licensing models where code is released under both a FOSS and a proprietary licence.

The most common FOSS business models in the server and workstation market segment tend to be support provision across multiple products (i.e. like IBM)[116] or support provision for a branded family of products (i.e. like Red Hat).[117] While dual-licensing used to be relatively common, the best known companies such as MySQL and Trolltech did not scale beyond being multi-million dollar enterprises and were instead acquired by multi-billion dollar corporations. Since then the visible side of

[116] http://www-03.ibm.com/linux/

[117] http://www.redhat.com/

their business has tended to be focused on the FOSS element of the product offering rather than the proprietary.

Embedded companies (those that make telephones, routers and other small computing devices) now frequently make use of FOSS. The business models applied tend towards mixed-model, with a FOSS platform being used to provide basic services, and perhaps a proprietary series of components to provide a differentiator. The LiMo Foundation's work in the mobile sphere[118] or MontaVista's products in the embedded networking sphere provide examples of this.[119]

In network services there are a great variety of companies using FOSS. Most notable is perhaps Google, which uses FOSS-based technologies to power its infrastructure, and makes a modified FOSS operating system available for its employees workstations.[120] Because Google primarily provides network services, rather than focusing on the distribution of software, the use of FOSS has very little impact on their business model except to reduce costs, and their modifications to FOSS code do not generally have to be distributed. This has come under some criticism as effectively using FOSS without fully participating in the paradigm.[121] However, regardless of what one thinks of their use of the code, Google's business model has proven highly successful. In essence, they used FOSS to facilitate infrastructure that would have cost billions to build as proprietary software for a far smaller sum, and they leveraged this advantage to provide services above the traditional limits of their corporate scale and funding.

[118] http://www.limofoundation.org/

[119] http://www.mvista.com/

[120] http://en.wikipedia.org/wiki/Google_platform

[121] http://ostatic.com/blog/google-touts-open-source-cred

Ultimately the number of possible business models applicable to FOSS make it impossible to pick out any one as a clear favourite. As with any field of business, the correct model depends on market segment analysis, an understanding of skills, and a prudent balance between maximisation of profit and sustainability. There is no 'FOSS business model' in the singular sense; the licences used in the field provide broad grants that foster a wide range of approaches.

Understanding The Governance Of FOSS

The early governance of FOSS was understandably centred on the licenses that govern FOSS transactions. There was a narrow focus on compliance because it was regarded as the critical issue for minimising risk in adoption and deployment, and that was the critical issue facing early users. However, as the stakeholders in the field became more sophisticated, so too did their approach to governance, and this lead to a shift in perspective towards understanding governance as a tool to maximise value while honouring obligations. This is a related but wider concept than reducing risk.

For early adopters of FOSS the most common problems encountered can be summarised as having their roots in two key issues; people didn’t read the licenses properly, or they read them but didn’t follow the terms. The solution to these problems were equally simple; people had to read the licenses in question and follow their terms. Nevertheless, new adopters frequently encountered issues, with some notable cases being GPL-violations.org versus Sitecom,[122] GPL-violations.org versus D-Link[123] and SFLC versus 14 companies.[124] A lack of understanding or a lack of process maturity can generally be proposed as a reasonable explanation for these occurrences.

As FOSS stakeholders became more understanding of how FOSS provides value - namely through collaboration between an ever-changing pool of third parties - they also became more nuanced in their understanding of the governance necessary to provide maximum benefit. This resulted in a shift from policy in the form of lists of accepted or rejected licenses, code or deployment approaches towards more nuanced processes that provided the flexibility to adopt new technology and adapt to changes in licensing or market demands. This tended to be intertwined with the evolution of participants in how they approach the field as a whole. If one understands the value of FOSS to be found in the collaborative energy centred around common frameworks, then stakeholder maturity will see an increasing shift from relative isolation as an entity to collaboration as a participant in a community.

While early FOSS governance used to be focused on understanding licenses as obligations, the mature governance of FOSS is about the questions that lifecycle management raises, namely “what type of code do you use and why?”, “how do you manage change to ensure continual improvement?”, “how do you ensure your obligations are met?” and “how is this applied through the supply chain from inception to end-of-life for each product or solution involved?” Stakeholders become more active

[122] http://gpl-violations.org/news/20040415-iptables.html

[123] http://www.jbb.de/judgment_dc_frankfurt_gpl.pdf

[124] http://www.linux-magazine.com/Online/News/SFLC-Files-Lawsuit-Against-14-Companies-for-GPL-Violations

buying or developing the processes to manage code, training people internally to obtain value while minimising risk, and doing the same for the supply chain on which they depend. This is a natural consequence of seeking to maximise potential through shared rules to improve collaboration.

The Emergence Of Market Solutions

There are many services, products and collaborative platforms that contribute to governance in the FOSS marketplace. None solves every challenge that the paradigm can raise, but many deliver utility to new entrants and experienced stakeholders alike, providing avenues for minimising risk, improving efficiency and dealing with suppliers or customers. One example is FOSSBazaar, a community for sharing governance data that was initiated by HP via the Linux Foundation, and which continues to over a broad range of general material and commentary today.[125] Others include comprehensive commercial solutions that have appeared from companies like BlackDuck Software[126] and OpenLogic[127] that deliver lifecycle management, the non-profit Linux Foundation compliance programme,[128] and independent FOSS projects like the Binary Analysis Tool.[129]

Collaboration is key to deriving value from FOSS and sustaining it through the development, deployment and support of products or solutions. This is not about code; the collaboration

[125] http://www.fossbazaar.org/

[126] http://www.blackducksoftware.com/

[127] http://www.openlogic.com/index2.php

[128] http://www.linuxfoundation.org/programs/legal/compliance

[129] http://www.binaryanalysis.org/en/home

that provides value is not limited to software, but is instead applicable to the approach required to obtain value in the modern market. It translates into platform management, and requires managers, programmers and legal experts to collaborate across organisational boundaries and nation borders.

Interaction and cooperation around stakeholders is far more than an aspiration or ad hoc arrangement in the increasingly mature FOSS-related economy. One example is that the Linux Foundation helps stakeholders collaborate around Linux in the US, Europe and Asia by organising meetings, working groups and conferences to encourage shared understanding and knowledge sharing.[130] Another is the European Legal Network, an invitation-based effort facilitated by Free Software Foundation Europe that helps 280 stakeholders collaborate across 4 continents, and which runs private mailing lists, special interest groups and conferences to share knowledge.[131]

It is impractical to attempt to list the degree to which collaboration - or crowd-sourcing - has permeated the global FOSS economy, but a cursory examination of the Asia-Pacific region is illustrative of how initiatives like the Linux Foundation and the European Legal Network are far from isolated. In Japan collaborative activities are organised by the government via the IPA[132] and the industry via Linux Foundation Japan,[133] while regional organisations like Ruby City Matsue[134] have also fostered

[130] http://www.linuxfoundation.org/

[131] http://fsfe.org/projects/ftf/network.en.html

[132] http://www.ipa.go.jp/software/open/ossc/index.html

[133] http://www.linuxfoundation.jp/

[134] http://www1.city.matsue.shimane.jp/sangyoushinkou/open/rubycitymatsue/ruby_city_projecting.html

enough momentum to host international conferences.[135] In Taiwan, the Open Source Software Foundry gives support and legal advice to help companies use FOSS,[136] and a new legal network modelled on the European Legal Network is also being prepared for launch.[137] In Korea, NIPA is collaborating with KOSS Law Center and FSFE to develop governance activities,[138] with tangible outcomes including the creation of a national legal network and the launch of a new international conference to share knowledge.[139]

Increased Governance And Collaboration Is Driven By The Market

Software is a knowledge product and FOSS is a management approach for this product. FOSS requires effective governance and collaboration to create maximum value for every stakeholder regardless of their individual product range or market segment. The required degree of cooperation is appearing in national, regional and global markets with a growing amount of shared structure visible due to the interconnected nature of the industry. The outstanding question is probably whether this trend will continue or some form of market pressure - be it litigation or alternative methods of deriving value from software proving more

[135] http://www.rubyworld-conf.org/en/

[136] http://www.openfoundry.org/

[137] http://osln.tw/doku.php?id=open_source_legal_network_taiwan

[138] http://www.oss.kr/7065

[139] http://www.kosslaw.or.kr/conference/conference01.php

attractive - will lead to a lack of long-term coherent governance for the field.

This is more than an idle question. Research by Gartner previously suggested that 85% of enterprises are already using FOSS in one capacity or another, and the remaining 15% expect to use it within twelve months of the survey.[140] These figures the type of market penetration figures previously suggested by UNU Merit, when in their 2007 report for the European Commission they suggested that "FLOSS-related services could reach a 32% share of all IT services by 2010, and the FLOSS-related share of the economy could reach 4% of European GDP by 2010."[141] Research shows no indication that the growth of FOSS will slow at any point in the near future, given fair market access.

This last point may prove to be crucially important. If competition drives innovation and provides an impartial method of determining the success or failure of product or business models, then it is important for fair and equitable competition to be fostered in markets regardless of the particular approach chosen by participants. It follows that access to information regarding interoperability and interaction between software components is therefore a key requirement in the modern IT market to foster such competition. Conversely, if fair access is not provided, then competitive paradigms like FOSS may be hindered in terms of future market penetration and opportunities despite their potential utility.

FOSS And Standardisation

[140] http://www.gartner.com/it/page.jsp?id=801412

[141] http://flossimpact.eu/

FOSS and standardisation is an area that has drawn increased interest in recent years, not least due to the challenges FOSS faces with regards market access and the ability to compete fairly (for a given value of fairly) against older and more established approaches to organising the creation, distribution and support of software knowledge products. This is best exemplified by the public debate over what became known as MS-OOXML, a next generation document format. It was suggested that the process was biased[142] and that the grants provided for the proposed standard were insufficient for FOSS.[143]

A great deal of the discussion surrounding standardisation and FOSS centred on patents. The reason for this are the are fundamentally different goals for patents and standards, as illustrated by Mr Karsten Meinhold, chairman of the ETSI IPR Special Committee, when he stated that "IPRs and Standards serve different purposes: IPRs are destined for private exclusive use, standards are intended for public, collective use."[144] FOSS, being also designed for public, collective use, tends not to fall into the normal categorisation of how IPR is positioned.

Patents in standards had previously been managed by grants such as RAND, and these were considered sufficient for proprietary software. However, that did not mean that such conditions facilitated fair market access and competition for all software paradigms competing in the market. For example, per-unit royalty payments would compromise the freedom of people to share the code, as would terms that did not permit sub-licensing.

[142] http://www.linuxjournal.com/node/1000294

[143] http://arstechnica.com/software/news/2008/03/sflc-ooxml-could-poses-patent-threat-to-gpl-licensed-software.ars

[144] http://ec.europa.eu/enterprise/ict/policy/standards/ws08ipr/presentations/21meinhold_en.pdf

Indeed, several FOSS licences have provisions regarding issues like patents to ensure that the four freedoms defined by the copyright licence continue in full to all subsequent users. The GPL is an example of such a licence, and others with the same or similar provisions actually make up the majority of the FOSS paradigm. For example, according to BlackDuck Software research 66.57% of projects use GPL family licences that explicitly prohibit the application of patent restrictions on covered software.[145] Excluding these licences from a standard would mean excluding 2/3 of the FOSS model participants from accessing that standard. That is quite a challenge for FOSS and for markets that seek to be open, competitive and genuinely innovative, though debate still continues regarding the best way to address the matter.

Globalisation And FOSS

Globalisation refers to the process of national economies becoming more open, economics becoming more ‘global’ than ‘national’, and to the reduction of national controls over economic matters.[146] In effect, changing the world from a loose organisation of states into a single giant canvas, and providing new opportunities for people to work together. This concept has profound implications for cooperative models of innovation and production, though it is not without its detractors.

The proposition of emerging Globalisation is contested by ‘globalisation scepticism’, a view summed up by Hirst and Thompson’s comment that “the closer we looked the shallower and more unfounded became the claims of the more radical

[145] http://www.BlackDucksoftware.com/oss

[146] Ramesh Mishra, *Globalisation and the Welfare State,* (Cheltenham: Edward Elgar, 1999), 3-4.

advocates of economic globalisation."[147] For these sceptics there are international economies but there is no evidence for a truly 'global' economy.[148] This is a valid criticism within the constraints of their definition, but it can equally be contested as relevant only in the context of purely economic, rather than cultural or communicative Globalisation.

Whether one defines Globalisation as an example of increasing global capitalism or as a deeper and more complex mix of political, cultural and financial connections, it suggests that the world is not merely a collection of states with limited communication and sharing potential. From that perspective, and therefore from the perspective of technology and business, it does not matter whether globalisation is a trend towards a global economy or a collection of increasingly interlinked international economies. Knowledge, goods and people are far more mobile now than ever before. Software, a technology that can be easily transferred through communication networks, is one of the greatest beneficiaries of this development. It follows that the current success of FOSS may therefore be partly explicable as a product of such Globalisation, and that it will inevitably continue to expand as long as the trend towards increased human interaction continues.

In this context it is worth noting that the concepts behind FOSS have blazed a trail in developing the norms required for massively distributed collaboration, and they have proven to be influential beyond the field of technology. A key example is that when Professor Laurence Lessig established the Creative Commons, and in doing so formalised an approach to foster

[147] Paul Hirst and Grahame Thompson, *Globalization in Question: The International Economy and the Possibilities*, 2nd ed. (Cambridge: Polity Press, 1999), p2.

[148] Ibid, p16.

increased engagement and exchange around cultural artefacts, he drew heavily on the concepts behind FOSS licenses. In the introduction to Lessig's primary book on cultural sharing, 'Free Culture,' he acknowledges that his insights do not exist in isolation, and states:

> "The inspiration for the title and for much of the argument of this book comes from the work of Richard Stallman and the Free Software Foundation. Indeed, as I reread Stallman's own work, especially the essays in Free Software, Free Society, I realise that all of the theoretical insights I develop here are insights Stallman described decades ago."[149]

Conclusion

FOSS is an approach to software that emphasises the freedoms provided to end users, with a particular focus on the ability of participants to use, study, share and improve technology. While occasionally misunderstood as being non-commercial, FOSS has always been conceptualised as something that allows commercial activity. It is framed by its licences, which range from providing a simple, non-perpetual grant of the receiving user freedom (as with the Modified BSD licence) through to providing such freedom in perpetuity via Copyleft and addressing issues such as patents (as with the GPL). While still relatively new, most concerns related to this approach to licensing have been substantially addressed in courts of law, in industry usage and in common understanding over the licence terms and their intent. Today FOSS has become a central component of mainstream IT.

[149] http://www.free-culture.cc/freecontent/, page 14.

The popularity of FOSS has produced a wealth of related terminology and perspectives, and this occasionally leads to some degree of confusion or misunderstanding. To address this it is necessary to go back to first principles in defining FOSS, understanding the concept of licensing that underpins it, and examining how it facilitates multiple development and business models. This leads to a number of useful observations. The first is that FOSS is a paradigm that facilitates a multitude of development and business models, barring only those inherently tied to the concept of per-unit software licence fees. The second is that FOSS benefits from globalisation, especially in the context of increasing long-distance cooperation facilitated by the Internet. This applies equally whether one is concerned with cooperation between like-minded professionals or with blurring the distinction between a developer and a user of technology. The third is that the licences that appear to best support this diversity of choice are those that provide both common rules for interaction (i.e. terms of using, studying, sharing and improving) while also delivering a mechanism for sustaining these rules for subsequent users (i.e. strong-Copyleft licences such as the GNU GPL).

Once FOSS is understood as a method of deriving value from knowledge products with an emphasis on collaboration, it naturally follows that its productive application depends on good governance and active collaboration. This type of management structure has been gradually developed by stakeholders using the same methods applied to the creation of creation of FOSS knowledge products. though it is worth noting that the current mechanisms does not fully explain how FOSS potential can be continually realised by an increasingly diverse eco-system of stakeholders. While it is evident that FOSS governance is increasingly sophisticated, it is equally evident that understanding which model is best suited to the long-term management of software is far from trivial in a world with complex supply chains, products deployed across a multitude of legal jurisdictions, and a vast array of stakeholders with a multitude of development, deployment and business models.

However, despite such difficulty in determining which precise governance model (or models) may be best suited to the long-term sustenance of FOSS, the indicators provided by the previous two decades suggest that its management will continue to be effectively refined by stakeholders. FOSS is well positioned because it facilitates sharing and cooperation in a world where such activities tend to easier, cheaper and more effective than ever before. It is therefore reasonable to assert that FOSS will continue to benefit from and drive increased openness and interoperability in the technology market for pragmatic reasons.

In conclusion, as the concepts underlying FOSS are applied to other creative works such as text, music or images, mainstream acceptance of this approach to developing and maintaining knowledge products will increase. Its governance - and therefore sustainability - will be refined as it scales, and any issues will gradually be worked out due to stakeholder requirements and market dynamics.

Shane Coughlan is an expert in communication methods and business development. He is well known for building bridges between commercial and non-commercial stakeholders in the technology sector. His professional accomplishments include establishing a legal department for the primary NGO promoting Free Software in Europe, building a professional network of over 270 legal counsel and technical experts across 4 continents, and aligning corporate and community interests to launch both the first law journal and first legal book dedicated to Free/Open Source Software. He has been part of the OIN licensing team since November 2009 and has lead the team since August 2013.

Shane has extensive knowledge of Internet technologies, management best practice, community building and Free/Open Source Software. His experience includes engagement with the server, desktop, embedded and mobile telecommunication industries. He does business in Europe, Asia and the Americas, and maintains a broad network of contacts.